UNIFIED
COMMUNICATIONS
SYSTEM

VITALPBX 4.0

COMPLETE GUIDE

BY JOSEPH MONTES
REVISION: 1

VitalPBX 4.0 -
Complete Guide

Table of Contents

Introduction

Hello, and welcome to a complete guide on how to use and configure VitalPBX, completely from scratch.

Throughout this set of lessons, you will be able to understand and configure VitalPBX from the ground up and become a VitalPBX expert.

We will go over every module that VitalPBX has to offer. We will also cover our different add-ons, troubleshooting scenarios, and many tools that will help you with your VitalPBX installation.

By the end, you will have a fully functioning PBX system full of capabilities for your business and your customers.

So, without any further ado, let's begin.

SECTION 1 - INSTALLATION

Introduction

VitalPBX can be installed in many different environments. Here, we will cover every one of them, and you will see that the process is very straightforward.

With Version 4, we have transitioned to using **Debian** as our base operating system. So, if you are familiar with this environment, it will be even easier for you.

You can install VitalPBX on dedicated hardware, virtual machines, or VPS service providers. Like **Amazon AWS**, **Google Cloud**, and **Digital Ocean**, among others.

We have an ISO image available for easy installation, or you can begin with a minimal installation of Debian and use our installation script.

Let's take a look at the first installation environment.

System Specifications

VitalPBX is very scalable in terms of the specifications required for installation. Depending on the number of concurrent calls you wish to have, you can determine the specifications for your server. We will go more in-depth on this subject in a later lesson. For now, the minimum server specifications are

- **2 Core CPU at 2.4 GHz.**
- **4GB of Memory**
- **50GB of Storage.**

This will allow you to have up to 50 extensions with 12 concurrent calls.

An internet connection is **required** to install VitalPBX fully and with its latest version.

Section 1, Lesson 1 - Virtual Machine Installations

Installing VitalPBX in a virtual environment is simple. You will need the following:

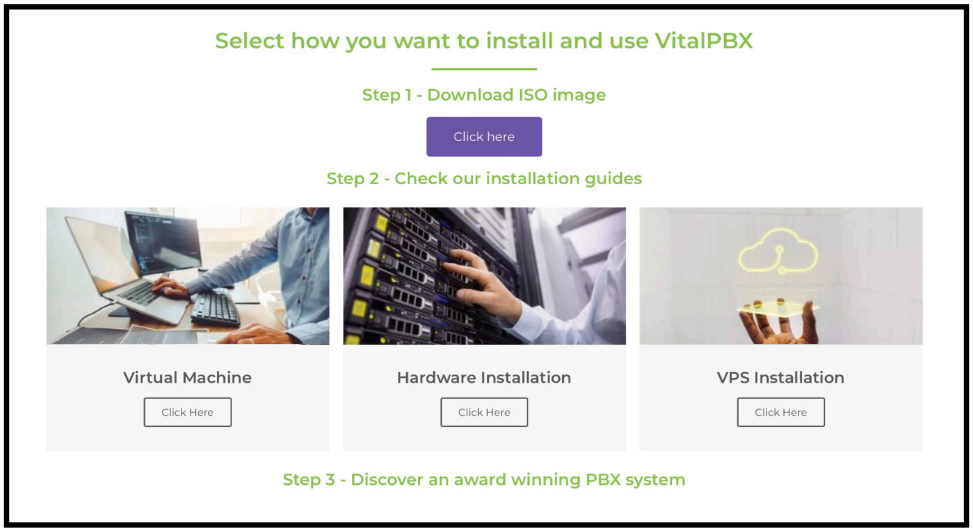
- A Virtual Machine capable computer.
- Virtualization Software. This can be **VM Ware**, **Oracle Virtual Box**, **ProxMox**, or any other **KVM-based virtualizer**.
- **VitalPBX's official ISO image** or a **Debian minimal installation ISO image**.
- An internet connection.

This installation process is the basis for hardware or a Debian minimal installation. So, you can refer to these steps if you are installing these other types.

Concerning system specifications, this will depend on the number of concurrent calls you wish to have. Again, we will go through how you can determine this in a later lesson. For now, use the minimum server specifications detailed in the previous section.

The Virtual Machine creation process will vary based on the virtualizer of your choice. In this lesson, we will be using **Oracle's Virtual Box**. It is free to use and accessible for most systems. We will cover some special occasions at the end of this section.

First, we will need to download the ISO from VitalPBX's official website at https://www.vitalpbx.com. This can be found under the **Download section** and by scrolling down.



Select how you want to install and use VitalPBX

Step 1 - Download ISO image

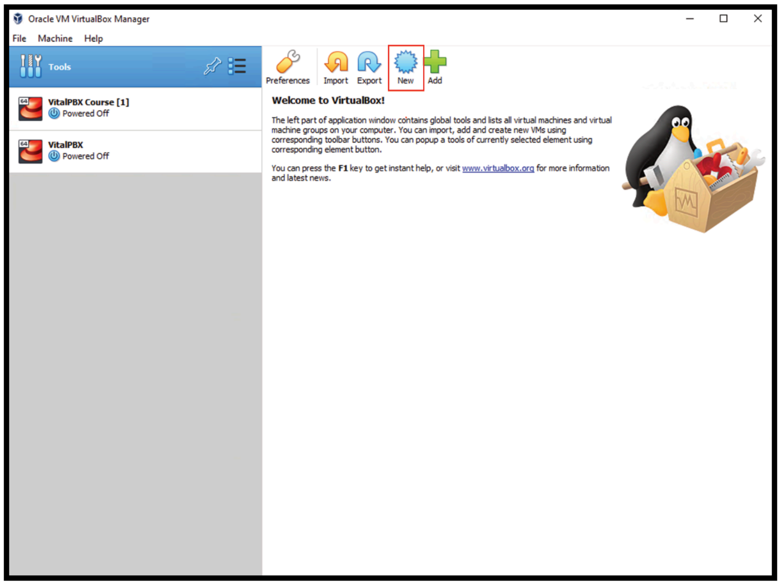
Step 2 - Check our installation guides

Step 3 - Discover an award winning PBX system

Image 1.1.1 - VitalPBX download section.

There, you will also find written guides on how to install VitalPBX in different environments.

Next, we will create a new virtual machine. For this, we will start Oracle's Virtual Box. Then, click on the **New** button.



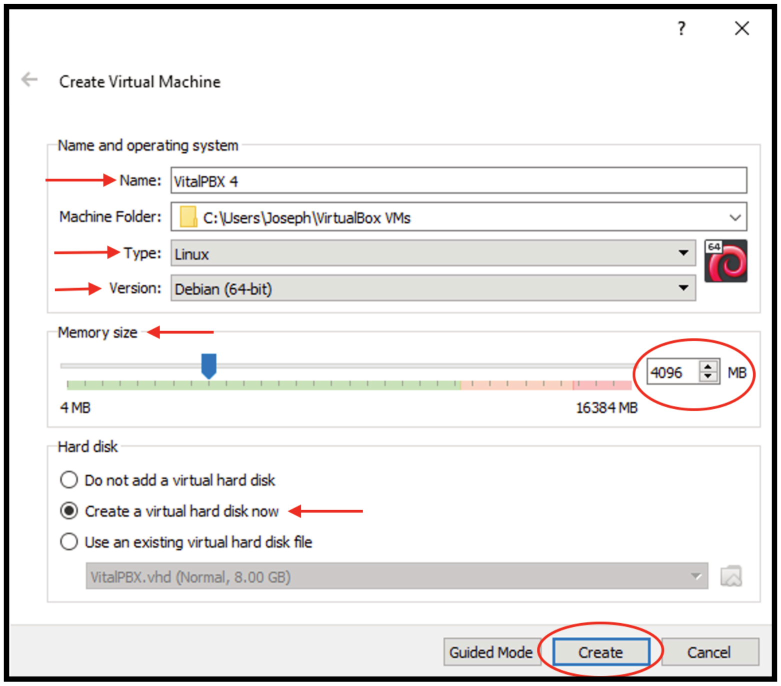
Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

Image 1.1.2 - Virtual Box new virtual machine.

Here, enter a name for your virtual machine. Then, select the type of virtual machine as **Linux** and the version as **Debian 64-bit**. Set the **Memory Size** as **4096 MB**. For **Hard Disk**, select **Create a virtual hard disk now**. Afterward, click on **Create**.



Create Virtual Machine

Name and operating system

Name: VitalPBX 4

Machine Folder: C:\Users\Joseph\VirtualBox VMs

Type: Linux

Version: Debian (64-bit)

Memory size

4096 MB

Hard disk

○ Do not add a virtual hard disk

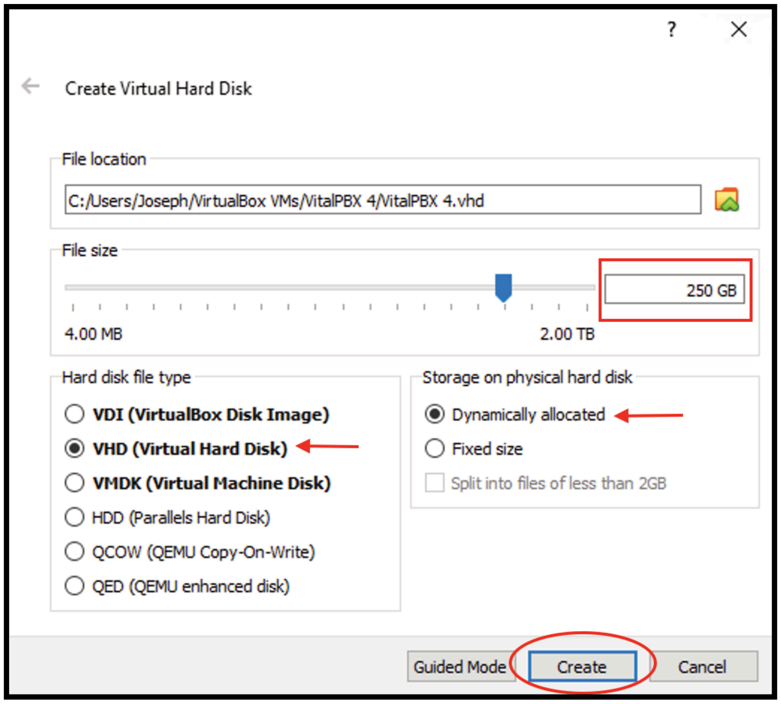
● Create a virtual hard disk now

○ Use an existing virtual hard disk file

VitalPBX.vhd (Normal, 8.00 GB)

Image 1.1.3 - Virtual Box, create virtual machine.

Next, you will be presented with the “Create Virtual Hard Disk” window. Here, enter the **hard disk size** of your choice. For **Hard disk file type**, select **Virtual Hard Disk**. For “Storage on physical hard disk,” choose **Dynamically Allocated**. Finally, click on **Create**. By dynamically allocating the storage, we don’t use the full size we entered immediately.



Create Virtual Hard Disk

File location: C:/Users/Joseph/VirtualBox VMs/VitalPBX 4/VitalPBX 4.vhd

File size: 250 GB

Hard disk file type:
- ○ VDI (VirtualBox Disk Image)
- ● VHD (Virtual Hard Disk)
- ○ VMDK (Virtual Machine Disk)
- ○ HDD (Parallels Hard Disk)
- ○ QCOW (QEMU Copy-On-Write)
- ○ QED (QEMU enhanced disk)

Storage on physical hard disk:
- ● Dynamically allocated
- ○ Fixed size
- □ Split into files of less than 2GB

Image 1.1.4 - Virtual Box, create virtual hard disk.

You will now have a virtual machine created. Next, we will need to configure this virtual machine to proceed with the installation. Right-click on the new virtual machine and select **settings**.

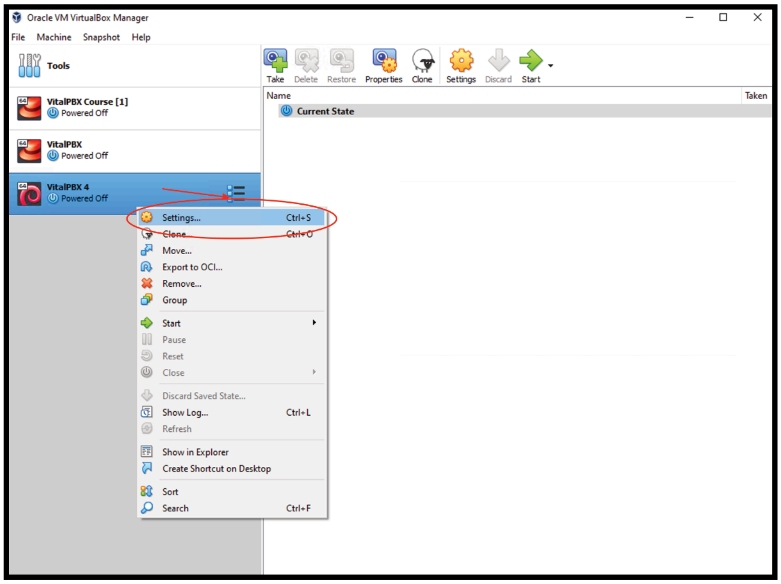
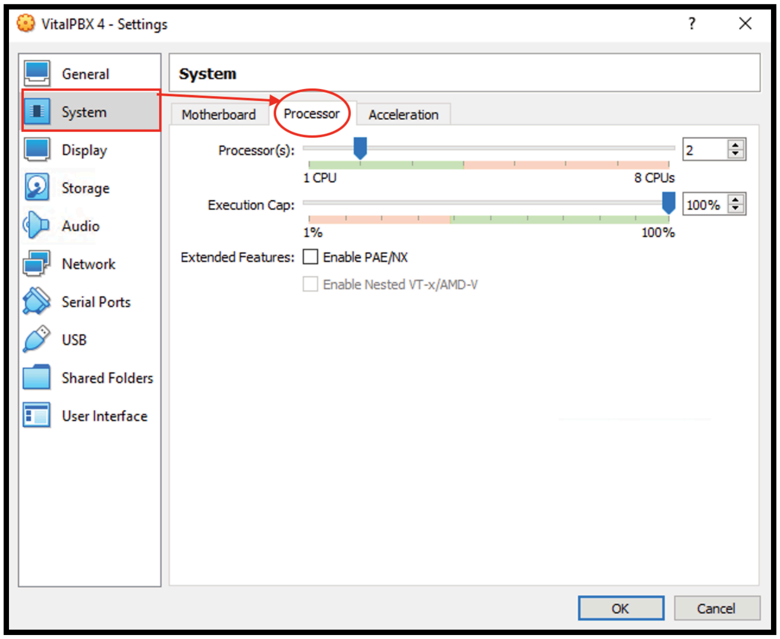


Image 1.1.5 - Virtual Box, settings option for the virtual machine.

Click on **System** and then on **Processor**. Here, we will set the number of processors to 2. You can choose more “Processors” depending on the number of cores your host machine has.



VitalPBX 4 - Settings

System

Processor(s): 2

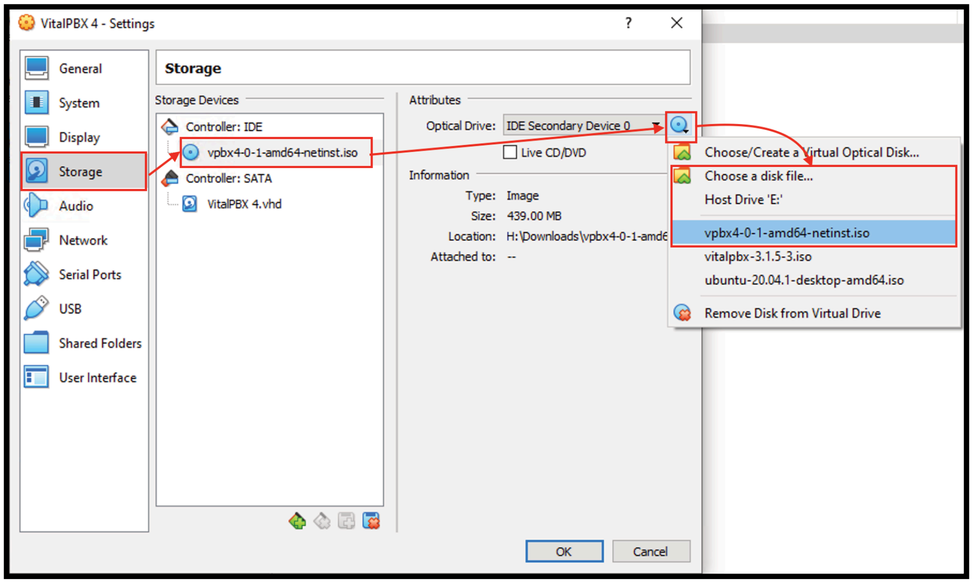
Execution Cap: 100%

Extended Features: □ Enable PAE/NX

□ Enable Nested VT-x/AMD-V

Image 1.1.6 - Virtual Box, virtual machine settings.

Next, click on **Storage**. Here, click on the **empty storage**, then the disk icon next to **Optical Drive**, and choose the ISO image we downloaded from VitalPBX’s Website.



VitalPBX 4 - Settings

Storage

Storage Devices

Controller: IDE

vpbx4-0-1-amd64-netinst.iso

Controller: SATA

VitalPBX 4.vhd

Attributes

Optical Drive: IDE Secondary Device 0

□ Live CD/DVD

Information

Type: Image

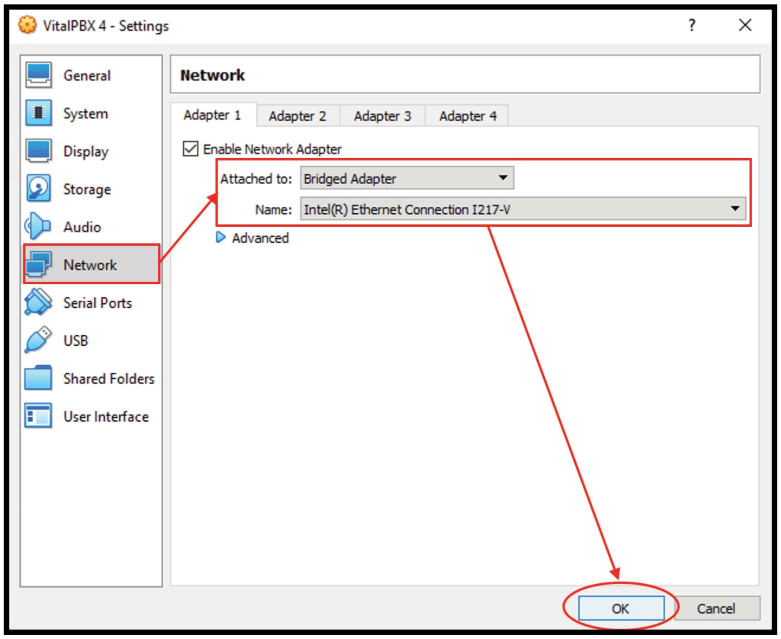
Size: 439.00 MB

Location: H:\Downloads\vpbx4-0-1-amd6

Attached to: --

Image 1.1.7 - Virtual Box, storage settings.

Finally, we will go to **Network** and change “Attached to” to **Bridged Adapter**. You can then select a network adapter on your host machine. This will allow this virtual machine to connect to your local network. Finally, click on **OK**.



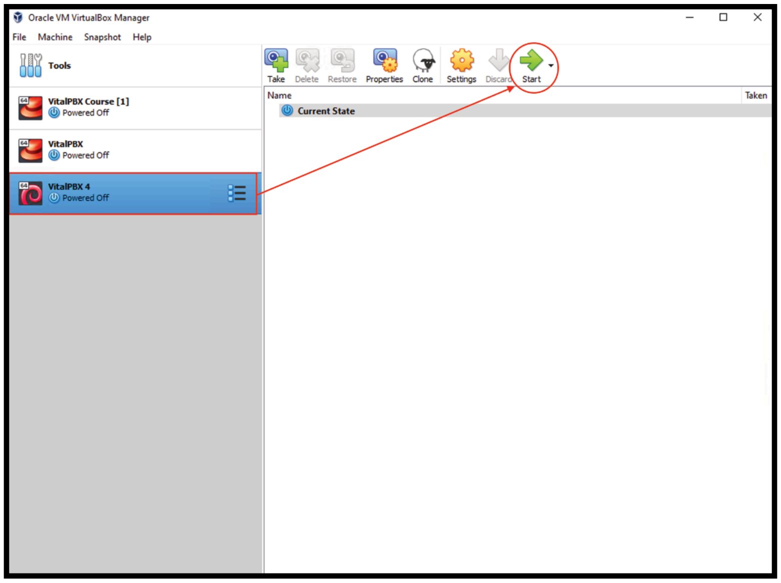
☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: Intel(R) Ethernet Connection I217-V

Image 1.1.8 - Virtual Box, virtual machine network settings.

Now that we have set up our virtual machine, we can turn it on. You can **double-click on your virtual machine** or click on the **green arrow start button**.



VitalPBX Course [1]
Powered Off

VitalPBX
Powered Off

VitalPBX 4
Powered Off

Image 1.1.9 - Virtual Box, starting a virtual machine.

Once you have initiated the virtual machine, you will be shown the installation process for Debian with VitalPBX pre-selected options. To proceed, press enter. Just be sure you are in the virtual machine by clicking on the window screen.

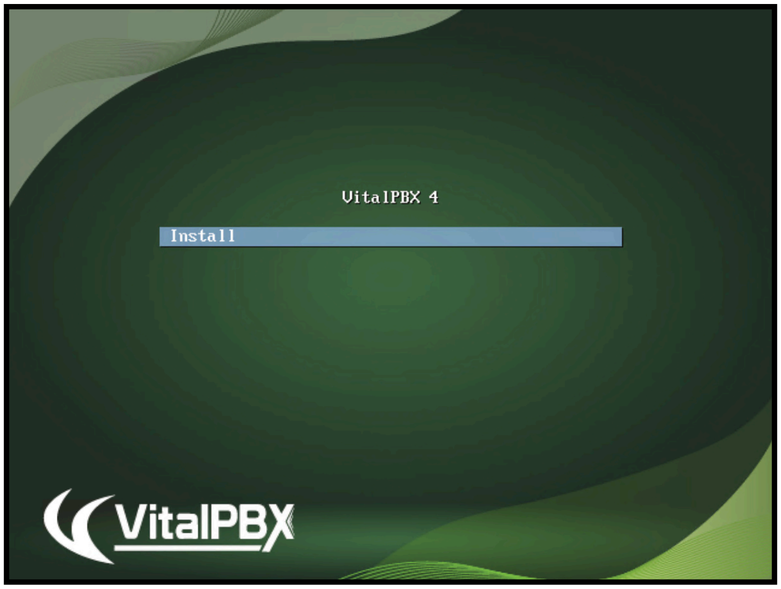
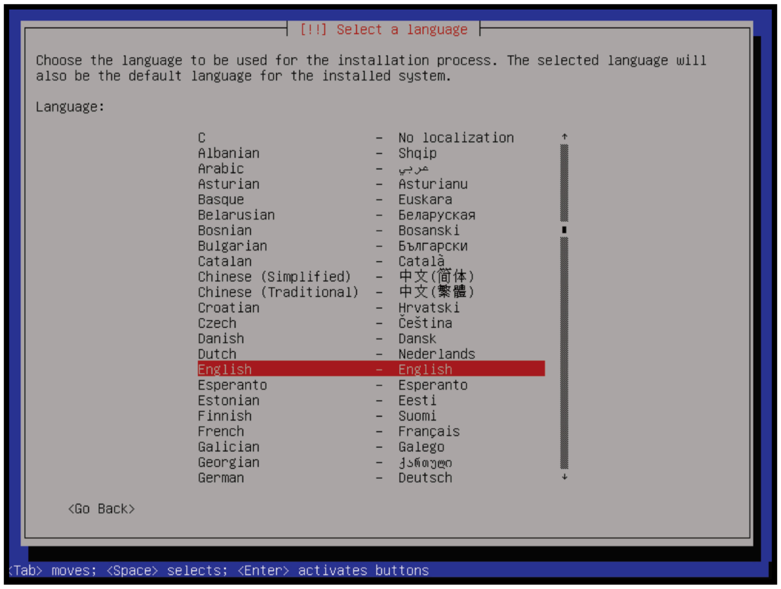


Image 1.1.10 - VitalPBX Installation first screen.

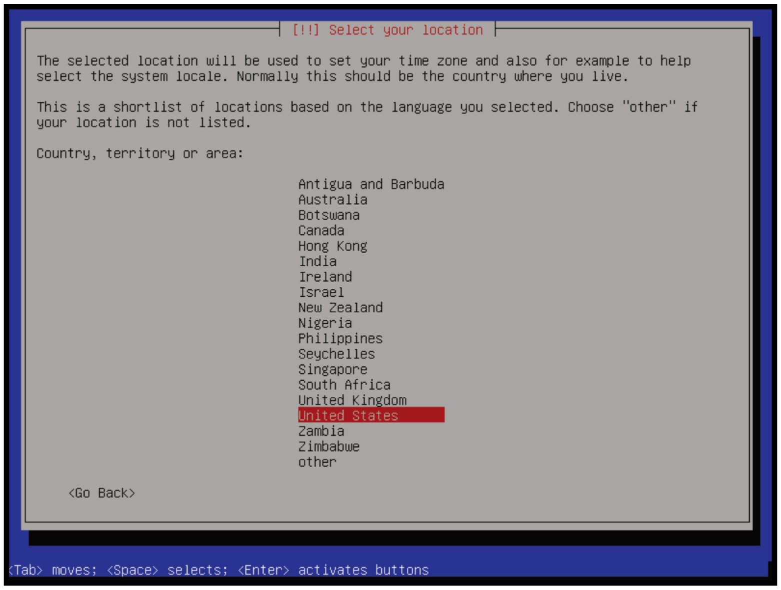
Afterward, select your preferred language. Press enter.



```
[!!] Select a language

Choose the language to be used for the installation process. The selected language will
also be the default language for the installed system.

Language:

                     C                      -  No localization
                     Albanian               -  Shqip
                     Arabic                 -  عربي
                     Asturian               -  Asturianu
                     Basque                 -  Euskara
                     Belarusian             -  Беларуская
                     Bosnian                -  Bosanski
                     Bulgarian              -  Български
                     Catalan                -  Català
                     Chinese (Simplified)   -  中文(简体)
                     Chinese (Traditional)  -  中文(繁體)
                     Croatian               -  Hrvatski
                     Czech                  -  Čeština
                     Danish                 -  Dansk
                     Dutch                  -  Nederlands
                     English                -  English
                     Esperanto              -  Esperanto
                     Estonian               -  Eesti
                     Finnish                -  Suomi
                     French                 -  Français
                     Galician               -  Galego
                     Georgian               -  ქართული
                     German                 -  Deutsch

    <Go Back>

<Tab> moves; <Space> selects; <Enter> activates buttons
```

Image 1.1.11 - Language selection screen.

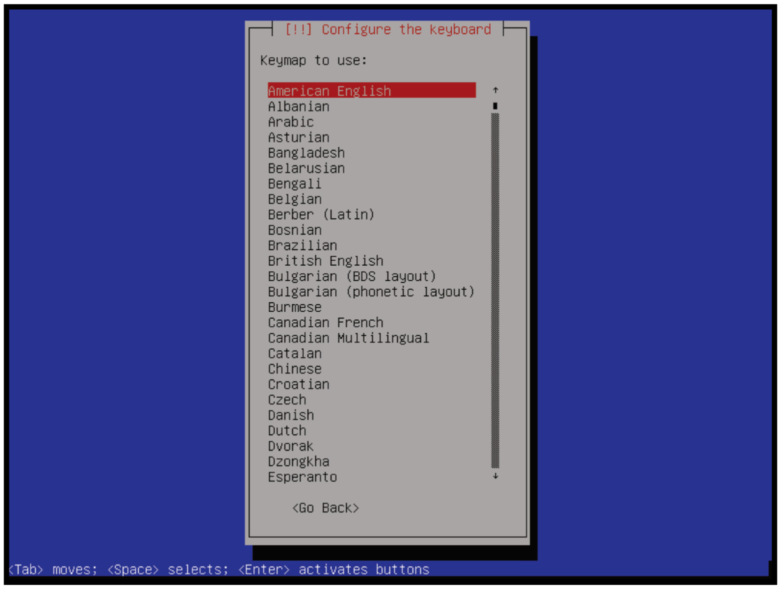
Next, you will need to choose your location. You can use the arrow keys and spacebar to pick. Once you have your location selected, you can press enter.



```
[!!] Select your location

The selected location will be used to set your time zone and also for example to help
select the system locale. Normally this should be the country where you live.

This is a shortlist of locations based on the language you selected. Choose "other" if
your location is not listed.

Country, territory or area:

                                  Antigua and Barbuda
                                  Australia
                                  Botswana
                                  Canada
                                  Hong Kong
                                  India
                                  Ireland
                                  Israel
                                  New Zealand
                                  Nigeria
                                  Philippines
                                  Seychelles
                                  Singapore
                                  South Africa
                                  United Kingdom
                                  United States
                                  Zambia
                                  Zimbabwe
                                  other

    <Go Back>

<Tab> moves; <Space> selects; <Enter> activates buttons
```

Image 1.1.12 - Locations selection screen.

Then, pick your keyboard layout and press enter.

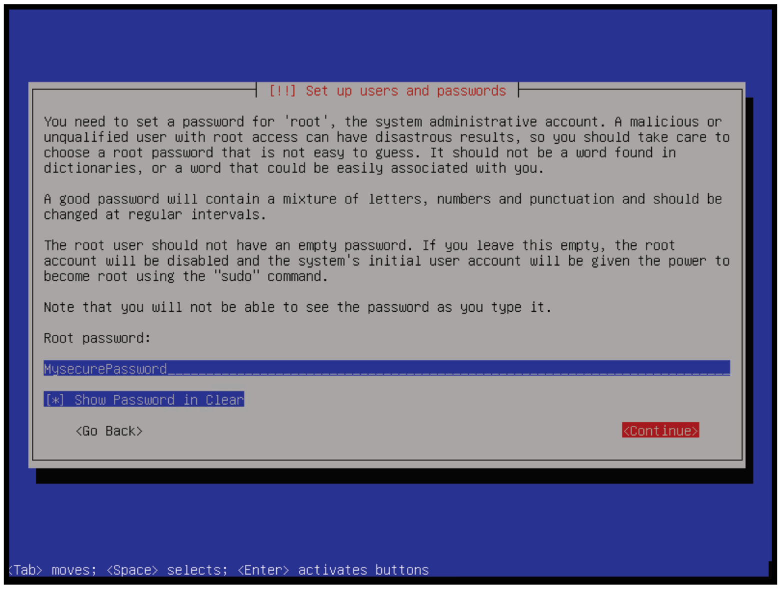


```
[!!] Configure the keyboard

Keymap to use:

 American English
 Albanian
 Arabic
 Asturian
 Bangladesh
 Belarusian
 Bengali
 Belgian
 Berber (Latin)
 Bosnian
 Brazilian
 British English
 Bulgarian (BDS layout)
 Bulgarian (phonetic layout)
 Burmese
 Canadian French
 Canadian Multilingual
 Catalan
 Chinese
 Croatian
 Czech
 Danish
 Dutch
 Dvorak
 Dzongkha
 Esperanto

    <Go Back>

<Tab> moves; <Space> selects; <Enter> activates buttons
```

Image 1.1.13 - Keyboard layout selection screen.

With all these options selected, the installation process will begin.

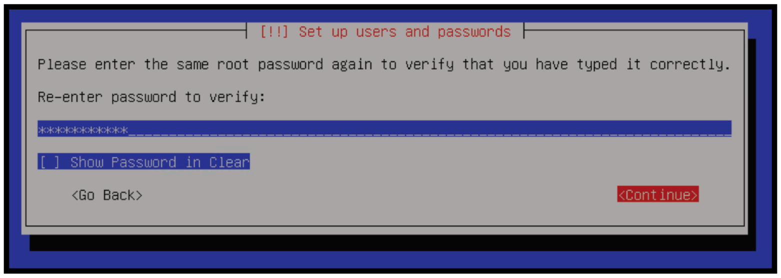
Afterward, you will be prompted to enter your root password. Enter a root password you won’t forget and is difficult to decipher. You can choose to view the password. Moving using the arrow keys to **“Show Password in Clear”** and pressing the space bar.



```
[!!] Set up users and passwords

You need to set a password for 'root', the system administrative account. A malicious or
unqualified user with root access can have disastrous results, so you should take care to
choose a root password that is not easy to guess. It should not be a word found in
dictionaries, or a word that could be easily associated with you.

A good password will contain a mixture of letters, numbers and punctuation and should be
changed at regular intervals.

The root user should not have an empty password. If you leave this empty, the root
account will be disabled and the system's initial user account will be given the power to
become root using the "sudo" command.

Note that you will not be able to see the password as you type it.

Root password:

MysecurePassword

[*] Show Password in Clear

    <Go Back>                                                          <Continue>

<Tab> moves; <Space> selects; <Enter> activates buttons
```

Image 1.1.14 - Root password selection screen.

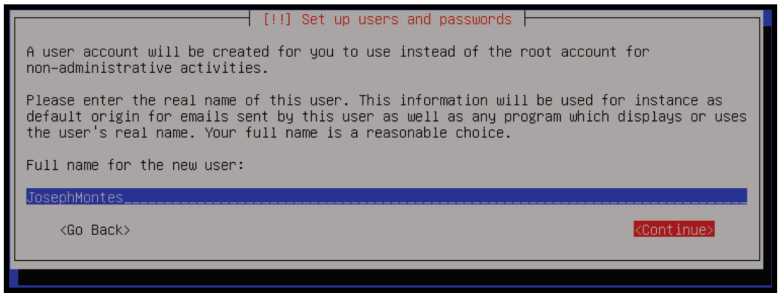
Then, navigate with the arrow keys to **Continue** and press enter. You will then need to re-enter the root password. Afterward, navigate to the **Continue** button and press enter.



```
[!!] Set up users and passwords

Please enter the same root password again to verify that you have typed it correctly.

Re-enter password to verify:

************

[ ] Show Password in Clear

    <Go Back>                                                          <Continue>
```

Image 1.1.15 - Root password re-enter screen.

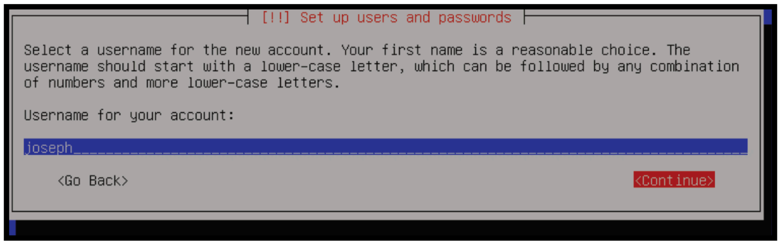
You will then be prompted to enter a name for a new Debian user. Your full name is recommended.



```
[!!] Set up users and passwords

A user account will be created for you to use instead of the root account for
non-administrative activities.

Please enter the real name of this user. This information will be used for instance as
default origin for emails sent by this user as well as any program which displays or uses
the user's real name. Your full name is a reasonable choice.

Full name for the new user:

JosephMontes

    <Go Back>                                                          <Continue>
```

Image 1.1.16 - New Debian user creation screen.

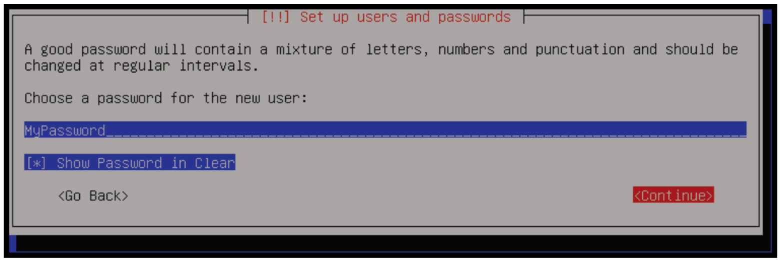
Next, you will be asked to create a username for the new user. We recommend using all lowercase letters and no spaces. Once entered, navigate to the **Continue button** and press enter.



```
[!!] Set up users and passwords

Select a username for the new account. Your first name is a reasonable choice. The
username should start with a lower-case letter, which can be followed by any combination
of numbers and more lower-case letters.

Username for your account:

joseph

    <Go Back>                                                              <Continue>
```

Image 1.1.17 - New Debian username creation screen.

Then, enter a new password for the Debian user. You will then be prompted to re-enter the password. On both screens, navigate to the **Continue** button and hit enter to move forward.

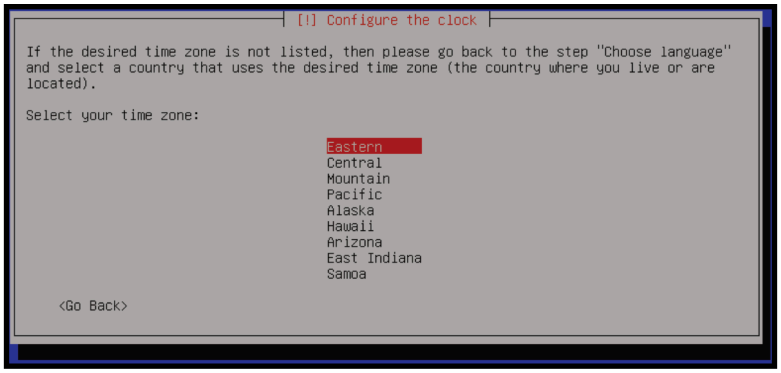


```
[!!] Set up users and passwords

A good password will contain a mixture of letters, numbers and punctuation and should be
changed at regular intervals.

Choose a password for the new user:

MyPassword

[*] Show Password in Clear

    <Go Back>                                                              <Continue>
```

Image 1.1.18 - New Debian username password screen.

Once again, we recommend you **use a strong password that you can remember easily**. Preferably, this password is **different** from the root password.

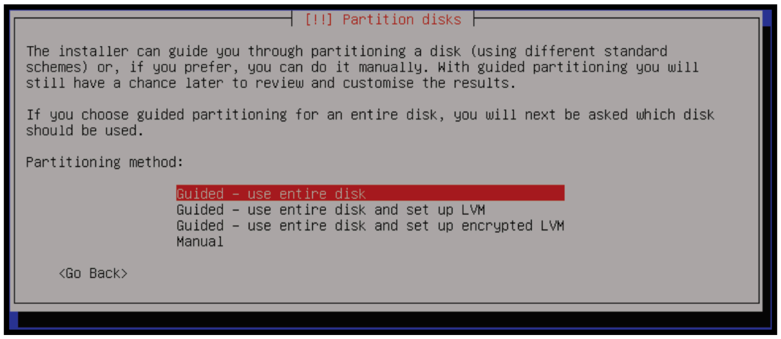
Next, you will be presented with the time zone selection screen. Navigate to your time zone and then press enter.



```
[!] Configure the clock

If the desired time zone is not listed, then please go back to the step "Choose language"
and select a country that uses the desired time zone (the country where you live or are
located).

Select your time zone:

                                     Eastern
                                     Central
                                     Mountain
                                     Pacific
                                     Alaska
                                     Hawaii
                                     Arizona
                                     East Indiana
                                     Samoa

    <Go Back>
```

Image 1.1.19 - Debian clock configuration screen.

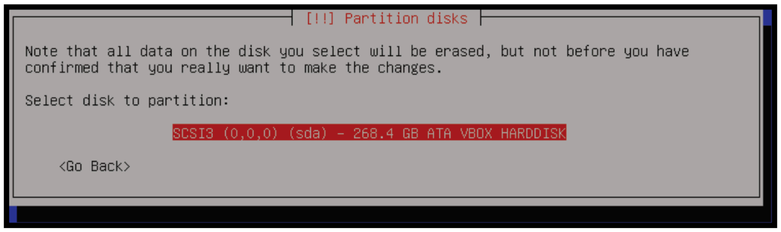
You will then be prompted to partition your disk. Here, we select the **"Guided - use entire disk"** option.



```
[!!] Partition disks

The installer can guide you through partitioning a disk (using different standard
schemes) or, if you prefer, you can do it manually. With guided partitioning you will
still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk
should be used.

Partitioning method:

                   Guided - use entire disk
                   Guided - use entire disk and set up LVM
                   Guided - use entire disk and set up encrypted LVM
                   Manual

    <Go Back>
```

Image 1.1.20 - Debian partition disks screen.

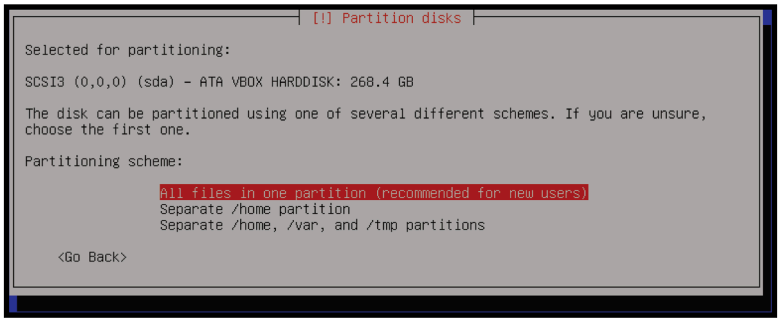
You will then see your disk partitions. We should only have one in this case. Press enter.



```
[!!] Partition disks

Note that all data on the disk you select will be erased, but not before you have
confirmed that you really want to make the changes.

Select disk to partition:

                   SCSI3 (0,0,0) (sda) - 268.4 GB ATA VBOX HARDDISK

    <Go Back>
```

Image 1.1.21 - Debian partition disks selection screen.

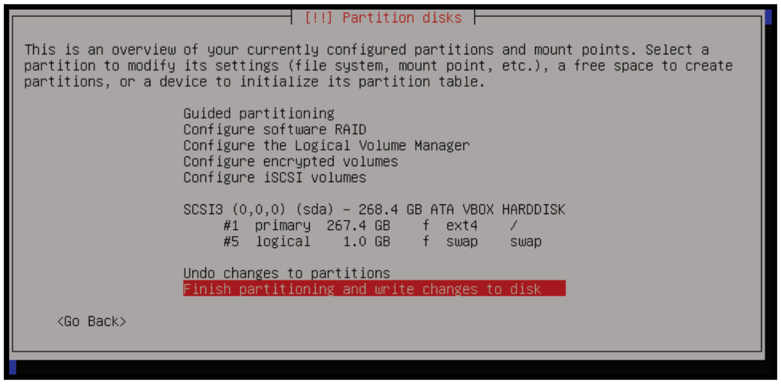
Afterward, you will need to select your partitioning scheme. Here we choose **“All files in one partition.”** Press enter.



```
[!] Partition disks

Selected for partitioning:

SCSI3 (0,0,0) (sda) - ATA VBOX HARDDISK: 268.4 GB

The disk can be partitioned using one of several different schemes. If you are unsure,
choose the first one.

Partitioning scheme:

                 All files in one partition (recommended for new users)
                 Separate /home partition
                 Separate /home, /var, and /tmp partitions

    <Go Back>
```

Image 1.1.22 - Debian partition disks screen, scheme selection screen.

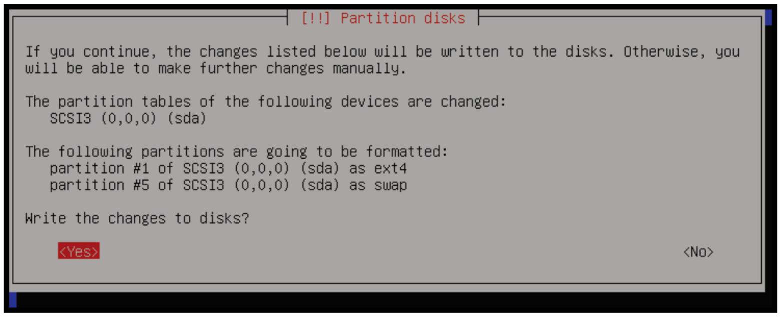
We then verify our screen partitioning, navigate to **“Finish partitioning and write changes to disk,”** and press enter.



```
[!!] Partition disks

This is an overview of your currently configured partitions and mount points. Select a
partition to modify its settings (file system, mount point, etc.), a free space to create
partitions, or a device to initialize its partition table.

                    Guided partitioning
                    Configure software RAID
                    Configure the Logical Volume Manager
                    Configure encrypted volumes
                    Configure iSCSI volumes

                    SCSI3 (0,0,0) (sda) - 268.4 GB ATA VBOX HARDDISK
                         #1  primary  267.4 GB    f  ext4    /
                         #5  logical    1.0 GB    f  swap    swap

                    Undo changes to partitions
                    Finish partitioning and write changes to disk

    <Go Back>
```

Image 1.1.23 - Debian partition disks verification screen.

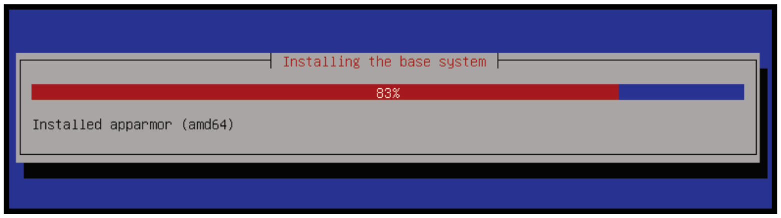
A warning will appear with the changes to occur. Navigate to **Yes** using the arrow keys, and press enter.



```
[!!] Partition disks

If you continue, the changes listed below will be written to the disks. Otherwise, you
will be able to make further changes manually.

The partition tables of the following devices are changed:
   SCSI3 (0,0,0) (sda)

The following partitions are going to be formatted:
   partition #1 of SCSI3 (0,0,0) (sda) as ext4
   partition #5 of SCSI3 (0,0,0) (sda) as swap

Write the changes to disks?

    <Yes>                                                                       <No>
```

Image 1.1.24 - Debian partition disks changes confirmation screen.

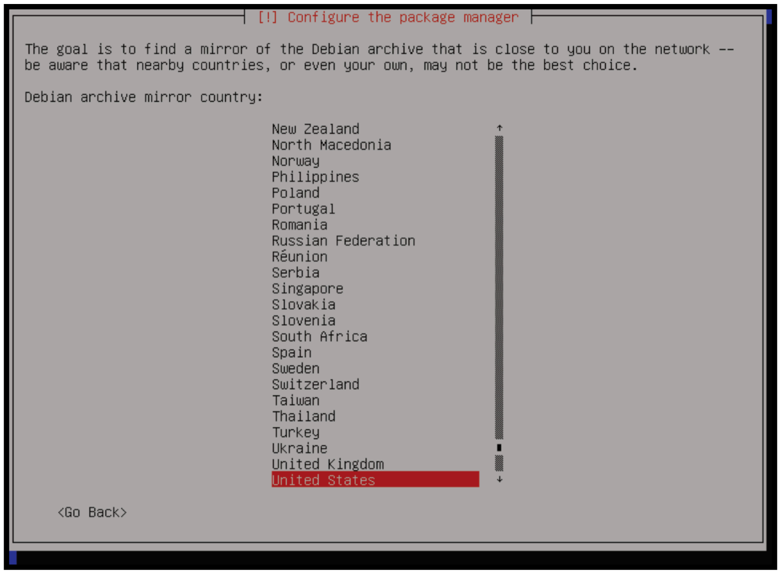
Your disk partitioning will start processing, and the operating system will begin the installation. This process will take a couple of minutes.



```
Installing the base system

                                    83%

Installed apparmor (amd64)
```

Image 1.1.25 - Debian installation process.

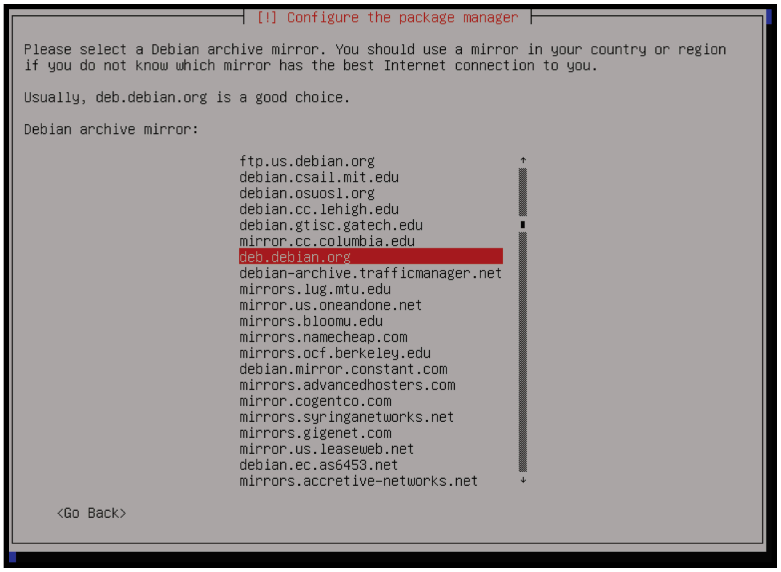
Next, you will need to select your package manager location. This will ensure that you get the closest mirrors to your area. This will assure faster download speeds and reliability. Navigate to your preferred location with the arrow keys and press enter.



```
[!] Configure the package manager

The goal is to find a mirror of the Debian archive that is close to you on the network --
be aware that nearby countries, or even your own, may not be the best choice.

Debian archive mirror country:

                                New Zealand
                                North Macedonia
                                Norway
                                Philippines
                                Poland
                                Portugal
                                Romania
                                Russian Federation
                                Réunion
                                Serbia
                                Singapore
                                Slovakia
                                Slovenia
                                South Africa
                                Spain
                                Sweden
                                Switzerland
                                Taiwan
                                Thailand
                                Turkey
                                Ukraine
                                United Kingdom
                                United States

    <Go Back>
```

Image 1.1.26 - Debian package manager selection screen.

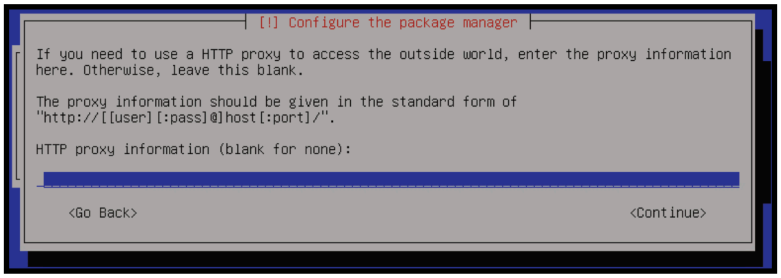
Select your preferred mirror. By default, the selection is deb.debian.org. If you don't have a preference, you can leave it as default and press enter.



```
[!] Configure the package manager

Please select a Debian archive mirror. You should use a mirror in your country or region
if you do not know which mirror has the best Internet connection to you.

Usually, deb.debian.org is a good choice.

Debian archive mirror:

                            ftp.us.debian.org
                            debian.csail.mit.edu
                            debian.osuosl.org
                            debian.cc.lehigh.edu
                            debian.gtisc.gatech.edu
                            mirror.cc.columbia.edu
                            deb.debian.org
                            debian-archive.trafficmanager.net
                            mirrors.lug.mtu.edu
                            mirror.us.oneandone.net
                            mirrors.bloomu.edu
                            mirrors.namecheap.com
                            mirrors.ocf.berkeley.edu
                            debian.mirror.constant.com
                            mirrors.advancedhosters.com
                            mirror.cogentco.com
                            mirrors.syringanetworks.net
                            mirrors.gigenet.com
                            mirror.us.leaseweb.net
                            debian.ec.as6453.net
                            mirrors.accretive-networks.net

    <Go Back>
```

Image 1.1.27 - Debian mirror selection screen.

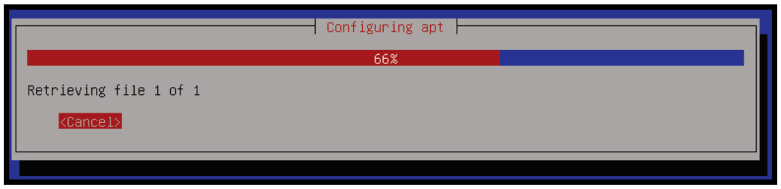
Afterward, you will be prompted to enter an internet proxy. If you don't use one, you can leave it blank. We navigate to the **Continue** button using the arrow keys and press enter.



```
[!] Configure the package manager

If you need to use a HTTP proxy to access the outside world, enter the proxy information
here. Otherwise, leave this blank.

The proxy information should be given in the standard form of
"http://[[user][:pass]@]host[:port]/".

HTTP proxy information (blank for none):

<Go Back>                                                              <Continue>
```

Image 1.1.28 - Debian internet proxy selection screen.

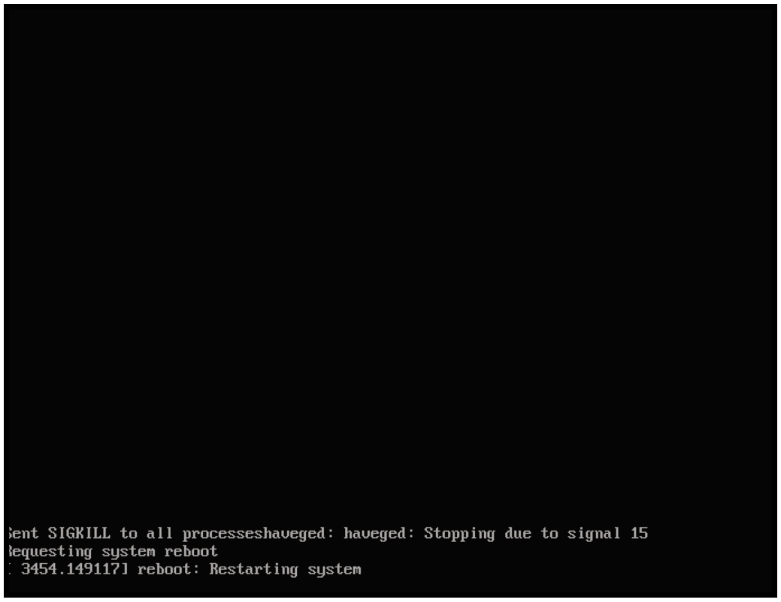
The installation of the operating system will continue.



```
Configuring apt

66%

Retrieving file 1 of 1

<Cancel>
```

Image 1.1.29 - Debian apt configuration.

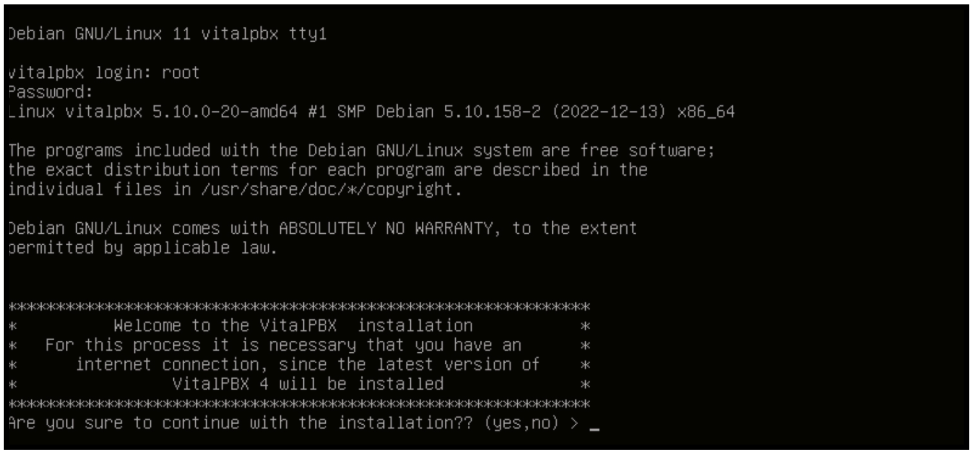
Once the installation is done, your virtual machine will reboot.



```
ent SIGKILL to all processeshaveged: haveged: Stopping due to signal 15
equesting system reboot
 3454.149117] reboot: Restarting system
```

Image 1.1.30 - Virtual Machine reboot.

After the virtual machine reboots, you will be prompted to log in. Log in using the **root user with the password you created**. Press enter.



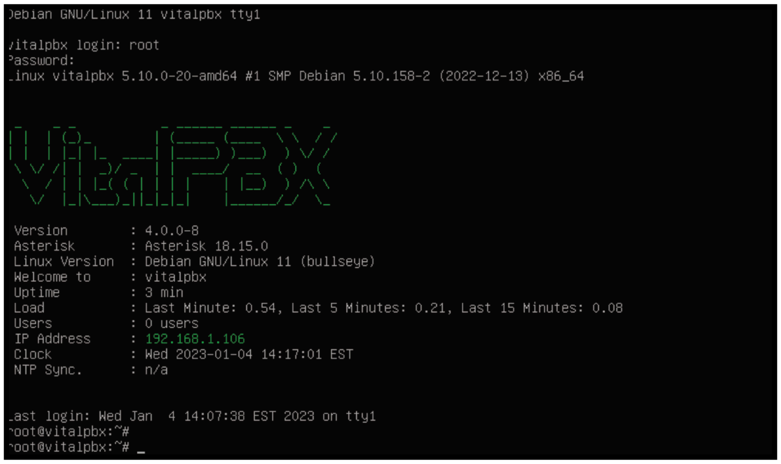
```
Debian GNU/Linux 11 vitalpbx tty1

vitalpbx login: root
Password:
Linux vitalpbx 5.10.0-20-amd64 #1 SMP Debian 5.10.158-2 (2022-12-13) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.


************************************************************
*          Welcome to the VitalPBX  installation           *
*   For this process it is necessary that you have an      *
*      internet connection, since the latest version of    *
*                VitalPBX 4 will be installed              *
************************************************************
Are you sure to continue with the installation?? (yes,no) > _
```

Image 1.1.31 - VitalPBX installation prompt

You will then be prompted to install VitalPBX. Enter **yes** in the prompt, and press enter.

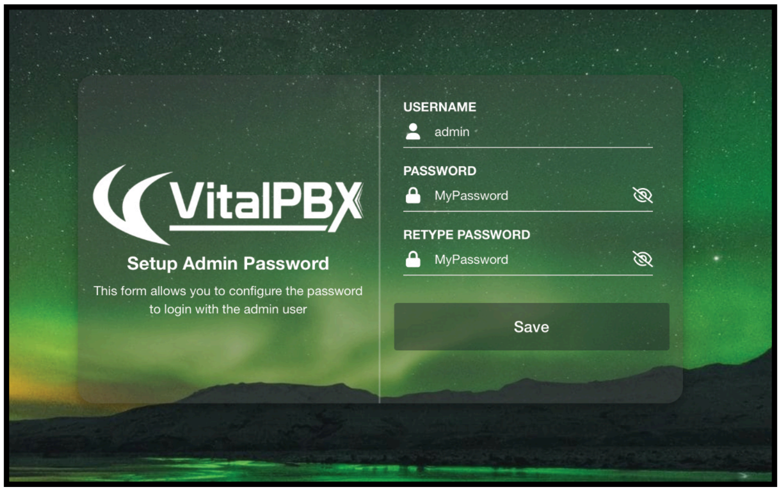
This will then update Debian and download and install the latest version of VitalPBX. After the installation is done, your system will reboot.

Once the system reboots, you can log in with the root username and password again. After you log in, you will see a nice banner with the VitalPBX Logo and your system information. You can copy the IP address and enter it into your web browser.



```
Debian GNU/Linux 11 vitalpbx tty1

vitalpbx login: root
Password:
Linux vitalpbx 5.10.0-20-amd64 #1 SMP Debian 5.10.158-2 (2022-12-13) x86_64


 Version        : 4.0.0-8
 Asterisk       : Asterisk 18.15.0
 Linux Version  : Debian GNU/Linux 11 (bullseye)
 Welcome to     : vitalpbx
 Uptime         : 3 min
 Load           : Last Minute: 0.54, Last 5 Minutes: 0.21, Last 15 Minutes: 0.08
 Users          : 0 users
 IP Address     : 192.168.1.106
 Clock          : Wed 2023-01-04 14:17:01 EST
 NTP Sync.      : n/a


Last login: Wed Jan  4 14:07:38 EST 2023 on tty1
root@vitalpbx:~#
root@vitalpbx:~# _
```

Image 1.1.32 - VitalPBX installation complete.


USERNAME
admin
PASSWORD
MyPassword
RETYPE PASSWORD
MyPassword
Save

VitalPBX
Setup Admin Password
This form allows you to configure the password to login with the admin user

Image 1.1.33 - VitalPBX, initial login screen.

When you enter the IP Address in your browser, you will be greeted with VitalPBX's initial login screen. Here, you can enter the **admin username and password**. By default, the admin user is just "admin," but you can choose any username of your liking.
The password you use can differ from the one you used for the root and Debian usernames. This is a separate user from the ones you created during installation. And with that, you have successfully installed VitalPBX. Congratulations!

Section 1, Lesson 2 - Dedicated Hardware Installation.

The installation process of VitalPBX in dedicated hardware is similar to how you do a virtual machine installation. So, we will show only some of the same steps again. Here, we'll only cover the preparation steps of getting you ready to install VitalPBX in dedicated hardware. You can refer to the Virtual Machine installation process for the rest of the steps.

When we say dedicated hardware, we mean that we will install the software on a computer or server that will only work as our VitalPBX installation. It is important to note that the computer or server you are installing VitalPBX on needs a dedicated monitor and keyboard to proceed with the installation.

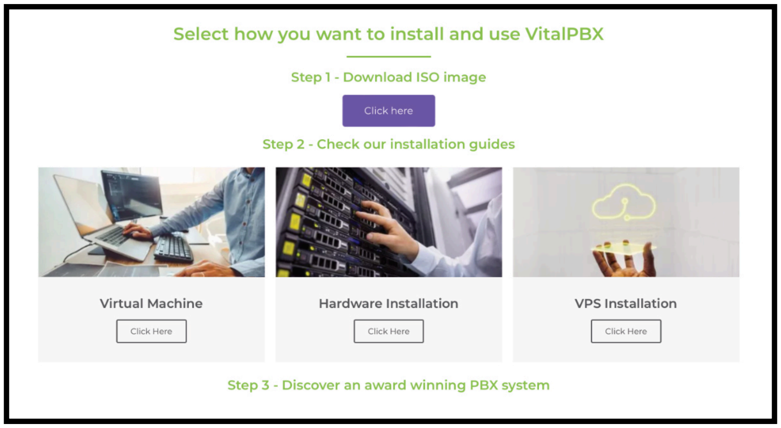

Select how you want to install and use VitalPBX
Step 1 - Download ISO image
Step 2 - Check our installation guides
Step 3 - Discover an award winning PBX system

Image 1.2.1 - VitalPBX download section.

So, we first need to download the ISO image from VitalPBX's official website at https://www.vitalpbx.com, just as we did in the previous lesson. This can be found under the **Download section** and by scrolling down. There, you will also find written guides on how to install VitalPBX in different environments.

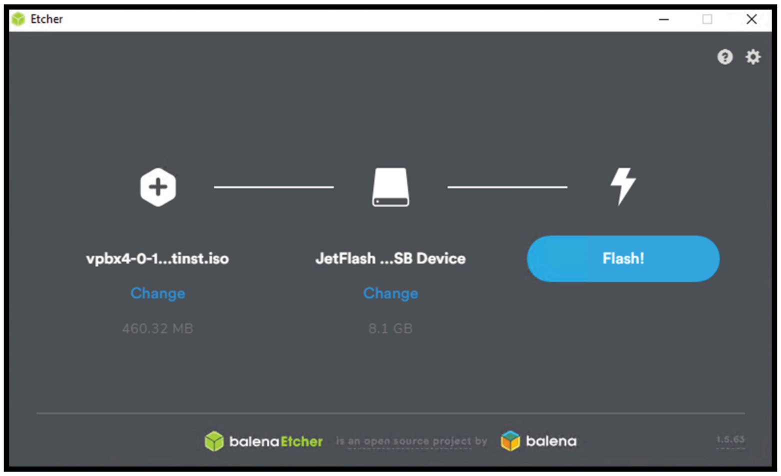


Image 1.2.2 - Preparing Balena Etcher to flash a USB flash drive.

Next, we'll need to flash the ISO image to a USB flash drive to install VitalPBX on dedicated hardware. To do this, we can download **Balena Etcher** at https://www.balena.io/etcher/. Balena Etcher is a free storage drive flashing tool that works in **Windows** or **macOS**.

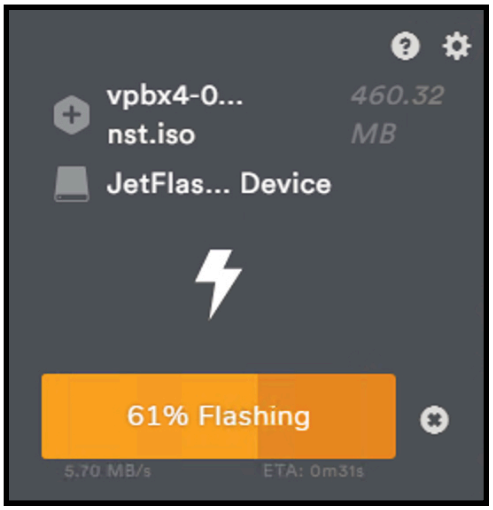


Image 1.2.3 - Balena Etcher flashing USB flash drive.

Once you have installed Balena Etcher and run it, choose the ISO image you have downloaded. Then, select any USB flash drive you have available.

The USB flash drive has to be **at least 2GB in storage size**.
Finally, click on the blue **"Flash!"** Button. This will start the flashing process, which can take a few minutes.

Once it's done, you can eject the drive safely.

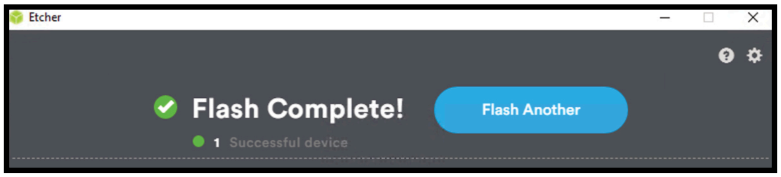


Image 1.2.4 - Balena Etcher done flashing.

Once you have securely ejected the USB flash drive, you can insert it into a USB port in your dedicated hardware. Proceed to turn on the server and press the **boot drive options key** while the system starts.

The boot drive option key varies depending on the motherboard manufacturer. Usually, it is the **Delete key**, **F2**, **F11**, or **F12**.
In the boot drive options, select the **USB flash drive you flashed**. Reboot your server or computer. The system will boot from the USB flash drive, and you can begin VitalPBX's installation process. To proceed, press enter.

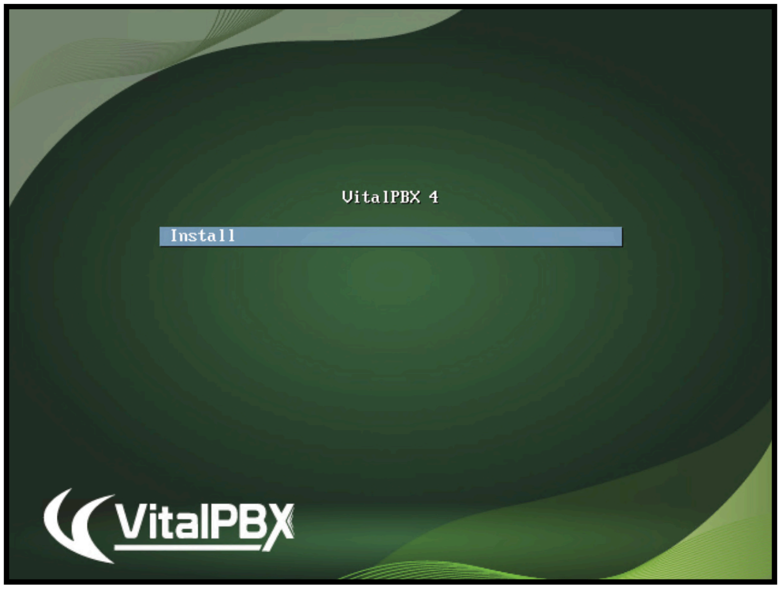


Image 1.2.5 - VitalPBX Installation first screen.

You can follow the same procedure in the Virtual Machine installation lesson, where we installed VitalPBX in a virtual machine.

Once you have followed these steps, you have successfully installed VitalPBX. Congratulations!

Section 1, Lesson 3 - Installing VitalPBX in a Debian and VPS Installation.

The process to install VitalPBX in an existing minimal installation of Debian is simple. You don't need to have the Desktop or Gnome environment for Debian to install VitalPBX.

This will be the same process to install VitalPBX in any **VPS service provider** like **Google Cloud**, **Digital Ocean**, and **Vultr**, among others. You will need to create a new **minimal Debian 64-bit-based system**.

Once you have Debian installed, you can run a couple of commands.

You will need to log in as the root user, as root permissions are required to run the script. You can also use the *"sudo su"* command.

If your Debian user is not in the sudoers list, you need to add it to be able to run the *"sudo su"* command. We will take a look into this later on.

First, download the VitalPBX installation script using the following command.

```
root@debian:~# wget https://repo.vitalpbx.com/vitalpbx/v4/apt/install_vpbx4.sh
```

Next, we will give the script execution permissions.

```
root@debian:~# chmod +x install_vpbx4.sh
```

Finally, we execute the script.

```
root@debian:~# ./install_vpbx4.sh
```

Once you execute the script, you will be prompted to enter "yes" to proceed. Enter the word **yes** in the prompt and press enter.

This will update the system and begin the VitalPBX installation process. When it ends, the system will **reboot**.

Once the system reboots, you can log in again using the root user. You are now greeted with a VitalPBX Logo and system information.

You can copy the IP address into your web browser and configure the **admin username and password**. This username and password differ from the root username and password or any other user you may have created within Debian.

And with this, you have successfully installed VitalPBX in a Debian-based installation. Congratulations!

Section 1, Lesson 5 - Installing VitalPBX on ARM-based systems.

With VitalPBX being based on Debian with version 4, you can now install VitalPBX in an ARM-based environment.

There is **no ISO image based on ARM** for VitalPBX, so you must start with a **minimal Debian 64-bit installation**. You can download the **ARM64 ISO Network Install image** from Debian's website at https://www.debian.org/CD/netinst.

This also means a **Raspberry Pi** system can be your VitalPBX server. For this, you'll flash a microSD card with a **Debian** or **Raspberry Pi OS** image.

In this lesson, I am installing VitalPBX on an M1-Max MacBook Pro. You can choose any virtualizer like **Parallels** or **UTM**. The process is mostly the same.
The installation process is a regular Debian Installation. Refer to our Virtual Machine installation lesson and check the troubleshooting tips we will share later.

Once the system reboots with ARM Debian installed, log in as root and run the following commands.

```
root@debian:~# wget https://repo.vitalpbx.com/vitalpbx/v4/apt/install_vpbx4.sh
```

Then give it execution permissions.

```
root@debian:~# chmod +x install_vpbx4.sh
```

Finally, we execute the script.

```
root@debian:~# ./install_vpbx4.sh
```

You will then be prompted to continue with the installation, so enter "yes" in the prompt and press enter.

Section 1, Lesson 6 - Troubleshoot Debian Installation.

You might find some questions during your installation, especially if you are unfamiliar with a Debian installation.

Here are some tips and recommendations concerning a Debian installation. If you use VitalPBX's ISO image, these options won't appear, and default values will be chosen.

But, if you make a custom Debian installation or an ARM installation, you might find some additional configuration options during installation.

1. **Hostname and Domain Name**
   In these fields, you can enter any information. For the hostname, if you use an FQDN (Fully Qualified Domain Name), use it here—for example, *sip.mydomain.com*. In the Domain Name option, you can enter your domain name. Like, *mydomain.com*.

2. **Scanning extra installation media**
   You may be prompted to “Scan extra installation media” when configuring the package manager. This option can be ignored, and respond “no.”

3. **The *“wget”* command is not available.**
   You might get an error when trying to download the installation script. The error is saying that the *“wget”* command does not exist. You can download and install the *“wget”* command with the following.

```
root@debian:~# apt-get install wget -y
```

4. **Software Selection.**
   By the end of the Debian installation, you will be asked which software you wish to include. We recommend you leave the list with everything unselected, as you will install all dependencies with the VitalPBX installation script.

These are the main differences between a custom Debian installation and installing from the VitalPBX ISO image.

Section 1, Lesson 7 - Post-Installation Options.

With your VitalPBX installation done, there are some things you can do for ease of configuration.

1. **Remote access using the root user.**
   You cannot remotely access the server using the root user by default. This can be changed by modifying the *“sshd_config”* file. First, log in as root directly on the terminal. Then, edit the following file using Nano.

```
root@debian~:# nano /etc/ssh/sshd_config
```

Change the line.

```
#PermitRootLogin prohibit-password
```

With.

```
#PermitRootLogIn yes
```

Save the document and exit.

Optionally, if you want to keep the root user from remotely accessing the server, you can add a Debian user to the “sudoers group.” Enter the following command.

```
root@debian:~# usermod -aG sudo username
```

2. **Change to a Static IP address.**
   By default, your system will pull an IP address using DHCP. It is recommended that your server use a static IP address instead. To change it, we will modify the interfaces file. Run the following command.

```
root@debian:~# nano /etc/network/interfaces
```

Change this block of text.

```
#The primary network interface
allow-hotplug eth0
iface eth0 inet dchp
```

To this.

```
#The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.1.200
netmask 255.255.255.0
gateway 192.168.1.1
```

Finally, enable the interface and reboot your system.

```
root@debian:~# ifup eth0
root@debian:~# reboot
```

Conclusion

And with this, we have now explored every way to install VitalPBX. This helps you understand the different environments in which to install VitalPBX. Now, you can have your VitalPBX installation done!

Let’s now proceed to the next section, so we can start configuring VitalPBX.

SECTION 2 - INITIAL CONFIGURATIONS

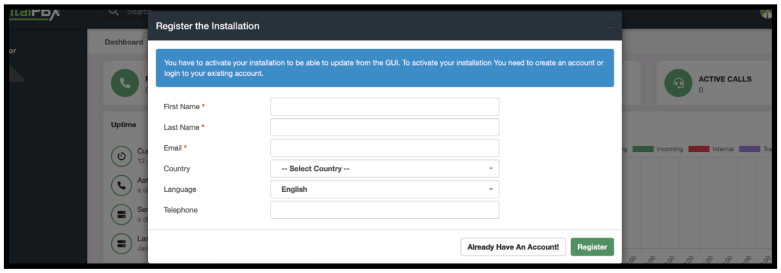
Introduction

With VitalPBX now installed, we can proceed with our configurations. But before we go through every module, let's go through various recommendations we have whenever you make your initial deployment.

With these recommendations, you can reduce issues or have various conveniences when configuring your VitalPBX system.

Section 2, Lesson 1 - System Registration and Interface Overview.

Whenever you first install VitalPBX, you will see the **Register the Installation** prompt. Here, you can enter your information to open an account for your system installations.



Register the Installation

You have to activate your installation to be able to update from the GUI. To activate your installation You need to create an account or login to your existing account.

First Name *

Last Name *

Email *

Country: -- Select Country --

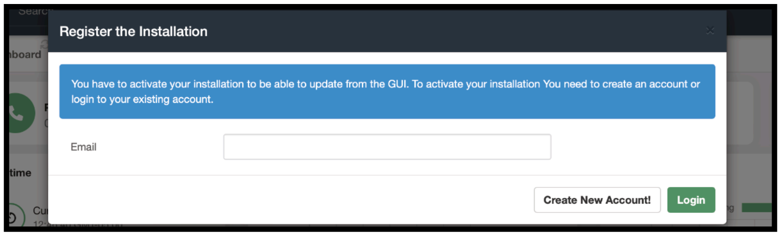
Language: English

Telephone

Image 2.1.1 - VitalPBX installation registration prompt.

This is separate from a vitalpbx.com account. This is so we can record the number of VitalPBX installs worldwide for internal use and is not shared with any third party.

If you have installed VitalPBX previously, you can click the **Already Have An Account** button and enter your email address. Afterward, click on **Register**.



Register the Installation

You have to activate your installation to be able to update from the GUI. To activate your installation You need to create an account or login to your existing account.

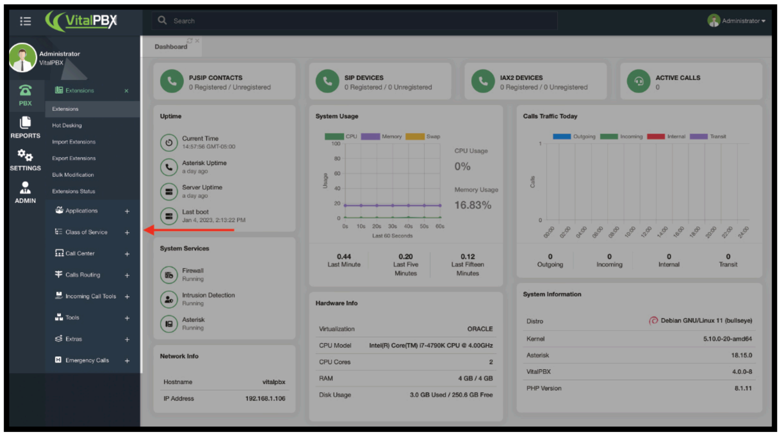
Email

Image 2.1.2 - VitalPBX installation registration with existing email address.

This step is necessary for you to update your VitalPBX installation.

Before we continue, let’s get familiar with the web interface. We have worked on making VitalPBX as intuitive as possible, so learning these core aspects will have you navigating the system in no time.

To the left, you have the **main navigation menu**. Here, you will find the **main categories**, **sub-categories**, and **modules**.



PJSIP CONTACTS
0 Registered / Unregistered

SIP DEVICES
0 Registered / 0 Unregistered

IAX2 DEVICES
0 Registered / 0 Unregistered

ACTIVE CALLS
0

Uptime

Current Time
14:57:56 GMT-05:00

Asterisk Uptime
a day ago

Server Uptime
a day ago

Last boot
Jan 4, 2023, 2:13:22 PM

System Usage

CPU Usage
0%

Memory Usage
16.83%

0.44 Last Minute
0.20 Last Five Minutes
0.12 Last Fifteen Minutes

Calls Traffic Today

0 Outgoing
0 Incoming
0 Internal
0 Transit

System Services

Firewall
Running

Intrusion Detection
Running

Asterisk
Running

Network Info

Hostname	vitalpbx
IP Address	192.168.1.106

Hardware Info

Virtualization	ORACLE
CPU Model	Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz
CPU Cores	2
RAM	4 GB / 4 GB
Disk Usage	3.0 GB Used / 250.6 GB Free

System Information

Distro	Debian GNU/Linux 11 (bullseye)
Kernel	5.10.0-20-amd64
Asterisk	18.15.0
VitalPBX	4.0.0-8
PHP Version	8.1.11

Image 2.1.3 - VitalPBX navigation menu.

On top, you will find the **general search bar.** Here, you can search for any module or configuration from VitalPBX. For example, let’s search for Extensions and press enter.

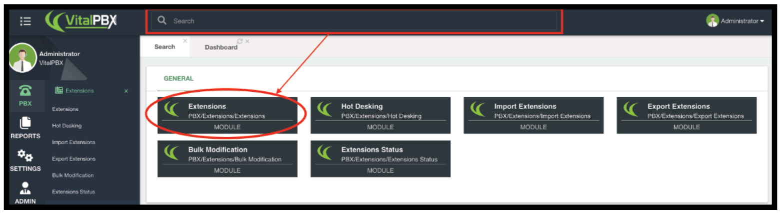


Image 2.1.4 - Search results.

You will see every item related to extensions in the results. Let’s click on **Extensions.** Whenever you are in a module where you can create multiple items, you will see this **list** icon in the top-right corner.

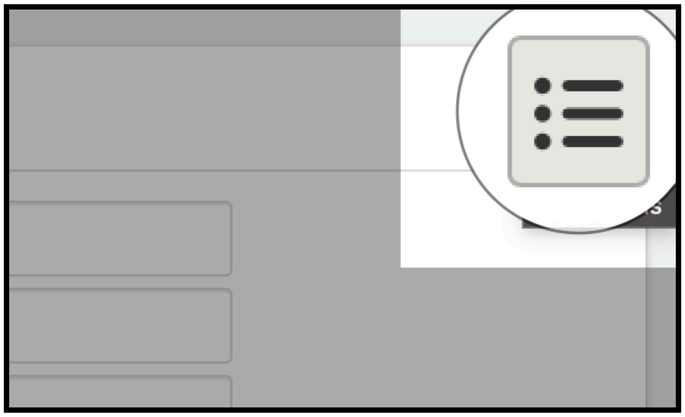


Image 2.1.5 - VitalPBX module list icon.

In this list, you can open and edit items you previously created. There is also a **Search Bar** to search the items based on their **name**, **number**, or **description**.

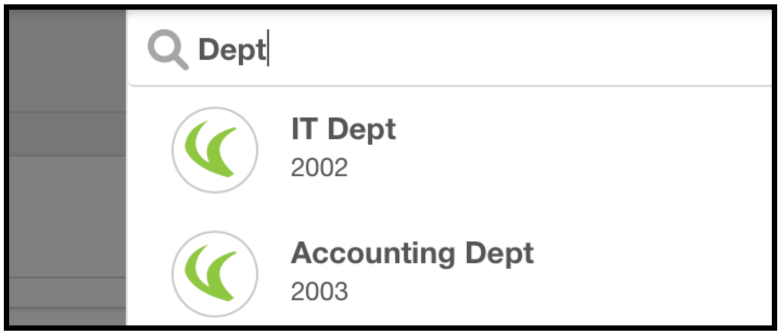
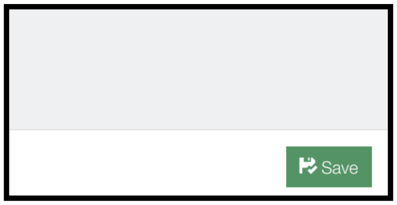


Image 2.1.6 - VitalPBX list search.

When configuring a new item, you will have a Save button in the bottom right-hand corner. If you edit an existing item, you will see an **Update**, **Delete**, and **Cancel** button. The **Update** button will update your configurations with the new ones, **Delete** will delete the item you are editing, and **Cancel** will not save any changes and will present you with a blank canvas.



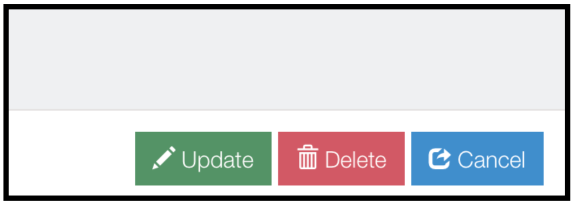


Image 2.1.7 - Save button.

Image 2.1.8 - Update, Delete, and Cancel Buttons.

This will be the same with many of our different modules. More options may appear depending on the module in the lower left-hand corner.

A red arrow icon will appear when you **save or update** most configurations. This will **Apply Changes** from the database to the system's configuration files.

Remember, **Saving your configurations** and **Applying changes** are two separate actions.

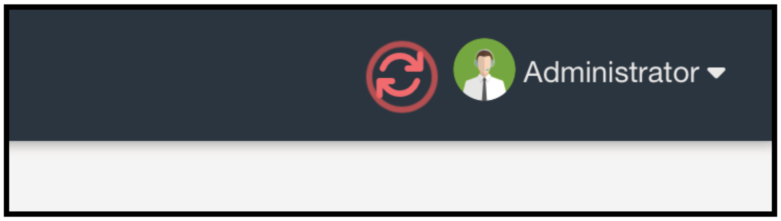


Image 2.1.9 - Apply Changes button.

Right next to it, you will have the **User Menu**. This will display various other settings you can access. We will see these later on.

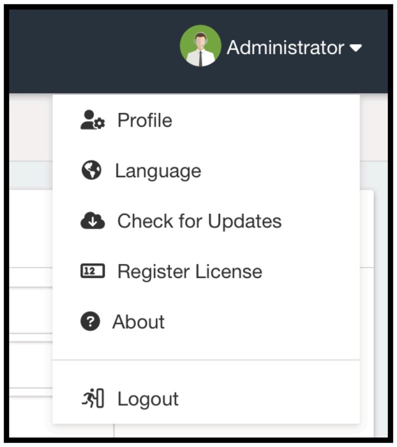


Image 2.1.10 - User menu.

While moving through multiple modules, you will notice that tabs start to appear. This **Multi-Tab system** comes turned on by default with VitalPBX, allowing you to navigate opened modules easily. You can close Tabs individually or close many by right-clicking on one. It is also possible to refresh a single tab in case you recently made a change in a different module.

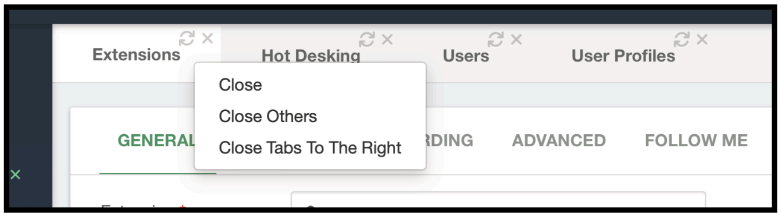
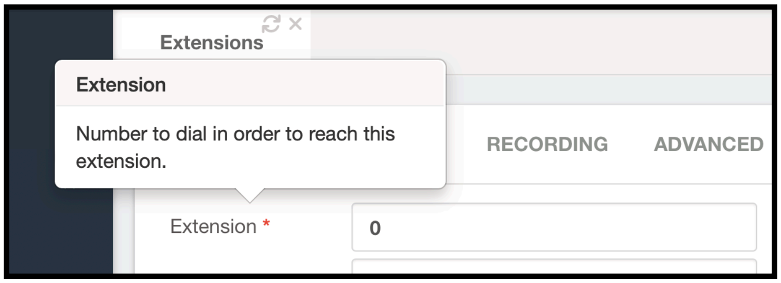


Image 2.1.11 - VitalPBX's tab system.

Finally, rest assured that you don't need to remember what every field in VitalPBX configures. Whenever you hover your mouse over a field title, you get a **Tool Tip**. This allows you to see a brief description of the field; most of them contain sample data for your understanding.



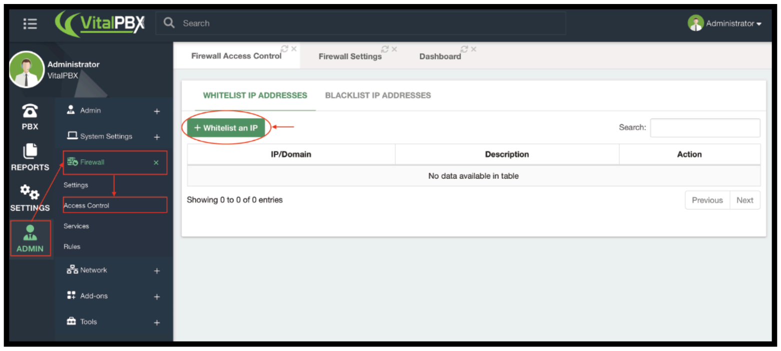
Extension

Number to dial in order to reach this extension.

Image 2.1.12 - Extension field tool tip.

Section 2, Lesson 2 - WhiteList IP Addresses.

Next, we need to add our IP addresses to the White List. This way, we won't get banned from the system. We can get blocked by the **Intrusion Detection system** if we enter the wrong password too many times in the web UI, phone device, or SSH connection.

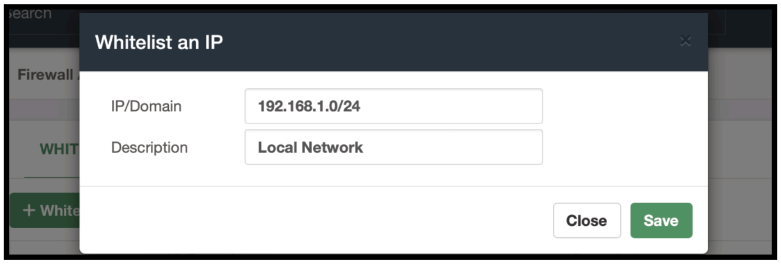


IP/Domain	Description	Action
	No data available in table	

Showing 0 to 0 of 0 entries

Image 2.2.1 - VitalPBX Firewall Access Control screen.

We will go to **Admin > Firewall > Access Control** to do this. On the **Whitelist IP Addresses** tab, we click the **+ Whitelist an IP** button.



IP/Domain: 192.168.1.0/24

Description: Local Network

Image 2.2.2 - Adding a network segment to the firewall whitelist.

Then, you can add any IP address you want to add to the list. This way, during your configuration, you don't get banned by the intrusion detection system by mistake.

For the **IP/Domain** field, you can add **individual IP addresses** or network segments using **CIDR** (Classless Inter-Domain Routing) format. For example, 192.168.1.0/24.

Once you have added the IP address or network segment, click **Save**.

Section 2, Lesson 3 - Users and User Profiles.

Next, we will look into how you can create new VitalPBX users and limit which modules they can access. This way, other users can log into VitalPBX, but you can restrict which options they can or can't configure.

Let's begin by creating a new **User Profile**. For example, let's create a profile that can only see the **CDR** (Call Detail Reporting). To do this, we will go to **Admin > Admin > User Profiles**.

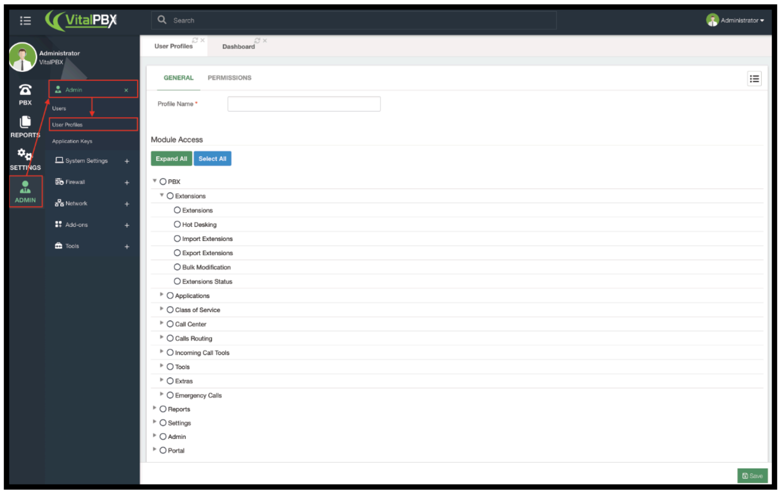
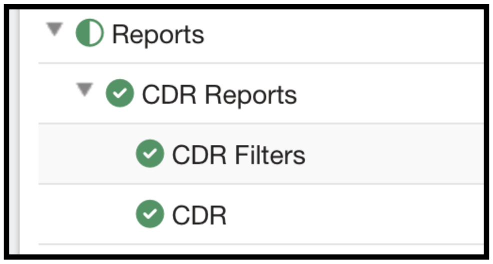


Image 2.3.1 - User Profiles, General Tab.

You can create a new User Profile by entering any name you desire under **Profile Name**. For this example, we can name it **CDR**.

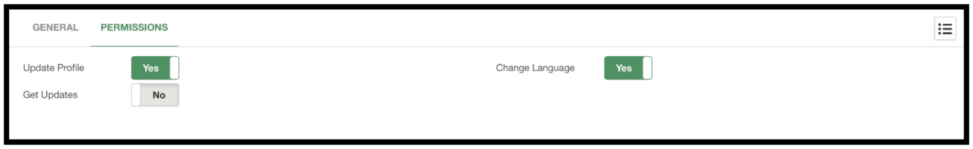
Then, on the lower part under **Module Access**, you can select the modules you wish to have access to with this User Profile. You can expand and select/deselect all the items with the buttons on top of this section. Here we can choose only the CDR modules for our sample profile.



- ◐ Reports
  - ✓ CDR Reports
    - ✓ CDR Filters
    - ✓ CDR

Image 2.3.2 - User Profiles, module access selection.

You can then head to the **Permissions** tab. Here, you can choose whether or not a user can update their profile, change their web UI language, or be able to update the system.

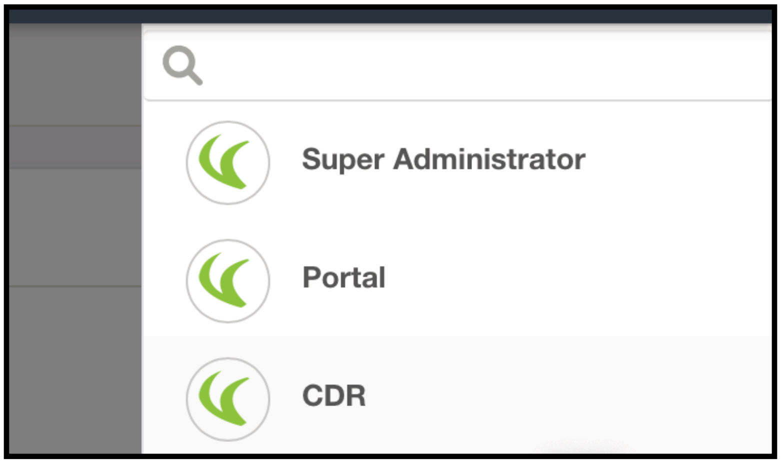


Update Profile	Yes	Change Language	Yes
Get Updates	No		

Image 2.3.2 - User Profiles, Permissions Tab.

We will leave this with the default values for this example. Click **Save**.

You will see your newly created profile listed with the default user profiles: **Administrator**, **Portal**, and if you use our **Multi-Tenant add-on**, **Tenant Administrator**.



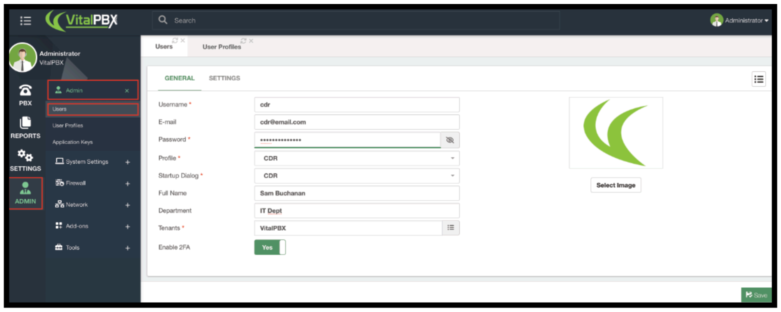
Super Administrator

Portal

CDR

Image 2.3.3 - User Profile list.

With the User Profile created, we can create a new user. For this, we will go to **Admin > Admin > Users**.



Username * cdr

E-mail cdr@email.com

Password * ••••••••••••••

Profile * CDR

Startup Dialog * CDR

Full Name Sam Buchanan

Department IT Dept

Tenants * VitalPBX

Enable 2FA Yes

Image 2.3.4 - Users module.

To make our user, we can configure the following.

- **Username -** This must not contain spaces or special characters.
- **Email -** This is an email address to identify this user.
- **Password -** The user's password to log in.
- **Profile -** We can select any User Profile we have created.
- **Startup Dialog -** This is the first screen the user sees when logging in.
- **Department -** This is used to identify the user further and is optional.
- **Tenants -** The tenant this user belongs to.
- **Enable 2FA -** This allows you to add Two-Factor Authentication to this user.
- **Profile Picture -** This is used to identify the user further with an image.

You can then click on **Save**.

Afterward, we go back to the user we just created, and in the bottom left-hand corner, we can see the option to add a **2FA (Two-Factor Authentication) application**. We recommend **Google Authenticator**.

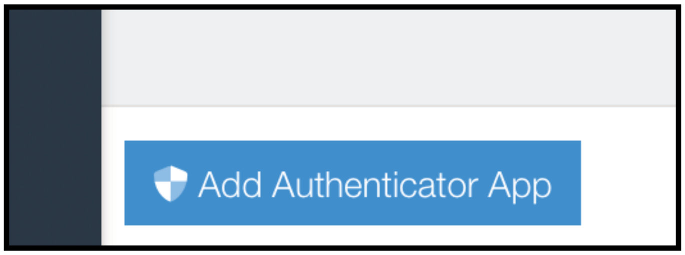
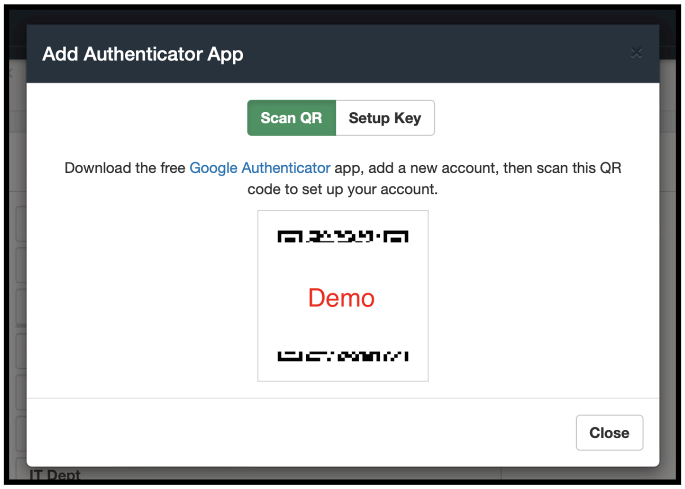


Image 2.3.5 - Add Authenticator App button.

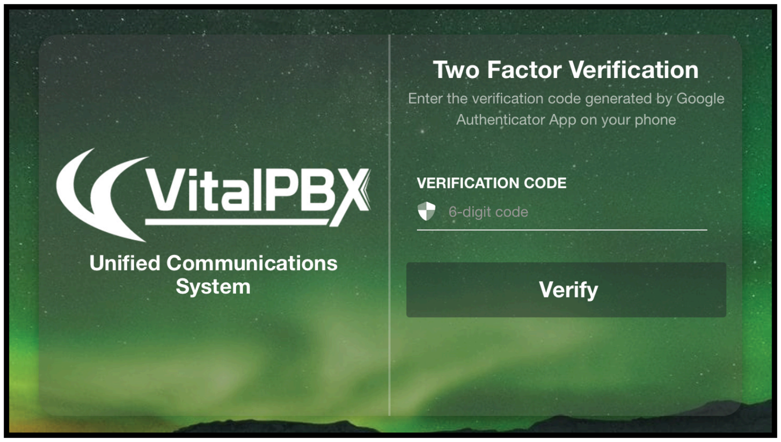
When you click on it, you can scan a **QR code** or use a **Setup Key** to configure the 2FA with your authenticator application.



Download the free Google Authenticator app, add a new account, then scan this QR code to set up your account.

Image 2.3.6 - Add authenticator app QR code prompt.

Once you have processed the QR code with your authenticator app, you can close the prompt, and when logging in with the user, you will be asked for a 2FA code to enter.



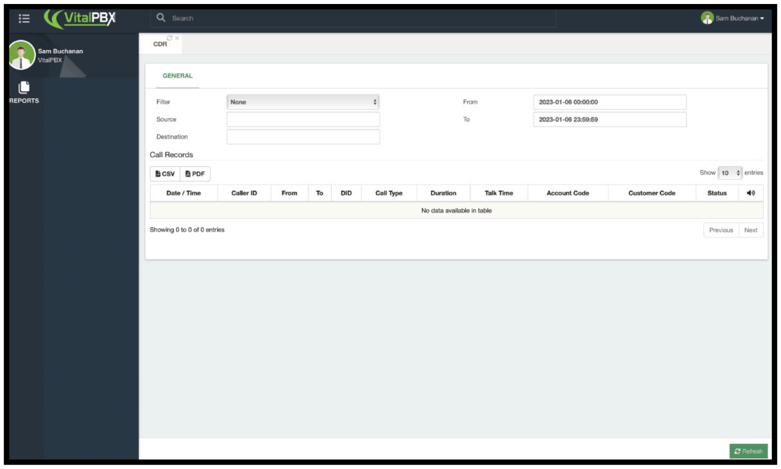
Two Factor Verification

Enter the verification code generated by Google Authenticator App on your phone

VERIFICATION CODE: 6-digit code

Image 2.3.7 - VitalPBX Login screen with 2FA active.

After you enter the code, you will successfully log in with the user. And you can see the effect of the User Profile you assigned to the user. In the case of this example, the user only has access to the CDR features.



Filter: None

Source:

Destination:

From: 2023-01-06 00:00:00

To: 2023-01-06 23:59:59

Call Records

Show 10 entries

Date / Time	Caller ID	From	To	DID	Call Type	Duration	Talk Time	Account Code	Customer Code	Status	🔊
No data available in table											

Showing 0 to 0 of 0 entries

Image 2.3.8 - Logged in with a new user with CDR access only.

Section 2, Lesson 4 - User Menu

Every user has access to the **user menu** located in the upper right-hand corner of the web UI for VitalPBX. Here, users can perform various actions concerning their profile and the system.

As we saw in the previous lesson, you can control the features you can see here. You can see every option available in this menu when logged in as the system administrator.

On a fresh installation, you have access to the following features.

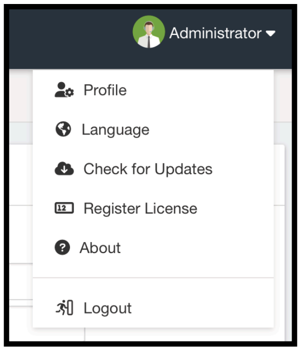
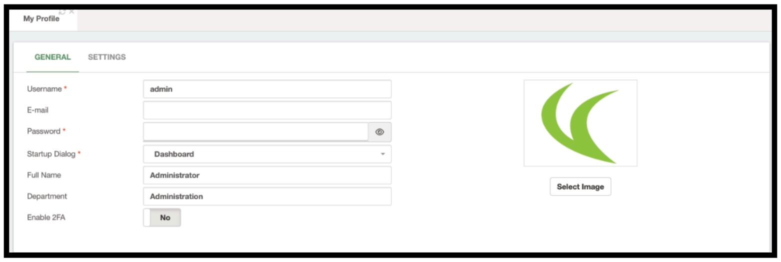


Image 2.4.1 - User menu.

When you click on **Profile**, you will see various options you can configure concerning your profile.

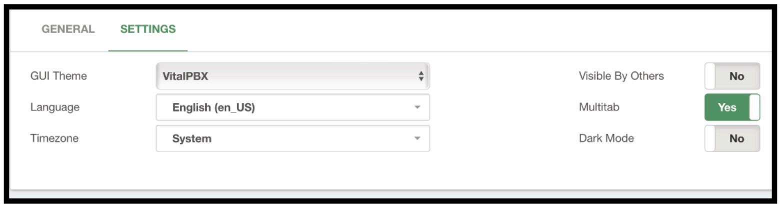


Username * admin
E-mail
Password *
Startup Dialog * Dashboard
Full Name Administrator
Department Administration
Enable 2FA No

Image 2.4.2 - My profile module.

You can change your **username, email, password, startup dialog, full name, and department and enable or disable 2FA**. You can also change your **profile picture** if you desire.

When you move over to the **Settings** tab, you can first change your web UI language, which only affects your user.



GUI Theme VitalPBX
Language English (en_US)
Timezone System
Visible By Others No
Multitab Yes
Dark Mode No

Image 2.4.3 - My Profile's settings tab.

If you are in a different timezone to the system, you can change it here whether or not your user is visible.

This is important if you don't want other users who can update user information to be able to modify your account. However, system administrators can see every user independently if this feature is used.

You can also enable or disable the **Multi-tab** feature.

And finally, you can enable **Dark Mode** if you enjoy a dark color palette for the interface. Keep in mind that if you enable Dark Mode, you must save the changes for the change to be persistent.
You can then save any changes in the bottom right-hand corner.

The next option in the User Menu is **Language**. This will change the web UI language. You can select from multiple options that VitalPBX has to offer. This, however, will not affect the voice prompts in the system. We will go over how you can influence this later on.

Afterward, we have the **Check for Updates** option. This is how you can update VitalPBX to the latest version. This way, updating the system through the Linux CLI (Command Line) is not obligatory.

Register License is how you register any of our Licensing Plans or the Starter License. We will go more in-depth on this in a later lesson.

The **About** option will give you information on your VitalPBX installation. This reflects the version you are running and the licensing type.

And Finally, you can **log out**.

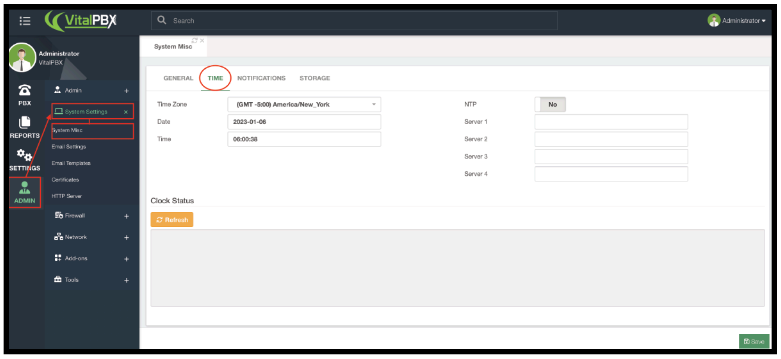
Additionally, quick links will appear here if you have any **Sonata Application or VitXi** installed.

Section 2, Lesson 5 - Time Zones and Email

Now, we will look into configuring the **Time Zone** for our system. This will be the base clock the system will use for any time-sensitive information like call dates and times in the CDR.

If you installed VitalPBX from the ISO on our website or made a Debian installation from scratch, the timezone was set at installation. However, you might need to change your timezone if you have used a VPS system that uses images to create your systems.

We will go to **Admin > System Settings > System Misc** to do this. From here, we will go to the **Time** tab. We will look into the other Tabs later.

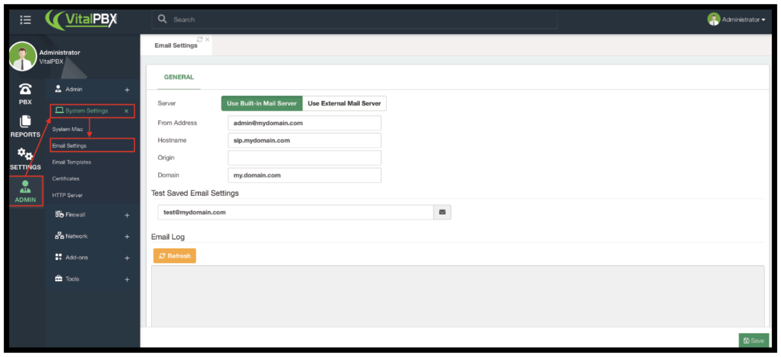


Time Zone (GMT -5:00) America/New_York
Date 2023-01-06
Time 06:00:38
NTP No
Server 1
Server 2
Server 3
Server 4
Clock Status

Image 2.5.1 - Time tab in the system miscellaneous module.

Here, you can change the time and date to your liking. You can also add **NTP servers** if you want specific servers for your system clock. In the bottom section, you can check the system's **Clock Status**. This will tell you the exact time for your system.

Next, let's look into configuring the **email client** for VitalPBX. We will go to **Admin > System Settings > Email Settings**.



Server: Use Built-in Mail Server
From Address: admin@mydomain.com
Hostname: sip.mydomain.com
Origin:
Domain: my.domain.com

Test Saved Email Settings
test@mydomain.com

Email Log

Image 2.5.2 - Email settings module.

In this module, you can configure the **external email client** or **built-in server** to send email messages from your VitalPBX installation. This is important so you can send notifications and emails concerning various configurations from your PBX.

This will be useful for sending welcome emails from templates we will look into later. This can also help with Voicemail to Email, Fax to Email, System Notifications, and more.

You can pick between using the built-in server or an external mail server. Additionally, you can use a Gmail account with the **External Mail Server** option. If you go with this option, make sure you create an application password for your Gmail account, or else email sending might fail.

Once you have entered your configurations, click **Save**.

You can test your configurations by entering an email address in the **Test Saved Email Settings** field and clicking the **envelope** button to its right.

In the **Email Log** section, if you click on **Refresh**, you will see the email log output, and you can use this to troubleshoot if there is an issue with receiving the test email.

Section 2, Lesson 6 - Notifications and Monitoring.

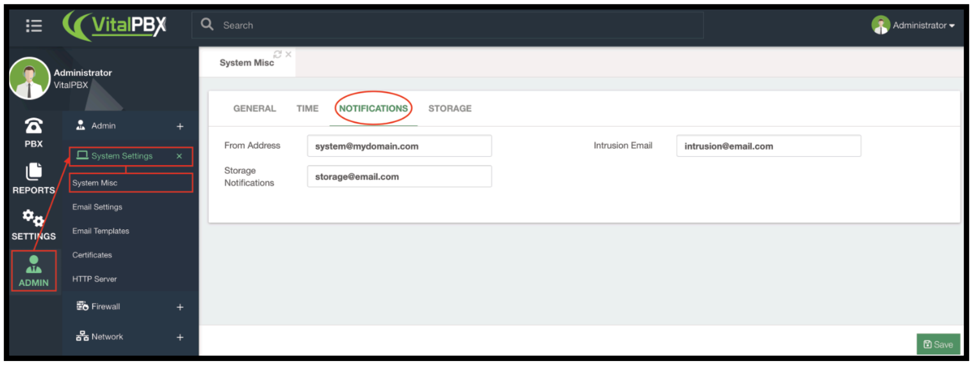
With our email settings in place, we will now configure various notifications available for VitalPBX.

For this, we will go to **Admin > System Settings > Monitoring**. You will find the **Notifications** and **Storage** tabs here.

In the **Notifications** tab, you can define a **From Address** from which the emails will be sent. This can be a different "From" address than the email you configured in the email settings.

Make sure you use the same domain as the one configured in your email settings and that your email server permits this.

You can also enter the emails to which **Storage** and **Intrusion Notifications** will be sent.



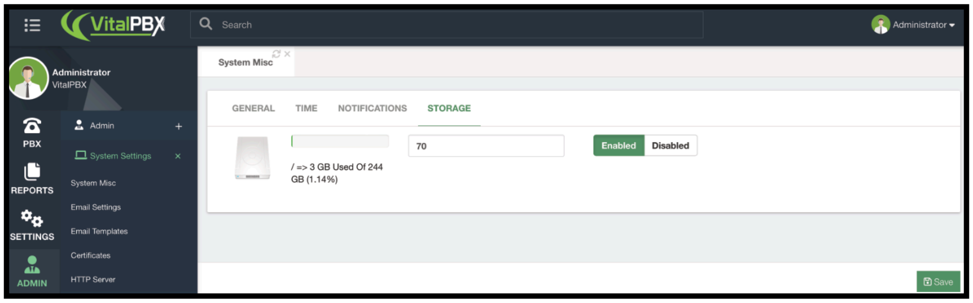
From Address: system@mydomain.com
Intrusion Email: intrusion@email.com
Storage Notifications: storage@email.com

Image 2.6.1 - System Miscellaneous's notifications tab.

You can then click on **Save**.

Next, we can move to the **Storage** tab.

You can configure the threshold to send you the storage notification email and enable/disable this notification.



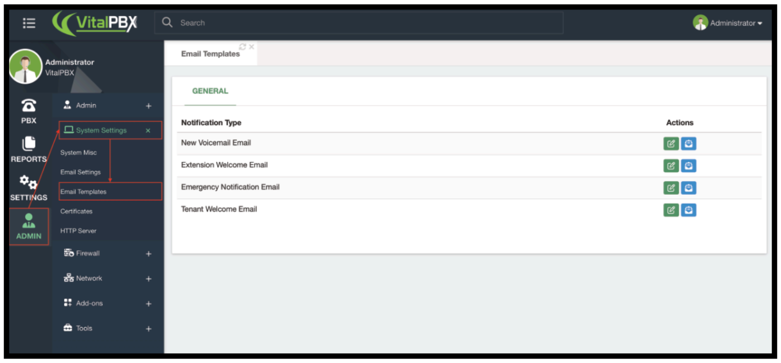
/ => 3 GB Used Of 244 GB (1.14%)
70
Enabled Disabled

Image 2.6.2 - System Miscellaneous's storage tab.

Once you set your setting, click on **Save**.

Section 2, Lesson 7 - Email Templates

VitalPBX has various Email Templates available for different types of notifications. These templates can be customized to fit your standards and branding. To customize the email templates, we will need to go to **Admin > System Settings > Email Templates**.



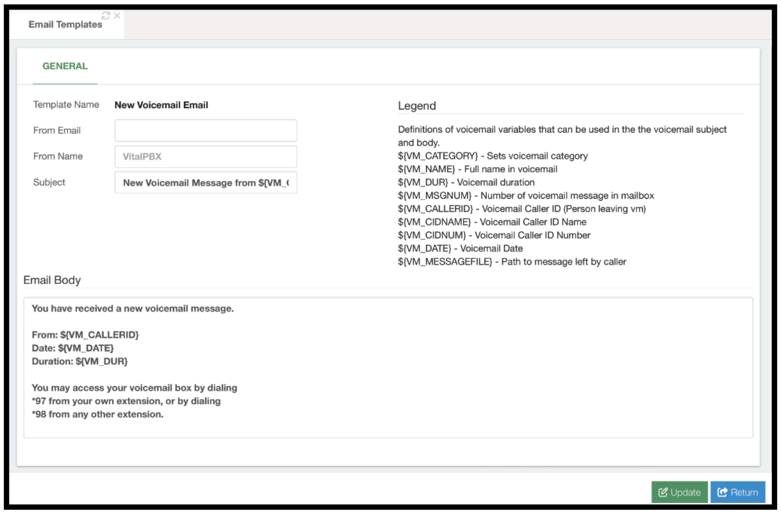
Notification Type	Actions
New Voicemail Email	
Extension Welcome Email	
Emergency Notification Email	
Tenant Welcome Email	

Image 2.7.1 - Email templates module.

Here, you can customize 5 email templates.

- **New Voicemail Email -** This is an email sent to an extension's email address(es) when an incoming call is sent to voicemail.
- **Extension Welcome Email -** This is an email notification sent to a user when their extension is created.
- **Emergency Notification Email -** This is an email notification sent when an Emergency Number is dialed.
- **Tenant Welcome Email -** This is an email notification sent to a new Tenant Administrator user when a tenant is created for them.
- **Missed Call -** This is an email sent to an extension user when a call is missed.

For these templates, you have two options under the **Action** column. You can edit the template, or you can send a test email to test how the email looks. Let's take a look at the *New Voicemail Email* template first. Click on the green **Edit** button in the *Action* column next to it.



Template Name: New Voicemail Email
From Email:
From Name: VitalPBX
Subject: New Voicemail Message from ${VM_(

Legend

Definitions of voicemail variables that can be used in the the voicemail subject and body.
${VM_CATEGORY} - Sets voicemail category
${VM_NAME} - Full name in voicemail
${VM_DUR} - Voicemail duration
${VM_MSGNUM} - Number of voicemail message in mailbox
${VM_CALLERID} - Voicemail Caller ID (Person leaving vm)
${VM_CIDNAME} - Voicemail Caller ID Name
${VM_CIDNUM} - Voicemail Caller ID Number
${VM_DATE} - Voicemail Date
${VM_MESSAGEFILE} - Path to message left by caller

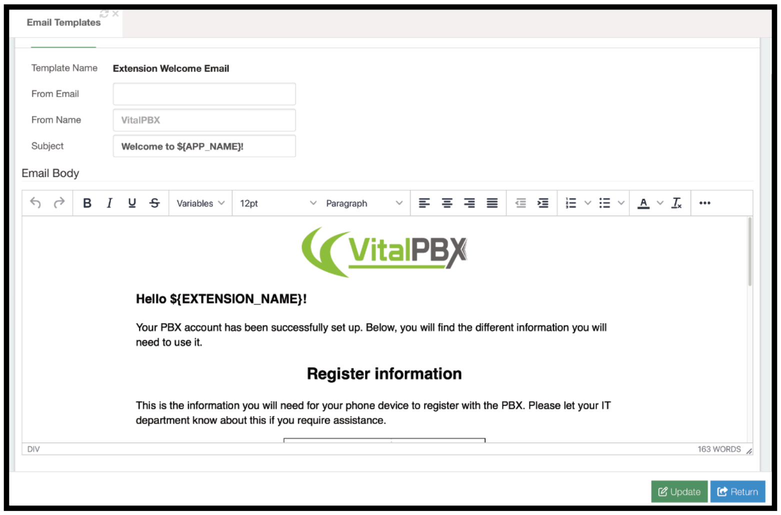
Email Body

```
You have received a new voicemail message.

From: ${VM_CALLERID}
Date: ${VM_DATE}
Duration: ${VM_DUR}

You may access your voicemail box by dialing
*97 from your own extension, or by dialing
*98 from any other extension.
```

Image 2.7.2 - Voicemail Email template.

Here, you can edit the **From Email and Name**, as well as the **Subject and Body**. To the right, we have a list of variables we can use to have dynamic information in our template. For now, we will leave these with the default values. If you make any changes, you can click on the green **Update** button in the bottom-right corner. To return to the email templates list, click on the blue **Return** button.

Now, let’s look at the **Extension Welcome Email**. Similarly, click on the green **Edit** button under the *Actions* column for this template.



Email Templates

Template Name	Extension Welcome Email
From Email	
From Name	VitalPBX
Subject	Welcome to ${APP_NAME}!

Email Body

Hello ${EXTENSION_NAME}!

Your PBX account has been successfully set up. Below, you will find the different information you will need to use it.

Register information

This is the information you will need for your phone device to register with the PBX. Please let your IT department know about this if you require assistance.

Image 2.7.3 - Extension Welcome Email template.

As you can see, this one looks a bit different than the *Voicemail Email template*. This is how the **Extension Welcome, Emergency Notification,** and **New Tenant** email templates look when you edit them.

You can still change the **From Name, From Email,** and **Subject** in the section above where we find the **Email Body**. The body uses a complete WYSIWYG (what you see is what you get) editor. This supports different font styling, links, and a full HTML editor. This way, you can make these welcome and notification emails as attractive and on-brand as you want.

By default, these come will a default template, but you can edit this to anything you like.

You will see that we still use variables to make the email contents dynamic. These can be found in the **Variables** dropdown menu. You can place these variables anywhere in the email body or subject line.

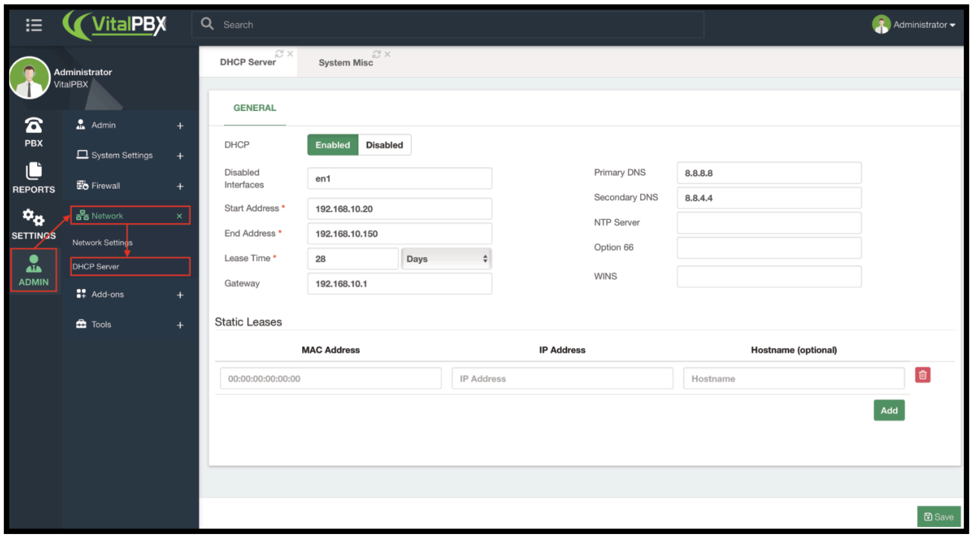
For now, we can leave these with their default content. If you make any changes, you can press the **Update** button in the lower right-hand corner.

As mentioned before, the **Emergency Notification** and **New Tenant** email templates work the same way as the **Extension Welcome**. Feel free to work on the different designs you can create incorporating the available variables to customize the emails sent.

Section 2, Lesson 8 - DHCP Configurations

Sometimes, you must work in an isolated network for your phone system. In these cases, you might be working solely with a network switch. Fortunately, VitalPBX has a built-in DHCP Server to give your devices an IP address.

To configure the DHCP settings, we must go to **Admin > Network > DHCP Server**.



DHCP: Enabled | Disabled

Disabled Interfaces	en1	Primary DNS	8.8.8.8
Start Address *	192.168.10.20	Secondary DNS	8.8.4.4
End Address *	192.168.10.150	NTP Server	
Lease Time *	28 Days	Option 66	
Gateway	192.168.10.1	WINS	

Static Leases

MAC Address	IP Address	Hostname (optional)
00:00:00:00:00:00	IP Address	Hostname

Image 2.7.1 - DHCP Server module.

You can set up a complete DHCP pool with an **IP address range**, **Lease Time**, **Gateway**, **DNS Servers**, and **NTP Server**. Additionally, you have **Option 66,** which defines the "Phone Provisioning URL," so your phone devices pull provisioning information and a **WINS** server. Under **Static Leases,** you can define static IP addresses for specific devices using their MAC address.

You can prevent the DHCP server from serving IP addresses to specific interfaces if you are connected with multiple network interfaces.

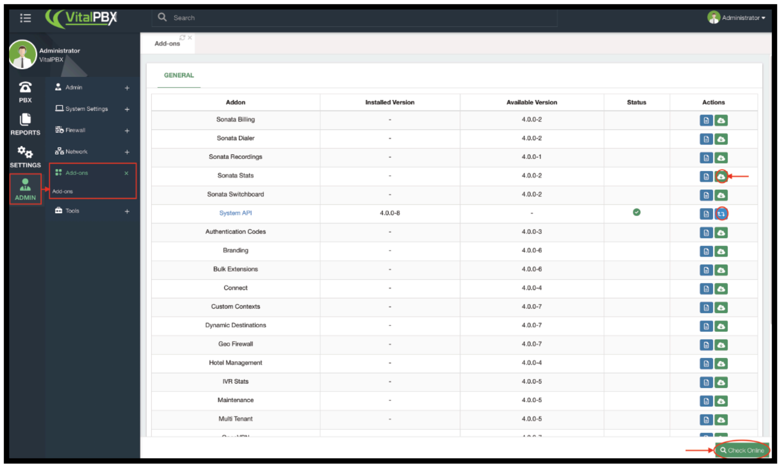
Ensure there is no other DHCP server in the network to avoid conflicts. Then, click "Save."

Section 2, Lesson 9 - Installing Add-Ons.

One of the significant advantages of VitalPBX is the ability to expand the features with our add-ons.

To install add-ons, we will first go to **Admin** > **Add-Ons** > **Add-Ons**.
Here, we will see the list of the add-ons that are currently available for installation. The first time you open this module, the list will be empty. Click the **Check Online** button to refresh the list with the latest add-ons.

To install the add-on, click the green **Download** button next to the add-on you wish to install. If an update is available, you will see an **Update** button. The blue arrow button allows you to reinstall an add-on. Clicking the blue **Information** button will show you information about the add-on. Once an add-on is installed, you will see a red **Uninstall** button.



Addon	Installed Version	Available Version	Status	Actions
Sonata Billing	-	4.0.0-2		🗎 ☁
Sonata Dialer	-	4.0.0-2		🗎 ☁
Sonata Recordings	-	4.0.0-1		🗎 ☁
Sonata Stats	-	4.0.0-2		🗎 ☁
Sonata Switchboard	-	4.0.0-2		🗎 ☁
System API	4.0.0-8	-	✓	🗎 ⟳
Authentication Codes	-	4.0.0-3		🗎 ☁
Branding	-	4.0.0-6		🗎 ☁
Bulk Extensions	-	4.0.0-6		🗎 ☁
Connect	-	4.0.0-4		🗎 ☁
Custom Contexts	-	4.0.0-7		🗎 ☁
Dynamic Destinations	-	4.0.0-7		🗎 ☁
Geo Firewall	-	4.0.0-7		🗎 ☁
Hotel Management	-	4.0.0-4		🗎 ☁
IVR Stats	-	4.0.0-5		🗎 ☁
Maintenance	-	4.0.0-5		🗎 ☁
Multi Tenant	-	4.0.0-5		🗎 ☁

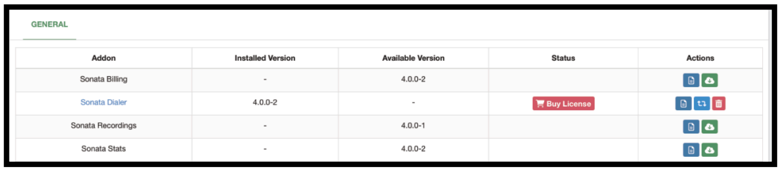
Image 2.9.1 - Add-Ons module.

Section 2, Lesson 10 - Licensing.

Many of our add-ons require a commercial license. These licenses can be purchased individually, or you can license multiple add-ons with a Licensing Plan Subscription.

You can purchase individual licenses through the store on our website at https://www.vitalpbx.com/vitalpbx-store/.

To apply the license to the add-on, we will go to **Admin** > **Add-Ons** > **Add-Ons**. Here, you will see the add-ons available. Be sure that you have installed the add-on first. Next to your installed add-on, you will see a red **Buy License** button.

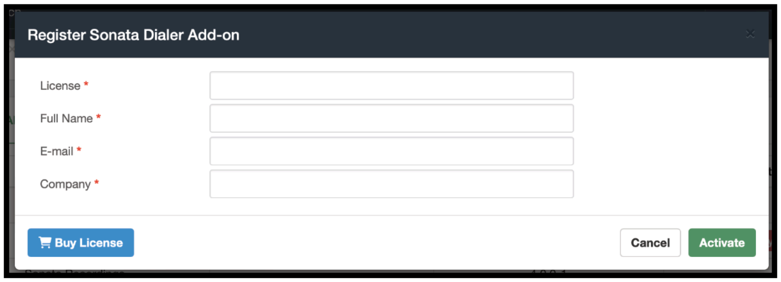


Addon	Installed Version	Available Version	Status	Actions
Sonata Billing	-	4.0.0-2		🗎 ☁
Sonata Dialer	4.0.0-2	-	Buy License	🗎 ⟳ 🗑
Sonata Recordings	-	4.0.0-1		🗎 ☁
Sonata Stats	-	4.0.0-2		🗎 ☁

Image 2.10.1 - Add-on with a Buy License button.

When you click the Buy License button, you will see a form to enter the licensing information.

You must enter the **License Key** sent to your email when you purchased the license. As well as your **name**, **email**, and **company** name. Then, click on **Activate**.



Register Sonata Dialer Add-on

License *	
Full Name *	
E-mail *	
Company *	

Image 2.10.2 - Add-on license registration form.

You can then refresh your browser and have the add-on registered.

If you subscribe to one of our licensing plans at https://www.vitalpbx.com/pbx-system-plans-and-pricing/, you must do the following.

Log in as the system administrator, and go to the **User Menu** in the upper right-hand corner. Click on **Register License**.

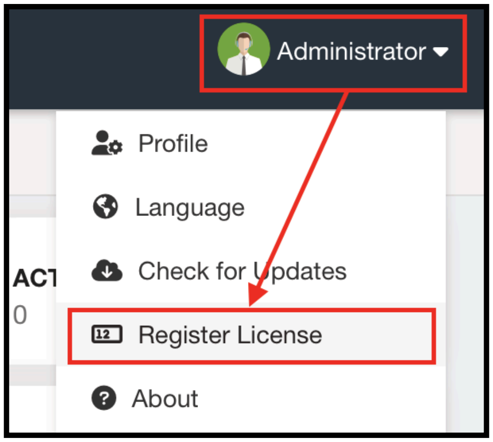
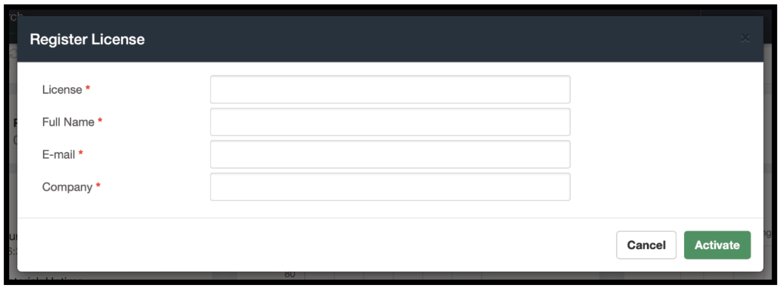


Image 2.10.3 - User Menu's Register License option.

You will then see a form to register your purchased plan.



Register License

License *	
Full Name *	
E-mail *	
Company *	

Image 2.10.4 - License registration form.

Here, you will enter the **License Key** sent to your email when you subscribed to a Licensing Plan. As well as your **name**, **email**, and **company** name. Finally, click on **Activate**.

This process will be the same if you have purchased a **Starter License**.

Remember that all of our commercial add-ons can be installed even if you don't have a license. You get access to all of their features for an unlimited time. The only limitation without an add-on or plan license is the number of items you can create within an add-on.

This way, you can test out all of our add-ons without worrying about features or time limits.

Conclusion

With this, we conclude with our initial configurations and overview of VitalPBX. We recommend that you go through this process whenever you do a new installation of VitalPBX.

This will ensure that you work in a proper environment and run into fewer issues down the line when configuring your PBX.

Now, let's continue with the rest of our VitalPBX configuration.

SECTION 3 - EXTENSIONS

Introduction

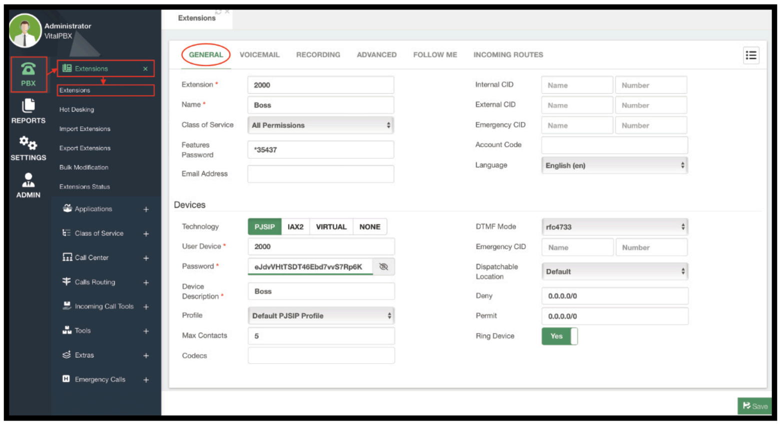
Let's begin with the essential feature and the bulk of the work for every PBX, the extensions. Configuring extensions in VitalPBX is a straightforward task.

Extensions are full of features that give you a rich experience when using a VitalPBX system. We will go through every option that you can configure so you understand the full capabilities of VitalPBX extensions.

Throughout this section, you will learn everything about how to configure, manipulate, register, and customize your extensions and the different modules around them.

Section 3, Lesson 1 - General Configurations

To start configuring your extensions, we will go to **PBX** > **Extensions** > **Extensions**.



Extensions

Extension *	2000
Name *	Boss
Class of Service	All Permissions
Features Password	*35437
Email Address	
Internal CID	Name / Number
External CID	Name / Number
Emergency CID	Name / Number
Account Code	
Language	English (en)

Devices

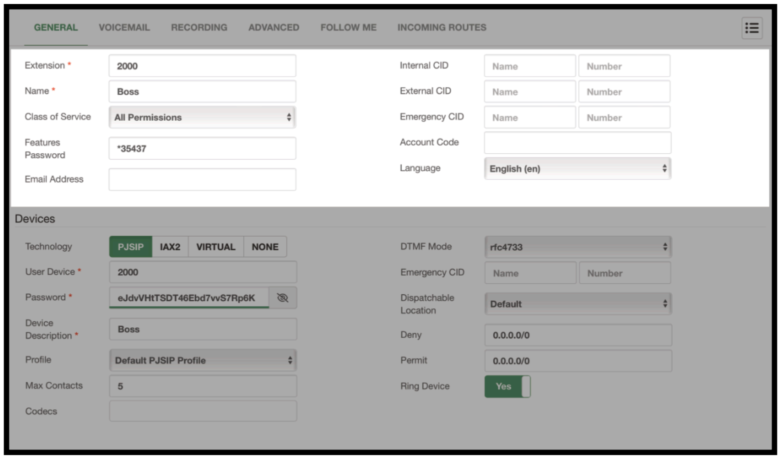
Technology	PJSIP / IAX2 / VIRTUAL / NONE (PJSIP selected)
User Device *	2000
Password *	eJdvVHtTSDT46Ebd7vvS7Rp6K
Device Description *	Boss
Profile	Default PJSIP Profile
Max Contacts	5
Codecs	
DTMF Mode	rfc4733
Emergency CID	Name / Number
Dispatchable Location	Default
Deny	0.0.0.0/0
Permit	0.0.0.0/0
Ring Device	Yes

Image 2.3.1 - Extensions module.

We will see the general configurations for our extensions on the General tab. Here we can create the extension number and its devices.

With VitalPBX, we use a multi-device technology so you can have multiple devices for the same extension number.
This is why the extensions module general tab is divided into the extension number and the devices section.
Let's begin by adding an **Extension Number** in the Extension Number Section. This number can be any set of numbers of the length you desire.

Then, add a **Name** to identify this extension and select a **Class of Service**. We will go more into classes of service in a later section.

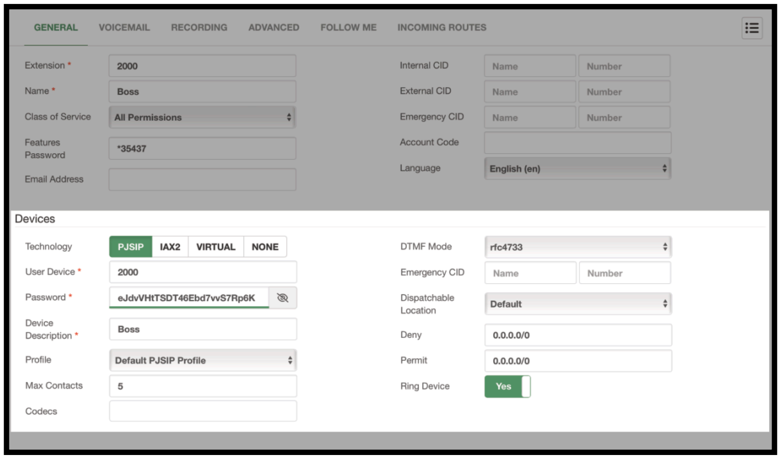


Extension * 2000
Name * Boss
Class of Service All Permissions
Features Password *35437
Email Address
Internal CID Name Number
External CID Name Number
Emergency CID Name Number
Account Code
Language English (en)

Devices

Technology PJSIP IAX2 VIRTUAL NONE
User Device * 2000
Password * eJdvVHtTSDT46Ebd7vvS7Rp6K
Device Description * Boss
Profile Default PJSIP Profile
Max Contacts 5
Codecs
DTMF Mode rfc4733
Emergency CID Name Number
Dispatchable Location Default
Deny 0.0.0.0/0
Permit 0.0.0.0/0
Ring Device Yes

Image 2.3.2 - Extension number section.



Extension * 2000
Name * Boss
Class of Service All Permissions
Features Password *35437
Email Address
Internal CID Name Number
External CID Name Number
Emergency CID Name Number
Account Code
Language English (en)

Devices

Technology PJSIP IAX2 VIRTUAL NONE
User Device * 2000
Password * eJdvVHtTSDT46Ebd7vvS7Rp6K
Device Description * Boss
Profile Default PJSIP Profile
Max Contacts 5
Codecs
DTMF Mode rfc4733
Emergency CID Name Number
Dispatchable Location Default
Deny 0.0.0.0/0
Permit 0.0.0.0/0
Ring Device Yes

Image 2.3.3 - Devices section.

Next, you can add an **Email Address**. This email address is used to notify the extension user or users, send them a **Welcome Email**, and use the **Voicemail to Email** feature. We will see later how to modify the welcome email.

You then have the **Internal**, **External**, and **Emergency** caller IDs. You can leave these blank. The Internal CID will populate with the extension name and number. You can customize the External and Emergency CIDs for outgoing calls through a Trunk. Later on, we will go more into how Emergency CIDs work. Finally, you can set the **Language** for the voice prompts for this extension.
Now, we can move over to the Device Section. Here we can create the user and password to register to our PBX. This device will be associated with the extension number above. So, whenever the extension number is dialed, this device will ring.

The first option we see is the technology for the device. Here you can choose between **PJSIP**, **IAX2**, **Virtual**, or **None**.

Note…

*With **VitalPBX 4 and onward**, we have moved entirely into **a PJSIP environment instead of SIP**. This is due to SIP being deprecated from the latest Asterisk versions and no longer supported.*

*This means that PJSIP uses **port 5060** with every VitalPBX 4 installation by **default**.*

Every SIP-based device or trunk will work with a PJSIP device or trunk created in VitalPBX. So there is no need to worry about compatibility with your existing or newer devices.

IAX2 is another classic device type for Asterisk. **Virtual** devices can be any destination number, i.e., a cellphone number, that will be dialed when the extension number is called. **None** is when you want to create an extension with no device. This is useful when you want to make a Virtual Extension that is used as a common Voicemail destination, for example.

Once we have selected our preferred technology, we can enter our **User Device** and **Password**. This is the username and password we will use to register our device. By default, a random password is generated. We can then add a **Description** to identify the device.

Next, we can select a **Device Profile**. We will go more into what this is later on.

We can also choose the number of **Max Contacts** that can register using the same user device and password we created previously. With PJSIP, you can have multiple registrations for the same user device.

You can then choose specific **Codecs** for your device. All available codecs will be used if you leave this blank.

Then, you can change the **DTMF Mode** on the right side if necessary.

It is possible to assign a device-specific **Emergency CID** and a **Dispatchable Location**. We will go further into these options in the Emergency Calls section.

Next, we have the **Deny** and **Permit** fields. Here, you can enter specific IP addresses or network segments where you can or cannot register to this User Device.

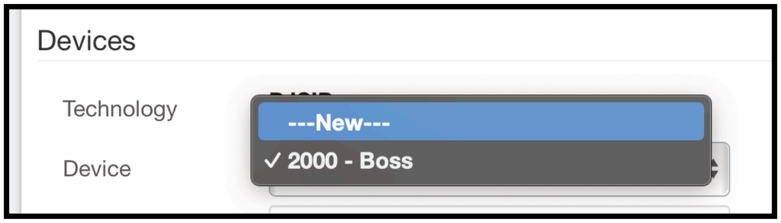
And finally, we can enable or disable **Ring Device**. This is a Multi-Device PBX system, so you might have multiple devices with the same extension number. So, you can choose which ones can ring.

With everything entered, click **Save**, and don't forget to **Apply Changes**.

You don't need to complete every single field to create an extension. By simply entering an extension number and name, the rest of the areas are populated and can be left by default.

You can return to the extension from the list in the top right-hand corner with the extension and device created.

While editing the extension, you will see that there is now a device drop-down menu.



Devices

Technology

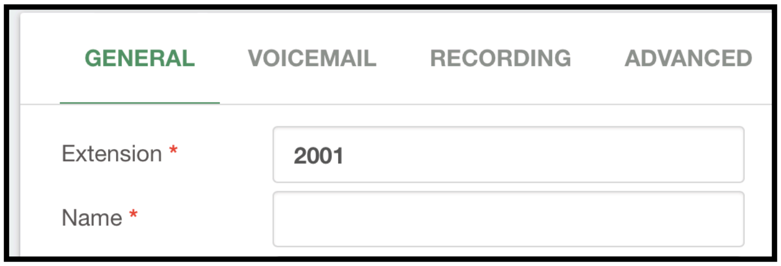
Device

Image 2.3.4 - Device drop-down menu.

Click on New to add another device to the same extension number.

You will also see a **Remove Device** button in the lower left-hand corner. This will remove a device from the extension. The device will now appear on the **Hot Desking** module, but we will see more about Hot Desking later.

Now that you have an extension number, whenever you create a new extension, you will see that the **Extension** field is populated with the following sequential number based on the extension number you have used.



Extension * 2001

Name *

Image 2.3.5 - Sequential extension numbers.

If you enter an existing extension number, you will be taken to the edit page for that extension.

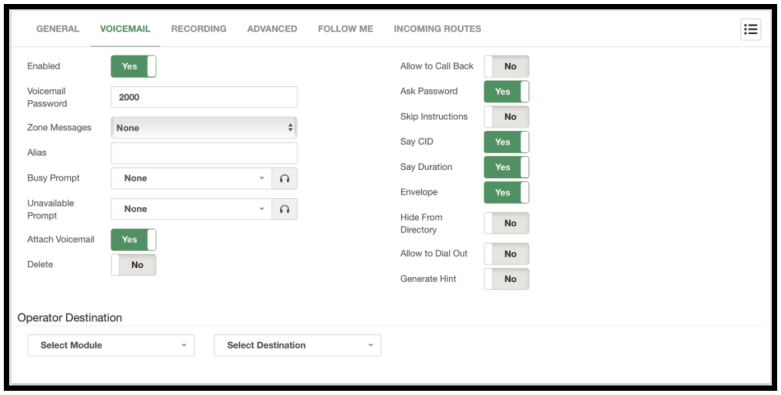
With this, you have created your first extension. Congratulations! We can configure some other features to make them more complex.

Section 3, Lesson 2 - Voicemail

Voicemail is a neat feature that comes with VitalPBX out of the box. Let's configure it for our extensions so we can receive a message whenever we are unavailable.

To do so, let's return to our extensions under **PBX** > **Extensions** > **Extensions**. Select an extension we have already created or create a new one.

Go to the **Voicemail** tab next to General.



Enabled: Yes
Voicemail Password: 2000
Zone Messages: None
Alias:
Busy Prompt: None
Unavailable Prompt: None
Attach Voicemail: Yes
Delete: No

Allow to Call Back: No
Ask Password: Yes
Skip Instructions: No
Say CID: Yes
Say Duration: Yes
Envelope: Yes
Hide From Directory: No
Allow to Dial Out: No
Generate Hint: No

Operator Destination

Select Module
Select Destination

Image 3.2.1 - Voicemail tab for an extension

The first option is to **Enable** or **Disable** voicemail for this extension. By default, this comes on for all extensions. Later on, we will see how we can turn this off by default.

Next, there's the **voicemail password**. If this is set to the same number as the extension number, the user must follow the initial setup process when entering their voicemail box. This is useful if you wish to reset the voicemail box for a user.

It is possible to choose a **Zone Message**; this will dictate how the voicemail envelope is read with the time format. Later on, we will see how you can create a zone message.

You can change the **alias** for the Dial by Name feature and have custom **Busy** and **Unavailable** voicemail prompts.

You can then Enable or Disable the **Attach Voicemail** and **Delete** options, which are part of the **Voicemail to Email** feature. So you can attach a .wav file to the email and delete it from the system once it is sent.

Then, you have multiple permissions to

- **Allow to Call Back** the voicemail sender.
- **Ask for the Password** when entering this extension’s voicemail box.
- **Skip Instructions,** so the caller won’t hear the instructions for leaving a message.
- **Say CID** before the voicemail message.
- **Say the Duration** of the voicemail message.
- Enable or disable the **Envelope playback** before a message.
- **Hide from the Directory** so the extension cannot be found with the Dial by Name feature.
- **Allow to Dial Out** from the voicemail box.
- **Generate a Hint** so other users can monitor this extension’s voicemail box.

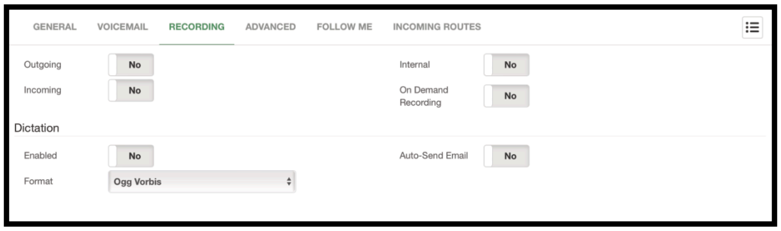
Ultimately, you have the **Operator Destination**, the destination when the caller dials “0” after leaving a message.

Remember to **Save/Update** and then **Apply Changes**.

Section 3, Lesson 3 - Extension Call Recording

Call recording is an essential feature that comes with VitalPBX out of the box. This feature can be enabled at many parts of the call process.

Let’s look at how we can do this at the extension level. We will go back to **PBX** > **Extensions** > **Extensions**. Select an existing extension or create a new one, then go to the **Recording** tab.



Outgoing	No	Internal	No
Incoming	No	On Demand Recording	No

Dictation

Enabled	No	Auto-Send Email	No
Format	Ogg Vorbis		

Image 3.3.1 - Recording tab in an extension.

Here, you can enable and disable call recordings for this extension. **Outgoing** and **Incoming** recordings are for calls coming in and out of the PBX through a trunk. **Internal** is for an extension to an extension call. **On-Demand Recording** allows you to use the ***3** feature code to record calls on demand.

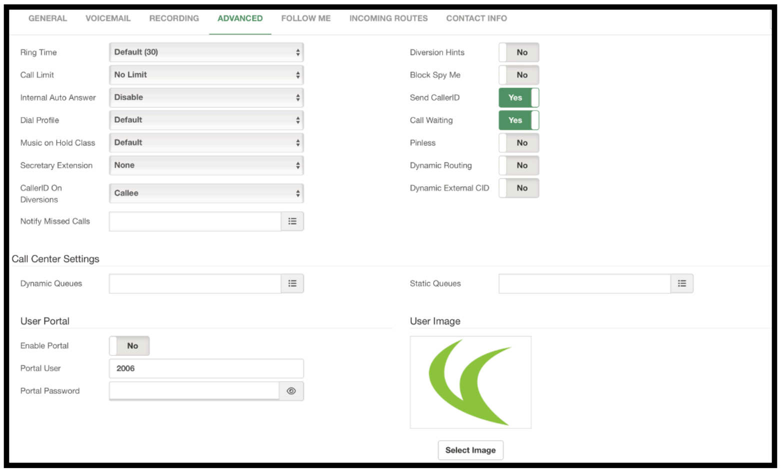
The **Dictation** section concerns the dictation feature, ***93**. This is a voice note reminder feature in VitalPBX. This allows you to record an audio file in the **Format** you choose here and **Auto-Send** it to your **Email**. This is the perfect tool to create quick notes directly from your registered device.

Once you have enabled the call recording types you want, you can click **Save/Update** and **Apply Changes**.

We will look into how you can listen to these recordings later.

Section 3, Lesson 4 - Advanced Extension Settings

Extensions have additional parameters you can configure to manage them more granularly. These options are available under **PBX > Extensions > Extensions**, and click on **Advanced**.



Ring Time	Default (30)	Diversion Hints	No
Call Limit	No Limit	Block Spy Me	No
Internal Auto Answer	Disable	Send CallerID	Yes
Dial Profile	Default	Call Waiting	Yes
Music on Hold Class	Default	Pinless	No
Secretary Extension	None	Dynamic Routing	No
CallerID On Diversions	Callee	Dynamic External CID	No
Notify Missed Calls			

Call Center Settings

Dynamic Queues		Static Queues	

User Portal

Enable Portal	No
Portal User	2006
Portal Password	

User Image

Image 3.4.1 - Advanced tab in an extension.

In this section, you can modify the following options.

- Change the **Ring Time**, so you can select how long the extension rings before going to voicemail or hanging up.
- Change the **Call Limit** of the extension's number of calls at once.
- Do an **Internal Auto Answer** to answer internal calls in the speaker automatically.
- Change the **Dial Profile** for transfer and ringback properties.
- Change the **Music on Hold Class** that plays when the user places a call on hold.
- Assign the **Secretary Extension**. We will see more about this feature later.
- You can also select the **Caller ID sent on Diversions**. This allows you to choose whether you send the caller's CID or the extension number when you have a call forward active.
- **Notify Missed Calls** using the Missed Calls email template. You can choose to send External, Internal, Queue, and/or Ring Group calls.
- Enable **Diversion Hints** if you need to monitor the diversion status from another extension. This comes set to **No** by default, as it adds a load to the system.
- You can **Block Spy Me** if you don't want anyone to use the Spy feature with this extension. This is useful for administrative extensions in a Call Center environment.
- Enable or Disable **Send Caller ID** if you wish to keep your extension anonymous within the company.
- Enable or Disable **Call Waiting** if you want to send an incoming call while the extension is busy to voicemail or hangup.
- **Pinless** will allow you to use various features with a PIN without entering any PIN.
- The **Dynamic Routing** feature allows you to automatically route calls with callers you were conversing with. For example, if you are on a call with a client and the call ends, the next time the caller calls, they will be routed back to the extension instead of going to a predefined destination, like an IVR or an operator.
- Lastly, you can enable or disable a **Dynamic External ID**. This is useful when you use the extension as a trunk from a third-party system—using the incoming call's CID as the Caller ID.

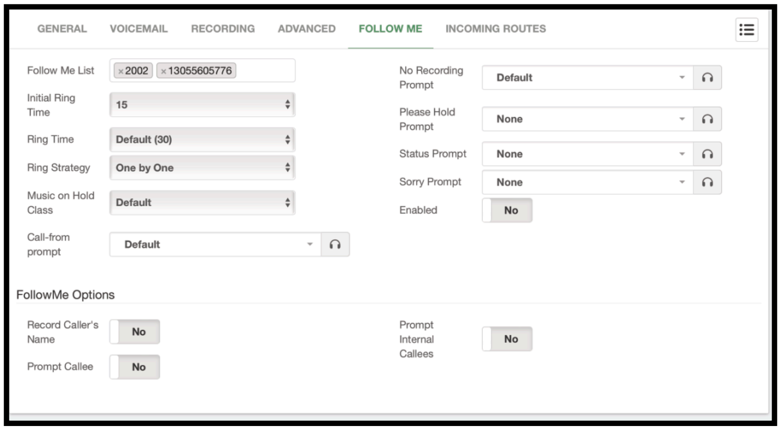
Next, you have **Call Center Settings**. This allows you to easily add the extension to multiple **queues** as a **Dynamic** or **Static** agent.

Finally, there's the **User Portal** section. Here you can define a **username**, **password**, and **profile picture** so the user can access the User Portal and manage their extension options. We will be taking a look into this User Portal later. You can enable and disable this feature at any time.

Section 3, Lesson 5 - Follow Me

Follow Me is a very nice feature if you or your users have a lot of mobility. With Follow Me, you can establish a thread of numbers that will be dialed if you are unavailable on your usual extension.

To configure the Follow Me options for an extension, we must go to **PBX > Extensions > Extensions**, and in the extension, go to the Follow Me tab at the top.



Follow Me List: ×2002 ×13055605776
Initial Ring Time: 15
Ring Time: Default (30)
Ring Strategy: One by One
Music on Hold Class: Default
Call-from prompt: Default
No Recording Prompt: Default
Please Hold Prompt: None
Status Prompt: None
Sorry Prompt: None
Enabled: No

FollowMe Options

Record Caller's Name: No
Prompt Callee: No
Prompt Internal Callees: No

Image 3.5.1 - Follow Me tab in an extension.

Here, you can establish your **Follow Me List**, which is the list of numbers to which the incoming call will be forwarded. When you are entering a number here, be sure to press the **Enter** key to add it to the list.

Then, you can set up the **Ring Time**, which is the amount of time in seconds the call will ring the extension before going through the Follow Me List.

You can also change the **Ring Strategy**, which is how the application will ring the Follow Me List. This can be done **One by One** or **Ring All**.

It is possible to set the **Music on Hold Class** that will play to the Caller while they wait for the call to connect with any number in the Follow Me List.

Next, you can change the audio prompts that will play back to the Caller and the Callee throughout the Follow Me operations.

- **Call-From Prompt** - This prompt will play back to the Callee before announcing the Caller's name if they have recorded it.
- **No Recording Prompt** - This prompt will play back to the Callee with a generic message announcing an incoming call.
- **Please Hold Prompt** - This prompt will play back to the Caller, asking them to hold while the application tries to connect them with the Callee.
- **Status Prompt** - This prompt will play back to the Caller, informing them that the Callee is not within reach of their extension. So, the application will try to reach them through the Follow Me List.
- **Sorry Prompt** - This prompt will play back to the Caller if the Callee is not reachable on any number from the Follow Me List.

Finally, you can **Enable or Disable** Follow Me for this extension. This can also be done through Feature Codes, which we will see later.

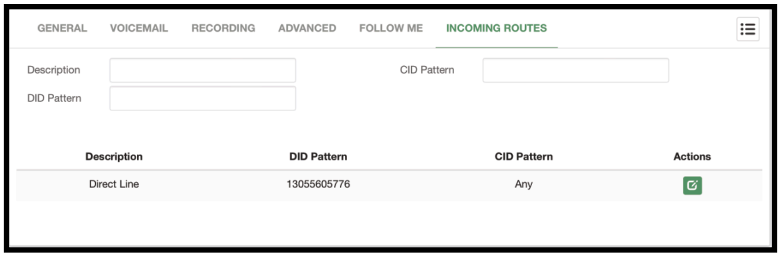
Now, underneath these configurations, we have more Follow Me options. Here, you can enable or disable the ability to **Record the Caller's Name**. As well as to enable or disable the ability to **prompt Internal or External Callees** to accept the incoming call. When you allow the Callees to be prompted, they can take or reject the incoming call.

With this, you will have a complete system to guide a call to multiple destinations in case where the user is unavailable to answer their extension. Remember to click on **Save** and then **apply changes**.

Section 3, Lesson 6 - Incoming Routes

It is pretty common for users to have a dedicated DID for incoming calls through your PSTN or VoIP Provider. We will see later how to route incoming calls with VitalPBX, but we can easily configure incoming routes directly from the Extensions module for extensions.

To do this, we will go to **PBX > Extensions > Extensions**, and then in the extension, go to the Incoming Routes tab.



Description

DID Pattern

CID Pattern

Description	DID Pattern	CID Pattern	Actions
Direct Line	13055605776	Any	✎

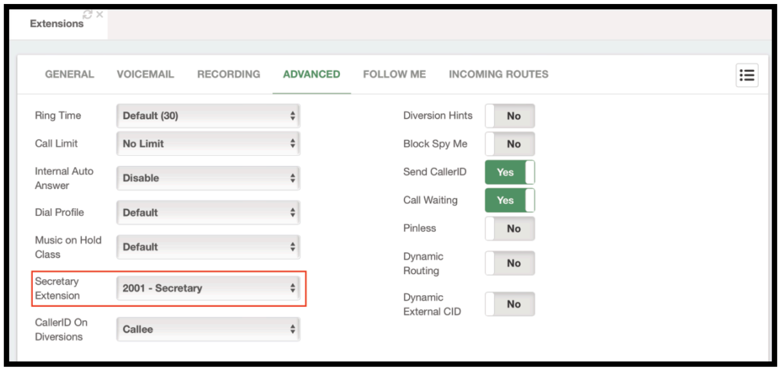
Image 3.6.1 - Incoming Routes tab in an extension.

The configuration of the incoming route is straightforward. You must add a **Description** to identify this route and enter the **DID and CID Pattern**. The DID pattern is the number assigned to this user, and the CID pattern is used so that only specific Caller IDs are routed through this incoming route. We will go deeper into incoming routes later. Once you **Save or Update**, you will see the assigned Incoming Routes in the table below. Here you can see the information and also have an **Action** button. This will take you to the Inbound Routes module so you can make any changes to the route.

Section 3, Lesson 7 - Boss/Secretary Feature

As mentioned before, you can assign a **Secretary Extension** to an Extension. This is a feature that we call Boss/Secretary. This comes from environments where the Boss has a personal assistant that aids in filtering calls for them. So by declaring a Secretary Extension, you can route incoming calls to the Boss extension to the Secretary Extension.

To use this feature, you must have at least two extensions created. Next, in the Boss Extension, go to the Advanced Tab. There, under **Secretary Extension** select an extension to fulfill this function.



Ring Time	Default (30)	Diversion Hints	No
Call Limit	No Limit	Block Spy Me	No
Internal Auto Answer	Disable	Send CallerID	Yes
Dial Profile	Default	Call Waiting	Yes
Music on Hold Class	Default	Pinless	No
Secretary Extension	2001 - Secretary	Dynamic Routing	No
CallerID On Diversions	Callee	Dynamic External CID	No

Image 3.7.1 - Secretary Extension setting in the Extension's Advanced tab.

Then, remember to **Update** and **Apply Changes**. Next, from the Boss Extension, use the feature code, ***36**. This toggle code allows you to turn the feature on and off.

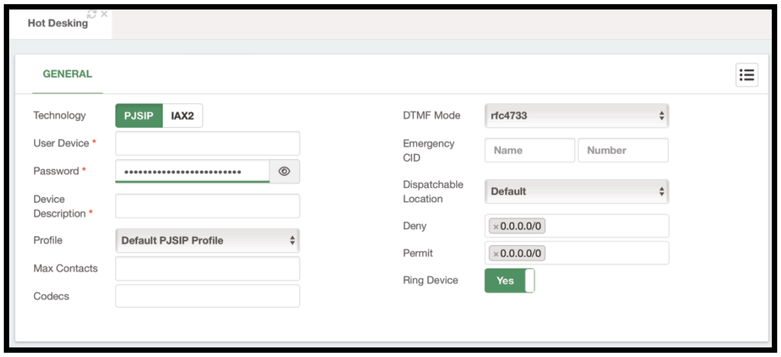
When the feature is on, you will notice that the Boss Extension cannot receive calls directly from any extension, but the Secretary Extension. Instead, all calls will go to the Secretary Extension. The Secretary Extension can then filter the calls and transfer them to the Boss Extension.

The Boss user can also create a **Whitelist** of numbers that can surpass the Secretary Extension and reach them directly. We will see more of this when we reach the **User Portal** lesson.

Section 3, Lesson 8 - Configuring a Hot Desking Device

Sometimes you may find yourself having more staff than devices. This is common in businesses like Call Centers that manage multiple shifts using the same offices or cubicles for employees. Here, Hot Desking devices are the perfect solution for this situation.

To create a Hot Desking device you must first go to **PBX > Extensions > Hot Desking**.



Technology	PJSIP IAX2
User Device *	
Password *	••••••••••••••••••••••••
Device Description *	
Profile	Default PJSIP Profile
Max Contacts	
Codecs	
DTMF Mode	rfc4733
Emergency CID	Name Number
Dispatchable Location	Default
Deny	×0.0.0.0/0
Permit	×0.0.0.0/0
Ring Device	Yes

Image 3.8.1 - Hot Desking module.

You will see that this is very similar to the **Devices** section for **Extensions**. Similarly to how you can create an extension without a device, a Hot Desking device is a device without a defined extension number. So by mixing these two features, you can dynamically exchange the extension associated with this device.

Creating a Hot Desking device is simple. You can just add a **User Device**, and a **Description**, and use the auto-generated **Password**. This is what you need to register a device. Remember to **Save** and **Apply Changes**. We have more settings you can configure, but you can refer to the Extension's General section for more information on the rest of the settings. For now, we will leave the rest blank and use the defaults.

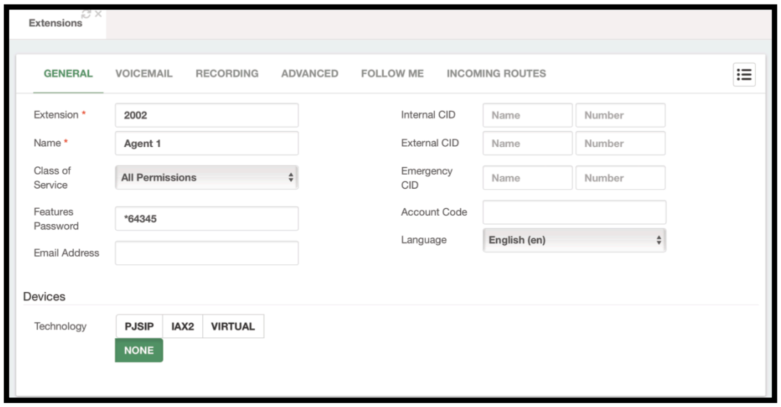
Remember that the User Device cannot have spaces or special characters.

Section 3, Lesson 9 - Using Hot Desking

To use your new Hot Desking device, you must first register the device. This can be any hardware that uses SIP, a softphone, or a mobile application. Once the device has registered using the User Device and Password, you will notice that you can't place any calls. This is due to no extension being associated with this device.

There is only one exception when it comes to Emergency calls, but we will see more about this later.

To be able to call other extensions, we must associate an extension number with it. For this, create an extension where the device is set to **NONE**.



Extension * 2002
Name * Agent 1
Class of Service All Permissions
Features Password *64345
Email Address
Internal CID Name Number
External CID Name Number
Emergency CID Name Number
Account Code
Language English (en)

Devices

Technology PJSIP IAX2 VIRTUAL NONE

Image 3.9.1 - Extension with device technology set to NONE.

Once you have the extension with the device set to NONE, use the feature code *80 in your Hot Desking device. You can then follow the prompts to enter the extension you wish to use, which is the extension with the device set to NONE. Afterward, it will ask you for a password. This password is the **Features Password** you set up in your extension. In the case of the example above, this is **64345*, but this can be anything of your liking or another generated Features Password.

After the prompts, you will have an extension associated with the device and you can now place calls.

To remove the extension, you can use the code ***80** again and follow the prompts to remove the extension.

You also have the option to use the feature code, ***90**. This will allow you Hot Desk an extension and log them into the queues they belong to. As mentioned before, this is a popular feature for Call Centers, so this code streamlines the login process.

Section 3, Lesson 10 - Import Extensions

Now that you can create extensions, you might find creating multiple extensions a bit tedious if you have tens or hundreds. Or, you might want to move multiple extensions from a VitalPBX server to another.

For this, we have our Import and Export extensions modules. These modules allow you to create and move multiple extensions at once.

To import multiple extensions at once is simple. For this, you must head over to **PBX > Extensions > Import Extensions**.

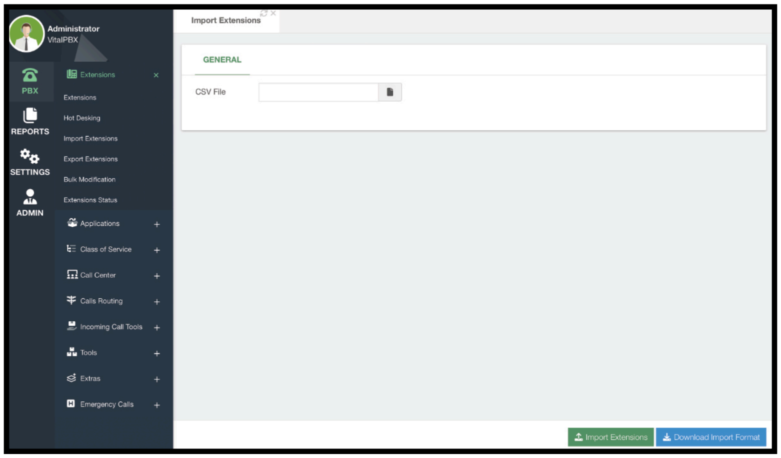
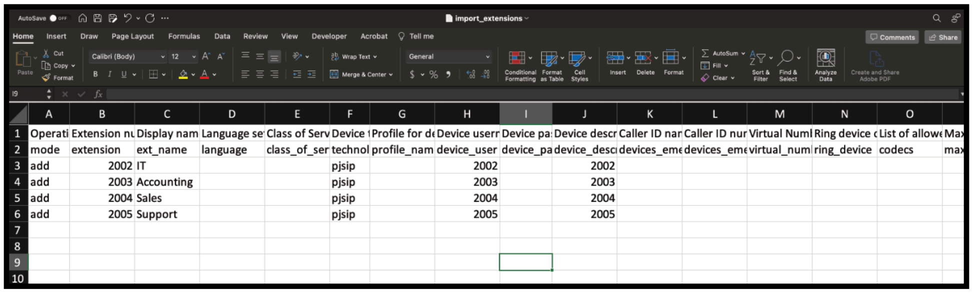


Image 3.10.1 - Import Extensions module.

This module allows you to upload a **CSV file** that contains the configurations for multiple extensions. For this, you can **Download the Import Format** by clicking on the button in the lower right-hand corner. You can open this in any number processor like Excel, Open Office, or Numbers. The file itself has all the columns for the same configurations we saw when creating an extension. On the first row, you will see text explaining each field as well.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	Operati	Extension nu	Display nam	Language se	Class of Serv	Device t	Profile for de	Device userr	Device pa	Device descr	Caller ID nar	Caller ID nur	Virtual Numl	Ring device c	List of allowe	Max
2	mode	extension	ext_name	language	class_of_ser	technol	profile_nam	device_user	device_pa	device_desci	devices_eme	devices_eme	virtual_numl	ring_device	codecs	max
3	add	2002	IT			pjsip		2002		2002						
4	add	2003	Accounting			pjsip		2003		2003						
5	add	2004	Sales			pjsip		2004		2004						
6	add	2005	Support			pjsip		2005		2005						
7																
8																
9																
10																

Image 3.10.2 - Extension's Import Format template.

It is important to note that you can add or remove extensions in mass with the import extensions feature. So the first column you will need to specify if you want to **add** or **delete** the extension number or device.

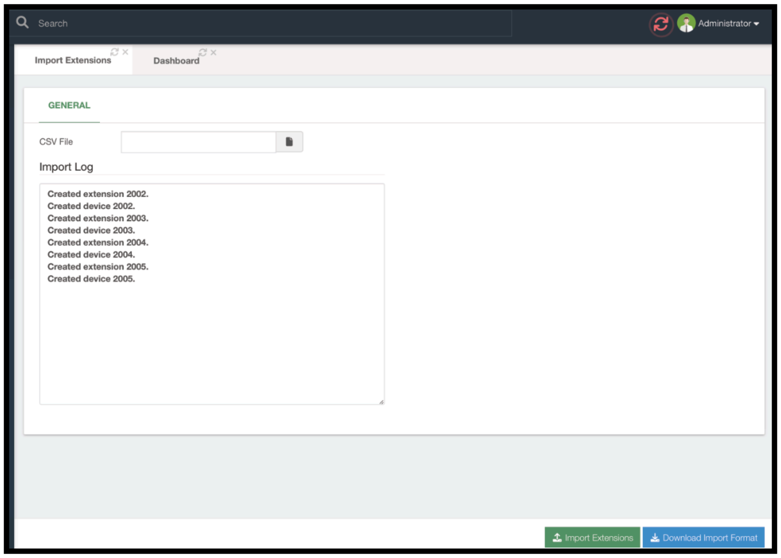
You can also add multiple extensions without a device if you leave the **technology** and **device_user** fields blank. Likewise, you can create Hot Desking devices by not specifying the **extension number** and **ext_name**.

It is possible to add multiple devices to a single extension number using the same extension number in the **extension** field, and using the **add_device** operation instead of *add*. There is no need to specify the **ext_name** field in this case.

You can leave most of the fields you don't want to specify blank, as they will use default values when nothing is implied. In the case of the **device_password** and **feature_password**, they are randomly generated.

Save the CSV file in **UTF-8 format** and be **separated by commas**.

Once you have your filled CSV file, you can go back to the Import Extensions module, and open the file you just created by clicking on the blank field next to **CSV File**. Then, click on the green **Import Extensions** button in the lower right-hand corner.



```
Created extension 2002.
Created device 2002.
Created extension 2003.
Created device 2003.
Created extension 2004.
Created device 2004.
Created extension 2005.
Created device 2005.
```

Image 3.10.3 - Extensions imported from the CSV file.

You will then be presented with an **Import Log** with the extensions that were created. If any errors occur, you will see them in this log. You can then correct the CSV file and import it again.

Using the **add** or **add_device** operations for existing extensions or devices will modify the existing extensions and devices.

Remember to **Apply Changes** after you Import the extensions.

Section 3, Lesson 11 - Export Extensions

With many extensions now added to your system you may want to back them up or move them to another VitalPBX installation. For this, you can export the extensions using our Export Extensions module. Let’s go to **PBX > Extensions > Export Extensions**.

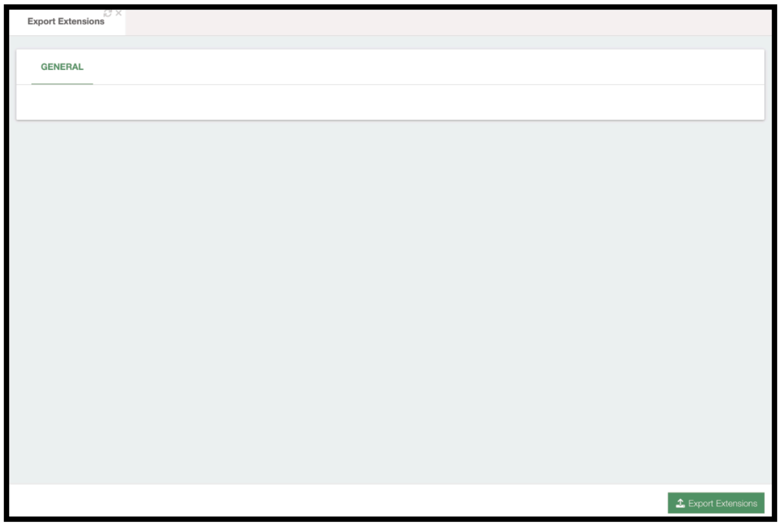
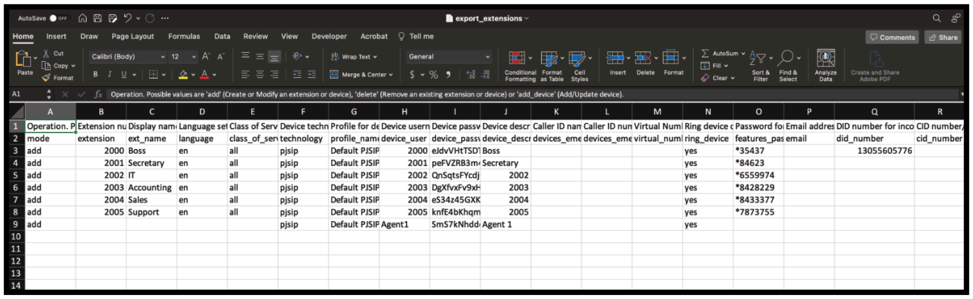


Image 3.11.1 - Export Extensions module.

This is one of our most simple modules. The only option here is the green **Export Extensions** button in the lower right-hand corner. When you click on this button a CSV file is generated with all the configurations for all the extensions in the system.



Operation. Possible values are 'add' (Create or Modify an extension or device), 'delete' (Remove an existing extension or device) or 'add_device' (Add/Update device).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Operation. P	Extension nu	Display nam	Language set	Class of Serv	Device techn	Profile for de	Device usern	Device passv	Device descr	Caller ID nam	Caller ID nun	Virtual Numl	Ring device c	Password fo	Email addres	DID number for inco	CID number/
2	mode	extension	ext_name	language	class_of_serv	technology	profile_name	device_user	device_passv	device_descr	devices_eme	devices_eme	virtual_numl	ring_device	features_pas	email	did_number	cid_number
3	add	2000	Boss	en	all	pjsip	Default PJSIP	2000	eJdvVHtTSDT	Boss				yes	*35437		13055605776	
4	add	2001	Secretary	en	all	pjsip	Default PJSIP	2001	peFVZRB3m	Secretary				yes	*84623			
5	add	2002	IT	en	all	pjsip	Default PJSIP	2002	QnSqtsFYcdj	2002				yes	*6559974			
6	add	2003	Accounting	en	all	pjsip	Default PJSIP	2003	DgXfvxFv9xH	2003				yes	*8428229			
7	add	2004	Sales	en	all	pjsip	Default PJSIP	2004	eS34z45GXK	2004				yes	*8433377			
8	add	2005	Support	en	all	pjsip	Default PJSIP	2005	knfE4bKhqm	2005				yes	*7873755			
9	add					pjsip	Default PJSIP	Agent1	SmS7kNhdd	Agent 1				yes				
10																		
11																		
12																		
13																		
14																		

Image 3.11.2 - Exported extensions CSV file.

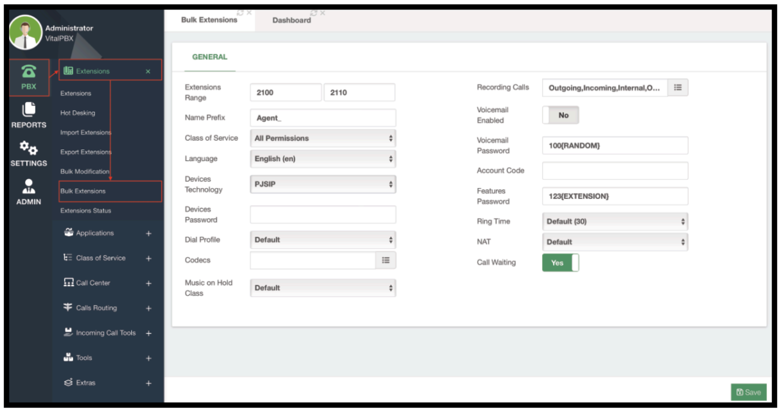
This CSV file has the same format as the one we saw for importing extensions. You can use this file with the **Import Extensions** module in another VitalPBX Installation.

Another popular use of this feature is for doing mass modifications for various extensions at once. You can export the extensions, then modify multiple configurations in multiple extensions, save the file, and then import it back to your VitalPBX installation. Remember, when using the **add** or **add_device** operations when importing extensions or devices, this modifies the existing extensions and devices in your VitalPBX installation.

Section 3, Lesson 12 - Bulk Extensions

Another way to add multiple extensions is using our **Bulk Extensions** add-on module. This is a free add-on module you can install that will allow you to generate multiple extensions directly from the VitalPBX Web UI.

To use the Bulk Extensions module, you need to install the add-on first from **Admin > Add-Ons > Add-Ons**. Once the add-on is installed, refresh the browser, and you will now see a new option under **PBX > Extensions > Bulk Extensions**.



Extensions Range: 2100 – 2110
Name Prefix: Agent_
Class of Service: All Permissions
Language: English (en)
Devices Technology: PJSIP
Devices Password:
Dial Profile: Default
Codecs:
Music on Hold Class: Default
Recording Calls: Outgoing,Incoming,Internal,O...
Voicemail Enabled: No
Voicemail Password: 100{RANDOM}
Account Code:
Features Password: 123{EXTENSION}
Ring Time: Default (30)
NAT: Default
Call Waiting: Yes

Image 3.12.1 - Bulk Extensions add-on module.

To use the module, you need to define an **Extension Range**, this is the extensions and the numbers you will use for these extensions. The rest of the fields are the same as you find when creating the device for an extension. Additionally, you can define the types of **Call Recording** you will use, and whether or not you are enabling **Voicemail** for the extensions.

For the different types of Passwords, you have variables you can use to standardize them with content based on the extension.

- **{EXTENSION} -** This will add the extension number to the field. You can combine it with plain text. E.g. *secure_password{EXTENSION}*.
- **{RANDOM} -** This will generate a random string of characters to make the password more secure. You can combine this with plain text as well. E.g. *password{RANDOM}*.

If you leave password fields blank, a random password will be generated for each extension created.

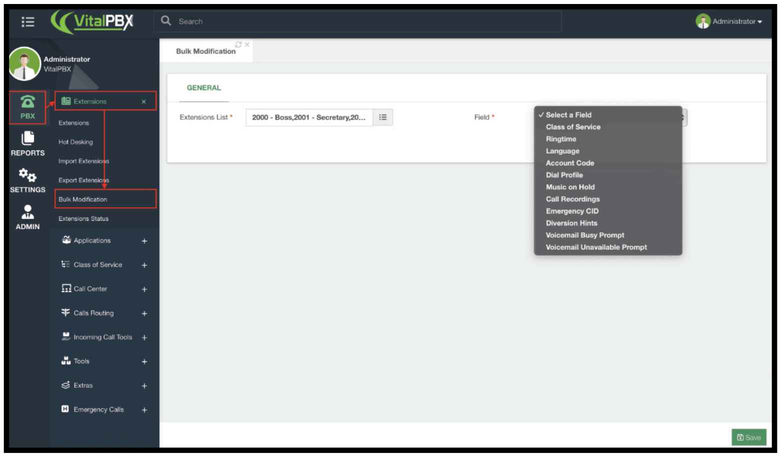
When you **Save** and **Apply Changes**, all of the extensions will be created.

Using the Bulk Extensions module can save you a lot of time by generating multiple extensions at once. You can even combine it with the Import and Export Extensions modules so you can generate the extensions with the Bulk Extensions module, export them, make any modifications to the CSV file, and import it again to modify the extensions with any changes you made.

Section 3, Lesson 13 - Bulk Modification

There are some cases where you need to make a quick modification to multiple extensions. For this, we have created the Bulk Modification module. Here, you can modify the same configuration to multiple extensions at once.

For us to start doing Bulk Modifications we must go to **PBX > Extensions > Bulk Modification.**

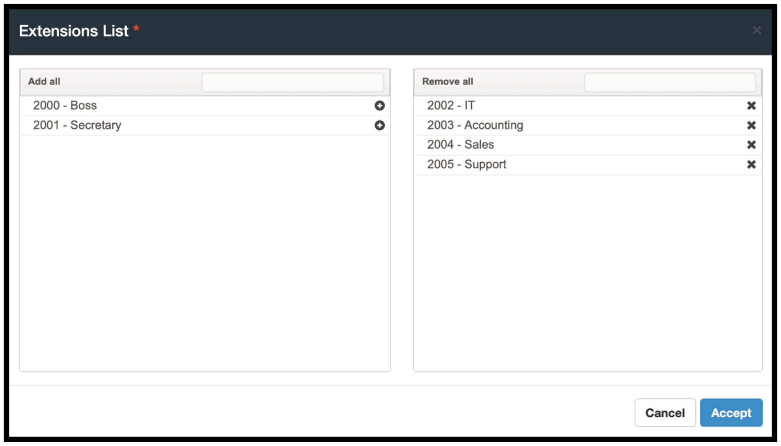


Extensions List * 2000 - Boss,2001 - Secretary,20...

Field *

Image 3.13.1 - Bulk Modification module.

In this module, you will first need to select the extensions you wish to modify. For this, click on the field next to **Extensions List**.



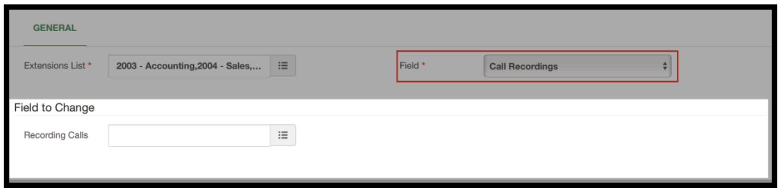
Add all	Remove all
2000 - Boss	2002 - IT
2001 - Secretary	2003 - Accounting
	2004 - Sales
	2005 - Support

Image 3.13.2 - Extensions List in the Bulk Modification module.

This is a very common modal in VitalPBX, where you have two list columns. Everything on the left is what can be selected, and everything on the right is what is selected. You can use the text box above the column to filter for specific terms, and use the **add all** or **remove all** options to move all the items in the column. In this example, we have selected the extensions we imported in the previous lesson. Click on **Accept**.

Next, we have the **Field** option. Here we choose the field we wish to modify for the selected extensions. These are various options we have already seen when creating an extension. You can go back to the initial lessons in this section to review each option.

For now, we will select the **Call Recordings** option. This will display a new section at the bottom. Each field we select will show you different options for that field. In the case of Call Recordings, you will get the call recording options.



Extensions List: 2003 - Accounting,2004 - Sales,...
Field: Call Recordings

Image 3.13.3 - Field to change section.

When you click on the field next to **Recording Calls** you will be shown a modal to choose the call recording options.

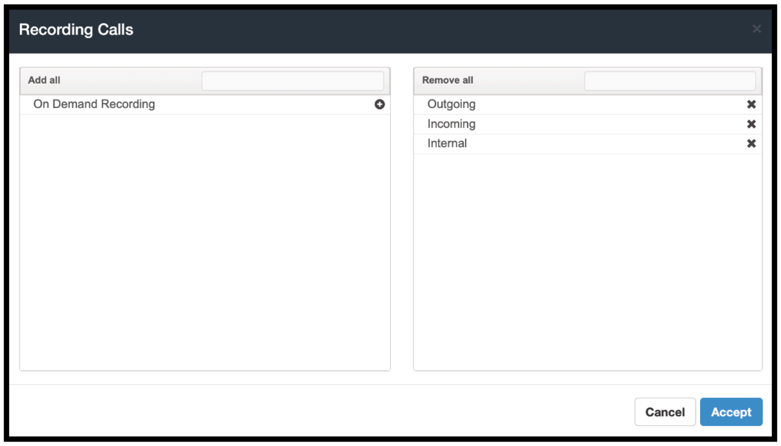
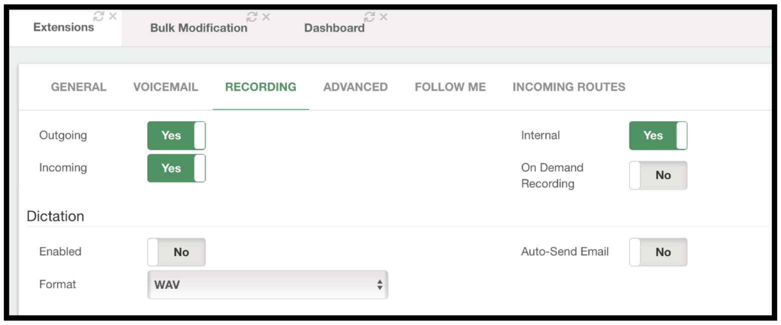


Image 3.13.4 - Call Recording options.

In this example, we will choose Outgoing, Incoming, and Internal. Then click **Accept**, and click on the green **Save** button in the bottom right-hand corner. Remember to **Apply Changes**. Go back to any of the extensions you modified, and you will see that the modifications have been applied.



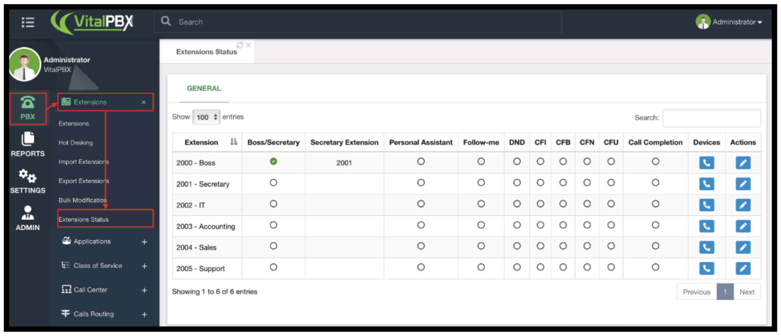
Outgoing: Yes
Incoming: Yes
Internal: Yes
On Demand Recording: No
Enabled: No
Auto-Send Email: No
Format: WAV

3.13.5 - Extension with the modifications applied.

Section 3, Lesson 14 - Extension Status

With various extensions now deployed, you need to easily monitor them. You can monitor their registration status and active diversions with the Extension Status module. The module also allows you to modify the status and configure the diversions.

To access the Extension Status Module, you must go to **PBX > Extensions > Extension Status**. The module will show you a list of all the extensions in the system with their **Call Diversions status** and **actions** you can do to these extensions.



Extension	Boss/Secretary	Secretary Extension	Personal Assistant	Follow-me	DND	CFI	CFB	CFN	CFU	Call Completion	Devices	Actions
2000 - Boss	✓	2001	○	○	○	○	○	○	○	○	☎	✎
2001 - Secretary	○		○	○	○	○	○	○	○	○	☎	✎
2002 - IT	○		○	○	○	○	○	○	○	○	☎	✎
2003 - Accounting	○		○	○	○	○	○	○	○	○	☎	✎
2004 - Sales	○		○	○	○	○	○	○	○	○	☎	✎
2005 - Support	○		○	○	○	○	○	○	○	○	☎	✎

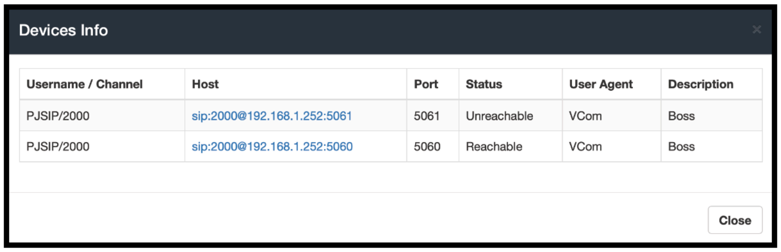
Showing 1 to 6 of 6 entries

Image 3.14.1 - Extensions Status module.

With this module, you cannot only monitor the status of diversions, but you can manage them as needed. By clicking the **radial button** under any **Diversion** next to an **Extension**, you can enable and disable them immediately. We can Enable and Disable the **Boss/Secretary**, **Personal Assistant**, **Follow-Me**, and **Do Not Disturb (DND)**—as well as various **call-forward diversions and Call Completion**.

These features can be activated by the end-users using feature codes, which we will see more about later, or BLF keys that have these codes configured. With this module, you can monitor if they have been activated by mistake and disable them.

Additionally, we have the **Devices** column. This option will show you the registration status of an extension for all their devices. This includes useful information such as the registration IP Address and the User Agent.

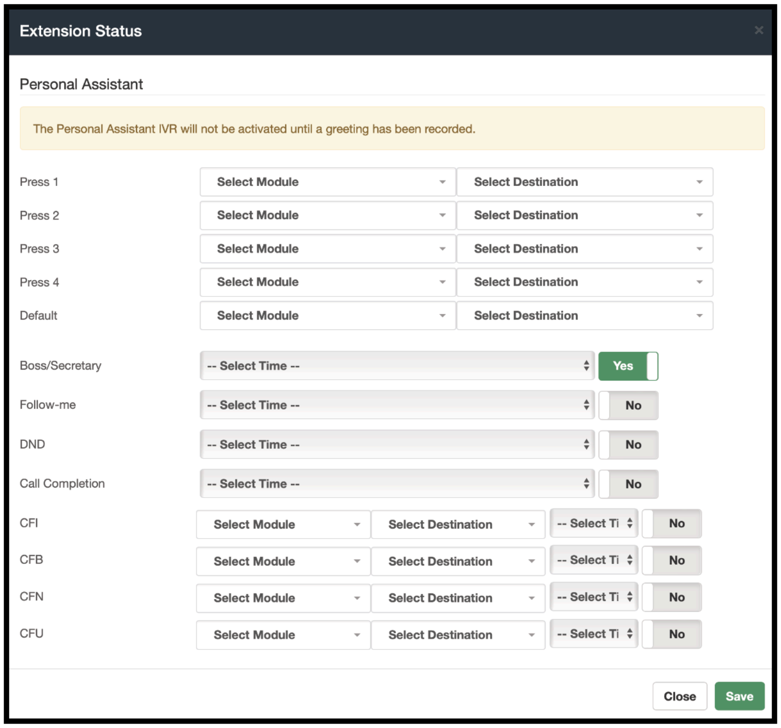


Username / Channel	Host	Port	Status	User Agent	Description
PJSIP/2000	sip:2000@192.168.1.252:5061	5061	Unreachable	VCom	Boss
PJSIP/2000	sip:2000@192.168.1.252:5060	5060	Reachable	VCom	Boss

Image 3.14.2 - Device Info for the Boss extension.

You can click on the **Host** link and this will take you to the device's web portal if available. Keep in mind this won't work if your server and devices are not on a LAN. To close the modal click on **Close**.

Lastly, we have the **Actions** Column. By clicking on the edit button next to an extension, you can configure the different diversions for this extension.



Extension Status

Personal Assistant

The Personal Assistant IVR will not be activated until a greeting has been recorded.

Press 1	Select Module	Select Destination
Press 2	Select Module	Select Destination
Press 3	Select Module	Select Destination
Press 4	Select Module	Select Destination
Default	Select Module	Select Destination

Boss/Secretary	-- Select Time --	Yes
Follow-me	-- Select Time --	No
DND	-- Select Time --	No
Call Completion	-- Select Time --	No

CFI	Select Module	Select Destination	-- Select Ti	No
CFB	Select Module	Select Destination	-- Select Ti	No
CFN	Select Module	Select Destination	-- Select Ti	No
CFU	Select Module	Select Destination	-- Select Ti	No

Image 3.14.3 - Extension diversions in the Diversion Status module.

Here you can configure the **Personal Assistant** for this extension. The Personal Assistant is a dedicated IVR for the extension which, when activated, will playback and allow calling parties to select up to any of 4 options to go to a defined destination.

You can also define the status of other diversions like **Boss/Secretary**, **Follow-Me**, **Do Not Disturb**, and **Call Completion**, instead of using their feature codes from the extension. You will notice that you can also assign a **Time Group** to automate when they are enabled or disabled. We will see more about *Time Groups* later.

> ***Call Completion*** *is a type of diversion we have not seen yet. This feature allows you to automatically dial someone back as soon as you are available. When this feature is enabled on your extension, and you are busy or unavailable, callers will be able to request a Call Completion. Therefore, when you become available again, VitalPBX will generate the call automatically, between the Caller and your Extension.*

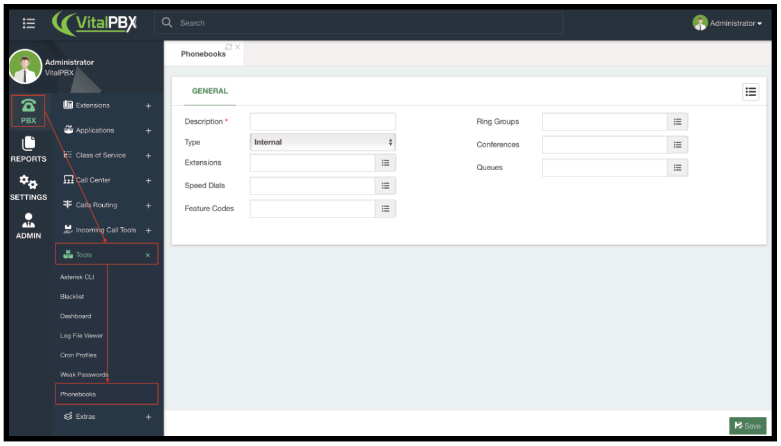
Lastly, you have four **Call Forward** options: **Immediate**, **on Busy**, **on No Answer**, and **Unavailable**. You can choose the destination for all of them, and establish a *Time Group* to automate when to enable or disable the Call Forwards.

Section 3, Lesson 15 - Phonebooks

With VitalPBX, you can install our free **Phonebooks** add-on module. This allows you to centralize your contacts in a single location. Once you have installed the Phonebooks add-on, you can find it under **PBX > Tools > Phonebooks**.

There are two types of phonebooks, **Internal** and **External**. **Internal Phonebooks** can have a list of **Extensions**, **Speed Dials**, **Feature Codes**, **Ring Groups**, **Conferences**, and **Queues**. Meanwhile, **External Phonebooks** can have a list of contacts with numbers external to the VitalPBX.

Let's make an internal phonebook first.

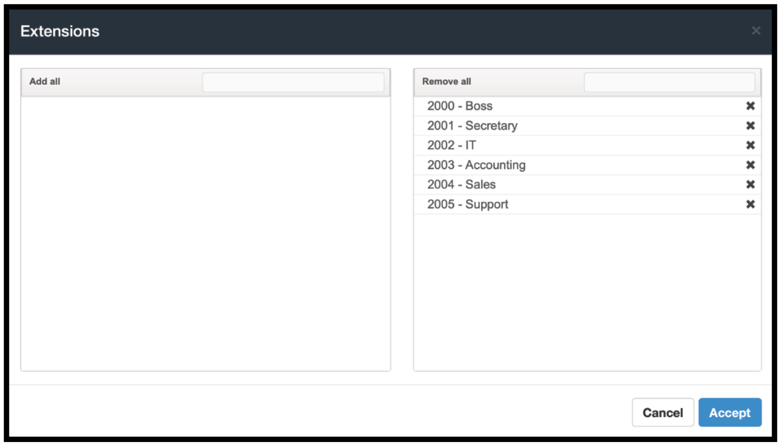


Type: Internal

Image 3.15.1 - Phonebooks module with an Internal Phonebook selected.

To create the phonebook, you will first need to enter a **description**. Next, you will need to select the type. By default, an internal phonebook is created.

Afterward, you can add any of the different options available. In this case, we will include the extensions we have created so far. For this, I click on the list icon on the **Extensions** field.

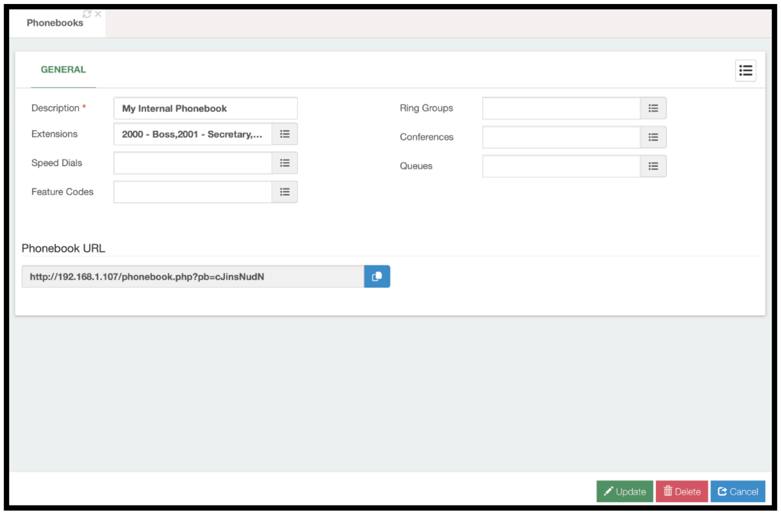


2000 - Boss ✖
2001 - Secretary ✖
2002 - IT ✖
2003 - Accounting ✖
2004 - Sales ✖
2005 - Support ✖

Image 3.15.2 - Extensions Selection in the Phonebooks module.

You are presented with the list of extensions. Remember, in this type of selection, everything you can select will appear in the left column, and everything that has been selected will appear in the right column. We press the **Add All** option to include all the extensions and then click on **Accept**.

Once you have selected everything you wish to add to this phonebook, click on the **Save** button in the bottom right-hand corner. The phonebook is created, and you can find it in the list in the upper right-hand corner of the module.



Description * My Internal Phonebook

Extensions 2000 - Boss,2001 - Secretary,...

Speed Dials

Feature Codes

Ring Groups

Conferences

Queues

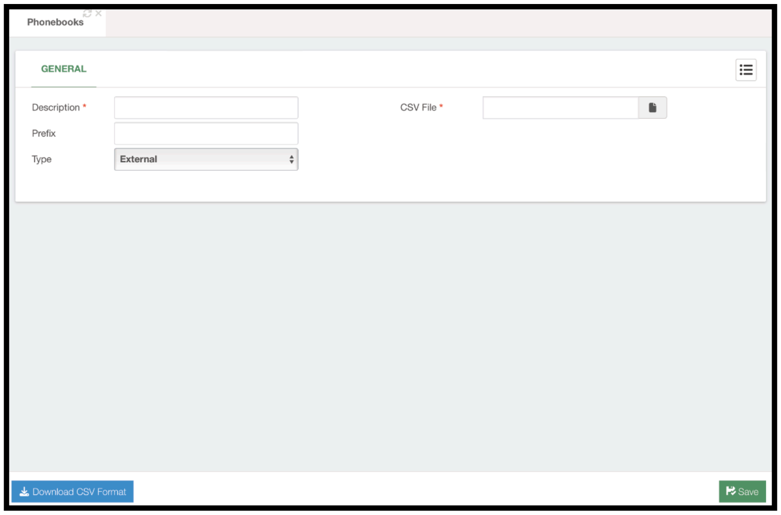
Phonebook URL

http://192.168.1.107/phonebook.php?pb=cJinsNudN

Image 3.15.3 - An Internal Phonebook created.

With the phonebook created you will now have a **Phonebook URL**. You can add this URL to any device that supports centralized phonebook URLs for *remote directories*. Additionally, you will now find a **Contact Info** tab in your extensions. This is to add additional information to be displayed with centralized phonebooks.

Now, let's create an external phonebook.



Description *

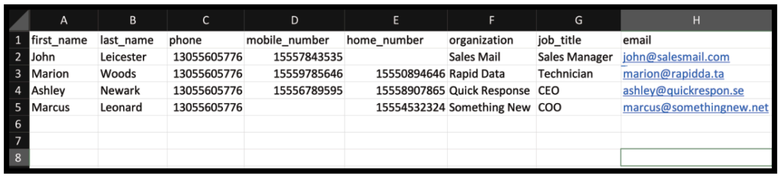
Prefix

Type External

CSV File *

Image 3.15.4 - External phonebook selected in the Phonebooks module.

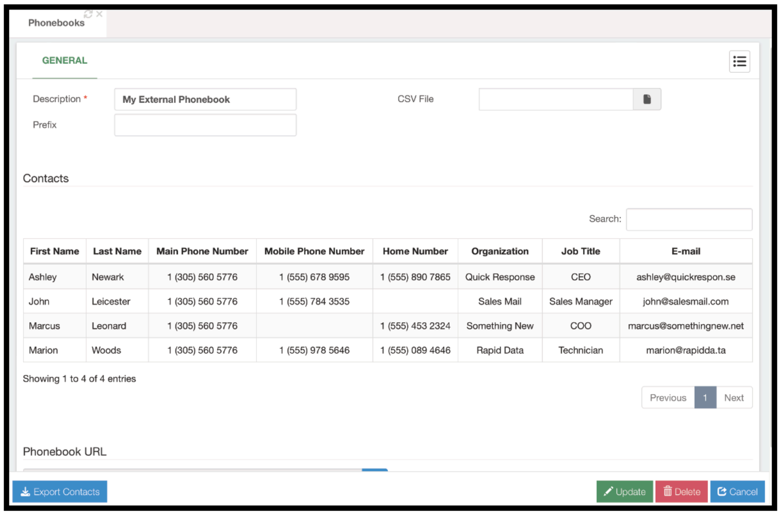
External phonebooks work a little differently, as you need a **CSV file** to import the contacts. A **CSV format template** is available with a button in the bottom left-hand corner of the module.



	A	B	C	D	E	F	G	H
1	first_name	last_name	phone	mobile_number	home_number	organization	job_title	email
2	John	Leicester	13055605776	15557843535		Sales Mail	Sales Manager	john@salesmail.com
3	Marion	Woods	13055605776	15559785646	15550894646	Rapid Data	Technician	marion@rapidda.ta
4	Ashley	Newark	13055605776	15556789595	15558907865	Quick Response	CEO	ashley@quickrespon.se
5	Marcus	Leonard	13055605776		15554532324	Something New	COO	marcus@somethingnew.net
6								
7								
8								

Image 3.15.5 - External Phonebooks CSV import file.

The CSV file is quite simple, storing the contact's **First and Last Name**, T**hree Phone Numbers**, **Organization**, **Job Title**, and **Email**. Once you have entered the contacts' information, save the CSV file, and in the Phonebooks module, upload with the CSV File field. Then, click the **Save** button in the lower right-hand corner.



Description * My External Phonebook

CSV File

Prefix

Contacts

Search:

First Name	Last Name	Main Phone Number	Mobile Phone Number	Home Number	Organization	Job Title	E-mail
Ashley	Newark	1 (305) 560 5776	1 (555) 678 9595	1 (555) 890 7865	Quick Response	CEO	ashley@quickrespon.se
John	Leicester	1 (305) 560 5776	1 (555) 784 3535		Sales Mail	Sales Manager	john@salesmail.com
Marcus	Leonard	1 (305) 560 5776		1 (555) 453 2324	Something New	COO	marcus@somethingnew.net
Marion	Woods	1 (305) 560 5776	1 (555) 978 5646	1 (555) 089 4646	Rapid Data	Technician	marion@rapidda.ta

Showing 1 to 4 of 4 entries

Phonebook URL

Image 3.15.6 - An External Phonebook created.

Now, you will see the external phonebook contacts listed in a table, similar to how they were listed in the CSV file.

You will also notice that external phonebooks have a **Prefix** field. The Prefix field in an external phonebook, allows you to add any prefix number needed to route the calls correctly. We will see more about outbound routes later, but sometimes you need additional numbers before a phone number so the call goes through the correct route. The prefix field allows you to add numbers that will be added automatically to the beginning of the contacts' phone numbers. This way when selecting a contact from a device using an external directory with an external phonebook, the call will be placed correctly.

You will now find the ability to **Export Contacts** as well, by clicking on the button in the lower left-hand corner. So if the contacts are modified, you can download the current listing of contacts. You can modify contacts by uploading a new CSV file.

Additionally, if you have a Starter License or any of our Licensing Plans, you get access to the **Add Contact** extended feature. This extended feature allows you to create a new contact directly from the Phonebooks module.

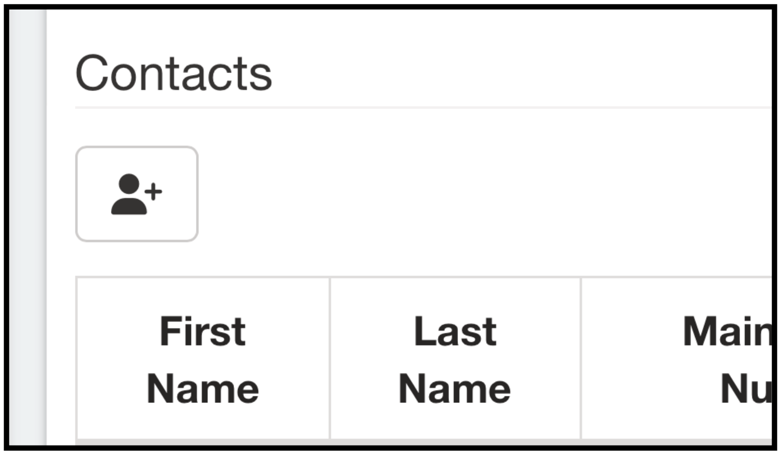
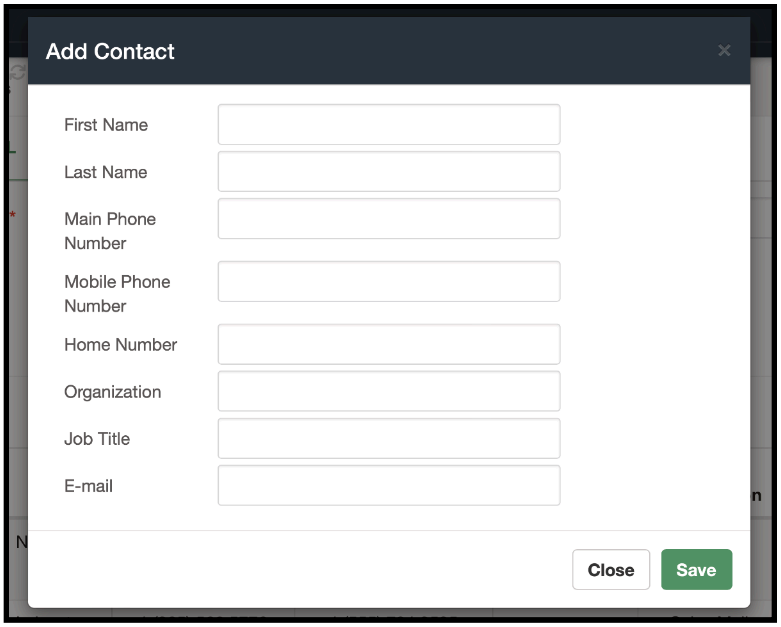


Image 3.15.7 - Add Contacts button in the Phonebooks Module.



Add Contact

First Name	
Last Name	
Main Phone Number	
Mobile Phone Number	
Home Number	
Organization	
Job Title	
E-mail	

Image 3.15.8 - External Phonebooks Add Contact form.

> ***Note:*** *Extended Features are additional configurations you can make when you have a Starter License or a Licensing Plan registered with your VitalPBX installation. We will go through these extended features as we progress with this guide.*

Now you have centralized phonebooks you can use with multiple devices. The devices will have the contacts updated automatically as you update the phonebook contacts.

Section 3, Lesson 16 - VitalPBX Connect

With VitalPBX we offer an application for iOS and Android called VitalPBX Connect. This is an application that will allow you to have your extension on your mobile device. This way, you will never miss a call again.

First here are some requirements for this application.

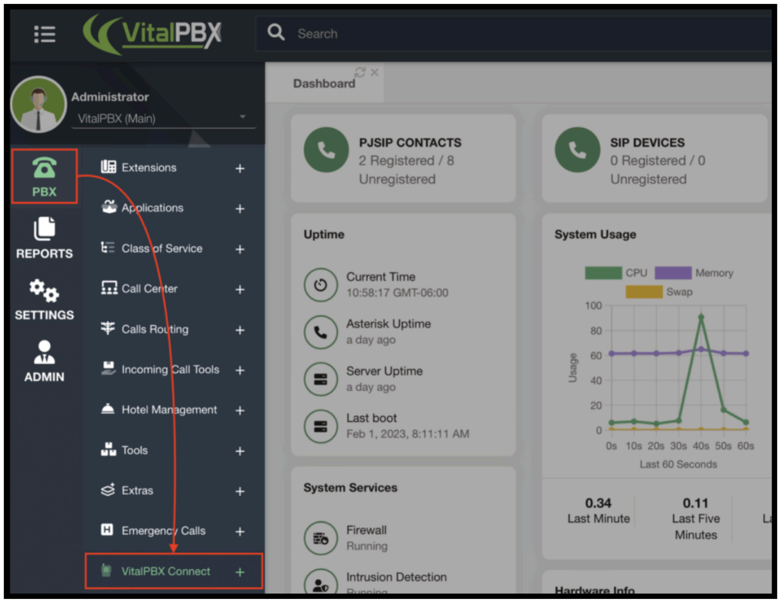
- Your VitalPBX server must have an FQDN (Fully Qualified Domain Name) pointing to it. Or have a Public IP Address so the application can register even when you are outside your network.
- The VitalPBX server must have access to the internet so the devices register using the registration server, and push notifications work properly.
- To use VitalPBX Connect, you will need to have the appropriate licensing. With our Licensing Plans, you get 2 included VitalPBX Connect devices. You will need to purchase additional licensing for more devices.
    - Licensing for VitalPBX Connect is offered in bundles of 10 devices for USD 12.00 MSRP. You can increment in intervals of 10 devices and the pricing adjusts accordingly.

With these requirements met, we can start by downloading the application. As mentioned earlier, VitalPBX Connect can be downloaded directly through the App Store for iOS or Google Play for Android. You can search for VitalPBX Connect by VitalPBX LLC.

You will notice that you need to scan a QR code to register the device. You will need to make some configurations in your VitalPBX installation to register your VitalPBX Connect device.

First, go to **Admin > Add-Ons > Add-Ons** and install the **Connect add-on**. If you don't see the add-on listed, click on the green **Check Online** button in the lower right-hand corner.

Once you have installed the add-on, refresh your browser. A new option will appear under the PBX menu.



PJSIP CONTACTS
2 Registered / 8 Unregistered

SIP DEVICES
0 Registered / 0 Unregistered

Current Time
10:58:17 GMT-06:00

Asterisk Uptime
a day ago

Server Uptime
a day ago

Last boot
Feb 1, 2023, 8:11:11 AM

0.34 Last Minute

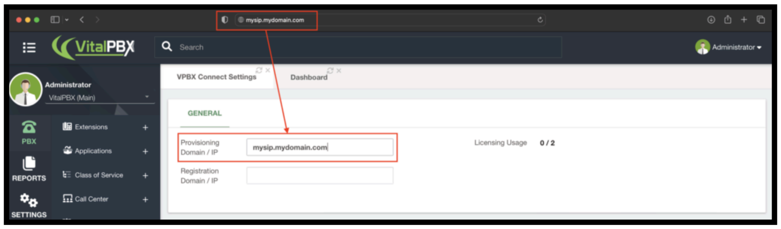
0.11 Last Five Minutes

Firewall
Running

Intrusion Detection
Running

Image 3.16.1 - VitalPBX Connect Menu Section.

Now, let's go to **PBX > VitalPBX Connect > Settings**. Here, we will configure our Provisioning Domain. This is the FQDN (Fully Qualified Domain Name) or Public IP Address of your VitalPBX installation. Click on **Save** and **Apply Changes**.



mysip.mydomain.com

VPBX Connect Settings | Dashboard

GENERAL

Provisioning Domain / IP: mysip.mydomain.com

Registration Domain / IP:

Licensing Usage 0 / 2

Image 3.16.2 - VitalPBX Connect Settings module.

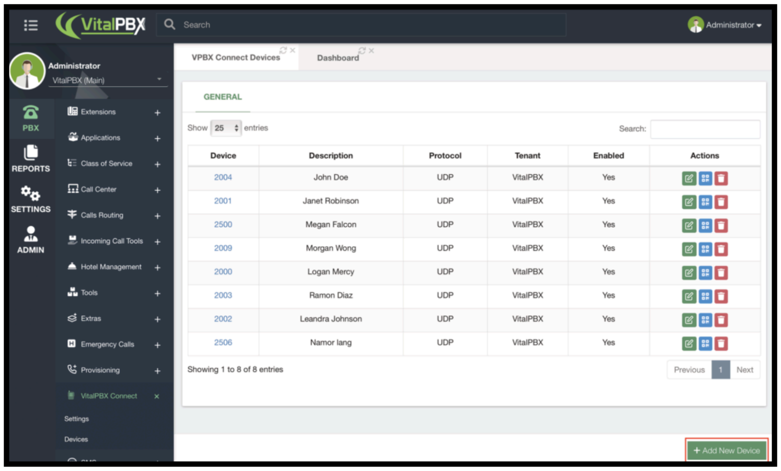
This will tell the VitalPBX Connect application where to get the provisioning information for the application. The data is provisioned directly by your VitalPBX installation to register your device.

You can use a different Registration Domain if you use a separate server for your registration. Usually, this server is the same one used for provisioning. So, you can leave it blank to use the same domain used before.

In this module, you can also see how many licenses you use for the VitalPBX Connect application and how many you have left.

Next, we can create the provisioning information for the devices in several ways. The first one is used mainly if you already have a device made for an extension, and you can easily create a new VitalPBX QR code.

For this option, go to **PBX > VitalPBX Connect > Devices.** Here you will see the list of devices tagged as Mobile Clients, and a provisioning QR code has been created for them.



VPBX Connect Devices | Dashboard

GENERAL

Show 25 entries | Search:

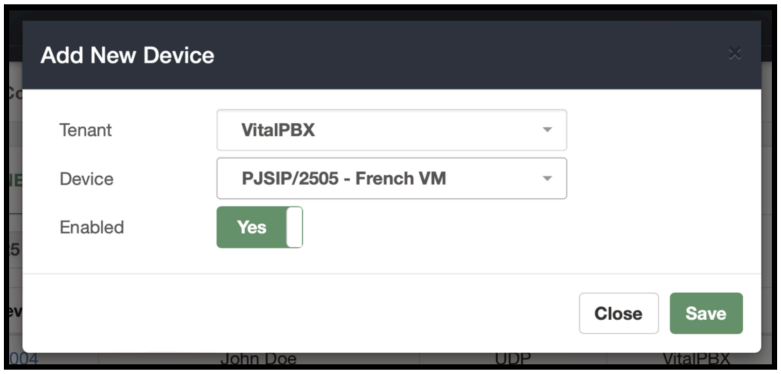
Device	Description	Protocol	Tenant	Enabled	Actions
2004	John Doe	UDP	VitalPBX	Yes	
2001	Janet Robinson	UDP	VitalPBX	Yes	
2500	Megan Falcon	UDP	VitalPBX	Yes	
2009	Morgan Wong	UDP	VitalPBX	Yes	
2000	Logan Mercy	UDP	VitalPBX	Yes	
2003	Ramon Diaz	UDP	VitalPBX	Yes	
2002	Leandra Johnson	UDP	VitalPBX	Yes	
2506	Namor lang	UDP	VitalPBX	Yes	

Showing 1 to 8 of 8 entries

Image 3.16.3 - VitalPBX Connect Devices Module.

You can click on **+ Add New Device** to add a new device. When adding a new device, you can choose a specific **tenant**. This will show you the list of available devices.

Keep in mind that the devices listed are those in that tenant. So if an extension has multiple devices, all will be listed. You can also **Enable/Disable** the device. Then click on **Save** and **Apply Changes**.



Add New Device

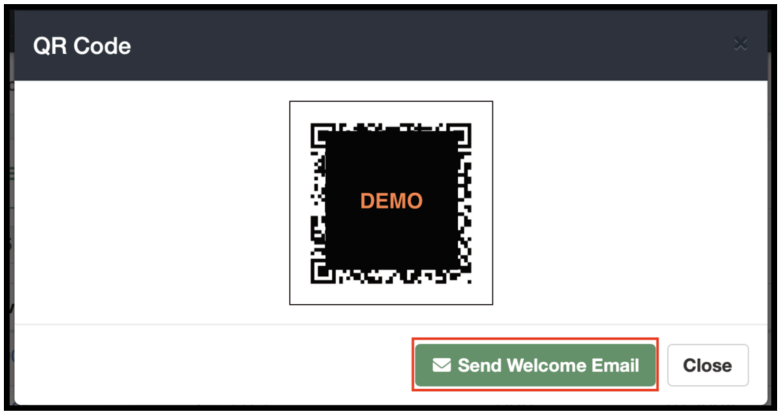
Tenant: VitalPBX

Device: PJSIP/2505 - French VM

Enabled: Yes

Image 3.16.4 - Adding a new VitalPBX Connect Device.

When you have multiple devices created, you will see various options in the **Actions** column. These allow you to **Edit** a device, **View the QR code**, and **Delete** a device. When you click on the blue **QR code button**, the QR code is displayed, and you will have the option to send it via email for the user to scan using the *Welcome Email template*.



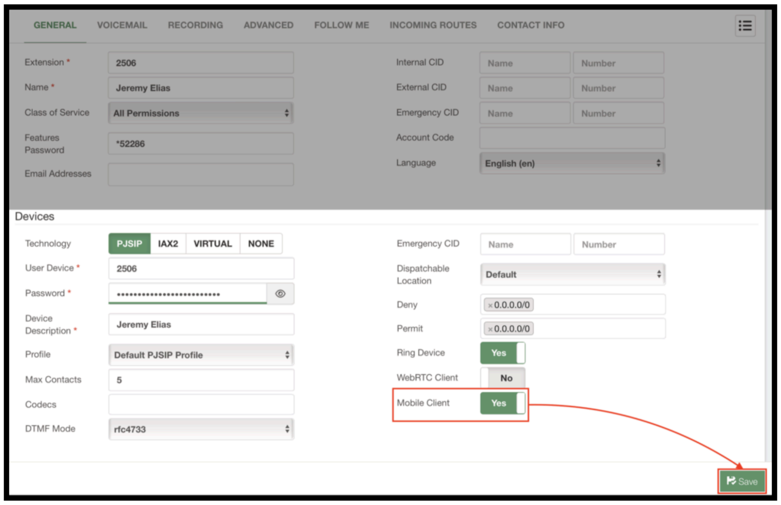
QR Code

Image 3.16.5 - VitalPBX Connect Device QR code.

The other way you can create a VitalPBX Connect device is when making a new extension. This will be the most common way you will create the VitalPBX Connect provisioning devices.

When creating a new extension, enable the **Mobile Client** option under the device section. When you **save** and **apply changes**, this will automatically generate the VitalPBX Connect device.

For new extensions, if you have configured the email client, and entered an email address on the email field for the extension, the extension welcome email can have the QR code so the end users can easily scan the code.



Extension * 2506
Name * Jeremy Elias
Class of Service: All Permissions
Features Password: *52286
Email Addresses:
Internal CID: Name / Number
External CID: Name / Number
Emergency CID: Name / Number
Account Code:
Language: English (en)

Devices

Technology: PJSIP | IAX2 | VIRTUAL | NONE
User Device * 2506
Password * ••••••••••••••••••••••••
Device Description * Jeremy Elias
Profile: Default PJSIP Profile
Max Contacts: 5
Codecs:
DTMF Mode: rfc4733
Emergency CID: Name / Number
Dispatchable Location: Default
Deny: 0.0.0.0/0
Permit: 0.0.0.0/0
Ring Device: Yes
WebRTC Client: No
Mobile Client: Yes

Image 3.16.6 - Mobile Client option enabled on an extension.

You can review the QR code by going back to the extension and clicking on the mobile phone icon next to the **Device**. This will show you the QR code as we saw previously and you will have the option of sending the extension welcome email again.

With the add-on module configured and the QR code generated, we can now go back to the application on our mobile devices and register our extensions.

The process is the same between iOS and Android. This article will use an iPhone as an example, but you can use an Android device.

First, open the VitalPBX Connect application on your device.

The first time you open the application, you will be greeted by the login screen. Here you will sign in with your provisioning information. For security purposes, VitalPBX generated a long random username and password. This is why using the QR code is much easier to provision your device.

Click on the **Scan QR button**. This will ask for camera permissions. Once you scan the QR code, the application will be provisioned, and your extension will register.

Once the device is provisioned, you will see the following screen.

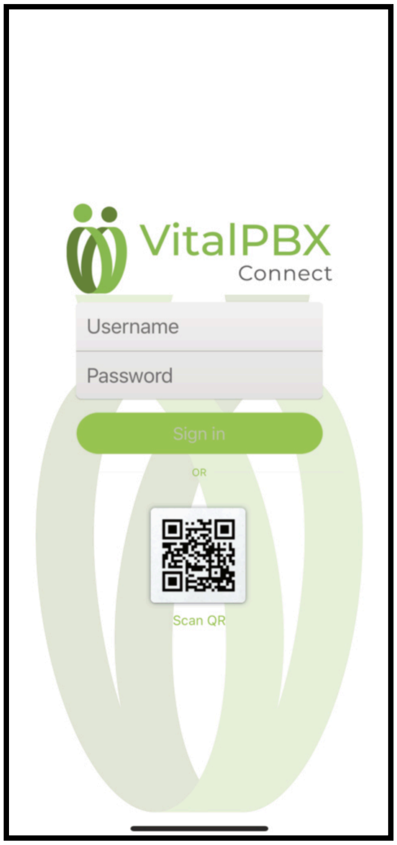
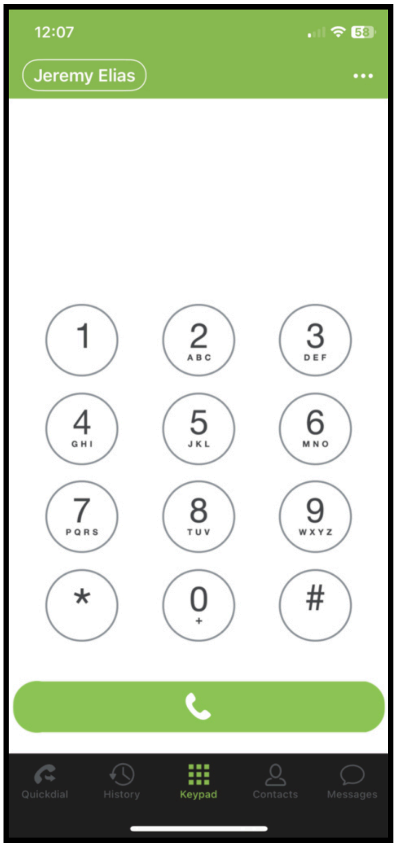


Image 3.16.7 - VitalPBX Connect Login Screen.



3.16.8 - VitalPBX Connect Logged in User.

The main screen for the application is the **Keypad** screen. Here you can initiate calls and check your voicemail.

On the top side, you will see your **extension's name** and the **settings menu**, represented by three dots.

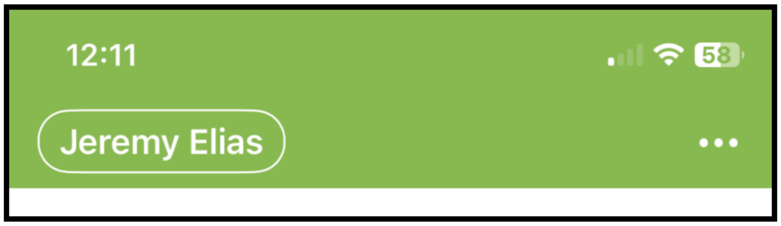
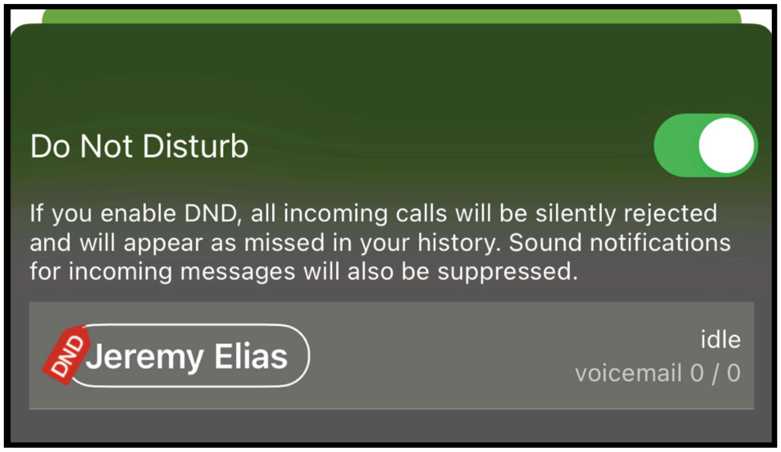


Image 3.16.9 - Upper section of Keypad Screen in VitalPBX Connect.

If you tap on your extension's name, you will see the **Do Not Disturb** option, which can be enabled or disabled. As well as information about your extension and voicemail.



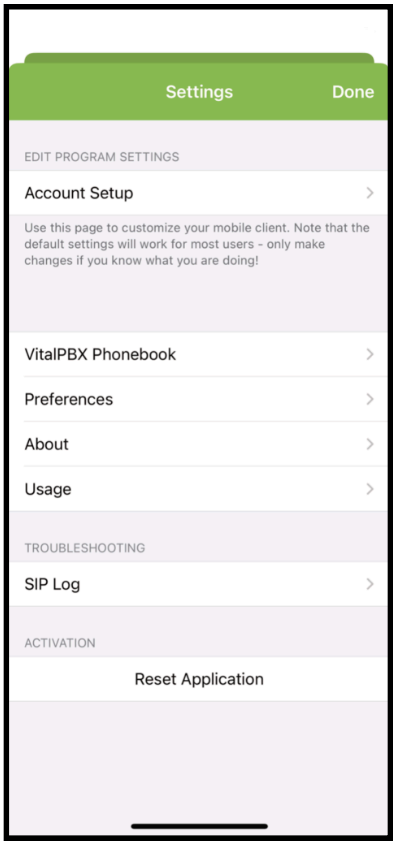
Do Not Disturb

If you enable DND, all incoming calls will be silently rejected and will appear as missed in your history. Sound notifications for incoming messages will also be suppressed.

Image 3.16.10 - Do not disturb feature in VitalPBX Connect.

You can tap the **Close** button at the bottom to close this screen.

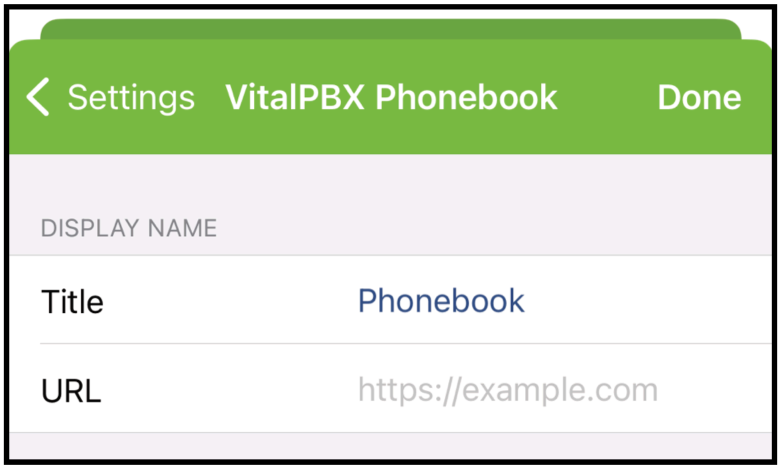
If you tap on the **Settings** option, represented by the three dots, you will see various settings for the application. Here, you will mostly access the VitalPBX Phonebook and Preferences options.



Use this page to customize your mobile client. Note that the default settings will work for most users - only make changes if you know what you are doing!

Image 3.16.11 - VitalPBX Connect Settings.

Under **VitalPBX Phonebook**, you can enter a phonebook URL generated by the Phonebooks add-on in your VitalPBX installation. This allows you to share a centralized phonebook with your end users, so they don't have to enter a contact list manually.



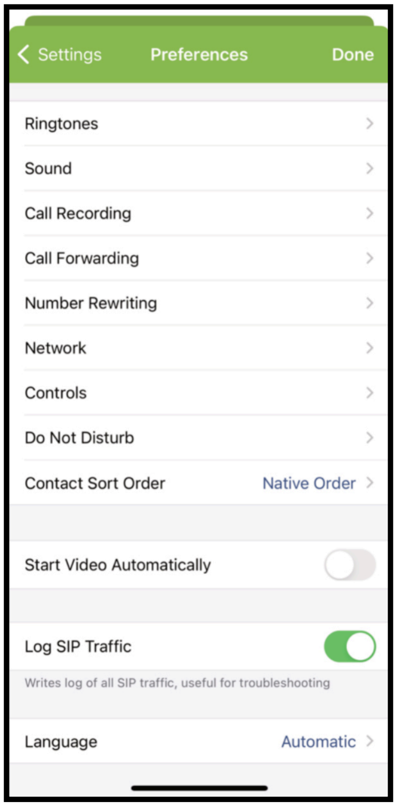
Title: Phonebook

URL: https://example.com

Image 3.16.12 - VitalPBX Phonebook option in VitalPBX Connect.

You can tap on **Done** after you have added the Phonebook URL.

Back in the Settings page, if you tap on **Preferences**, you will see various options you can configure to change the application's behavior.



Contact Sort Order: Native Order

Start Video Automatically: off

Log SIP Traffic: on

Writes log of all SIP traffic, useful for troubleshooting

Language: Automatic

Image 3.16.13 - VitalPBX Connect Preferences.

Here you can change **Ringtones, Call Recording options, Call Forward, add Do Not Disturb rules, and much more**. We encourage you to check out all available options to customize your VitalPBX Connect experience.

You can save any changes by tapping on **Done**.

Back on the application's main screen, you can see other pages you can access at the bottom.

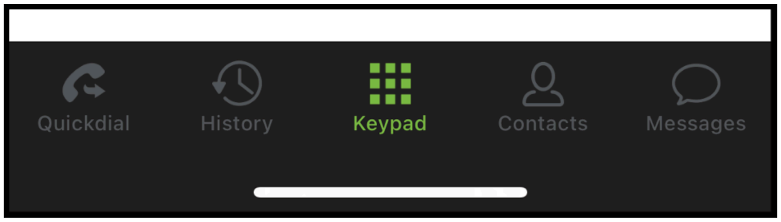
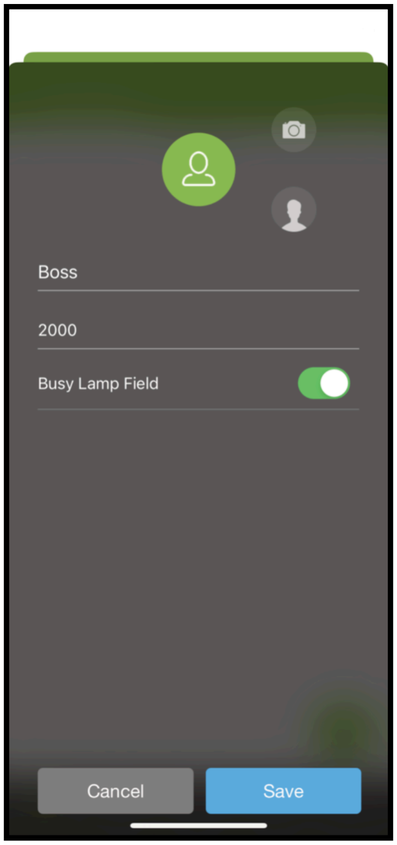


Image 3.16.14 - VitalPBX Connect Pages Navigation.

The first one from the right is the **Quickdial** page. Here, you can add contacts for quick access to call them. To add a quick-dial contact, tap on **Edit** and then the green **+ (plus) button**.



Boss

2000

Busy Lamp Field

Image 3.16.15 - VitalPBX Connect
Quick Dial contact.

When adding a new quick-dial contact, you can enable the **BLF (Busy Lamp Field)** option to monitor their presence. This has to be a local extension number to use the BLF feature. You can also add a profile picture for this quick-dial contact.

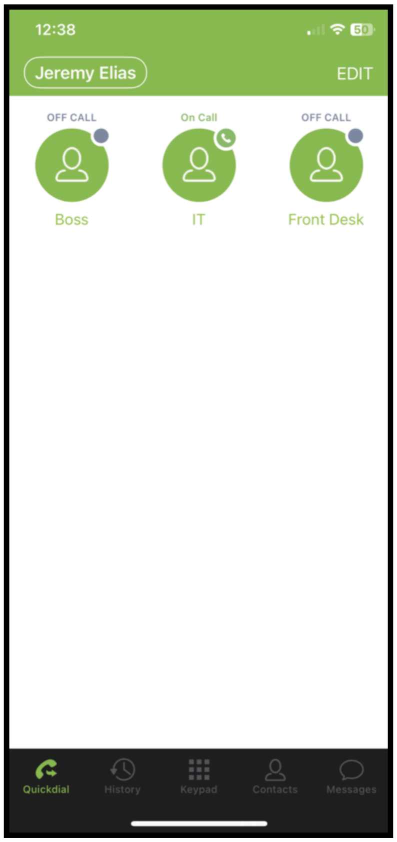


Image 3.16.16 - VitalPBX Connect Quickdial page.

Next is the History page. Here you can see your complete call history.

You can **edit** the list, **remove items**, and **export** the list using the share button in the upper right-hand corner.

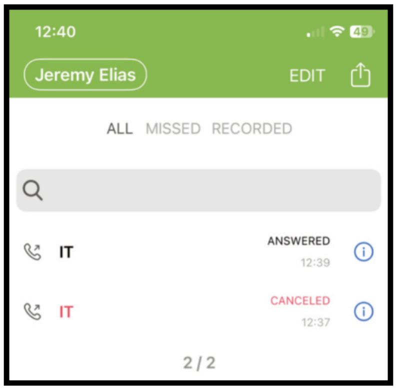


Image 3.16.17 - VitalPBX Connect Call History page.

In the middle, we have our **Keypad**, as we saw before. Here you can initiate calls. If you tap the green call button without dialing any number, you will recall the last number you dialed.

During a call, you will see the following screen.

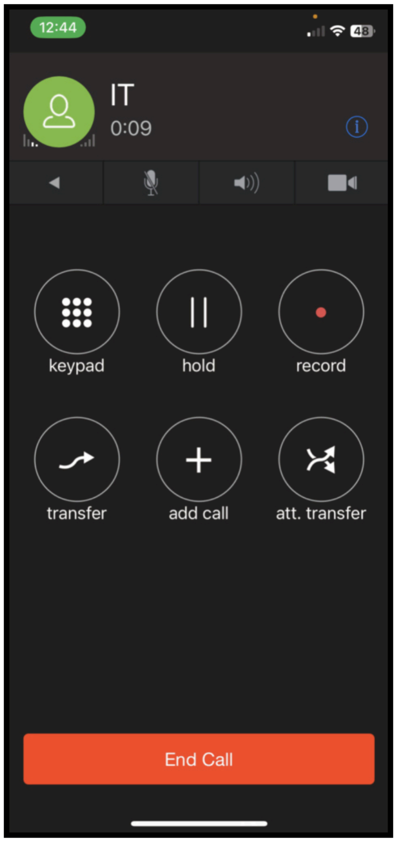
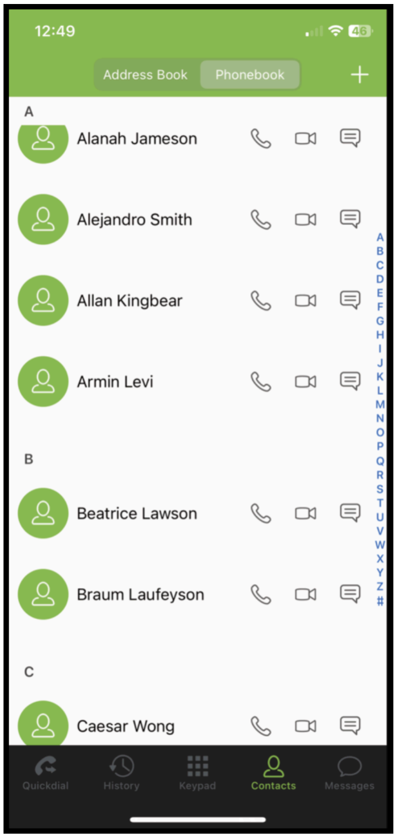


Image 3.16.18 - VitalPBX Connect call screen.

While on a call, you can **mute/unmute**, turn on **speaker** mode, and **initiate your video**. You can also show a **keypad** to perform DTMF actions, place the call on **hold**, **record** the call locally on your device, do **attended or direct transfer**, and **add calls** to create a 3-way conference. At the bottom, you can **End the Call**.

Next, we have the Contacts page. Here you will see the contacts stored in your mobile device and another tab for the central phonebook we added through the application's settings.



A

Alanah Jameson

Alejandro Smith

Allan Kingbear

Armin Levi

B

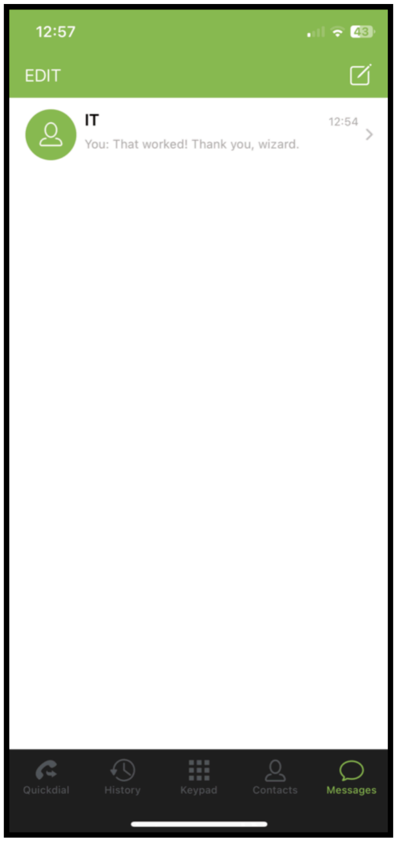
Beatrice Lawson

Braum Laufeyson

C

Caesar Wong

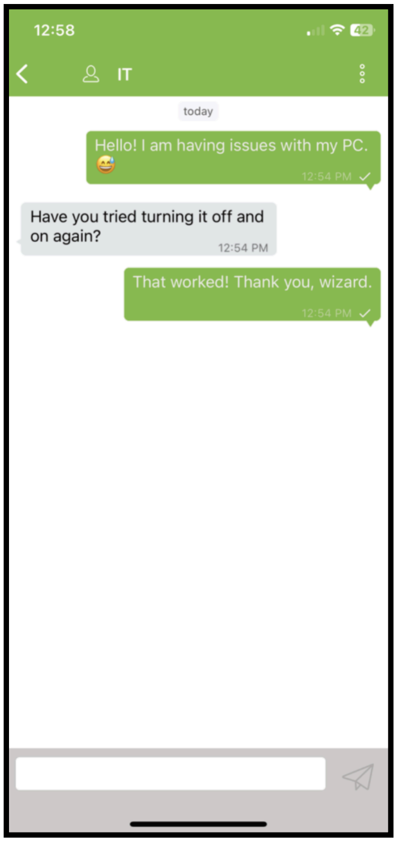
Image 3.16.19 - VitalPBX Connect Contacts page.



IT 12:54

You: That worked! Thank you, wizard.

Image 3.16.20 - VitalPBX Connect Message list.



Hello! I am having issues with my PC. 😅 12:54 PM

Have you tried turning it off and on again? 12:54 PM

That worked! Thank you, wizard. 12:54 PM

Image 3.16.21 - VitalPBX Connect Message conversation.

You can refresh the VitalPBX phonebook tab whenever you open this page or by sliding the contact list down.

This way, your phonebook is always up-to-date with the contacts added on VitalPBX's Phonebook add-on.

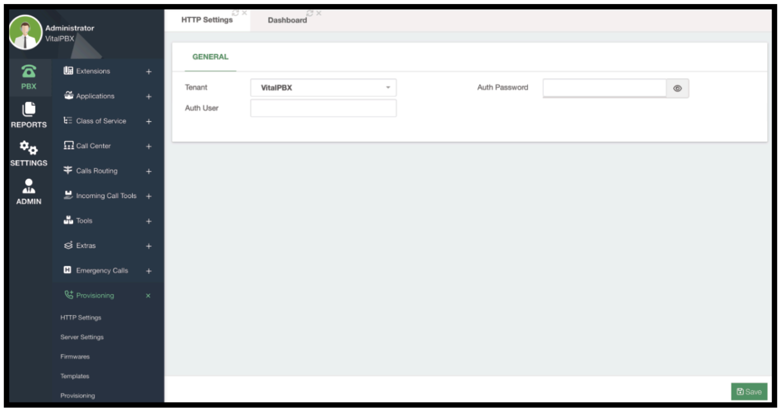
Finally, there is the **Messages** page. Here you can send and receive SIP-based messages with other users in the server that have devices that support SIP-based messaging, such as other VitalPBX Connect devices, VitXi, or select desktop phones. We will see later with the SMS add-on module, you can also send and receive SMS messages through this page, giving your end clients a complete experience of placing and receiving calls, as well as SMS messages. All from a single application. The messaging feature supports regular text and emojis.

You can **edit** and **remove** conversations or **start** a new one with the button in the upper right-hand corner.

With this, you have a complete mobile client that is feature-rich and allows you to have mobility with your extension.

Section 3, Lesson 17 - Phone Provisioning

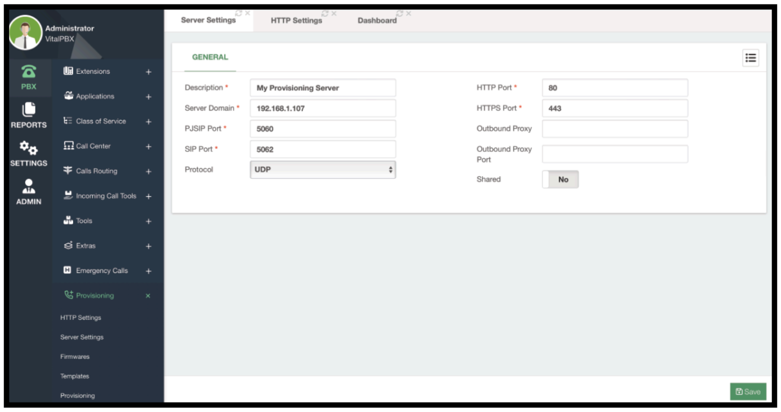
With VitalPBX you can configure end-client devices from the Web UI by using the **Phone Provisioning** module. This add-on module supports most of the most popular brands and their supported models. Once you have the Phone Provisioning add-on module installed, we will begin by going to **PBX > Provisioning > HTTP Settings**. Here, you can assign an **Auth User and Password** that is used by the devices to have access to the provisioning files. This is optional, so this setting can be skipped. You can also select the tenant this Auth User and Password works for. We will see more about Multi-Tenancy later. With the **Auth User and Password** set, you can press the **Save** button in the bottom right-hand corner.



Tenant: VitalPBX
Auth User:
Auth Password:

Image 3.17.1 - Phone provisioning HTTP settings.

Next, we will go to **PBX > Provisioning > Server Settings**. Here, you will enter the **Server and Port Information** to be used by multiple devices during provisioning.

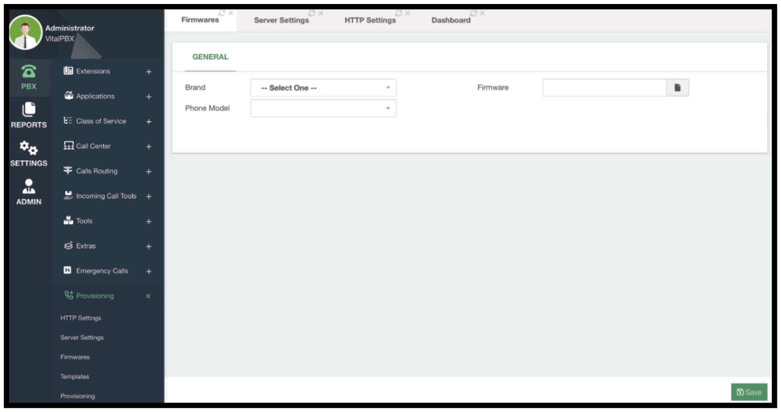


Description *: My Provisioning Server
Server Domain *: 192.168.1.107
PJSIP Port *: 5060
SIP Port *: 5062
Protocol: UDP
HTTP Port *: 80
HTTPS Port *: 443
Outbound Proxy:
Outbound Proxy Port:
Shared: No

Image 3.17.2 - Phone provisioning server settings.

The **Shared** option allows you to share these server settings with other tenants. For the time being, we will leave everything by default, and simply add a **Description** and the **Server Domain**, which can be the VitalPBX server's *IP address* or a valid *FQDN* (Fully-Qualified Domain Name), and then click on the **Save** button in the bottom-right corner.

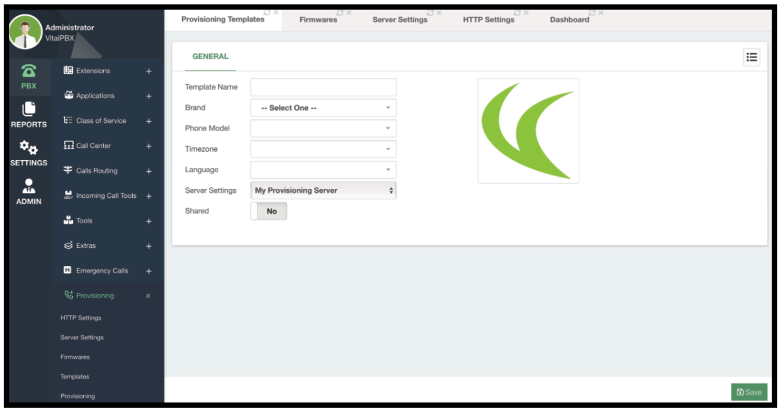
Next, we can go to **PBX > Provisioning > Firmwares**. Here, you can upload any firmware binaries for different phone models. You can choose the **Brand and Phone Model** and upload the **Firmware** binaries. You can then click on the **Save** button in the bottom right-hand corner. For now, we are not going to upload any binaries.



Brand: -- Select One --
Phone Model:
Firmware:

Image 3.17.3 - Phone provisioning firmwares module.

We can now start creating templates that we can use with multiple devices. For this, we will go to **PBX > Provisioning > Templates**.



Template Name:
Brand: -- Select One --
Phone Model:
Timezone:
Language:
Server Settings: My Provisioning Server
Shared: No

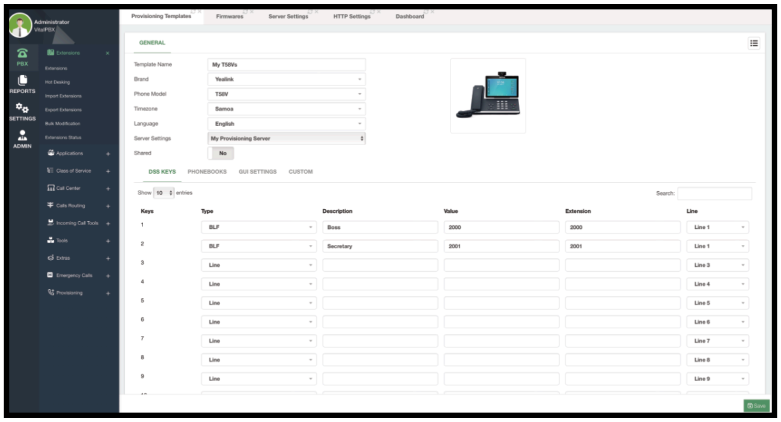
Image 3.17.4 - Phone provisioning templates module.

Here, you will create the templates for the global configurations to be used by the different devices. To create a template, you will need to add a **Template Name**, and then select the **Brand and Phone Model**. For this example, we will be creating a template for a Yealink T58V.

With the brand and phone model selected, you will be presented with an image of the device you are creating a template for. Then, if possible, you can choose the **Timezone and Interface Language** information.

Next, you can select the **Server Settings** we have created previously. Since we only have one, then this is the default server settings that are used. Then, you can click on **Share** toggle, to share the template with other tenants.

Now, below these settings, you will find additional options you can configure for devices. These options will vary based on the phone model.



Template Name	My T58Vs
Brand	Yealink
Phone Model	T58V
Timezone	Samoa
Language	English
Server Settings	My Provisioning Server
Shared	No

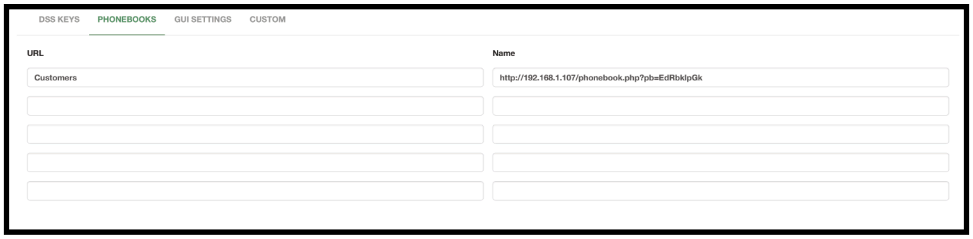
Show 10 entries Search:

Keys	Type	Description	Value	Extension	Line
1	BLF	Boss	2000	2000	Line 1
2	BLF	Secretary	2001	2001	Line 1
3	Line				Line 3
4	Line				Line 4
5	Line				Line 5
6	Line				Line 6
7	Line				Line 7
8	Line				Line 8
9	Line				Line 9

Image 3.17.5 - Phone provisioning template for a T58V. DSS Keys page.

The **DSS Keys** page will have the number of DSS keys the phone models support. The options available will vary to match the options the phone model supports through the provisioning file.

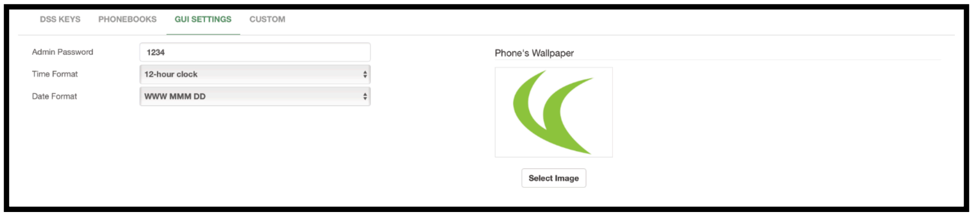
If the device supports it, you can add **Phonebook URLs** for remote directories.



URL	Name
Customers	http://192.168.1.107/phonebook.php?pb=EdRbklpGk

Image 3.17.6 - Phonebooks page for a provisioning template.

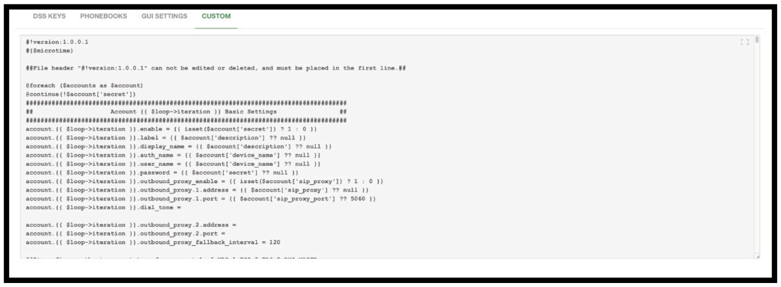
In the **GUI Settings** page, if the device supports it, you can change the **Admin Password**, **Time, and Date Format**. Additionally, you can upload a **Wallpaper Image** if the device allows it. Just make sure that you upload an image in the format supported by the device. To know the format, please refer to the device's documentation.



Admin Password	1234
Time Format	12-hour clock
Date Format	WWW MMM DD

Image 3.17.7 - GUI Settings for a phone provisioning template.

Finally, you have the **Custom** page. The custom page is available for every phone model. This is the full configuration file displayed in case you want to enter any custom configuration that is not presented in the module's UI.

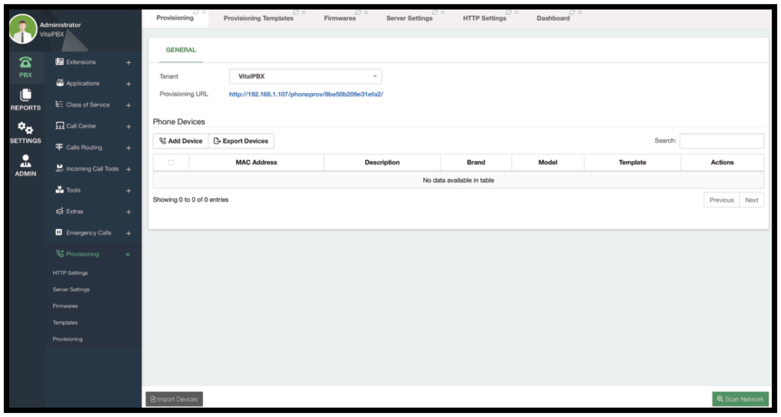


```
#!version:1.0.0.1
#{$microtime}

##File header "#!version:1.0.0.1" can not be edited or deleted, and must be placed in the first line.##

@foreach ($accounts as $account)
@continue(!$account['secret'])
#######################################################################################
##                     Account {{ $loop->iteration }} Basic Settings                  ##
#######################################################################################
account.{{ $loop->iteration }}.enable = {{ isset($account['secret']) ? 1 : 0 }}
account.{{ $loop->iteration }}.label = {{ $account['description'] ?? null }}
account.{{ $loop->iteration }}.display_name = {{ $account['description'] ?? null }}
account.{{ $loop->iteration }}.auth_name = {{ $account['device_name'] ?? null }}
account.{{ $loop->iteration }}.user_name = {{ $account['device_name'] ?? null }}
account.{{ $loop->iteration }}.password = {{ $account['secret'] ?? null }}
account.{{ $loop->iteration }}.outbound_proxy_enable = {{ isset($account['sip_proxy']) ? 1 : 0 }}
account.{{ $loop->iteration }}.outbound_proxy.1.address = {{ $account['sip_proxy'] ?? null }}
account.{{ $loop->iteration }}.outbound_proxy.1.port = {{ $account['sip_proxy_port'] ?? 5060 }}
account.{{ $loop->iteration }}.dial_tone =

account.{{ $loop->iteration }}.outbound_proxy.2.address =
account.{{ $loop->iteration }}.outbound_proxy.2.port =
account.{{ $loop->iteration }}.outbound_proxy_fallback_interval = 120
```

Image 3.17.8 - Custom page for a phone provisioning template.

With the template set up, you can go ahead and click the **Save** button in the bottom right-hand corner.

Now, we can go to the final section for the Phone Provisioning module, the provisioning itself. For this, we go to **PBX > Provisioning > Provisioning**.



Tenant: VitalPBX

Provisioning URL: http://192.168.1.107/phoneprov/8be50b209e31efa2/

☐	MAC Address	Description	Brand	Model	Template	Actions
No data available in table						

Showing 0 to 0 of 0 entries

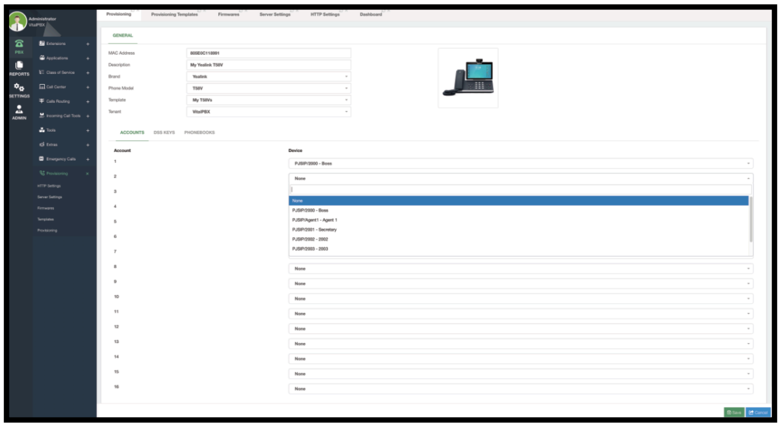
Image 3.17.9 - Provisioning section for the Phone Provisioning module.

Here is where you will add the information for the devices to be provisioned. You will see at the top where you can select the **Tenant** to view their devices, as well as the **Provisioning URL**. We will talk more about this URL in just a moment. To add a device you can click the **Add Device** button.

When adding a device, you will need to enter its **MAC address** and a **Description**. The MAC address can have not the colons. Then, select the **Brand and Phone Model**. This will show you the list of templates available for this phone model. In this case, I chose the Yealink T58V and the template we created previously. You can then select the **Tenant** to which this device belongs.

Beneath these options and the image of the device, you will see three pages. The first one pertains to the **Accounts** to add to the device. The number of accounts will vary depending on the phone model.

The **Devices** listed for the accounts will depend on the selected tenant. What you will see are devices rather than extension numbers only, as you can have multiple devices for the same extension number with VitalPBX.



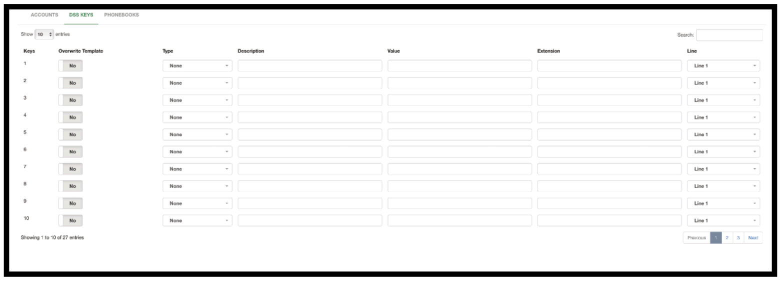
MAC Address	805E0C118991
Description	My Yealink T58V
Brand	Yealink
Phone Model	T58V
Template	My T58Vs
Tenant	VitalPBX

Account	Device
1	PJSIP/2000 - Boss
2	None
3	
4	
5	
6	
7	
8	None
9	None
10	None
11	None
12	None
13	None
14	None
15	None
16	None

None
PJSIP/2000 - Boss
PJSIP/Agent1 - Agent 1
PJSIP/2001 - Secretary
PJSIP/2002 - 2002
PJSIP/2003 - 2003

Image 3.17.10 - Adding a device to provision.

Next in the **DSS Keys** page, you can add custom DSS keys for this device in specific. You can see the option to Overwrite the Template's DSS Key configuration and configure a custom option for this device.



Show 10 entries

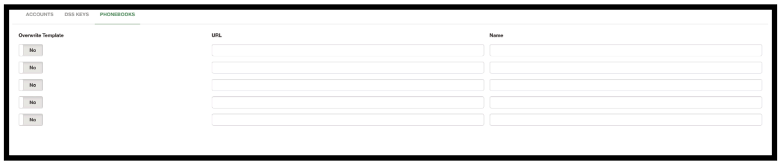
Search:

Keys	Overwrite Template	Type	Description	Value	Extension	Line
1	No	None				Line 1
2	No	None				Line 1
3	No	None				Line 1
4	No	None				Line 1
5	No	None				Line 1
6	No	None				Line 1
7	No	None				Line 1
8	No	None				Line 1
9	No	None				Line 1
10	No	None				Line 1

Showing 1 to 10 of 27 entries

Image 3.17.11 - Device specific DSS key configuration.

If you leave this blank, only the template's configurations will be applied to the device. Finally, we have the **Phonebooks** page. Similarly to the DSS keys page, this will add remote directories to this device specifically if you toggle the **Overwrite Template** option.

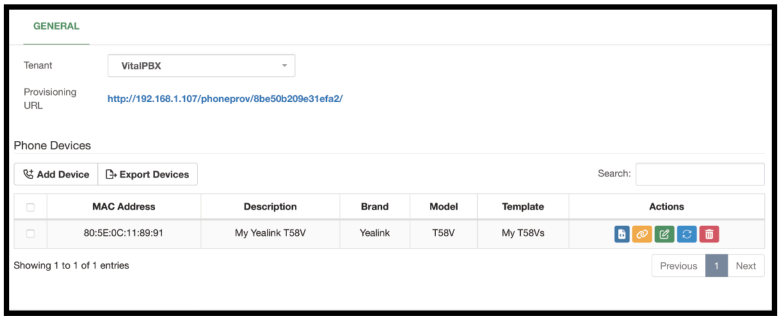


Overwrite Template	URL	Name
No		
No		
No		
No		
No		

Image 3.16.12 - Device specific Phonebook URLs configuration.

In this case, we will leave these pages blank to use the template's configuration, and then we will click on the **Save** button in the lower right-hand corner. This will add the device to the

Phone Devices list, showing us some basic information and a couple of actions we can do from the **Actions** column.



Tenant: VitalPBX

Provisioning URL: http://192.168.1.107/phoneprov/8be50b209e31efa2/

Phone Devices

☐	MAC Address	Description	Brand	Model	Template	Actions
☐	80:5E:0C:11:89:91	My Yealink T58V	Yealink	T58V	My T58Vs	

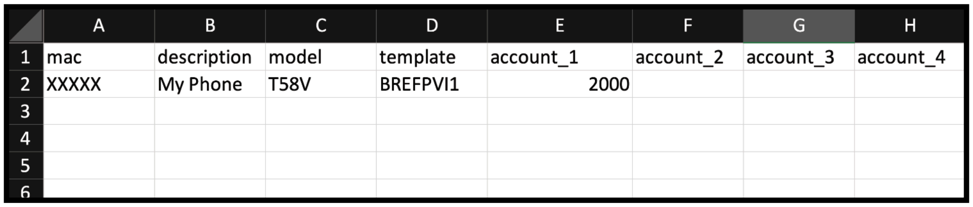
Showing 1 to 1 of 1 entries

Image 3.17.13 - Provisioning section with a device added.

From the Actions column, we can perform the following actions.

- **View the Provisioning File** - This allows you to view the final provisioning file so you can check that everything is correct.
- **Copy the Provisioning URL** - This allows you to copy the provisioning URL so you can apply it to the device or the network using Option 66.
- **Edit** - This lets you edit the configurations for this device.
- **Reboot** - This sends a reboot request to the extension numbers associated with this device. Keep in mind that if multiple devices have the same extension number, they will be rebooted as well.
- **Delete** - This deletes the device from the provisioning list.

Additionally, you can **import and export** multiple devices from the provisioning list. To import them click on the **Import Devices** button in the bottom left-hand corner. This will show you a screen where you can upload a **CSV file** with the configuration for multiple devices. A template is available with the **Download CSV Template** button.



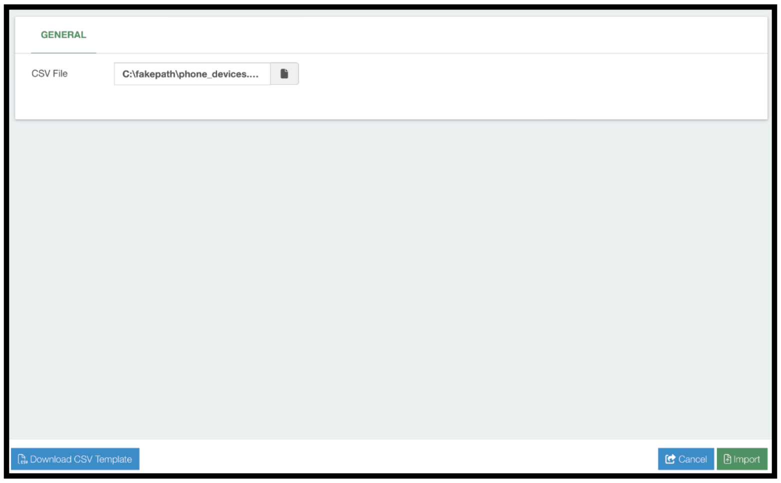
	A	B	C	D	E	F	G	H
1	mac	description	model	template	account_1	account_2	account_3	account_4
2	XXXXX	My Phone	T58V	BREFPVI1	2000			
3								
4								
5								
6								

Image 3.17.14 - Phone provisioning device CSV file.

The CSV file is simple with a field for the **MAC addresses**, **Description**, **Model**, **Template**, and **20 Accounts**.

The Model and Template nomenclature can be retrieved from the template itself. Go to **PBX > Provisioning > Templates**, and select a template from the list in the upper right-hand corner of the module.

You will see that a **Unique Name** has been created for this specific template. You can also see what is the nomenclature used for the **Phone Model** itself. This needs to match exactly in the CSV file so the devices are added correctly.



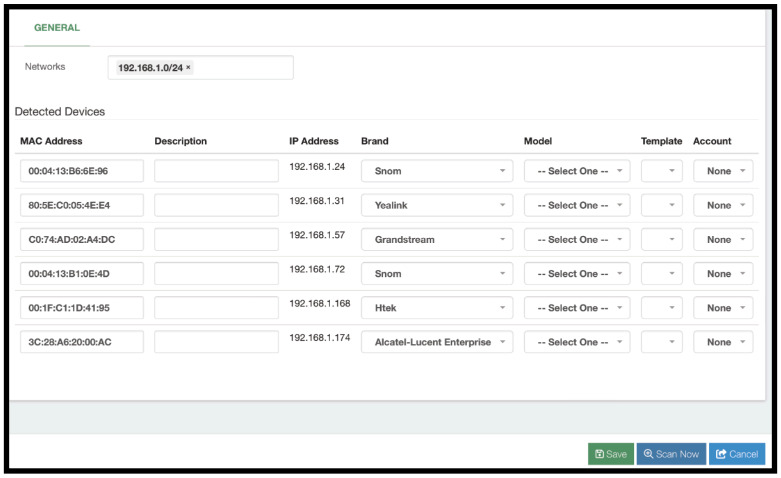
GENERAL

CSV File: C:\fakepath\phone_devices....

Image 3.17.15 - Importing devices for the phone provisioning module.

With the CSV file saved, you can upload the file to the module and click the **Import** button in the lower right-hand corner. Similarly, you can export the devices by clicking the **Export Devices** button. This will generate a CSV file you can use to import in another VitalPBX installation or use as a backup to import later.

One last way to create devices is by **scanning the network**. To do this, click on the **Scan Network** button in the lower right-hand corner. You will need to specify the reachable networks, and then click the **Scan Now** button in the lower right-hand corner. This will scan the local network in search of devices available. Keep in mind that you need to specify the **network mask** for it to scan a network. Also, this feature will only work in **LAN** environments, where the server is located in the same network as the devices. This will ***not*** work in **VPS environments**.



GENERAL

Networks: 192.168.1.0/24 ×

Detected Devices

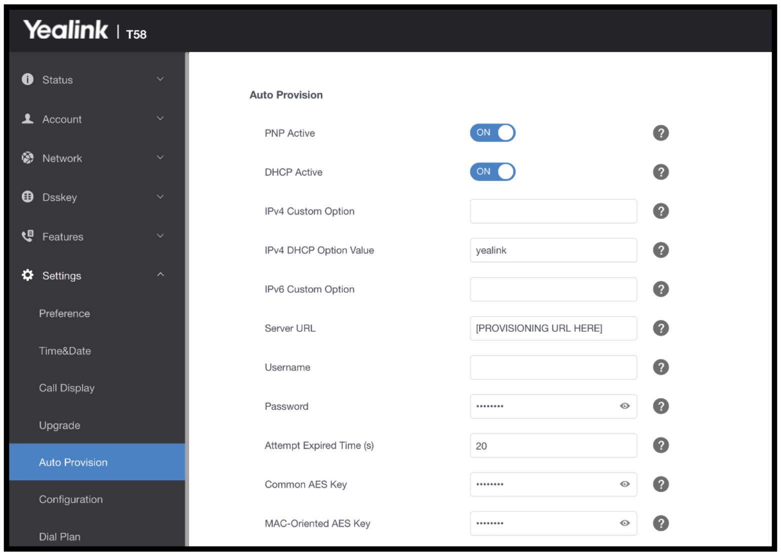
MAC Address	Description	IP Address	Brand	Model	Template	Account
00:04:13:B6:6E:96		192.168.1.24	Snom	-- Select One --		None
80:5E:C0:05:4E:E4		192.168.1.31	Yealink	-- Select One --		None
C0:74:AD:02:A4:DC		192.168.1.57	Grandstream	-- Select One --		None
00:04:13:B1:0E:4D		192.168.1.72	Snom	-- Select One --		None
00:1F:C1:1D:41:95		192.168.1.168	Htek	-- Select One --		None
3C:28:A6:20:00:AC		192.168.1.174	Alcatel-Lucent Enterprise	-- Select One --		None

Image 3.17.16 - Phone provisioning network scan.

Once the network is scanned, you can quickly add a description to the devices, and one account for fast configuration.

After adding the configurations you can click on the **Save** button in the lower right-hand corner.

With the devices added by any of the means mentioned. You will now be able to provision these devices. To do so, you will use the **Provisioning URL** that was created for the tenant. This provisioning URL can be added directly to the devices. The location of the URL will vary based on the device model.

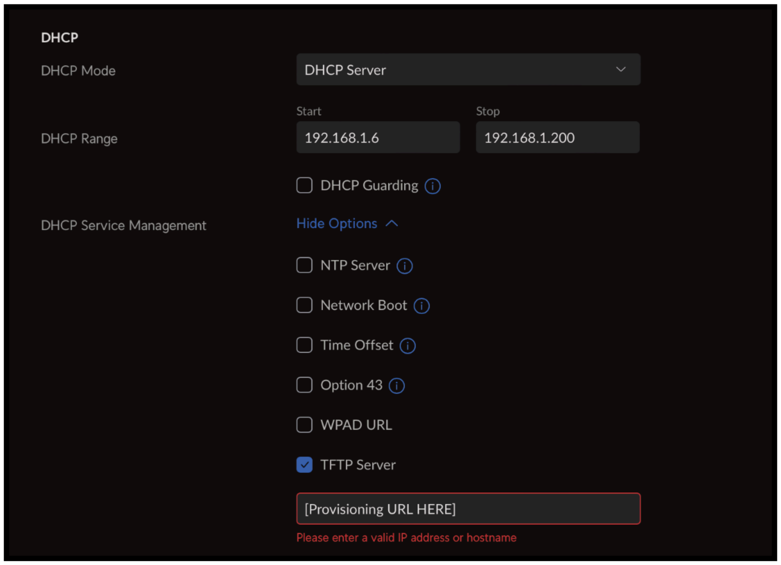


Auto Provision

PNP Active	ON
DHCP Active	ON
IPv4 Custom Option	
IPv4 DHCP Option Value	yealink
IPv6 Custom Option	
Server URL	[PROVISIONING URL HERE]
Username	
Password	••••••••
Attempt Expired Time (s)	20
Common AES Key	••••••••
MAC-Oriented AES Key	••••••••

Image 3.17.17 - Provisioning URL location for a Yealink T58V device.

Note that if you used an **Auth Username and Password**, this needs to be added to the device.



DHCP

DHCP Mode	DHCP Server
DHCP Range	Start 192.168.1.6 Stop 192.168.1.200

DHCP Service Management

□ DHCP Guarding

Hide Options

□ NTP Server

□ Network Boot

□ Time Offset

□ Option 43

□ WPAD URL

☑ TFTP Server

[Provisioning URL HERE]

Please enter a valid IP address or hostname

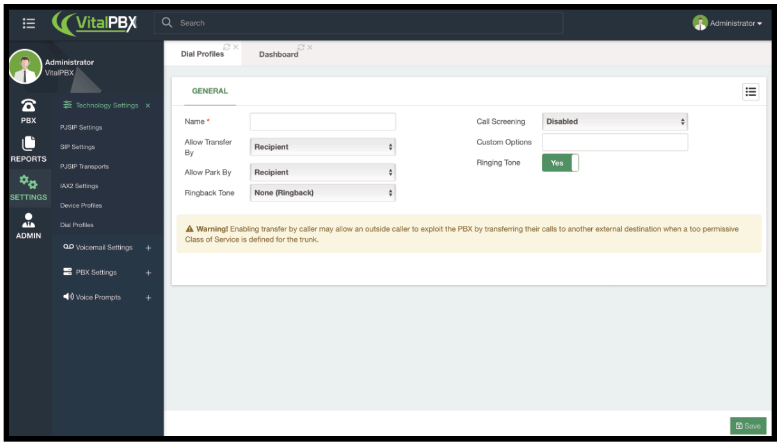
Image 3.17.18 - Setting the provisioning URL using Option 66.

Optionally, you can add the Provisioning URL to the **DHCP options** in your network configuration. The option used is **Option 66**. This will push the Provisioning URL to the devices at boot when obtaining an IP address from the DHCP pool. The location of this option will vary based on your Network configurations.

Now, you can power cycle the devices and they will retrieve their configurations from the Phone Provisioning module in VitalPBX. You can continue to make changes to these configurations and reboot the devices from your VitalPBX installation.

Section 3, Lesson 18 - Dial Profiles

You might have noticed that various areas in VitalPBX, especially Extensions, use what is called **Dial Profiles**. Dial Profiles are configuration templates you can use to easily configure common settings among multiple features at once. To configure a Dial Profile, we must go to **Settings > Technology Settings > Dial Profiles**.



Name *

Allow Transfer By: Recipient

Allow Park By: Recipient

Ringback Tone: None (Ringback)

Call Screening: Disabled

Custom Options

Ringing Tone: Yes

⚠ **Warning!** Enabling transfer by caller may allow an outside caller to exploit the PBX by transferring their calls to another external destination when a too permissive Class of Service is defined for the trunk.

Image 3.18.1 - Dial Profiles module in VitalPBX.

To create a Dial Profile you must enter a **Name** to recognize it. Then, we have various options we can modify for this Dial Profile. We can select who is allowed to **Transfer or Park** the calls. Either the *Recipient, Caller, or Both*. This will allow the selected option to use the feature code to transfer or park the call.

> ***Note:*** *Be sure that you know why you want to change this option, as stated by the* ***Warning*** *in the module,* ***"Enabling transfer by the caller may allow an outside caller to exploit the PBX by transferring their calls to another external destination when a too permissive Class of Service is defined for the trunk."***

You can also change the **Ringback Tone** the extensions or features will use. This can playback the default ringing tone, or any Music on Hold class you have created. You can turn off the **Ringing Tone** altogether as well.

You can then turn on **Call Screening** and select between having it *Disabled, Always Ask, or Only Once*. This will prompt the incoming caller to record their name, which will be played back when the call is answered. Finally, you can enter **Custom Options**, which are custom dial parameters you can run that are not specified in the GUI.

For now, we will continue to use the Default Dial Profile, but if you have made a new one, make sure to click on the **Save** button in the lower right-hand corner, and **apply changes**.

Conclusion

With this, we have now covered everything surrounding extensions. This is where the bulk of the work for any PBX configuration is situated. As you can see, there is a plethora of options you can follow to make the most out of your VitalPBX installation. For now, let's move on to the next section regarding another key point for your VitalPBX configuration, which is your PBX connectivity.

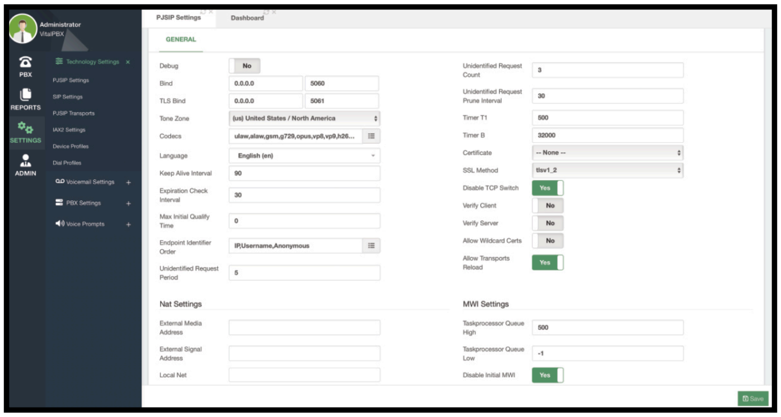
SECTION 4 - CONNECTIVITY

Introduction

Now that you have internal communications established, we must begin looking into connecting our VitalPBX with the outside world, or other PBX systems. In this section, we will explore everything regarding **Trunks, Routing, Basic Permissions, and Emergency** scenarios.

Section 4, Lesson 1 - NAT Settings and Firewall Overview

When you are planning on accessing VitalPBX from remote locations, and your server is located in a LAN Network you must configure your router to forward the necessary ports for registration and voice traffic. This means that you will need to be able to translate incoming data from your Public network with a Public IP address to the VitalPBX LAN IP address. For this, you can configure the **NAT Settings** for PJSIP so VitalPBX can translate that communication. To do this, we will need to go to **Settings > Technology Settings > PJSIP Settings**.



Debug	No	
Bind	0.0.0.0	5060
TLS Bind	0.0.0.0	5061
Tone Zone	(us) United States / North America	
Codecs	ulaw,alaw,gsm,g729,opus,vp8,vp9,h26...	
Language	English (en)	
Keep Alive Interval	90	
Expiration Check Interval	30	
Max Initial Qualify Time	0	
Endpoint Identifier Order	IP,Username,Anonymous	
Unidentified Request Period	5	

Nat Settings

External Media Address	
External Signal Address	
Local Net	

Unidentified Request Count	3
Unidentified Request Prune Interval	30
Timer T1	500
Timer B	32000
Certificate	-- None --
SSL Method	tlsv1_2
Disable TCP Switch	Yes
Verify Client	No
Verify Server	No
Allow Wildcard Certs	No
Allow Transports Reload	Yes

MWI Settings

Taskprocessor Queue High	500
Taskprocessor Queue Low	-1
Disable Initial MWI	Yes

Image 4.1.1 - PJSIP Settings module.

Here, you will see general configurations for PJSIP. Explaining all the features here is out of the scope of the VitalPBX guide, but we are interested in the bottom section called NAT Settings.

The only two fields we need to configure are **External Media Address, External Signal Address, and Local Net**. Both *External Media and External Signal Addresses* will need your Public IP Address. The Local Net field will require the Local Network where your VitalPBX installation is located.

The format for the Local Net field uses the dot cero for the whole network followed by a slash and the Netmask in CIDR or dotted format. E.g. 192.168.0.0/24 or 192.168.168.0.0/255.255.255.0.

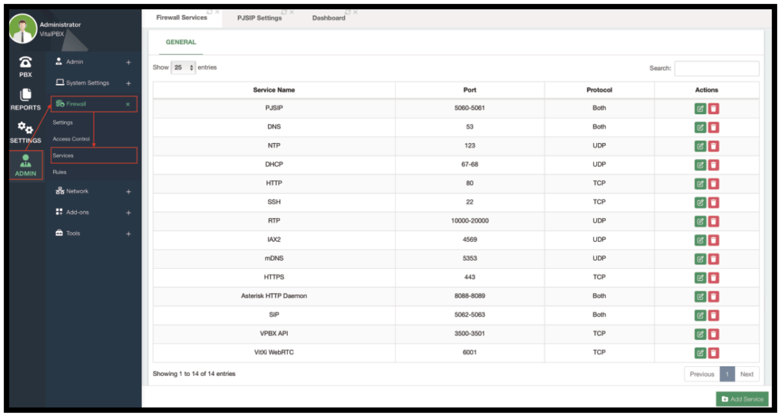
With this, you can place calls normally when registered remotely.

As mentioned previously, NAT configurations are mostly necessary when you have your VitalPBX in a local environment and want to publish it with a Public IP Address/FQDN. And as we said, you will need to port forward ports on your router to be able to do this.

The necessary ports are as follows.

- **5060,5061** - These are used for PJSIP registration.
- **10,000-20,000** - These are ports used for RTP traffic (Voice).

Additional ports can be forwarded depending on your use. You can see a full list of ports used under **Admin > Firewall > Services.**



Service Name	Port	Protocol	Actions
PJSIP	5060-5061	Both	
DNS	53	Both	
NTP	123	UDP	
DHCP	67-68	UDP	
HTTP	80	TCP	
SSH	22	TCP	
RTP	10000-20000	UDP	
IAX2	4569	UDP	
mDNS	5353	UDP	
HTTPS	443	TCP	
Asterisk HTTP Daemon	8088-8089	Both	
SIP	5062-5063	Both	
VPBX API	3500-3501	TCP	
VitXi WebRTC	6001	TCP	

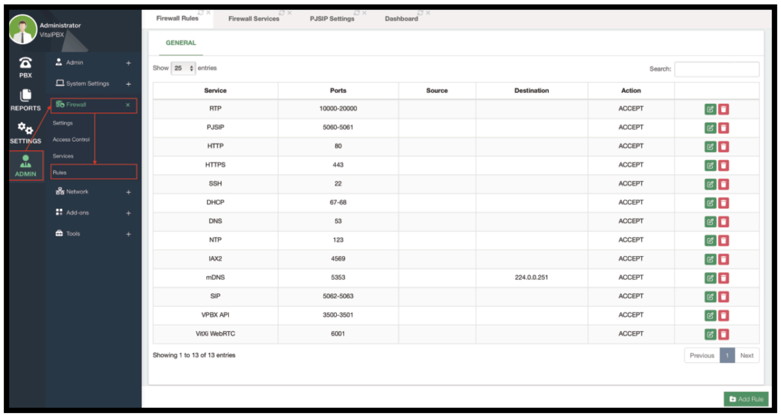
Showing 1 to 14 of 14 entries

Image 4.1.2 - Firewall Services module.

By default, these are ports used.

Service Name	Port	Protocol
PJSIP	5060 - 5061	UDP, TCP
DNS	53	UDP, TCP
NTP	123	UDP
DHCP	67-68	UDP
HTTP	80	TCP
SSH	22	TCP
RTP	10000-20000	UDP
IAX2	4569	UDP
mDNS	5353	UDP
HTTPS	443	TCP
Asterisk HTTP Daemon	8088-8089	UDP, TCP
VPBX API	3500-3501	TCP
VitXi WebRTC	6001	TCP

All of these ports can be modified with the green Edit button under the Actions column. You can also control how you treat these ports in VitalPBX under **Admin > Firewall > Rules**.



Service	Ports	Source	Destination	Action	
RTP	10000-20000			ACCEPT	
PJSIP	5060-5061			ACCEPT	
HTTP	80			ACCEPT	
HTTPS	443			ACCEPT	
SSH	22			ACCEPT	
DHCP	67-68			ACCEPT	
DNS	53			ACCEPT	
NTP	123			ACCEPT	
IAX2	4569			ACCEPT	
mDNS	5353		224.0.0.251	ACCEPT	
SIP	5062-5063			ACCEPT	
VPBX API	3500-3501			ACCEPT	
VitXi WebRTC	6001			ACCEPT	

Showing 1 to 13 of 13 entries

Image 4.1.3 - Firewall Rules module.

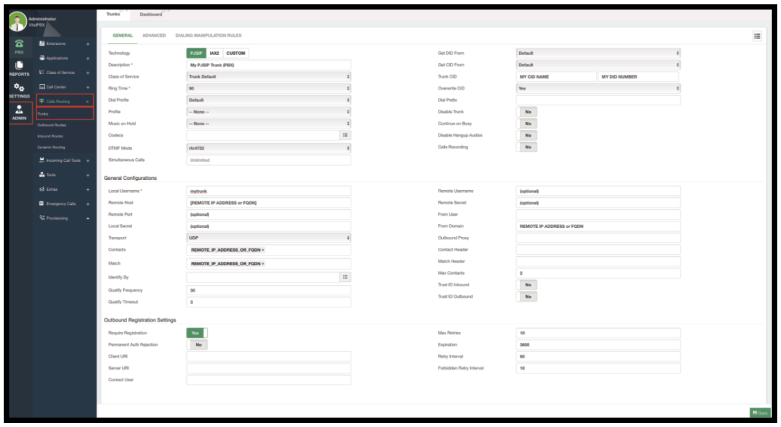
Here, you can choose if you are going to **ACCEPT, DROP, or REJECT** incoming packets through these ports. We will see in the VitXi Guide, how we manage the configuration of these ports for additional configurations.

Being able to modify and configure these ports and firewall rules is another way to secure your VitalPBX installation.

Section 4, Lesson 2 - Trunks

You now have internal communication between extensions, and with NAT Settings and your firewall configurations, you can connect from anywhere. Now, we will need to connect with the outside world. This can be done by configuring **Trunks**. Trunks are connections between VitalPBX and **VoIP providers or other PBX/VoIP systems**.

We must go to PBX > Calls Routing > Trunks to start your trunk configurations.



Technology	PJSIP / IAX2 / CUSTOM
Description *	My PJSIP Trunk (PBX)
Class of Service	Trunk Default
Ring Time *	90
Dial Profile	Default
Profile	-- None --
Music on Hold	-- None --
Codecs	
DTMF Mode	rfc4733
Simultaneous Calls	Unlimited
Get DID From	Default
Get CID From	Default
Trunk CID	MY CID NAME / MY DID NUMBER
Overwrite CID	Yes
Dial Prefix	
Disable Trunk	No
Continue on Busy	No
Disable Hangup Audios	No
Calls Recording	No

General Configurations

Local Username *	mytrunk
Remote Host	[REMOTE IP ADDRESS or FQDN]
Remote Port	(optional)
Local Secret	(optional)
Transport	UDP
Contacts	REMOTE_IP_ADDRESS_OR_FQDN ×
Match	REMOTE_IP_ADDRESS_OR_FQDN ×
Identify By	
Qualify Frequency	30
Qualify Timeout	3
Remote Username	(optional)
Remote Secret	(optional)
From User	
From Domain	REMOTE IP ADDRESS or FQDN
Outbound Proxy	
Contact Header	
Match Header	
Max Contacts	2
Trust ID Inbound	No
Trust ID Outbound	No

Outbound Registration Settings

Require Registration	Yes
Permanent Auth Rejection	No
Client URI	
Server URI	
Contact User	
Max Retries	10
Expiration	3600
Retry Interval	60
Forbidden Retry Interval	10

Image 4.2.1 - PJSIP Trunk in the Trunks module.

Here, you can create as many trunks as you need. We will not go through all the fields that the trunk configurations have, as you can leave most of them by default unless it is required by the other PBX or VoIP operator.

First, we will go through the most basic requirements to connect with another PBX/VoIP system.

> ***Note:*** *Remember, with VitalPBX 4 and onward, we have deprecated the SIP protocol. This is due to SIP no longer being supported. PJSIP is now the main protocol used. PJSIP is completely backward compatible with SIP trunks. Even if your VoIP Provider uses a SIP trunk, you can easily connect that SIP trunk with a PJSIP trunk.*

The first step we need to do is to select the technology for this trunk. This can be found in the first section at the top. With VitalPBX 4, we can choose between **PJSIP, IAX2, or Custom**. Most of the trunks you will create will be using **PJSIP**. **IAX2** trunks can be used when connecting two Asterisk-based systems, and **Custom** can be used to connect with other protocols, like H323, for legacy PBX systems. For now, select PJSIP.

Next, you will need to enter a **Description**. This can be anything to identify the trunk. The other field you can configure from here is the **Caller ID Name and Number** used for this trunk, under **Trunk CID**. You can use this to **Overwrite the External CID** from an extension when dialing out through this trunk. Under **Overwrite CID**, you can choose between **Yes**, to overwrite the

CID always, **No**, so the CID is not overwritten, or **If Not Provided**, when an extension or module does not have an *External CID*.

Trunks can also be **enabled or disabled** at any time. This way if you want to turn off a trunk without deleting it.

You can leave the rest of the fields for this first section with their default values. For the time being.
Optionally, some additional options you can configure are as follows.

- **Class of Service -** We will look more in-depth about this later, but this is used to control the outgoing calls through this trunk.
- **Simultaneous Calls -** The number of simultaneous calls that can go through this trunk.
- **Get DID and Get CID From -** The header from where you will retrieve the DID or CID information.
- **Dial Prefix -** A prefix number that is automatically added if needed at the beginning of the number you are calling through this trunk.
- **Calls Recording -** This will record all the calls going through this trunk. Please note that enabling call recording in multiple stages of the call, i.e. Extensions, Routes, and Trunks, will generate duplicates of recordings.

Most of these can be left with their default values, and are changed only if the remote system requires it.

With everything set in the first section, we can go to the **General Section**. Here, is where all of our connection configurations will go. From here, we first enter a **Local Username**. This username is mostly for identification, but keep in mind that if you are connecting to another PBX system that requires authentication, this will be the username you need to enter in that system. If you are using User Authentication in the remote system, you will also need to enter a **Local Secret** as well.

Next, we enter the **Remote Host**. This is the IP Address or FQDN for the system we wish to connect with. The **Remote Port** is optional and is used only if the port for the connection is different in the remote system. When left blank, the default port will be used, i.e. 5060 for PJSIP.

Then we have **Contacts** and **Match**. These are the PJSIP configurations needed to have a successful connection. *Contact* is how the remote system connects with your VitalPBX. This can be set in SIP URI format, ***sip:REMOTE_IP_ADDR:5060***. This setting is not always needed for the connection. *Match* allows us to match the connection from a specific FQDN or IP Address.

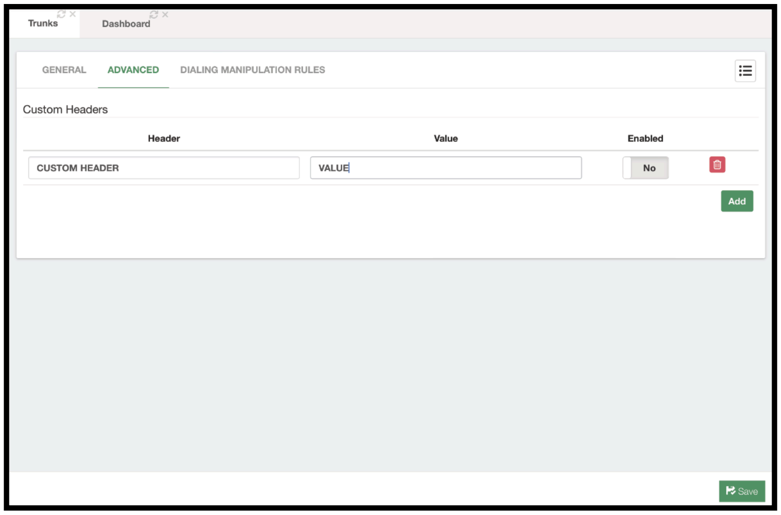
With only these configurations you would have a connection with IP Authentication ready. You can **Save** and **Apply Changes** from here. Additionally, you can use Registration Authentication by entering the **Remote Username and Secret**. These would be the username and password created in the remote system. When using this option, you will need to enable the **Require Registration** option in the **Outbound Registration Settings**.

Optionally, and this depends on the remote system, you can modify the **From Domain** header and add the remote IP Address or FQDN.

This is everything you need to create a Trunk with **User Authentication**. You can now **Save and Apply Changes**.

Additional features that exist for the General Configurations and Outbound Registration are optional and are only modified if required by the remote system.

If you require entering a header that is unavailable via the GUI, you can head over to the Advanced Tab and enter any **custom header** with their respective **values**.
Once again, explaining all the options for PJSIP would go out of scope with this manual and guide, but they are here if you understand what is required by the remote system you want to connect with.

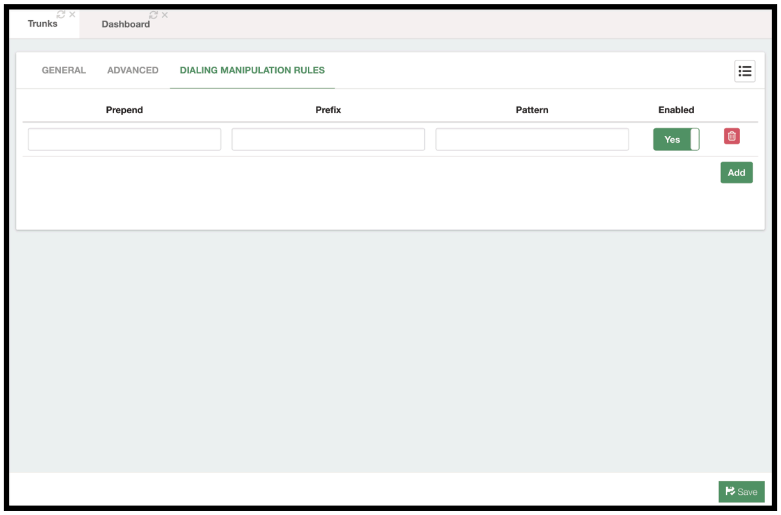


Custom Headers

Header	Value	Enabled
CUSTOM HEADER	VALUE	No

Image 4.2.2 - Trunks Advanced Tab.

Finally, we have the **Dialing Manipulation Rules**. This is not required for outbound calling but allows you to manipulate the dialed number depending on the trunk. So you can use this to manipulate the dialed numbers going through this trunk.



Prepend	Prefix	Pattern	Enabled
			Yes

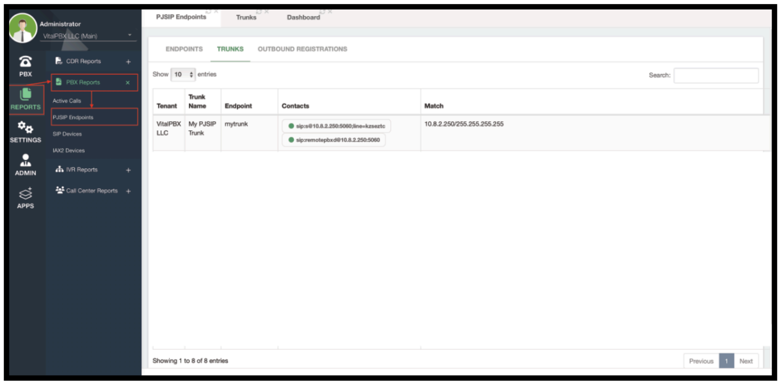
Image 4.2.3 - Trunks Dialing Manipulation Rules

Now you have a fully functioning trunk connecting you with another PBX system or VoIP Provider.

There is a sleuth of VoIP providers nowadays, so covering them all in a single guide would be impossible, so we created multiple blogs on our website that cover many VoIP providers we have worked with. You can check them out at the following URL, https://vitalpbx.com/?s=PJSIP.

> ***Note:*** *you may have noticed a lack of DAHDI, or telephony interface options for telephony interface cards. With VitalPBX 3 and onward support for DAHDI has been deprecated, and is no longer supported. In case you need to connect VitalPBX with an E1, T1, FXO, or FXS interface, we recommend using a Telephony Gateway appliance that will accept the telephony connection and connect with VitalPBX via PJSIP trunk.*

To verify the connection status of your newly created trunk, you can head over to **Reports > PBX Reports > PJSIP Endpoints**. Here, you can go to the Trunks tab and see the status of your trunk.



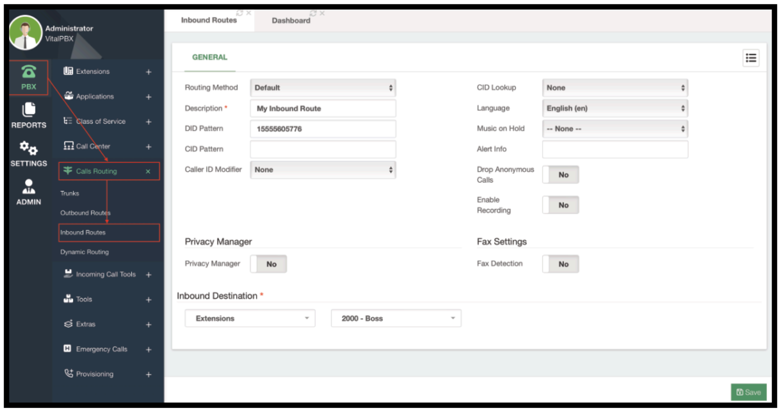
Tenant	Trunk Name	Endpoint	Contacts	Match
VitalPBX LLC	My PJSIP Trunk	mytrunk	● sip:s@10.8.2.250:5060;line=kzseztc ● sip:remotepbxd@10.8.2.250:5060	10.8.2.250/255.255.255.255

Showing 1 to 8 of 8 entries

Image 4.2.4 - Trunk registration in the PJSIP Endpoints report.

Section 4, Lesson 3 - Inbound Routes

Now that you have a trunk created we need to be able to place incoming and outgoing calls through it, these are called routes. First, let's deal with incoming calls using **Inbound Routes**. For this, we need to go to **PBX > Calls Routing > Inbound Routes**.



Routing Method: Default
Description *: My Inbound Route
DID Pattern: 15555605776
CID Pattern:
Caller ID Modifier: None
CID Lookup: None
Language: English (en)
Music on Hold: -- None --
Alert Info:
Drop Anonymous Calls: No
Enable Recording: No

Privacy Manager

Privacy Manager: No

Fax Settings

Fax Detection: No

Inbound Destination *

Extensions | 2000 - Boss

Image 4.3.1 - Inbound Routes module.

The first thing we need to do is choose our **Routing Method**. There are two types of routing methods, **Default and DID Range**. Let's begin with the default routing method. Next, we need to add a **Description** to identify this inbound route.

The default routing method identifies the DID or phone number the calling party is dialing and routes it to a destination.

You can enter a specific **DID Pattern** so whenever someone dials that pattern calls get routed. Adding a **CID (Caller ID) Pattern** will make it so only when someone with this specific Caller ID calls this DID pattern, will get routed with this inbound route. If you leave both the DID and CID pattern fields empty, this means that anyone calling any DID pattern with any CID pattern will get routed by this inbound route. This is what we usually call an ***Any/Any*** route.

The last thing we need to configure an inbound route is the **Inbound Destination**. This is an extension or module that can take an incoming call in VitalPBX. This can be an IVR, a Queue, a Ring Group, etc. For the example, I set it to the Boss Extension, but you can use anything of your choosing.

You can now **Save** and **Apply Changes**. If you followed the Extension creation section for this guide, you might remember that we assigned an Incoming Route to the Boss extension. You can find the list of all incoming routes in the list located in the upper right-hand corner.

We have more tools that you can use for incoming calls. We will see more of them later, but for now, these are other configurations you can have with inbound routes.

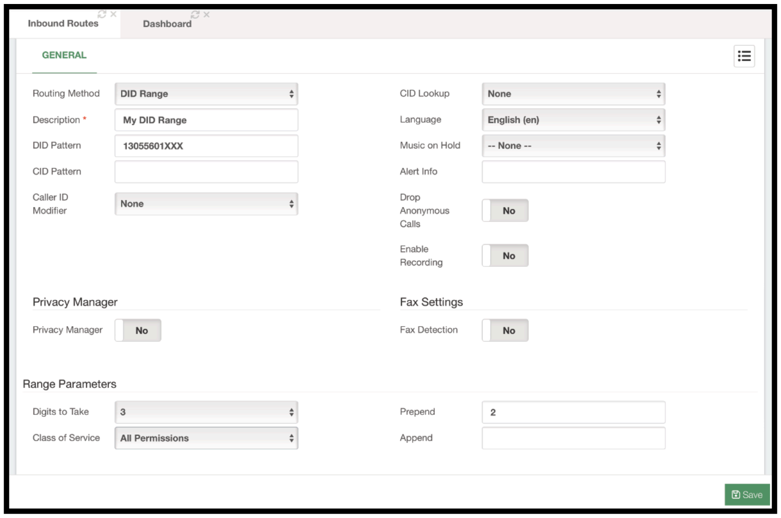
- **Language -** Voice prompt language will be changed when calling through this inbound route.
- **Music on Hold -** You can set the MOH that plays back when calling through this inbound route.
- **Alert Info -** This is a header that will be added to calls going through this inbound route. The Alert-Info header can be used for custom ringtones with endpoint devices.
- **Drop Anonymous Calls -** Any call without a caller ID is identified as Anonymous. With this option, you can drop any call that comes without a CID.
- **Enable Recording -** This will enable call recording of calls going through this inbound route. Again, remember if you enable call recording in multiple stages of an incoming call, duplicates of that call recording will be generated.
- **Privacy Manager -** This feature will prompt the calling party to enter their caller ID information if they don't have one.
- **Fax Detection -** This will detect if the incoming call is a fax message. With this enabled, you can change the Detection Time in seconds and set a Fax Destination. Keep in mind this will add a delay to incoming calls as it will need to detect the Fax tones.

So now, you will have an inbound route. If you connected your VitalPBX installation using a trunk with your VoIP Provider, you can try calling any number associated with that trunk to test the incoming call.

If everything matches, the call will now be redirected to the Inbound Destination.

Optionally, you can also create an inbound route with the **DID Range** routing method. What this method allows you to do, is to have multiple routes with a range of DIDs.

Let's say, you have a range of DIDs that goes from 13055601000-13055601100. You want to associate the last three digits of the DIDs with extensions you have created. So, what we can do in this case is change the routing method from Default to DID Range. And for the DID Pattern, we can enter 13055601XXX. The Xs in this case means any number between 0 and 9. We will see more about how variables work in patterns in the *Outbound Routes* lesson.



Routing Method	DID Range	CID Lookup	None
Description *	My DID Range	Language	English (en)
DID Pattern	13055601XXX	Music on Hold	-- None --
CID Pattern		Alert Info	
Caller ID Modifier	None	Drop Anonymous Calls	No
		Enable Recording	No

Privacy Manager

Privacy Manager	No

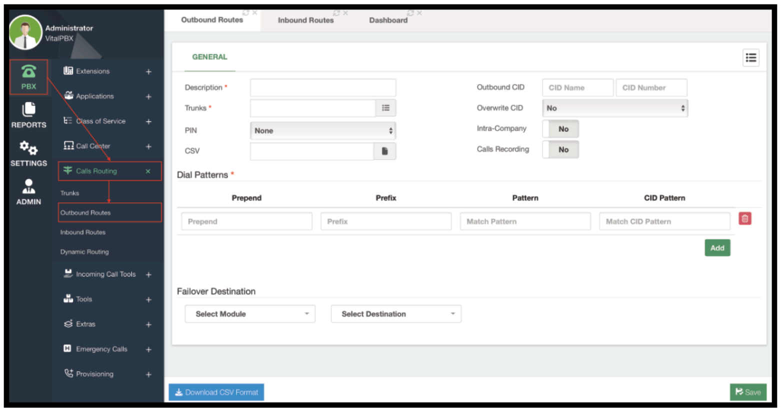
Fax Settings

Fax Detection	No

Range Parameters

Digits to Take	3	Prepend	2
Class of Service	All Permissions	Append	

Image 4.3.2 - Inbound Route with DID Range.



Description *

Trunks *

PIN: None

CSV

Outbound CID: CID Name | CID Number

Overwrite CID: No

Intra-Company: No

Calls Recording: No

Dial Patterns *

Prepend	Prefix	Pattern	CID Pattern
Prepend	Prefix	Match Pattern	Match CID Pattern

Failover Destination

Select Module | Select Destination

Image 4.4.1 - Outbound Routes module.

You will notice that instead of a single destination, we now have the **Range Parameters**. Here we will establish the **Digits to Take**. So for our example, we will take the last 3 digits from our range. Then we have **Prepend and Append**. Prepend will add digits before the digits we took, and Append will add them after. Let's say our extensions go from 2000 to 2100. Since we took the last three digits from our range which goes from 000 to 100, we can prepend a 2, so the numbering matches with our extensions.

For **Class of Service**, we can leave the default All Permissions CoS. We will see more about Class of Service later.

With these configurations, whenever someone calls any DID from our Range, calls will be routed to the matching extension. This way, we don't need to create an incoming route for every extension. Remember to **Save** and **Apply Changes.**

Section 4, Lesson 4 - Outbound Routes

Now that you have the incoming calls dealt with, let's move on to outgoing calls. For this, we can go to **PBX > Calls Routing > Outbound Routes**. **Outbound Routes** will allow you to place outgoing calls through a trunk.

To create your outbound route, you must add a **Description** to identify this outbound route. Next, we select a trunk. This can be the trunk we have created previously.

Afterward, we have the **Dial Patterns**. This is the heart of outbound routes. Outbound routes will identify the number you dial using these patterns, and route them through the trunk. There are three components for a dial pattern: **Prepend, Prefix, Pattern, and CID Pattern**.

Prepend adds the value entered to the beginning of the number you dial. So, if we add a 1 here, for example, whenever you dial a number that matches this pattern, 1 is added automatically at the beginning.

Prefix, is a number that is removed from the beginning of the number you dial. Say you want to use the number 9 to always use this specific outbound route. You establish 9 as a prefix, and dial the number with the 9 at the beginning, *i.e. 913055605776*. The 9 is removed and you will be sending 13055605776, which is accepted by the PBX or VoIP system the trunk is connected to.

The **Pattern** is the base for the Dial Patterns. With this, you establish the numbers that can be dialed. Patterns become powerful with the available variables. This way, you don't have to configure every existing number possibility.
Here is an explanation of the available patterns and some examples of how they work.

- **X -** This is any number between 0 and 9.
- **Z -** This is any number between 1 and 9.
- **N -** This is any number between 2 and 9.
- **. -** The dot is a Wild Card that matches 1 or more digits immediately. This means that from the dot onward, you can have as many digits as you need.
- **! -** The exclamation mark or bang is a Wild Card that matches 0 or more digits immediately. This means that from the bang onward, you can have no more digits or as many as you need.
- **[1,2,3-9] -** Whenever you encapsulate digits in square brackets, you establish the numbers that can be used in this character space. You can establish individual digits separated by a comma or a range of digits separated by a dash (-).
- **[a-z] -** Matches the characters in lowercase. Remember, in telephony, letters are also valid destinations to dial.
- **[A-Z] -** Matches the characters in uppercase. Again, in telephony, letters are also valid destinations to dial.

With these variables, let's say we want to match all numbers that are valid in the USA. For this, we can type down the following pattern.

NXXNXXXXXX

This pattern can be interpreted as follows. The area code (first three digits), **NXX**, means that the first digit can be any number from 1 to 9, and the second and third digits are any number from 0 to 9. Then, the telephone prefix (middle three digits) acts the same way with **NXX**. Finally, the Line Number, **XXXX**, makes the last four digits any number from 0000 to 9999.

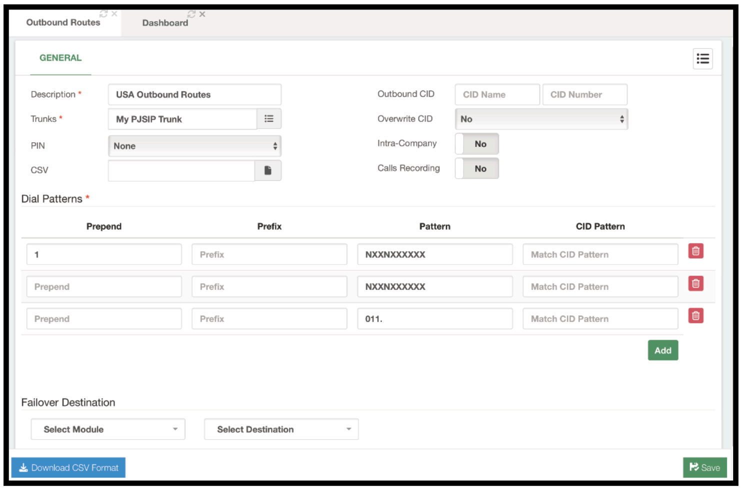
This way, you now support all valid US numbers, with a single dial pattern. Now, sometimes, you need to add the country code at the beginning and the end-user might dial it or not. For this, you can support this by adding the 1 at the beginning of the pattern, i.e. **1NXXNXXXXXX**. As well as adding the 1 on the prepend field so it is added automatically to the pattern without the 1 at the beginning.

Next, let's tackle international numbers. International numbers may vary in length. So here, wild cards are of great use. We can create the following pattern to support international dialing from the USA.

011.

In the USA we use **011** followed by the international number, including the country code, to dial internationally. So, we can use the dot wild card to support any phone number of any length. Whenever an end-user dials **011+International Number**, no matter how long the international number is, the call will be routed correctly.

In the end, the Outbound Route will look like the following image.



Description * USA Outbound Routes

Trunks * My PJSIP Trunk

PIN None

CSV

Outbound CID CID Name CID Number

Overwrite CID No

Intra-Company No

Calls Recording No

Dial Patterns *

Prepend	Prefix	Pattern	CID Pattern
1	Prefix	NXXNXXXXXX	Match CID Pattern
Prepend	Prefix	NXXNXXXXXX	Match CID Pattern
Prepend	Prefix	011.	Match CID Pattern

Failover Destination

Select Module Select Destination

Image 4.4.2 - Outbound route created in the outbound routes module.

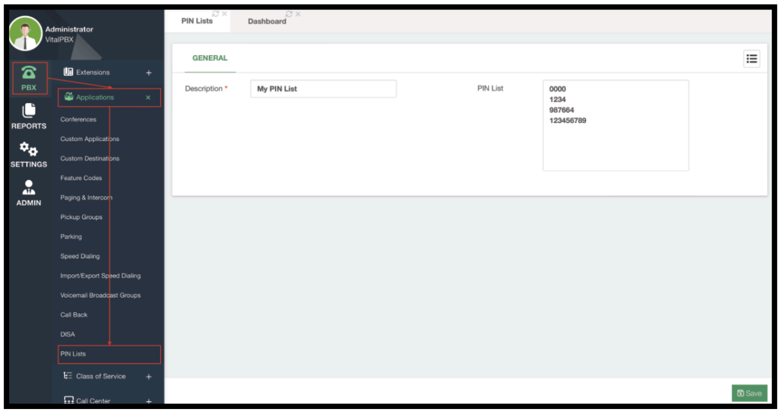
These three specific patterns will allow users in the USA to dial **Locally and Long Distance**, as well as **Internationally**. Number patterns will vary depending on your country or where you are routing the calls, so hopefully these examples help you understand how patterns work so you can create Dial Patterns that match your numbering patterns. Remember to **Save** and **Apply Changes**.

If you have multiple patterns you can import a **CSV file** with the patterns. The CSV Format can be downloaded from the **Download CSV Format** button in the lower left-hand corner. Additionally, you can set a **CID Name and Number** and **Overwrite the Caller ID** at the outbound route level. This way, anytime a call goes through this outbound route, the CID is modified. If you enable the **Intra-Company** option, your extension's **Internal Caller ID** is used instead of the **External Caller ID**. This is useful if the connection is with another PBX system, where you would like to show the *Extension Number* instead of a long number. Finally, you can enable **Call Recording** at the outbound route level. Just remember that if you enable call recording in multiple stages of a call, duplicate recordings will be created.

Section 4, Lesson 5 - PIN Lists

You may have noticed another option for outbound routes called **PIN List**. PIN lists add a level of security and permissions to outgoing calls. Whenever you add a PIN list to an outbound route, you will be prompted to enter a valid PIN to be able to use the outbound route. To create a PIN list, we will go to **PBX > Applications > PIN Lists**.

For a PIN list, we will need to add a **Description** to identify this list and add the **list of PINs** that can be used. PINs can be of any length and are separated by a line break. You can create as many PINs and PIN lists as you need. You can then **Save** and **Apply Changes**.



Description * My PIN List

PIN List
0000
1234
987664
123456789

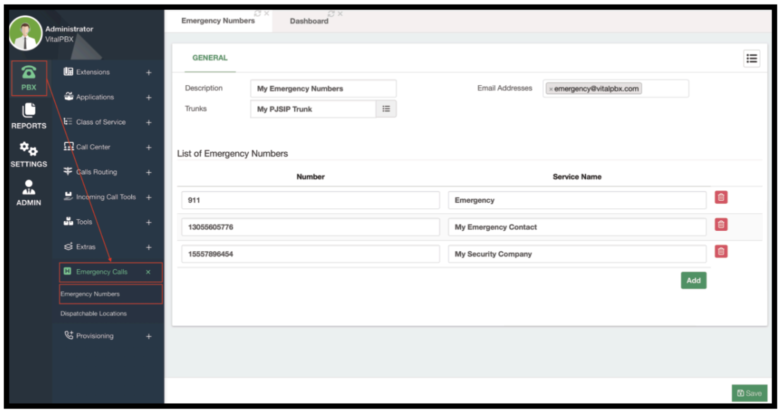
Image 4.5.1 - PIN Lists module.

Afterward, you can go ahead and add the PIN list to your outbound route, so any time a call goes through the outbound route, users will be prompted to enter a PIN from the list to be able to complete the call.

Section 4, Lesson 6 - Emergency Numbers

A necessary feature in every PBX is the handling of **Emergency Numbers**. In VitalPBX we take safety seriously, so we have created a workflow to consider emergency calls in an intuitive manner. So first, we will go to **PBX > Emergency Calls > Emergency Numbers**.

Here, we will first add a **Description** to identify the Emergency number list. Next, we can add as many **Email Addresses** as needed. Whenever we call these numbers, an email is sent to these addresses with the information of where and who placed the emergency call.
The email is based on the email templates we saw previously in our initial configurations. This template can be modified at **Admin > System Settings > Email Templates**.



Description: My Emergency Numbers

Trunks: My PJSIP Trunk

Email Addresses: emergency@vitalpbx.com

List of Emergency Numbers

Number	Service Name
911	Emergency
13055605776	My Emergency Contact
15557896454	My Security Company

Image 4.6.1 - Emergency Numbers module.

We can then select the **Trunk** through which the emergency calls will go through. Emergency calls do not go through outbound routes.

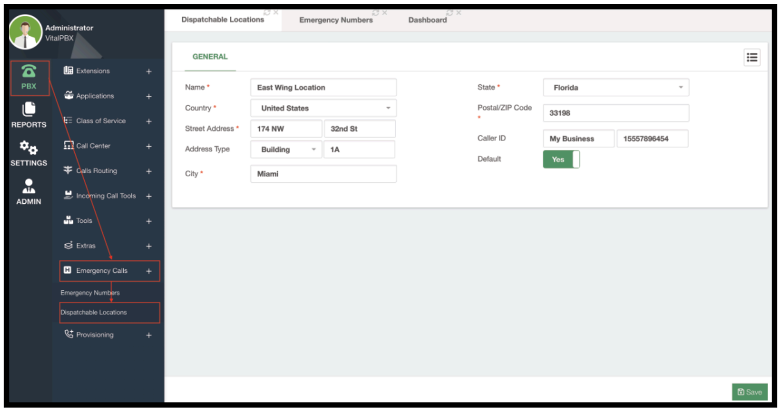
This is so calls are assured to go through no matter if you have a PIN list or the appropriate permissions for an outbound route. You might also use a separate service for emergency calls, so you can set that specific trunk here.

Finally, we have the **List of Emergency Numbers**. Here, we add the **Numbers** and a S**ervice Name** to identify these numbers. We can add as many numbers as we need. Remember to **Save** and **Apply Changes**.

With this, we have established the Emergency Numbers. Emergency calls can be placed from any registered device, including Hot Desking devices with no extension assigned. So no matter what, you can always place an emergency call.

Section 4, Lesson 7 - Dispatchable Locations

The next step for *Emergency Calls* is to establish the **Dispatchable Locations**. For this, we go to **PBX > Emergency Calls > Dispatchable Locations**.



Name * East Wing Location
Country * United States
Street Address * 174 NW 32nd St
Address Type Building 1A
City * Miami
State * Florida
Postal/ZIP Code * 33198
Caller ID My Business 15557896454
Default Yes

Image 4.7.1 - Dispatchable Locations module.

Dispatchable locations allow us to identify the location from where an emergency call is placed. This location can be added to the email sent with the *Emergency Call Template* to the email addresses we added to the *Emergency Numbers*.

For the dispatchable locations, we add a **Name** to identify this location, and enter the **Address** and **Caller ID** information we will use for this location. Usually, when you have an e911 service, you identify a location using special Caller ID information. So, you can associate that CID information with your dispatchable location. You can also set this location as default, so even if you don't specify a dispatchable location for an extension or hot desking device, the default location will be used. You can then **Save** and **Apply Changes**. You can create as many dispatchable locations as you need.

Now, you can assign the dispatchable locations to any extension under the device settings, and to hot desking devices. *Extensions and Hot Desking* devices have their own **Emergency CID information** so the *Caller ID information* is different from any *External Caller ID*.

With this, you now have an *Emergency Call* flow that will ensure that your end-users can place emergency calls without any hassle and are safe to call at any time.

Conclusion

With these configurations, you now have an essential PBX system with the ability to place and receive calls. You can call internally and externally. This is the basis of any PBX system, so congratulations, you essentially have a fully working PBX! Now let's go ahead and make our system more complex to make the most out of it.

SECTION 5 - CLASS OF SERVICE

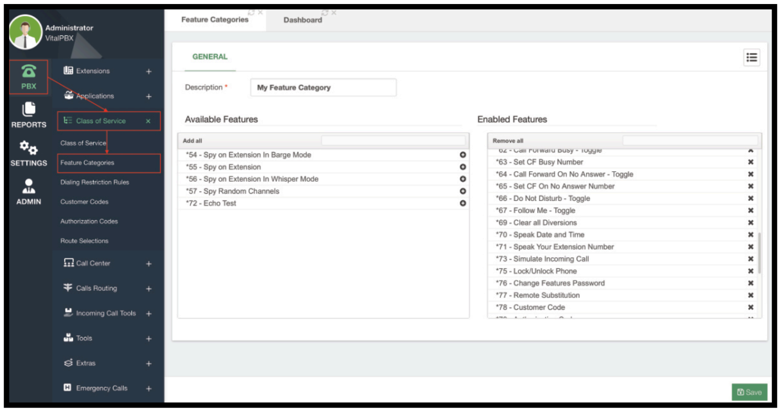
Introduction

Now that you are able to place calls to the outside world with the Trunks and Outbound Routes you have created, you might want to control how the outgoing calls are placed. You might want to restrict outgoing calls to a schedule, or you might want only certain end-users to be able to dial certain numbers. For all of this, you can use what we call a **Class of Service**. Classes of Service are an essential tool for you to have full control of the outgoing calls from your VitalPBX installation.

Classes of Service are composed of three main options Feature Categories, Dial Restriction Rules, and Route Selections. We can configure any of these three features to create our Class of Service.

Section 5, Lesson 1 - Feature Categories

With the **Feature Categories,** we can establish which feature codes extensions or modules can be used with a Class of Service. To create the categories we must go to **PBX > Class of Service > Feature Categories**. This is a very simple module, as all you need is a **Description**, and select which are the **Enabled Features**.



Description * My Feature Category

Available Features

- *54 - Spy on Extension In Barge Mode
- *55 - Spy on Extension
- *56 - Spy on Extension In Whisper Mode
- *57 - Spy Random Channels
- *72 - Echo Test

Enabled Features

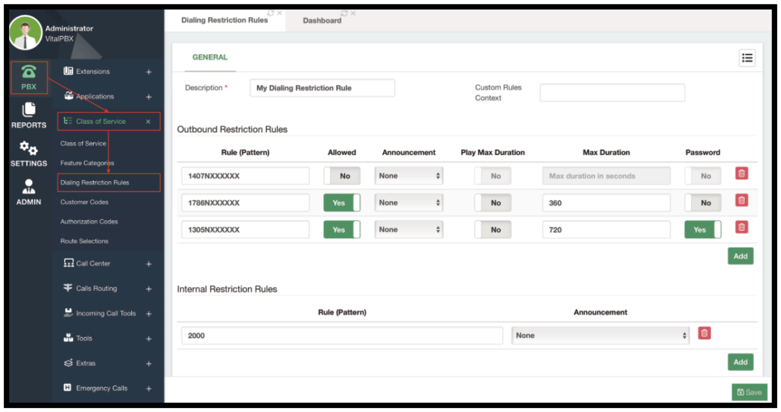
- *62 - Call Forward Busy - Toggle
- *63 - Set CF Busy Number
- *64 - Call Forward On No Answer - Toggle
- *65 - Set CF On No Answer Number
- *66 - Do Not Disturb - Toggle
- *67 - Follow Me - Toggle
- *69 - Clear all Diversions
- *70 - Speak Date and Time
- *71 - Speak Your Extension Number
- *73 - Simulate Incoming Call
- *75 - Lock/Unlock Phone
- *76 - Change Features Password
- *77 - Remote Substitution
- *78 - Customer Code

Image 5.1.1 - Feature Categories module.

Everything in the left column is what you have available, and everything on the right is what is enabled. You can create as many categories as you need. You can click on **Add or Remove All** to move everything in a column to the other side. There is also a search bar to search for specific features. Then, remember to **Save** and **Apply Changes**.

Section 5, Lesson 2 - Dialing Restriction Rules

We can now establish **Dialing Restriction Rules** we can use with a Class of Service. As the name implies, with these rules we can restrict the numbers that users with these rules applied with a Class of Service can dial. To configure these rules we must go to **PBX > Class of Service > Dialing Restriction Rules**.



Description * My Dialing Restriction Rule

Custom Rules Context

Outbound Restriction Rules

Rule (Pattern)	Allowed	Announcement	Play Max Duration	Max Duration	Password
1407NXXXXXX	No	None	No	Max duration in seconds	No
1786NXXXXXX	Yes	None	No	360	No
1305NXXXXXX	Yes	None	No	720	Yes

Internal Restriction Rules

Rule (Pattern)	Announcement
2000	None

Image 5.2.1 - Dialing Restriction Rules module.

Dialing restriction rules can easily become very powerful and specific. First, you must enter a **Description** to identify the rule. Next, we have the **Outbound Restriction Rules**. These are rules for all calls going through a *Trunk*. To create the rule you can enter a **Pattern** and you can use the same variables as the ones we saw in the *Outbound Routes* lesson. Then, you can select if it is **Allowed** to dial the pattern.
Optionally, you can play back an **Announcement** when someone dials this pattern. This announcement is a custom audio you upload. We will see more on how you can do that later. You can enter a **Max Duration** in seconds for how long someone can call the dialed pattern, and whether or not you **Play the Max Duration Time**.

Finally, you can choose if the extension's Feature **Password** is needed to be able to dial the rule's pattern. You are able to add as many rules as you need to make the rule as specific as you need.

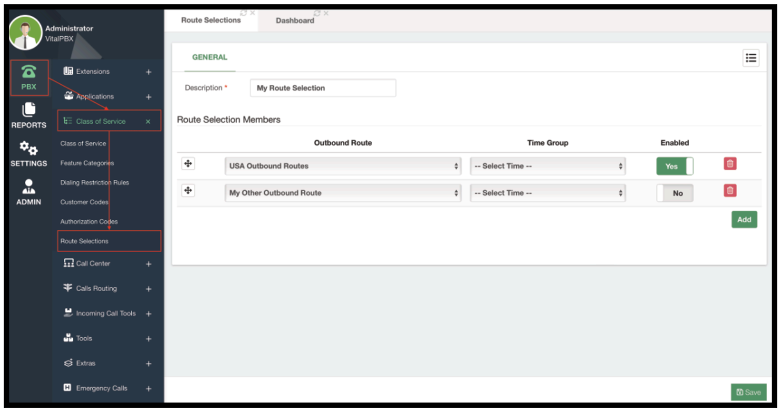
At the bottom, you get the **Internal Restriction Rules**. Here, you can add restrictions to internal numbering, like *extension numbers*, *custom applications*, or *any code* in the *internal numbering plan*. You can play an **Announcement** whenever someone with this rule in their Class of Service dials the restricted pattern. You can then **Save** and **Apply Changes**.

Additionally, if you understand Asterisk contexts, and have created a custom context that can help you with advanced dialing restrictions, you can declare the context in the **Custom Rules Context** field.

And with this, you now have a set of rules to control where anyone with these rules in the Class of Service is able to dial. You can restrict international calling or maybe long distance. Maybe you want to restrict the ability to dial the Boss extension completely. Maybe you have a Hotel that allows calling the front desk, but not other rooms. Or you might want to separate the Call Center agent and the management extensions so they cannot call each other. The rules you create will vary depending on how to wish to control the flow of the calls.

Section 5, Lesson 3 - Route Selection

Next, let’s take a look at **Route Selections**. Route selections allow you to prioritize outbound routes based on time and selection. To create your route selections we must go to **PBX > Class of Service > Route Selections.**



Description * My Route Selection

Route Selection Members

Outbound Route	Time Group	Enabled
USA Outbound Routes	-- Select Time --	Yes
My Other Outbound Route	-- Select Time --	No

Image 5.3.1 - Route Selections module.

This module is a very straightforward one. First, you must enter a **Description** to identify the route selection. Next, we have the **Route Selection Members**. These are the outbound routes you want to associate with this route selection, by selecting the **Outbound Route** from the dropdown menu. Next, you can select a **Time Group** to establish a schedule of when you are going to use this route. Finally, if you want to **Enable** or not this route for the route selection.

You can then click on **Save** and **Apply Changes**.

We will see more about Time Groups in the next lesson, but this helps you see how to select the specific routes for a Class of Service. You can create as many route selections as you need.

Section 5, Lesson 4 - LCR (Least Cost Routing)

Let’s make a small tangent that is greatly associated with route selections. This is the concept of **Least-Cost routing**.

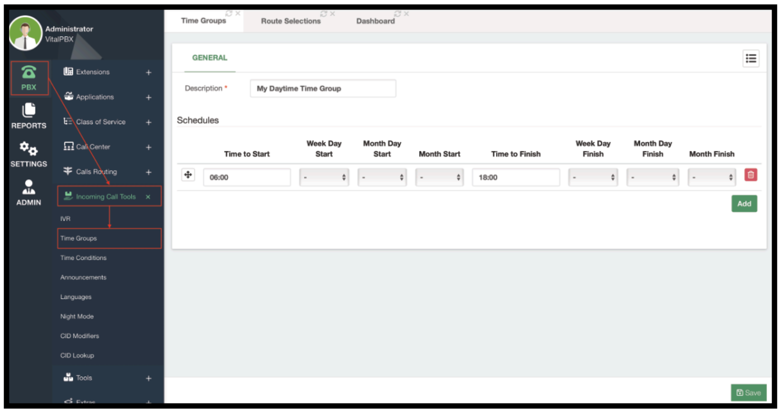
Imagine the following scenario, you have two VoIP Providers. One gives you excellent calling rates during the daytime, and the other gives you a great deal for calls during the nighttime. Obviously, as a business, you want to save the most amount of money.

So, you can create a **Route Selection** that utilizes **Time Groups** to schedule the time and days that your calls go through specific trunks.

Here, we are assuming that you have already created your trunks and outbound routes with the appropriate dial patterns.

So now, let's go to **PBX > Incoming Call Tools > Time Groups**. Here, we are going to create schedules that will match the time when it is most beneficial for us to go through our outbound routes.

First, let's create our daytime time group. This is a simple **Time Group**, but we will see more complex ones later. For now, we establish the Time to Start as *06:00 hours* and the Time to Finish as *18:00 hours*. What this does is it makes this Time Group **TRUE** when the time matches in this timeframe. So, any time of the day between **06:00** in the morning and **18:00** in the evening will match this Time Group. Remember to **Save** and **Apply Changes**.



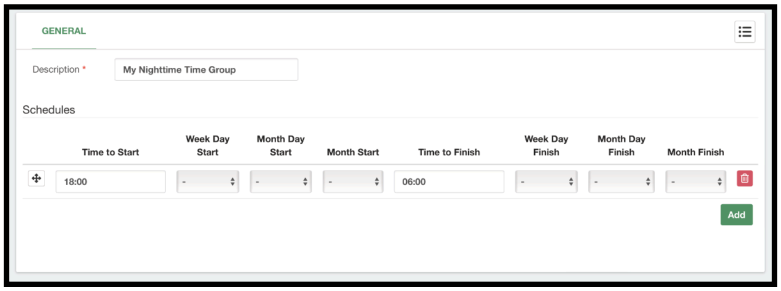
Description * My Daytime Time Group

Schedules

Time to Start	Week Day Start	Month Day Start	Month Start	Time to Finish	Week Day Finish	Month Day Finish	Month Finish
06:00	-	-	-	18:00	-	-	-

Image 5.4.1 - Daytime Time Group example.

For the Nighttime Time Group, it is similar.



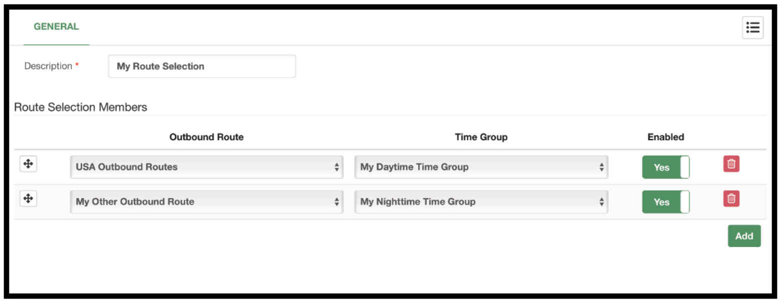
Description * My Nighttime Time Group

Schedules

Time to Start	Week Day Start	Month Day Start	Month Start	Time to Finish	Week Day Finish	Month Day Finish	Month Finish
18:00	-	-	-	06:00	-	-	-

Image 5.4.2 - Nighttime Time Group example.

We make the time group's **Time to Start** at **18:00** hours and the **Time to Finish** at **06:00** hours. **Save** and **Apply Changes**.



Description * My Route Selection

Route Selection Members

Outbound Route	Time Group	Enabled
USA Outbound Routes	My Daytime Time Group	Yes
My Other Outbound Route	My Nighttime Time Group	Yes

Image 5.4.3 - Route Selection with Time Groups.

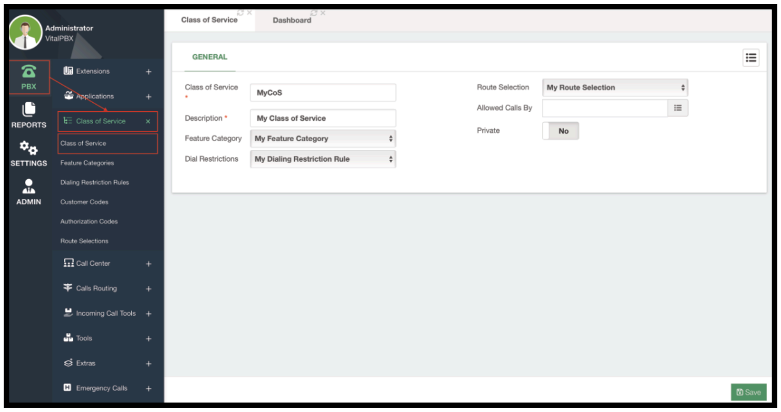
Now, you just need to create a **Route Selection** with both of your outbound routes and select the **Time Groups** accordingly.

Don't forget to **Save** and **Apply Changes**. With this, you now have a route selection you can use with a Class of Service for a least-cost routing environment. This means that whenever the time matches, you will be using the outbound route and trunk that gives you the best rate at all times.

Time Groups can be more complex, as you can specify the days of the week, for example, during weekdays you will use one of the outbound routes and the other during the weekend. We will see more complex Time Groups later on, but for now, this example gives you a clear idea of how to do Least Cost Routing with VitalPBX.

Section 5, Lesson 5 - Class of Service

Now that you have a Feature Category, a Dialing Restriction Rule, and a Route Selection, we can apply these to a **Class of Service** to make use of them. To create the Class of Service, you must go to **PBX > Class of Service > Class of Service**.



Class of Service * MyCoS
Description * My Class of Service
Feature Category My Feature Category
Dial Restrictions My Dialing Restriction Rule
Route Selection My Route Selection
Allowed Calls By
Private No

Image 5.5.1 - Class of Service module.

To create a Class of Service, you must first enter a **Class of Service name**. This name cannot contain spaces, but dashes and underscores are allowed. Next, enter a **Description** for easier search-ability.

Afterward, it is just a matter of selecting the **Feature Category**, **Dial Restriction**, and **Route Selection** you have created previously. Notice that it is not completely necessary to add the three of these to a Class of Service, as you also have the option to allow **All Feature Categories**, **No Dialing Restrictions**, and **All Routes**. In fact, the default **All Permissions** Class of Service or CoS has these unrestrictive permissions.

The last thing you can enable in a Class of Service is the **Private** option. With the Private option, you can restrict internal communication to only extensions and modules that use this Class of Service. So only if you have this Class of Service assigned, you will be able to dial other extensions or modules that use this same Class of Service. You can white-list other extensions or modules by adding other Classes of Service to the **Allowed Calls By** field.

So any extension or module that has a Class of Service declared in the allowed calls by field will be able to dial extensions or module that has this Class of Service. Remember to **Save** and **Apply Changes**.

You can now use this Class of Service in any extension or module that requires one. If you add this Class of Service now to a registered extension, you will notice that the rules we have applied will be effective.

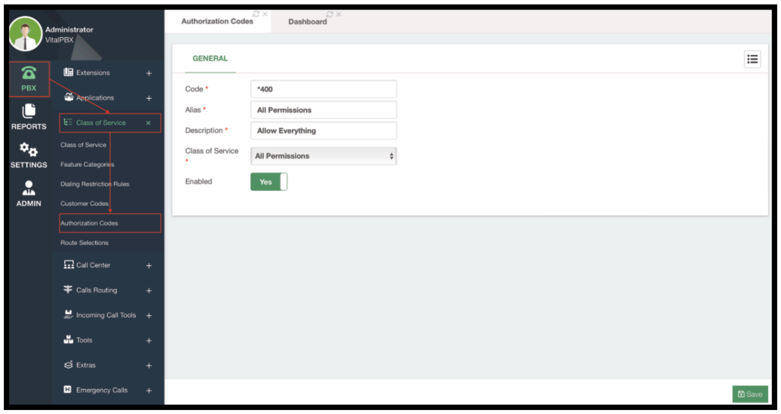
Section 5, Lesson 6 - Authorization Codes

Associated with Classes of Service are **Authorization Codes**. These codes allow you to change the Class of Service for a single call. Let's say that you are a higher-up who usually has no restrictions on your calls, but you are roaming by general public extensions that have restrictions.

You can use authorization codes to change the class of service to a more permissive one. To create the authorization code, go to **PBX > Class of Service > Authorization Codes**.

First, enter what the **Code** will be. This can be comprised of numbers and the star (*) symbol.

Next, add an **Alias**, which can be anything of your liking. The alias is used for security reasons, where instead of the security code the alias will appear in the CDR (Call Detail Records.) Afterward, add a **Description** to easily search for the authorization code, select the **Class of Service** this code will use, and **Enable** it. **Save** and **Apply Changes**.



Code * : *400
Alias * : All Permissions
Description * : Allow Everything
Class of Service * : All Permissions
Enabled : Yes

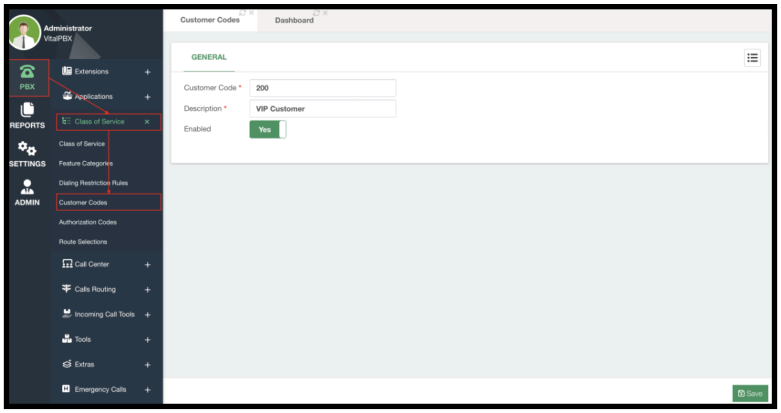
Image 5.6.1 - Authorization Codes module.

To use the Authorization Code, from an extension that has a restrictive Class of Service applied dial the code ***79**. This will prompt you to enter the authorization code. Enter the code we have created. Now dial the number you wish to call, and you will notice that it will allow you based on the authorization code's Class of Service.

Section 5, Lesson 7 - Customer Codes

You might have noticed there is an additional module in the Class of Service section. This is the **Customer Codes** module. This is a simple module that is not really related to Classes of Service but to outbound calling as a whole.

Customer codes can be used to identify outgoing calls in the CDR reports. To create a customer code, you must go to **PBX > Class of Service > Customer Codes**.



Customer Code * 200

Description * VIP Customer

Enabled Yes

Image 5.7.1 - Customer Codes module.

The module is self-explanatory. You just need to enter the **Customer Code** you are going to use, a **Description** to identify the customer code, and whether or not the code is **Enabled**. **Save** and **Apply Changes**.

To use the customer code dial from a registered extension the code ***78**. This will prompt you to enter the customer code. Enter the customer code you just created. Next, you will be prompted to dial the number you wish to call. Once the call is finished, this will add the Customer Code to the CDR so you can easily search for these calls. We will see more about the CDR later, but for now, this is how you can use Customer codes.

Conclusion

This will cover the Class of Service section. VitalPBX is proud to come from a Telephony background, so we deemed this to be an essential feature for any PBX system. Classes of Service allow powerful ways for you to control the flow of outgoing calls from your VitalPBX, so make use of them to make sure you have full control of your calls.

SECTION 6 - CALL CENTER

Introduction

One of the industries that VitalPBX excels at is the Call Center industry. We possess various tools that can aid with your Call Center needs, so let's take a look at some tools that come in the VitalPBX WebUI and some add-ons that can enhance the experience.

Section 6, Lesson 1 - Call Center as a Concept

For us to begin talking about the different features VitalPBX has for Call Center environments, let's take a look into what Call Centers are and how they usually operate so we all have a clear idea of how we can use these tools to make the most out of our VitalPBX installation.

First, there are mainly two types of Call Centers, Inbound and Outbound.

Inbound Call Centers have key points for their operation, among these are the following.

- **ACD or Automatic Call Distribution -** This is a system that automatically assigns incoming calls to available agents.
- **Receive Calls -** Inbound Call Centers will only receive calls. Agents do not place outgoing calls to customers. Customers will be calling the call center.
- **Calls are terminal -** This means that the reason for the customer calling must be resolved by the agent or the appropriate party.

In inbound call centers the agents must be able to assess the inquiry the caller or customer has placed. Be it positively or negatively, when a customer calls into the service, the assessment must be finished in a single call. Calls that bounce in an inbound call center are usually seen in a bad view.

Outbound Call Centers, on the other hand, manage these key points.

- **Place Calls -** Outbound Call Centers will be the ones calling the customers. Usually through a list of contacts.
- **Are Initial Calls -** Since the customer is not the one initiating the conversation, it is possible for the assessment to take multiple calls or re-schedule calls for a later date.
- **Use a Dialer -** Instead of an ACD, outbound call centers use a Dialer. This can be a predictive or a progressive dialer that automatically calls the contact list and assigns the call to an agent.

Outbound call centers will use either a progressive or predictive dialer system. Progressive dialers are more elegant in our opinion as they will call the customer only when the agent is available. Predictive dialers are better when you have a massive team of agents assuring that an agent is available at any time. In predictive dialers, if an agent is not available it can cause situations where the customer is called and they listen to ambiance noise or music on hold instead of a person, causing more disconnections.

In this guide, we will be looking into VitalPBX's inbound call center capabilities. For Outbound, we have our Sonata Dialer application, but to learn more about this you can refer to the Sonata Dialer manual.

With this known, let's dwell more into ACD and how it works, calls are managed with the following key points.

- ACD assigns the incoming calls by incoming order to the first available agent.
- The system will answer the call immediately and, if needed, keep the calls in queue until they can be answered by an available agent.
- The balance of the workforce between agents guarantees that each caller receives a quick response and professional service.

Here are some general rules of thumb you can consider when developing your Call Center environment.

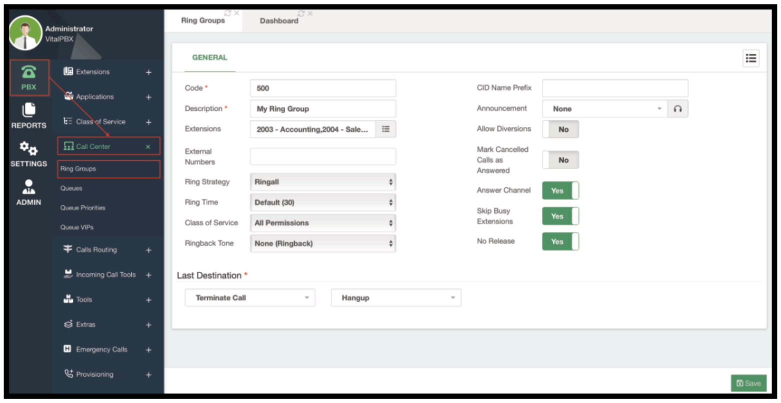
- If you have more channels, meaning people calling in, than agents, use a queue.
- If you have more agents than people calling in, use a Ring Group.
- Queues are more efficient than Ring Groups since you can have a call on hold for as long as you configure it. Queues are unlimited by default in VitalPBX.
- In some rare cases, Ring Groups will work better than Queues for some Call Centers. Especially if you are not expecting that many calls.
- Callers can receive information about wait time and position in a queue.
- Queues can have static or dynamic agents, while users in a ring group will always be part of a ring group.
- Multiple Ring Strategies are available.

Regarding Static and Dynamic agents, the difference is straightforward. Static agents a logged in a queue indefinitely, while dynamic agents can log in or out at any moment. Both types of agents can pause within a queue at any moment.

With these concepts clear, you might get a clearer idea of how a Call Center operates. As mentioned previously, VitalPBX has multiple tools that can aid in any of the aspects we went through. We have even more tools that expand on these concepts, and we will go through these in this guide.

Section 6, Lesson 2 - Ring Groups

As mentioned in the previous lesson, there are two ways you can distribute calls to a group of people. This can be with a Queue or a Ring Group. The most simple of these is the **Ring Group**. With Ring Groups you can easily call multiple extensions at once. To start creating our ring group, we must go to **PBX > Call Center > Ring Groups**.



Code * 500
Description * My Ring Group
Extensions 2003 - Accounting,2004 - Sale...
External Numbers
Ring Strategy Ringall
Ring Time Default (30)
Class of Service All Permissions
Ringback Tone None (Ringback)
CID Name Prefix
Announcement None
Allow Diversions No
Mark Cancelled Calls as Answered No
Answer Channel Yes
Skip Busy Extensions Yes
No Release Yes
Last Destination *
Terminate Call
Hangup

Image 6.2.1 - Ring Groups module.

The first thing you will enter is a **Code** for the ring group. This is the number you will need to dial to reach this ring group. Then, add a **Description** so we can identify the ring group.

Afterward, we must include the **Extensions** that belong to the ring group. These are the extensions that will be called when calling the ring group.

Optionally, members of this ring group can also be an **External Number**. Keep in mind, when using external numbers, you need to choose an appropriate **Class of Service** so the module can reach the external number.

Next, choose the Ring Strategy that best fits your needs, be it **Ring All** or **One by one**.

There are a couple of options that can help the members of a ring group to identify incoming calls. **CID Name Prefix** will add anything you enter in this field to the caller's Caller ID Name. This way, you get to see that the call is from this ring group. You can also add an **Announcement**, which will playback to the ring group member before connecting the incoming call.

Lastly, let's add a **Last Destination**, which is the destination for the call when the **Ring Time** runs out and none of the members answers the call. With this, you can **Save** and **Apply Changes**.

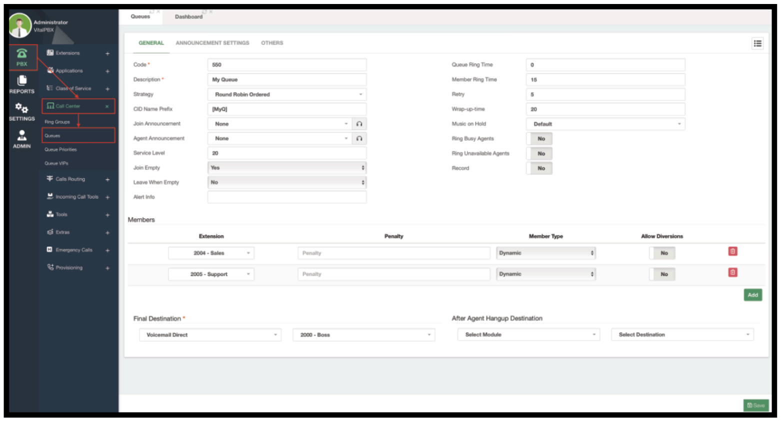
Some additional configurations you can perform in this module are as follows.

- **Ringback Tone -** This is a Music on Hold class to playback while someone is calling the ring group.
- **Allow Diversions -** With this feature enabled, the ring group members can enable any call diversions, i.e. call forwards, which will affect the incoming calls to the ring group.
- **Mark Cancelled Calls as Answered -** This will mark the calls as Answered if one of the ring group members answers the calls. With this disabled, calls will appear as missed in the extensions for the members who did not answer the call.
- **Answer Channel -** This is the default behavior for a ring group, as the ring group module will answer the call, and then connect it with any member that answers their phone. With this feature turned off, calls will not be marked as answered until a member answers the call. Keep in mind, that if you turn it off, audio (i.e. Music on Hold) may not playback, as the call is still ringing the system.
- **Skip Busy Extensions -** If this is turned on, any extension with a busy status will be skipped.
- **No Release -** This adds the no-release (/n) flag to the ring group members. This avoids creating multiple items in the CDR. Nevertheless, some phones may present issues on transfers when this feature is enabled, therefore it can be disabled.

And with this, you now have a Ring Group created. You can dial the ring group code from any device that does not belong to the ring group and members will start to ring. As you can see this is a very straightforward module that easily allows you to dial multiple extensions.

Section 6, Lesson 3 - Queues

Now, let’s take a look into **Queues**. Queues are the basis for all inbound Call Centers. These will allow you to manage multiple calls and automatically assign them to available agents. queues in VitalPBX possess multiple tools that will allow you to manage incoming calls easily and effectively. To start creating queues we must go to **PBX > Call Center > Queues**.



Code *	550	Queue Ring Time	0
Description *	My Queue	Member Ring Time	15
Strategy	Round Robin Ordered	Retry	5
CID Name Prefix	[MyQ]	Wrap-up-time	20
Join Announcement	None	Music on Hold	Default
Agent Announcement	None	Ring Busy Agents	No
Service Level	20	Ring Unavailable Agents	No
Join Empty	Yes	Record	No
Leave When Empty	No		
Alert Info			

Members

Extension	Penalty	Member Type	Allow Diversions
2004 - Sales	Penalty	Dynamic	No
2005 - Support	Penalty	Dynamic	No

Final Destination *		After Agent Hangup Destination	
Voicemail Direct	2000 - Boss	Select Module	Select Destination

Image 6.3.1 - Queues module General tab.

The Queues module is comprised of multiple tabs, the **General**, **Announcement Settings**, and **Others** tabs.

We will begin with the main configurations in the General tab. This contains the general configuration for our queues and here is where we add our **Members** also known as **Agents** in a queue.

First, we will add a **Code** to our queue. This is how we are able to call a queue internally. And we add a **Description** so we can identify it easily.

Next, we have our **Ring Strategies**. We must choose one of the following.

- **Ring All -** This will ring all the available agents at once. Note: that this is not a recommended strategy if you want to have accurate statistics for your agent reports.
- **Least Recent -** This will ring the agent with the least recent assigned call.
- **Fewest Calls -** This will ring the agent with the fewest calls assigned.
- **Random -** This strategy will ring any agent at random.
- **Round Robin Memory -** Calls will be assigned in order of login, and the queue will remember which was the last agent assigned in the order.
- **Linear -** This will ring the agents linearly in order of login. This strategy will always begin with the first agent and only go to the next one if the agent is busy.
- **Weight Random -** Will randomly ring an agent, based on their penalty.
- **Round Robin Ordered -** Will ring an agent in order of login. Will continue to the next agent based on the previous agent rung.

It is very important you choose the ring strategy appropriately. This way calls are distributed as evenly as possible according to your operations. We recommend using any of the Round Robin options as these are the most fair among the agents.

Options like Least Recent and Fewest Calls may overwhelm a single agent with calls until they reach the number of calls answered by other agents. This is evident especially if you have mixed shifts where a new agent logs in and does not have the same amount of calls answered as the other available agents. With Linear, the first agent can get overwhelmed if they complete calls often, and all new calls will always be assigned to them. Linear is only recommended if you know that calls will last longer.

Afterward, you can add a **CID Name Prefix**. This will help the agent identify the queue, especially if they belong to multiple queues. Similarly, you can add an **Agent Announcement**, which will playback to the agent prior to connecting the call with the caller.

A **Join Announcement** is an audio that will playback to the caller when joining the queue. This is usually a welcome or disclaimer message that tells the caller the queue they have called into or that the call will be recorded for quality assurance and training purposes.

Next, we configure the **Queue and Member Ring Time**. Queue Ring Time is the time in seconds a caller will stay in the queue before going to the Final Destination, while the Member Ring Time is the time in seconds the call rings an agent before going to the next available agent respecting the ring strategy. If the Queue Ring Time is set to zero, the caller will wait indefinitely in queue until an agent answers the call. In between ringing members, the **Retry** time in seconds will be considered.

Then, we have the **Wrap-Up Time**. This is the time agents will have between getting calls assigned. During this time, agents will not get calls assigned allowing them some breathing time, and to wrap up any process from their previous call.

Afterward, we can start adding members to the Queue. Members have four aspects to configure.

- **Extension -** This is the user we will add as a member to our queue.
- **Penalty -** This is how we prioritize agents within a queue. Agents with a lower penalty will be prioritized when assigning calls. Only when these agents are busy or unavailable, agents with higher penalty will be considered.
- **Member Type -** Agents can be either Static or Dynamic members. Static agents will be permanently logged into a Queue, while Dynamic agents can log in or out of a Queue. By default, agents will always be Dynamic. We recommend static agents when you prioritize a device or cubicle assigned to a queue, rather than considering individual agents. Dynamic agents will give more accurate statistical reports.
- **Allow Diversions -** This will allow agents to enable call diversions like Do Not Disturb and Call Forwarding.

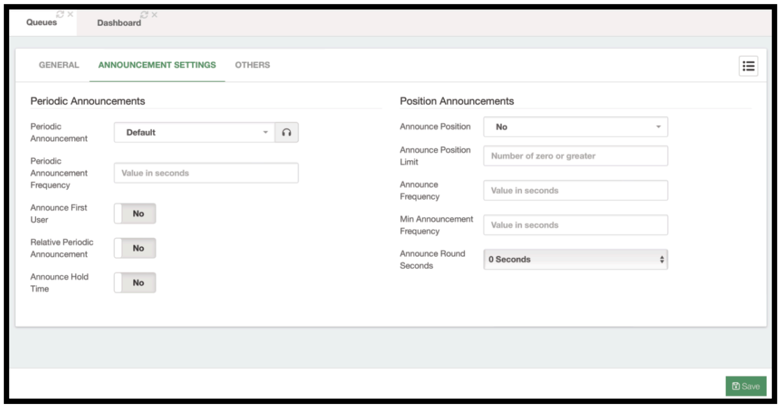
You can add as many agents as you need to a queue, and remove them at any time.

Lastly, we have the **Final Destination** where the call will be sent if the members/agents do not answer the call in time. And the **Destination After the Agent Hangs Up**. If the agent hangs up first, you can send the call to another destination. This can be something like a satisfaction survey with a specific extension or an IVR tree using our IVR Stats commercial add-on. We will see more about IVR stats later.

With these configurations, you can **Save** and **Apply Changes**, and have a queue ready.

Additionally, you can configure the following.

- **Service Level -** This is used for statistic reports. This is the amount of time it takes the agents to answer a call in the queue.
- **Join Empty -** Allows the callers to join the queue when there are no agents. The available options are:
    - **Yes -** If set to yes, this allows the callers to join the queue without agents.
    - **No -** If set to no, callers cannot join the queues without agents.
    - **Strict -** Callers cannot join a queue without agents or only unavailable agents. Unavailable Agents mean that the agents are paused or busy.
    - **Loose -** Callers can join a queue without agents and the agents are unavailable.
    - **No + InUse -** Callers cannot join a queue without agents or agents that are busy on a call.
- **Leave Empty -** This will kick callers out of the queue when the last agent leaves the queue and the caller is still waiting in the queue. The available options are:
    - **Yes -** When set to yes, callers will be kicked out of the queue when the last available agent leaves.
    - **No -** When set to no, callers will not be kicked out of the queue when the last available agent leaves.
    - **Strict -** This will kick the callers from the queue when the last agent leaves or becomes unavailable. An unavailable agent means that the agent is on a call or on pause.
    - **Loose -** This is the same as strict, but paused queue agents will not be considered as unavailable.
    - **Yes + InUse -** This will kick the caller from the queue when the last available agent leaves or gets on a call.
- **Alert Info -** This will add to the alert-info header so agent devices can have a different ringtone.
- **Music on Hold -** This is the Music on Hold class that plays back while the caller is in the queue.
- **Ring Busy Agents -** If enabled, this will ring agents even if they are on a call.
- **Ring Unavailable Agents -** If enabled, this will ring agents even if they are not registered on the PBX.
- **Record -** If enabled, this will record all the calls in the queue. Remember, if you enable call recording in multiple stages of a call, be it in a trunk, inbound route, or at an extension level, for example, multiple instances of the call will be recorded.



Periodic Announcements

Periodic Announcement: Default

Periodic Announcement Frequency: Value in seconds

Announce First User: No

Relative Periodic Announcement: No

Announce Hold Time: No

Position Announcements

Announce Position: No

Announce Position Limit: Number of zero or greater

Announce Frequency: Value in seconds

Min Announcement Frequency: Value in seconds

Announce Round Seconds: 0 Seconds

Image 6.3.2 - Queues’ Announcement Settings.

Next, we have the **Announcement Settings** we can add to our queue.

Here we can configure our **Periodic Announcements**. This includes a **Periodic Announcement** we can playback based on the **Periodic Announcement Frequency**. This is an audio that usually tells the caller that they are still in the queue and will be connected to the next available agent. We will see later how to upload these sound files.

Additionally, we have the following options.

- **Announce First User -** This option will tell the caller that they are the next in line.
- **Relative Periodic Announcement -** This option considers the length of the Periodic Announcement, so the frequency is considered at the end of the periodic announcement's length.
- **Announce Hold Time -** This will estimate the hold time based on previous calls, and announce the time to the caller.

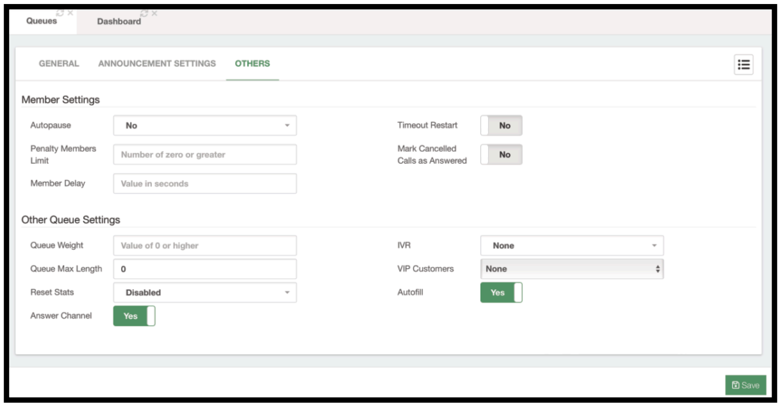
Then we have the **Position Announcements**, which include the following options.

- **Announce Position -** This lets you choose if you want to announce the position to the caller. The available options are:
  - **No -** If the announce position is set to no, the queue position will not be announced to the caller.
  - **Yes -** if set to yes, the queue position will be announced to the caller.
  - **Limit -** if set to limit, the Announce Position Limit will be considered. The position will be limited to the Announce Position Limit value. This means that if the position of the caller exceeds the Announce Position Limit the queue position will not be announced to the caller.
  - **More -** if set to more, this means that the position will be announced to the caller if the Announce Position Limit is exceeded.
- **Announce Frequency -** This is how often the position and hold time are announced to the caller. In smaller Call Centers, it is more difficult to estimate the hold time for callers. So this can be a frustrating feature in these smaller environments. By setting it to 0 the feature is disabled.
- **Minimum Announcement Frequency -** This defines the minimum amount of time that must elapse before announcing the position or holding time again. This is used to prevent these announcements from playing too often.
- **Announce Round Seconds -** This is used to round the holding time if the time is non-zero. You can choose in intervals of 5 seconds between 0 and 50 seconds.

Remember to **Save** and **Apply Changes** if you made any changes in this section.

> ***Note:*** *The Announce First User, Hold time, and Position can become a double-edged sword. We will see in a moment where you can add priority to other calls based on the queue or caller. So the estimations in holding time and queue position may change, and a caller can go from the first position to the second or their hold time may increase. This can frustrate the caller, so we recommend you use these features at your discretion.*

Finally, we have the **Others** tab for additional settings.



Member Settings

Autopause: No
Penalty Members Limit: Number of zero or greater
Member Delay: Value in seconds
Timeout Restart: No
Mark Cancelled Calls as Answered: No

Other Queue Settings

Queue Weight: Value of 0 or higher
Queue Max Length: 0
Reset Stats: Disabled
Answer Channel: Yes
IVR: None
VIP Customers: None
Autofill: Yes

6.3.3 - Queues' Others tab.

Here, we find the following **Member Settings**.

- **Auto pause -** This will pause the agents automatically in a queue if they do not answer multiple calls in a row. You can choose between the following options.
  - **Yes -** Will pause the agent in this specific queue.
  - **No -** Will not pause the agent even if they don't answer incoming calls.
  - **All -** This will pause the agent in all the queues they belong to.
- **Penalty Members Limit -** This option disregards the consideration of the agents' penalty if there are too few members logged in the queue. No penalty is considered if there are X or fewer agents in the queue.
- **Member Delay -** This is a delay in seconds you can add before connecting the caller with the agent.
- **Timeout Restart -** Resets the timeout for an agent to answer if a BUSY or CONGESTION status is received from the channel. This is useful if the agent is allowed to reject or cancel a call.
- **Mark Cancelled Calls as Answered -** This is useful if you are using the Ring All strategy. This will mark calls that have been answered by one of the ringed members as answered. Whereas if this is disabled, calls would be marked as missed.

Finally, we have **Other Queue Settings** that allow us the following.

- **Queue Weight -** If the agents belong to multiple queues, the queue weight gives priority to calls from this queue over the others.
- **Queue Max Length -** This is the number of callers that can be in a queue. A value of 0 means that there can be any number of callers waiting in the queue.
- **Reset Stats -** For this feature, you need to create a cron job. We will learn more about cron jobs later. This feature allows you to reset the current statistics for a queue.
- **Answer Channel -** This feature is enabled by default. This allows the queue to answer the incoming call. If it is disabled, calls will not be answered until an agent is available. Notice that if this feature is disabled, no announcement or music on hold will play.
- **IVR -** This allows you to set an IVR while the caller is waiting in the queue.
- **VIP customers -** This is a list of customers that have a higher priority over other callers for this queue.

- **Autofill -** With this feature enabled, the queue behaves more like a real queue. If there are multiple agents available, multiple calls are assigned to those agents at the same time.

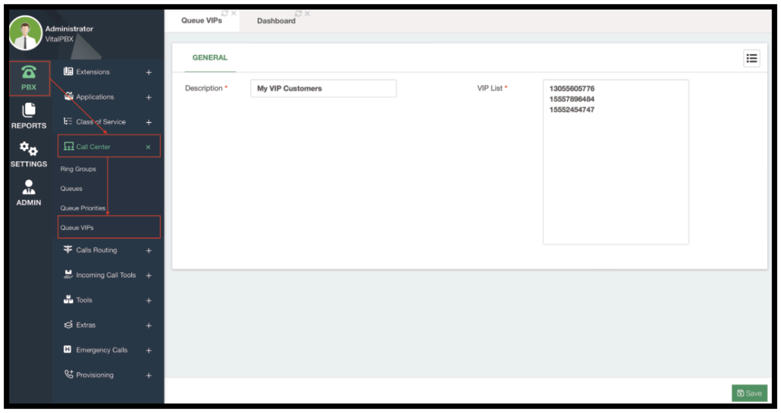
This was the last tab for the Queues module. As you can see this module is very extensive and allows you to make granular changes to your queues. Queues are the basis for any inbound call center and with VitalPBX you can create as many queues with as many agents as you need.

Section 6, Lesson 4 - Queue VIPs

As we saw while creating a queue, we are able to assign **VIP Customers** to give them a higher priority over other callers in a queue. To create this list of VIP customers we must go to **PBX > Call Center > Queue VIPs**.

This module is very simple to follow. All you need to add is a **Description** to identify the list and the list of numbers on the **VIP List**. The VIP List must have the Caller ID numbers for the customers just as you expect to get them from your VoIP Provider. Each number must be separated using a line break.

You can add as many numbers as you need to any number of Queue VIP lists.



Description * My VIP Customers

VIP List *
13055605776
15557896484
15552454747

Image 6.4.1 - Queue VIPs module.

Section 6, Lesson 5 - Agent Codes

In this short lesson, we will outline a cheat sheet of feature codes your agents and call center members need to know for everyday operations.

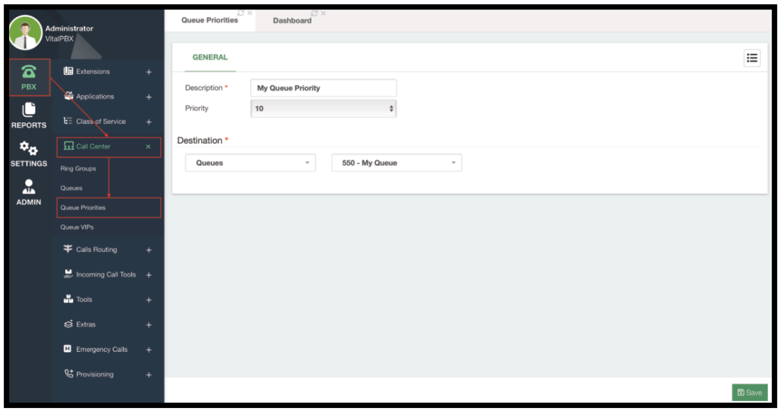
- ***50 -** Log in and Log out from a specific queue.
- ***51 -** Pause and Resume an agent from a specific queue.
- ***52 -** Log in and Log out the agent from all the queues the agent is a member of.
- ***53 -** Pause and Resume an agent from all the queues they are a member of.
- ***54 -** Spy on an extension in Barge Mode. This allows you to listen to listen to a call an agent is on and participate in the conversation.
- ***55 -** Spy on an extension. This allows you to listen to a call without the agent or the callers knowing you are listening.

- ***56 -** Spy on an extension in Whisper Mode. This is normally known as coaching. This feature allows you to listen to a call and be able to speak with your agent without the caller knowing you are in the conversation.
- ***57 -** Spy on a random channel. This will allow you to listen to a call from a random agent.
- ***80 -** This allows you to Hot Desk into a device.
- ***90 -** This code allows you to Hot Desk into a device and log into your queues in a single step. Dial *90#QUEUE_NUMBER#FEATURES_PASSWORD, and this will log you into the queue using your features password.

These are the most common feature codes you will use in everyday operations in a Call Center. Every code has a prompt that will guide you through the process of operation.

Section 6, Lesson 6 - Queue Priorities

We have now seen how we can prioritize callers inside a queue with Queue VIPs, and how you can add weight to a queue so it has higher priority over other queues. You can also add priority to a queue dynamically to a queue using Queue Priorities. To create a Queue Priority you must go to **PBX > Call Center > Queue Priorities**.



Description * My Queue Priority
Priority 10
Destination *
Queues
550 - My Queue

Image 6.6.1 - Queue Priorities module.

The module is simple to use. You must add a **Description** to identify the priority, followed by the **Priority** level and the **Destination**.

The Priority level establishes its priority, the higher the better. And the destination is always a queue. You can then set this Queue Priority as a destination in an IVR or inbound route, and the priority is set.

Section 6, Lesson 7 - Commands for Queues and Agents

As a system administrator, you might want to monitor various queues and agents with their status. Here are some commands you will find useful when monitoring and troubleshooting queues and agents. These commands are run from the Asterisk CLI. This is accessed by initiating an SSH connection to your VitalPBX installation and running the asterisk -rvvvv command.

- **Queue Show -** This command shows you the logged-in agents and their current status in a queue. If no queue is entered, all queues are shown.

```
queue show Q<queue_number>
```

- **Queue Add Member -** This adds a new member to a queue. Using a dial string (such as SIP/6001) to a queue with an optional penalty, member name, and a state interface.

```
queue add member <dial string> to <queue> [[[penalty <penalty>] as <member name>] state_interface <interface>]
```

- **Queue Remove Member -** Remove a specific channel from a queue.

```
queue remove member <channel> from <queue>
```

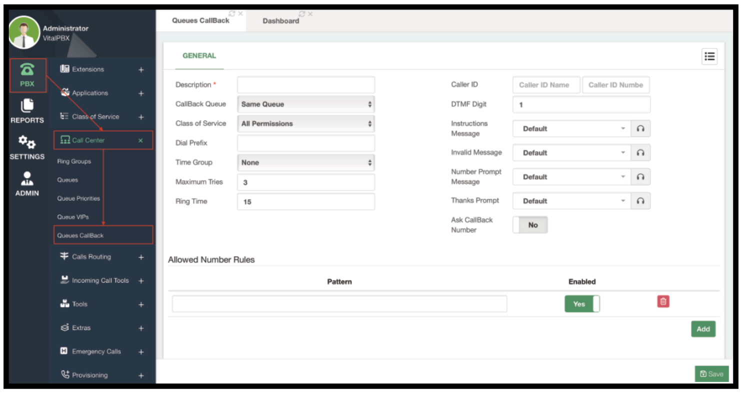
- **Queue Reset Stats -** Resets the gathered statistics from a queue.

```
queue reset stats Q<queue>
```

These commands will help you with troubleshooting and assessment of your queues.

Section 6, Lesson 8 - Queues Callback

We all know that waiting in queue can be a little frustrating, especially if there are long hold times at any given moment. This can lower customer satisfaction if they have to call into a business and wait for long times to be tended to. For this, we have created the **Queues Callback** add-on module. This module allows calling customers to opt-in for a callback rather than waiting in a queue. Once their turn is reached and an agent is available, VitalPBX will call them back automatically.



Description *
CallBack Queue: Same Queue
Class of Service: All Permissions
Dial Prefix
Time Group: None
Maximum Tries: 3
Ring Time: 15
Caller ID: Caller ID Name | Caller ID Numbe
DTMF Digit: 1
Instructions Message: Default
Invalid Message: Default
Number Prompt Message: Default
Thanks Prompt: Default
Ask CallBack Number: No

Allowed Number Rules

Pattern	Enabled
	Yes

Image 6.8.1 - Queues Callback module.

This is a *commercial add-on*, so, to use it you need to install it first under **Admin > Add-Ons > Add-Ons**. Since it is a commercial add-on without a license you can create **one single** Queue Callback with all its features for an unlimited amount of time. Licensing is available **individually** for the add-on or with our **Call Center and Carrier Plus** Licensing plans.

Once you have the add-on installed, you can go to **PBX > Call Center > Queues Callback**. To create a queue callback, we must enter a **Description** to identify this callback. Next, we can select the **Callback Queue**. This is the queue where we are going to send the callback numbers. By default, it is the *same queue*, but you can choose *any other queue*. Remember that every module that places an outgoing call also requires a **Class of Service**, so it has the permissions to reach the numbers to dial.

If we are using a **Dial Prefix** to use a specific Outbound Route, you can add it as well. We will then define the **Maximum Tries** and **Ring Time**. This is how many times VitalPBX is going to try to reach the caller, and for how long it's going to wait for an answer.

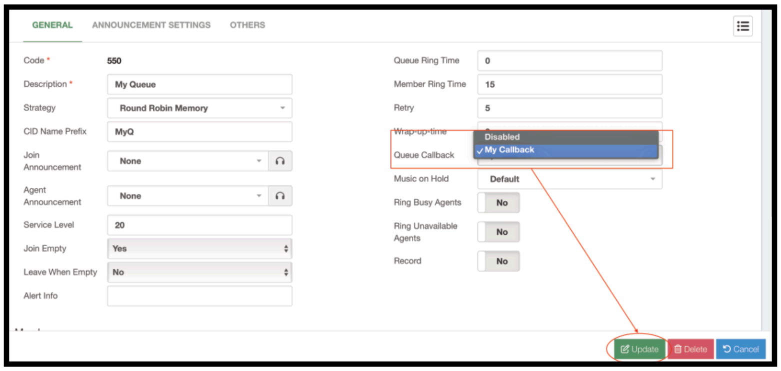
The rest can be left with the default values for the time being. The other important part to configure is the **Ask Callback Number** option. When this is enabled callers will be able to enter the number they wish to be called back to. To prevent any misuse of this option, you can establish the **Allowed Number Rules**. These are dial pattern rules that need to be followed so the caller can enter a valid number. You can establish whether a pattern is enabled or not.

With this, you can **Save** and **Apply Changes**. The additional configurations you can add are as follows.

- **Caller ID -** You can set the Caller ID Name and Number the module will use. This is helpful so the customer knows who is calling them.
- **DTMF Digit -** This is the digit the caller will press to opt for a callback while waiting in a queue. By default it is 1.
- **Instructions Message -** This is a message that replaces the Periodic Announcement in a queue.
- **Invalid Message -** This is a message that will playback when the caller enters a number that is not allowed in the Allowed Number Rules.
- **Number Prompt Message -** This is a message that prompts the caller what digit to press to opt-in for a callback.
- **Thanks Prompt -** This is a message that plays after a caller opts in for a callback.

With the callback created, to use it, you need to assign it to a queue. For this let's go back to **PBX > Call Center > Queues**. You will now see that there is a new field in the **General Tab** for a **Queue Callback**.

You can add this to any new or existing queue. Here, select the callback you created from the dropdown menu.



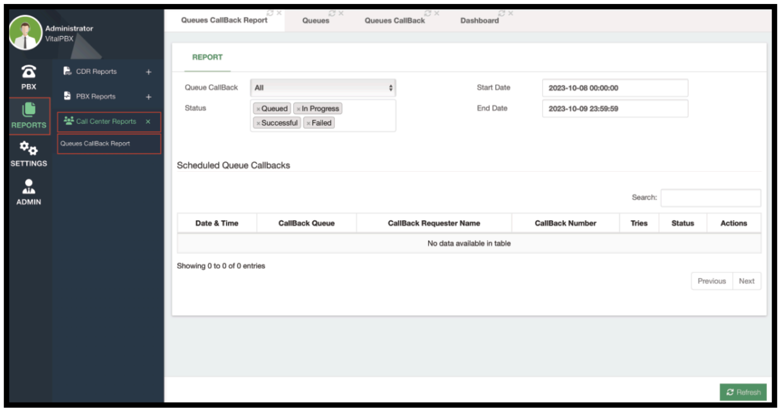
Code *	550	Queue Ring Time	0
Description *	My Queue	Member Ring Time	15
Strategy	Round Robin Memory	Retry	5
CID Name Prefix	MyQ	Wrap-up-time	
Join Announcement	None	Queue Callback	Disabled ✓ My Callback
Agent Announcement	None	Music on Hold	Default
Service Level	20	Ring Busy Agents	No
Join Empty	Yes	Ring Unavailable Agents	No
Leave When Empty	No	Record	No
Alert Info			

Image 6.8.2 - Queue callback assigned to a queue.

You can then click on **Update or Save** and then **Apply Changes**. Keep in mind that adding a Queue Callback will replace the *Periodic Announcement* with the *Instruction Message prompt*, and will also **disable** the ability to *associate an IVR with the Queue*.

Your users will now be able to opt in for a callback instead of waiting in the queue.

You can monitor the different callbacks from the new **Queues Callback Reports** under **Reports > Call Center Reports > Queues Callback Reports**. Here you will see a table of the different callers that have opted for a callback.



Queue CallBack: All

Status: × Queued × In Progress × Successful × Failed

Start Date: 2023-10-08 00:00:00

End Date: 2023-10-09 23:59:59

Scheduled Queue Callbacks

Search:

Date & Time	CallBack Queue	CallBack Requester Name	CallBack Number	Tries	Status	Actions
No data available in table						

Showing 0 to 0 of 0 entries

Image 6.8.3 - Queues Callback Report.

Here, you can filter the view by **Queue Callback**, **Status**, and **Start and End Date**. The Scheduled Queue Callbacks table has information on the **Date and Time** the caller opted for the callback, the **Callback Queue**, the **Callback Requester Name**, the **Callback Number**, the number of **Tries**, the **Status** of the callback, and **Actions** such as delete callback.
Using the Queues Callback module will give your callers a higher satisfaction rate as they don't have to hold on to a queue, and have more use of their time. This is a great addition to any Call Center environment.

Conclusion

With this, we conclude with the Call Center tools in VitalPBX's WebUI. Additionally, we recommend the Sonata Suite Guides, and VitXi, our browser-based softphone solution with Call Center features built-in. This joint set of tools will ensure you have a powerful Call Center environment.

SECTION 7 - INCOMING CALL TOOLS

Introduction

We have now seen ways for us to manage incoming calls with VitalPBX. Now it is time that we make our configuration more advanced with additional tools to make incoming calls more effective and efficient. With these tools, you can manage and manipulate the incoming calls. This way, your callers will always have a great experience calling into your business.

Section 7, Lesson 1 - Time Groups and Time Conditions

Businesses will tend to incoming calls within certain hours, or some will do a 24x7 service. Depending on the time of the day, you might tend to incoming calls in different ways. Or even on holidays, you might divert calls to another destination. With VitalPBX this is very easy using **Time Groups** and **Time Conditions**. This allows us to have incoming calls based on a schedule.

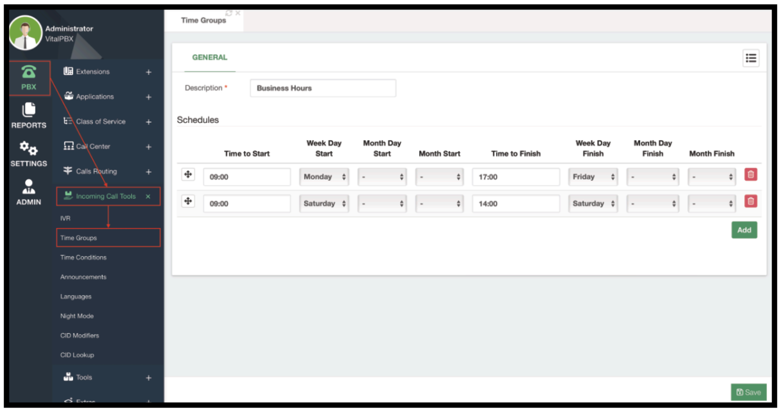
We have seen Time Groups in previous sections with the Least Cost Routing configurations, but in this section, we will add more details to the Time Groups.

So, to begin our configurations we must go to **PBX > Incoming Call Tools > Time Groups**. For this guide, we will do an example of setting business hours for our Incoming Calls Routing.

First, let's add a **Description** to identify this time group. Next, we will add our schedules. In this example, we will define the Business hours as follows.

- **Monday - Friday, from 09:00 to 17:00**
- **Saturday, from 09:00 to 14:00**
- **Sundays, closed**

To configure this, we will need two defined schedules. The first one will have the **Time to Start at 09:00, Week Day Start as Monday, Time to Finish at 17:00, and Week Day Finish as Friday**. Next, click the green **Add** button, and set the **Time to Start at 09:00, Week Day Start as Saturday, Time to Finish at 14:00, and Week Day Finish as Saturday**.



Description * Business Hours

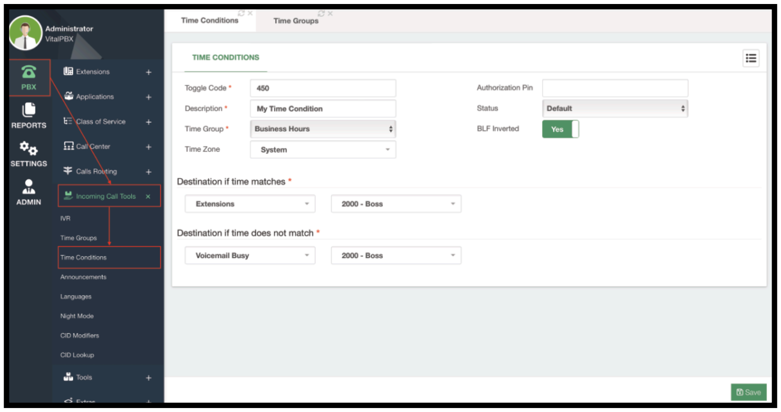
Schedules

Time to Start	Week Day Start	Month Day Start	Month Start	Time to Finish	Week Day Finish	Month Day Finish	Month Finish
09:00	Monday	-	-	17:00	Friday	-	-
09:00	Saturday	-	-	14:00	Saturday	-	-

Image 7.1.1 - Time Group with business hours example.

With these settings, whenever the system clock is within the defined schedules, the time group will return TRUE, matching our business hours.

Now that we have a Time Group with our Business Hours, we can apply them to a **Time Condition**. For this, we go to **PBX > Incoming Calls Tools > Time Conditions**.
Time conditions allow us to choose a destination based on a time group. First, we will define a **Toggle Code**, which will allow us to manually toggle the current status of the time condition. Next, we set a **Description** to identify the time condition, and we select the **Time Group** we have created.



Toggle Code * 450
Description * My Time Condition
Time Group * Business Hours
Time Zone System
Authorization Pin
Status Default
BLF Inverted Yes

Destination if time matches *
Extensions 2000 - Boss

Destination if time does not match *
Voicemail Busy 2000 - Boss

Image 7.1.2 - Time Conditions module.

Afterward, we have an **extended feature** only available with the **Starter License** or any of our **Licensing Plans**. This is the **Time Zone** option. This allows you to define a time zone specific to this time condition.

Then, you can enter an **Authorization PIN** for security purposes, so you need to enter this PIN when changing the status manually using the toggle code. You can also change the **Status** from here as well. You can choose one of the following options.

- **Temporary Matched -** The time condition will redirect calls to the If Time Matches Destination temporarily, until the next time the Time Group does not match.
- **Temporary Unmatched -** The time condition will redirect calls to the If Time Does Not Match Destination temporarily, until the next time the Time Group matches.
- **Permanently Matched -** The time condition will redirect calls to the If Time Matches Destination temporarily until this is changed directly.
- **Permanently Unmatched -** The time condition will redirect calls to the If Time Does Not Match Destination temporarily until this is changed directly.

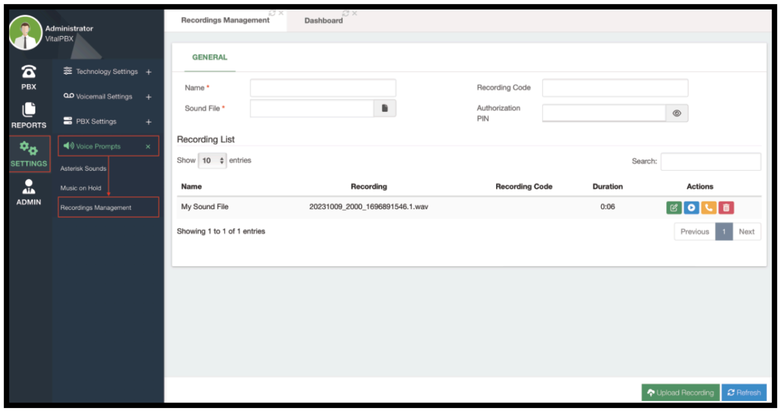
Afterward, you can enable or disable the **BLF Inverted**. By default, when monitoring Time Conditions with a BLF key, the light is green when the time matches and red when the time does not match.

Finally, we have the **Destination if the time matches and does not match**. This is the principle of how time conditions work. When the Status is set to Default, the time condition will be based on the time group. When the system time matches the time group you will be taken to a destination, and when it doesn't you are taken to the other destination. You can choose the destination of your liking for these options.
With this configured, you can now **Save** and **Apply Changes**.

Now, you have a time condition created with a time group. You can add this time condition to an IVR or Inbound Route, for example. This way, whenever someone calls into your VitalPBX, they will taken to the time condition and if the time matches they are taken to destination A, and if it doesn't match, they are taken to Destination B.

Section 7, Lesson 2 - Recordings Management

In multiple modules, you are able to have custom recordings used for announcements or prompts. You can upload your own sound files to playback in the different modules. To upload your own sound files we must go to **Settings > Voice Prompts > Recordings Management**.



Name *

Sound File *

Recording Code

Authorization PIN

Recording List

Show 10 entries

Search:

Name	Recording	Recording Code	Duration	Actions
My Sound File	20231009_2000_1696891546.1.wav		0:06	

Showing 1 to 1 of 1 entries

Previous 1 Next

Image 7.2.1 - Recordings Management module.

Before we upload a sound file, let's talk about the sound file format. It does not really matter the format you use as VitalPBX will automatically convert the sound file to the appropriate format. But, it helps with volume levels and clarity if you upload a file that is already using the appropriate format. The format goes as follows.

- **8,000 Hz**
- **16-bit PCM**
- **WAV file**
- **A-LAW algorithm**

You can use any audio processing software to convert this sound file, Audacity™ is a free open-source software that we have used to create these sound files.

With your sound file created, we can now go back to the Recordings Management module. Here, we will enter a **Name** for our recording, select the **Sound File** we have created, and click on the green **Upload Recording** button in the lower right-hand corner. And **Apply Changes**.

Optionally, you can enter a **Recording Code** so you can re-record the sound file from any extension, and add an **Authorization PIN**, to protect the recording from being re-recorded by anyone.

You can now use the sound file in various modules as we saw in Queues, Queues Callbacks, Extensions, and in any other module that uses custom audio files.

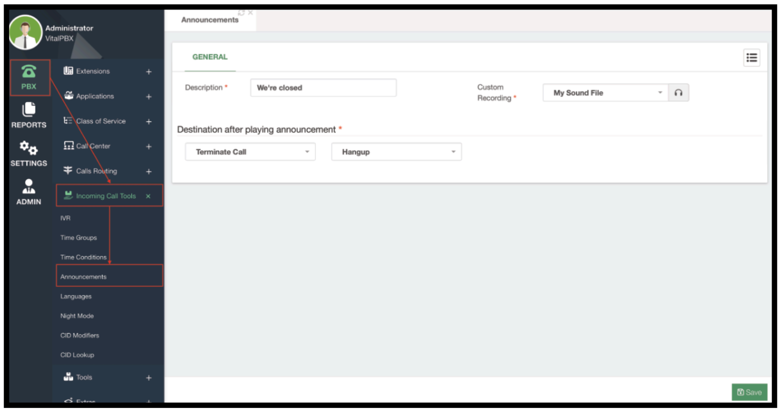
In the bottom section, you can see your **Recording List**. This will display information about the different recordings and sound files you have, including their **Name**, **Recording Sound File**, **Recording Code**, and **Duration**.

Additionally, you have some actions you can take over these recordings. You can edit them to change their **Name**, **Recording code**, and **Authorization PIN**. You can play the recording back and download the file. With the yellow handset icon, you can prompt an extension to re-record the sound file. And with the red trash bin icon, you can delete the recording.

Additionally, you can record these files from any extension using the feature code ***92**. This feature code will allow you to record sound files directly so you can use them for the various prompts in VitalPBX. There's also a blue Refresh button in the lower right-hand corner to refresh the list. This way you can see the latest recording you have done.

Section 7, Lesson 3 - Announcements

Sometimes you might want to simply playback an audio in your call flow. For this, we have **Announcements**. To create an announcement, we must go to **PBX > Incoming Call Tools > Announcement**.



Description * We're closed

Custom Recording * My Sound File

Destination after playing announcement *

Terminate Call | Hangup

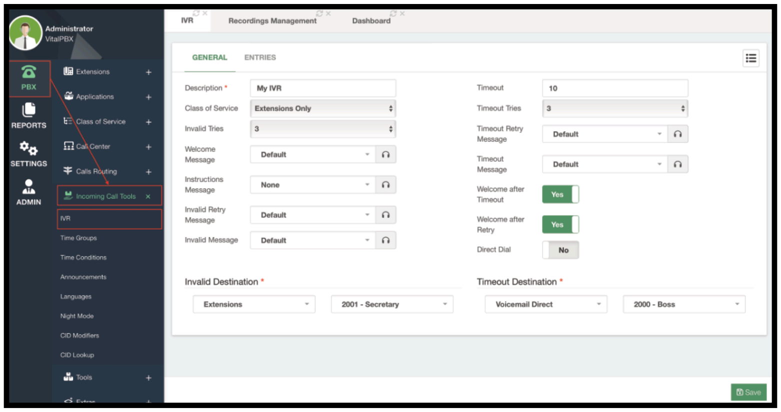
Image 7.3.1 - Announcements module.

Announcements are very simple to configure, you just need to enter a **Description** to identify your announcement, select a **Custom Recording** you uploaded using the Recordings Management module, and set a **Destination** after the recording plays.

Afterward, click on **Save** in the bottom right-hand corner, and **Apply Changes**. You can now use these announcements as a destination in IVRs, Time Conditions, Inbound Routes, and anywhere you want to playback a sound file.

Section 7, Lesson 4 - IVR

One of the most essential features in a PBX system is the automated attendant or **IVR** (*Interactive Voice Response*). With VitalPBX you can easily configure multiple IVRs that will help you reach any destination of your choosing. To configure an IVR, you must go to **PBX > Incoming Call Tools > IVR**.



Description * My IVR
Class of Service Extensions Only
Invalid Tries 3
Welcome Message Default
Instructions Message None
Invalid Retry Message Default
Invalid Message Default
Timeout 10
Timeout Tries 3
Timeout Retry Message Default
Timeout Message Default
Welcome after Timeout Yes
Welcome after Retry Yes
Direct Dial No
Invalid Destination * Extensions 2001 - Secretary
Timeout Destination * Voicemail Direct 2000 - Boss

Image 7.4.1 - IVR module's General Tab.

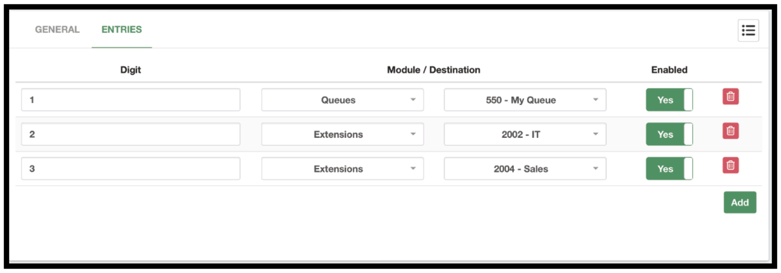
There are two tabs in the IVR module, the **General** and the **Entries** tabs. In the **General** tab, as the name implies, we will do general configurations for how the IVR menu will operate. First, we will add a **Description** to identify this IVR menu. Next, we will define the number of **Invalid Tries.** This is how many times a caller can input invalid responses before being transferred to the **Invalid Destination**, which can be any destination of your liking. To the right of this, we have the **Timeout time** and the **Timeout Tries**. This is the time you will give the caller to input anything, and the tries are the number of times the caller can timeout.

Next, we have a set of messages that will playback to the caller. The **Welcome and the Instructions Messages** will playback to the caller when they reach this IVR.

The welcome and instructions messages are separate, as you can playback just the instructions message after the *Timeout* and *Invalid Retries*. The welcome message is only played back after these retries if you have the **Welcome After Timeout and Welcome After Retry enabled**. The **Invalid Retry and Timeout Retry Messages** are played back after every Invalid and Timeout retry. The **Invalid and Timeout Messages** are played back right before the call goes to the *Invalid or Timeout Destinations*. These messages are sound files that you have uploaded using our *Recordings Management* module, as we saw in a previous lesson.

Finally, you have the **Direct Dial** feature. When enabled, this will allow callers to dial numbers directly from the IVR. If this is enabled, we have a default **Class of Service** called *Extensions Only* pre-selected. This is for security purposes, as you need to be careful about the numbers callers can dial from the IVR. If you have a permissive Class of Service, callers might be able to dial valid outbound numbers and this can lead to abuse of your trunks. So, if you require callers to Direct Dial anything else than your local extensions directly, make sure you create an appropriate Class of Service so they only dial what you expect them to dial.

Now, let's move over to the **Entries** tab. This is the basis for the IVR to operate. Here, you define the digits the caller will be able to press and take them to their selected destination. First, you will enter the **Digit** the caller can enter, then select the **Module/Destination** this will take them, and finally, you can **Enable or Disable** this option, as well as Delete it. You can add multiple digits, by clicking the green **Add** button.



Digit	Module / Destination		Enabled
1	Queues	550 - My Queue	Yes
2	Extensions	2002 - IT	Yes
3	Extensions	2004 - Sales	Yes

Image 7.4.2 - IVR module's Entries Tab.

With this, you can now click on **Save** in the lower right-hand corner, and **Apply Changes**.

> ***Note:*** *We can indeed create as many entry options as we want with the IVR module in VitalPBX. As a general recommendation and rule of thumb, we don't recommend adding more than 4 entries at a time for an IVR menu. This is for general customer satisfaction and operation. Customers will start to forget the options they are given after hearing more than 4 entries and might need to listen to the instructions message again. If you truly require more than 4 options, the best solution is to create sub-menus. This means that an IVR entry option will take you to another IVR menu with the rest of the options.*

Now that you have an IVR menu created, you can use this as a destination for your *inbound routes*. So whenever a customer calls your business, they will be presented with an elegant menu and options to direct them to their desired destination.

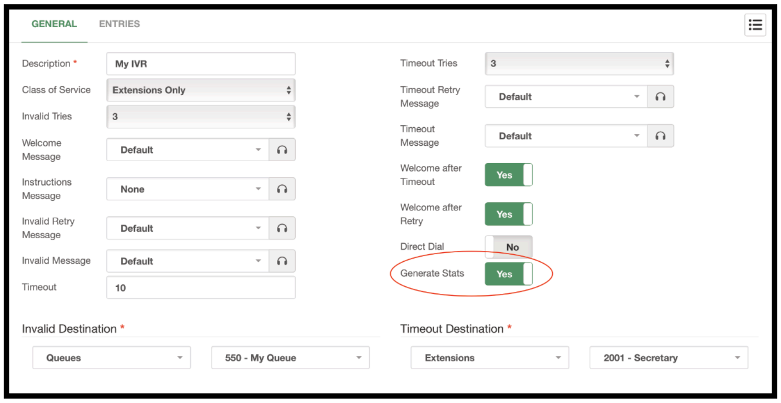
Section 7, Lesson 5 - IVR Stats and IVR-Based Surveys

If you want to measure which are the most popular options in an IVR menu, the **IVR Stats** add-on module is the best way to do this. This is also useful if you want to create surveys your callers can enter their answers and see how your business is doing.

The IVR Stats module is a commercial add-on, that requires a license. Without a license, you can only monitor one IVR menu. This module can be licensed individually, or with any of our licensing plans.

Before we take a look at the module, you must modify your IVR menus. A new option called **Generate Stats** will appear once you have the IVR Stats add-on installed. You must enable this option to see information in the IVR Stats module.

Afterward, you can go to **Reports > IVR Reports > IVR Stats**.

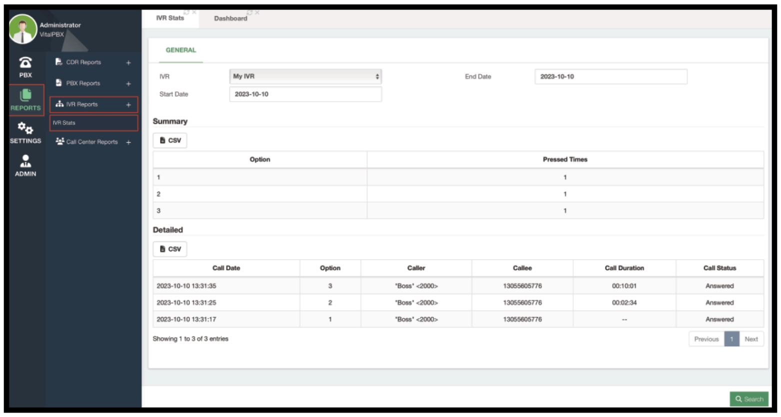


Description * My IVR
Class of Service Extensions Only
Invalid Tries 3
Welcome Message Default
Instructions Message None
Invalid Retry Message Default
Invalid Message Default
Timeout 10
Timeout Tries 3
Timeout Retry Message Default
Timeout Message Default
Welcome after Timeout Yes
Welcome after Retry Yes
Direct Dial No
Generate Stats Yes

Invalid Destination * Queues 550 - My Queue
Timeout Destination * Extensions 2001 - Secretary

Image 7.5.1 - Generate Stats option in an IVR menu.

The module is simple. You can select the **IVR** you wish to monitor, as well as a **Start and End Date** for a date range. Click on the **Search** button in the lower right-hand corner, and you will be shown a **Summary** and **Detailed** report for the IVR menu. The summary includes the entries and the number of times they have been pressed. While the detailed report has the call information related to the option pressed. Both of the reports can be **exported to a CSV file**. With this in mind, one of the most common use cases for the IVR stats is to **create surveys** your callers can answer.



IVR My IVR
Start Date 2023-10-10
End Date 2023-10-10

Summary

Option	Pressed Times
1	1
2	1
3	1

Detailed

Call Date	Option	Caller	Callee	Call Duration	Call Status
2023-10-10 13:31:35	3	"Boss" <2000>	13055605776	00:10:01	Answered
2023-10-10 13:31:25	2	"Boss" <2000>	13055605776	00:02:34	Answered
2023-10-10 13:31:17	1	"Boss" <2000>	13055605776	--	Answered

Showing 1 to 3 of 3 entries

Image 7.5.2 - IVR Stats module.

To create a survey you must create what we call an **IVR Tree**. An IVR tree is simply multiple IVR menus linked together. You have a main IVR menu we call the **IVR Trunk**, and each entry for this menu can lead to **IVR Branches**, which can lead to **IVR Leaves**.

An example can look as follows.

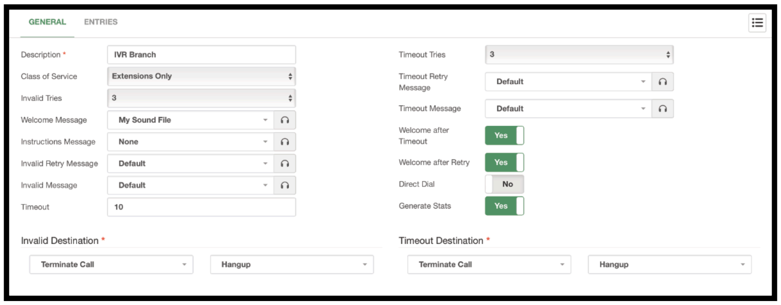
- **IVR Trunk** (Main Menu with the Main Question, e.g. *“How was our service?”*)
    - **IVR Branch** (Answer 1, e.g. *“Good”,* This branch has another question, e.g. *“Would you recommend us?”*)
        - **IVR Leaf 1** (Answer 1, e.g. “Yes”)
        - **IVR Leaf 2** (Answer 2, e.g. “No”)
    - **IVR Leaf 3** (Answer 2, e.g. *“Regular”*)
    - **IVR Leaf 4** (Answer 3, e.g. *“Bad”*)

The trunk and each branch would be an IVR menu. Leaves would be terminal destinations like announcements, voicemail boxes, or custom contexts.

You can add other IVR branches to your IVR branches, to expand on questions. You can make your surveys as simple or as complex as you need them to be.

The example above is a simple survey asking ***“How was our service?”*** The caller can answer ***“Good”*** by pressing 1, ***“Regular”*** by pressing 2, and ***“Bad”*** by pressing 3. The **IVR Branch 1** has the additional question ***“Would you recommend us?”*** The customer can answer ***“Yes”*** by pressing 1, and ***“No”*** by pressing 2. So, to create this survey, you must create 2 IVR menus, one for the trunk and the other one for the branch.

> ***Note:*** *You will notice that Entries are required to create the IVR menu, but if you don’t have the IVR menu branches created, you cannot set them as entry destinations. For this, you can create the branches first with their IVR leaves or use temporary entries first, and move from end to beginning to create your IVR tree.*



Description * IVR Branch

Class of Service Extensions Only

Invalid Tries 3

Welcome Message My Sound File

Instructions Message None

Invalid Retry Message Default

Invalid Message Default

Timeout 10

Timeout Tries 3

Timeout Retry Message Default

Timeout Message Default

Welcome after Timeout Yes

Welcome after Retry Yes

Direct Dial No

Generate Stats Yes

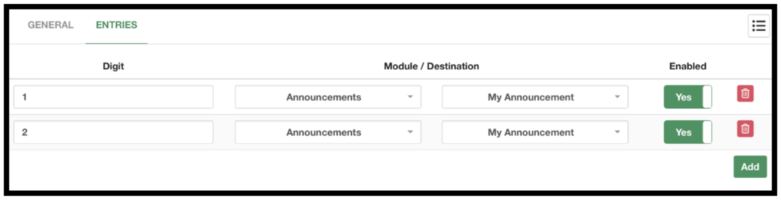
Invalid Destination * Terminate Call Hangup

Timeout Destination * Terminate Call Hangup

Image 7.5.3 - IVR Branch example.

First, we create our IVR Branch. The question will be the **Welcome Message**, so we also enable the **Welcome After Timeout** and **Welcome After Retry** options. For now, the **Invalid and Timeout Destinations** will be set to Hang Up the call. Remember to enable the **Generate Stats** option.

For our Entries, we add two entries with Digits 1 and 2 respectively. Ideally, these entries will take you to an **Announcement** that plays back a sound file thanking the caller for answering the survey.

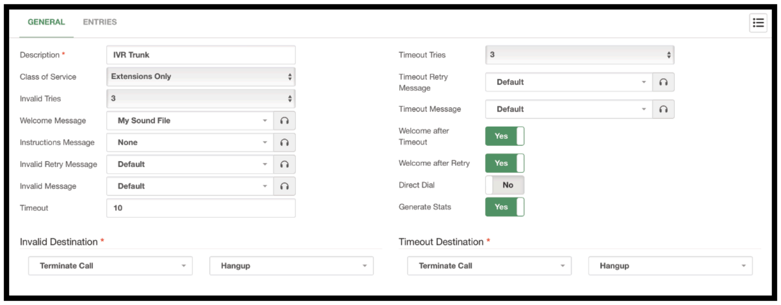


Digit	Module / Destination		Enabled
1	Announcements	My Announcement	Yes
2	Announcements	My Announcement	Yes

Image 7.5.4 - IVR Branch example entries.

You can now **Save** and **Apply Changes**.

Afterward, we can create the IVR Trunk. Similarly, the question will be set on the Welcome Message, and we enable the **Welcome After Timeout** and **Welcome After Retry** options. The **Invalid and Timeout Destinations** will be set to Hang Up the call. Remember to enable the **Generate Stats** option.



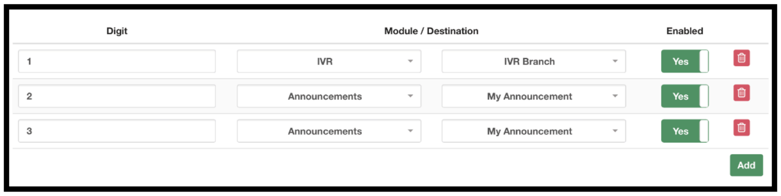
Description *	IVR Trunk	Timeout Tries	3
Class of Service	Extensions Only	Timeout Retry Message	Default
Invalid Tries	3	Timeout Message	Default
Welcome Message	My Sound File	Welcome after Timeout	Yes
Instructions Message	None	Welcome after Retry	Yes
Invalid Retry Message	Default	Direct Dial	No
Invalid Message	Default	Generate Stats	Yes
Timeout	10		

Invalid Destination *: Terminate Call, Hangup

Timeout Destination *: Terminate Call, Hangup

Image 7.5.5 - IVR Trunk example.

For the entries, based on our example layout, we create three entries. The first one will take you to the IVR Branch we created, and the second and third would ideally take you to an **Announcement** that thanks the caller for answering the survey.



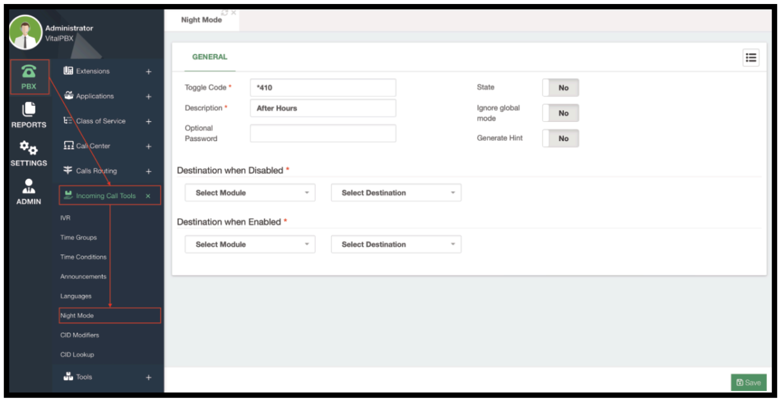
Digit	Module / Destination		Enabled
1	IVR	IVR Branch	Yes
2	Announcements	My Announcement	Yes
3	Announcements	My Announcement	Yes

Image 7.5.6 - IVR Trunk example entries.

With this, you can now add the survey as a destination in your VitalPBX. This can be on a Queue, for example, set as the **Destination if the Agent Hangs Up First**. So the caller can answer the survey, and you can monitor the answers with the IVR Stats module for the **IVR Tree**. IVR menus can be much more than just selectors for destinations.

Section 7, Lesson 6 - Night Mode

You might want to change the destination for incoming calls on command. For this, you make use of the **Night Mode** module. This module allows you to toggle between two destinations. To create our night modes, we must go to **PBX > Incoming Call Tools > Night Mode**.



Toggle Code * *410
Description * After Hours
Optional Password
State No
Ignore global mode No
Generate Hint No

Destination when Disabled *
Select Module ▾ Select Destination ▾

Destination when Enabled *
Select Module ▾ Select Destination ▾

Image 7.6.1 - Night Mode module.

All we need to do for our night mode is to enter a **Toggle Code** to enable and disable it from any extension, add a **Description** to identify it, and select the **Destinations when it is Enabled and Disabled**. Then we can **Save** and **Apply Changes**.

Additionally, you can add an **Optional Password**, so you need to enter this password to enable or disable the night mode. You can change the **State** of the night mode from here, instead of dialing the Toggle Code. You can also **Ignore the Global Mode**, so it is not affected by the feature code ***81** for the **Global Night Mode Toggle**. Lastly, you can enable **Generate Hint** so you can use the generated BLF Hint code to monitor the night mode status with a BLF key.

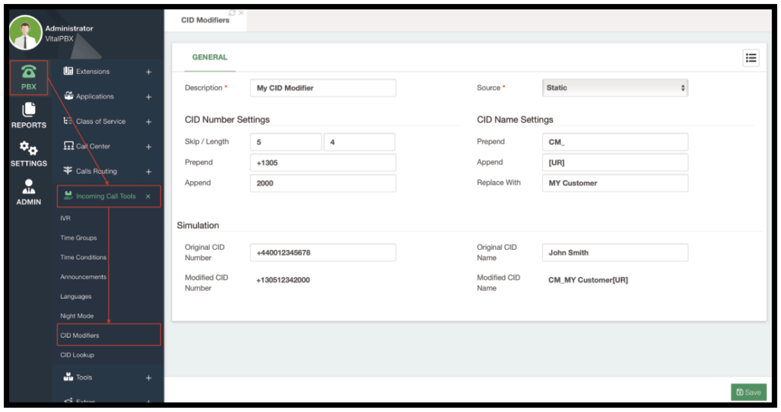
You can now use this night mode as a destination, and change the destinations with the toggle code.

Section 7, Lesson 7 - Caller ID Modifier

For standard procedures or verification, you might want to modify the Caller ID of incoming calls so you can identify them better. For this, you can use the **CID Modifiers** module. With this module, you are able to manipulate the Caller ID information for incoming calls so you can change it to whatever you need.

A common use is to add information to know the inbound route or trunk the call is coming from, or the DID the caller dialed to reach us. This way we can utilize the appropriate salutation for the incoming call.

To create your CID modifiers you must go to **PBX > Incoming Call Tools > CID Modifiers**. There are three types of **Sources** for the manipulation of CID information. **Static**, which is the standard way, by defining a hard-coded way to manipulate the CID information. **MySQL** to perform a MySQL query to modify the CID information based on the MySQL response. And finally, **HTTPS/HTTP Request**, so the CID information is based on the response.



CID Modifiers

Description * My CID Modifier

Source * Static

CID Number Settings

Skip / Length 5 4

Prepend +1305

Append 2000

CID Name Settings

Prepend CM_

Append [UR]

Replace With MY Customer

Simulation

Original CID Number +440012345678

Modified CID Number +130512342000

Original CID Name John Smith

Modified CID Name CM_MY Customer[UR]

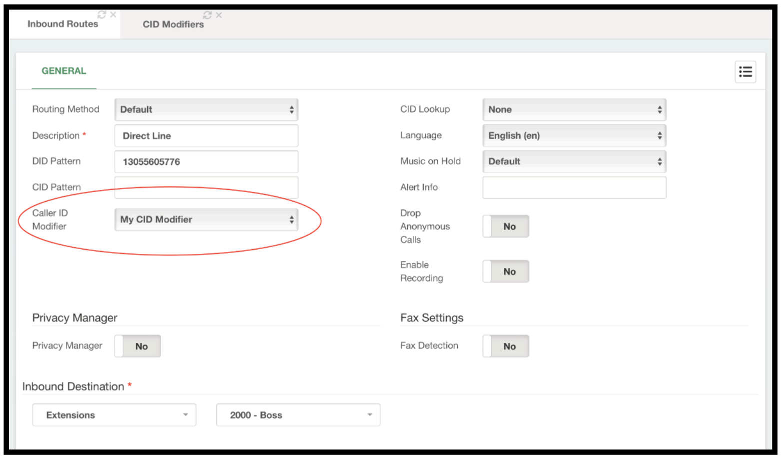
Image 7.7.1 - CID Modifiers module.

First, we must enter a **Description** to identify this CID Modifier. For a static CID modifier, we have **CID Number and Name Settings**. For the CID Number, you can enter how many characters to **Skip**, the total **length** of characters you are keeping, any characters or digits to **Prepend** before the CID Number, and any characters or digits to **Append** after the CID Number.

For the CID Name Settings, you can **Prepend** anything before the CID Name, **Append** anything after the CID Name, and outright **Replace** what the CID Name is.

Below the CID Number and Name settings, you have a **Simulation** section. This section allows you to enter sample CID information so you can see how this is modified.

You can then **Save** and **Apply Changes**. CID Modifiers can then be added to **Inbound Routes** under **PBX > Calls Routing > Inbound Routes**. This will apply the CID modifier through every incoming call that goes through this Inbound Route.



Routing Method Default

Description * Direct Line

DID Pattern 13055605776

CID Pattern

Caller ID Modifier My CID Modifier

CID Lookup None

Language English (en)

Music on Hold Default

Alert Info

Drop Anonymous Calls No

Enable Recording No

Privacy Manager

Privacy Manager No

Fax Settings

Fax Detection No

Inbound Destination *

Extensions 2000 - Boss

Image 7.7.2 - Inbound Route with a Caller ID Modifier.

For MySQL and HTTP/HTTPS requests, you can use the following variables for your queries.

- **[CIDNUM] -** This will return the Caller ID Number.
- **[CIDNAME] -** This will return the Caller ID Name.

This type of Source also provides the necessary fields for you to establish your connection.

A sample MySQL query can be as follows.

```
select cidname, cidnumber from customer where phone = ‘[CIDNUM]’
```

The response should return the cidname and cidnumber columns for the CID information to be modified.

Similarly with HTTPS/HTTP requests. A sample query can be as follows.

```
https://mycrm.com/cid.php?caller=[CIDNUM]&callername=[CIDNAME]
```

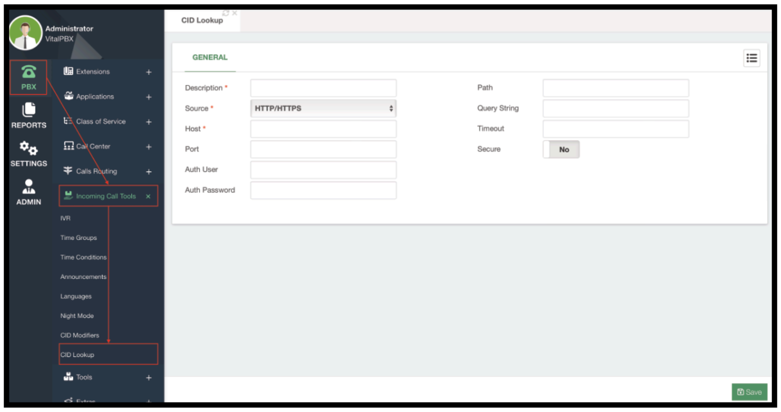
The URL execution must return a result in **JSON format** containing **cidname** and **cidnumber** attributes in order to modify the CID information.

You will now have the Caller ID information just how you need it, and operate following your standards and information handled.

Section 7, Lesson 8 - Caller ID Lookup

Many times, the incoming call may not have CID name information. If you are expecting calls from callers you have already logged or registered in a Database, the **CID Lookup** will allow you to look in those tables so you can add the CID name information.

To create the CID Lookups, we must go to **PBX > Incoming Call Tools > CID Lookup**.



Description *
Source * HTTP/HTTPS
Host *
Port
Auth User
Auth Password
Path
Query String
Timeout
Secure No

Image 7.8.1 - CID Lookup module.

The CID lookup can use any of 4 sources, an **HTTP/HTTPS request**, a **MySQL query**, **OpenCNAM**, and a **VitalPBX Phonebook.** You can use the **[CIDNUM]** variable in your query to retrieve the CID Name information from your database.

Any of these source types will show you the necessary fields to perform your connection and enter your query.

A sample HTTPS/HTTP request can be as follows.

```
https://mycrm.com/cid.php?cidnum=[CIDNUM]&customer=vip
```

For MySQL, the query can look as follows.

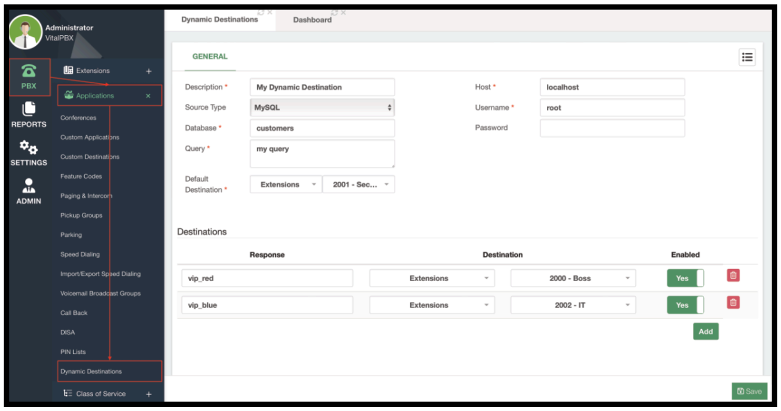
```
select `name` from `customers` where `number` = ‘[CIDNUM]’
```

The easiest option is to use VitalPBX Phonebooks. Since this allows you to upload a contact list using a CSV file, any contact in the phonebook can be used for CID lookup. Simply choose a Phonebook you have created and this is now your database for CID Names.

You can then apply the CID Lookup to your inbound routes, and the CID Name information will be populated by the CID Lookup response.

Section 7, Lesson 9 - Dynamic Destinations

An even more advanced way to route your incoming calls is using our **Dynamic Destinations** add-on module. This is a free add-on module you can install in your VitalPBX installation. Once the add-on is installed, you can go to **PBX > Applications > Dynamic Destinations**.



Description * My Dynamic Destination
Source Type MySQL
Database * customers
Query * my query
Default Destination * Extensions 2001 - Sec...
Host * localhost
Username * root
Password

Destinations

Response	Destination		Enabled
vip_red	Extensions	2000 - Boss	Yes
vip_blue	Extensions	2002 - IT	Yes

Image 7.9.1 - Dynamic Destinations module.

You can create multiple Dynamic Destinations, so you need to enter a **Description** to identify this dynamic destination. Dynamic destinations will query either a **MySQL** database or an **HTTP/HTTPS request** with a URL. Depending on the **Source Type** you choose, the required fields will appear so you can perform your **connection** and **query**.

A sample MySQL query can be as follows.

```
select ‘VIP_CUSTOMER’ as `response` from `customers` where `number` = ‘[CIDNUM]’
```

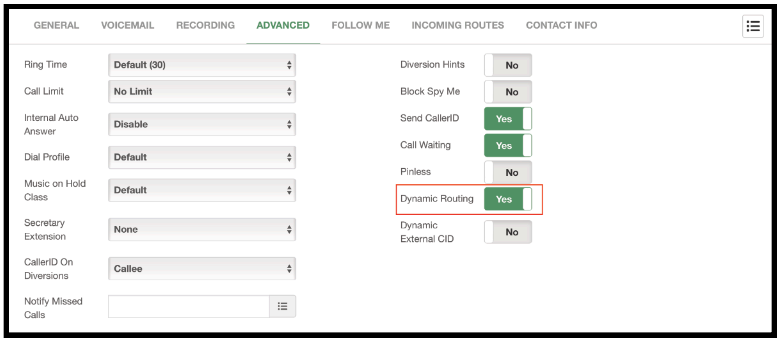
You can use the **[CIDNUM]** variable to add the Caller ID Number to your query. Depending on the **Response**, you can choose a **Destination**.

You can then use the Dynamic Destination as a destination for any module, for example, an inbound route.

Section 7, Lesson 10 - Dynamic Routing

As we saw in the Extensions section, you can enable a feature called **Dynamic Routing**. With dynamic routing, you can create dynamic routes to connect the calling party with the extension they were speaking with. This way, if you have a caller speaking with an extension, and they hangup or the call is dropped, the next time the caller calls your business, the Dynamic Routing module will check their Caller ID information and connect them with the extension, independently if there is another route for that DID.

To use this feature, you must first enable it on the extensions under the **PBX > Extensions > Extensions > Advanced**. Remember to **Save/Update** and **Apply Changes**.

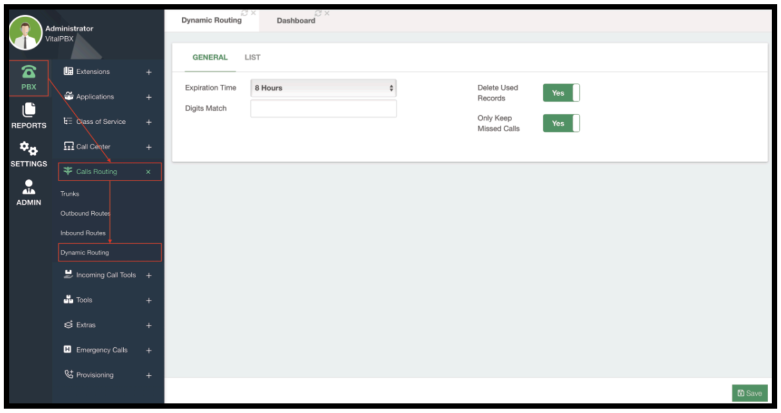


Ring Time: Default (30)
Call Limit: No Limit
Internal Auto Answer: Disable
Dial Profile: Default
Music on Hold Class: Default
Secretary Extension: None
CallerID On Diversions: Callee
Notify Missed Calls:

Diversion Hints: No
Block Spy Me: No
Send CallerID: Yes
Call Waiting: Yes
Pinless: No
Dynamic Routing: Yes
Dynamic External CID: No

Image 7.10.1 - Dynamic Routing enabled for an extension.

Once this is enabled, you can modify how the feature behaves by going to **PBX > Calls Routing > Dynamic Routing**.

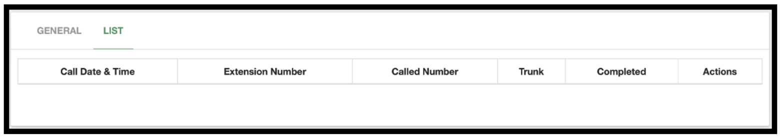


Expiration Time: 8 Hours
Digits Match:
Delete Used Records: Yes
Only Keep Missed Calls: Yes

Image 7.10.2 - Dynamic Routing module’s General Tab.

In the **General Tab**, you can change the **Expiration Time** for how long you will keep the Dynamic Route. You can then establish the number of **Digits to Match** for the caller's Caller ID number. You can then enable the ability to **Delete Used Records**, which will delete the dynamic route once it is used. This means that when the callers call again and are routed to the extension they were talking to the record will be deleted, and the next time they call, they will go through the normal routing. Finally, you can choose to **Only Keep Missed Calls**. This means that only outgoing calls that are missed by the extension will be saved to the Dynamic Routing List.

Next, we have the Dynamic Routing **List**. This is a table that will show you all the current dynamic routes with the caller information. You can also delete records from the Actions column.



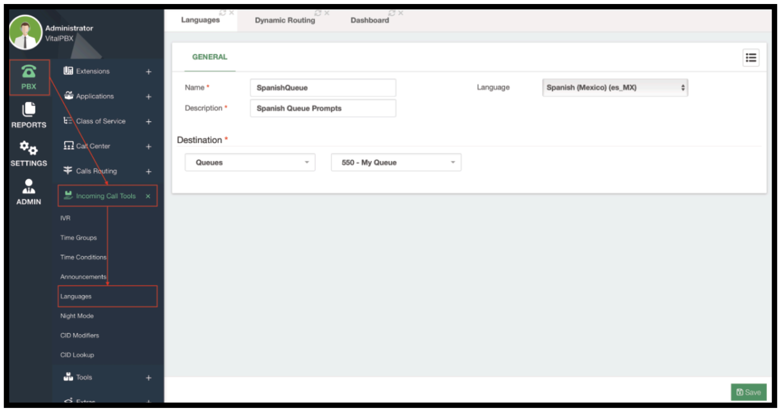
Call Date & Time	Extension Number	Called Number	Trunk	Completed	Actions

Image 7.10.3 - Dynamic Routing List.

If you made any changes here, you can **Save** and **Apply Changes**.

Section 7, Lesson 11 - Languages

Multiple businesses nowadays tend to their customers in multiple languages. It is important we are able to consider this possibility and our VitalPBX responds with the appropriate voice prompts. For this, we can intervene with a **Language** router. This will go in between our intended destination and our origin point, like an inbound route or IVR. To create this Language router we must go to **PBX > Incoming Call Tools > Languages**.



Name * SpanishQueue

Description * Spanish Queue Prompts

Language Spanish (Mexico) (es_MX)

Destination * Queues 550 - My Queue

Image 7.11.1 - Languages module.

Now, enter a **Name** and a **Description** to identify this language router. The name cannot have any spaces or special characters. Select the **Language** from the dropdown menu. For more language options refer to the **Asterisk Sounds** section and lesson. Finally, set your intended **Destination**. **Save** and **Apply Changes**.

With this set, you can place this language router between your origin and intended destination. This is most commonly an IVR where you have the option to *"press 2 for Spanish,"* for example. This entry is set to the language router we created, and the call is then taken to the Final Destination we set. Remember to **Save/Update** and **Apply Changes**.

Now all the voice prompts from here on will have the new language prompts play back.

Conclusion

With this, you have a more advanced experience with your incoming calls, giving all of the possibilities to your callers. With the use of all these tools, you can ensure that your calls will always be routed to the appropriate destination no matter the time, destination, or language.

SECTION 8 - APPLICATIONS

Introduction

We have now seen various essential destinations we can use with VitalPBX to tend to calls. Now, let's take a look into some additional applications we can use for a more feature-rich experience with your VitalPBX installation. Everything from Conferences, Speed Dials, Parking, and much more to come in this section.

Section 8, Lesson 1 - Feature Codes

We have now seen various codes we can use to perform multiple actions in VitalPBX. Here is the full list of feature codes you can use. You can find the **Feature Codes** under **PBX > Applications > Feature Codes**. Most of these codes can be modified to another code, as long as it does not clash with the existing numbering plan. Meaning that if the code is not used anywhere else in the system you can change it to anything. They can also be **Enabled or Disabled** from here. This is different from using a *Feature Category*, since this is a global setting, meaning that the feature is Enabled or Disabled for the whole system.

- **BLACKLIST**
    - ***30 - Blacklist a Number.** This feature code is used to enter a number you wish to blacklist.
    - ***31 - Remove Number From Blacklist.** This feature code is used to remove a number from the blacklist.
    - ***32 - Blacklist Last Caller.** This feature code is used to block the last caller from the extension dialing this code.

- **BUSINESS SERVICES**
    - ***34 - Wakeup Call.** This feature code is used to set up a wakeup call to the extension dialing this code.
    - ***35 - Remote Wakeup Call.** This feature code is used to set up a wakeup call to a remote extension. This feature comes disabled by default, so it needs to be enabled before use.
    - ***37 - Speak Last Number.** This feature code is used to listen to the Last Number that called the extension dialing this code.

- ***38 - Reminder.** This feature code is used to set up a reminder to the extension dialing this code. The user will record a message and set up the time when the reminder is played back to them.

- **CALL COMPLETION (CCSS)**
    - ***40 -** Call Completion (Toggle). This feature code toggles the Call Completion feature on and off.
    - ***41 -** Cancel Call Completion. This feature code cancels a call completion set.
- **CALL CENTER**
    - ***50 - Add/Remove Queue Agent.** This feature code will log in or log out an agent from a specific queue.
    - ***51 - Pause/Resume Queue Agent.** This feature code will Pause or Resume an agent from a specific queue.
    - ***52 - Queues Log In/Log Out.** This feature code will log in or log out an agent from all the queues they belong to.
    - ***53 - Queues Pause/Resume.** This feature code will Pause or Resume an agent from all the queues they belong to.
    - ***54 - Spy on Extension in Barge Mode.** This feature code will spy an extension's call in barge mode, meaning that the extension that dials this code is able to speak with both parties in the call they barged into. This feature is disabled by default. You need to enable it to use it.
    - ***55 - Spy on Extension.** This feature code will allow the extension that dialed the code to listen to the call on another extension. This feature is disabled by default and needs to be enabled to be used.
    - ***56 - Spy on Extension in Whisper Mode.** This feature code allows the extension that dialed this code to listen to another extension's call, but can only speak to the other extension not the third party in the call. This feature is disabled by default and needs to be enabled to be used.
    - ***57 - Spy Random Channels.** This feature code allows the extension that dialed this code to listen to a random extension's call. This feature is disabled by default and needs to be enabled to be used.
    - ***80 - Hot Desking.** This feature code allows you to assign an extension number with their device set to NONE to a Hot Desking device with no extension number.
    - ***90 - Hot Desking CC.** This feature code allows you to assign an extension number with their device set to NONE to a Hot Desking device with no extension number, and immediately log them into their queues. You can use the shortcode of ***90#[Extension Number]#[Features Password]** to assign the extension to the hot desking device and log them into their queues with their feature passwords in a single line.

- **CALL FORWARD**
    - ***36 - Boss/Secretary (Toggle).** This feature code will toggle the Boss/ Secretary feature on and off for the extension that dialed the code. They need to have a Secretary extension assigned for this feature to work.
    - ***58 - Call Forward Immediately (Toggle).** This feature code will toggle on and off call forwarding immediately to a number set.
    - ***59 - Set Call Forward Immediately Number.** This feature code allows you to set the number to call forward immediately for the extension that dialed this code.

- ***60 - Call Forward Unavailable (Toggle).** This feature code allows you to toggle on and off call forwarding to a number set when the extension that dialed this code is unavailable.
- ***61 - Set Call Forward Unavailable Number.** This feature code allows you to set the number to call forward when the extension that dialed this code is unavailable.
- ***62 - Call Forward Busy (Toggle).** This feature code allows you to toggle on and off call forwarding to a number set when the extension that dialed this code is busy.
- ***63 - Set Call Forward Busy Number.** This feature code allows you to set the number to call forward when the extension that dialed this code is busy.
- ***64 - Call Forward on No Answer (Toggle).** This feature code allows you to toggle on and off call forwarding to a number set when the extension that dialed this code does not answer the incoming call.
- ***65 - Set Call Forward on No Answer Number.** This feature code allows you to set the number to call forward when the extension that dialed this code does not answer the incoming call.
- ***66 - Do Not Disturb (Toggle).** This feature code allows you to toggle on and off the do not disturb feature for the extension that dialed this code.
- ***67 - Follow Me (Toggle).** This feature code allows you to toggle on and off the follow me feature for the extension that dialed this code.
- ***69 - Clear all Diversions.** This feature code allows you to disable any diversions that are turned on for the extension that dialed this code. This can be any Call Forward, Boss/Secretary, DND, and Follow Me.
- ***96 - Personal Assistant (Toggle).** This feature allows you to toggle on and off the Personal Assistant for the extension that dialed this code.

- **ON CALL FEATURES**
  - ***0 - Disconnect Call.** This feature code allows you to disconnect your current call. This code is always enabled and cannot be changed.
  - ***00 - Pause/Resume Outbound Call Recording.** This feature code allows you to Pause and Resume the recording of your current outbound call. This code is always enabled and cannot be changed.
  - ***07 - Direct Pickup.** This feature code allows you to pick up a call ringing in another extension directly. To use this code dial *07+Extension Number that is ringing.
  - ***08 - Pickup Group.** This feature code allows you to pick up a call ringing another extension that belongs to your same Pickup Group.
  - ***1 - Pause/Resume Inbound Call Recording.** This feature code allows you to Pause and Resume call recording on your current inbound call. This code cannot be disabled or changed.
  - ***2 - Attended Transfer.** This feature code allows you to perform an attended transfer of your current call. An attended transfer is when the party you wish to transfer the call to needs to answer your call before you transfer the call to them. This code cannot be disabled or changed.
  - ***3 - One Touch Recording.** This feature code allows you to perform an on-demand recording of your current call. You need to have on-demand recordings enabled on your extension to use this feature. This code cannot be disabled or changed.
  - ***4 - Park Call.** This feature code allows you to Park calls to an available parking space in the default parking. This code cannot be changed or disabled.

  - **#1 - Blind Transfer.** This feature code allows you to perform a blind transfer of your current call. A blind transfer means that the call is transferred immediately to the party you wish to transfer the call to. This code is always enabled and cannot be changed.

- **PHONEBOOK DIRECTORY**
  - **411 - Dial By Name Directory.** This feature code allows you to access the dial-by-name directory. In this directory, you can partially dial the name of an extension to find it.
- **TEST SERVICES**
  - ***70 - Speak Date and Time.** This feature code will playback the current system Date and time.
  - ***71 - Speak Your Extension Number.** This feature code will playback the extension number from the extension that dialed this code.
  - ***72 - Echo Test.** This feature code will allow you to perform an echo test. This is a test to hear the delay between the system and the extension. Whatever you say will be played back to you. The less delay there is, the better the connection.
  - ***73 - Simulate Incoming Call.** This feature code will allow you to simulate an incoming by having the system call you back after you dial the code.

- **SPECIAL FEATURES**
  - ***75 - Lock/Unlock Phone.** This feature code will allow you to toggle the lock on the extension that dialed this code.
  - ***76 - Change Features Password.** This feature code will allow you to change the Features Password from the extension that dialed this code.
  - ***77 - Remote Substitution.** This feature code will allow you to place a call from any extension as if you did it from another extension number with their privileges. You will be asked for the other extension's feature password to perform this operation.
  - ***78 - Customer Code.** This feature code will allow you to input a customer code you have created so it appears in the CDR information.
  - ***79 - Authorization Code.** This feature code will allow you to enter an Authorization Code you have created to change the Class of Service for the extension that dialed this code for a single call.
  - ***81 - Night Mode All.** This feature code enables all of the Night Modes created. Unless the Night Mode has the Ignore Global option enabled.
  - ***82 - Paging.** This feature code will allow you to page a specific extension number. You need to dial *82+Extension Number to use this code.
  - ***83 - Paging Duplex.** This feature code allows you to page an extension and the other party will be able to talk back. You need to dial *83+Extension Number to use this code.
  - ***88 - Anonymous Calling.** This feature code allows you to call another extension without showing your Caller ID.

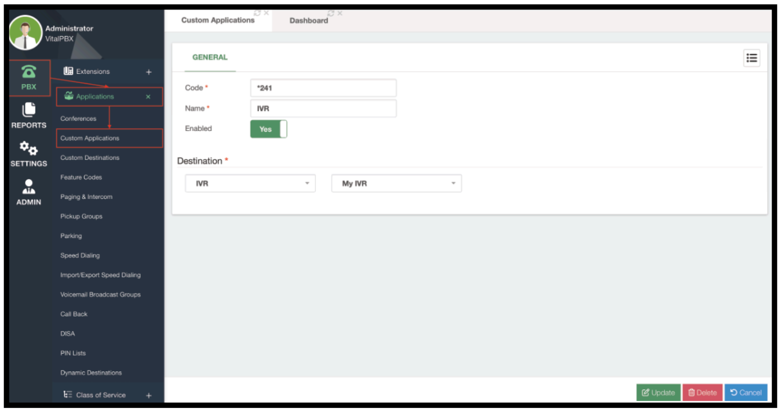
- **RECORDINGS & ANNOUNCEMENTS**
  - ***92 - Custom Recording.** This feature code allows you to create a custom recording. You can then see these recordings in the Recordings Management module.
  - ***93 - Dictation.** This feature code allows you to create dictation recordings. The Extension needs to have this feature enabled.

- ***94 - Record Personal Assistant Message.** This feature code allows you to record your Personal Assistant menu message.
- ***95 - Send Voicemail Message.** This feature code allows you to send a voicemail message to an extension.
- ***97 - Direct Voicemail.** This feature code allows you to access the extension that dialed this code's voicemail box.
- ***98 - Remote Voicemail.** This feature code allows you to access a remote voicemail box. The Features Password is needed to access the voicemail box remotely.

Section 8, Lesson 2 - Custom Applications and Destinations

You may find the need to be able to dial certain destinations that do not use a code, like IVRs, for example. Or maybe, you wish to have an external number as a destination for a module. You can perform these tasks using **Custom Applications** and **Custom Destinations**.

Let's start with **Custom Applications** by going to **PBX > Applications > Custom Applications**.



Code * *241
Name * IVR
Enabled Yes
Destination *
IVR
My IVR

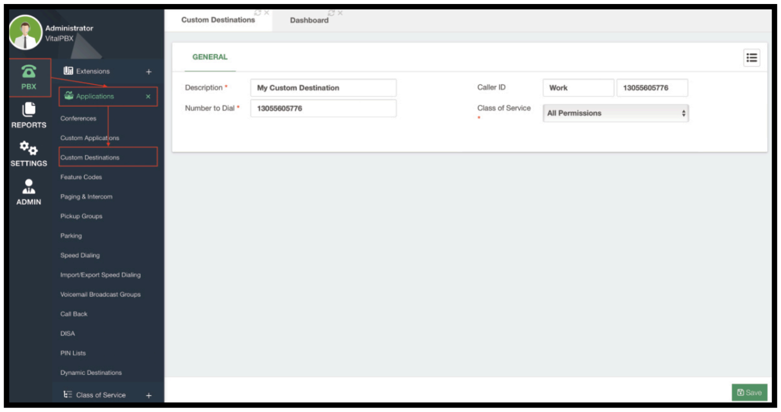
Image 8.2.1 - Custom Applications module.

A custom application allows you to assign a **Code** to a **Destination** that does not use one, for example, IVRs. So with the code and destination, all you need is to add a **Name** and **Enable** it. Then **Save** and **Apply Changes**.

You can then dial the code you just created and you will reach the destination assigned to it.

Now, let's say you wish to add a code or feature external to your VitalPBX installation and use it as a destination. Like a code in another PBX system, or a cellphone number. For this, we can create a Custom Destination under **PBX > Applications > Custom Destinations**.
Here, you just add a **Description** to identify the destination, and the **Number to Dial**, add **Caller ID** information if needed, and set a **Class of Service**. Now, you can set this as a destination in any module, and no matter the Class of Service the extension or module might have, the Custom Destination's Class of Service is the one considered. This way, if you have a restrictive Class of Service, you can use these Custom Destinations to reach external features.

You can even use a Custom Application that uses a Custom Destination to reach those features.



Description * My Custom Destination

Number to Dial * 13055605776

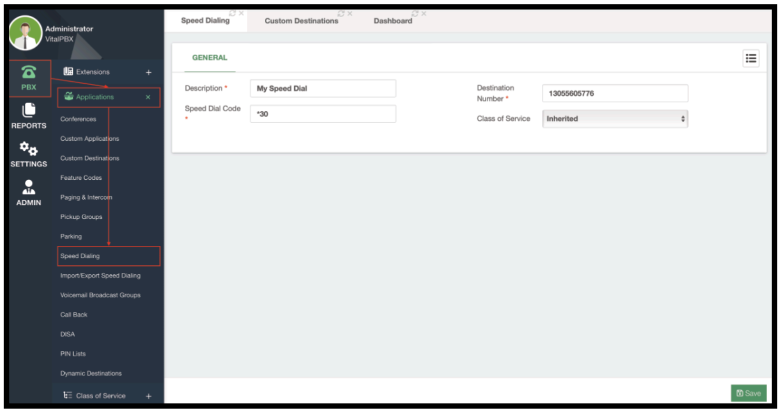
Caller ID Work 13055605776

Class of Service * All Permissions

Image 8.2.2 - Custom Destinations.

Section 8, Lesson 3 - Speed Dialing

If you or multiple people are calling a long number often, using a **Speed Dial** is the best way to shorten it. To create a speed dial you must go to **PBX > Applications > Speed Dialing**.



Description * My Speed Dial

Speed Dial Code * *30

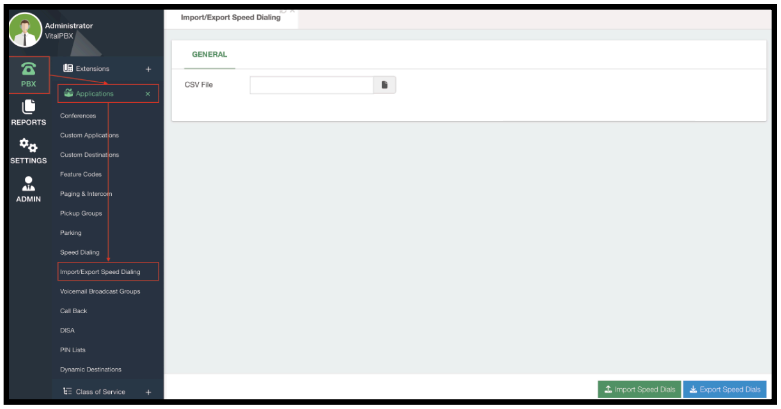
Destination Number * 13055605776

Class of Service Inherited

Image 8.3.1 - Speed Dialing module.

Simply add a **Description** to identify this speed dial, set the **Speed Dial Code**, and enter the **Destination** number. The **Class of Service** can be inherited from the extension or module that dialed this speed dial or any Class of Service you specify. Then **Save** and **Apply Changes**. Now, whenever you dial the Speed Dial code, you will be calling the longer destination number.

Additionally, you can import and export Speed Dials using the **Import/Export Speed Dialing** module. This module is located at **PBX > Applications > Import/Export Speed Dialing**. You can upload a **CSV file** and click on the green **Import Speed Dials** button in the lower right-hand corner. To export the Speed Dials you have created, click on the blue **Export Speed Dials** button in the lower right-hand corner.

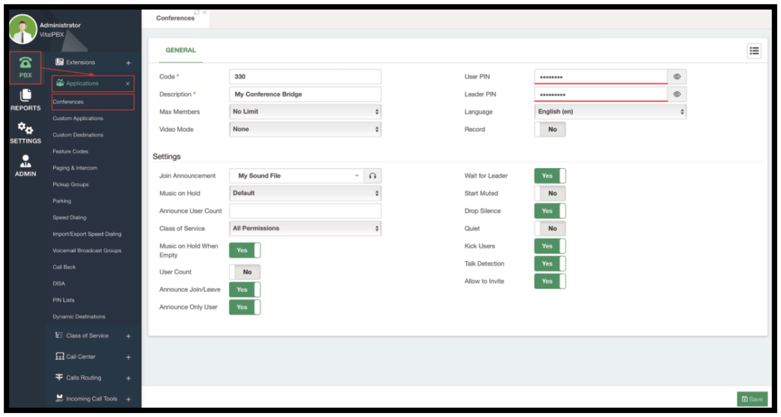


CSV File

Image 8.3.2 - Import/Export Speed Dialing module.

Section 8, Lesson 4 - Conference Bridges

It is common for multiple people to join in a single conversation. With VitalPBX you can host audio conference rooms for multiple people in what we call **Conference Bridges**. To create one, you must go to **PBX > Applications > Conferences**.



Code *	330	User PIN	•••••••••
Description *	My Conference Bridge	Leader PIN	••••••••••
Max Members	No Limit	Language	English (en)
Video Mode	None	Record	No

Settings

Join Announcement	My Sound File	Wait for Leader	Yes
Music on Hold	Default	Start Muted	No
Announce User Count		Drop Silence	Yes
Class of Service	All Permissions	Quiet	No
Music on Hold When Empty	Yes	Kick Users	Yes
User Count	No	Talk Detection	Yes
Announce Join/Leave	Yes	Allow to Invite	Yes
Announce Only User	Yes		

Image 8.4.1 - Conference Bridges module.

With a conference, more than 3 people can join in a single call. They can all call into the conference bridge from either any extension or an incoming call to join the conversation.

First, you must enter the **Code** used to reach this conference bridge and add a **Description** to identify it. We recommend you set a **User and a Leader PIN** to identify users and leaders for this conference bridge. All other settings can be left by default and then **Save** and **Apply Changes**. Additionally, you can customize the Conference Bridge with the following settings.

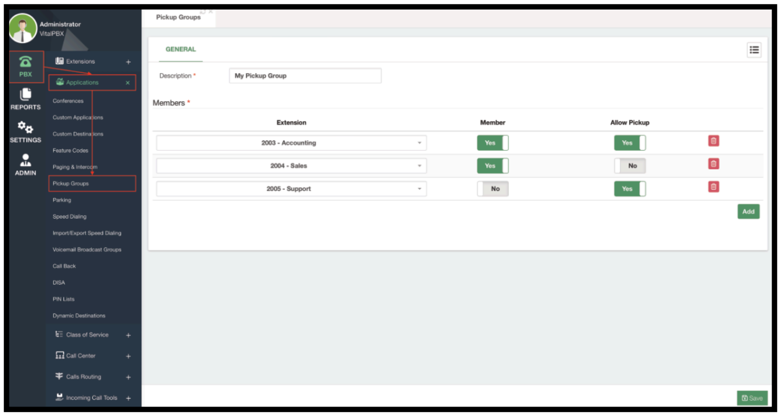
- **Max Members -** This is used to limit the number of participants in this conference bridge.

- **Video Mode -** You can enable this feature to have video in the conference bridge. It is important to know that this will only work for extensions that have devices that support PJSIP/SIP-based video calling. This will not work for external callers doing an incoming call to the VitalPBX installation. You can choose between these options:
    - **None -** This is used to disable the video mode feature.
    - **Follow Talker -** This is used to show the video of who's currently talking in the conference.
    - **Admin -** This will show the video for leaders only at the conference.
    - **Last Admin -** This will show the video for the last leader to join the conference.
    - **Selective Forwarding Unit -** This sets a multi-stream operation for a multi-stream video conference so multiple people are shown at once.
- **Language -** This sets the language for the voice prompts in the conference bridge.
- **Record -** When enabled, this will record the conference call.
- **Join Announcement -** This is an announcement that is played back when someone joins the conference.
- **Music on Hold -** This is a Music on Hold class that will play when a user is waiting.
- **Announce User Count -** You can set this to Yes to enable it, No to disable it (this is the default value), and a positive number to announce the number of users to everyone when the number of members is above this number.
- **Class of Service -** You can assign a Class of Service to a conference bridge since you can enable inviting external numbers. This determines the numbers you can dial from within the conference.
- **Music on Hold When Empty -** This determines if MoH will play while there is only one person in the conference.
- **User Count -** Announces to new callers the number of participants in the conference.
- **Announce Join/Leave -** Will announce conference members when someone joins or leaves the conference. Users will be asked to record their names before joining the conference.
- **Announce Only User -** This will announce the last member that they are the only member left in the conference.
- **Wait for Leader -** Users will not be able to speak with each other until a Leader joins.
- **Start Muted -** All users start muted when they join the conference.
- **Drop Silence -** When a member is not speaking, silence will not be transmitted, saving bandwidth and improving performance.
- **Quiet -** This will mute all voice prompts from this conference.
- **Kick Users -** All users are kicked when the last leader leaves the conference.
- **Talk Detection -** This determines whether to send event notifications when a user starts and ends speaking to the Asterisk Management Interface (AMI).
- **Allow to Invite -** This allows inviting external people once you are in the conference dialing ****** or **0**.

Section 8, Lesson 5 - Pickup Groups

It is common for extensions to be located in near physical locations. So, there might be occasions when a user is not at their desk and their extension starts to ring. Another user can pick up the call, but instead of moving physically to the other desk, they can pick up the call from their extension. We already saw that you can use the ***07+Extension Number** to directly pick up a call. But if this occurs frequently, you can assign these extensions to a **Pickup Group**.

To create a Pickup Group, you must go to **PBX > Applications > Pickup Groups**.



Description * My Pickup Group

Members *

Extension	Member	Allow Pickup
2003 - Accounting	Yes	Yes
2004 - Sales	Yes	No
2005 - Support	No	Yes

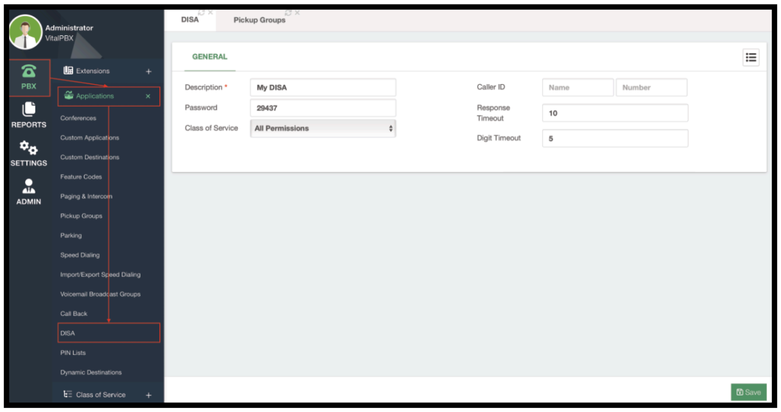
Image 8.5.1 - Pickup Groups module.

First, enter a **Description** to identify the Pickup Group. Then, we need to add **Members**. You can add as many members as you want by clicking the green **Add** button in the right-hand corner. And you can remove them by clicking the red **Trash** button to the right.

Members are comprised of an **Extension** number, whether or not they are **Members**, and whether or not they are **Allowed to Pick up**. If you are a member, your calls can be picked up, and if you are allowed to pick up you can pick up the calls from another member. So, you can combine these options to select who can pick up calls and whose calls can be picked up within the Pickup Group.

Section 8, Lesson 6 - DISA

Now let's look into DISAs. DISA stands for Direct Inward System Access. This means that this is a feature that allows you to use internal features from a PBX from an external call. This way, you can call your VitalPBX from an external number and start using any internal feature like Custom Applications, call other extensions directly, or any other feature that uses an internal code.



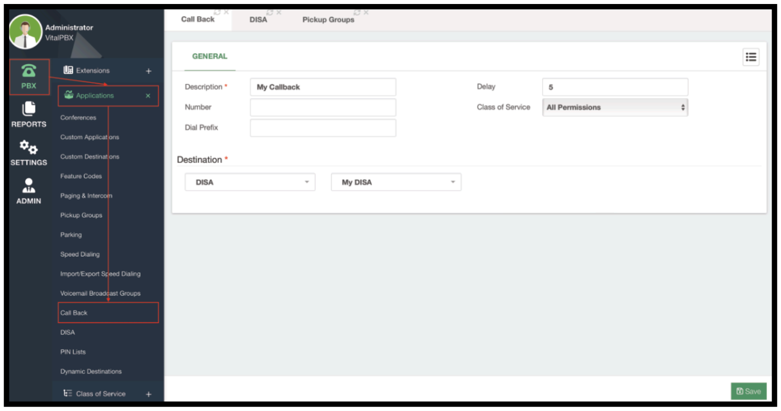
Description * My DISA
Password 29437
Class of Service All Permissions
Caller ID Name Number
Response Timeout 10
Digit Timeout 5

Image 8.6.1 - DISA module.

To create a DISA we must first go to **PBX > Applications > DISA**. A DISA only requires a **Description** so you can identify it. A **Password** is automatically generated and is highly recommended since this will allow you to dial numbers based on the **Class of Service** assigned. You can then establish a **Caller ID Name and Number** to use since you can place outgoing calls from the DISA. Keep in mind that whenever you place an incoming call to a DISA and then make an outgoing call you will have two calls established through your trunk.
Finally, you have the **Response** and **Digit Timeouts**. These are the amount of time in seconds you have to enter a number, and the time in between digits pressed.
You can then **Save** and **Apply Changes**. Now you can use the DISA to call into your VitalPBX and dial internal codes and numbers as if you are an extension in the system.

Section 8, Lesson 7 - Call Back

Next, let's take a look at the **Call Back** module. The Call Back module allows you to set an automatic call from VitalPBX to a phone number and connect it with a destination. Usually, you will have a callback for each user that utilizes this feature, with their phone numbers set. To create a callback we must go to **PBX > Applications > Call Back**.



Description * My Callback
Number
Dial Prefix
Delay 5
Class of Service All Permissions
Destination *
DISA
My DISA

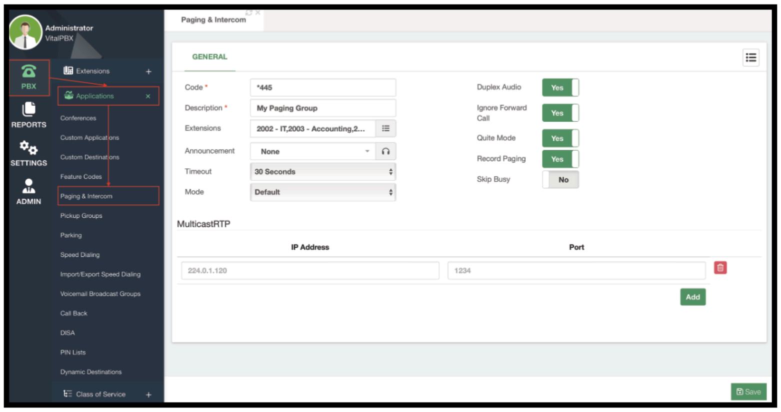
Image 8.7.1 - Call Back module.

For a callback, we first set a **Description** so we can identify it. Next, we have the **Number** and **Prefix** field. These are used to establish a static number to callback when you reach this callback destination. The prefix is used if it is needed with your outbound routes. If the Number field is left blank, then the Caller ID Number is used instead. You can add a **Delay** in seconds from when the incoming call ends and the system places the callback. You need a **Class of Service** since this module places an outbound call, so it needs to have permission to dial the set number. Finally, set the **Destination** to where the callback will connect with the outgoing call. **Save** and **Apply Changes**.

This feature is usually used so it is the business that is charged for the calls instead of your personal phone number. We see this most often used with the DISA module as it will your external phone number and connect you with a DISA so you can use internal features or place outgoing calls from the business's VitalPBX instead of your personal devices.

Section 8, Lesson 8 - Paging and Intercom

Up next, we have the **Paging and Intercom** module. This module allows you to create what we call **Paging Groups**. Paging groups can be comprised of devices that support auto-answer, or dedicated devices for Paging and Intercom. These dedicated devices can be SIP-based paging speakers, for example. To create our paging groups, we must go to **PBX > Applications > Paging and Intercom.**



Code * : *445
Description * : My Paging Group
Extensions: 2002 - IT,2003 - Accounting,2...
Announcement: None
Timeout: 30 Seconds
Mode: Default
Duplex Audio: Yes
Ignore Forward Call: Yes
Quite Mode: Yes
Record Paging: Yes
Skip Busy: No

MulticastRTP

IP Address	Port
224.0.1.120	1234

Image 8.8.1 - Paging and Intercom module.

First, you need to enter the **Code** you will use to reach this paging group, this can use any digits and the star (*) symbol. Next, add a **Description** to identify this paging group. Afterward, include the **Extensions** you wish to page. Again, if these are regular desktop devices, they need to support auto-answering with speaker mode. Otherwise, this will not work. If you want to be able to have a two-way conversation with the paged extensions, you can enable **Duplex Audio**, so you can speak both ways. With this, you can **Save** and **Apply Changes**. Now you can dial the code you created and page the extensions assigned to the paging group.

Additionally, you can add the following settings.

- **Announcement -** this is an announcement that will play on the selected extensions when you call this paging group.
- **Timeout -** This is the time in seconds after which the paging group call will timeout.
- **Mode -** Here you establish the Default mode, which is to speak from the device you called the paging group, or Announcement, to playback the announcement set to this paging group.
- **Ignore Forward Call -** When this is enabled, if the extension in the paging group has a call forward enabled, this will be ignored, and the extension will ring with auto-answer anyway.
- **Quiet Mode -** This will disable the tone that plays when calling this paging group announcing that the extension is being paged.
- **Record Paging -** This will record the call whenever the paging group is called.
- **Skip Busy -** This will ignore the status of the extension if the extension is currently in a call and ring the device anyway.

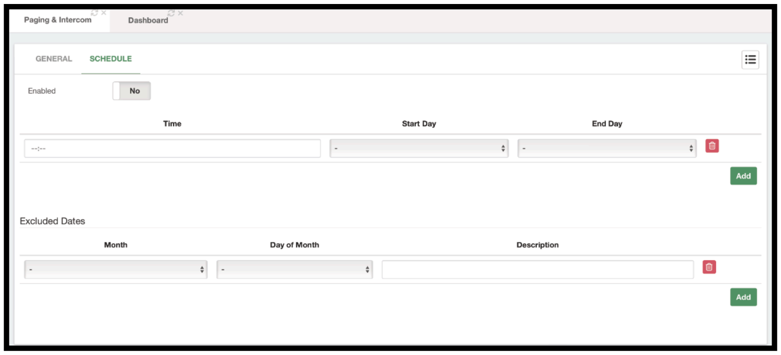
Finally, you can also establish **MulticastRTP IP Addresses** through a specific **Port**. If the devices you wish to page, like a dedicated paging device, use MulticastRTP you can establish its IP Address and port to cast the RTP packets. You can add as many IP Addresses as you need by clicking the green **Add** button in the right-hand corner.

Section 8, Lesson 9 - Paging Pro

Taking advantage of the announcement abilities with the Paging and Intercom module, we have created a commercial add-on module called **Paging Pro**. This commercial add-on module allows you to schedule the announcements to playback in the paging group devices automatically.

This is useful for environments with constant announcements that need to be played in a schedule. For example, transportation terminals, schools, warehouses, etc.

To use the Paging Pro add-on, you need to install it under **Admin > Add-Ons > Add-Ons**. Once the add-on is installed the feature will expand the Paging and Intercom module. So, we can go back to **PBX > Applications > Paging and Intercom**.



Enabled: No

Time	Start Day	End Day
--:--	-	-

Excluded Dates

Month	Day of Month	Description
-	-	

Image 8.9.1 - Paging Pro scheduling feature.

You will now see a **Schedule** tab. Here you can Enable or Disable the scheduling of the automatic announcement. Then, you will see fields to add dates and times of when the announcements will be played. The **Time** is the time of the day it will play the announcement, **Start and End Day** is a range of days that the announcements will be scheduled.

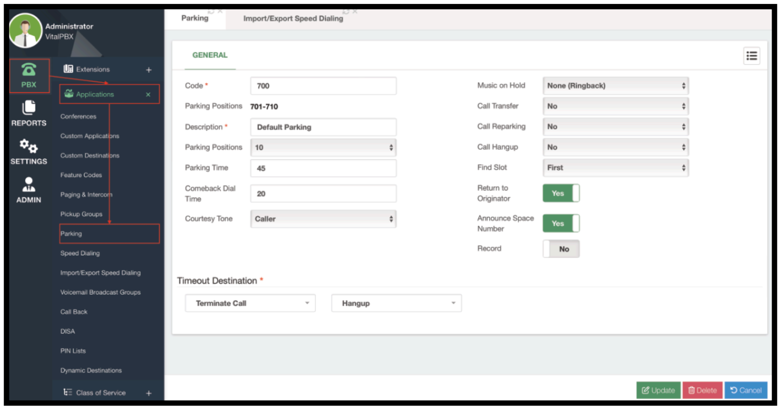
Next, you can **exclude specific dates** where the announcements will not playback. These can be special holidays or off dates where you don't want the announcements to page the paging groups.

This is a great tool to automate processes within your business and offer a greater experience to your customers.

Section 8, Lesson 10 - Call Parking

Next up we have **Call Parking**. Call parking is best understood if you think of it as a literal parking lot, where you can park ongoing calls into spaces within the parking lot. By default, we have the **Default Parking 700** with 10 spaces available. To create a Parking, you must go to **PBX > Applications > Parking**.

The first thing you will need to establish is the **Code** for this parking. The code works a little bit differently with parking. The code is used to reach the parking lot as a whole. The parking spaces start from the next number from your code and are defined by the number of **Parking Positions** you establish. For example, if you set the code to 900 and define 10 parking positions, your parking spaces are from 901 to 910. You can leave the rest of the fields with the default values and **Save** and **Apply Changes**.



Code * 700
Parking Positions 701-710
Description * Default Parking
Parking Positions 10
Parking Time 45
Comeback Dial Time 20
Courtesy Tone Caller
Music on Hold None (Ringback)
Call Transfer No
Call Reparking No
Call Hangup No
Find Slot First
Return to Originator Yes
Announce Space Number Yes
Record No
Timeout Destination * Terminate Call Hangup

Image 8.10.1 - Call Parking module.

Additionally, you can configure the following options.

- **Parking Time -** This is the amount of time in seconds a call can stay in a parking lot before returning.
- **Comeback Dial Time -** This is the amount of time in seconds a parked call will dial the device that originally parked the call after timing out in the parking space.
- **Courtesy Tone -** This option will play back a courtesy tone when picking up a parked call. This can playback to the caller, the callee, or both.
- **Music on Hold -** This is the Music on Hold class that plays while in a parking space.
- **Call Transfer -** This is used to specify who can execute a DTMF-based transfer in the call when picking up a parked call. You can choose between the caller, the callee, both, or no.
- **Call Re-Parking -** This is used to specify who can execute DTMF-based parking in the call when picking up a parked call. You can choose between the caller, the callee, both, or no.
- **Call Hangup -** This is used to specify who can execute a DTMF-based hangup in the call when picking up a parked call. You can choose between the caller, the callee, both, or no.
- **Find Slot -** This is the order in which the calls get parked. You can choose to use the first available position or use the next position based on the last space position used.
- **Return to Originator -** If this is enabled, after the parked call times out it will return to the device that parked the call. If it is disabled, the call will be taken to the **Timeout Destination**.
- **Announce Space Number -** If enabled, the system will announce the parking position in which the call was parked.
- **Record -** If enabled, this will record the parked call.

To use the call parking you have a couple of options. If you use the feature code ***4** this will park the call using the **Default Parking**. The parking position used will be based on the Find Slot option you chose. If you have Announce Space Number enabled, you will hear the parking position in which the call was parked. You can then call this parking space position from any device to retrieve the parked call.

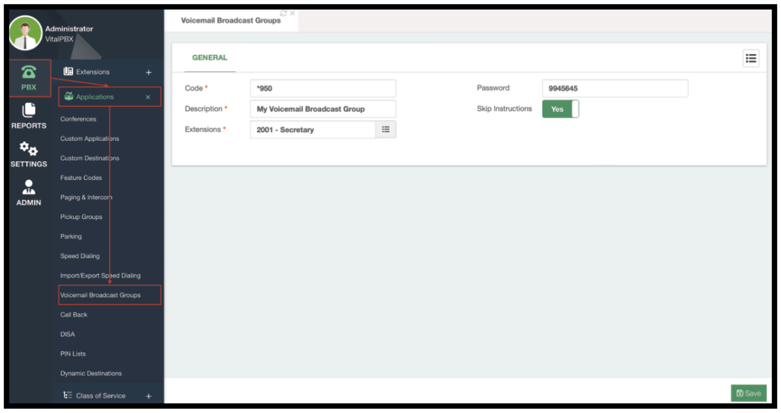
You can also transfer calls to a specific parking position. For example, you can blind transfer a call to parking position 705. Now, anyone can dial 705 to retrieve the parked call.

Parking positions can be set to BLF keys with devices that support monitoring parking spaces. This way, you can monitor the parking space from your device and retrieve parked calls in that space with a single button press. This same BLF key can be used to park calls as well.

Call Parking is a nice way to set an ongoing call aside and retrieve it from any device. This is similar to how calls were placed on hold in traditional PBX systems, and anyone on the same server can retrieve the call.

Section 8, Lesson 11 - Voicemail Broadcast Groups

A simple way to leave a message to multiple extensions is using **Voicemail Broadcast Groups**. With voicemail broadcast groups you can send a voicemail message to the voicemail box of multiple extensions. To create a Voicemail Broadcast group you must go to **PBX > Applications > Voicemail Broadcast Groups**.



Code * *950

Description * My Voicemail Broadcast Group

Extensions * 2001 - Secretary

Password 9945645

Skip Instructions Yes

Image 8.11.1 - Voicemail Broadcast Groups module.

First, you need to establish the **Code** to reach this Voicemail Broadcast Group. The code can be any number and star (*) symbol. Next, we add a **Description** to identify this group. Lastly, we select the **Extensions** to include in this group. Only extensions with voicemail enabled are eligible to be part of Voicemail Broadcast Groups. Additionally, you can set a **Password** to this group so you need to enter a password when dialing the code. Finally, you can enable or disable **Skip Instructions**, to skip the voice prompts for this feature and leave a voicemail message immediately when calling this Voicemail Broadcast Group.

Now, you can **Save** and **Apply Changes**. Whenever you call the code for this voicemail broadcast group, you will be able to leave a voicemail message to multiple extensions at once.

Conclusion

With all of these applications, you can make a more advanced use of your VitalPBX installation. This adds to the everyday use of VitalPBX, other than placing and receiving calls. You can now page extensions, leave voicemail messages to multiple extensions, place a call to a DISA to use internal features from an external call, park your calls, and much more. Making your experience richer and fully featured.

Now, this is not everything that VitalPBX can do. There are many more features we will continue to explore throughout this guide, especially with our add-ons.

Let's Continue exploring everything that VitalPBX can offer.

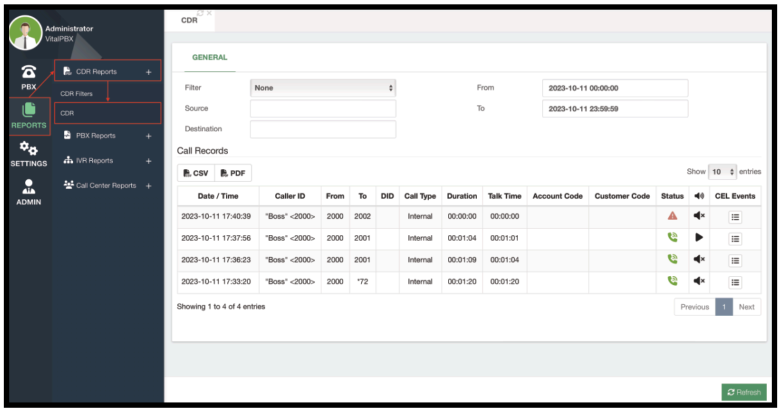
SECTION 9 - REPORTS

Introduction

VitalPBX offers various reports you can use to monitor your different calls and devices. Let's explore the CDR reports for a detailed view of your calls, and how to filter them. As well as various PBX reports to see the status of your devices and trunks.

Section 9, Lesson 1 - CDR Reports

The CDR or Call Detail Records/Reporting allows you to see the full history of calls in your VitalPBX installation. This way you can track calls to the time and second they occurred. To go to the CDR reports we must go to **Reports > CDR Reports > CDR**.



Filter: None

Source:

Destination:

From: 2023-10-11 00:00:00

To: 2023-10-11 23:59:59

Call Records

Show 10 entries

Date / Time	Caller ID	From	To	DID	Call Type	Duration	Talk Time	Account Code	Customer Code	Status	🔊	CEL Events
2023-10-11 17:40:39	"Boss" <2000>	2000	2002		Internal	00:00:00	00:00:00			⚠	🔇	☰
2023-10-11 17:37:56	"Boss" <2000>	2000	2001		Internal	00:01:04	00:01:01			☎	▶	☰
2023-10-11 17:36:23	"Boss" <2000>	2000	2001		Internal	00:01:09	00:01:04			☎	🔇	☰
2023-10-11 17:33:20	"Boss" <2000>	2000	*72		Internal	00:01:20	00:01:20			☎	🔇	☰

Showing 1 to 4 of 4 entries

Image 9.1.1 - CDR Reports module.

As mentioned previously, all calls placed and received will be displayed here. In the top section, you can filter out the calls for a specific search. We will see how to create **advanced filters** in the next lesson. You can quickly filter by the **Source** and **Destination**, as well as a **Date Range** by establishing the **From** and **To** date range. You can then click the green **Refresh** button in the lower right-hand corner to apply the filters.

Next, you have the **Call Records** table. You can choose how many **entries to see per page**. This table shows you the **Date and Time** the call was placed/received, the **Caller ID** information, **From** where the call was placed, **To** where the call was made, the **DID** number called, the **Call Type**, the **Duration** of the call from the moment it started ringing, the **Talk Time** from when the call was answered, the **Account Code** if it was used, the **Customer Code** if it was used, the **Status**, and the **Call Recording.**

If you have a Starter License or are subscribed to any of our Licensing Plans, you will have the **CEL Events** extended feature. We will see more about CEL events, and how to enable and configure them later.

This table of records can be **exported in CSV or PDF format**. You can then process this information using a number processing application.

You can listen to a call recording by clicking on the **Play icon** in the call recording column. This will open a modal with the call recording and will show a download button to download the call recording file.

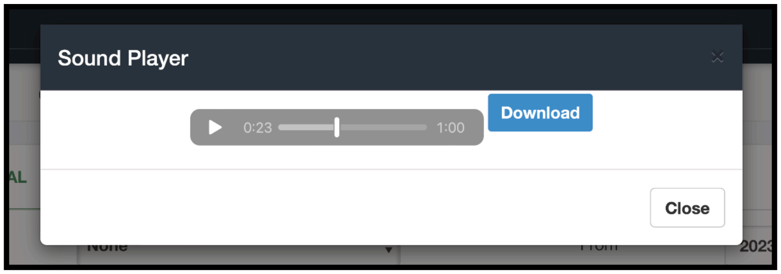
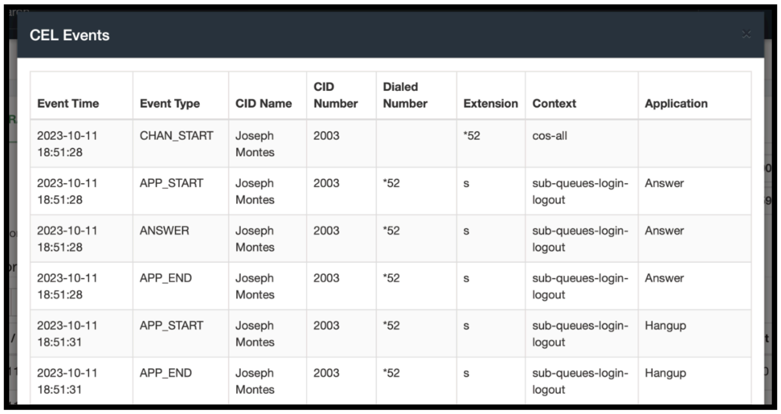


Image 9.1.2 - Call Recording playback from a CDR record.

This way you can listen to call recordings directly and download the files. For a more advanced way to manage and download call recordings, please refer to the Sonata Recordings lessons and manual.


CEL Events

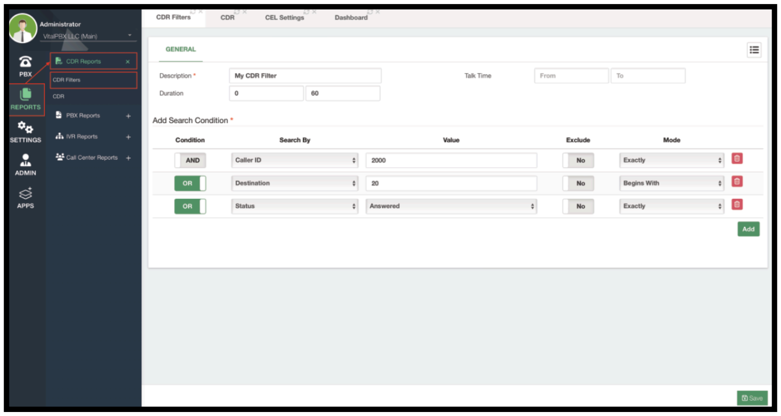
Event Time	Event Type	CID Name	CID Number	Dialed Number	Extension	Context	Application
2023-10-11 18:51:28	CHAN_START	Joseph Montes	2003		*52	cos-all	
2023-10-11 18:51:28	APP_START	Joseph Montes	2003	*52	s	sub-queues-login-logout	Answer
2023-10-11 18:51:28	ANSWER	Joseph Montes	2003	*52	s	sub-queues-login-logout	Answer
2023-10-11 18:51:28	APP_END	Joseph Montes	2003	*52	s	sub-queues-login-logout	Answer
2023-10-11 18:51:31	APP_START	Joseph Montes	2003	*52	s	sub-queues-login-logout	Hangup
2023-10-11 18:51:31	APP_END	Joseph Montes	2003	*52	s	sub-queues-login-logout	Hangup

Image 9.1.3 - CEL Events in the CDR report.

When you have the **CEL Events** extended feature, you can click on the **hamburger (three lines) icon** under the CEL Events column for a specific call. CEL Events can show you the trajectory of a call and allow you to monitor and troubleshoot how a call behaves.
CDR is a useful and powerful tool to monitor the call history for the whole VitalPBX. Now let's look at how we can filter the CDR records in a more advanced way.

Section 9, Lesson 2 - CDR Filters

Our CDR records can become massive over time, and we might want to filter our results for a more specific set of calls. To do this, we can create **CDR filters** we can use over and over with our CDR reports. To create our CDR filters we must go to **Reports > CDR Reports > CDR Filters**.



Description * My CDR Filter

Talk Time From To

Duration 0 60

Add Search Condition *

Condition	Search By	Value	Exclude	Mode
AND	Caller ID	2000	No	Exactly
OR	Destination	20	No	Begins With
OR	Status	Answered	No	Exactly

Image 9.2.1 - CDR Filters module.

First, we will need to enter a **Description** to identify the CDR filter. Next, we can add **Duration** and **Talk Time** Filters. This will limit the search to calls longer than a certain amount of seconds, but shorter than a certain amount of seconds. Duration is a call from the moment it starts ringing, and Talk Time is a call from the moment it is answered.

Afterward, we have **Search Conditions**. These are the heart and soul of our CDR filters. With these search conditions, we can make our CDR filters as specific as possible.

Search Conditions are comprised of the following aspects.

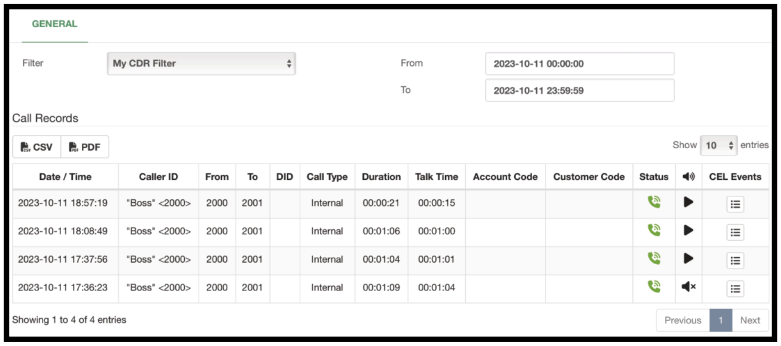
- **Condition -** This is the type of condition you are adding. There are two options, AND or OR. AND means that the condition has to be met for the filter to apply, and OR means that this is an optional condition. If you have experience with MySQL-type searches, these condition types will make more sense as these filters are basically a MySQL search.
- **Search By -** This is the field that this condition will consider in the search. You can choose between the following options.
  - **Caller ID -** This will use the Caller ID information field.
  - **Source -** This will use the source field. The source is who originated the call.
  - **Destination -** This will use the destination field. The destination is the number dialed.
  - **DID -** This will use the DID field. The DID is the DID number dialed. This is only for incoming calls.

    - **Account Code -** This will use the Account Code field. An account code is created in Extensions and helps filter by group of extensions using the same account code.
    - **Customer Code -** This will use the Customer Code field. A customer code is created in the customer code module. This is used to filter by a common customer type for easier searching.
    - **Status -** This will use the Status field. The status is the outcome status of a call. This can be Answered, No Answer, Busy, or Failed.
    - **Call Type -** This will use the Call Type field. The call type is the type of call that was made. This can be Internal, for calls between extensions, Incoming, for calls received through a trunk, Outgoing, for outgoing calls made through a trunk, and Transit, for incoming calls that came through a trunk and were routed to another trunk.
- **Value -** This is the value used for the Search By option.
- **Exclude -** This will determine if the condition will be included or excluded in the search.
- **Mode -** This is how the value is considered. You can consider the value Exactly as it is, Beginning with, is Contained within, or Ending with. Status and Call Type can only have Exact modes.

You can create as many Search Conditions as you need to make the CDR filter as specific as you want by clicking the green **Add** button in the right-hand corner. Remove them by clicking the red **Trash** button next to the condition. You can then proceed to **Save**.

For example, we can make a filter that searches for calls made by extension 2000, by setting a condition that uses the **Source of 2000** and is matched **exactly**. Then we can have another condition saying that calls from extension 2000 were to other extensions, by setting an **AND condition** with the **Destination** set to **Beginning with 20**, as our extensions begin with 20. And we only want to see Answered calls so we add an **AND condition** with the **Status** set to **Answered**.

We can apply our CDR filter in the CDR reports at **Reports > CDR Reports > CDR**, and click **Refresh**.



Filter: My CDR Filter
From: 2023-10-11 00:00:00
To: 2023-10-11 23:59:59

Call Records

Show 10 entries

Date / Time	Caller ID	From	To	DID	Call Type	Duration	Talk Time	Account Code	Customer Code	Status	🔊	CEL Events
2023-10-11 18:57:19	"Boss" <2000>	2000	2001		Internal	00:00:21	00:00:15			☎	▶	☰
2023-10-11 18:08:49	"Boss" <2000>	2000	2001		Internal	00:01:06	00:01:00			☎	▶	☰
2023-10-11 17:37:56	"Boss" <2000>	2000	2001		Internal	00:01:04	00:01:01			☎	▶	☰
2023-10-11 17:36:23	"Boss" <2000>	2000	2001		Internal	00:01:09	00:01:04			☎	🔇	☰

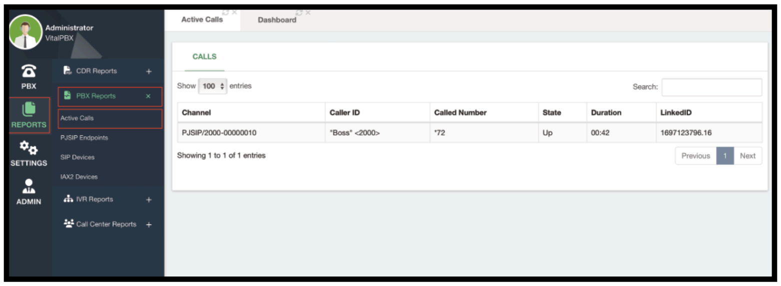
Showing 1 to 4 of 4 entries

Image 9.2.2 - CDR report with CDR Filter applied.

As you can see, the CDR report will show calls that match our filter.

Section 9, Lesson 3 - PBX Reports

Next, we will see PBX reports. These are a set of reports that will help you monitor the registration of devices and trunks, as well as active calls in your VitalPBX in real-time. Let's start with **Active Calls** by going to **Reports > PBX Reports > Active Calls**.



Channel	Caller ID	Called Number	State	Duration	LinkedID
PJSIP/2000-00000010	"Boss" <2000>	*72	Up	00:42	1697123796.16

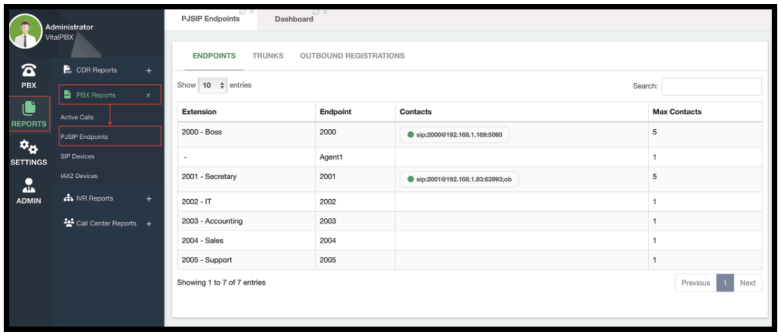
Showing 1 to 1 of 1 entries

Image 9.3.1 - Active Calls module.

Here you will see all of the active calls in your VitalPBX at that moment in real-time. If you place a call to another extension or a feature code, you will see the call appear and display information about that particular code.

Next, we will see modules specific to **PJSIP, SIP, and IAX technologies**. Within these modules, you will see information about the **extensions**, **trunks**, and **outbound registrations** in your VitalPBX.

We will only see the PJSIP module as this is basically the same for SIP and IAX devices. So we go to **Reports** > **PBX Reports** > **PJSIP Endpoints**.



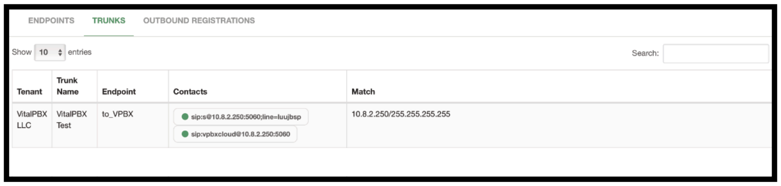
Extension	Endpoint	Contacts	Max Contacts
2000 - Boss	2000	● sip:2000@192.168.1.169:5060	5
-	Agent1		1
2001 - Secretary	2001	● sip:2001@192.168.1.83:63993;ob	5
2002 - IT	2002		1
2003 - Accounting	2003		1
2004 - Sales	2004		1
2005 - Support	2005		1

Showing 1 to 7 of 7 entries

Image 9.3.2 - PJSIP Endpoints module's Endpoints tab.

The module will have three tabs. In the case of PJSIP, we first have the **Endpoints** tab these are the devices created for our extensions. You can see the extension number they are associated with, as well as the registration status for the devices. Since PJSIP can have multiple registrations for a single device, you may see multiple contacts in the contacts column. What is important to see here is that contacts have a **status indicator**. This indicator will shift from red to green depending on the **registration status**. All of this is updated in real-time. This way, you are able to monitor the registration of your devices. So if you turn off or de-register a device you will see it turn red or disappear from the table in real time.

The next tab is the **Trunks** tab.

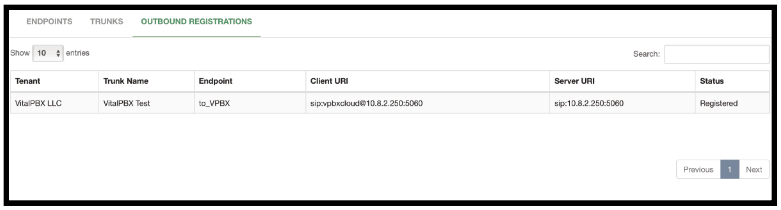


Tenant	Trunk Name	Endpoint	Contacts	Match
VitalPBX LLC	VitalPBX Test	to_VPBX	● sip:s@10.8.2.250:5060;line=luujbsp ● sip:vpbxcloud@10.8.2.250:5060	10.8.2.250/255.255.255.255

Image 9.3.3 - PJSIP Endpoints module's Trunks tab.

In the trunks tab, you can see your trunks and the registered devices. Similarly to the extensions, you have a status indicator to see if your trunk is registered successfully. The green indicator would mean that the trunk is registered successfully.

Trunks can be extended to the **Outbound Registrations** tab if you are using registration authentication with your trunks. This means you have a username and secret to register your trunks. This tab will give you additional information about your registration status.



Tenant	Trunk Name	Endpoint	Client URI	Server URI	Status
VitalPBX LLC	VitalPBX Test	to_VPBX	sip:vpbxcloud@10.8.2.250:5060	sip:10.8.2.250:5060	Registered

Image 9.3.4 - PJSIP Endpoints module's Outbound Registrations tab.

As mentioned before, this is the same idea for the SIP and IAX devices reports. With these reports, you can monitor in real-time the registration status of your extensions and trunks.

Conclusion

These are powerful and useful reports you can use to monitor the activity in your VitalPBX and have a better insight into the different devices and calls your users have. Additional reports are added when you add the Queues Callback and IVR stats add-on modules. To see these in action, please refer to the IVR Stats and Queues Callback section and lessons. The different reports VitalPBX offers make it easy for you to know how your VitalPBX is standing.

For Call Center Queue and Agent reports and for reports based on the call recordings in your VitalPBX, as well as real-time call monitoring for your different agents, please refer to the Sonata Stats, Sonata Recording, and Sonata Switchboard manuals, guides, and lessons.

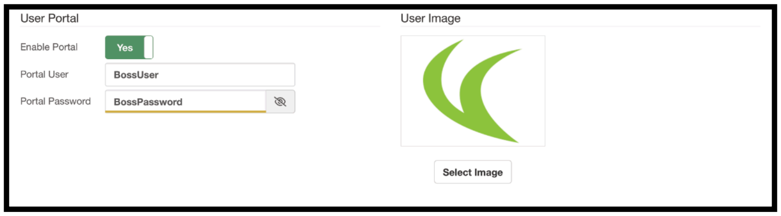
SECTION 10 - USER PORTAL

Introduction

End users also have the ability to have a **User Portal** where they can manage their extensions and use different features on the VitalPBX user interface. This portal is a graphical way for end-users to monitor and manage their Voicemails, Call History, and different diversions for their extension.

Section 10, Lesson 1 - User Portal

To use a **User Portal**, you first need to enable it on the extension for the end-user. This can be done under **PBX > Extensions > Extension > Advanced**. Here, you will find the User Portal section at the bottom of the Advanced tab.


User Portal

Enable Portal: Yes

Portal User: BossUser

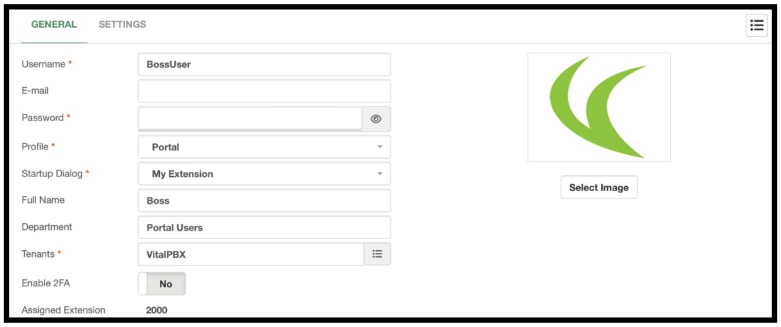
Portal Password: BossPassword

User Image

Image 10.1.1 - User Portal section in an Extension.

First, you need to set **Enable Portal** to **Yes**. Then, add a **Portal User and Password**. Optionally, you can add a **User Image** to identify the user. This image must be a square image in PNG or JPEG format. Then you can **Save/Update** and **Apply Changes**.

By enabling the User Portal and adding a Portal User and Password, the user will also appear in the Users module under **Admin > Admin > Users**. They will have the **Portal** User Profile assigned.



Username *: BossUser

E-mail:

Password *:

Profile *: Portal

Startup Dialog *: My Extension

Full Name: Boss

Department: Portal Users

Tenants *: VitalPBX

Enable 2FA: No

Assigned Extension: 2000

Image 10.1.2 - Portal user in the Users module.

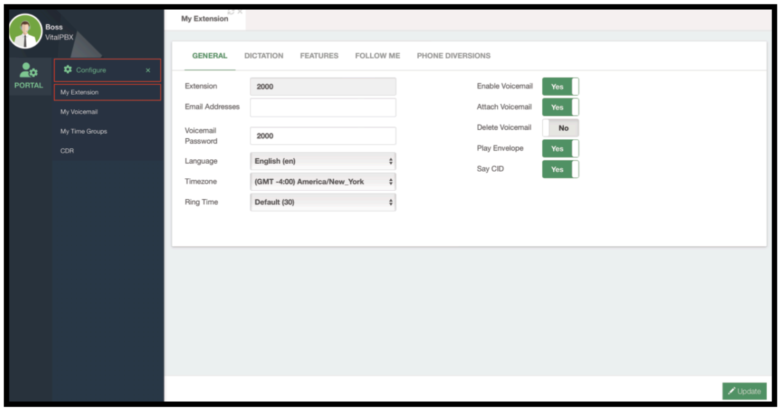
Here, you can add additional features or modifications by making changes to the User Profile and changing the settings of the user if needed.

We can now log out as the Administrator and Log back in using the Portal User and Password we created.

Once you log in with the Portal User and Password, you will be greeted with the **My Extension** module. This is located under **Portal > Configure > My Extension**. The My Extension module is where you can perform various actions over your assigned extension as an end-user portal user.

First, you have the **General Tab**. Here you can change the following options.

- **Email Address -** This is the email address associated with the portal user's extension for the different notifications sent by the system.
- **Voicemail Password -** This is the password used to access the portal user's voicemail box using the *97 feature code.
- **Language -** This is the language used for the different voice prompts for the various features in VitalPBX.
- **Timezone -** This is the time zone associated with the voicemails and CDR call history for this portal user.
- **Ring Time -** This is the ring time for the portal user's extension before going to voicemail or hanging up.
- **Enable Voicemail** - This feature will enable or disable voicemail for this extension.
- **Attach Voicemail -** This is whether or not the extension will attach the voicemail recording file with the voicemail to email feature.
- **Delete Voicemail -** This will delete the voicemail recording file after the voicemail is sent using the voicemail-to-email feature.
- **Play Envelope -** This is whether or not the system will add the voicemail information at the beginning of the voicemail recording file.
- **Say CID -** This is whether or not the system will add the Caller ID information to the voicemail recording file.

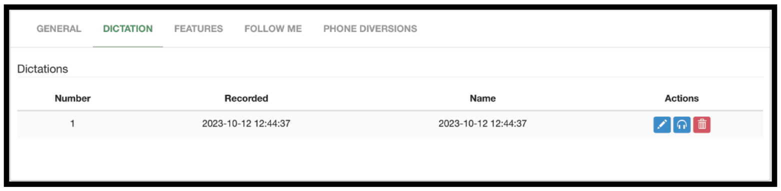


Extension: 2000
Email Addresses:
Voicemail Password: 2000
Language: English (en)
Timezone: (GMT -4:00) America/New_York
Ring Time: Default (30)
Enable Voicemail: Yes
Attach Voicemail: Yes
Delete Voicemail: No
Play Envelope: Yes
Say CID: Yes

Image 10.1.3 - User Portal's My Extension module.

After any changes to these configuration options, you can press the green **Update** button in the lower right-hand corner.

Next, we have the **Dictation Tab**. The dictation tab will have a graphical way to view any dictations recorded by the portal user. Remember, for this to work, you need to enable dictation to the portal user's extension. Then, whenever the portal user records a dictation using the feature code ***93** these dictation recordings can be played back from the Dictation Tab.



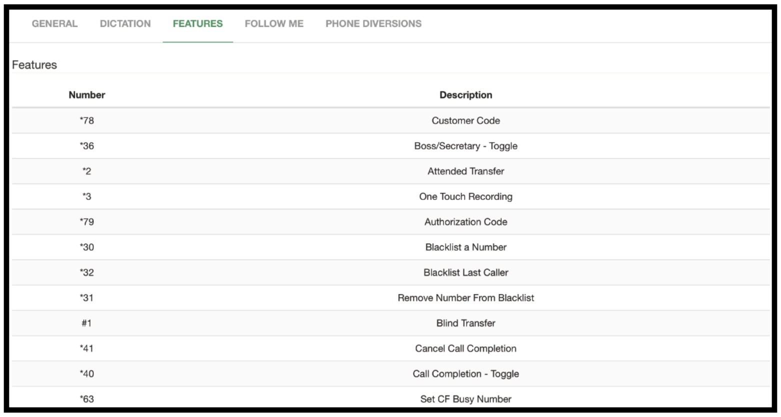
Dictations

Number	Recorded	Name	Actions
1	2023-10-12 12:44:37	2023-10-12 12:44:37	✎ 🎧 🗑

Image 10.1.4 - My Extension’s Dictation tab.

The **Dictations** table will show you the dictation’s **number**, the date when it was **recorded**, and the dictation **name**. In the **Actions** column, you can **edit** the file name, **playback** the dictation, and delete the dictation.

Remember, dictations are voice notes you can record, so you can listen back to them at any time from your User Portal.



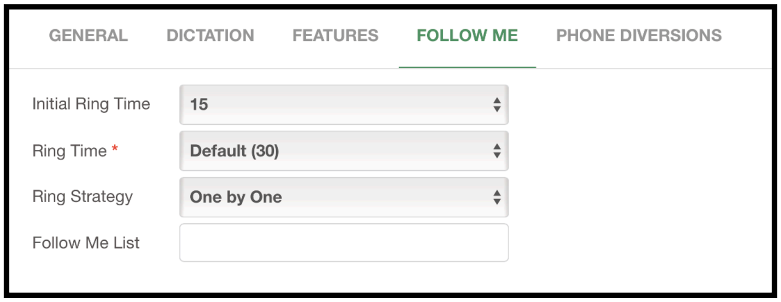
Features

Number	Description
*78	Customer Code
*36	Boss/Secretary - Toggle
*2	Attended Transfer
*3	One Touch Recording
*79	Authorization Code
*30	Blacklist a Number
*32	Blacklist Last Caller
*31	Remove Number From Blacklist
#1	Blind Transfer
*41	Cancel Call Completion
*40	Call Completion - Toggle
*63	Set CF Busy Number

Image 10.1.5 - My Extension’s Features tab.

Next, we have the **Features Tab**. Here portal users can see the different **feature codes** they can dial from their extensions.

This is a simple cheat sheet they can use to know what a feature code does.



Initial Ring Time	15
Ring Time *	Default (30)
Ring Strategy	One by One
Follow Me List	

Image 10.1.6 - My Extension’s Follow Me tab.

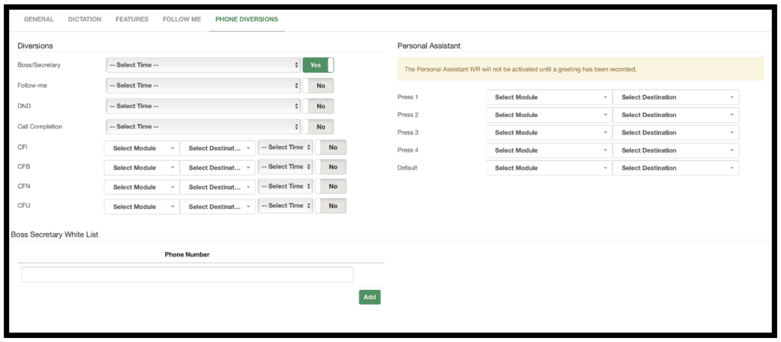
Afterward, we have the **Follow Me Tab**. Here, the portal user can configure their **Follow Me Settings**.

The settings include the following options.

- **Initial Ring Time -** This is the time in seconds the extension devices will ring before going to the Follow Me List.
- **Ring Time -** This is the time in seconds the module will call the numbers in the Follow Me List.
- **Ring Strategy -** This is the strategy the module will use to call the numbers in the Follow Me List. This can be Ring All or One-by-One.
- **Follow Me List -** This is the list of numbers the module will contact if the extension does not answer within the Initial Ring Time.

The end-user is able to enable Follow Me using the feature code ***67** afterward.

Finally, there's the **Phone Diversions** Tab. Here the portal user is able to manage their phone diversions.



Diversions

Boss/Secretary	-- Select Time --			Yes
Follow-me	-- Select Time --			No
DND	-- Select Time --			No
Call Completion	-- Select Time --			No
CFI	Select Module	Select Destinat...	-- Select Time	No
CFB	Select Module	Select Destinat...	-- Select Time	No
CFN	Select Module	Select Destinat...	-- Select Time	No
CFU	Select Module	Select Destinat...	-- Select Time	No

Personal Assistant

The Personal Assistant IVR will not be activated until a greeting has been recorded.

Press 1	Select Module	Select Destination
Press 2	Select Module	Select Destination
Press 3	Select Module	Select Destination
Press 4	Select Module	Select Destination
Default	Select Module	Select Destination

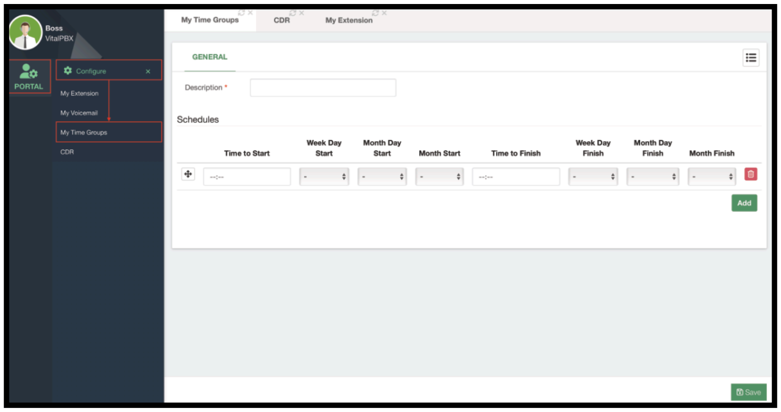
Boss Secretary White List

Phone Number

Image 10.1.7 - My Extension's Phone Diversions tab.

Similar to what we saw in the Extension Status module, here the end-user is able to manage the different diversions. They can **Enable and Disable** their Boss/Secretary, Follow Me, Do Not Disturb, Call Completion, and Multiple Call Forwards.

Portal users can create their own **Time Groups** by going to **Portal > Configure > My Time Groups**. These are created in the same way we did as the super administrator.



Description *

Schedules

Time to Start	Week Day Start	Month Day Start	Month Start	Time to Finish	Week Day Finish	Month Day Finish	Month Finish
--:--	-	-	-	--:--	-	-	-

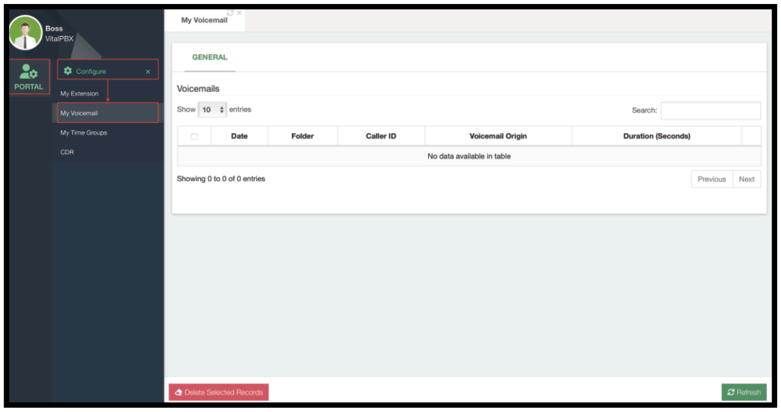
Image 10.1.8 - My Time Groups module.

The portal user can then associate their time groups with their diversions to automate their activation.

If the portal user is using the Boss/Secretary feature, they are able to add a white list in the Phone Diversions tab as well. This way, the numbers added to this white list will skip the Boss/Secretary and be able to reach the extension directly. You can add as many numbers as you need by clicking the green **Add** button.

Lastly, you have the Personal Assistant configurations. We will see this in more detail in the next lesson.

Continuing with the User Portal, the next module is My Voicemail under **Portal > Configure > My Voicemail**. This is a visual way for the end-user to see their voicemail box.



Voicemails

Show 10 entries Search:

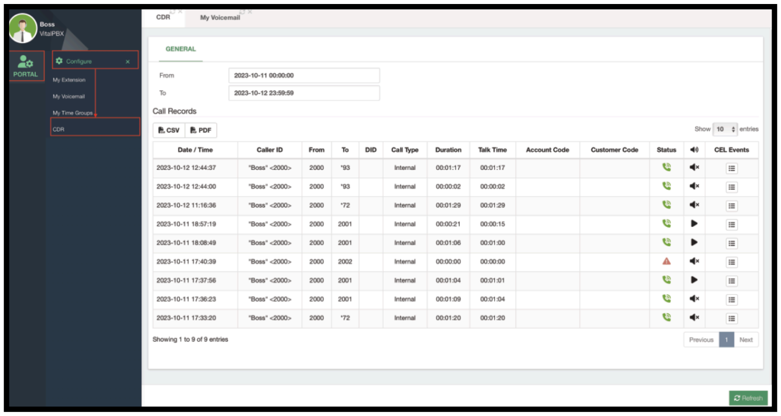
□	Date	Folder	Caller ID	Voicemail Origin	Duration (Seconds)
No data available in table					

Showing 0 to 0 of 0 entries

Image 10.1.9 - My Voicemail module.

The end-user can see detailed information about the voicemail messages, play them back, and delete multiple records.

Finally, with the User Portal, end-users can see their call history under **Portal > Configure > CDR**.



From 2023-10-11 00:00:00

To 2023-10-12 23:59:59

Call Records

Show 10 entries

Date / Time	Caller ID	From	To	DID	Call Type	Duration	Talk Time	Account Code	Customer Code	Status	🔊	CEL Events
2023-10-12 12:44:37	"Boss" <2000>	2000	*93		Internal	00:01:17	00:01:17			☎	🔇	☰
2023-10-12 12:44:00	"Boss" <2000>	2000	*93		Internal	00:00:02	00:00:02			☎	🔇	☰
2023-10-12 11:16:36	"Boss" <2000>	2000	*72		Internal	00:01:29	00:01:29			☎	🔇	☰
2023-10-11 18:57:19	"Boss" <2000>	2000	2001		Internal	00:00:21	00:00:15			☎	▶	☰
2023-10-11 18:08:49	"Boss" <2000>	2000	2001		Internal	00:01:06	00:01:00			☎	▶	☰
2023-10-11 17:40:39	"Boss" <2000>	2000	2002		Internal	00:00:00	00:00:00			⚠	🔇	☰
2023-10-11 17:37:56	"Boss" <2000>	2000	2001		Internal	00:01:04	00:01:01			☎	▶	☰
2023-10-11 17:36:23	"Boss" <2000>	2000	2001		Internal	00:01:09	00:01:04			☎	🔇	☰
2023-10-11 17:33:20	"Boss" <2000>	2000	*72		Internal	00:01:20	00:01:20			☎	🔇	☰

Showing 1 to 9 of 9 entries

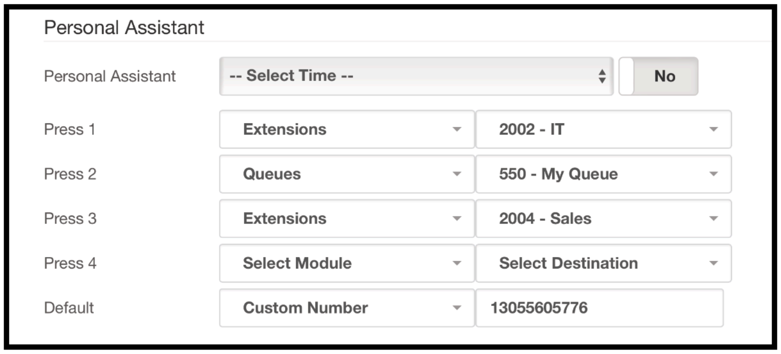
Image 10.1.10 - User Portal's CDR module.

Here, the portal user is able to see their full call history and listen to their call recordings.

Section 10, Lesson 2 - Personal Assistant

End-users have access to a feature called **Personal Assistant**. The personal assistant feature allows end-users to have a dedicated IVR menu. This allows them to record a sound file with their menu options so when this feature is active callers will have options to press to reach destinations they set.

To use the Personal Assistant, the end user first needs to record the sound file that will playback, this is called the **Personal Assistant Message**. They can do this by dialing the ***94** feature code. Once they have recorded a personal assistant message, they can head over to **Portal > Configure > My Extension > Phone Diversions**.



Personal Assistant

Personal Assistant	-- Select Time --	No
Press 1	Extensions	2002 - IT
Press 2	Queues	550 - My Queue
Press 3	Extensions	2004 - Sales
Press 4	Select Module	Select Destination
Default	Custom Number	13055605776

Image 10.2.1 - Personal Assistant in the User Portal.

The portal user can now configure the personal assistant options. The personal assistant feature can have up to 4 options callers can press. If they don't enter any option, callers will be taken to the Default option.

The personal assistant can also have a **Time Group** assigned to automate the activation. End-users can also activate the Personal Assistant feature with the ***96** feature code.

Once the personal assistant is activated if you call the extension, you will be greeted by the personal assistant message. You can press any option and be taken to the assigned destination.

Conclusion

The User Portal is an easy way for end-users to manage their extension and the different features assigned to it. Remember that additional options can be added to users by making changes in the Users module and the portal user profile.

End-users can have full control over their extensions and this is a good way for them to manipulate their diversions. The personal assistant is a nice way for them to give their callers options to reach popular destinations if the end-user leaves their extension.

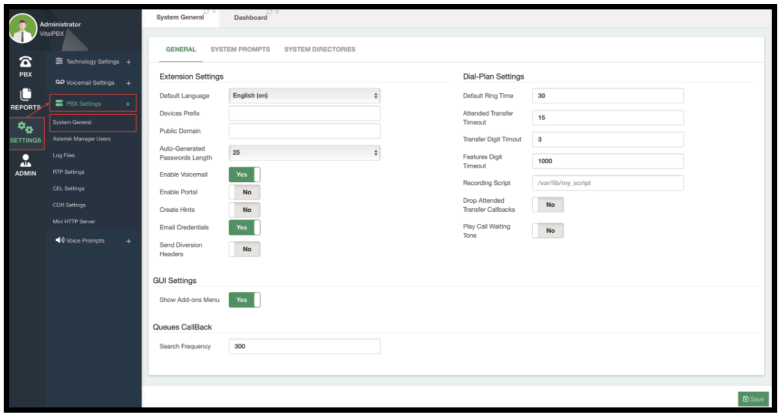
SECTION 11 - GENERAL PBX CONFIGURATIONS

Introduction

In this next section, we will see some general configurations you can do with your VitalPBX. These are more advanced configurations and their location so you can add a more personal aspect to your VitalPBX installation.

Section 11, Lesson 1 - System General

There are various default configurations you can set for your VitalPBX. These are general system configurations. To set these you must go to **Settings > PBX Settings > System General**.



Extension Settings

Default Language: English (en)
Devices Prefix:
Public Domain:
Auto-Generated Passwords Length: 25
Enable Voicemail: Yes
Enable Portal: No
Create Hints: No
Email Credentials: Yes
Send Diversion Headers: No

Dial-Plan Settings

Default Ring Time: 30
Attended Transfer Timeout: 15
Transfer Digit Timout: 3
Features Digit Timeout: 1000
Recording Script: /var/lib/my_script
Drop Attended Transfer Callbacks: No
Play Call Waiting Tone: No

GUI Settings

Show Add-ons Menu: Yes

Queues CallBack

Search Frequency: 300

Image 11.1.1 - System General module's General tab.

First, we have the **General** tab. Here we find the default **Extension, Dial Plan, GUI, and Queues Callback Settings**.

For Extensions, we can set the following default settings.

- **Default Language -** This is the voice prompt language used for new extensions.
- **Devices Prefix -** This is a prefix automatically added to device users.
- **Public Domain -** This is the FQDN (Fully Qualified Domain Name) used for various URL settings. This will overwrite the hostname set for this installation.
- **Auto-Generated Passwords Length -** This is the default length for auto-generated passwords.
- **Enable Voicemail -** This is whether or not voicemail is enabled by default for extensions.
- **Enable Portal -** This is whether or not the User Portal is enabled by default.
- **Create Hints -** This is whether or not Hints are generated for new extensions. Keep in mind that enabling hints for all extensions can take a toll on system performance.
- **Email Credentials -** If enabled, this will send the extension welcome email to new extensions.
- **Send Diversion Headers -** If enabled, additional call diversion headers will be sent such as the different Call Forwards. Some SBCs might complain if these are enabled.

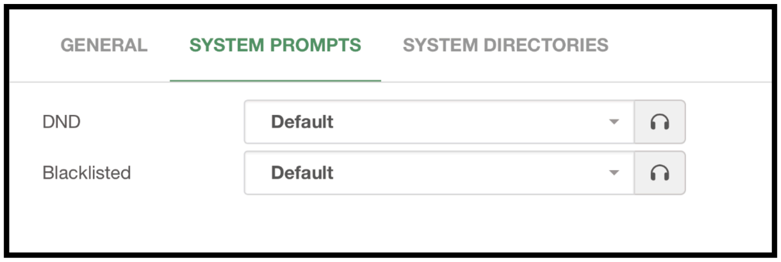
For the Dial Plan Settings, we can establish the following default settings.

- **Default Ring Time -** This is the default time in seconds an extension will ring before going to voicemail or hangup.
- **Attended Transfer Timeout -** This is the default time in seconds where a call is placed after the last digit is dialed for an attended transfer.
- **Transfer Digit Timeout -** This is the default time in seconds between digits pressed during a call transfer.
- **Features Digit Timeout -** This is the default time in milliseconds between digits for a feature during a call.
- **Recording Script -** This is a custom script that can be run after a call is recorded.
- **Drop Attended Transfer Callbacks -** When set to No, Asterisk will call the transfer initiator back after an early hangup if the transferred party does not answer the attended transfer.
- **Play Call Waiting Tone -** When set to Yes a tone will play when an extension receives a new call while on an existing call.

Finally, we have the **Show Add-Ons Menu** for the GUI Settings, which will hide direct access to the Sonata Suite and VitXi from the login screen if it is set to No. And if you have the **Queues Callback** module installed, you are able to set the **Search Frequency** which is how often the Queues Callback will search for queued calls for a callback.

If you make any changes in this module, you can then **Save** and **Apply Changes**.

Next, we have the System Prompts tab. Here, you can change the voice prompts used when you have **Do Not Disturb** turned on, and the prompt that plays back when a **blacklisted** number calls your VitalPBX and you didn't set a destination for blacklisted calls.



DND: Default

Blacklisted: Default

Image 11.1.2 - System Prompts tab in System General.

Once you select the sound recordings for these voice prompts, you can **Save** and **Apply Changes**.

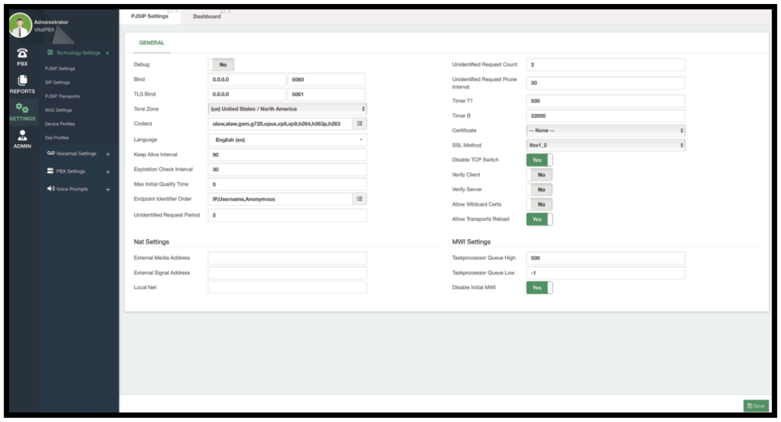
Lastly, we have the **System Directories** tab, which will show you the directories for popular Asterisk locations.

- **Asterisk AGI Directory -** /var/lib/asterisk/agi-bin
- **Asterisk Directory -** /etc/asterisk
- **Asterisk Log Directory -** /var/log/asterisk
- **Asterisk Modules Directory -** /usr/lib/asterisk/modules
- **Asterisk Sound Directory -** /var/lib/asterisk/sounds
- **Asterisk Spool Directory -** /var/spool/asterisk
- **Asterisk Libraries Directory -** /var/lib/asterisk

Section 11, Lesson 2 - PJSIP Settings

Now let's take a look into various advanced settings we can configure for the technologies VitalPBX manages. Note that we are not covering every single option throughout this guide as these get more specific to the technology and go out of the scope of this guide. This is more for you to know where you will find these settings should you need to modify them. To learn more about these settings and what they are used for, we recommend you look into Asterisk's official documents about the topic and documentation dedicated to the technologies themselves. You can also use the **tooltips** for each option by hovering over a field title. Here we just make sure that they are available to you. We will first cover PJSIP Settings as this is the main technology for VitalPBX.

First, let's go to **Settings > Technology Settings > PJSIP Settings**. Here you will find general settings for the PJSIP protocol.



GENERAL

Debug: No
Bind: 0.0.0.0 5060
TLS Bind: 0.0.0.0 5061
Tone Zone: (us) United States / North America
Codecs: ulaw,alaw,gsm,g729,opus,vp8,vp9,h264,h263p,h263
Language: English (en)
Keep Alive Interval: 90
Expiration Check Interval: 30
Max Initial Qualify Time: 0
Endpoint Identifier Order: IP,Username,Anonymous
Unidentified Request Period: 5

Unidentified Request Count: 3
Unidentified Request Prune Interval: 30
Timer T1: 500
Timer B: 32000
Certificate: -- None --
SSL Method: tlsv1_2
Disable TCP Switch: Yes
Verify Client: No
Verify Server: No
Allow Wildcard Certs: No
Allow Transports Reload: Yes

Nat Settings

External Media Address:
External Signal Address:
Local Net:

MWI Settings

Taskprocessor Queue High: 500
Taskprocessor Queue Low: -1
Disable Initial MWI: Yes

Image 11.2.1 - PJSIP Settings module.

We visited this module when we looked into the NAT Settings. But aside from this, here you can define default **Codecs** PJSIP extensions and Trunks will use. You can also change the general **Language** that is used for voice prompts used with PJSIP endpoints. Here you can also change the **Bind** and **TLS Bind** ports. If you change these ports, you must also make the change in the *Firewall* so your devices can register with the new ports.

Other important aspects you will find here are the **Certificate** and **SSL Method**, but we will see more about this when we take a look into the security features with VitalPBX.

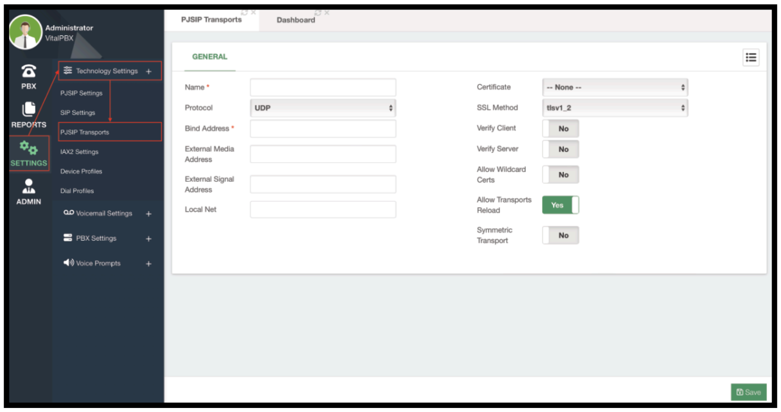
Most of the time, these features will be left with the default value, but if you make any changes here, you can **Save** and **Apply Changes**.

Now, let's look into **PJSIP Transports**. These can be created under **Settings > Technology Settings > PJSIP Transports**.

PJSIP Transports are usually used when the server in which VitalPBX is installed has **multiple network interfaces** and you have SIP/PJSIP Trunks connected through these interfaces. When you configure the *NAT settings*, the IP address used for the **SDP (Session Description Protocol)** is the one defined in the *NAT settings*. Thus your *Trunk Provider* will reject the calls because you are not sending the IP Address assigned to you for their service.

Here is where PJSIP Transports come into play so you can create a transport where *NAT settings* are not used and can then be assigned to a Trunk. This way, the trunk at the moment of creating the SDP uses the IP assigned to the interface connected with the Trunk Provider.

This is a very particular case, and it is uncommon with internet-based VoIP providers. Default Transports for UDP, TCP, WebRTC, and Microsoft© Teams™ trunks are created automatically, so there is no need to create these and they cannot be modified or deleted. So creating PJSIP Transports is rare unless you need it for your use case.



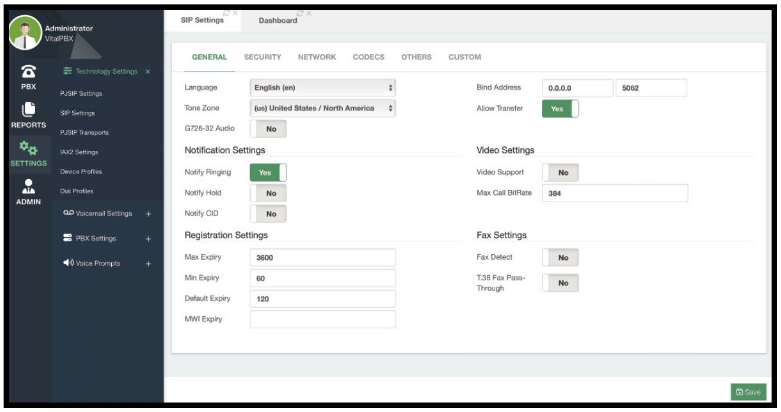
Name *
Protocol: UDP
Bind Address *
External Media Address
External Signal Address
Local Net
Certificate: -- None --
SSL Method: tlsv1_2
Verify Client: No
Verify Server: No
Allow Wildcard Certs: No
Allow Transports Reload: Yes
Symmetric Transport: No

Image 11.2.2 - PJSIP Transports module.

Section 11, Lesson 3 - SIP Settings (Legacy)

Next, we have the **SIP Settings** under **Settings > Technology Settings > SIP Settings**. As mentioned previously, SIP is no longer supported officially and has been deprecated by Asterisk. These settings are here solely to support migrations from VitalPBX 3 to VitalPBX 4. This is in the case you had SIP extensions or trunks created with your VitalPBX 3 backup and restored in a VitalPBX 4 installation. This way, your backup can restored, and we recommend you migrate your extensions and trunks to PJSIP as soon as possible. Our Extensions Import/Export can help with this process, but Trunks need to be reconfigured from the ground up.

This module is mostly ignored by newer installations that start with VitalPBX 4 and will be removed in future versions of VitalPBX as this technology will no longer be available. No new SIP Devices and Trunks can be created. Rest assured that PJSIP is easy to adopt as it is backward compatible with any SIP device and SIP Trunk.



Language: English (en)
Tone Zone: (us) United States / North America
G726-32 Audio: No
Bind Address: 0.0.0.0 5062
Allow Transfer: Yes

Notification Settings
Notify Ringing: Yes
Notify Hold: No
Notify CID: No

Video Settings
Video Support: No
Max Call BitRate: 384

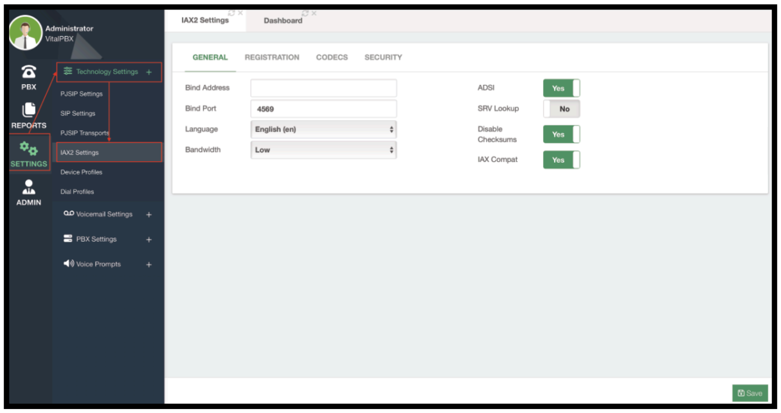
Registration Settings
Max Expiry: 3600
Min Expiry: 60
Default Expiry: 120
MWI Expiry

Fax Settings
Fax Detect: No
T.38 Fax Pass-Through: No

Image 11.3.1 - SIP Settings module.

Section 11, Lesson 4 - IAX2 Settings

Next, we have the IAX2 Settings under **Settings > Technology Settings > IAX2 Settings**. Here we can configure various options for IAX2. Again, most of the time these do not need to be changed, and in this guide, we explain only the most common to configure.

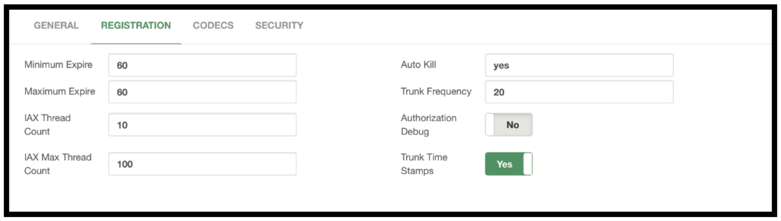


Bind Address		ADSI	Yes
Bind Port	4569	SRV Lookup	No
Language	English (en)	Disable Checksums	Yes
Bandwidth	Low	IAX Compat	Yes

Image 11.4.1 - IAX2 Settings General Tab.

In the **General Tab**, you can change the **Bind Port**, which is the port used to register IAX2 devices and trunks. If you change this port, you also need to change it in the *Firewall Services*. You can also change the default **Language** for the voice prompts that IAX2 devices use. Additionally, you can set the **Bandwidth** which will determine the codec to use from the selected codec list.

Next, we have the **Registration Tab.** This tab holds settings used for different timings and counts.



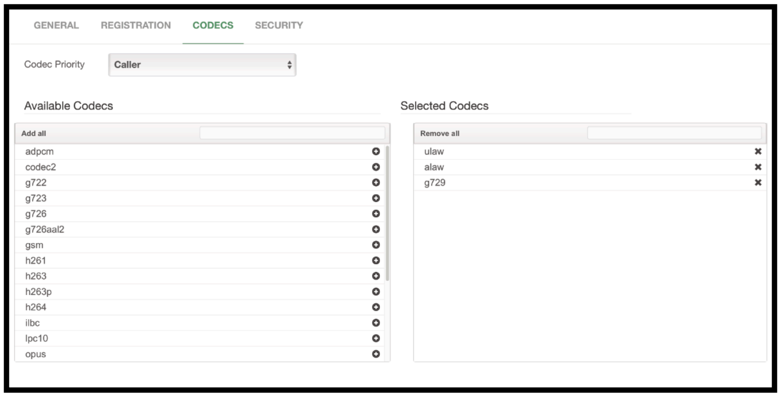
Minimum Expire	60	Auto Kill	yes
Maximum Expire	60	Trunk Frequency	20
IAX Thread Count	10	Authorization Debug	No
IAX Max Thread Count	100	Trunk Time Stamps	Yes

Image 11.4.2 - IAX2 Settings Registration Tab.

Afterward, there's the **Codecs Tab**. This tab allows you to choose who has the **Codec Priority**. This can be one of the following options.

- **Caller (Default) -** This considers the caller's preferred order over the host's.
- **Host -** This considers the hosts's preferred order over the caller's.
- **Disabled -** This disables the consideration of codecs altogether. This is the original behavior prior to the addition of preferences being added.
- **Reqonly -** This is the same as Disabled. The call is only accepted if the requested format is available.

Below, you can choose the codecs to be used with IAX2 devices and trunks. In the left column, you have the **Available Codecs**, and to the right the **Selected Codecs**.



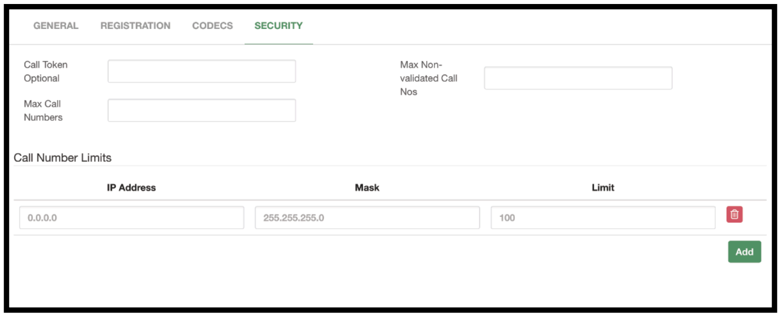
Codec Priority: Caller

Available Codecs
adpcm
codec2
g722
g723
g726
g726aal2
gsm
h261
h263
h263p
h264
ilbc
lpc10
opus

Selected Codecs
ulaw
alaw
g729

Image 11.4.3 - IAX2 Settings Codecs Tab.

Finally, we have the **Security Tab** for the IAX2 Settings. Here you will find various security parameters that you can add to IAX2 devices.



Call Number Limits

IP Address	Mask	Limit
0.0.0.0	255.255.255.0	100

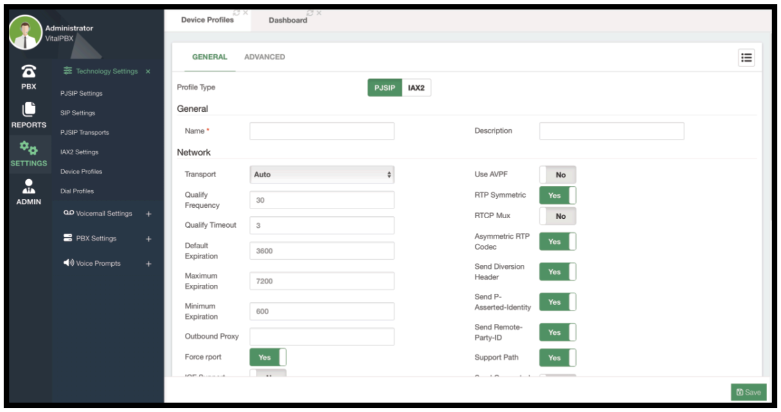
Image 11.4.4 - IAX2 Settings Security Tab.

Section 11, Lesson 5 - Device Profiles

Next, we have **Device Profiles.** Device profiles allow you to configure advanced settings for the different technologies available. Once again, we aren't going through every single option as these are specific to each technology and we recommend you reach out to the Asterisk documentation for the specifics of each option. You will find the description of each field at the end of this lesson. We will create some device profiles throughout various lessons as well.

The Device Profiles can be created for **PJSIP** and **IAX2** devices and trunks. **Default Profiles** for **SIP, PJSIP, IAX2, and WebRTC** have already been created. These are the ones used by default for these technologies. You can modify these default profiles if needed, but this is uncommon.

To create a profile you only need to select the **Profile Type**, add a **Name**, and a **Description**. You will then find additional configuration parameters to configure the specific profile. If the option you are looking to configure is not available in the UI options, you can go to the **Advanced Tab** to add the specific **header parameter** with its **value.** Afterward, you can **Save** and then **Apply Changes**.



Profile Type: PJSIP | IAX2

General

Name *:
Description:

Network

Transport: Auto
Qualify Frequency: 30
Qualify Timeout: 3
Default Expiration: 3600
Maximum Expiration: 7200
Minimum Expiration: 600
Outbound Proxy:
Force rport: Yes
Use AVPF: No
RTP Symmetric: Yes
RTCP Mux: No
Asymmetric RTP Codec: Yes
Send Diversion Header: Yes
Send P-Asserted-Identity: Yes
Send Remote-Party-ID: Yes
Support Path: Yes

Image 11.5.1 - Device Profiles module.

For **PJSIP Device Profiles** we find the following options.

Network

- **Transport -** desired transport configuration.
- **Qualify Frequency -** the interval between attempts to rate the contact to reach it. If 0 never qualify. Time in seconds.
- **Qualify Timeout -** If the contact does not respond to the OPTIONS request before the time out, the contact is marked as unavailable. If the value is 0 there is no timeout.
- **Force rport -** force use of the return port.
- **ICE support -** enable the ICE mechanism to help traverse NAT.
- **Rewrite Contact -** this allows the contact header to be rewritten with the source IP address port.
- **Remove Existing -** this allows registration to succeed by displacing any existing contacts that now exceed the “Max Contacts” count. Any removed contacts are the next to expire. The behavior is beneficial when “Rewrite Contact” is enabled and “Max Contacts” is greater than one. The removed contact is likely the old contact created by “Rewrite Contact” that the device is refreshing.
- **Use AVPF -** determine if res_pjsip will use and enforce the use of AVPF for this endpoint.
- **RTP Symmetric -** enforce that RTP must be symmetric.
- **RTCP Mux -** with this option enabled, Asterisk will try to negotiate the use of the **rtcp-mux** attribute on all media streams. This will result in RTP and RTCP being sent and received on the same port. This switches the demultiplexing logic to the application rather than the transport layer. This option is useful when interoperating with WebRTC endpoints as they enforce the use of this option.
- **Asymmetric RTP Codec -** allows the send and receive RTP codec to differ.
- **Send Diversion Header -** send the forward header, transmitting the forwarding information to the called user agent.
- **Send P-Asserted Identity -** send the Header P-Asserted Identity.
- **Send Remote-Party-ID -** send the Header Remote-Party-ID.

- **WebRTC -** when set to **Yes**, this also enables the following settings required for basic WebRTC support to work: **rtcp_mux, use_avpf, ice_support, and use_received_transport**.

Media

- **Media Encryption -** this determines if **res_pjsip** will use and enforce the use of media encryption for this endpoint.
- **Direct Media -** this determines if media can flow directly between endpoints.
- **Received Media Transport -** this determines if res_pjsip will use the media transport received in the offer SDP in the corresponding response SDP.
- **Optimistic Media Encryption -** this determines whether encryption should be used if possible, but does not terminate the session if not achieved.
- **Disable NAT Direct Media -** direct media session disable is updated when NAT obstructs the media session.

DTLS

- **DTLS certificate -** certificate to use with DTLS connections.
- **DTLS Setup -** if we are willing to accept connections, connect with the other party, or both. Valid options are:
- **Active -** we want to connect with the other party.
- **Passive -** we only want to accept connections.
- **Actpass -** we will do both. This value will be used in outbound SDP when offered and for inbound SDP offers when a remote party sends actpass.
- **DTLS Verify -** verify that the provided peer certificate is valid.
- **DTLS Fingerprint Hash -** the hash to use for the fingerprint in SDP.
- **DTLS Rekey interval -** interval in which to renegotiate the TLS session and reactivate the SRTP session. If this is not configured or the provided value is 0, reordering will be disabled.

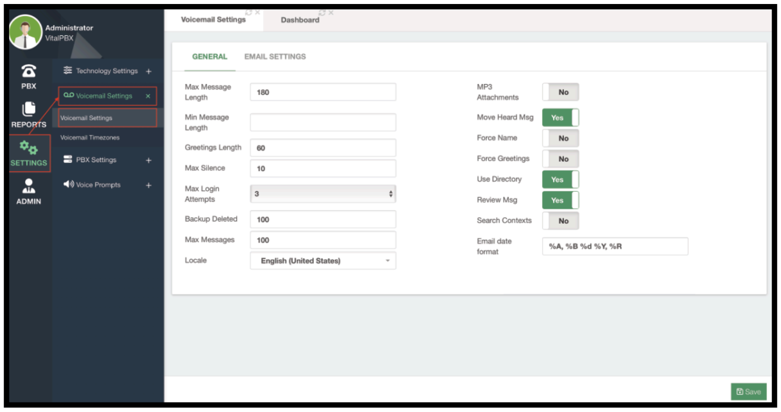
For **IAX2 Device Profiles**, we have the following options.

Network

- **Host -** hostname, or device address.
- **Type -** this defines the type of device.
  - **User -** this option is a device that makes calls, and requires authentication.
  - **Peer -** this option is a trunk device, accompanied by the Host.
  - **Friend -** this option is a combination of “User” and “Peer”.
- **Call Token -** this uses requirecalltoken for authentication.
- **Qualifier Frequency -** this defines the interval of qualifying in seconds. A value of zero will disable this feature.
- **Qualifier Timeout -** this defines the maximum response time in milliseconds before a device is considered unreachable. A value of zero will disable this feature.
- **Transfer -** allow transfers from this device

Section 11, Lesson 6 - Voicemail Settings

Now let's take a look into **Voicemail Settings**. These can be found under **Settings > Voicemail Settings > Voicemail Settings**. Voicemail settings are global settings you can configure for your voicemail management.



Max Message Length: 180
Min Message Length:
Greetings Length: 60
Max Silence: 10
Max Login Attempts: 3
Backup Deleted: 100
Max Messages: 100
Locale: English (United States)
MP3 Attachments: No
Move Heard Msg: Yes
Force Name: No
Force Greetings: No
Use Directory: Yes
Review Msg: Yes
Search Contexts: No
Email date format: %A, %B %d %Y, %R

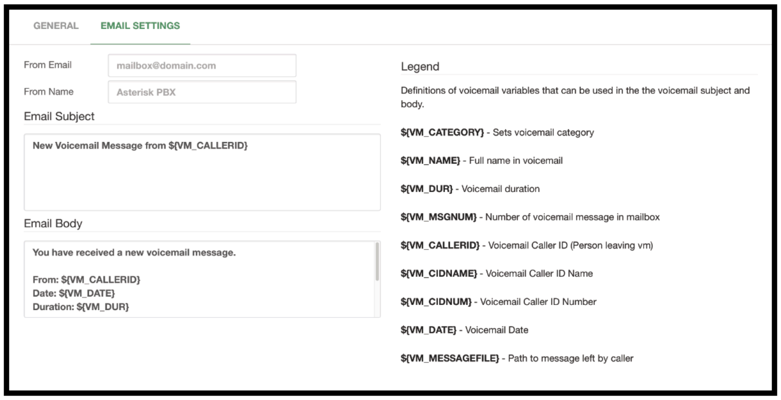
Image 11.6.1 - Voicemail Settings module.

Here you will find various settings that affect global settings for voicemail in your system. You can configure the following settings.

- **Max Message Length -** This is the maximum voicemail message length in seconds.
- **Min Message Length -** This is the minimum voicemail message length in seconds.
- **Greetings Length -** This is the maximum greeting length in seconds. The Greeting is the recording of the caller's name.
- **Max Silence -** This is the maximum silence length in seconds before VitalPBX ends the voicemail recording.
- **Max Login Attempts -** This is the maximum number of attempts users have to log into their voicemail box.
- **Backup Deleted -** This is the maximum number of deleted messages saved in the deleted folder.
- **Max Messages -** The maximum number of messages you can have per voicemail box. If set to 0, the voicemail box will be greetings only.
- **Locale -** This is the locale used for dates. For more locales, you will need to install them directly to the operating system. If you use multiple languages, we recommend not using letters for the date format.
- **MP3 Attachments -** This is an extended feature, only available when you have a Starter License or any of our licensing plans. This will convert the voicemail recordings to MP3 files when attaching them using the voicemail-to-email feature.
- **Move Heard Msg -** This will move heard voicemail messages to the OLD voicemail folder automatically.
- **Force Name -** This will force the users to record their names for their voicemail box.
- **Force Greetings -** Similar to Force Name, where callers will be forced to record a greeting.
- **Use Directory -** Allows you to find entries for forward/compose from the voicemail menu.
- **Review Msg -** This allows the callers to review their message before sending it to the extension's voicemail box.

- **Email Date Format -** Here you can define the date format for outgoing emails. You must use **standard strftime format string**. The most common options are as follows.
    - **%A -** Full day name.
    - **%a -** Abbreviated day name.
    - **%d or %e -** Day of the month in number.
    - **%B -** Full month name.
    - **%b or %h -** Abbreviated month name.
    - **%m -** Month number.
    - **%Y -** Full year with the century.
    - **%y -** Year abbreviation without the century.

Additionally, you will find the **Email Settings** tab. This is the same as the template found under the Email Templates module.



From Email: mailbox@domain.com

From Name: Asterisk PBX

Email Subject

New Voicemail Message from ${VM_CALLERID}

Email Body

You have received a new voicemail message.

From: ${VM_CALLERID}
Date: ${VM_DATE}
Duration: ${VM_DUR}

Legend

Definitions of voicemail variables that can be used in the the voicemail subject and body.

${VM_CATEGORY} - Sets voicemail category

${VM_NAME} - Full name in voicemail

${VM_DUR} - Voicemail duration

${VM_MSGNUM} - Number of voicemail message in mailbox

${VM_CALLERID} - Voicemail Caller ID (Person leaving vm)

${VM_CIDNAME} - Voicemail Caller ID Name

${VM_CIDNUM} - Voicemail Caller ID Number

${VM_DATE} - Voicemail Date

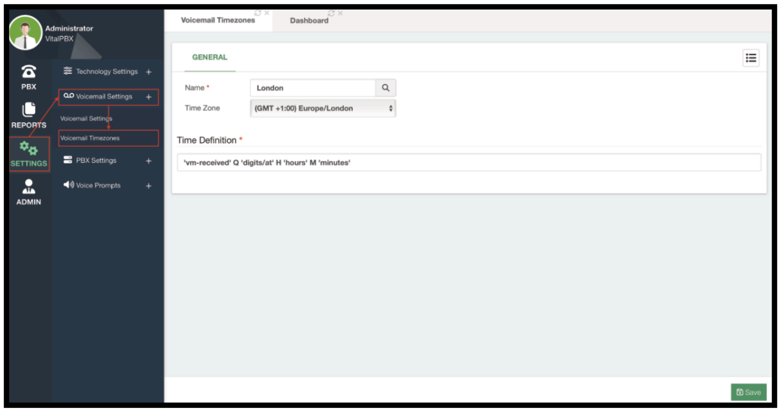
${VM_MESSAGEFILE} - Path to message left by caller

Image 11.6.2 - Voicemail Settings's Email Settings Tab.

Here, you will find the template used for the voicemail-to-email feature. You can see the list of variables you can use for the email body. If you made any changes, go ahead and **Save** and **Apply Changes**.

Section 11, Lesson 7 - Voicemail Timezones

Your end-users can be located in different time zones from the VitalPBX installation. This is why you can create **Voicemail Timezones**, so each user can have the appropriate time zone for their voicemail messages' envelope. For this, we must go to **Settings > Voicemail Settings > Voicemail Timezones**.



Name * London

Time Zone (GMT +1:00) Europe/London

Time Definition *

'vm-received' Q 'digits/at' H 'hours' M 'minutes'

Image 11.7.1 - Voicemail Timezones module.

First, you must enter the **Name** of the Voicemail Timezone to create, and select the **Time Zone** to consider.

Next, we have the **Time Definition.** This is what will be played back in the envelope for the voicemail message a caller leaves us. The supported values are as follows.

- **'filename' -** This is any Asterisk Sound to play. The filename must be placed between single ticks and in the exact casing of the file without the file extensions.
- **${VAR} -** This is a variable substitution. You can use any variable from the Voicemail Email Template, such as *${VM_CIDNAME}*, *${VM_CIDNUM}*, etc.
- **A or a -** Day of the week (Monday, Tuesday, …).
- **B, b, or h -** Month name (January, February, …).
- **d or e -** Numeric day of the month (First, Second, Third, …, Thirty-First).
- **Y -** Year.
- **I -** Hour in a 12-hour format.
- **H -** Hour in a 24-hour format (preceded by "oh").
- **K -** Hour in a 24-hour format (not preceded by "oh").
- **M -** Minute, with 00 pronounced as "o'clock."
- **N -** Minute, with 00 pronounced as "hundred." (US Military Time)
- **P or p -** AM or PM.
- **Q -** This is used for "today", "yesterday", or ABdY. (Note: this is not *standard strftime*)
- **R -** 24-hour time including minutes.

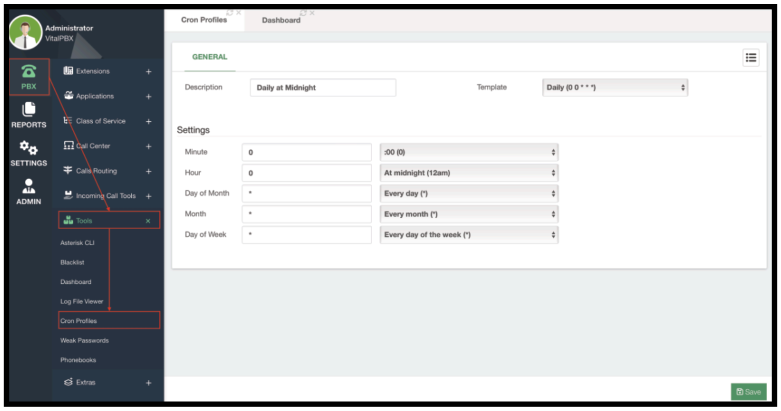
We already have some Voicemail timezones for eastern, central, central 24, military, and European. An example of a Time Definition can be as follows.

- ***Eastern, Time Zone: America/New York* - 'vm-received' Q 'digits/at' IMp**

This reads in the GMT (-04:00) Timezone as *"Voicemail Received Today at 12:45 PM"*.

Section 11, Lesson 8 - Cron Profiles

You might want to automate various scripts and operations within VitalPBX. For this, you can create **Cron Profiles**. Cron profiles are cron jobs that are executed in a specific schedule to automate different tasks within your VitalPBX server. To create a cron profile you must go to **PBX > Tools > Cron Profiles**.



Description: Daily at Midnight

Template: Daily (0 0 * * *)

Settings

Minute	0	:00 (0)
Hour	0	At midnight (12am)
Day of Month	*	Every day (*)
Month	*	Every month (*)
Day of Week	*	Every day of the week (*)

Image 11.8.1 - Cron Profiles module.

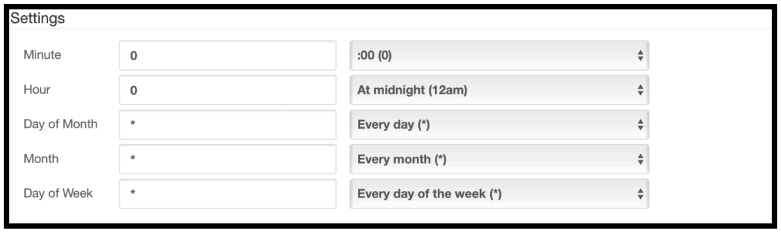
First, enter a **Description** to identify this cron profile.

Cron jobs can be a little complicated to wrap your mind around if you have never worked with them before. This is why we have included a **Template** section to select some of the most common schedules.

Cron jobs are usually defined in a row of five columns. These have been displayed in the **Settings** section. The columns correspond to the **Minutes, Hours, Day of the Month, Month, and Day of the Week**. You can enter any number or a star (*) for each field. The number will correspond to its value for the field. For example, for 12:45 PM you enter 12 for the hours and 45 for the minutes. The same goes for the month, any number between 1-12, for the day of the month 1-31, and for the day of the week 0-6. The star (*) means every interval of the field. For example, a star next to the month field means every month, and the same goes for each field. Keep in mind that the time is strictly in a 24-hour format.

For simplicity, we have also added a template for every field. This will have the most common option for each field.

If we use the template called Daily, the settings will accommodate to run every day at 00:00 hours (midnight). It will look like in the following image. You can then click on **Save and Apply Changes**.



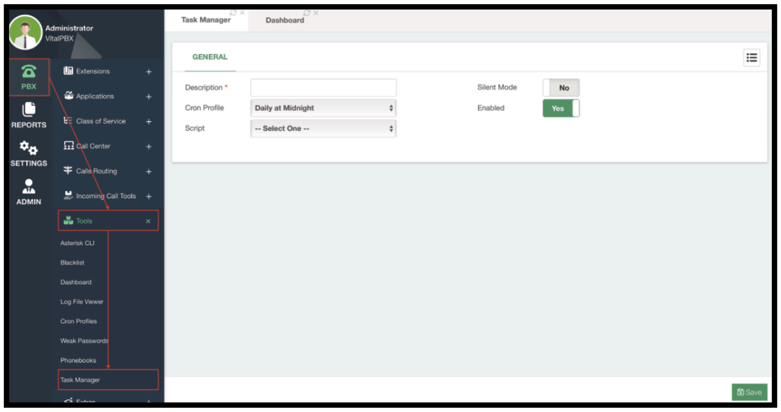
Settings

Minute	0	:00 (0)
Hour	0	At midnight (12am)
Day of Month	*	Every day (*)
Month	*	Every month (*)
Day of Week	*	Every day of the week (*)

Image 11.8.2 - Example of a Cron Profile setting.

Section 11, Lesson 9 - Task Manager

Now that you have a cron profile created, we can use it in any module to run a task automatically. We can also install the **Task Manager** add-on module. This is a free add-on module you can install from the add-ons module under **Admin > Add-Ons > Add-Ons**. Once the add-on is installed you can find it under **PBX > Tools > Task Manager**.



Description *
Cron Profile: Daily at Midnight
Script: -- Select One --
Silent Mode: No
Enabled: Yes

Image 11.9.1 - Task Manager module.

The Task Manager module will allow you to run any script you have created automatically using a Cron Profile. First, you need to upload your script to the following path.

```
/var/lib/vitalpbx/scripts
```

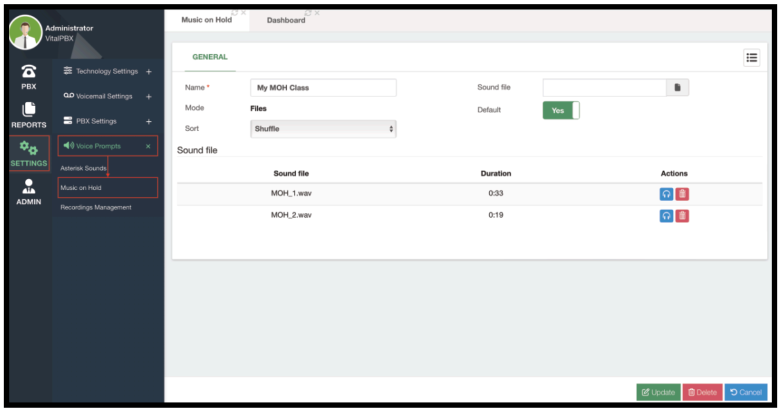
Next, you go back to the Task Manager module, add a **Description** to identify this task, select the **Cron Profile** you have created, and select the **Script** you uploaded. You can decide if the task is **Enabled** or not. You can then **Save** and **Apply Changes**. Optionally, you can enable the task in **Silent Mode**. This means that the task will run with any output it has suppressed.

The script you create can be anything that you want to perform from your VitalPBX server. This would be a **bash script** that you would normally run from the Linux CLI, so this task is not limited to affecting anything in VitalPBX per se, but it can deal with anything you want to perform from this server.

For example, you can create a script that copies the call recordings from the recordings directory to another one, or you can copy the backup files from the backups directory to another one. Then, with the Task Manager, you can run this script under a schedule to automate this function.

Section 11, Lesson 10 - Music on Hold

An important aspect of a PBX system is to be able to control the **Music on Hold**. This allows us to have a personalized experience for our callers, and give a more professional look to our business. To configure our music on hold, we must go to **Settings > Voice Prompts > Music on Hold**.



Name * My MOH Class

Mode Files

Sort Shuffle

Sound file

Default Yes

Sound file

Sound file	Duration	Actions
MOH_1.wav	0:33	
MOH_2.wav	0:19	

Image 11.10.1 - Music on Hold module with Files mode.

In this module, we create a **Music on Hold Classes**. A class is a playlist of sound files that will play when Music on Hold is used. Music on Hold is usually abbreviated as MOH.

First, you must enter a **Name** to identify this *Music on Hold Class*. Then, we select the **Mode** for this MOH, which can be either **Files** or **Custom**. Files will allow you to upload WAV files that can play based on the **Sort** option you select. You can sort in a **Linear** fashion, so the **Sound Files** play in the order they are uploaded, or in a **Shuffle** fashion and playback in a random order.

The first file is uploaded when you click on **Save**. To upload more files, you must go back to the MOH Class, select a new sound file, and click on **Update**. You will see the list of sound files at the bottom of your MOH class. You can playback the sound files or delete them if necessary.

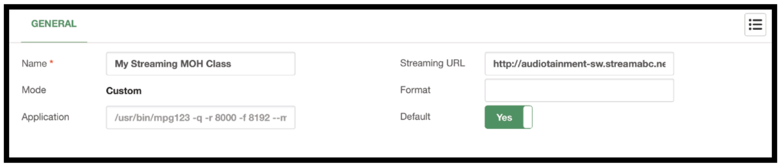
Additionally, you can set the MOH class as the **Default** MOH class. When you enable this option, every module that uses an MOH class will use this MOH class instead of the default music on hold that comes with VitalPBX. If you set a new MOH class as default, this will disable the option for any other MOH class that had this enabled. You cannot have two default MOH classes at once.

Once you have all your sound files uploaded, you can then **Apply Changes**.

The other mode available is **Custom**. This is an extended feature for the MOH module, where you can use a streaming URL to playback instead of predetermined sound files.

To create a custom MOH class, you must enter a Name to identify it and select the Custom mode. Then, you have the **Application** field. Here, you can select the application to playback the streaming MOH. By default, we use the **mpg123** application to playback the music, but you can use any other and enter the parameters here. You can leave this field blank to use the default values.

Next, we have the **Streaming URL**. This is the URL that has the music stream to playback. This can come from a streaming server or a streaming service that provides this URL.



GENERAL

Name *	My Streaming MOH Class	Streaming URL	http://audiotainment-sw.streamabc.ne
Mode	Custom	Format	
Application	/usr/bin/mpg123 -q -r 8000 -f 8192 --m	Default	Yes

Image 11.10.2 - Music on Hold module with Custom mode.

Make sure that the streaming URL you use does not use HTTPS. This is due to Asterisk not being able to process HTTPS URLs.

The **Format** field specifies the format option that the application will provide to Asterisk. The options you can enter here are the formats that Asterisk can accept, like **ulaw, alaw, wav, and mp3**. You can leave this field blank.

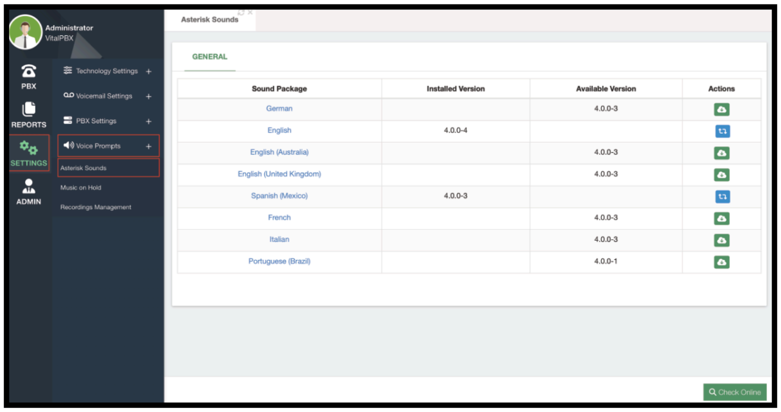
You can then **Save** and **Apply Changes**.

> ***Note:*** *With Debian 11, there is currently a bug (as of October 2023) with the FFMPEG libraries that may playback static noise. This may be fixed in a future version of Debian 11 or Debian 12.*

When using a streaming service, make sure that you take into account **Copyright** based on your local copyright laws, as well as the **quality, reliability, and content** of the streaming music. Remember that any caller will be able to listen to this stream when reaching your VitalPBX.

Section 11, Lesson 11 - Asterisk Sounds

As mentioned in a previous lesson, businesses nowadays may tend to users in multiple languages. This means that you must be able to provide the voice prompts in the caller's language. With VitalPBX English and Spanish are available by default, but additional languages can be added with the **Asterisk Sounds** module. This can be found under **Settings > Voice Prompts > Asterisk Sounds**.

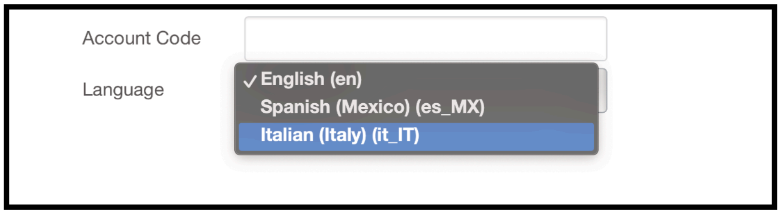


Sound Package	Installed Version	Available Version	Actions
German		4.0.0-3	
English	4.0.0-4		
English (Australia)		4.0.0-3	
English (United Kingdom)		4.0.0-3	
Spanish (Mexico)	4.0.0-3		
French		4.0.0-3	
Italian		4.0.0-3	
Portuguese (Brazil)		4.0.0-1	

Image 11.11.1 - Asterisk Sounds module.

The list of available voice prompt languages will appear blank in the beginning. To see the list of the latest languages, click on the green **Check Online** button. Next to the available language, you will find the **Actions** column. Here you can click the green **Install** button. This will install the voice prompts for that language. Once installed, you will see a blue **Reinstall** button, and a red **Trash** button to delete the voice prompt.

Once the language is installed, you can select more language options for voice prompts.



Account Code

Language: ✓ English (en)

Image 11.11.2 - Voice prompt language installed and available.

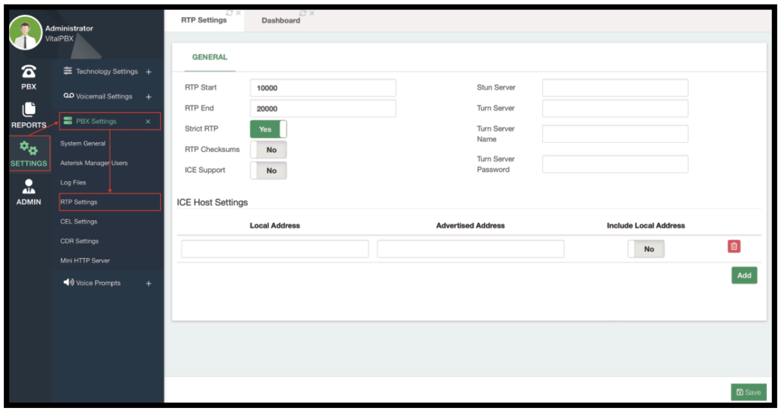
If you see that the voice prompts for a specific language are not available, we have created the following article so you can translate the prompts directly, https://vitalpbx.com/blog/how-to-translate-for-free-your-pbx-voice/.

In this article, we show you a small application we have created that uses Google© Translate™ to translate the voice prompts. You can then verify the translation and record the prompts, or use the following article to use Microsoft© Azure's™ Text-to-Speech to record these prompts, https://vitalpbx.com/blog/free-voice-guide-with-azure-free/. In this other article, we use another small application we have created to connect with Microsoft© Azure's™ TTS capabilities to record the prompts for us.

Afterward, you can send the translated recordings to us at sales@vitalpbx.com, and we will be able to add them to a future version of VitalPBX.

Section 11, Lesson 12 - RTP Settings

Next, we will take a look at the **RTP Settings**. These are found under **Settings > PBX Settings > RTP Settings**.



RTP Start: 10000

RTP End: 20000

Strict RTP: Yes

RTP Checksums: No

ICE Support: No

Stun Server

Turn Server

Turn Server Name

Turn Server Password

ICE Host Settings

Local Address	Advertised Address	Include Local Address
		No

Image 11.12.1 - RTP Settings module.

If you require to change the RTP port range, these can be modified with the **RTP Start and End fields**. By default, these go from port **10,000 to 20,000**. If you change these ports, make sure you also make the changes in the *Firewall Services* under **Admin > Firewall > Firewall Services**.

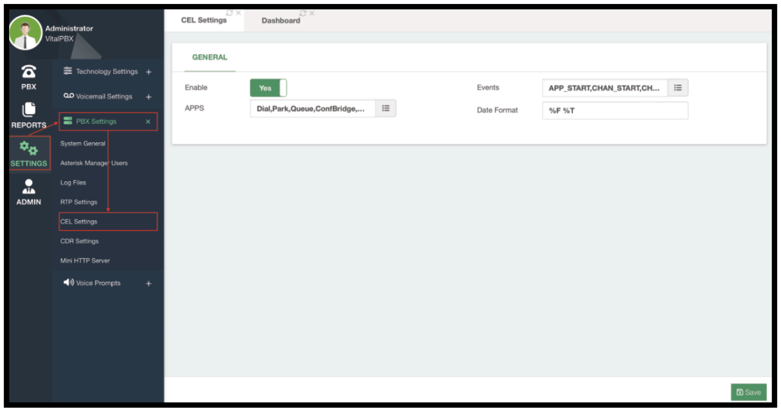
These fields can be left with their default values. You can enable or disable **Strict RTP**, and **RTP Checksums**. We recommend you leave Strict RTP on for security purposes. If it is disabled, VitalPBX will not drop packets that come from any source that is not the source for the RTP stream. If you are using a **STUN** or **TURN** server you can enter the necessary information here.

Additionally, you can enable **ICE Support** in this module. If you are using an ICE server, you can enter your settings under the **ICE Host Settings**. Where the **Local Address** is a LAN IP Address, and the **Advertised Address** is a Public IP Address.

If you made any changes here, **Save** and **Apply Changes**.

Section 11, Lesson 13 - CEL Settings

If you have a Starter License or any of our Licensing Plans, you automatically obtain extended features for your VitalPBX installation. One of these is the CEL events that you can see from the CDR module. These events can be modified in the **CEL Settings** module. To get there, we must go to **Settings > PBX Settings > CEL Settings**.



Enable: Yes
APPS: Dial,Park,Queue,ConfBridge,...
Events: APP_START,CHAN_START,CH...
Date Format: %F %T

Image 11.13.1 - CEL Settings module.

The first thing you will notice is that CEL will be disabled by default. This is why you might not see CEL events in the CDR. You can **enable** them so you can start logging the events. We have it disabled by default, since logging too many CEL events can fill your storage quickly if it is too small.

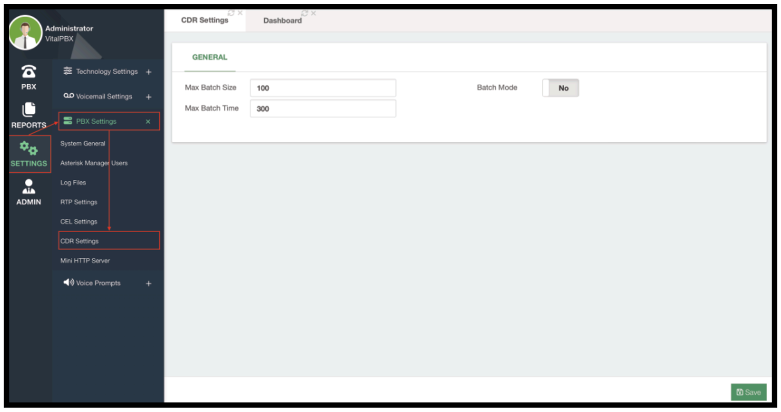
Here you can select the **APPs** and **Events** you wish to log for the CEL events. You can also change the **Date Format** using the **strftime** format.

You can then **Save** and **Apply Changes**.

With the CEL Events enabled, if you place a new call and check the CEL Events field in the CDR reports, you will now see information being logged.

Section 11, Lesson 14 - CDR Settings

If you are using a small server with a large number of calls per minute, alleviating the server usage is key. So, various optimizations can be made so your server is not fully loaded in peak time. One of these optimizations can be done with how the CDR operates with the **CDR Settings**.



Max Batch Size: 100

Max Batch Time: 300

Batch Mode: No

Image 11.14.1 - CDR Settings module.

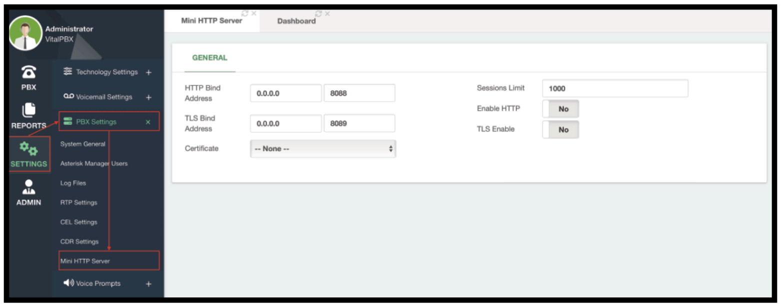
In this module, you can enable **Batch Mode** for the CDR. With batch mode, instead of logging every call in the CDR as each call ends, the data will be stored in a buffer. The **Max Batch Size** is how many calls are stored in the buffer, and the **Max Batch Time** is how often in seconds the calls are transferred from the buffer to the CDR logs and Database.

> ***Note:*** *When Batch Mode is enabled, there is a risk of data loss after unsafe Asterisk termination. For example, Power Loss, Software Crash, Kill -9, etc. So keep this in mind when using this feature.*

If you made any changes, you can **Save** and **Apply Changes**.

Section 11, Lesson 15 - Mini HTTP Server

With VitalPBX, you can host an additional web application using the **Mini HTTP Server**.



HTTP Bind Address: 0.0.0.0 8088

TLS Bind Address: 0.0.0.0 8089

Certificate: -- None --

Sessions Limit: 1000

Enable HTTP: No

TLS Enable: No

Image 11.15.1 - Mini HTTP Server module.

This module is typically used with WebRTC applications such as VitXi, so you can refer to the VitXi manual for more information on its use with that add-on application.

By default, we use ports **8088** and **8089** for the **HTTP and TLS Bind addresses**. If you change these, make sure you also change it in the *Firewall Services* under **Admin > Firewall > Firewall Services**.

To use this mini HTTP server, you need to **Enable HTTP**, and if you are using a **Certificate**, you need to **Enable TLS**. You can also change the **Sessions Limit** of how many WebSocket/HTTP sessions can be connected at the same time, by default this is 1000 sessions.

If you made any changes here, **Save** and **Apply Changes**.

Conclusion

These General PBX Configurations allow you to have full control over the technical aspects of your VitalPBX. It is important for you to know where these settings are located in the case where you need to make changes. The configurations can be very technical, but we have already provided the common defaults, so they would be mostly used on special occasions.

Additionally, tools like MOH and Asterisk Sounds make for a more professional appeal with your VitalPBX Installation.

SECTION 12 - MULTI-TENANT

Introduction

One of the most sought-after capabilities with VitalPBX is its Multi-Tenant add-on module. This is a very powerful tool so you can make the most of your VitalPBX Server. In this section, we will learn what Multi-Tenancy is and how you can use it to manage multiple companies with a single VitalPBX installation.

Section 12, Lesson 1 - Multi-Tenancy as a Concept

First, we must understand what Multi-Tenancy is, and how we can take advantage of it for our business.

Multi-Tenancy by definition is:

> *"A software architecture that allows a single instance to provide services for multiple clients."*

In other words, it is a shared system between multiple clients but operates in a flexible way and acts as if it is exclusive to a client. So, clients will feel as if they have a dedicated server for their use, but in reality, they are a fraction of a bigger system.

Multi-Tenant systems such as VitalPBX utilize three key mechanisms to ensure a secure and custom environment for each tenant.

- **It provides a mechanism for resource distribution.** This means it reduces hardware costs as this mechanism will automatically allocate resources based on use. There is no need to make calculations or assumptions about the hardware needs of each tenant. Thus, you optimize the use of the server hardware, allowing you to integrate more tenants in a single server.
- **It utilizes a security isolation mechanism.** This prevents invalid access and isolates tenants from each other. This way, tenants cannot see or access another tenant's information or configurations.
- **It has a customization mechanism.** This provides a UI model for each tenant, bringing back that theme of individuality between tenants. This supports access control, processes, and data through management settings. In simple words, each tenant will have their configurations presented in a personal format.

The advantages of using a Multi-Tenant system include the following.

- **Economy -** Once again, having a Multi-Tenant environment allows you to have a more economic model for development and maintenance costs. Instead of deploying multiple virtual machines locally or with VPS (Virtual Private Server) providers, you distribute the cost of a single server among multiple customers.
- **Easy to Update -** In a Multi-Tenant environment, like the one VitalPBX provides, you have easier administrative labor since it is a single instance, when you update the main system, all of your tenants update as well. You only need to update a single instance.

- **Information Security -** Each tenant has their configurations and data secured as each tenant has separate configuration files and are separated at a database level. No tenant is able to access another tenant's sensitive information or configurations.
- **Server Resource Optimization -** As mentioned previously, resources are optimized so you can manage more tenants in a single server instance.

Now let's make a clear comparison between Multi-Tenancy and Multi-Instancing. So, what is Multi-Instance? By definition, it is as follows.

> *"Multi-Instance is a type of architecture that allows for multiple companies to run on their own instance, with their own database, operating system, and application stack on the same hardware platform."*

So, you can think of Multi-Instance as multiple miniature virtual machines or containers that run as fully independent instances, hence the name Multi-Instance.

Some characteristics of Multi-Instancing can be as follows.

- **Independent Logic Storage -** This means that each instance is an independent installation with its own allocated storage.
- **Independent Operating Systems -** As each instance is a full installation, they may have their own operating system to run the application stack separately from other instances.
- **Independent Networking -** Similar to the previous point, given they are independent installations, each instance needs to have its own networking configuration.
- **One Database Per System -** Each instance will need to have its own database to operate.

Some disadvantages of running a Multi-Instance environment can be:

- **Too Individual -** This means that management for the instances has to be provided for each individual instance. This can become too tedious in bigger customer bases.
- **Repetitive Actions -** As each instance is its own full installation, you may need to perform similar configurations repeatedly for each one.
- **More Resource Expensive -** Since you need to allocate hardware resources to create each instance, you may overcompensate the amount of resources needed for an instance to prevent any unusual peaks in usage.

Let's look into Multi-Tenant with VitalPBX and how it works. VitalPBX adheres closely to the true definition of Multi-Tenancy, where you have **Shared Hardware Resources**. So there's no need to allocate resources to tenants, and tenants will only use as many resources as they need. It will all be managed in a single server. It is **Easy to Update for All Clients**, since by updating VitalPBX from the main tenant/system with the Administrator account, all tenants are updated as well. There can be **Communication Between Tenants**, as you can create special trunks so tenants will be able to call each other, even if they have the same numbering plan. Each tenant can have their **Connectivity to their Own VoIP Provider**, giving them more freedom or a sense of individuality. Additionally, you can **Centralize Trunking and Routes**, so instead of configuring trunks per tenant, if you provide them their numbers and routes, you can share a main trunk and outbound routes from the main tenant/system. Finally, once again, **Only One Server or Virtual Machine** is needed to manage multiple tenants.

In this diagram, you can get a bird's eye view of how Multi-Tenancy with VitalPBX works.

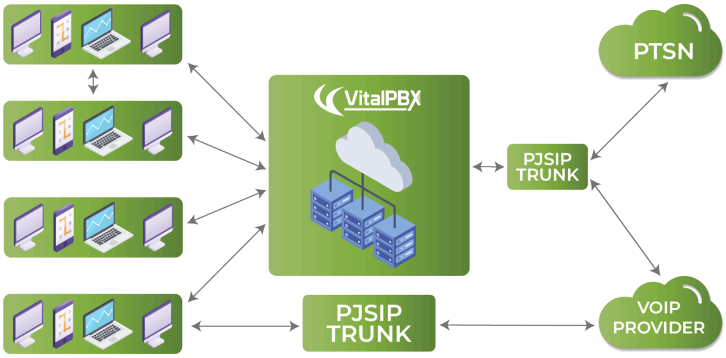


Image 12.1.1 - VitalPBX Multi-Tenant concept diagram.

With the concept explained, you can see that a Multi-Tenant environment allows you to manage multiple companies with great ease and effectiveness. Now, let's start creating our tenants in VitalPBX.

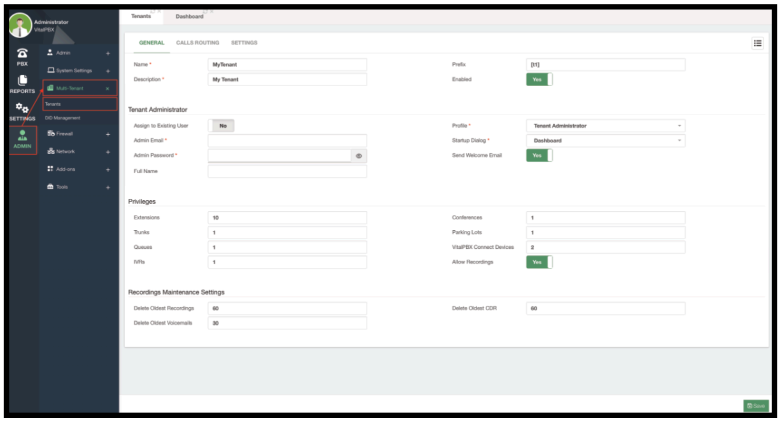
Section 12, Lesson 2 - Tenant Configuration

To start creating a Multi-Tenant environment with VitalPBX, you will first need to install the Multi-Tenant module under **Admin > Add-Ons > Add-Ons**.

> ***Note:*** *The Multi-Tenant module in VitalPBX is a commercial add-on. This means that you need to purchase a license to be able to configure more than one tenant. Licensing for the Multi-Tenant add-on can be purchased for a bundle of tenants through the store on the VitalPBX website, or you can subscribe to the Carrier Plus Licensing Plan so you can create as many tenants as your system can handle.*
>
> *If you use the Carrier Plus Licensing Plan, even though software-wise we don't limit the number of tenants, we recommend up to 100 tenants in a single server. This is based on our recommended maximum hardware specifications.*
>
> - *16-core CPU at 3.2 Ghz*
> - *At least 32 GB of Memory*
> - *At least 4 TB of Storage*
>
> *This is mostly an Asterisk limitation to the number of tenant configuration files it can handle. If you go above this number of tenants you might start experiencing a decrease in the quality experience for end users and administrators in terms of system slowdowns, especially when applying changes.*

Once the add-on is installed, you can refresh your browser, and you will have the Multi-Tenant menu option under **Admin > Multi-Tenant**.

Now, we must go to **Admin > Multi-Tenant > Tenants**. Here is where we will configure all of our tenants.



Name * MyTenant
Description * My Tenant
Prefix [t1]
Enabled Yes

Tenant Administrator

Assign to Existing User No
Admin Email *
Admin Password *
Full Name
Profile * Tenant Administrator
Startup Dialog * Dashboard
Send Welcome Email Yes

Privileges

Extensions 10
Trunks 1
Queues 1
IVRs 1
Conferences 1
Parking Lots 1
VitalPBX Connect Devices 2
Allow Recordings Yes

Recordings Maintenance Settings

Delete Oldest Recordings 60
Delete Oldest Voicemails 30
Delete Oldest CDR 60

Image 12.2.1 - Multi-Tenant Tenants module.

Here you will first see the **General Tab**. This is where the main configurations of a tenant will go. First, add a **Name** and a **Description** to identify this tenant. The name cannot have any spaces or special characters, while the description can.

Next, we have the **Prefix**, which is a key element to how Multi-Tenancy works with VitalPBX. The prefix is used throughout the tenant's configurations to make their devices and trunks unique. With the prefix, you can have the same numbering plan for every tenant without it clashing between tenants. By taking advantage of how VitalPBX separates the extension numbers and devices, you can have extension 1000 in tenant A and extension 1000 in tenant B.

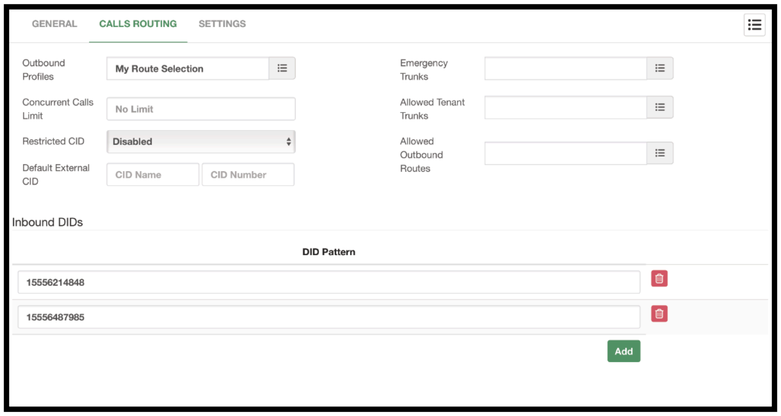
You can **enable or disable** a tenant at any time.

Next, we have the **Tenant Administrator** section. Here, you can create a new user that can manage the tenant. This is similar to how you create a user in the Users module under **Admin > Admin > Users**. You can select a User Profile, which has the Tenant Administrator profile by default. You can make changes to this user profile or make a new one if you want to add or remove access to certain modules in the VitalPBX Web UI. When creating a new Tenant Administrator, you can send the **Tenant Welcome Email**. This uses the email template found under **Admin > Settings > Email Templates**. This can be used to notify the Tenant Administrator that a new tenant has been created for them and give them the credentials so they can manage their new tenant. Optionally, you can change the Startup Dialog screen they will see when they log in.

Next, we have the **Privileges** section. Here you can establish the number of items the tenant will be able to create. You can set the number of **Extensions, Trunks, Queues, IVRs, Conference Bridges, Parking Lots, and VitalPBX Connect Devices (If the module is installed)**.

Afterward, you can **Allow Recordings** for the tenant. When allowing recordings on a tenant, you will see the **Recording Maintenance Settings**. This uses our *Maintenance* add-on module, which we will see more about later. Here, you can establish how long you are going to keep Call Recordings, Voicemails, and CDR items. The fields accept any number of days you wish to keep these items before they are deleted.

With the general settings set, we can now go to the **Calls Routing** tab.



Outbound Profiles: My Route Selection
Concurrent Calls Limit: No Limit
Restricted CID: Disabled
Default External CID: CID Name | CID Number
Emergency Trunks:
Allowed Tenant Trunks:
Allowed Outbound Routes:

Inbound DIDs

DID Pattern
15556214848
15556487985

Image 12.2.2 - Tenants module Calls Routing tab.

Here is where you are going to manage the incoming and outgoing calls for this tenant.

With VitalPBX you can configure your Trunks and Outbound Routes on the Main Tenant/System and share them with your tenants. This way, if you are managing the DID numbers and trunks for your tenants, you don't need to configure a trunk for each tenant and create the outbound routes over and over.

For this, once you have your trunk and outbound routes configured on the main tenant, you can use what we call **Outbound Profiles** to share them with your tenants. Outbound profiles are basically *Route Selections* that contain the outbound routes you wish to share.

So, go to **PBX > Class of Service > Route Selections**. Create a Route Selection with your outbound routes, and **Save** and **Apply Changes**.

Then, under the Outbound Profiles field, you can select the Route Selection you have created. This way, the Route Selection is applied to the *All Permissions Class of Service* within the tenant, and the tenant will be able to place outgoing calls.

Extensions within the tenant with this *Class of Service*, or any *Class of Service* with this *Route Selection* will be able to place outgoing calls through the trunk in the main tenant. Any changes you make from the main tenant to these routes or trunks will be automatically applied to all tenants using it.

You can also set a **Concurrent Calls Limit**, to limit the number of concurrent outgoing calls the tenant can have.

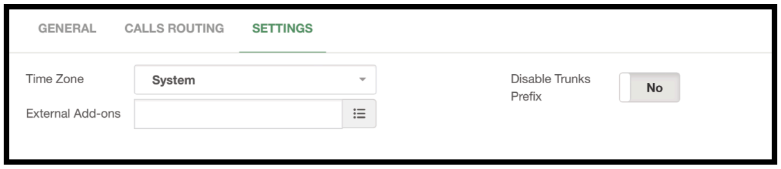
For incoming calls, you can declare the DIDs you want to assign to the tenant under the **Inbound DIDs** list. Whenever a call comes to the main tenant toward this DID, the call will be automatically routed to the tenant, and within the tenant, you can create an inbound route to route the call to any destination.

Next, you can define how to treat the outgoing CID (Caller ID) information with the **Restrict CID** option. Here you can choose between the following.

- **Disabled -** This will allow the tenant to use any CID information they send for outgoing calls.
- **Default -** This will use the values entered in the Default External CID fields.
- **DIDs -** This will allow outgoing calls only if the tenant uses any of the numbers defined under the Inbound DIDs list. This will prevent tenants from using any CID information that is not an assigned DID number.

Optionally, you can define specific **Emergency Trunks**, if these differ from the outbound profiles. As well as **Allowed Tenant Trunks** to give the tenant the ability to create a trunk with other tenants. We will see more about Tenant Trunks in the next lesson. Finally, the Allowed Outbound Routes is an older way of assigning an outbound route from the Main Tenant to the different tenants. This is not used as often anymore.

Finally, we have the **Settings** tab. Here we can define additional settings like a **Time Zone** for this tenant, in case it is located in a different time zone to the main system. Select which **External add-ons** from the Sonata Suite and VitXi you will allow the tenants to have access from the User Menu in the upper right-hand corner.



Time Zone: System

External Add-ons:

Disable Trunks Prefix: No

Image 12.2.3 - Tenants module Settings tab.

Finally, you can **Disable the Trunks Prefix**, this will remove the prefix defined in the general tab from the trunks created within the tenant. This can be useful if you need to create more than one trunk in different tenants from the same provider using the Username Authentication method.

> ***Note:*** *Disabling the Trunk Prefix for multiple tenants may lead to device name duplicity, meaning that there are multiple trunks using the same Local Username. This will eventually harm the performance of Asterisk, leading to an application crash.*

With all of these configurations set, you can now **Save** and **Apply Changes**.

As the Administrator of the Main Tenant, you can navigate between tenants easily. Refresh your browser and in the upper left-hand corner under your user's name, you will see that you now have a drop-down menu.

Here, you will find the list of the different tenants. You can also search for a specific tenant.

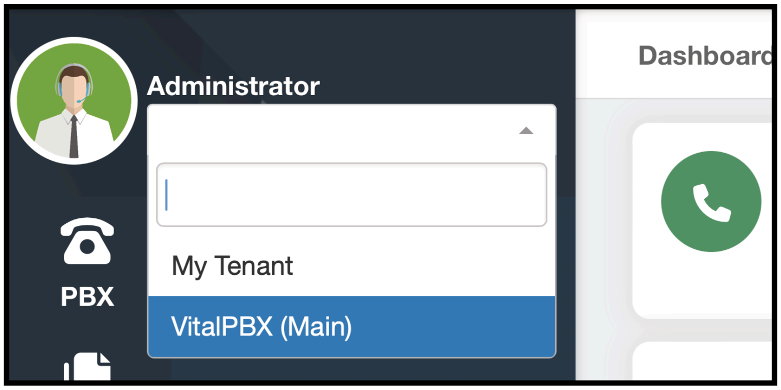


Image 12.2.4 - Tenant selection drop-down menu.

From here, if you click on the tenant, you will enter the tenant and be able to manage it.

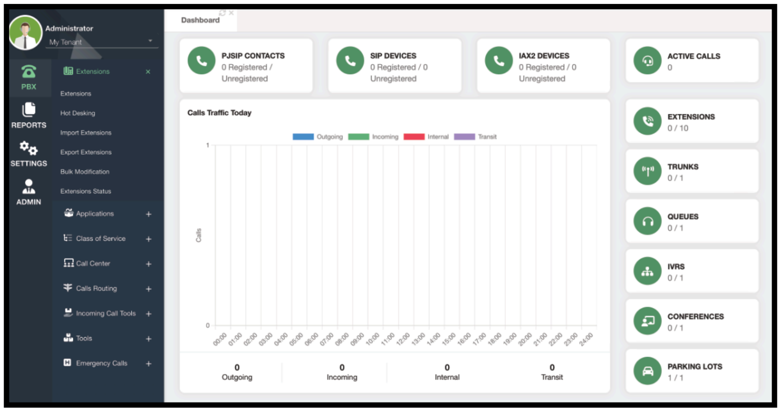
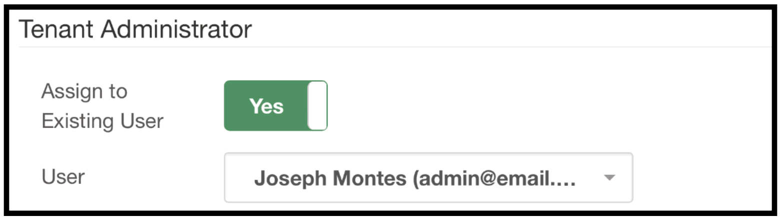


Image 12.2.5 - Tenant Dashboard.

As you can see, the dashboard is a little bit different from the Main Tenant's dashboard, as it contains information relevant to this tenant only. Otherwise, the modules and navigation are the same as any other VitalPBX installation. Here, tenant administrators can start to configure their extensions, IVRs, Queues, and more. Just as if they had their own VitalPBX installation.

If you log out and use the credentials for the Tenant Administrator you created, you will see this same tenant dashboard and configurations, without the ability to access the Main Tenant. You can assign multiple tenants to the same Tenant Administrator user account. When creating a tenant, you can select Assign to Existing User, which will show you a list of available users.



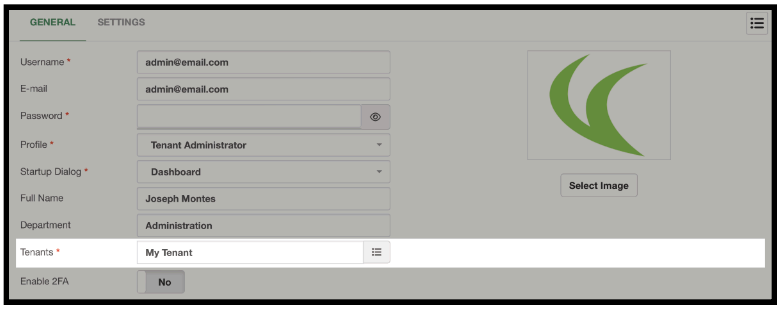
Tenant Administrator

Assign to Existing User: Yes

User: Joseph Montes (admin@email....

Image 12.2.6 - Assigning a Tenant to an existing user.

Additionally, you can go to **Admin > Admin > Users**, and when creating a user this way, you can select which tenants this user has access to.



Username * admin@email.com
E-mail admin@email.com
Password *
Profile * Tenant Administrator
Startup Dialog * Dashboard
Full Name Joseph Montes
Department Administration
Tenants * My Tenant
Enable 2FA No

Image 12.2.7 - Assigning a tenant to user in the Users module.

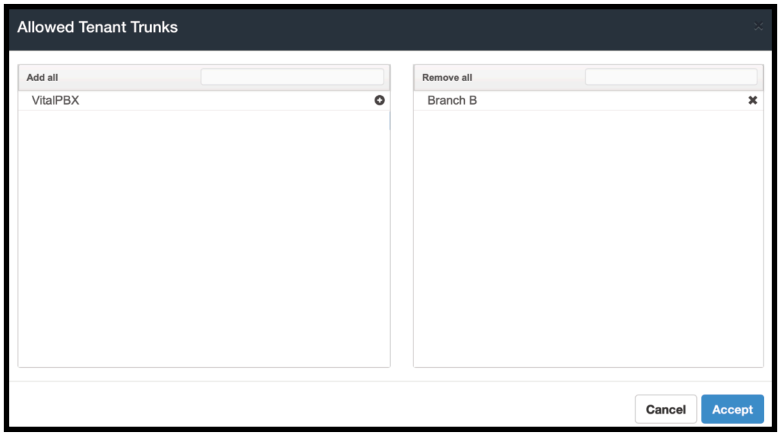
When a user has access to multiple tenants they will also have the tenant selection drop-down menu to navigate between tenants.

Having the ability to assign multiple tenants to a single user gives you the ability to create a **Reseller Environment**, where these Tenant Administrators can manage their tenants and sell the PBX service to their customers. With all of this covered, you now have a full Multi-Tenant environment with VitalPBX!

Section 12, Lesson 3 - Tenant Trunks (Inter-Tenant Communications)

Let's say you have a case where a customer has multiple branches and wants to have a tenant for each branch, but also they want extensions from Branch A to be able to call extensions on Branch B.

For this, we have the ability to create **Tenant Trunks**. Tenant Trunks allow us to create a direct connection between tenants without having to go through the network under a traditional PJSIP Trunk. Since both tenants are inside the same server, they can communicate directly within the server. To use Tenant Trunks, you must have at least two tenants who need to communicate with each other. Next, go to each tenant's Calls Routing Tab and click on the **Allowed Tenant Trunks** tab.



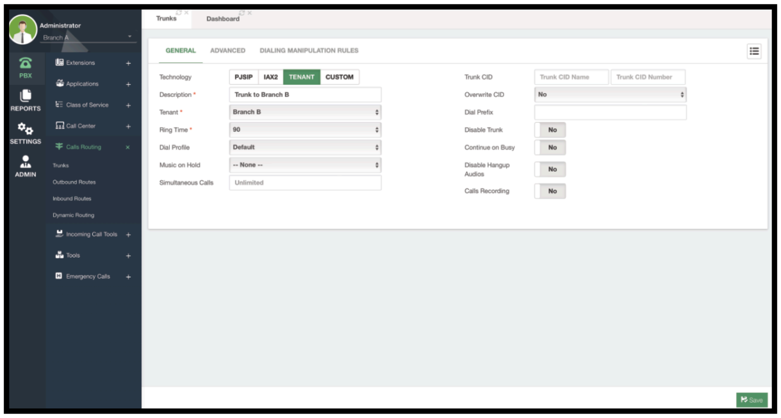
VitalPBX
Branch B

Image 12.3.1 - Allowed Tenant Trunks selection.

On each tenant, you must select the tenants with which they can have a tenant trunk. So, based on our example, in Branch A add *Branch B* as an allowed tenant trunk, and in Branch B, add *Branch A* as an *Allowed Tenant Trunk*.

Once you have configured the Allowed Tenant Trunk, **Save** and **Apply Changes**.

Now, navigate to each tenant and go to **PBX > Calls Routing > Trunks**.



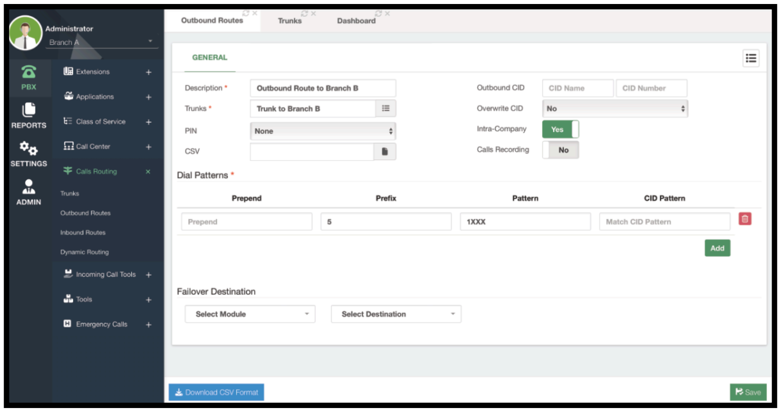
Technology	PJSIP · IAX2 · **TENANT** · CUSTOM
Description *	Trunk to Branch B
Tenant *	Branch B
Ring Time *	90
Dial Profile	Default
Music on Hold	-- None --
Simultaneous Calls	Unlimited
Trunk CID	Trunk CID Name · Trunk CID Number
Overwrite CID	No
Dial Prefix	
Disable Trunk	No
Continue on Busy	No
Disable Hangup Audios	No
Calls Recording	No

Image 12.3.2 - Configuring a Tenant Trunk within a Tenant.

Here, you will see that an additional technology option is now available with the option for **Tenant**. When you select Tenant, an abridged selection of options will be available. We can leave most of them by default. We add a **Description** to identify the Tenant Trunk, and we select the **Tenant** with whom we will connect.

Then, we **Save** and **Apply Changes**.

Now we can head over to **PBX > Calls Routing > Outbound Routes**.



Description *	Outbound Route to Branch B
Trunks *	Trunk to Branch B
PIN	None
CSV	
Outbound CID	CID Name · CID Number
Overwrite CID	No
Intra-Company	Yes
Calls Recording	No

Dial Patterns *

Prepend	Prefix	Pattern	CID Pattern
Prepend	5	1XXX	Match CID Pattern

Failover Destination

Select Module · Select Destination

Image 12.3.3 - Configuring an Outbound Route within a Tenant.

Here we are going to create an outbound route using the Tenant Trunk we have created.

It is possible to have the same numbering plan for both tenants. Meaning, that you can have extensions 1000-1100 in the tenant for Branch A, and extensions 1000-1100 in the tenant for Branch B. So to be able to specify that you are calling Branch B from Branch A, we will use a **Prefix**. In this example, we are using 5 as a prefix for calls going to the other branch's tenant, but you can use any other number you like.

For the **Pattern**, we will use the following value.

```
1XXX
```

With this, we can dial any number from 1000 to 1999. So to dial to the other branch's tenant, we dial something like this.

```
51000
```

The 5 is then dropped when calling the other tenant, and extension 1000 is reached. This way, extension 1000 on Branch A can call extension 1000 on Branch B.

Additionally, we recommend you enable the **Intra Company** option, as this will use the *Internal Caller ID* information, instead of the *External Caller ID*.

Remember, you need to create the Tenant Trunk and Outbound Route within each tenant. Once you have configured both tenants, you are now able to place calls between each tenant using the **Tenant Trunks** and dial using the prefix and pattern from the **Outbound Routes**.

Conclusion

With the understanding of Multi-Tenant as a concept and the tools that VitalPBX provides, you now have a full Multi-Tenant environment you can use to provide a PBX as a service for your customers. This way, you do not have to create a new VitalPBX installation for every customer, allowing you to reduce your costs and maintenance.

Additionally, you can now share your Trunks and Outbound Routes with your tenants, so you don't have to perform repetitive tasks of creating a trunk and routes for each tenant. And every time you update VitalPBX from the Main Tenant, all of your tenants will be up to date automatically.

You now have a system to create a direct communication channel between each tenant using Tenant Trunks. Allowing you to place calls between tenants with ease.

As you can see, managing multiple tenants with a single VitalPBX instance is easy, and beneficial if you are looking to provide a PBX as a service for your customers, or manage multiple branches.

SECTION 13 - HIGH AVAILABILITY

Introduction

With VitalPBX we have researched a couple of options for **High Availability**. With these options, you can create a redundancy system for your VitalPBX installations. So if a VitalPBX server fails for any reason, a secondary server comes online to take its place.

Section 13, Lesson 1 - High Availability Using DRBD

Let's take a look at the first option, which creates a high availability environment between two VitalPBX instances.

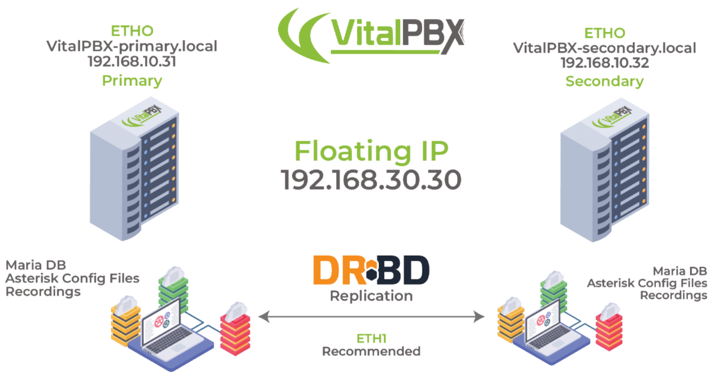


Image 13.1.1 - VitalPBX DRBD High Availability Concept Diagram.

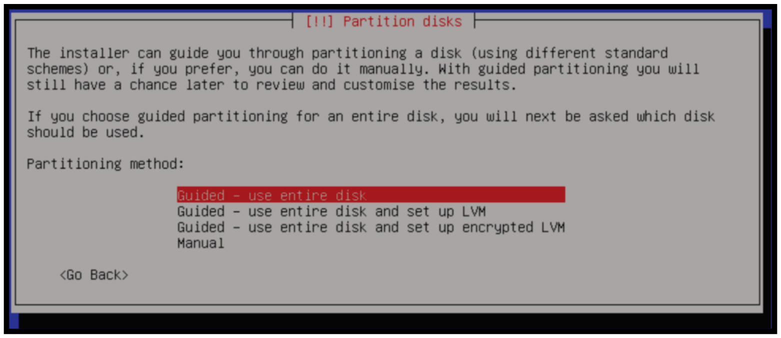
In this High Availability environment, we will be using DRBD or Distributed Replicated Block Device. This involves ensuring critical systems and services are available with minimal downtime in case of a failure. DRBD enables real-time data replication between nodes to ensure data availability and integrity.

First, let's look into the requirements for this type of High Availability.

- **Physical Infrastructure and Networking -** Two or more identical nodes (servers) to implement redundancy. These need to have the same hardware specifications and VitalPBX licensing. This means that if you are using a Carrier Plus license, each server will need its own license. This will ensure that both servers have the same permissions when the configurations are being replicated. We also need a reliable and low-latency network connection between the nodes. This can be a dedicated replication network (preferred) or a shared network if it's of high quality.
- **Operating System / VitalPBX version -** Nodes should run the same operating system using the same version and the same version of VitalPBX.
- **Disk Partitioning -** The storage device to be replicated should be partitioned and accessible on both nodes. Each node should have sufficient storage space to accommodate replication and data.
- **Network Configuration -** Each node should have static IP addresses and resolve correctly in the local DNS system or in the /etc/hosts file of the other node. Host names should be consistent across nodes.

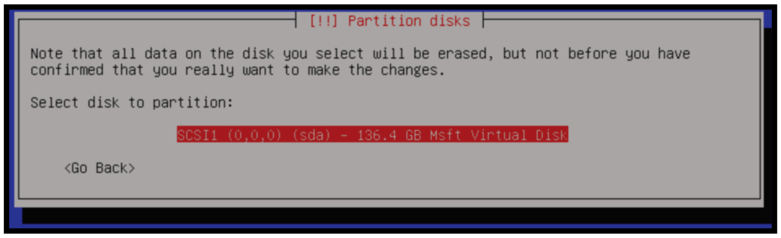
- **DRBD -** Install and configure DRBD on both nodes. Configure DRBD resources that define which devices will be replicated and how replication will be established. At the time of installation leave the largest amount of space on the hard drive to store the variable data on both servers. Define node roles: primary and secondary nodes.

With these requirements met and understood, we can start by installing Debian and VitalPBX on **two** servers. You can start by following the **Installation Section** for this guide. When you get to the partitions part of the installation, you must select **Guided - use the entire disk**.



```
[!!] Partition disks

The installer can guide you through partitioning a disk (using different standard
schemes) or, if you prefer, you can do it manually. With guided partitioning you will
still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk
should be used.

Partitioning method:

                    Guided - use entire disk
                    Guided - use entire disk and set up LVM
                    Guided - use entire disk and set up encrypted LVM
                    Manual

    <Go Back>
```

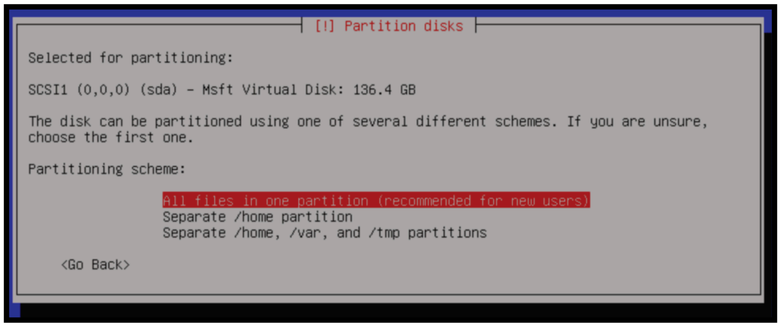
Image 13.1.2 - Debian installation disk partition wizard.



```
[!!] Partition disks

Note that all data on the disk you select will be erased, but not before you have
confirmed that you really want to make the changes.

Select disk to partition:

                   SCSI1 (0,0,0) (sda) - 136.4 GB Msft Virtual Disk

    <Go Back>
```

Image 13.1.3 - Debian installation disk partition disk selection.

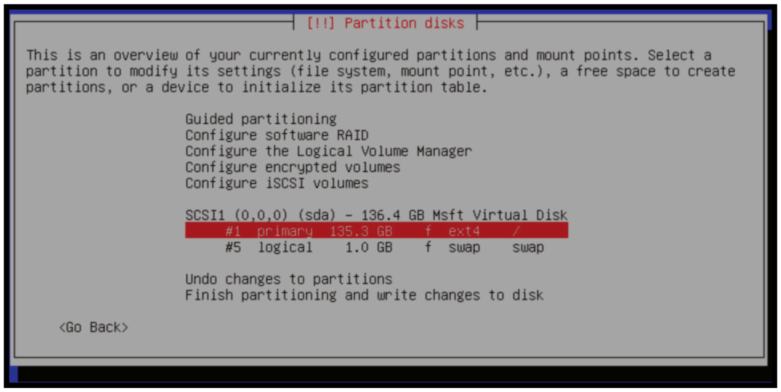
Next, select the option **All files in one partition (recommended for new users)**.



```
[!] Partition disks

Selected for partitioning:

SCSI1 (0,0,0) (sda) - Msft Virtual Disk: 136.4 GB

The disk can be partitioned using one of several different schemes. If you are unsure,
choose the first one.

Partitioning scheme:

                  All files in one partition (recommended for new users)
                  Separate /home partition
                  Separate /home, /var, and /tmp partitions

    <Go Back>
```

Image 13.1.4 - Debian installation disk partition partitioning scheme.

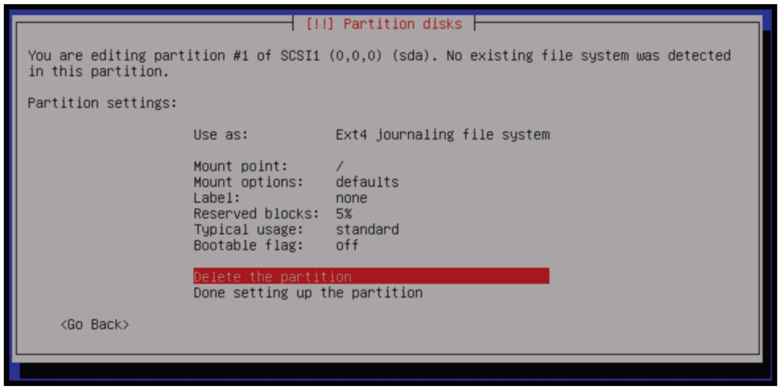
On the next screen select the **#1 Primary** partition to delete it.



```
                         [!!] Partition disks

This is an overview of your currently configured partitions and mount points. Select a
partition to modify its settings (file system, mount point, etc.), a free space to create
partitions, or a device to initialize its partition table.

                    Guided partitioning
                    Configure software RAID
                    Configure the Logical Volume Manager
                    Configure encrypted volumes
                    Configure iSCSI volumes

                    SCSI1 (0,0,0) (sda) - 136.4 GB Msft Virtual Disk
                         #1  primary  135.3 GB    f  ext4     /
                         #5  logical    1.0 GB    f  swap     swap

                    Undo changes to partitions
                    Finish partitioning and write changes to disk

    <Go Back>
```

Image 13.1.5 - Debian installation disk partition partition selection.

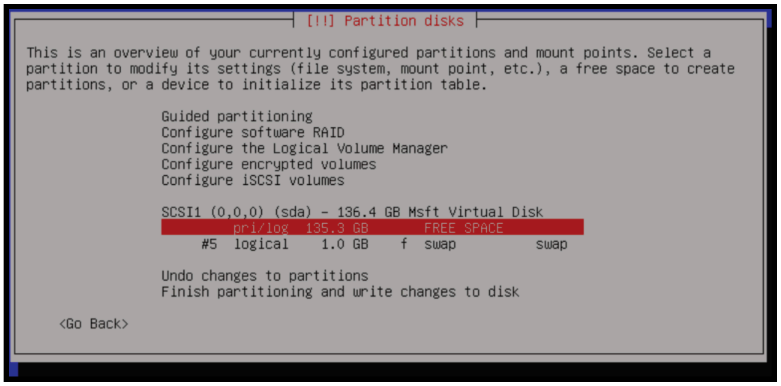
Delete the partition **#1 Primary** partition to create **Free Space**.



```
                         [!!] Partition disks

You are editing partition #1 of SCSI1 (0,0,0) (sda). No existing file system was detected
in this partition.

Partition settings:

                     Use as:           Ext4 journaling file system

                     Mount point:      /
                     Mount options:    defaults
                     Label:            none
                     Reserved blocks:  5%
                     Typical usage:    standard
                     Bootable flag:    off

                     Delete the partition
                     Done setting up the partition

    <Go Back>
```

Image 13.1.6 - Debian installation disk partition deletion.

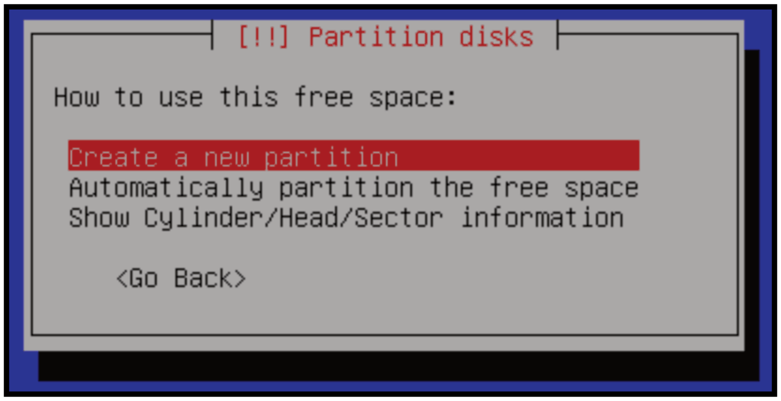
With the partition deleted, we select the **pri/log FREE SPACE** option.



```
                         [!!] Partition disks

This is an overview of your currently configured partitions and mount points. Select a
partition to modify its settings (file system, mount point, etc.), a free space to create
partitions, or a device to initialize its partition table.

                 Guided partitioning
                 Configure software RAID
                 Configure the Logical Volume Manager
                 Configure encrypted volumes
                 Configure iSCSI volumes

                 SCSI1 (0,0,0) (sda) - 136.4 GB Msft Virtual Disk
                          pri/log  135.3 GB        FREE SPACE
                      #5  logical    1.0 GB    f  swap          swap

                 Undo changes to partitions
                 Finish partitioning and write changes to disk

    <Go Back>
```

Image 13.1.7 - Debian installation disk partition free space selection.

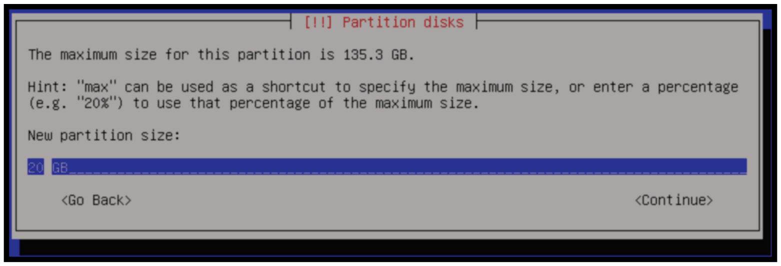
You will now select how to use the free space. Select the **Create a new partition** option.



```
[!!] Partition disks

How to use this free space:

 Create a new partition
 Automatically partition the free space
 Show Cylinder/Head/Sector information

    <Go Back>
```

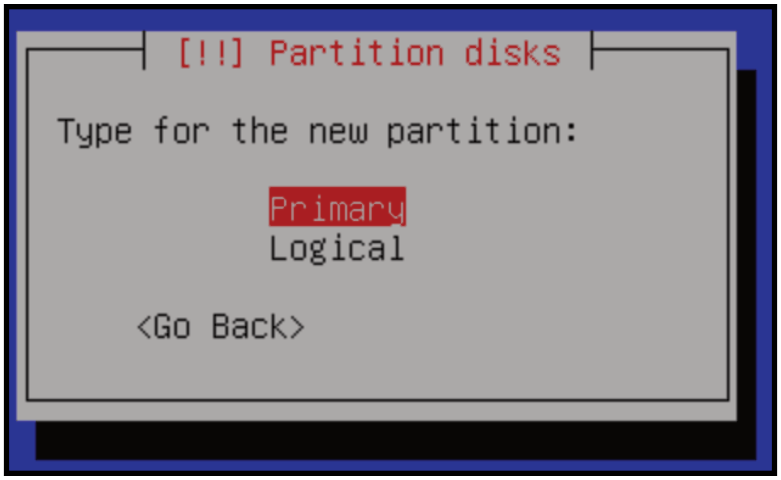
Image 13.1.8 - Debian installation creating a new partition.

Now, change the capacity of this partition to 20GB. This partition is solely for the OS and its applications. We make sure that it has enough space for the future. As a rule of thumb, this partition must be at least 20GB or 10% of your total storage space. So if you have a 1TB drive, you would allocate 100GB, for example.



```
[!!] Partition disks

The maximum size for this partition is 135.3 GB.

Hint: "max" can be used as a shortcut to specify the maximum size, or enter a percentage
(e.g. "20%") to use that percentage of the maximum size.

New partition size:

20 GB

    <Go Back>                                                        <Continue>
```

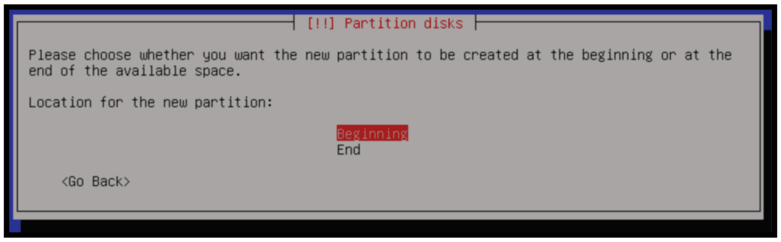
Image 13.1.9 - Debian installation partition size.



```
[!!] Partition disks

Type for the new partition:

          Primary
          Logical

    <Go Back>
```

Image 13.1.10 - Debian installation partition type.

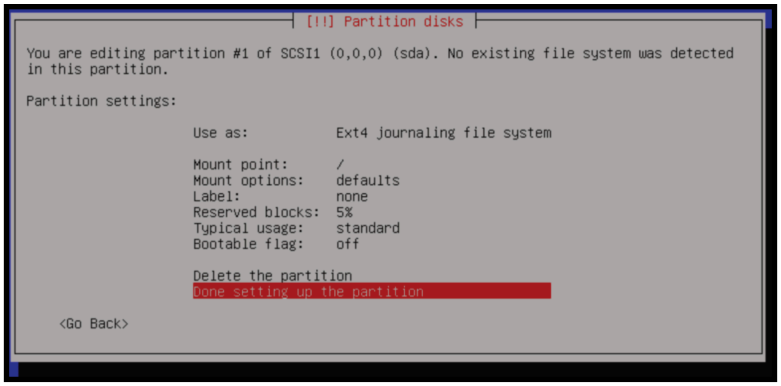
We then define this partition as a **Primary Partition**. Afterward, we will select the location for this partition to be the **Beginning**.



```
[!!] Partition disks

Please choose whether you want the new partition to be created at the beginning or at the
end of the available space.

Location for the new partition:

                                        Beginning
                                        End

    <Go Back>
```

Image 13.1.11 - Debian installation partition location.

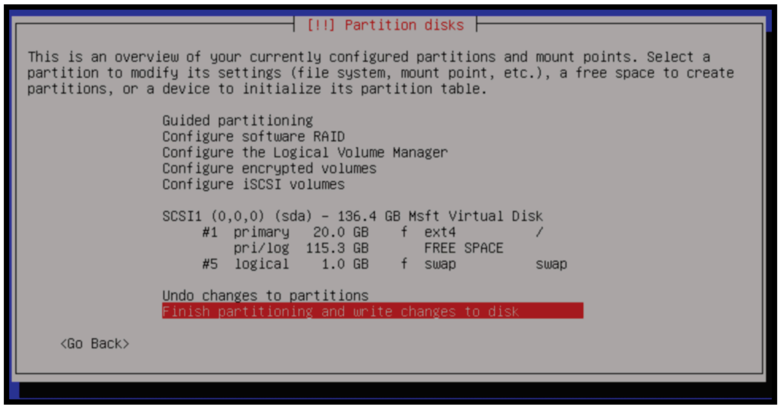
With this set, we will be shown a summary of the changes to this partition. Select the option **Done setting up the partition.**



```
[!!] Partition disks

You are editing partition #1 of SCSI1 (0,0,0) (sda). No existing file system was detected
in this partition.

Partition settings:

                     Use as:          Ext4 journaling file system

                     Mount point:     /
                     Mount options:   defaults
                     Label:           none
                     Reserved blocks: 5%
                     Typical usage:   standard
                     Bootable flag:   off

                     Delete the partition
                     Done setting up the partition

    <Go Back>
```

Image 13.1.12 - Debian installation partition summary.

Next, we are shown the partitions to be set on the drive. **Select the option Finish partitioning and write changes to disk.**

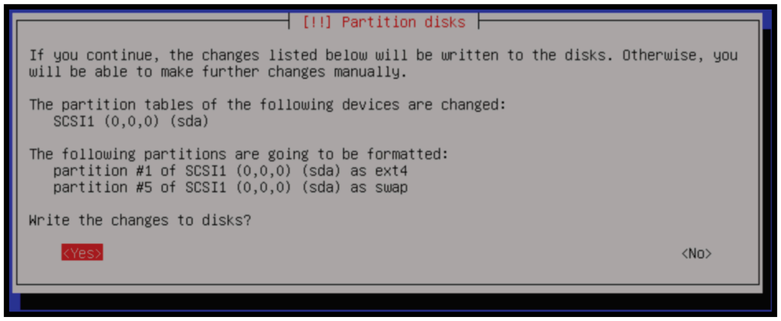


```
[!!] Partition disks

This is an overview of your currently configured partitions and mount points. Select a
partition to modify its settings (file system, mount point, etc.), a free space to create
partitions, or a device to initialize its partition table.

                 Guided partitioning
                 Configure software RAID
                 Configure the Logical Volume Manager
                 Configure encrypted volumes
                 Configure iSCSI volumes

                 SCSI1 (0,0,0) (sda) - 136.4 GB Msft Virtual Disk
                      #1  primary   20.0 GB    f  ext4          /
                          pri/log  115.3 GB       FREE SPACE
                      #5  logical    1.0 GB    f  swap          swap

                 Undo changes to partitions
                 Finish partitioning and write changes to disk

    <Go Back>
```

Image 13.1.13 - Debian installation disk partitions.

Later we will be using the rest of the **FREE SPACE** that is available.

Finally, we are shown a summary of the changes to be made on the drive. Select Yes to the question: **Write the changes to disks**.



```
[!!] Partition disks

If you continue, the changes listed below will be written to the disks. Otherwise, you
will be able to make further changes manually.

The partition tables of the following devices are changed:
   SCSI1 (0,0,0) (sda)

The following partitions are going to be formatted:
   partition #1 of SCSI1 (0,0,0) (sda) as ext4
   partition #5 of SCSI1 (0,0,0) (sda) as swap

Write the changes to disks?

    <Yes>                                                                  <No>
```

Image 13.1.14 - Debian installation drive changes summary.

You can then proceed with the installation as normal, you can follow the steps in the *Installation Section* for this guide. This **includes** the installation of **VitalPBX** using the **VPS installation script**.

Remember, the installation process with the partitioning needs to be done **twice**. **One for each server** in our high-availability environment.

With the installation done, we can start configuring our servers. It is a good idea to write down the networking information beforehand so we can work more orderly in our high-availability environment. For this guide, we will be using the following information.

Name	Primary Server	Secondary Server
Hostname	vitalpbx-primary.local	vitalpbx-secondary.local
IP Address	192.168.10.31	192.168.10.32
Netmask	255.255.255.0	255.255.255.0
Gateway	192.168.10.1	192.168.10.1
Primary DNS	8.8.8.8	8.8.8.8
Secondary DNS	8.8.4.4	8.8.4.4

Next, we will allow remote access using the root user on **both** servers. This will allow us to SSH login with the root user.

From the CLI, we will use nano to edit the sshd_config file.

```
root@debian:~# nano /etc/ssh/sshd_config
```

Change the following line.

```
#PermitRootLogin prohibit-password
```

With the following.

```
PermitRootLogIn yes
```

Save the changes and **exit** nano. Then, **restart** the sshd service.

```
root@debian:~# systemctl restart sshd
```

With this, you can now SSH login with the root user and password. This will make it easier to copy and paste the commands from this guide. Remember, this has to be done on **both** servers.

Once you are logged in with an SSH connection, we will set the static IP addresses for **both** servers. For this, we will use nano to modify the **interfaces** configuration file.

```
root@debian:~# nano /etc/network/interfaces
```

Here, change the following lines.

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet dchp
```

For the Primary Server, enter the following.

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.10.31
netmask 255.255.255.0
gateway 192.168.10.1
```

For the Secondary Server, enter the following.

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.10.32
netmask 255.255.255.0
gateway 192.168.10.1
```

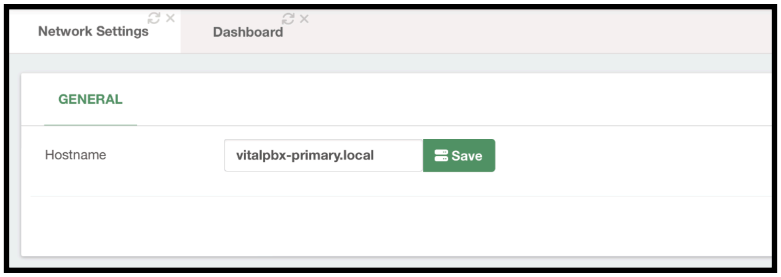
Now, reboot **both** servers and log in via SSH.

Note: *Your installation may have a different name for the primary network interface. Make sure that you are using the correct name for your interface.*

Next, we will install dependencies on **both** servers.

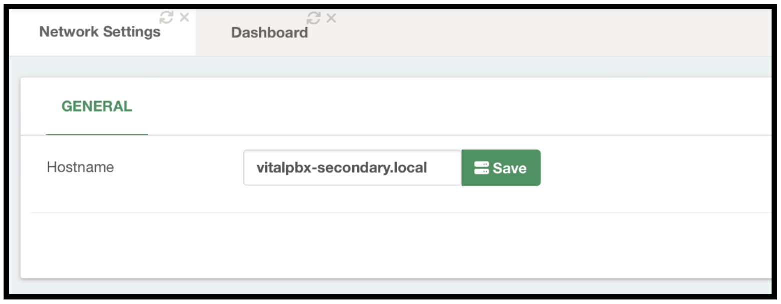
```
root@debian:~# apt -y install drbd-utils corosync pacemaker pcs chrony xfsprogs
```

With the dependencies installed, we must set the hostnames for **both** VitalPBX servers. For this, we go to **Admin > System Settings > Network Settings** in the VitalPBX Web UI.



Hostname: vitalpbx-primary.local

Image 13.1.15 - Primary Server hostname.



Hostname: vitalpbx-secondary.local

Image 13.1.16 - Secondary Server hostname.

After setting the hostname, click the green **Save** button. With the hostname set from the Web UI, we will now configure the hostnames in the **hosts file** for each server.

Set the **hostname** on the Primary Server with the following command.

```
root@debian:~# hostname vitalpbx-primary.local
```

And in the Secondary Server as follows.

```
root@debian:~# hostname vitalpbx-secondary.local
```

Afterward, on both servers modify the **hosts file** using nano.

```
root@debian:~# nano /etc/hosts
```

Add the following lines.

```
192.168.10.31 vitalpbx-primary.local

192.168.10.32 vitalpbx-secondary.local
```

This way, **both** servers will be able to see each other using their hostnames.

Now, we will create a **new partition** to allocate the rest of the available space for **both servers**. For this, we will use the **fdisk command**.

```
root@debian:~# fdisk /dev/sda
```

Answer as follows on the presented prompts.

```
Command (m for help): n

Partition type:

  p   primary (3 primary, 0 extended, 1 free)

  e   extended

Select (default e): p

Selected partition 3 (take note of the assigned partition number as we will need it later)

First sector (35155968-266338303, default 35155968): [Enter]

Last sector, +sectors or +size{K,M,G} (35155968-266338303, default 266338303): [Enter]

Using default value 266338303

Partition 4 of type Linux and of size 110.2 GiB is set

Command (m for help): t

Partition number (1-4, default 4): 3

Hex code (type L to list all codes): 8e

Changed type of partition 'Linux' to 'Linux LVM'

Command (m for help): w
```

Then restart **both** servers so that the new table is available.

```
root@debian:~# reboot
```

With the servers rebooted, we will proceed with the **HA (High Availability) cluster** configuration.

Now, we will create an **authorization key** for the access between both servers. This way, we can access both servers *without* entering credentials every time.

Create an **authorization key** in the Primary Server.

```
root@debian:~#  ssh-keygen -f /root/.ssh/id_rsa -t rsa -N '' >/dev/null

root@debian:~#  ssh-copy-id root@192.168.10.32

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

root@192.168.10.32's password: (remote server root’s password)


Number of key(s) added: 1


Now try logging into the machine, with:   "ssh 'root@192.168.10.32'"

and check to make sure that only the key(s) you wanted were added.

root@debian:~#
```

Next, create an **authorization key** in the Secondary Server.

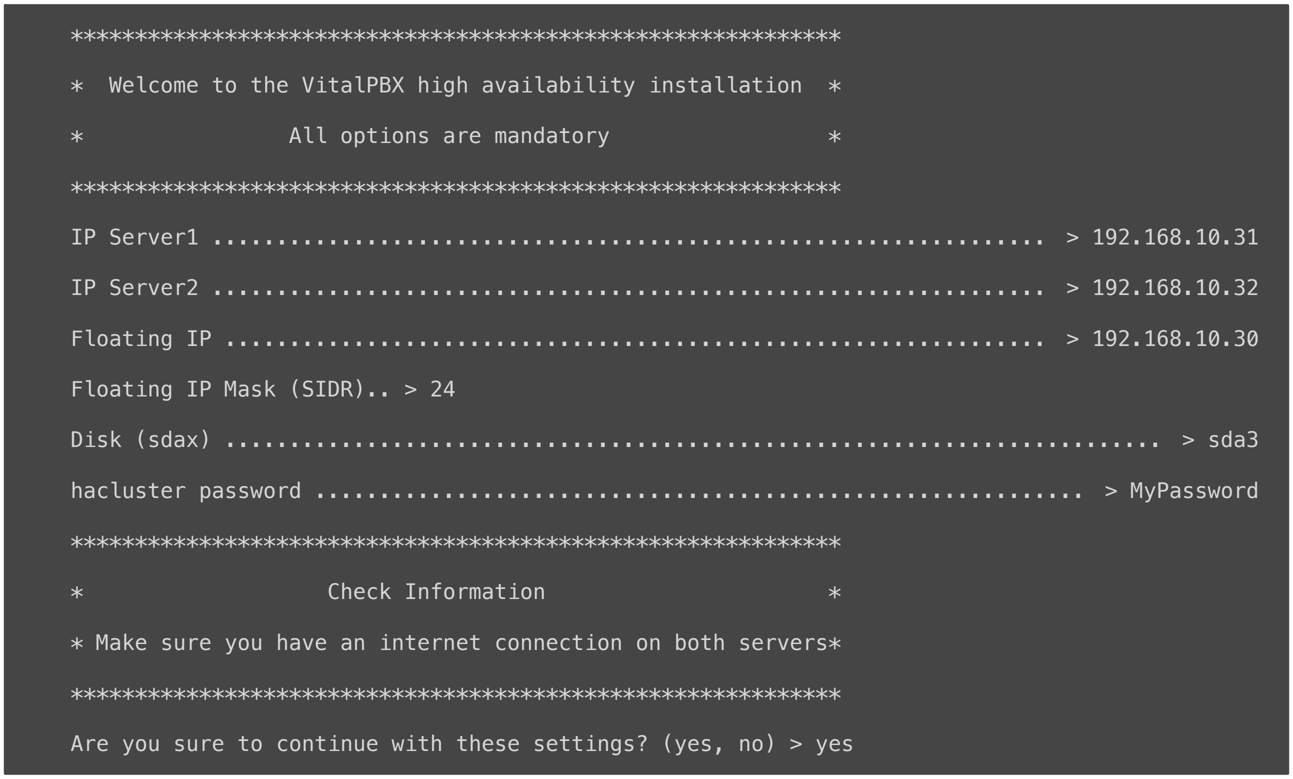
```
root@debian:~# ssh-keygen -f /root/.ssh/id_rsa -t rsa -N '' >/dev/null

root@debian:~# ssh-copy-id root@192.168.10.31

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

root@192.168.10.31's password: (remote server root’s password)


Number of key(s) added: 1


Now try logging into the machine, with:   "ssh 'root@192.168.10.31'"

and check to make sure that only the key(s) you wanted were added.

root@debian:~#
```

Now, we can proceed in two ways. One is using a **script** we made, or you can follow the **manual step-by-step**. If you proceed with the following script, you can skip all the steps until you reach the **add-on installation** in the lesson.

Afterward, you can download and run the following **script** from the Primary Server, using these commands.

```
root@debian:~# mkdir /usr/share/vitalpbx/ha

root@debian:~# cd /usr/share/vitalpbx/ha

root@debian:~# wget https://raw.githubusercontent.com/VitalPBX/vitalpbx4_drbd_ha/main/vpbxha.sh

root@debian:~# chmod +x vpbxha.sh

root@debian:~# ./vpbxha.sh
```

You will then be prompted to enter the information for the servers in the cluster.



```
************************************************************
*  Welcome to the VitalPBX high availability installation  *
*                All options are mandatory                 *
************************************************************
IP Server1 .................................................................. > 192.168.10.31
IP Server2 .................................................................. > 192.168.10.32
Floating IP ................................................................. > 192.168.10.30
Floating IP Mask (SIDR).. > 24
Disk (sdax) .......................................................................... > sda3
hacluster password ............................................................ > MyPassword
************************************************************
*                   Check Information                      *
* Make sure you have an internet connection on both servers*
************************************************************
Are you sure to continue with these settings? (yes, no) > yes
```

> ***Note:*** *The* ***hacluster*** *password can be anything of your liking. It does not have to be an existing password for any user in any node.*

> ***Note:*** *Before doing any high-availability testing, make sure that the data has finished synchronizing. To do this, use the* ***cat /proc/drbd*** *command.*

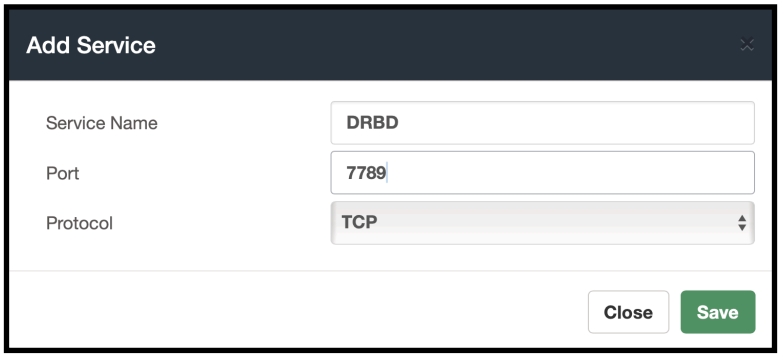
The script will start configuring the HA cluster for you. ***CONGRATULATIONS!*** You now have a high-availability environment with VitalPBX 4!

The following steps are if you want to proceed with the cluster configuration **manually**, rather than using the provided script. You can skip these steps if you decide to use the script and proceed to the **add-on installation** in the next lesson.

To configure the HA cluster **manually**, first, we need to configure the Firewall. This can be done by adding the services and rules from the **VitalPBX Web UI**. Here is the list of services we will configure. This needs to be configured in **both** servers.

Protocol	Port	Description
TCP	2224	This protocol is needed by the pcsd Web UI and required for node-to-node communication. It is crucial to open port 2224 in such a way that pcs from any node can talk to all nodes in the cluster, including itself. When using the Booth cluster ticket manager or a quorum device you must open port 2224 on all related hosts, such as Booth arbiters or the quorum device host.
TCP	3121	Pacemaker's crmd daemon on the full cluster nodes will contact the pacemaker_remoted daemon on Pacemaker Remote nodes at port 3121. If a separate interface is used for cluster communication, the port only needs to be open on that interface. At a minimum, the port should open on Pacemaker Remote nodes to full cluster nodes. Because users may convert a host between a full node and a remote node, or run a remote node inside a container using the host's network, it can be useful to open the port to all nodes. It is not necessary to open the port to any hosts other than nodes.
TCP	5403	Required on the quorum device host when using a quorum device with corosync-qnetd. The default value can be changed with the -p option of the corosync-qnetd command.
UDP	5404	Required on corosync nodes if corosync is configured for multicast UDP.
UDP	5405	Required on all corosync nodes (needed by corosync)
TCP	21064	Required on all nodes if the cluster contains any resources requiring DLM (such as clvm or GFS2)
TCP, UDP	9929	Required to be open on all cluster nodes and booth arbitrator nodes to connections from any of those same nodes when the Booth ticket manager is used to establish a multi-site cluster.
TCP	7789	Required by DRBD to synchronize information.

In the VitalPBX Web UI for both servers go to **Admin > Firewall > Services**. Add the services from the table above by clicking the **Add Service** button.


Add Service

Service Name: DRBD

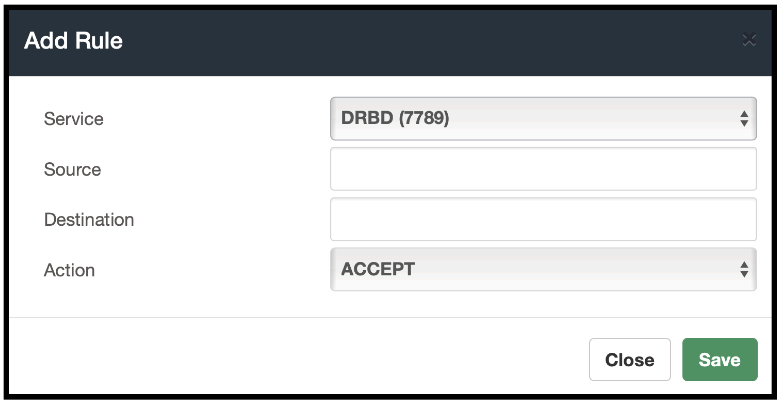
Port: 7789

Protocol: TCP

Image 13.1.17 - Adding HA cluster services to VitalPBX's Firewall.

With all the services added, **Apply Changes**.

Next, we go to **Admin > Firewall > Rules** to add the rules to **ACCEPT** all the services we just created.



Add Rule

Service: DRBD (7789)

Source:

Destination:

Action: ACCEPT

Image 13.1.18 - Adding HA cluster rules to VitalPBX's Firewall.

With all the rules added, **Apply Changes**. Remember, you need to add the services and rules to **both** servers' firewalls.

Now, Let's create a directory where we are going to mount the volume with all the information to be replicated in **both** servers.

```
root@debian:~# mkdir /vpbx_data
```

Afterward, we will format the new partition we made in **both** servers using the following commands.

```
root@debian:~# mke2fs -j /dev/sda3

root@debian:~# dd if=/dev/zero bs=1M count=500 of=/dev/sda3; sync
```

With all of this done, we can proceed to configure **DRBD** on both servers. Start by loading the module and enabling the service in **both** nodes using the following command.

```
root@debian:~# modprobe drbd

root@debian:~# systemctl enable drbd.service
```

Then, create a new **global_common.conf** file in **both** servers.

```
root@debian:~# mv /etc/drbd.d/global_common.conf /etc/drbd.d/global_common.conf.orig

root@debian:~# nano /etc/drbd.d/global_common.conf
```

Add the following content.

```
global {
  usage-count no;
}
common {
  net {
    protocol C;
  }
}
```

Save and **Exit** nano. Next, create a new configuration file called **drbd0.res** for the new resource named **drbd0** in **both** servers using nano.

```
root@debian:~# nano /etc/drbd.d/drbd0.res
```

Add the following content.

```
resource drbd0 {
startup {
        wfc-timeout  5;
    outdated-wfc-timeout 3;
        degr-wfc-timeout 3;
    outdated-wfc-timeout 2;
}
syncer {
    rate 10M;
    verify-alg md5;
}
net {
    after-sb-0pri discard-older-primary;
    after-sb-1pri discard-secondary;
    after-sb-2pri call-pri-lost-after-sb;
}
handlers {
    pri-lost-after-sb "/sbin/reboot";
}
```

```
    on vitalpbx-primary.local {
        device /dev/drbd0;
        disk /dev/sda3;
        address 192.168.10.31:7789;
        meta-disk internal;
    }
    on vitalpbx-secondary.local {
        device /dev/drbd0;
        disk /dev/sda3;
        address 192.168.10.32:7789;
        meta-disk internal;
        }
    }
```

Save and **Exit** nano.

> ***Note:*** *Although the access interfaces can be used, which in this case is ETH0. It is recommended to use an interface (i.e. ETH1) for synchronization, this interface must be directly connected between both servers.*

Now, initialize the metadata storage in each node by executing the following command in **both** servers.

```
root@debian:~# drbdadm create-md drbd0
Writing meta data...
New drbd meta data block successfully created.
```

Afterward, define the Primary Server as the **DRBD primary node first**.

```
root@debian:~# drbdadm up drbd0
root@debian:~# drbdadm primary drbd0 --force
```

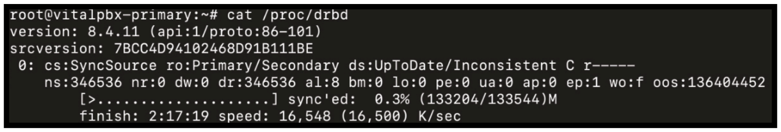
Then, on the Secondary Server, run the following command to start the **drbd0**.

```
root@debian:~# drbdadm up drbd0
```

You can check the current status of the synchronization while it is being performed, using the following command.

```
root@debian:~# cat /proc/drbd
```

Here is an example of the output of this command.



```
root@vitalpbx-primary:~# cat /proc/drbd
version: 8.4.11 (api:1/proto:86-101)
srcversion: 7BCC4D94102468D91B111BE
 0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r-----
    ns:346536 nr:0 dw:0 dr:346536 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:136404452
        [>....................] sync'ed:  0.3% (133204/133544)M
        finish: 2:17:19 speed: 16,548 (16,500) K/sec
```

Image 13.1.18 - DRBD synchronization progress.

In order to test the DRBD functionality, we must create a file system, mount the volume, write some data in the Primary Server, and finally switch the primary node to the Secondary Server.

Run the following commands in the Primary Server to create an **XFS file system** in the **/dev/drbd0** directory, and mount it to the **/vpbx_data** directory.

```
root@debian:~# mkfs.xfs /dev/drbd0

root@debian:~# mount /dev/drbd0 /vpbx_data
```

Create some data using the following command in the Primary Server.

```
root@debian:~# touch /vpbx_data/file{1..5}
```

Run the following command to list the content of the /vpbx_data directory.

```
root@debian:~# ls -l /vpbx_data
```

The command will return the following list.

```
total 0

-rw-r--r-- 1 root root 0 Nov 17 11:28 file1

-rw-r--r-- 1 root root 0 Nov 17 11:28 file2

-rw-r--r-- 1 root root 0 Nov 17 11:28 file3

-rw-r--r-- 1 root root 0 Nov 17 11:28 file4

-rw-r--r-- 1 root root 0 Nov 17 11:28 file5
```

Now, let's switch the primary node **"Primary Server"** to the secondary node **"Secondary Server"** to check if the data replication works.

We will need to unmount the volume **drbd0** in the Primary Server and change it from the primary node to the secondary node, and we will turn the Secondary Server into the primary node.

In the Primary Server, run the following commands.

```
root@debian:~# umount /vpbx_data

root@debian:~# drbdadm secondary drbd0
```

Change the secondary node to the primary node, by running this command on the Secondary Server.

```
root@debian:~# drbdadm primary drbd0 --force
```

In the Secondary Server, mount the volume and check if the data is available with the following command.

```
root@debian:~# mount /dev/drbd0 /vpbx_data

root@debian:~# ls -l  /vpbx_data
```

The command should return something like this.

```
total 0

-rw-r--r-- 1 root root 0 Nov 17 11:28 file1

-rw-r--r-- 1 root root 0 Nov 17 11:28 file2

-rw-r--r-- 1 root root 0 Nov 17 11:28 file3

-rw-r--r-- 1 root root 0 Nov 17 11:28 file4

-rw-r--r-- 1 root root 0 Nov 17 11:28 file5
```

As you can see the data is being replicated, since these files were created in the Primary Server, and we are seeing them in the Secondary Server.

Now, let's normalize the Secondary Server. Unmount the volume **drbd0** and set it as the secondary node. In the Secondary Server, run the following commands.

```
root@debian:~# umount /vpbx_data

root@debian:~# drbdadm secondary drbd0
```

Then, normalize the Primary Server. Turn it into the primary node, and mount the **drbd0** volume to the **/vpbx_data** directory. In the Primary Server, run the following commands.

```
root@debian:~# drbdadm primary drbd0

root@debian:~# mount /dev/drbd0 /vpbx_data
```

With the replication working, let's configure the cluster for high availability. Create a password for the hacluster user on **both** servers.

```
root@debian:~# echo hacluster:Mypassword | chpasswd
```

> ***Note:*** *The hacluster password can be anything of your liking. This does not have to be a password for the root or any other user.*

Then, start the PCS service on **both** servers, using the following command.

```
    root@debian:~# systemctl start pcsd
```

We must enable the **PCS, Corosync, and Pacemaker services** to start on **both** servers, with the following commands.

```
    root@debian:~# systemctl enable pcsd.service

    root@debian:~# systemctl enable corosync.service

    root@debian:~# systemctl enable pacemaker.service
```

Now, let's authenticate as the **hacluster** user using **PCS Auth** in the Primary Server. Enter the following commands.

```
    root@debian:~# pcs cluster destroy

    root@debian:~# pcs host auth vitalpbx-primary.local vitalpbx-secondary.local -u hacluster -p Mypassword
```

The command should return the following.

```
    vitalpbx-primary.local: Authorized

    vitalpbx-secondary.local: Authorized
```

Next, use the **PCS cluster setup** command in the Primary Server to generate and synchronize the corosync configuration.

```
    root@debian:~# pcs cluster setup cluster_vitalpbx vitalpbx-primary.local vitalpbx-
secondary.local --force
```

Start the cluster in the Primary Server, with the following commands.

```
    root@debian:~# pcs cluster start --all

    root@debian:~# pcs cluster enable --all

    root@debian:~# pcs property set stonith-enabled=false

    root@debian:~# pcs property set no-quorum-policy=ignore
```

> ***Note:*** *It's recommended to* ***prevent resources from moving after recovery****. In most circumstances, it is highly desirable to prevent healthy resources from being moved around the cluster. Moving resources always requires a period of downtime. For complex services such as databases, this period can be quite long.*

To prevent resources from moving after recovery, run this command in the Primary Server.

```
    root@debian:~# pcs resource defaults update resource-stickiness=INFINITY
```

Now, create the resource for the use of a **Floating IP Address**, with the following commands in the Primary Server.

```
    root@debian:~# pcs resource create ClusterIP ocf:heartbeat:IPaddr2 ip=192.168.10.30
cidr_netmask=24 op monitor interval=30s on-fail=restart

    root@debian:~# pcs cluster cib drbd_cfg

    root@debian:~# pcs cluster cib-push drbd_cfg --config
```

Then, create the **resource** to use **DRBD.** use the following commands in the Primary Server.

```
    root@debian:~# pcs cluster cib drbd_cfg

    root@debian:~# pcs -f drbd_cfg resource create DrbdData ocf:linbit:drbd drbd_resource=drbd0
op monitor interval=60s

    root@debian:~# pcs -f drbd_cfg resource promotable DrbdData promoted-max=1 promoted-node-
max=1 clone-max=2 clone-node-max=1 notify=true

    root@debian:~# pcs cluster cib fs_cfg

    root@debian:~# pcs cluster cib-push drbd_cfg --config
```

Next, create the **file system** for the **automated mount point**, using the following commands in the Primary Server.

```
    root@debian:~# pcs cluster cib fs_cfg

    root@debian:~# pcs -f fs_cfg resource create DrbdFS Filesystem device="/dev/drbd0"
directory="/vpbx_data" fstype="xfs"

    root@debian:~# pcs -f fs_cfg constraint colocation add DrbdFS with DrbdData-clone INFINITY
with-rsc-role=Master

    root@debian:~# pcs -f fs_cfg constraint order promote DrbdData-clone then start DrbdFS

    root@debian:~# pcs -f fs_cfg constraint colocation add DrbdFS with ClusterIP INFINITY

    root@debian:~# pcs -f fs_cfg constraint order DrbdData-clone then DrbdFS

    root@debian:~# pcs cluster cib-push fs_cfg --config
```

Stop and disable all services on **both** servers, using the following commands.

```
    root@debian:~# systemctl stop mariadb

    root@debian:~# systemctl disable mariadb

    root@debian:~# systemctl stop fail2ban

    root@debian:~# systemctl disable fail2ban

    root@debian:~# systemctl stop asterisk

    root@debian:~# systemctl disable asterisk

    root@debian:~# systemctl stop vpbx-monitor

    root@debian:~# systemctl disable vpbx-monitor
```

Create the **resource** for the use of **MariaDB** in the Primary Server, using the following commands.

```
    root@debian:~# mkdir /vpbx_data/mysql

    root@debian:~# mkdir /vpbx_data/mysql/data

    root@debian:~# cp -aR /var/lib/mysql/* /vpbx_data/mysql/data

    root@debian:~# chown -R mysql:mysql /vpbx_data/mysql

    root@debian:~# sed -i 's/var\/lib\/mysql/vpbx_data\/mysql\/data/g' /etc/mysql/mariadb.conf.d/
50-server.cnf
```

Change the **MariaDB Path** on the Secondary Server as well, using the following command.

```
    root@debian:~# sed -i 's/var\/lib\/mysql/vpbx_data\/mysql\/data/g' /etc/mysql/mariadb.conf.d/
50-server.cnf
```

Now, run the following commands in the Primary Server to create the MariaDB resource.

```
    root@debian:~# pcs resource create mysql service:mariadb op monitor interval=30s

    root@debian:~# pcs cluster cib fs_cfg

    root@debian:~# pcs cluster cib-push fs_cfg --config

    root@debian:~# pcs -f fs_cfg constraint colocation add mysql with ClusterIP INFINITY

    root@debian:~# pcs -f fs_cfg constraint order DrbdFS then mysql

    root@debian:~# pcs cluster cib-push fs_cfg --config
```

Set the paths for the Asterisk service in **both** servers, using the following commands.

```
    root@debian:~# sed -i 's/RestartSec=10/RestartSec=1/g'  /usr/lib/systemd/system/
asterisk.service

    root@debian:~# sed -i 's/Wants=mariadb.service/#Wants=mariadb.service/g'  /usr/lib/systemd/
system/asterisk.service

    root@debian:~# sed -i 's/After=mariadb.service/#After=mariadb.service/g'  /usr/lib/systemd/
system/asterisk.service
```

Next, create the **resource** for **Asterisk** in the Primary Server, using the following commands.

```
root@debian:~# pcs resource create asterisk service:asterisk op monitor interval=30s

root@debian:~# pcs cluster cib fs_cfg

root@debian:~# pcs cluster cib-push fs_cfg --config

root@debian:~# pcs -f fs_cfg constraint colocation add asterisk with ClusterIP INFINITY

root@debian:~# pcs -f fs_cfg constraint order mysql then asterisk

root@debian:~# pcs cluster cib-push fs_cfg --config

root@debian:~# pcs resource update asterisk op stop timeout=120s

root@debian:~# pcs resource update asterisk op start timeout=120s

root@debian:~# pcs resource update asterisk op restart timeout=120s
```

Copy the **Asterisk and VitalPBX folders and files** to the **DRBD partition** in the Primary Server using the following commands.

```
root@debian:~# tar -zcvf var-asterisk.tgz /var/log/asterisk

root@debian:~# tar -zcvf var-lib-asterisk.tgz /var/lib/asterisk

root@debian:~# tar -zcvf var-lib-vitalpbx.tgz /var/lib/vitalpbx

root@debian:~# tar -zcvf etc-vitalpbx.tgz /etc/vitalpbx

root@debian:~# tar -zcvf usr-lib-asterisk.tgz /usr/lib/asterisk

root@debian:~# tar -zcvf var-spool-asterisk.tgz /var/spool/asterisk

root@debian:~# tar -zcvf etc-asterisk.tgz /etc/asterisk


root@debian:~# tar xvfz var-asterisk.tgz -C /vpbx_data

root@debian:~# tar xvfz var-lib-asterisk.tgz -C /vpbx_data

root@debian:~# tar xvfz var-lib-vitalpbx.tgz -C /vpbx_data

root@debian:~# tar xvfz etc-vitalpbx.tgz -C /vpbx_data

root@debian:~# tar xvfz usr-lib-asterisk.tgz -C /vpbx_data

root@debian:~# tar xvfz var-spool-asterisk.tgz -C /vpbx_data

root@debian:~# tar xvfz etc-asterisk.tgz -C /vpbx_data

root@debian:~# chmod -R 775 /vpbx_data/var/log/asterisk


root@debian:~# rm -rf /var/log/asterisk

root@debian:~# rm -rf /var/lib/asterisk

root@debian:~# rm -rf /var/lib/vitalpbx

root@debian:~# rm -rf /etc/vitalpbx

root@debian:~# rm -rf /usr/lib/asterisk
```

```
root@debian:~# rm -rf /var/spool/asterisk

root@debian:~# rm -rf /etc/asterisk


root@debian:~# ln -s /vpbx_data/var/log/asterisk /var/log/asterisk

root@debian:~# ln -s /vpbx_data/var/lib/asterisk /var/lib/asterisk

root@debian:~# ln -s /vpbx_data/var/lib/vitalpbx /var/lib/vitalpbx

root@debian:~# ln -s /vpbx_data/etc/vitalpbx /etc/vitalpbx

root@debian:~# ln -s /vpbx_data/usr/lib/asterisk /usr/lib/asterisk

root@debian:~# ln -s /vpbx_data/var/spool/asterisk /var/spool/asterisk

root@debian:~# ln -s /vpbx_data/etc/asterisk /etc/asterisk


root@debian:~# rm -rf var-asterisk.tgz

root@debian:~# rm -rf var-lib-asterisk.tgz

root@debian:~# rm -rf var-lib-vitalpbx.tgz

root@debian:~# rm -rf etc-vitalpbx.tgz

root@debian:~# rm -rf usr-lib-asterisk.tgz

root@debian:~# rm -rf var-spool-asterisk.tgz

root@debian:~# rm -rf etc-asterisk.tgz
```

Now, configure the **symbolic links** on the Secondary Server with the following commands.

```
root@debian:~# rm -rf /var/log/asterisk

root@debian:~# rm -rf /var/lib/asterisk

root@debian:~# rm -rf /var/lib/vitalpbx

root@debian:~# rm -rf /etc/vitalpbx

root@debian:~# rm -rf /usr/lib/asterisk

root@debian:~# rm -rf /var/spool/asterisk

root@debian:~# rm -rf /etc/asterisk


root@debian:~# ln -s /vpbx_data/var/log/asterisk /var/log/asterisk

root@debian:~# ln -s /vpbx_data/var/lib/asterisk /var/lib/asterisk

root@debian:~# ln -s /vpbx_data/var/lib/vitalpbx /var/lib/vitalpbx

root@debian:~# ln -s /vpbx_data/etc/vitalpbx /etc/vitalpbx

root@debian:~# ln -s /vpbx_data/usr/lib/asterisk /usr/lib/asterisk

root@debian:~# ln -s /vpbx_data/var/spool/asterisk /var/spool/asterisk

root@debian:~# ln -s /vpbx_data/etc/asterisk /etc/asterisk
```

Then, create the **VitalPBX Service** in the Primary Server, using the following commands.

```
root@debian:~# pcs resource create vpbx-monitor service:vpbx-monitor op monitor interval=30s

root@debian:~# pcs cluster cib fs_cfg

root@debian:~# pcs cluster cib-push fs_cfg --config

root@debian:~# pcs -f fs_cfg constraint colocation add vpbx-monitor with ClusterIP INFINITY

root@debian:~# pcs -f fs_cfg constraint order asterisk then vpbx-monitor

root@debian:~# pcs cluster cib-push fs_cfg --config
```

Create the **Fail2Ban Service** in the Primary Server, using the following commands.

```
root@debian:~# pcs resource create fail2ban service:fail2ban op monitor interval=30s

root@debian:~# pcs cluster cib fs_cfg

root@debian:~# pcs cluster cib-push fs_cfg --config

root@debian:~# pcs -f fs_cfg constraint colocation add fail2ban with ClusterIP INFINITY

root@debian:~# pcs -f fs_cfg constraint order asterisk then fail2ban

root@debian:~# pcs cluster cib-push fs_cfg --config
```

Initialize the **Corosync and Pacemaker services** in the Secondary Server with the following commands.

```
root@debian:~# systemctl restart corosync.service

root@debian:~# systemctl restart pacemaker.service
```

Note: *All configurations are stored in the* ***/var/lib/pacemaker/cib/cib.xml*** *file.*

Now let’s see the cluster status by running the following command in the Primary Server.

```
root@debian:~# pcs status resources
```

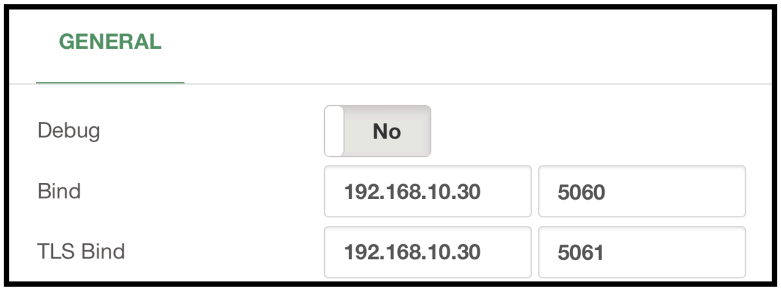
This command will return the following.

```
  * ClusterIP(ocf::heartbeat:IPaddr2):      Started vitalpbx-primary.local

  * Clone Set: DrbdData-clone [DrbdData] (promotable):

    * Masters: [ vitalpbx-primary.local ]

    * Slaves: [ vitalpbx-secondary.local ]

  * DrbdFS   (ocf::heartbeat:Filesystem):   Started vitalpbx-primary.local

  * mysql    (service:mysql):       Started vitalpbx-primary.local

  * asterisk (service:asterisk):    Started vitalpbx-primary.local

  * vpbx-monitor     (service:vpbx-monitor):       Started vitalpbx-primary.local

  * fail2ban (service:fail2ban):    Started vitalpbx-primary.local
```

> ***Note:*** *Before doing any high availability testing, make sure that the data has finished synchronizing. To do this, use the* ***cat /proc/drbd*** *command.*

With our cluster configured, we now must configure the **bind address**. Managing the bind address is critical when using multiple IP addresses on the same NIC (Network Interface Card). This is our case when using a Floating IP address in this HA cluster. In this circumstance, Asterisk has a habit of listening for SIP/IAX on the virtual IP address, but replying on the base address of the NIC, causing phones and trunks to fail to register.

In the Primary Server, go to **Settings > Technology Settings >PJSIP Settings**, and configure the Floating IP address in the Bind and TLS Bind fields.



Debug: No

Bind: 192.168.10.30 5060

TLS Bind: 192.168.10.30 5061

Image 13.1.19 - Bind and TLS Bind settings in PJSIP Settings for HA cluster.

Now that the Bind address is set. We will create the **Bascul** command in **both** servers. This command will allow us to easily move the services between nodes. This will essentially allow us to move between the Primary and Secondary Servers.

To start creating the bascul command, we can begin by downloading the following file using **wget** on **both** servers.

```
root@debian:~# wget https://raw.githubusercontent.com/VitalPBX/vitalpbx4_drbd_ha/main/bascul
```

Or we can create it from scratch using nano on **both** servers.

```
    root@debian:~# nano bascul
```

And add the following content.

```
    #!/bin/bash
    # This code is the property of VitalPBX, LLC
    # License: Proprietary
    # Date: 1-Aug-2023
    # Change the status of the servers, the Master goes to Standby and the Standby becomes
Master.
    #Function to draw a progress bar
    #You must pass as an argument the number of seconds that the progress bar will run
    #progress-bar 10 --> It will generate a progress bar that will run per 10 seconds
    set -e
    progress-bar() {
            local duration=${1}
            already_done() { for ((done=0; done<$elapsed; done++)); do printf ">"; done }
            remaining() { for ((remain=$elapsed; remain<$duration; remain++)); do printf " ";
done }
            percentage() { printf "| %s%%" $(( (($elapsed)*100)/($duration)*100/100 )); }
            clean_line() { printf "\r"; }
            for (( elapsed=1; elapsed<=$duration; elapsed++ )); do
                    already_done; remaining; percentage
                    sleep 1
                    clean_line
            done
            clean_line
    }
    server_a=`pcs status | awk 'NR==11 {print $4}'`
    server_b=`pcs status | awk 'NR==11 {print $5}'`
    server_master=`pcs status resources | awk 'NR==1 {print $5}'`
    #Perform validations
    if [ "${server_a}" = "" ] || [ "${server_b}" = "" ]
    then
```

```
        echo -e "\e[41m There are problems with high availability, please check with the command
*pcs status* (we recommend applying the command *pcs cluster unstandby* in both servers) \e[0m"

        exit;

    fi

    if [[ "${server_master}" = "${server_a}" ]]; then

            host_master=$server_a

            host_standby=$server_b

    else

            host_master=$server_b

            host_standby=$server_a

    fi

    arg=$1

    if [ "$arg" = 'yes' ] ;then

            perform_bascul='yes'

    fi

    # Print a warning message and ask the user if they want to continue

    echo -e "************************************************************"

    echo -e "*     Change the roles of servers in high availability     *"

    echo -e "*\e[41m WARNING-WARNING-WARNING-WARNING-WARNING-WARNING-WARNING  \e[0m*"

    echo -e "*All calls in progress will be lost and the system will be *"

    echo -e "*     be in an unavailable state for a few seconds.        *"

    echo -e "************************************************************"

    #Perform a loop until the users confirm if they want to proceed or not

    while [[ $perform_bascul != yes && $perform_bascul != no ]]; do

            read -p "Are you sure to switch from $host_master to $host_standby? (yes, no) > "
perform_bascul

    done

    if [[ "${perform_bascul}" = "yes" ]]; then

            #Unstandby both nodes

            pcs node unstandby $host_master

            pcs node unstandby $host_standby

            #Do a loop per resource

            pcs status resources | awk '{print $2}' | while read -r resource ; do

                    #Skip moving the ClusterIP resource, it will be moved at the end
                    if [[ "${resource}" != "ClusterIP" ]] && [[ "${resource}" != "Clone" ]] && [[ "${resource}" != "Masters:" ]] && [[ "${resource}" != "Slaves:" ]]; then

                            echo "Moving ${resource} from ${host_master} to ${host_standby}"

                            pcs resource move ${resource} ${host_standby}

                    fi

            done

            sleep 5 && pcs node standby $host_master & #Standby current Master node after five seconds

            sleep 20 && pcs node unstandby $host_master & #Automatically Unstandby current Master node after$

            #Move the ClusterIP resource to the standby node

            echo "Moving ClusterIP from ${host_master} to ${host_standby}"

            pcs resource move ClusterIP ${host_standby}

            #End the script

            echo "Turning ${host_standby} to Master"

            progress-bar 10

            echo "Done"

    else

            echo "Nothing to do, bye, bye"

    fi

    sleep 15

    role
```

Save and **Exit** nano. Next, add permissions and move to the **/usr/local/bin** directory using the following commands in **both** servers.

```
    root@debian:~# chmod +x bascul

    root@debian:~# mv bascul /usr/local/bin
```

Now, create the Role command in **both** servers. You can download the following file using **wget**.

```
    root@debian:~# wget https://raw.githubusercontent.com/VitalPBX/vitalpbx4_drbd_ha/main/role
```

Or you can create the file using nano on **both** servers.

```
    root@debian:~# nano role
```

Add the following content.

```
    #!/bin/bash
    # This code is the property of VitalPBX, LLC
    # License: Proprietary
    # Date: 10-oct-2022
    # Show the Role of the Server.
    #Bash Color Codes
    green="\033[00;32m"
    txtrst="\033[00;0m"
    linux_ver=`cat /etc/os-release | grep -e PRETTY_NAME | awk -F '=' '{print $2}' | xargs`
    vpbx_version=`aptitude versions vitalpbx | awk '{ print $2 }'`
    server_master=`pcs status resources | awk 'NR==1 {print $5}'`
    host=`hostname`
    if [[ "${server_master}" = "${host}" ]]; then
            server_mode="Master"
    else
            server_mode="Standby"
    fi
    logo='
     _    _ _           _ ______ ______ _    _
    | |  | (_)_        | (_____ (____  \ \  / /
    | |  | |_| |_  ____| |_____) )___)  ) \/ /
     \ \/ /| |  _)/ _ | | ____/  __  ( )  (
      \  / | | |_( ( | | | |    | |__)  ) /\ \\
       \/  |_|\___)_||_|_|_|    |______/_/  \_\\
    '
    echo -e "
    ${green}
    ${logo}
    ${txtrst}
     Role           : $server_mode
     Version        : ${vpbx_version//[[:space:]]}
     Asterisk       : `asterisk -rx "core show version" 2>/dev/null| grep -ohe 'Asterisk
[0-9.]*'`
```

```
     Linux Version  : ${linux_ver}

     Welcome to     : `hostname`

     Uptime         : `uptime | grep -ohe 'up .*' | sed 's/up //g' | awk -F "," '{print $1}'`

     Load           : `uptime | grep -ohe 'load average[s:][: ].*' | awk '{ print "Last Minute: "
$3" Last 5 Minutes: "$4" Last 15 Minutes "$5 }'`

     Users          : `uptime | grep -ohe '[0-9.*] user[s,]'`

     IP Address     : ${green}`ip addr | sed -En 's/127.0.0.1//;s/.*inet (addr:)?(([0-9]*\.){3}
[0-9]*).*/\2/p' | xargs `${txtrst}

     Clock          :`timedatectl | sed -n '/Local time/ s/^[ \t]*Local time:\(.*$\)/\1/p'`

     NTP Sync.      :`timedatectl |awk -F: '/NTP service/ {print $2}'`

    "

    echo -e ""

    echo -e "************************************************************"

    echo -e "*                  Servers Status                          *"

    echo -e "************************************************************"

    echo -e "Master"

    pcs status resources

    echo -e ""

    echo -e "Servers Status"

    pcs cluster pcsd-status
```

Save and **Exit** nano. Next, we copy it to the **/etc/profile.d/** and permissions directory using the following commands on **both** servers.

```
    root@debian:~# cp -rf role /etc/profile.d/vitalwelcome.sh

    root@debian:~# chmod 755 /etc/profile.d/vitalwelcome.sh
```

Now, add execution permissions and move to the /usr/local/bin directory using the following commands on **both** servers.

```
    root@debian:~# chmod +x role

    root@debian:~# mv role /usr/local/bin
```

Afterward, we will create the **drbdsplit** command in **both** servers. **Split-Brain** can be caused by intervention by cluster management software or human error during a period of failure for network links between cluster nodes, causing both nodes to switch to the primary role while disconnected. Split-brain occurs when both high availability nodes switch into the primary role while disconnected. This behavior can allow data to be modified on either node without being replicated on the peer, leading to two diverging sets of data on each node, which can be difficult to merge. The drbdsplit command allows us to recover from split-brain in case it ever happens to us.

To create the drbdsplit command, we can download the following file using the wget command on **both** servers.

```
    root@debian:~# wget https://raw.githubusercontent.com/VitalPBX/vitalpbx4_drbd_ha/main/
drbdsplit
```

Or we can create it from scratch using nano on **both** servers.

```
    root@debian:~# nano drbdsplit
```

Add the following content.

```
    #!/bin/bash

    set -e

    # This code is the property of VitalPBX, LLC

    # License: Proprietary

    # Date: 1-Agu-2023

    # DRBD split-brain recovery

    #

    drbdadm secondary drbd0

    drbdadm disconnect drbd0

    drbdadm -- --discard-my-data connect drbd0

    echo "Disk Status"

    drbdadm status
```

Save and **Exit** nano. Now, add permissions and move it to the **/usr/local/bin** directory using the following commands on **both** servers.

```
    root@debian:~# chmod +x drbdsplit

    root@debian:~# mv drbdsplit /usr/local/bin
```

With this, you have a full high availability environment! **CONGRATULATIONS**, you now have high availability with VitalPBX 4.

Section 13, Lesson 2 - Installing Add-Ons in High Availability

Now we can see **how to install add-ons in a high-availability environment**. We will take an in-depth look into the process of adding **add-ons that have a service running** on the servers. This includes the **Sonata Suite, VitXi, OpenVPN, and Maintenance** add-on modules. These add-ons require special steps for installation in high-availability environments.

Let’s begin with the Sonata Suite. The first application of Sonata Suite we will look into installing is **Sonata Switchboard**.

Before proceeding, make sure that the Primary Server is set as a **Master**. You can use the **role command** to verify this.

First, go to the **Floating IP Address** on your browser. There, install Sonata Switchboard as normal from **Admin > Add-Ons > Add-Ons**. Then, go to the Sonata Switchboard URL, which is ***https://FloatingIPAddress/switchboard*** by default. Follow the installation wizard and finish installing Sonata Switchboard.

Next, **copy the files** that we are going to replicate from the Primary Server to the Secondary Server. Run the following commands in the Primary Server.

```
root@debian:~# tar -zcvf switchboard-settings.tgz /etc/switchboard/settings.conf

root@debian:~# tar xvfz switchboard-settings.tgz -C /vpbx_data/

root@debian:~# rm -rf /etc/switchboard/settings.conf

root@debian:~# ln -s /vpbx_data/etc/switchboard/settings.conf  /etc/switchboard/settings.conf

root@debian:~# rm -rf switchboard-settings.tgz
```

Afterward, run the **bascul command** on the Primary Server to turn the Secondary Server into the Master.

```
root@debian:~# bascul
```

Now, go back to the **Floating IP Address** on your browser. You will now be navigating on the Secondary Server. Here, install Sonata Switchboard as you did in the Primary Server by installing the add-on and following the Wizard.

Then, create the symbolic link for the **settings.conf** file in the Secondary Server.

```
root@debian:~# rm -rf /etc/switchboard/settings.conf

root@debian:~# ln -s /vpbx_data/etc/switchboard/settings.conf  /etc/switchboard/settings.conf
```

Finally, execute the **bascul command** once again from the Primary Server to return the cluster back to normal where the Primary Server is set as the Master.

```
root@debian:~# bascul
```

With this, you now have Sonata Switchboard installed in your high-availability environment.

Let’s move on to **Sonata Recordings**. The process will be very similar to the Sonata Switchboard. With the difference of adding the Sonata Recordings service automation at the end.

First, go to the **Floating IP Address** on your browser. Here, install Sonata Recordings from the add-ons module under **Admin > Add-Ons > Add-Ons**. Once installed, go to the Sonata

Recordings URL, which is ***https://FloatingIPAddress/recordings*** by default. There, follow the wizard to finish with the installation.

Next, **copy the files** that we are going to replicate from the Primary Server to the Secondary Server. Run the following commands on the Primary Server.

```
root@debian:~# tar -zcvf recording-settings.tgz /etc/rec-manager/settings.conf

root@debian:~# tar xvfz recording-settings.tgz -C /vpbx_data/

root@debian:~# rm -rf /etc/rec-manager/settings.conf

root@debian:~# ln -s /vpbx_data/etc/rec-manager/settings.conf  /etc/rec-manager/settings.conf

root@debian:~# rm -rf recording-settings.tgz
```

Now, execute the **bascul command** on the Primary Server to turn the Secondary Server into the Master.

```
root@vitalpbx-master:~# bascul
```

Afterward, go back to the **Floating IP Address** on your browser. You will now be navigating the Secondary Server. Here, **install** Sonata Recordings as you did on the Primary Server.

Then, create the symbolic link for the **settings.conf** file on the Secondary Server.

```
root@debian:~# rm -rf /etc/rec-manager/settings.conf

root@debian:~# ln -s /vpbx_data/etc/rec-manager/settings.conf  /etc/rec-manager/settings.conf
```

Now, execute the **bascul command** on the Primary Server to return the cluster to normal where the Primary Server is the Master.

```
root@debian:~# bascul
```

Finally, add the **Sonata Recordings service** automation. Run the following commands on **both** servers.

```
root@debian:~# systemctl stop recordings

root@debian:~# systemctl disable recordings
```

Then, run the following commands on the Primary Server

```
    root@debian:~# pcs resource create sonata-recordings service:recordings op monitor
interval=30s

    root@debian:~# pcs cluster cib fs_cfg

    root@debian:~# pcs cluster cib-push fs_cfg --config

    root@debian:~# pcs -f fs_cfg constraint colocation add sonata-recordings with ClusterIP
INFINITY

    root@debian:~# pcs -f fs_cfg constraint order vpbx-monitor then sonata-recordings

    root@debian:~# pcs cluster cib-push fs_cfg --config
```

With this, you now have Sonata Recordings installed in your high-availability environment.

Let’s move towards **Sonata Billing**. The process will be exactly the same as the Sonata Switchboard.

First, go to the **Floating IP Address** and install Sonata Billing from the add-ons module under **Admin > Add-Ons > Add-Ons**. Once installed, go to the Sonata Billing URL, which is ***https://FloatingIPAddress/billing*** by default. There, follow the wizard and finish the installation.

Next, **copy the files** that we are going to replicate from the Primary Server to the Secondary Server. Run the following commands on the Primary Server.

```
    root@debian:~# tar -zcvf billing-settings.tgz /etc/billing/settings.conf

    root@debian:~# tar xvfz billing-settings.tgz -C /vpbx_data/

    root@debian:~# rm -rf /etc/billing/settings.conf

    root@debian:~# ln -s /vpbx_data/etc/billing/settings.conf  /etc/billing/settings.conf

    root@debian:~# rm -rf billing-settings.tgz
```

Now, execute the **bascul command** on the Primary Server to turn the Secondary Server into the Master.

```
    root@debian:~# bascul
```

Go back to the **Floating IP Address** on your browser. You will now be navigating the Secondary Server. Here, install Sonata Billing as you did in the Primary Server.

Then, create the symbolic link for the **settings.conf** file on the Secondary Server.

```
    root@debian:~# rm -rf /etc/billing/settings.conf

    root@debian:~# ln -s /vpbx_data/etc/billing/settings.conf  /etc/billing/settings.conf
```

Finally, execute the **bascul command** once again from the Primary Server to return the cluster back to normal where the Primary Server is set as the Master.

```
    root@debian:~# bascul
```

With this, you now have Sonata Billing installed in your high-availability environment.

Let’s continue with **Sonata Stats**. This process is similar to Sonata Recordings but has some additional files to copy.

First, go to the **Floating IP Address** on your browser and install Sonata Stats from the add-ons module under **Admin > Add-Ons > Add-Ons**. Then, go to the Sonata Stats URL which is ***https://FloatingIPAddress/stats*** by default. There, follow the wizard to finish with the installation.

Next, **copy the files** that we are going to replicate from the Primary Server to the Secondary Server. Run the following commands on the Primary Server.

```
    root@debian:~# tar -zcvf stats-env.tgz /usr/share/queues-stats/backend/.env

    root@debian:~# tar -zcvf stats-wizard.tgz /var/lib/sonata/stats/wizard.conf

    root@debian:~# tar xvfz stats-env.tgz -C /vpbx_data/

    root@debian:~# tar xvfz stats-wizard.tgz -C /vpbx_data/

    root@debian:~# rm -rf /usr/share/queues-stats/backend/.env

    root@debian:~# ln -s /vpbx_data/usr/share/queues-stats/backend/.env  /usr/share/queues-stats/backend/.env

    root@debian:~# rm -rf /var/lib/sonata/stats/wizard.conf

    root@debian:~# ln -s /vpbx_data/var/lib/sonata/stats/wizard.conf  /var/lib/sonata/stats/wizard.conf

    root@debian:~# rm -rf stats-env.tgz

    root@debian:~# rm -rf stats-wizard.tgz
```

Now, execute the **bascul command** on the Primary Server to turn the Secondary Server into the Master.

```
    root@debian:~# bascul
```

Go back to the **Floating IP Address** on your browser. You will now be navigating the Secondary Server. Here, install Sonata Stats as you did in the Primary Server.

Then, create the symbolic link for the **.env** file on the Secondary Server.

```
    root@debian:~# rm -rf /usr/share/queues-stats/backend/.env

    root@debian:~# ln -s /vpbx_data/usr/share/queues-stats/backend/.env /usr/share/queues-stats/
backend/.env

    root@debian:~# rm -rf /var/lib/sonata/stats/wizard.conf

    root@debian:~# ln -s /vpbx_data/var/lib/sonata/stats/wizard.conf  /var/lib/sonata/stats/
wizard.conf

    root@debian:~# cd /usr/share/queues-stats/backend/

    root@debian:~# php artisan config:cache && php artisan route:cache && php artisan view:cache
```

Now, execute the **bascul command** on the Primary Server to return the cluster to normal where the Primary Server is the Master.

```
    root@debian:~# bascul
```

Finally, add the **Sonata Stats service** automation. Run the following commands on **both** servers.

```
    root@debian:~# systemctl stop sonata-stats

    root@debian:~# systemctl disable sonata-stats
```

Then, run the following commands in the Primary Server.

```
    root@debian:~# pcs resource create sonata-stats service:sonata-stats op monitor interval=30s

    root@debian:~# pcs cluster cib fs_cfg

    root@debian:~# pcs cluster cib-push fs_cfg --config

    root@debian:~# pcs -f fs_cfg constraint colocation add sonata-stats with ClusterIP INFINITY

    root@debian:~# pcs -f fs_cfg constraint order vpbx-monitor then sonata-stats

    root@debian:~# pcs cluster cib-push fs_cfg --config
```

With this, you now have Sonata Billing installed in your high-availability environment.

Let’s finish with the Sonata Suite with **Sonata Dialer**. The process is similar to the Sonata Stats.

First, go to the **Floating IP Address** on your browser, and install Sonata Dialer from the add-ons module under **Admin > Add-Ons > Add-Ons**. Once installed, go to the Sonata Dialer URL which is ***https://FloatingIPAddress/dialer*** by default. There, follow the wizard to finish the installation.

Next, **copy the files** that we are going to replicate from the Primary Server to the Secondary Server. Run the following commands on the Primary Server.

```
    root@debian:~# tar -zcvf dialer-env.tgz /usr/share/dialer/backend/.env

    root@debian:~# tar -zcvf dialer-wizard.tgz /var/lib/sonata/dialer/wizard.conf

    root@debian:~# tar xvfz dialer-env.tgz -C /vpbx_data/

    root@debian:~# tar xvfz dialer-wizard.tgz -C /vpbx_data/

    root@debian:~# rm -rf /usr/share/dialer/backend/.env

    root@debian:~# ln -s /vpbx_data/usr/share/dialer/backend/.env  /usr/share/dialer/backend/.env

    root@debian:~# rm -rf /var/lib/sonata/dialer/wizard.conf

    root@debian:~# ln -s /vpbx_data/var/lib/sonata/dialer/wizard.conf  /var/lib/sonata/dialer/
wizard.conf

    root@debian:~# rm -rf dialer-env.tgz
```

Now, execute the **bascul command** on the Primary Server to turn the Secondary Server into the Master.

```
    root@debian:~# bascul
```

Go back to the **Floating IP Address** on your browser. You will now be navigating the Secondary Server. Here, install Sonata Stats as you did in the Primary Server.

Then, create the symbolic links for the **.env** file on the Secondary Server.

```
    root@debian:~# rm -rf /usr/share/dialer/backend/.env

    root@debian:~# ln -s /vpbx_data/usr/share/dialer/backend/.env /usr/share/dialer/backend/.env

    root@debian:~# rm -rf /var/lib/sonata/dialer/wizard.conf

    root@debian:~# ln -s /vpbx_data/var/lib/sonata/dialer/wizard.conf  /var/lib/sonata/dialer/
wizard.conf

    root@debian:~# cd /usr/share/dialer/backend/

    root@debian:~# php artisan config:cache && php artisan route:cache && php artisan view:cache
```

Now, execute the **bascul command** on the Primary Server to return the cluster to normal where the Primary Server is the Master.

```
    root@debian:~# bascul
```

Finally, add the **Sonata Dialer service** automation. Run the following commands on **both** servers.

```
    root@debian:~# systemctl stop sonata-dialer

    root@debian:~# systemctl disable sonata-dialer
```

Then, run the following commands in the Primary Server.

```
    root@debian:~# pcs resource create sonata-dialer service:sonata-dialer op monitor
interval=30s

    root@debian:~# pcs cluster cib fs_cfg

    root@debian:~# pcs cluster cib-push fs_cfg --config

    root@debian:~# pcs -f fs_cfg constraint colocation add sonata-dialer with ClusterIP INFINITY

    root@debian:~# pcs -f fs_cfg constraint order vpbx-monitor then sonata-dialer

    root@debian:~# pcs cluster cib-push fs_cfg --config
```

With this, you now have Sonata Dialer installed in your high-availability environment. Now all of the Sonata Suite is up and running in your high-availability environment!

Let's move towards **VitXi**, our WebRTC solution.

Beforehand, make sure that your cluster is normalized with the Primary Server set as the Master. You can use the **role command** to verify this.

First, go to the **Floating IP Address** on your browser, and follow the installation procedure for VitXi from **VitXi's manual**. VitXi installation requires you to install the add-on from ***Admin > Add-Ons > Add-ons***, make various VitalPBX configurations, and follow the installation wizard from VitXi's URL which is ***https://FloatingIPAddress/webrtc*** by default. This process is detailed in VitXi's manual.

With VitXi completely installed and VitalPBX configured, **copy the files** that we are going to replicate from the Primary Server to the Secondary Server. Run the following commands on the Primary Server.

```
    root@debian:~# tar -zcvf vitxi-env.tgz /usr/share/vitxi/backend/.env

    root@debian:~# tar -zcvf vitxi-storage.tgz /usr/share/vitxi/backend/storage

    root@debian:~# tar -zcvf vitxi-wizard.tgz /var/lib/vitxi/wizard.conf

    root@debian:~# tar xvfz vitxi-env.tgz -C /vpbx_data/

    root@debian:~# tar xvfz vitxi-storage.tgz -C /vpbx_data/

    root@debian:~# tar xvfz vitxi-wizard.tgz -C /vpbx_data/


    root@debian:~# rm -rf /usr/share/vitxi/backend/.env

    root@debian:~# rm -rf /usr/share/vitxi/backend/storage

    root@debian:~# rm -rf /var/lib/vitxi/wizard.conf


    root@debian:~# ln -s /vpbx_data/usr/share/vitxi/backend/.env /usr/share/vitxi/backend/.env

    root@debian:~# ln -s /vpbx_data/usr/share/vitxi/backend/storage  /usr/share/vitxi/backend/
storage

    root@debian:~# ln -s /vpbx_data/var/lib/vitxi/wizard.conf  /var/lib/vitxi/wizard.conf
    root@debian:~# rm -rf vitxi-env.tgz

    root@debian:~# rm -rf vitxi-storage.tgz

    root@debian:~# rm -rf vitxi-wizard.tgz
```

Run the **bascul command** on the Primary Server to turn the Secondary Server into the Master.

```
    root@debian:~# bascul
```

Go back to the **Floating IP Address** on your browser. You will now be navigating the Secondary Server. Here, install Sonata Stats as you did in the Primary Server.

Then, create the symbolic links for the **.env** file on the Secondary Server.

```
    root@debian:~# rm -rf /usr/share/vitxi/backend/.env

    root@debian:~# ln -s /vpbx_data/usr/share/vitxi/backend/.env /usr/share/vitxi/backend/.env

    root@debian:~# rm -rf /var/lib/vitxi/wizard.conf

    root@debian:~# ln -s /vpbx_data/var/lib/vitxi/wizard.conf  /var/lib/vitxi/wizard.conf

    root@debian:~# rm -rf /usr/share/vitxi/backend/storage

    root@debian:~# ln -s /vpbx_data/usr/share/vitxi/backend/storage  /usr/share/vitxi/backend/
storage

    root@debian:~# cd /usr/share/vitxi/backend/

    root@debian:~# php artisan config:cache && php artisan route:cache && php artisan view:cache
```

Now, execute the **bascul command** on the Primary Server to return the cluster to normal where the Primary Server is the Master.

Finally, add the **Sonata Dialer service** automation. Run the following commands on **both** servers.

```
    root@debian:~# systemctl stop vitxi

    root@debian:~# systemctl disable vitxi
```

Then, run the following commands in the Primary Server.

```
    root@debian:~# pcs resource create vitxi service:vitxi op monitor interval=30s

    root@debian:~# pcs cluster cib fs_cfg

    root@debian:~# pcs cluster cib-push fs_cfg --config

    root@debian:~# pcs -f fs_cfg constraint colocation add vitxi with ClusterIP INFINITY

    root@debian:~# pcs -f fs_cfg constraint order vpbx-monitor then vitxi

    root@debian:~# pcs cluster cib-push fs_cfg --config
```

With this, you now have VitXi installed in your high-availability environment!

We can now move towards the **OpenVPN** add-on module.

First, go to the **Floating IP Address** on your browser, and install the OpenVPN add-on from the add-ons module under **Admin > Add-Ons > Add-Ons**.

Once installed, **copy the files** that we are going to replicate from the Primary Server to the Secondary Server. Run the following commands on the Primary Server.

```
root@debian:~# tar -zcvf etc-openvpn.tgz /etc/openvpn

root@debian:~# tar xvfz etc-openvpn.tgz -C /vpbx_data/

root@debian:~# rm -rf /etc/openvpn

root@debian:~# ln -s /vpbx_data/etc/openvpn /etc/openvpn

root@debian:~# rm -rf etc-openvpn.tgz
```

Run the **bascul command** on the Primary Server to turn the Secondary Server into the Master.

```
root@debian:~# bascul
```

Go back to the **Floating IP Address** on your browser. You will now be navigating the Secondary Server. Here, install the OpenVPN add-on as you did in the Primary Server.

Then, create the symbolic link for the **OpenVPN Path** on the Secondary Server.

```
root@debian:~# rm -rf /etc/openvpn

root@debian:~# ln -s /vpbx_data/etc/openvpn /etc/openvpn
```

Finally, add the **OpenVPN service** automation. Run the following commands on **both** servers.

```
root@debian:~# systemctl stop openvpn

root@debian:~# systemctl disable openvpn
```

Then, run the following commands in the Primary Server.

```
root@debian:~# pcs resource create openvpn service:openvpn op monitor interval=30s

root@debian:~# pcs cluster cib fs_cfg

root@debian:~# pcs cluster cib-push fs_cfg --config

root@debian:~# pcs -f fs_cfg constraint colocation add openvpn with ClusterIP INFINITY

root@debian:~# pcs -f fs_cfg constraint order vpbx-monitor then openvpn

root@debian:~# pcs cluster cib-push fs_cfg --config
```

With this, you now have OpenVPN installed in your high-availability environment!

Let’s move toward the **Maintenance** add-on. This is the final add-on that requires special steps for installation in a high-availability environment.

First, go to the **Floating IP Address** on your browser and install the Maintenance add-on module from the add-ons module under **Admin > Add-Ons > Add-Ons**.

Once installed, execute the **bascul command** on the Primary Server to turn the Secondary server into the Master.

```
root@debian:~# bascul
```

Go back to the **Floating IP Address** on your browser. You will now be navigating the Secondary Server. Install the Maintenance add-on as you did on the Primary Server.

Afterward, run the **bascul command** on the Primary Server once again to return the cluster to normal, where the Primary Server is set as the Master.

```
root@debian:~# bascul
```

Because the **Maintenance cron file** is copied to a shared directory at **/etc/cron.d/vpbx_maintenace** that has other system cron services active, we recommend first creating the Maintenance on the Primary Server first, then using the bascul command to switch the server roles, and finally create the same maintenance on the Secondary Server.

Since the databases are synchronized, it is only necessary to go to **Admin > Tools > Maintenance** and click on **Save** and **Apply Changes**.

With this, you now have the Maintenance add-on installed in a high-availability environment.

These are all the add-ons that require special attention and steps for installation in a High Availability environment. The rest of the add-ons can be installed similarly to how the Maintenance add-on was installed. The general steps for the rest of the add-ons are as follows.

1. Verify that the server cluster is **normalized** where the Primary Server is set as the Master. This can be verified with the **role command**.
2. Go to the **Floating IP Address** on your browser.
3. Install the add-on on the Primary Server from the add-ons module under **Admin > Add-Ons > Add-Ons**.
4. Use the **bascul command** to switch the servers’ roles.
5. Install the add-on on the Secondary Server from the add-ons module under **Admin > Add-Ons > Add-Ons**.
6. Use the **bascul command** again to normalize the server cluster and have the Primary Server as the Master.

With this done, you now have a full high-availability environment with add-ons in VitalPBX 4!

Section 13, Lesson 3 - DRBD High-Availability Monitoring and Updates

Now, let’s take a look into additional configurations and processes you can do to troubleshoot your high-availability environment using DRBD.

Let’s look into a **custom context** that will allow us to **monitor the current state of the high-availability environment** from any registered extension.

First, we will create a new context using **nano**. Run the following command from the Primary Server.

```
root@debian:~# nano /etc/asterisk/vitalpbx/extensions__61-ha-test.conf
```

Add the following content.

```
[cos-all](+)

exten => *777,1,NoOp(Say Asterisk IP Address)

 same => n,Answer()

 same => n,Set(ASTERISK_IP=${SHELL(/usr/bin/hostname -I | awk -F " " '{print $1}')})

 same => n,NoOp(IP Address: ${ASTERISK_IP})

 same => n,SayDigits(${CUT(ASTERISK_IP,.,1)})

 same => n,Playback(letters/dot)

 same => n,SayDigits(${CUT(ASTERISK_IP,.,2)})

 same => n,Playback(letters/dot)

 same => n,SayDigits(${CUT(ASTERISK_IP,.,3)})

 same => n,Playback(letters/dot)

 same => n,SayDigits(${CUT(ASTERISK_IP,.,4)})

 same => n,PlayBack(silence/1&vm-goodbye)

 same => n,Hangup()
```

Instead of ***777**, you can use any other code of your liking that doesn’t interfere with your numbering plan.

For us to be able to use this context, we must **restart the Asterisk Dial Plan**. Run the following command from the Primary Server.

```
root@debian:~# asterisk -rx"dialplan reload”
```

To test the custom context, dial ***777** or your code of choosing from any registered extension.

> ***Note:** as a **Warning**, **if you must turn off the servers** for any reason, when turning them back on **always try to start with the Primary Server**. Wait for the Primary Server to fully start, and **then turn on the Secondary Server**.*

Let's look into **updating VitalPBX or any Add-On**. The steps for updating VitalPBX or any add-on are as follows.

1. Go to the **Floating IP Address** on your browser, and log in as the administrator.
2. Update VitalPBX from the Web UI. Under the user menu in the upper right-hand corner, click on **Check for Updates**.
3. Run the **bascul command** on the Primary Server. This will turn the Secondary Server into the Master.
4. Go back to the **Floating IP Address** on your browser, and log in as the administrator.
5. Update VitalPBX from the Web UI.
6. Run the **bascul command** again to normalize the server cluster where the Primary Server is the Master.

With this, you will have updated both servers. Basically, you update the Primary Server first, then switch to the Secondary Server and update it next. Then, you normalize the cluster.

Section 13, Lesson 4 - Changing One of the Servers in DRBD High Availability

There could be occasions where you need to change one of the servers. This can be due to one of them getting damaged or hardware modification.

Let's look into **changing the Secondary Server**.

First, we will proceed to destroy the cluster on the Primary Server.

```
root@debian:~# pcs cluster stop

root@debian:~# pcs cluster destroy

root@debian:~# systemctl disable pcsd.service

root@debian:~# systemctl disable corosync.service

root@debian:~# systemctl disable pacemaker.service

root@debian:~# systemctl stop pcsd.service

root@debian:~# systemctl stop corosync.service

root@debian:~# systemctl stop pacemaker.service
```

Next, if we still have access to the Secondary Server, we destroy the cluster as well on the Secondary Server.

```
root@debian:~# pcs cluster stop --force

root@debian:~# pcs cluster destroy

root@debian:~# systemctl disable pcsd.service

root@debian:~# systemctl disable corosync.service

root@debian:~# systemctl disable pacemaker.service

root@debian:~# systemctl stop pcsd.service

root@debian:~# systemctl stop corosync.service

root@debian:~# systemctl stop pacemaker.service
```

Since when destroying the cluster the DRBD volume is unmounted, in the Primary Server we must **mount it again manually** to avoid interrupting the normal operation of our services. Run the following commands on the Primary Server.

```
root@debian:~# drbdadm up drbd0

root@debian:~# drbdadm primary drbd0 --force

root@debian:~# mount /dev/drbd0 /vpbx_data
```

Now, enable the services on the Primary Server to make sure our server continues to work as normal.

```
root@debian:~# systemctl enable asterisk

root@debian:~# systemctl restart asterisk

root@debian:~# systemctl enable mariadb

root@debian:~# systemctl restart mariadb

root@debian:~# systemctl enable fail2ban

root@debian:~# systemctl restart fail2ban

root@debian:~# systemctl enable vpbx-monitor

root@debian:~# systemctl restart vpbx-monitor
```

At this moment, the Primary Server is now working as normal. We must now configure the replica with our new server.

First, prepare the new server following the steps for the installation in Section 13, Lesson 1, with the disk partition and the free space. Then enable remote access with the root user. Change the IP address to a static IP address. Install the dependencies. Change the hostname in the Web UI and CLI using the same hostname as the old Secondary Server. Create the sda3 partition using the fdisk command. And finally, configure the firewall with the ports established in the table from lesson 1. These are the steps for the manual HA setup presented in lesson 1. This should only be configured in the new Secondary Server.

Now, we will proceed to format the new partition on the new Secondary Server using the following commands.

```
root@debian:~# mkdir /vpbx_data

root@debian:~# mke2fs -j /dev/sda3

root@debian:~# dd if=/dev/zero bs=1M count=500 of=/dev/sda3; sync
```

Load the DRBD module and enable the service on the new Secondary Server using the following commands.

```
root@debian:~# modprobe drbd

root@debian:~# systemctl enable drbd.service
```

Next, we must create new configuration files called **drbd0.res** in the **/etc/drbd.d** directory on the new Secondary Server for the new resource named drbd0.

```
root@debian:~# nano /etc/drbd.d/drbd0.res
```

Add the following content.

```
resource drbd0 {

startup {

        wfc-timeout  5;

      outdated-wfc-timeout 3;

        degr-wfc-timeout 3;

      outdated-wfc-timeout 2;

}

syncer {

      rate 10M;

      verify-alg md5;

}

net {

    after-sb-0pri discard-older-primary;

    after-sb-1pri discard-secondary;

    after-sb-2pri call-pri-lost-after-sb;

}

handlers {

    pri-lost-after-sb "/sbin/reboot";

}
on vitalpbx-primary.local {
      device /dev/drbd0;
      disk /dev/sda3;
      address 192.168.10.31:7789;
      meta-disk internal;
}
on vitalpbx-secondary.local {
      device /dev/drbd0;
      disk /dev/sda3;
      address 192.168.10.32:7789;
      meta-disk internal;
      }
}
```

> ***Note:*** *Although the access interfaces can be used, which in this case is ETH0. It is recommended to use an interface (ETH1) for synchronization, this interface must be directly connected between both servers.*

Initialize the metadata storage on each server by executing the following command on the new Secondary Server.

```
root@debian:~# drbdadm create-md drbd0
Writing meta data...
New drbd meta data block successfully created.
```

Run the following command on the new Secondary Server to start the drbd0.

```
root@debian:~# drbdadm up drbd0
```

You can check the current status of the synchronization using the **cat /proc/drbd** command.

Once the synchronization is done, you can follow the steps from **Section 13, Lesson 1** for **configuring the cluster** starting with the **hacluster user and password** up to the creation of the **fail2ban service**. Then, follow the steps for creating the **bascul command** and the creation of the **role command**. All of this is for the new Secondary Server.

After this, you will now have replaced the old Secondary Server with a new one!

Now, let’s look into **changing the Primary Server**. This method differs from changing the Secondary Server.

First, we proceed to destroy the cluster in the Primary Server if we still can. Run the following commands in the Primary Server.

```
root@debian:~# pcs cluster stop

root@debian:~# pcs cluster destroy

root@debian:~# systemctl disable pcsd.service

root@debian:~# systemctl disable corosync.service

root@debian:~# systemctl disable pacemaker.service

root@debian:~# systemctl stop pcsd.service

root@debian:~# systemctl stop corosync.service

root@debian:~# systemctl stop pacemaker.service
```

Next, we do the same in the Secondary Server with the following commands.

```
root@debian:~# pcs cluster stop --force

root@debian:~# pcs cluster destroy

root@debian:~# systemctl disable pcsd.service

root@debian:~# systemctl disable corosync.service

root@debian:~# systemctl disable pacemaker.service

root@debian:~# systemctl stop pcsd.service

root@debian:~# systemctl stop corosync.service

root@debian:~# systemctl stop pacemaker.service
```

Since by destroying the cluster the DRBD unit is unmounted, in the Secondary Server, we must mount it again manually to avoid interrupting the normal operation of our services. Run the following commands in the Secondary Server.

```
root@debian:~# drbdadm up drbd0

root@debian:~# drbdadm primary drbd0 --force

root@debian:~# mount /dev/drbd0 /vpbx_data
```

Now, enable the services on the Secondary Server to make sure that our server continues to work as normal.

```
root@debian:~# systemctl enable asterisk

root@debian:~# systemctl restart asterisk

root@debian:~# systemctl enable mariadb

root@debian:~# systemctl restart mariadb

root@debian:~# systemctl enable fail2ban
```

```
root@debian:~# systemctl restart fail2ban

root@debian:~# systemctl enable vpbx-monitor

root@debian:~# systemctl restart vpbx-monitor
```

With this, our Secondary Server is now working as normal. We now proceed to configure the replica with our new server.

First, prepare the new server following the steps for the installation in Section 13, Lesson 1, with the disk partition and the free space. Then enable remote access with the root user. Change the IP address to a static IP address. Install the dependencies. Change the hostname in the Web UI and CLI using the same hostname as the old Primary Server. Create the sda3 partition using the fdisk command. And finally, configure the firewall with the ports established in the table from lesson 1. These are the steps for the manual HA setup presented in lesson 1. This should only be configured in the new Primary Server.

Now, proceed to format the new partition on the new Primary Server using these commands.

```
root@debian:~# mkdir /vpbx_data

root@debian:~# mke2fs -j /dev/sda3

root@debian:~# dd if=/dev/zero bs=1M count=500 of=/dev/sda3; sync
```

Then, load the module on the new Primary Server and enable the service with the following commands.

```
root@debian:~# modprobe drbd

root@debian:~# systemctl enable drbd.service
```

Next, back up the original global_common.conf file, and create a new one using nano on the Primary Server. Use the following commands on the new Primary Server.

```
root@debian:~# mv /etc/drbd.d/global_common.conf /etc/drbd.d/global_common.conf.orig

root@debian:~# nano /etc/drbd.d/global_common.conf
```

Add the following content.

```
global {

  usage-count no;

}

  common {

net {

  protocol C;

  }

}
```

Next, we must create new configuration files called **drbd0.res** in the **/etc/drbd.d** directory on the new Primary Server for the new resource named drbd0.

```
root@debian:~# nano /etc/drbd.d/drbd0.res
```

Add the following content.

```
resource drbd0 {
startup {
        wfc-timeout  5;
      outdated-wfc-timeout 3;
        degr-wfc-timeout 3;
      outdated-wfc-timeout 2;
}
syncer {
      rate 10M;
      verify-alg md5;
}
net {
    after-sb-0pri discard-older-primary;
    after-sb-1pri discard-secondary;
    after-sb-2pri call-pri-lost-after-sb;
}
handlers {
    pri-lost-after-sb "/sbin/reboot";
}
on vitalpbx-primary.local {
      device /dev/drbd0;
      disk /dev/sda3;
      address 192.168.10.31:7789;
      meta-disk internal;
}
on vitalpbx-secondary.local {

      device /dev/drbd0;

      disk /dev/sda3;

      address 192.168.10.32:7789;

      meta-disk internal;

      }

}
```

> ***Note:*** *Although the access interfaces can be used, which in this case is ETH0. It is recommended to use an interface (ETH1) for synchronization, this interface must be directly connected between both servers.*

Initialize the metadata storage on each server by executing the following command on the new Primary Server.

```
root@debian:~# drbdadm create-md drbd0

Writing meta data...

New drbd meta data block successfully created.
```

On the new Primary Server, run the following command to start the drbd0.

```
root@debian:~# drbdadm up drbd0
```

You can check the current status of the synchronization while it's being performed. The **cat /proc/drbd command** displays the creation and synchronization progress of the resource.

Once the synchronization is done, you can follow the steps from **Section 13, Lesson 1** for **configuring the cluster** starting with the **hacluster user and password** up to the creation of the **fail2ban service**. Then, follow the steps for creating the **bascul command** and the creation of the **role command**. All of this is for the new Primary Server.

After this, you will now have replaced the old Primary Server with a new one!

With this, you now have the necessary tools to run VitalPBX in a high-availability environment. Allowing you to rest assured that you have two servers working together for minimal to no downtime.

Section 13, Lesson 5 - DRBD High Availability Useful Commands

```
root@debian:~# bascul
```

This command is used to change roles between high-availability servers. If all is well, a confirmation question should appear if we wish to execute the action. The bascul command permanently moves services from one server to another. If you want to return the services to the main server you must execute the command again.

```
root@debian:~# role
```

This command shows the status of the current server.

```
root@debian:~# pcs resource refresh --full
```

This command is used to poll all resources even if the status is unknown.

```
root@debian:~# pcs node standby host
```

This command stops the server node where it is running. The node status is stopped.

```
root@debian:~# pcs node unstandby host
```

In some cases, the bascul command does not finish tilting, which causes one of the servers to be on standby (stopped), with this command the state is restored to normal.

```
root@debian:~# pcs resource delete
```

This command removes the resource so it can be created.

```
root@debian:~# pcs resource create
```

This command creates the resource.

```
root@debian:~# corosync-cfgtool -s
```

This command is used to check whether cluster communication is happy.

```
root@debian:~# ps axf
```

This command confirms that Corosync is functional, we can check the rest of the stack. The pacemaker has already been started, so verify the necessary processes are running.

```
root@debian:~# pcs status
```

This command allows you to check the pcs status output.

```
root@debian:~# crm_verify –L –V
```

This command allows you to check the validity of the configuration.

```
root@debian:~# drbdadm status
```

This command shows the integrity status of the disks that are being shared between both servers in high availability. If for some reason the status of Connecting or Standalone returns to us, wait a while and if the state remains it is because there are synchronization problems between both servers, and you should execute the drbdsplit command.

```
root@debian:~# cat /proc/drbd
```

This command shows you the state of your device is kept in /proc/drbd.

```
root@debian:~# drbdadm connect drbd0
```

This command shows you the status on the other node (the split brain survivor), if its connection state is also StandAlone, you will need to enter.

```
root@debian:~# drbdadm role drbd0
```

This command is another way to check the role of the block device.

```
root@debian:~# drbdadm primary drbd0
```

This command allows you to switch the DRBD block device to Primary using drbdadm.

```
root@debian:~# drbdadm secondary drbd0
```

This command allows you to switch the DRBD block device to Secondary using drbdadm.

```
root@debian:~# /usr/share/vitalpbx/ha/ ./vpbxha.sh
```

This script creates the cluster automatically.

```
root@debian:~# /usr/share/vitalpbx/ha/ ./destroy.sh
```

This script will completely destroy the cluster, leaving the DRBD intact.

```
root@debian:~# /usr/share/vitalpbx/ha/ ./rebuild.sh
```

This script recreates the cluster starting from the fact that the DRBD is already configured on both servers.

Section 13, Lesson 6 - High Availability using Proxmox™

Another way to have a high-availability environment we have studied is using **Proxmox™ VE**. **Proxmox VE** is a complete **open-source server management platform** for enterprise virtualization. It tightly integrates the **KVM hypervisor** and **Linux Containers (LXC)**, software-defined **storage**, and **networking** functionality, into a **single platform**. With the built-in web-based user interface, you can manage **virtual machines and containers**, **high availability for clusters**, or the **built-in disaster recovery tools** with ease. You can explore Proxmox as an option to create your VitalPBX instances in your local data center infrastructure.

To start using Proxmox VE for an HA environment, there are some prerequisites that need to be fulfilled.

- **System Requirements**

  For production servers, high-quality server equipment is needed. Proxmox VE supports clustering, which means that multiple installations of Proxmox VE can be centrally managed thanks to the built-in cluster functionality. Proxmox VE can use local storage such as (DAS), SAN, and NAS, as well as shared and distributed storage (Ceph).

- **Recommended Hardware**
  - **CPU -** Intel EMT64 or AMD64 with Intel VT/AMD-V CPUs running at 3.5 GHz.
  - **Memory -** At least 4 GB for OS services and Proxmox VE. More memory needs to be designated for guests. For Ceph or ZFS additional memory is required, approximately 1GB of memory for every TB of storage used.
  - **Fast and redundant storage -** Better results with SSD drives.
    - **OS storage -** Hardware RAID with battery-protected write cache ("BBU") or non-RAID with ZFS cache and SSD.
    - **VM storage -** For local storage, use hardware RAID with battery-backed write cache (BBU) or no RAID for ZFS. Neither ZFS nor Ceph supports a hardware RAID controller. Shared and distributed storage is also possible.
  - **Redundant Gigabit NICs -** Additional NICs depending on preferred storage technology and cluster configuration: 10 Gbit and higher are also supported.
  - **For PCI(e) pass-through -** a CPU with CPU flag VT-d/AMD-d is required.

- **For Testing (Minimal Hardware for Testing Only)**
  - **CPU -** 64-bit (Intel EMT64 or AMD64)
  - **Motherboard -** Intel VT/AMD-V compatible CPU/motherboard (for full KVM virtualization support)
  - **Memory -** Minimum 2 GB of RAM
  - **Storage -** 500GB HDD
  - **Network -** One Gigabit NIC

Additionally, we are going to need an NFS Storage system. For this, in this example, we are going to be using another server to work as a NAS server.

With these prerequisites fulfilled, we can proceed to **install Proxmox VE.**

Proxmox needs a clean hard drive because it will remove all partitions and data from the hard drive during installation.

First, download Proxmox VE's ISO image from Proxmox's official website, https://proxmox.com/en/downloads/category/iso-images-pve.

If you are going to install it on physical hardware, you must flash the ISO image on a USB drive of at least 8GB. We recommend the Balena Etcher software, https://www.balena.io/etcher/.

Afterward, we can install Proxmox VE on dedicated hardware using the USB drive. Once we boot into the image we will be greeted with the following screen.

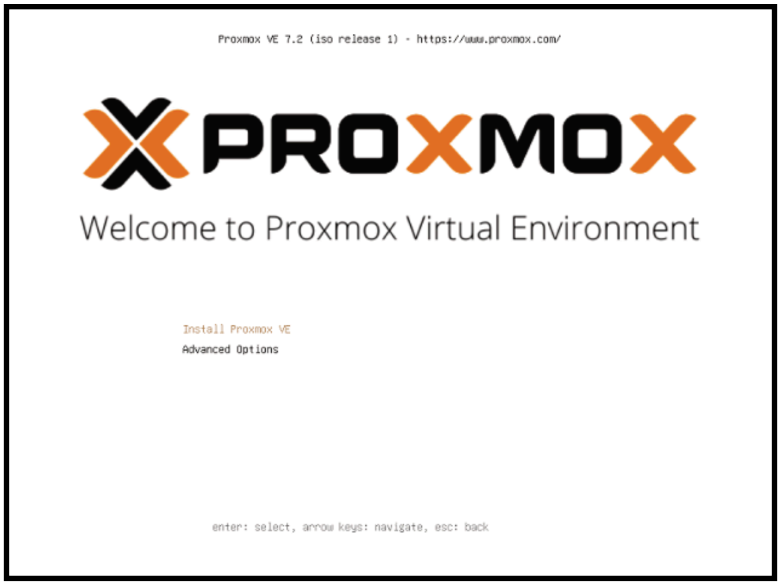
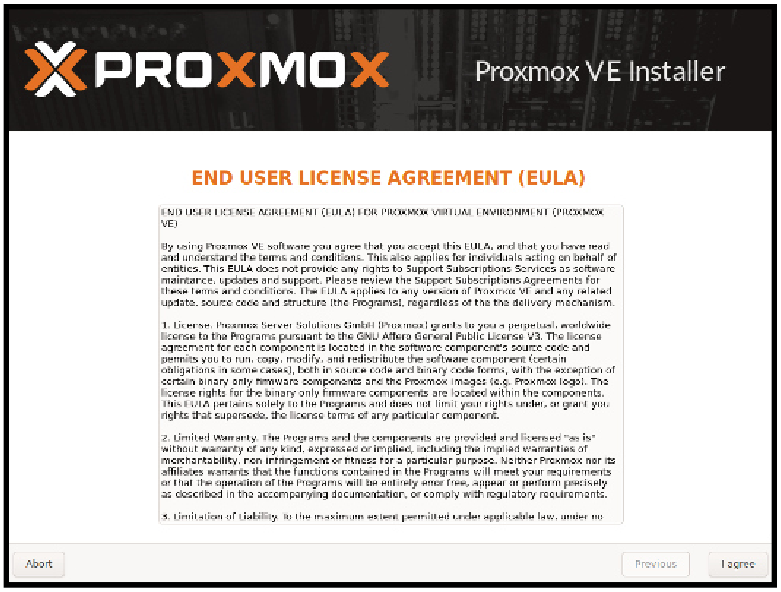

Proxmox VE 7.2 (iso release 1) - https://www.proxmox.com/
PROXMOX
Welcome to Proxmox Virtual Environment
Install Proxmox VE
Advanced Options
enter: select, arrow keys: navigate, esc: back

Image 13.6.1 - Proxmox installation welcome screen.

Select the **Install Proxmox VE** option, and press **Enter**. After a few seconds, the EULA will appear. Click on the **I agree** button.



END USER LICENSE AGREEMENT (EULA)

END USER LICENSE AGREEMENT (EULA) FOR PROXMOX VIRTUAL ENVIRONMENT (PROXMOX VE)

By using Proxmox VE software you agree that you accept this EULA, and that you have read and understand the terms and conditions. This also applies for individuals acting on behalf of entities. This EULA does not provide any rights to Support Subscriptions Services as software maintance, updates and support. Please review the Support Subscriptions Agreements for these terms and conditions. The EULA applies to any version of Proxmox VE and any related update, source code and structure (the Programs), regardless of the the delivery mechanism.

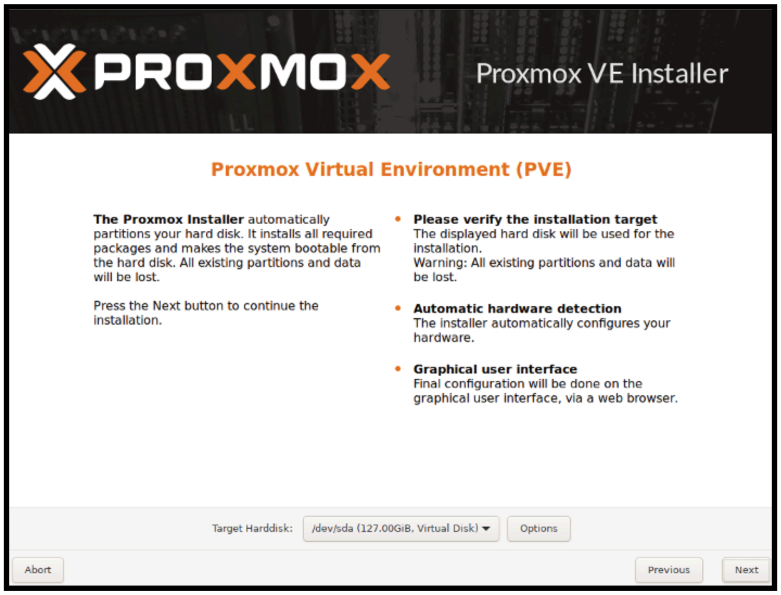
1. License. Proxmox Server Solutions GmbH (Proxmox) grants to you a perpetual, worldwide license to the Programs pursuant to the GNU Affero General Public License V3. The license agreement for each component is located in the software component's source code and permits you to run, copy, modify, and redistribute the software component (certain obligations in some cases), both in source code and binary code forms, with the exception of certain binary only firmware components and the Proxmox images (e.g. Proxmox logo). The license rights for the binary only firmware components are located within the components. This EULA pertains solely to the Programs and does not limit your rights under, or grant you rights that supersede, the license terms of any particular component.

2. Limited Warranty. The Programs and the components are provided and licensed "as is" without warranty of any kind, expressed or implied, including the implied warranties of merchantability, non infringement or fitness for a particular purpose. Neither Proxmox nor its affiliates warrants that the functions contained in the Programs will meet your requirements or that the operation of the Programs will be entirely error free, appear or perform precisely as described in the accompanying documentation, or comply with regulatory requirements.

3. Limitation of Liability. To the maximum extent permitted under applicable law, under no

Image 13.6.2 - Proxmox EULA screen.

Afterward, you must select the hard disk where you want to install Proxmox. Once selected, click on **Next.**



Proxmox Virtual Environment (PVE)

The Proxmox Installer automatically partitions your hard disk. It installs all required packages and makes the system bootable from the hard disk. All existing partitions and data will be lost.

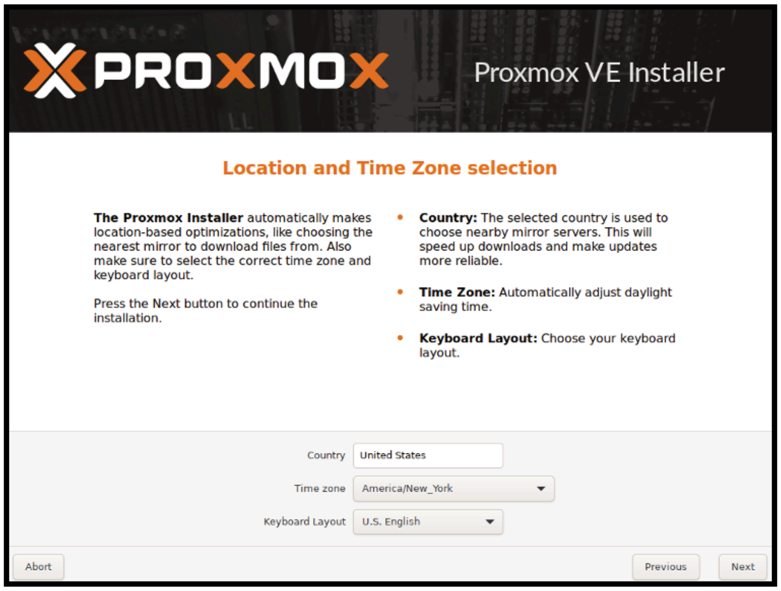
Press the Next button to continue the installation.

- **Please verify the installation target**
  The displayed hard disk will be used for the installation.
  Warning: All existing partitions and data will be lost.
- **Automatic hardware detection**
  The installer automatically configures your hardware.
- **Graphical user interface**
  Final configuration will be done on the graphical user interface, via a web browser.

Target Harddisk: /dev/sda (127.00GiB, Virtual Disk)

Image 13.6.3 - Proxmox hard disk selection screen.

On the next screen, enter your location, time zone, and language. Then click on **Next**.



Location and Time Zone selection

The Proxmox Installer automatically makes location-based optimizations, like choosing the nearest mirror to download files from. Also make sure to select the correct time zone and keyboard layout.

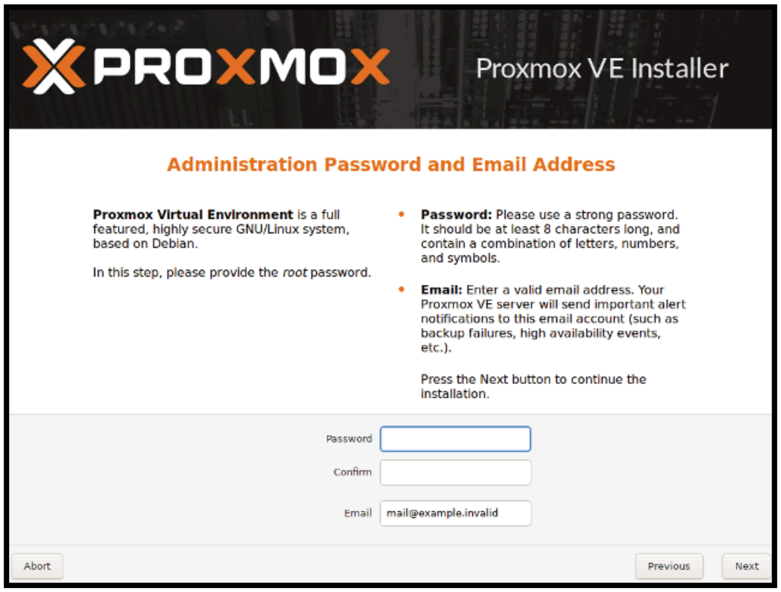
Press the Next button to continue the installation.

- **Country:** The selected country is used to choose nearby mirror servers. This will speed up downloads and make updates more reliable.
- **Time Zone:** Automatically adjust daylight saving time.
- **Keyboard Layout:** Choose your keyboard layout.

Country: United States
Time zone: America/New_York
Keyboard Layout: U.S. English

Image 13.6.4 - Proxmox location and time zone selection screen.

Now you must enter the password for the root user which is used both in the Proxmox web interface and in the server console via SSH. Since this is a very sensitive user, we recommend a fairly complex password.



Administration Password and Email Address

Proxmox Virtual Environment is a full featured, highly secure GNU/Linux system, based on Debian.

In this step, please provide the *root* password.

- **Password:** Please use a strong password. It should be at least 8 characters long, and contain a combination of letters, numbers, and symbols.
- **Email:** Enter a valid email address. Your Proxmox VE server will send important alert notifications to this email account (such as backup failures, high availability events, etc.).

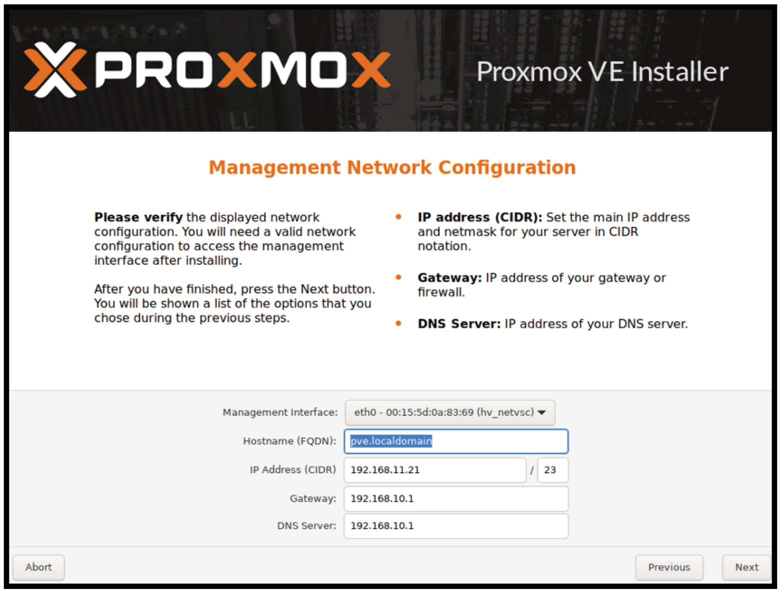
Press the Next button to continue the installation.

Password
Confirm
Email mail@example.invalid

Image 13.6.5 - Proxmox root user password screen.

For the Email address, you can use any email address of your liking. Click on **Next**.

Afterward, we must enter our Networking configurations.



Management Network Configuration

Please verify the displayed network configuration. You will need a valid network configuration to access the management interface after installing.

After you have finished, press the Next button. You will be shown a list of the options that you chose during the previous steps.

- **IP address (CIDR):** Set the main IP address and netmask for your server in CIDR notation.
- **Gateway:** IP address of your gateway or firewall.
- **DNS Server:** IP address of your DNS server.

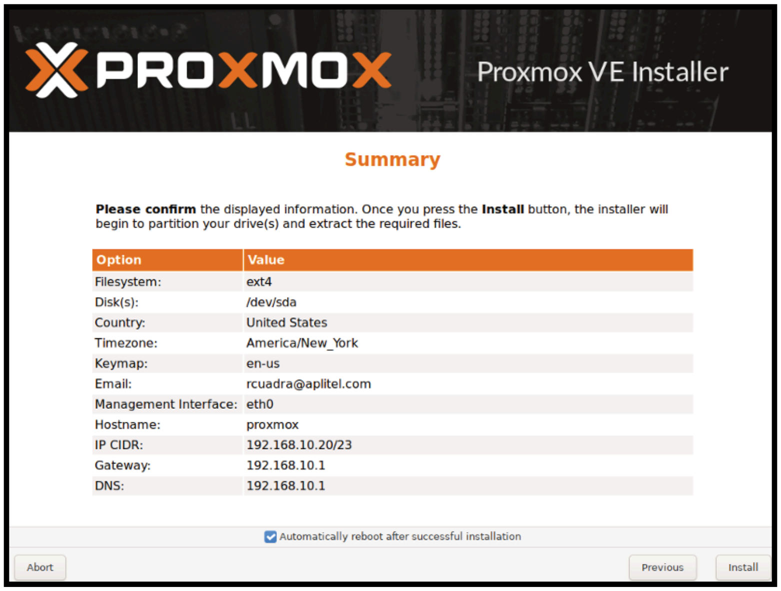
Management Interface: eth0 - 00:15:5d:0a:83:69 (hv_netvsc)
Hostname (FQDN): pve.localdomain
IP Address (CIDR) 192.168.11.21 / 23
Gateway: 192.168.10.1
DNS Server: 192.168.10.1

Image 13.6.6 - Proxmox network configuration screen.

- **Management Interface -** Select the interface through which we will manage Proxmox.
- **Hostname (FQDN) -** A valid FQDN is recommended. For local testing, we will use a .local domain.
- **IP Address (CIDR) -** Enter the IP address for the Proxmox server, using the CIDR format for the netmask.
- **Gateway -** Enter the default gateway for the Proxmox server.
- **DNS Server -** Enter the IP address for a DNS server to solve for server names. **i.e.** 8.8.8.8

Once the network has been configured, click on **Next**.

Finally, you will be presented with a summary of your installation configurations. Verify the information and click on **Install**.



Summary

Please confirm the displayed information. Once you press the **Install** button, the installer will begin to partition your drive(s) and extract the required files.

Option	Value
Filesystem:	ext4
Disk(s):	/dev/sda
Country:	United States
Timezone:	America/New_York
Keymap:	en-us
Email:	rcuadra@aplitel.com
Management Interface:	eth0
Hostname:	proxmox
IP CIDR:	192.168.10.20/23
Gateway:	192.168.10.1
DNS:	192.168.10.1

☑ Automatically reboot after successful installation

Image 13.6.7 - Proxmox installation summary screen.

The installation will proceed and may take a few minutes depending on your hardware.

Once the installation is done, enter the **IP Address** plus the port **8006** on your browser, **i.e. 192.168.20.220:8006**.

Another option to install Proxmox is by installing a minimal installation of Debian. For this, you can follow the Debian installation instructions in **Section 1, Lesson 1**. **During the installation, set the hostname for your Proxmox installation**. Later we will show you the scheme we used for our example.

Once you have Debian installed, make sure you have **SSH** and **WGET** installed.

```
root@debian:~# apt-install -y ssh wget
```

Then, make the root user available to log in via SSH, using nano.

```
root@debian:~# nano /etc/ssh/sshd_config
```

Go to the following line.

```
#PermitRootLogin prohibit-password
```

Change it to

```
PermitRootLogin yes
```

Save and **Exit** nano.

Next, we must set the static IP address for this server using nano.

```
root@debian:~# nano /etc/network/interfaces
```

Go to the following line.

```
iface eth0 inet dhcp

Change it to the following content.

iface eth0 inet static

address 192.168.20.220

netmask 255.255.255.0

gateway 192.168.20.1

dns-nameservers 8.8.8.8 8.8.4.4 1.1.1.1 1.1.0.0
```

Save and **Exit** nano.

> ***Note:*** *Your interface name and IP Address will vary from these instructions. This information is based on the scheme we present later. Make sure that you are using the interface name and appropriate IP Address for your environment.*

Now, we must make changes to the hosts file so the hostname resolves the server IP Address. Use nano to modify the hosts file.

```
    root@debian:~# nano /etc/hosts
```

Go to the following line.

```
    127.0.1.1 prox1.vitalpbx.local prox1
```

Change it to the following.

```
    192.168.20.220 prox1.vitalpbx.local prox1
```

Save and **Exit** nano. You can check your configurations with the following command.

```
    root@debian:~# hostname --ip-address
```

With this setup, we must add the Proxmox VE repositories. Run the following command.

```
    root@debian:~# echo "deb [arch=amd64] http://download.proxmox.com/debian/pve bullseye pve-no-
subscription" > /etc/apt/sources.list.d/pve-install-repo.list
```

Next, add the Proxmox repository key as root.

For Debian 11.

```
    root@debian:~# wget https://enterprise.proxmox.com/debian/proxmox-release-bullseye.gpg -O /
etc/apt/trusted.gpg.d/proxmox-release-bullseye.gpg

    # verify

    sha512sum /etc/apt/trusted.gpg.d/proxmox-release-bullseye.gpg

    7fb03ec8a1675723d2853b84aa4fdb49a46a3bb72b9951361488bfd19b29aab0a789a4f8c7406e71a69aabbc727c9
36d3549731c4659ffa1a08f44db8fdcebfa  /etc/apt/trusted.gpg.d/proxmox-release-bullseye.gpg
```

For Debian 12.

```
    root@debian:~# wget https://enterprise.proxmox.com/debian/proxmox-release-bookworm.gpg -O /
etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg

    # verify

    sha512sum /etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg

    7da6fe34168adc6e479327ba517796d4702fa2f8b4f0a9833f5ea6e6b48f6507a6da403a274fe201595edc86a8446
3d50383d07f64bdde2e3658108db7d6dc87 /etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg
```

Afterward, update the repository and system by running the following command.

```
    root@debian:~# apt update && apt full-upgrade
```

Then, install the Proxmox kernel.

For Debian 11.

```
root@debian:~# apt install pve-kernel-5.15
```

For Debian 12.

```
root@debian:~# apt install pve-kernel-6.2
```

Reboot the system.

```
root@debian:~# systemctl reboot
```

Once the system has rebooted, install the Proxmox packages using the following command.
For Debian 11.

```
root@debian:~# apt install proxmox-ve postfix open-iscsi
```

For Debian 12.

```
root@debian:~# apt install proxmox-ve postfix open-iscsi chrony
```

Then, remove the original Linux Kernel.

For Debian 11.

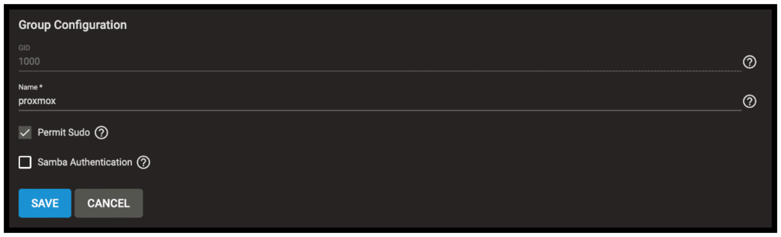
```
root@debian:~# apt remove linux-image-amd64 ‘linux-image-5.10*'
```

For Debian 12.

```
root@debian:~# apt remove linux-image-amd64 ‘linux-image-6.1*'
```

Afterward, update the GRUB.

```
root@debian:~# update-grub
```



Group Configuration

GID
1000

Name *
proxmox

☑ Permit Sudo

☐ Samba Authentication

Image 13.6.12 - TrueNAS group configurations.

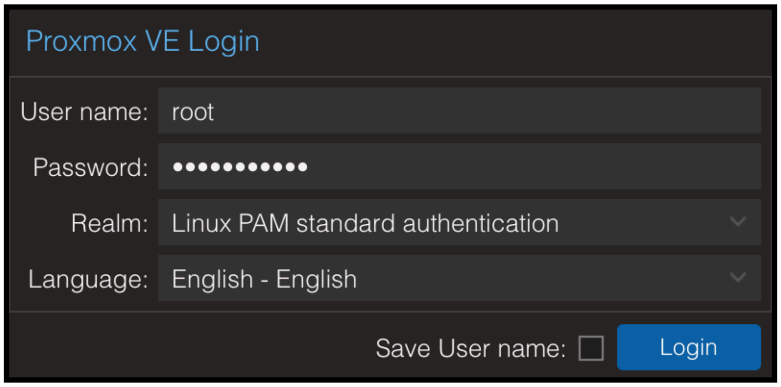
The **os-prober** package will scan all the partitions to create dual-boot GRUB entries. This can include the ones assigned to virtual machines, which we don’t want to add a boot entry. If you didn’t install Proxmox as a dual-boot with another OS, it is recommended to remove the os-prober package using the following command.

```
root@debian:~# apt remove os-prober
```

Finally, reboot the system using the following command.

```
root@debian:~# reboot
```

Once the system reboots, go to the **Proxmox IP Address using port 8006** on your browser, **i.e. 192.168.20.220:8006**. Here, login using the root user and password.



Proxmox VE Login

User name: root

Password: ●●●●●●●●●●●●

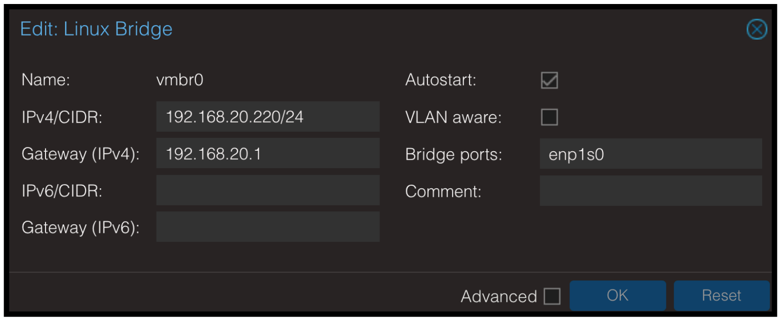
Realm: Linux PAM standard authentication

Language: English - English

Save User name: □ Login

Image 13.6.8 - Proxmox login prompt.

Go to **Datacenter > Your Proxmox node > System > Network**, and click **Create**. Here, create a **Linux Bridge** and call it **vmbr0**. Add the **IP Address** of your server **with the netmask in CIDR format**, the **gateway**, and enter the **network interface name** under the **Bridge port**. Then click **OK**.



Edit: Linux Bridge

Name: vmbr0

IPv4/CIDR: 192.168.20.220/24

Gateway (IPv4): 192.168.20.1

IPv6/CIDR:

Gateway (IPv6):

Autostart: ☑

VLAN aware: □

Bridge ports: enp1s0

Comment:

Advanced □ OK Reset

Image 13.6.9 - Linux Bridge creation in Proxmox.

> ***Note:*** *During this step, you may have to remove the IP address from the Network interface listed here as well. For this, you must* ***click the interface****, and then click on* ***Edit****. There, you can* ***remove the IP Address and Gateway****. Then, click* ***OK****.*

After the installation, you will see a message saying “You do not have a valid subscription for this server.” To remove this message, you will need to purchase a valid subscription. For this, you can go to the following link, https://www.proxmox.com/en/proxmox-ve/pricing.

For High Availability, you must complete this process three times on three separate servers. Ideally with similar hardware specifications. This is due to us needing what is called a **quorum**. The cluster of Proxmox servers will vote on which server the virtual machines will be transferred to.

For the latest instructions on how to install Proxmox in Debian, you can check the Proxmox wiki at https://pve.proxmox.com/wiki/Category:Installation.

For this example, we will be creating and using the following.

- **3 Physical Servers** with similar hardware specifications.
  - 16GB of Memory
  - 256GB of Storage
  - 4-Core CPU
- A **NAS Server** with enough capacity to store all the **Virtual Machines** we are going to create.

We will be using the following scheme.

- **Server 1**
  - **Name:** prox1.vitalpbx.local
  - **IP Address:** 192.168.20.220

- **Server 2**
  - **Name:** prox2.vitalpbx.local
  - **IP Address:** 192.168.20.221

- **Server 3**
  - **Name:** prox3.vitalpbx.local
  - **IP Address:** 192.168.20.222

- **NFS Server**
  - **Name:** truenas.vitalpbx.local
  - **IP Address:** 192.168.20.219

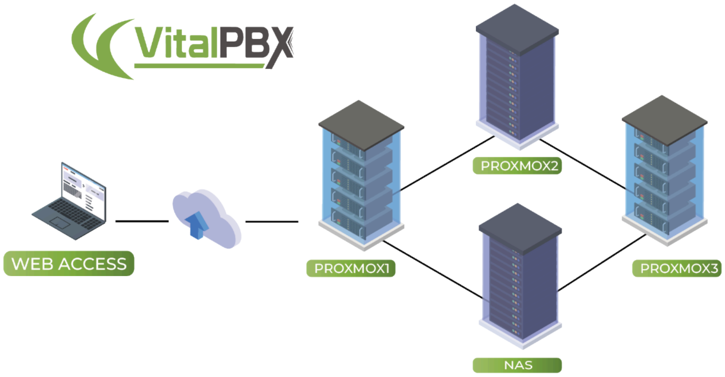


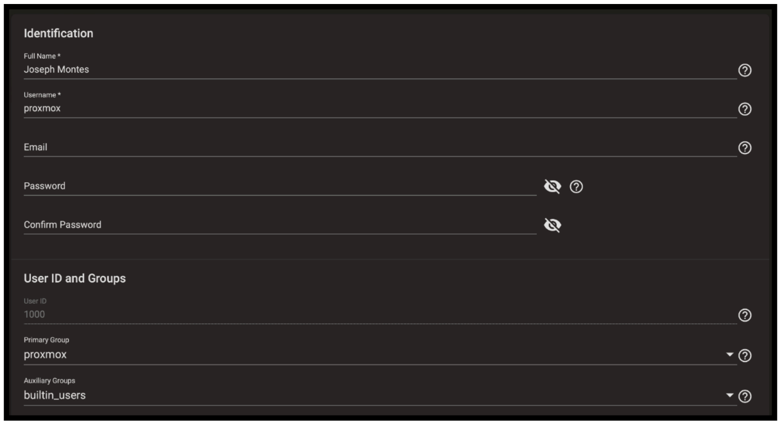
Image 13.6.9 - Proxmox High Availability diagram.

This diagram shows the infrastructure we are going to be building. We have the three Proxmox servers and an NFS server where we are going to be storing our Virtual Machines.

The next step is to configure our NFS storage. For this example, we will be using a **TrueNAS** server to create our NFS shares. You can use any NAS server that allows you to create an NFS share.

Since this part will vary based on the NAS or Storage Server you choose to use, we will go briefly on what needs to be configured.

On our TrueNAS installation, we will first create a **new user and password** that we are going to use to log into our NFS share. We go to **Accounts > Users**.



Identification

Full Name * Joseph Montes

Username * proxmox

Email

Password

Confirm Password

User ID and Groups

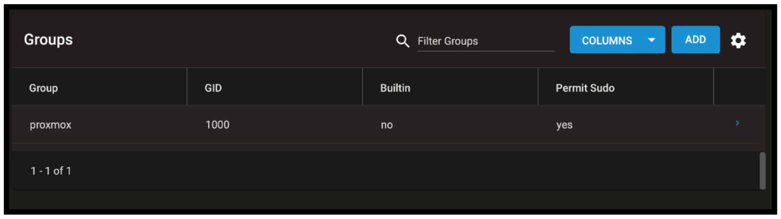
User ID 1000

Primary Group proxmox

Auxiliary Groups builtin_users

Image 13.6.10 - User creation in TrueNAS.

Here, a new group has been created as well that we have named **proxmox**. You can verify this under **Accounts > Groups**.

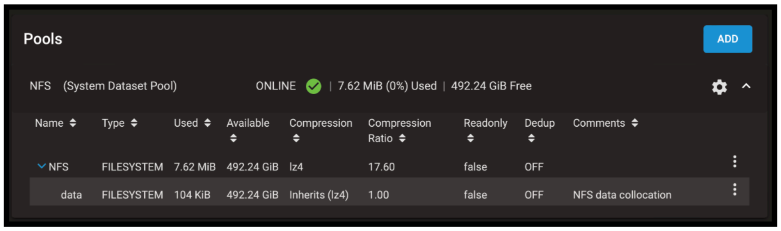


Group	GID	Builtin	Permit Sudo
proxmox	1000	no	yes

1 - 1 of 1

Image 13.6.11 - TrueNAS group.

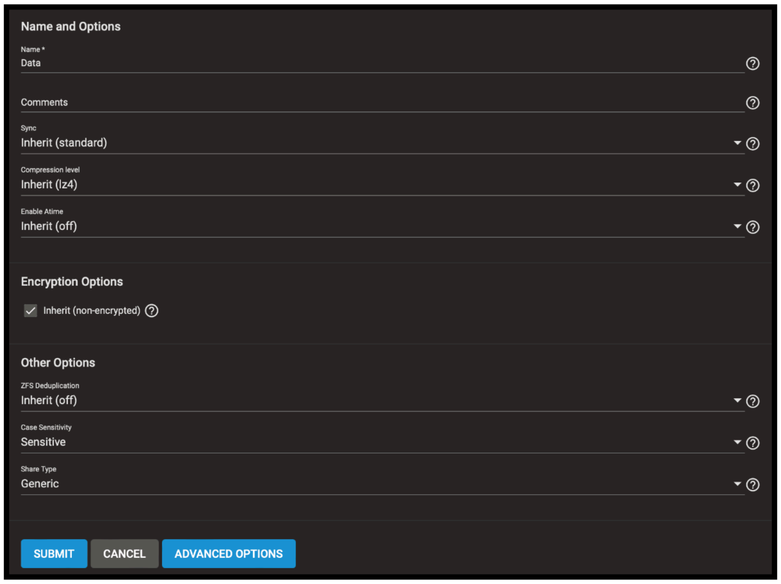
With the user and group created, we then created a **new Pool with a Dataset** where we are going to be storing our Virtual Machines. For this, we go to **Storage > Pools**.



NFS (System Dataset Pool) ONLINE ✓ | 7.62 MiB (0%) Used | 492.24 GiB Free

Name	Type	Used	Available	Compression	Compression Ratio	Readonly	Dedup	Comments
NFS	FILESYSTEM	7.62 MiB	492.24 GiB	lz4	17.60	false	OFF	
data	FILESYSTEM	104 KiB	492.24 GiB	Inherits (lz4)	1.00	false	OFF	NFS data collocation

Image 13.6.13 - TrueNAS Pool and Dataset.



Name and Options

Name * Data

Comments

Sync: Inherit (standard)

Compression level: Inherit (lz4)

Enable Atime: Inherit (off)

Encryption Options

☑ Inherit (non-encrypted)

Other Options

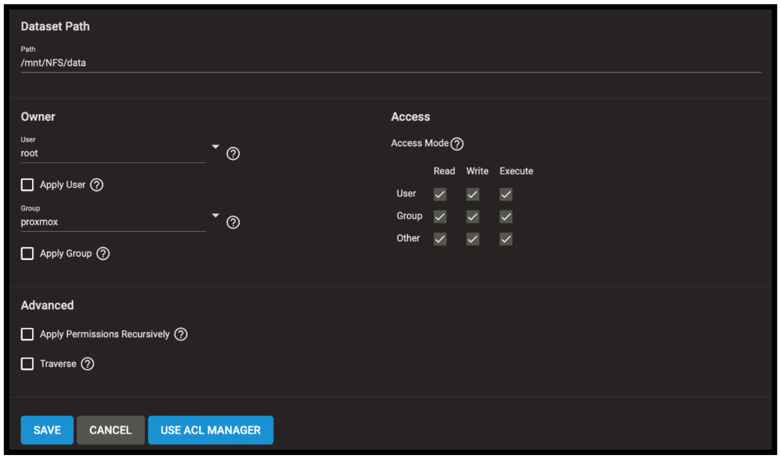
ZFS Deduplication: Inherit (off)

Case Sensitivity: Sensitive

Share Type: Generic

Image 13.6.14 - TrueNAS Dataset configurations.

Then, we edit the permissions for the dataset we named **data**, add our **group**, and modify the **Access Mode** to allow us to read and write data to the dataset. Remember to check the **Apply Group** box before clicking on **Save**.



Dataset Path

Path: /mnt/NFS/data

Owner

User root

□ Apply User

Group: proxmox

□ Apply Group

Access

Access Mode

	Read	Write	Execute
User	☑	☑	☑
Group	☑	☑	☑
Other	☑	☑	☑

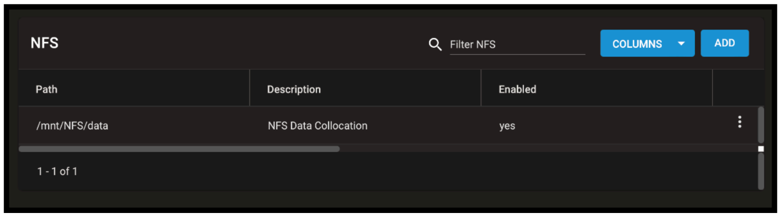
Advanced

□ Apply Permissions Recursively

□ Traverse

Image 13.6.15 - TrueNAS dataset permissions.

Next, we go to **Sharing > Unix Shares (NFS)** to create our NFS share. We point this share to the dataset we created previously.

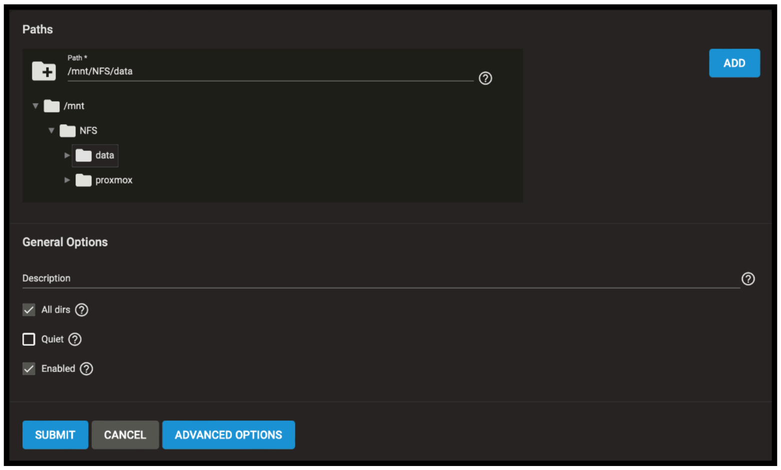


NFS

Path	Description	Enabled
/mnt/NFS/data	NFS Data Collocation	yes

1 - 1 of 1

Image 13.6.16 - TrueNAS NFS shares.



Paths

Path *: /mnt/NFS/data

General Options

Description

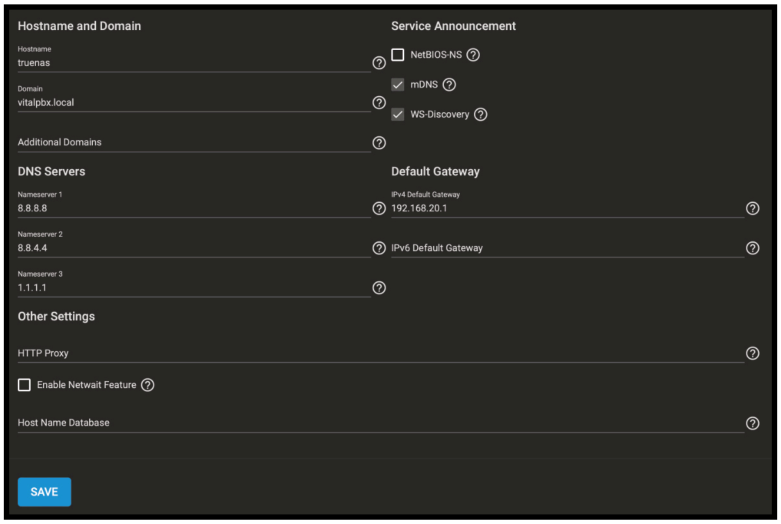
☑ All dirs

□ Quiet

☑ Enabled

Image 13.6.17 - TrueNAS NFS share configuration.

Finally, remember to have the networking configured so you can reach this server from our Proxmox nodes. Under **Network > Global Configurations**, we configure our **DNS servers**, **hostname**, and **Default Gateway**.



Hostname and Domain

Hostname: truenas

Domain: vitalpbx.local

Additional Domains

Service Announcement

□ NetBIOS-NS

☑ mDNS

☑ WS-Discovery

DNS Servers

Nameserver 1: 8.8.8.8

Nameserver 2: 8.8.4.4

Nameserver 3: 1.1.1.1

Default Gateway

IPv4 Default Gateway: 192.168.20.1

IPv6 Default Gateway

Other Settings

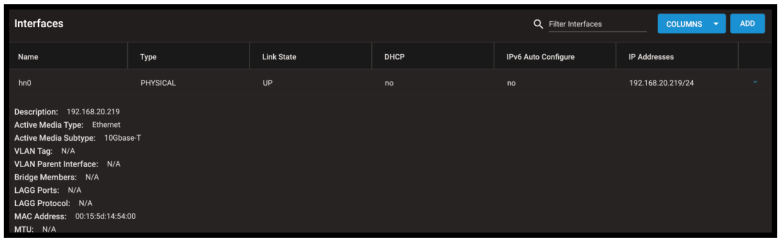
HTTP Proxy

□ Enable Netwait Feature

Host Name Database

Image 13.6.18 - TrueNAS Network Global configurations.

And under **Network > Interfaces**, we configure the static IP Address for our server.



Interfaces

Name	Type	Link State	DHCP	IPv6 Auto Configure	IP Addresses
hn0	PHYSICAL	UP	no	no	192.168.20.219/24

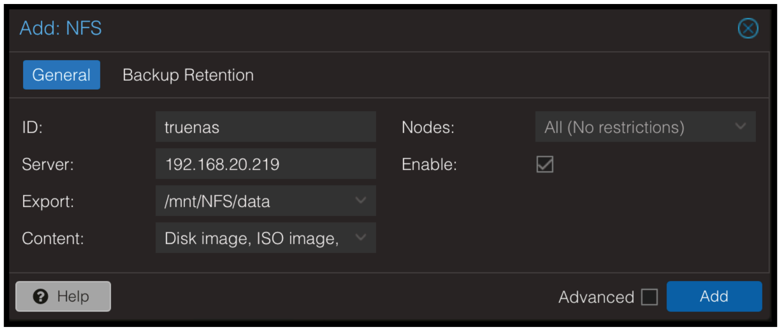
Description: 192.168.20.219
Active Media Type: Ethernet
Active Media Subtype: 10Gbase-T
VLAN Tag: N/A
VLAN Parent Interface: N/A
Bridge Members: N/A
LAGG Ports: N/A
LAGG Protocol: N/A
MAC Address: 00:15:5d:14:54:00
MTU: N/A

Image 13.6.19 - TrueNAS Network Interface configurations.

With this, you now have an NFS share available to store your Virtual Machines in Proxmox.

Now that we have three Proxmox servers and our NAS server with the NFS shares configured, we can configure our connection to the NAS server and create our cluster for High Availability using Proxmox.

Go to the first Proxmox node and go to **Datacenter > Storage**. Click **Add** and select **NFS**. Here, we configure the **ID** to identify this NFS share, add the **server IP address**, set the path for our data set under **Export**, and select the **content** we want to include in this storage.



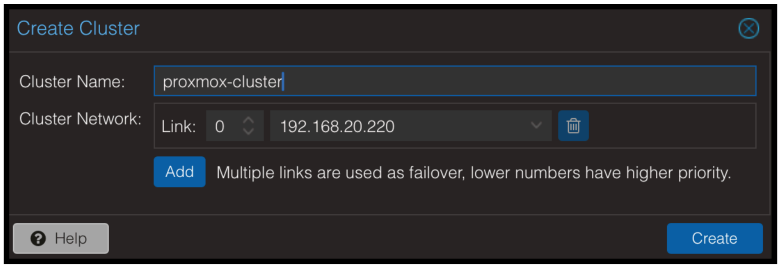
Add: NFS

ID:	truenas	Nodes:	All (No restrictions)
Server:	192.168.20.219	Enable:	☑
Export:	/mnt/NFS/data		
Content:	Disk image, ISO image,		

Image 13.6.20 - Adding the NFS share to Proxmox.

Under node, you will only see the current proxmox node available. You can go to every node and add the NFS share **using the same ID**. You can also come back to this Share on the first node after we create the cluster, and add the other two nodes afterward.

Next, we configure our Cluster. On the first node, go to **Datacenter > Cluster** and click on **Create Cluster**. Once you enter the **Cluster Name**, click on **Create**.



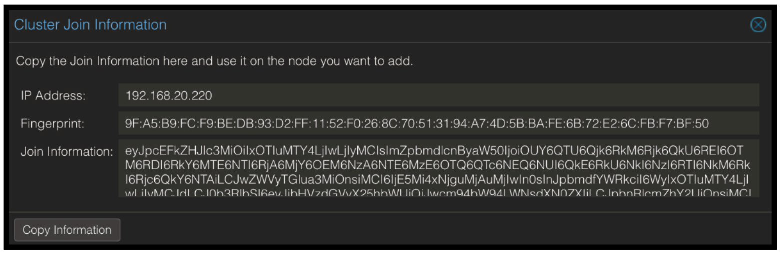
Create Cluster

Cluster Name: proxmox-cluster
Cluster Network: Link: 0 192.168.20.220

Multiple links are used as failover, lower numbers have higher priority.

Image 13.6.21 - Cluster creation in Proxmox.

Go back to **Datacenter > Cluster**, and click on **Join Information**. This will show you the information we need to add to the other two nodes. Click on **Copy Information**.



Cluster Join Information

Copy the Join Information here and use it on the node you want to add.

IP Address: 192.168.20.220

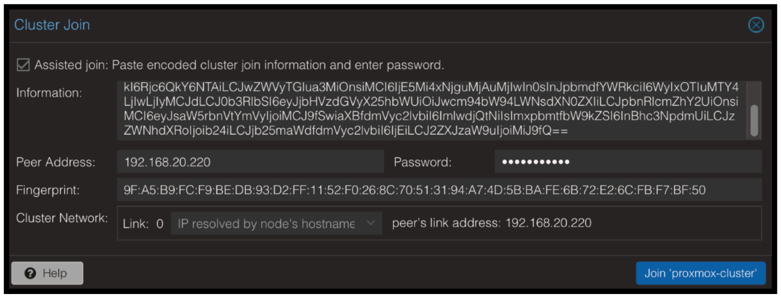
Fingerprint: 9F:A5:B9:FC:F9:BE:DB:93:D2:FF:11:52:F0:26:8C:70:51:31:94:A7:4D:5B:BA:FE:6B:72:E2:6C:FB:F7:BF:50

Join Information:

```
eyJpcEFkZHJlc3MiOiIxOTIuMTY4LjIwLjIyMCIsImZpbmdlcnByaW50IjoiOUY6QTU6Qjk6RkM6Rjk6QkU6REI6OT
M6RDI6RkY6MTE6NTI6RjA6MjY6OEM6NzA6NTE6MzE6OTQ6QTc6NEQ6NUI6QkE6RkU6NkI6NzI6RTI6NkM6Rk
I6Rjc6QkY6NTAiLCJwZWVyTGlua3MiOnsiMCI6IjE5Mi4xNjguMjAuMjIwIn0sInJpbmdfYWRkciI6WyIxOTIuMTY4LjI
wLjIyMCJdLCJ0b3RlbSI6eyJjbHVzdGVyX25hbWUiOiJwcm94bW94LWNsdXN0ZXIiLCJpbnRlcmZhY2UiOnsiMCI
```

Image 13.6.22 - Cluster Join Information in Proxmox.

Go to the second Proxmox node go to **Datacenter > Cluster** and click **Join Cluster**. Here, you will need to enter the Join Information for the first node.



Cluster Join

☑ Assisted join: Paste encoded cluster join information and enter password.

Information:

```
kI6Rjc6QkY6NTAiLCJwZWVyTGlua3MiOnsiMCI6IjE5Mi4xNjguMjAuMjIwIn0sInJpbmdfYWRkciI6WyIxOTIuMTY4
LjIwLjIyMCJdLCJ0b3RlbSI6eyJjbHVzdGVyX25hbWUiOiJwcm94bW94LWNsdXN0ZXIiLCJpbnRlcmZhY2UiOnsi
MCI6eyJsaW5rbnVtYmVyIjoiMCJ9fSwiaXBfdmVyc2lvbiI6ImlwdjQtNiIsImxpbmtfbW9kZSI6InBhc3NpdmUiLCJz
ZWNhdXRoIjoib24iLCJjb25maWdfdmVyc2lvbiI6IjEiLCJ2ZXJzaW9uIjoiMiJ9fQ==
```

Peer Address: 192.168.20.220

Password: ••••••••••••

Fingerprint: 9F:A5:B9:FC:F9:BE:DB:93:D2:FF:11:52:F0:26:8C:70:51:31:94:A7:4D:5B:BA:FE:6B:72:E2:6C:FB:F7:BF:50

Cluster Network: Link: 0 IP resolved by node's hostname peer's link address: 192.168.20.220

Image 13.6.23 - Cluster Join in Proxmox, as seen on the second node.

You will need to enter the root password for the first Proxmox node here. Click on **Join 'proxmox-cluster'**. After a few minutes, the second node is added to the cluster.

Repeat these two steps on the third Proxmox node, by adding the Join Information and root password from the first node. Then click on **Join 'proxmox-cluster'** and wait a few minutes.

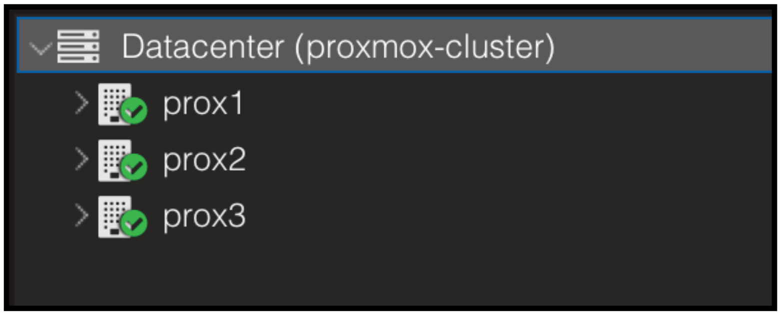
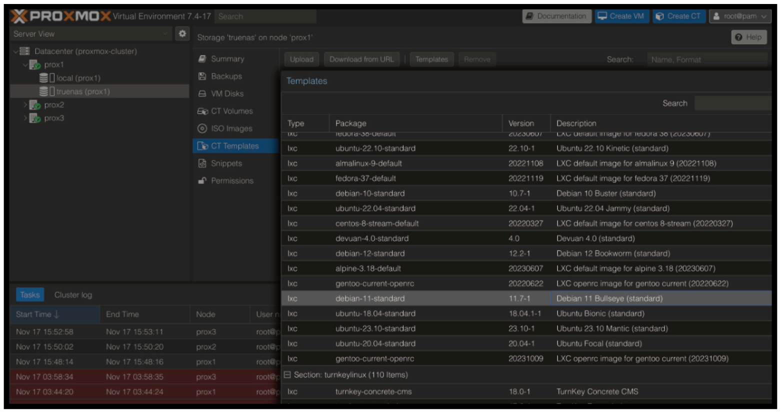


Image 13.6.24 - All Proxmox nodes in a cluster, as seen on the first node.

You will now see that all your Proxmox nodes are listed under Datacenter. From here, if you have not added the second and third nodes to the NFS storage configured under **Datacenter > Storage**. Go back to that share, and edit it to add the second and third nodes.

Now, we will create a VitalPBX instance. On the first Proxmox node, go to **Datacenter > prox1 > truenas (prox1)**. Here, click on **CT Templates** and then click on **debian-11-standard**. Then, click on **Download**.



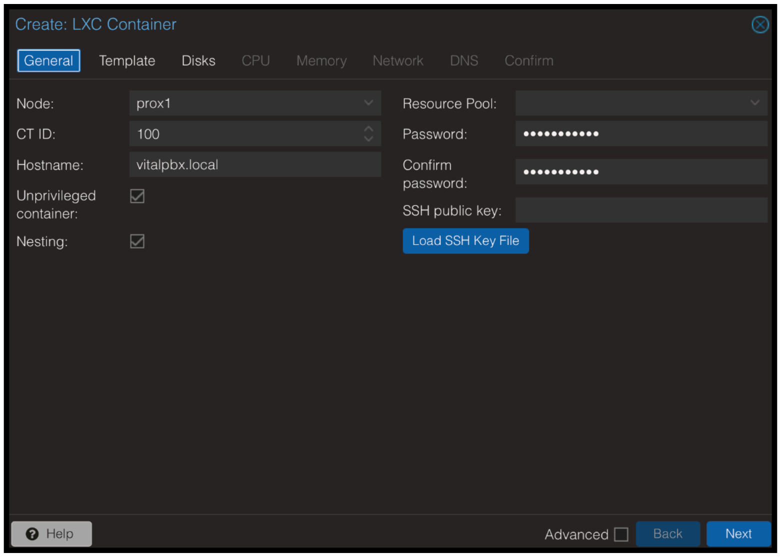
Type	Package	Version	Description
lxc	fedora-38-default	20230607	LXC default image for fedora 38 (20230607)
lxc	ubuntu-22.10-standard	22.10-1	Ubuntu 22.10 Kinetic (standard)
lxc	almalinux-9-default	20221108	LXC default image for almalinux 9 (20221108)
lxc	fedora-37-default	20221119	LXC default image for fedora 37 (20221119)
lxc	debian-10-standard	10.7-1	Debian 10 Buster (standard)
lxc	ubuntu-22.04-standard	22.04-1	Ubuntu 22.04 Jammy (standard)
lxc	centos-8-stream-default	20220327	LXC default image for centos 8-stream (20220327)
lxc	devuan-4.0-standard	4.0	Devuan 4.0 (standard)
lxc	debian-12-standard	12.2-1	Debian 12 Bookworm (standard)
lxc	alpine-3.18-default	20230607	LXC default image for alpine 3.18 (20230607)
lxc	gentoo-current-openrc	20220622	LXC openrc image for gentoo current (20220622)
lxc	debian-11-standard	11.7-1	Debian 11 Bullseye (standard)
lxc	ubuntu-18.04-standard	18.04.1-1	Ubuntu Bionic (standard)
lxc	ubuntu-23.10-standard	23.10-1	Ubuntu 23.10 Mantic (standard)
lxc	ubuntu-20.04-standard	20.04-1	Ubuntu Focal (standard)
lxc	gentoo-current-openrc	20231009	LXC openrc image for gentoo current (20231009)
⊟ Section: turnkeylinux (110 Items)			
lxc	turnkey-concrete-cms	18.0-1	TurnKey Concrete CMS

Start Time ↓	End Time	Node	User n
Nov 17 15:52:58	Nov 17 15:53:11	prox3	root@p
Nov 17 15:50:02	Nov 17 15:50:20	prox2	root@p
Nov 17 15:48:14	Nov 17 15:48:16	prox1	root@p
Nov 17 03:58:34	Nov 17 03:58:35	prox3	root@p
Nov 17 03:44:20	Nov 17 03:44:24	prox1	root@p

Image 13.6.25 - Downloading a Debian 11 container template in Proxmox.

With this, we can now create a container where we are going to install VitalPBX 4. Go back to **Datacenter > prox1 > truenas (prox1)** and click on **Create CT** in the upper right-hand corner.

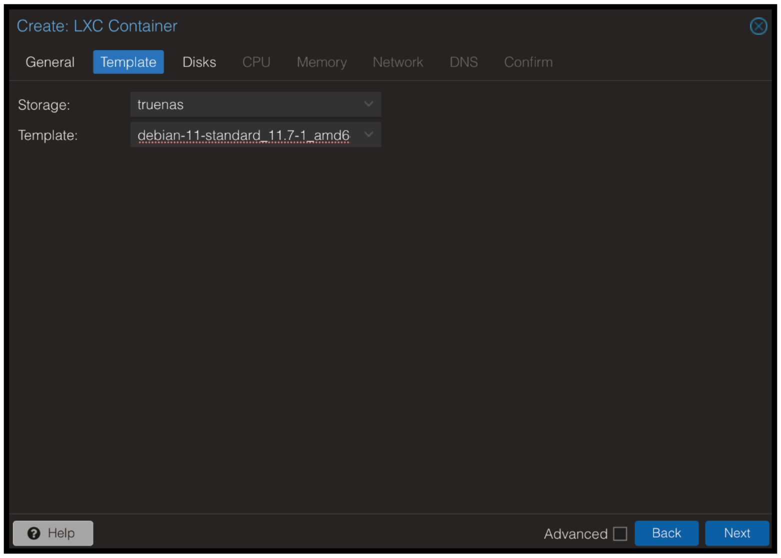
First, in the **General tab**, we are going to add the **Hostname** and root **Password**.



Node:	prox1	Resource Pool:	
CT ID:	100	Password:	•••••••••••
Hostname:	vitalpbx.local	Confirm password:	•••••••••••
Unprivileged container:	☑	SSH public key:	
Nesting:	☑		

Image 13.6.26 - Creating LXC Container in Proxmox. General tab.

Next, on the **Template tab**, select the NFS **Storage** we created and the **Template** we downloaded.

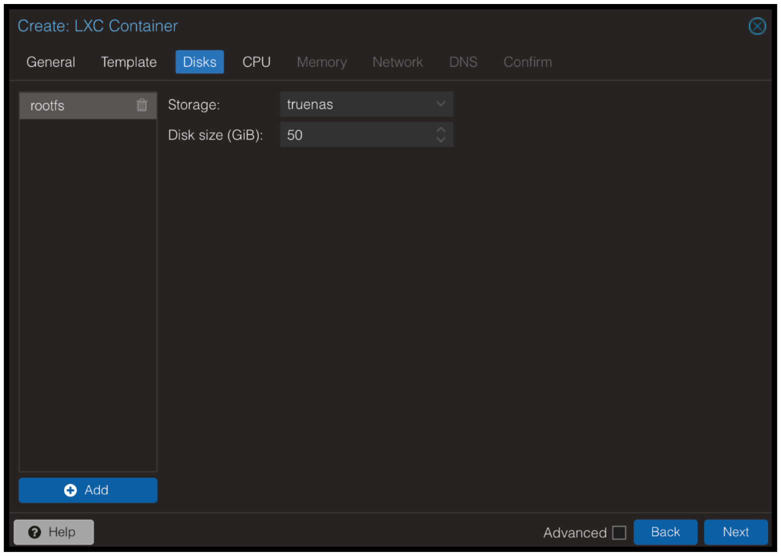


Storage: truenas

Template: debian-11-standard_11.7-1_amd6

Image 13.6.27 - Creating LCX Container in Proxmox. Template tab.

Then, on the **Disks tab**, select the NFS **Storage** and enter the **Disk size in GiB** for the Virtual Machine.

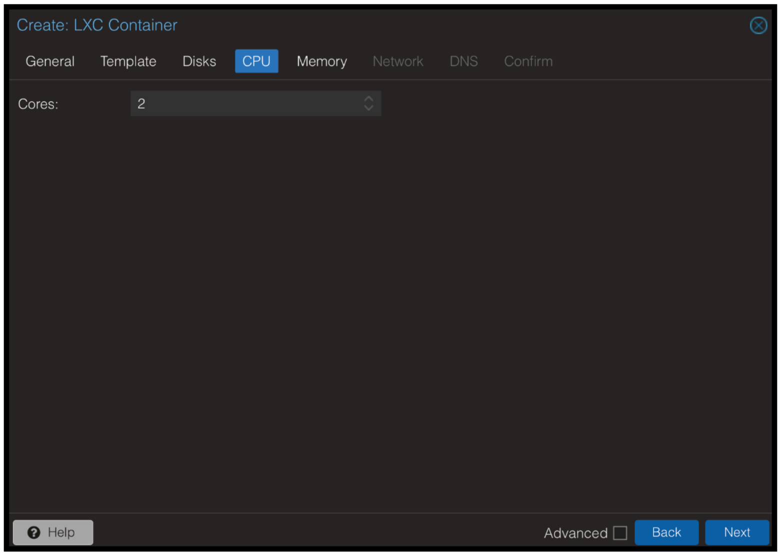


Storage: truenas

Disk size (GiB): 50

Image 13.6.28 - Creating LCX Container in Proxmox. Disks tab.

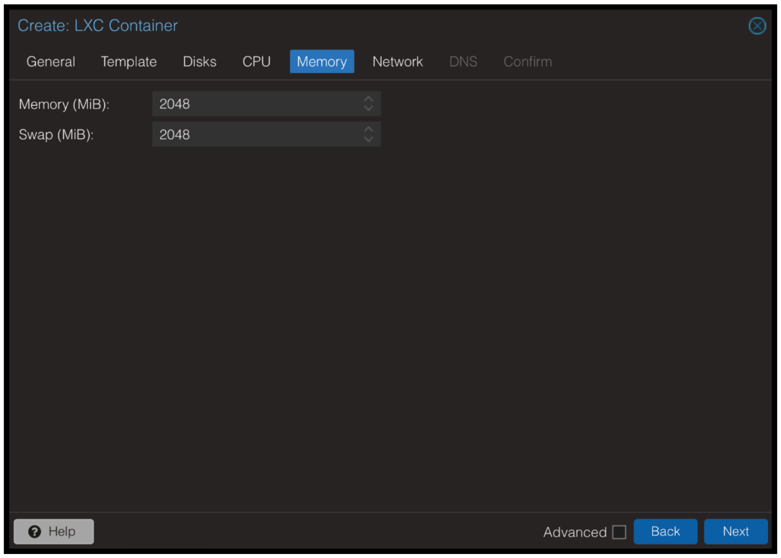
Afterward, on the **CPU tab**, enter the number of **cores** you want to give to this Virtual Machine.



Cores: 2

Image 13.6.29 - Creating LCX Container in Proxmox. CPU tab.

Under the **Memory tab**, enter the amount of **Memory and Swap in MiB** you wish to give to this Virtual Machine.

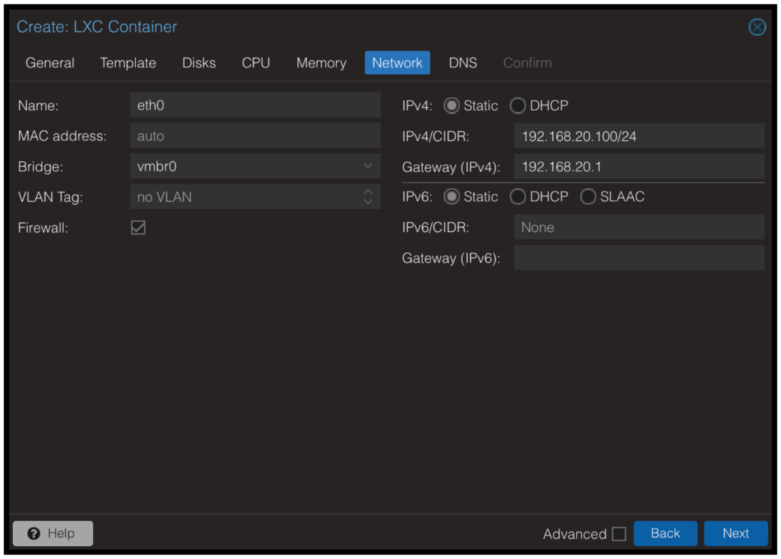


Memory (MiB): 2048

Swap (MiB): 2048

Image 13.6.30 - Creating LCX Container in Proxmox. Memory tab.

Then, on the **Network tab**, enter a **static IP address and Gateway** for the Virtual Machine.



Create: LXC Container

Name: eth0

MAC address: auto

Bridge: vmbr0

VLAN Tag: no VLAN

Firewall: ☑

IPv4: ● Static ○ DHCP

IPv4/CIDR: 192.168.20.100/24

Gateway (IPv4): 192.168.20.1

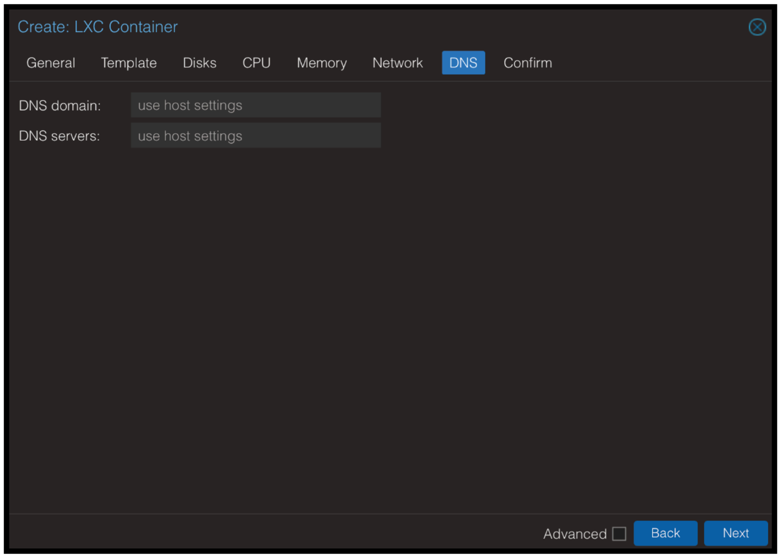
IPv6: ● Static ○ DHCP ○ SLAAC

IPv6/CIDR: None

Gateway (IPv6):

Image 13.6.31 - Creating LCX Container in Proxmox. Network tab.

Under the **DNS tab**, you can configure the **DNS servers** the Virtual Machine will use. You can leave these fields blank to use the host settings.



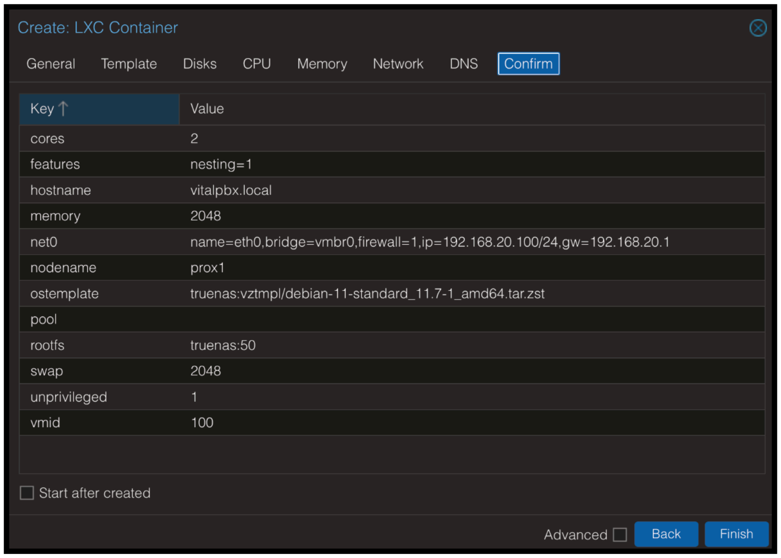
Create: LXC Container

DNS domain: use host settings

DNS servers: use host settings

Image 13.6.32 - Creating LCX Container in Proxmox. DNS tab.

Finally, you can verify the configurations under the **Confirm tab**.



Create: LXC Container

Key ↑	Value
cores	2
features	nesting=1
hostname	vitalpbx.local
memory	2048
net0	name=eth0,bridge=vmbr0,firewall=1,ip=192.168.20.100/24,gw=192.168.20.1
nodename	prox1
ostemplate	truenas:vztmpl/debian-11-standard_11.7-1_amd64.tar.zst
pool	
rootfs	truenas:50
swap	2048
unprivileged	1
vmid	100

□ Start after created

Image 13.6.33 - Creating LCX Container in Proxmox. Confirm tab.

Click on **Finish**, and wait a few minutes. The instance with Debian 11 is now ready.

With the Debian 11 instance created, we can install VitalPBX 4 on it. On the left navigation menu, you will now see the instance listed. **Right-click** the instance, and click on **Start**.

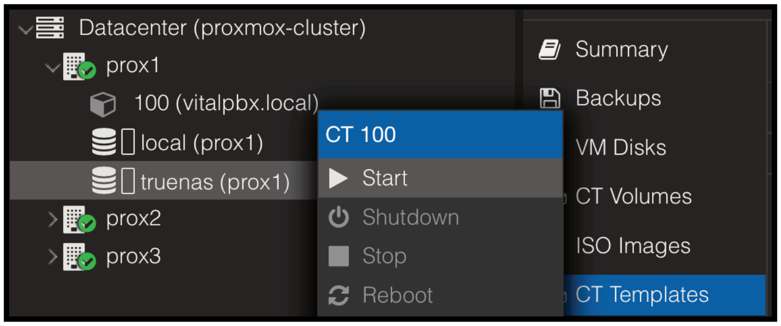


Image 13.6.34 - Starting container instance in Proxmox.

Once started, a console window will pop up. Run the following commands to install VitalPBX.

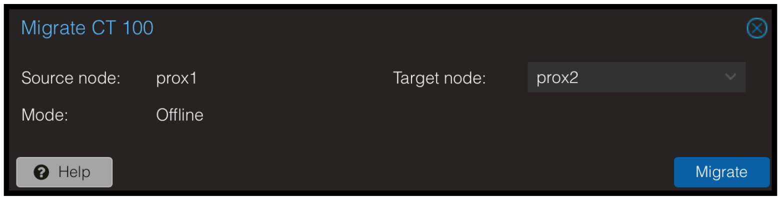
```
root@debian:~# apt update

root@debian:~# apt -y upgrade

root@debian:~# wget https://repo.vitalpbx.com/vitalpbx/v4/apt/install_vpbx4.sh

root@debian:~# chmod +x install_vpbx4.sh

root@debian:~# ./install_vpbx4.sh
```

After a few minutes, VitalPBX will have been installed and the Virtual Machine will reboot. You now have a VitalPBX instance on your Proxmox server!

With the instance created, you can now migrate it around the cluster between nodes. You can do this **Hot** (With the server ON) or **Cold** (With the server OFF).

To perform a **Cold Migration**, turn off the VitalPBX instance, by running a **poweroff** command from the console.

Right-click the instance, and click **Migrate**. You will be presented with a prompt selecting where you want to migrate the instance.



Migrate CT 100

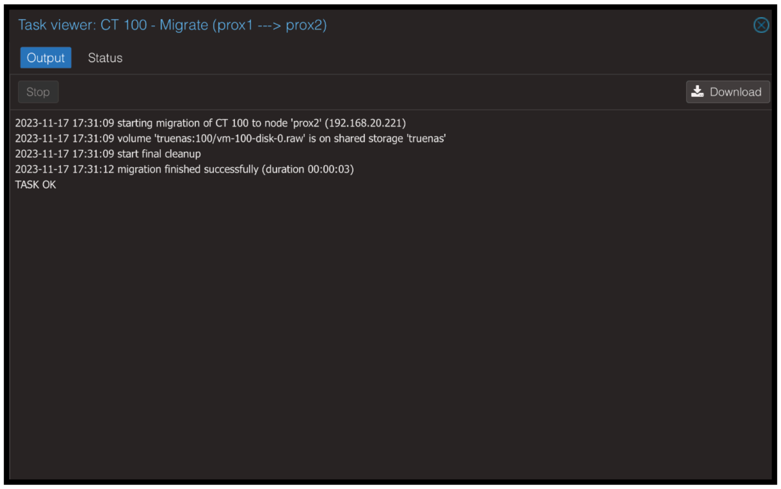
Source node: prox1

Target node: prox2

Mode: Offline

Image 13.6.35 - Cold migration of an instance between nodes in Proxmox.

Once the migration is complete, you will see the following message.



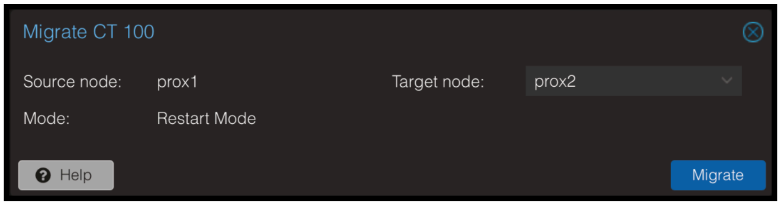
Task viewer: CT 100 - Migrate (prox1 ---> prox2)

```
2023-11-17 17:31:09 starting migration of CT 100 to node 'prox2' (192.168.20.221)
2023-11-17 17:31:09 volume 'truenas:100/vm-100-disk-0.raw' is on shared storage 'truenas'
2023-11-17 17:31:09 start final cleanup
2023-11-17 17:31:12 migration finished successfully (duration 00:00:03)
TASK OK
```

Image 13.6.36 - Cold migration of an instance between nodes OK in Proxmox.

The instance will now be in the second Proxmox node.

For a **Hot Migration**, while the VitalPBX instance is **turned on** in the first Proxmox node, **right-click** on it and click **Migrate**.



Migrate CT 100

Source node: prox1

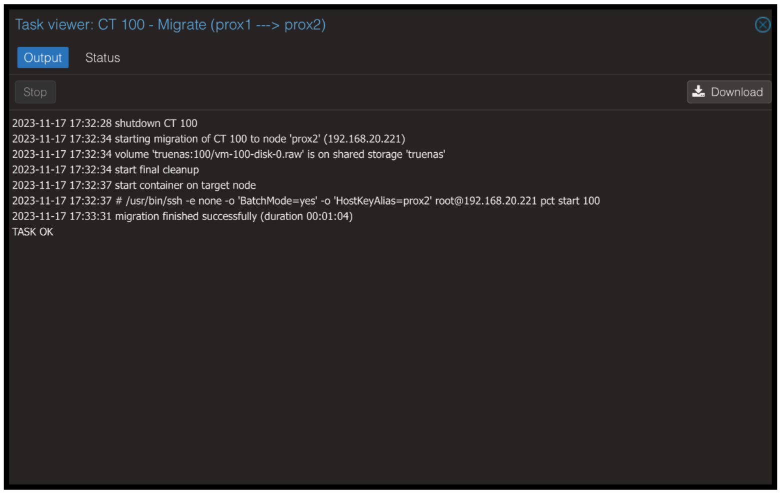
Target node: prox2

Mode: Restart Mode

Image 13.6.37 - Hot migration of an instance between nodes in Proxmox.

You will see that the mode now says **Restart Mode**. This means that the server will be restarted once it has migrated.

After the migration, you will see the following message.



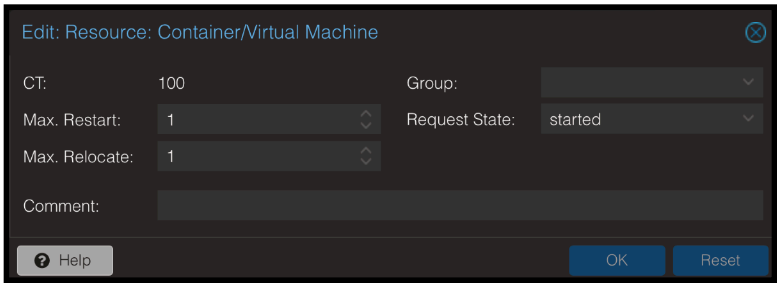
```
2023-11-17 17:32:28 shutdown CT 100
2023-11-17 17:32:34 starting migration of CT 100 to node 'prox2' (192.168.20.221)
2023-11-17 17:32:34 volume 'truenas:100/vm-100-disk-0.raw' is on shared storage 'truenas'
2023-11-17 17:32:34 start final cleanup
2023-11-17 17:32:37 start container on target node
2023-11-17 17:32:37 # /usr/bin/ssh -e none -o 'BatchMode=yes' -o 'HostKeyAlias=prox2' root@192.168.20.221 pct start 100
2023-11-17 17:33:31 migration finished successfully (duration 00:01:04)
TASK OK
```

Image 13.6.38 - Hot migration of an instance between nodes OK in Proxmox.

With the migration done, the instance is now on the second Proxmox node already started.

Now that we have a VitalPBX instance that we can migrate between nodes, we can set up our cluster to work in High Availability in Proxmox.

Go to **Datacenter > HA > Resources** and click **Add.** Here, enter the VitalPBX instance's **ID number** and click **Add.**



CT:	100	Group:	
Max. Restart:	1	Request State:	started
Max. Relocate:	1		
Comment:			

Image 13.6.39 - Adding the Virtual Machine instance as a resource for HA.

You will see that the cluster has the quorum OK and that the VitalPBX instance is now a resource for High Availability.

To test the High Availability, you can turn off the first Proxmox node and after some time the VitalPBX instance will automatically move from the first node to another one.

The time it takes to move the instance will depend on your server hardware and network speed.

When turning the first Proxmox node off, you will need to access the cluster from the second or third node to manage the cluster.

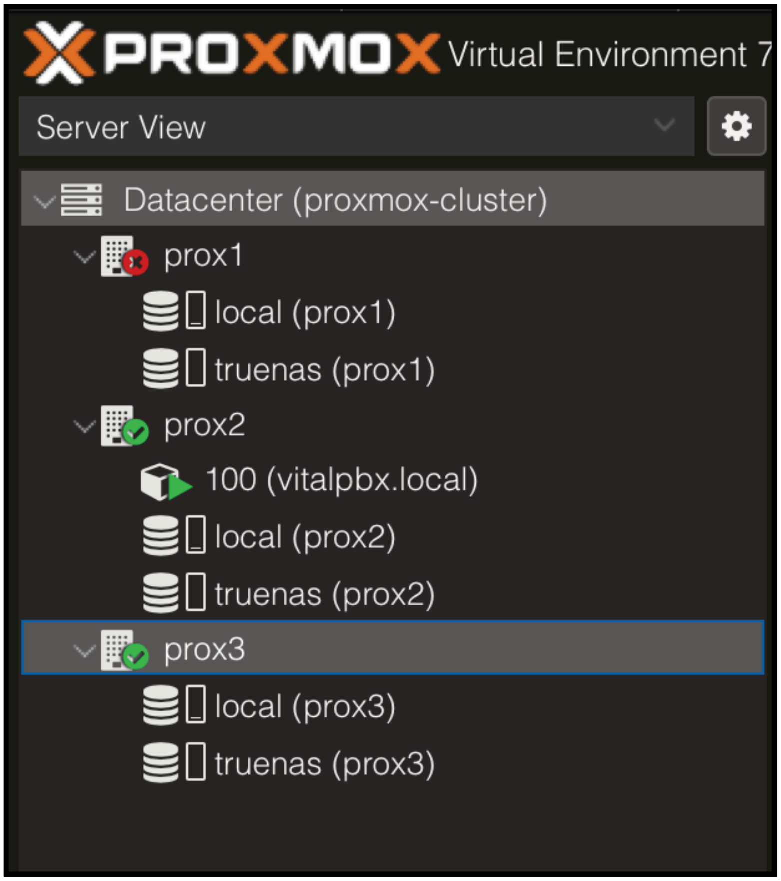


Image 13.6.40 - First Proxmox node is down, and the instance migrated to the second node.

With this done, you now have a full High Availability Environment using Proxmox!

One thing that differs between HA with Proxmox and using DRBD is that with Proxmox, there is no need to have multiple VitalPBX licenses. Since in this case, it is the same machine that is migrated over multiple Proxmox nodes, the licensing is moved alongside the virtual machine.

Section 13, Lesson 7 - Multi-Instance with Proxmox

We saw previously how you can have a Multi-Tenant environment with VitalPBX. Now, let's look at the other side of the coin, deploying a Multi-Instance server using Proxmox. This is quite simple as Proxmox VE is a virtual machine environment.

For this, you can follow the steps from the previous lesson to create a container. Once you have your container, install VitalPBX. With VitalPBX installed, you can now power off the container/virtual machine, and right-click the instance on the Proxmox navigation menu on the left. This will show you the **Create Template** action. By creating a template, you can easily deploy a new VitalPBX instance to your liking and as your server supports it.

Now, you can have many VitalPBX instances running. Keep in mind, when running multiple instances, you need to keep in mind the resources allocated. Each VitalPBX instance will be completely independent and each will need its own license. As such, each instance will need to be updated on its own.

With the template created, to deploy a new instance, you can right-click the instance again and click **Clone**.

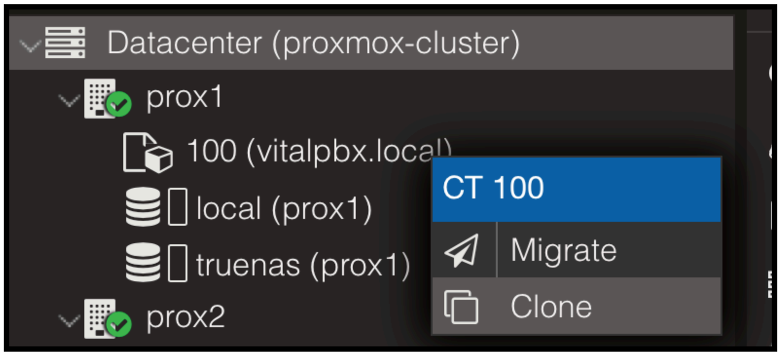


Image 13.7.1 - Clone action for an instance in Proxmox.

When creating a CT template, you can no longer start that machine. So you can use this first instance only to clone it into new instances. Once cloned, the new instance will have the IP address set on the original template. So we recommend you first set the network address for the new cloned CT or VM first thing. As well as using different hostnames when you create a new instance.

Conclusion

With a High Availability Environment, you can ensure that your downtime will be minimal or close to zero. This is a good way to gain peace of mind for you and your customers.

Following the steps that were used throughout this section, you can have a complex system to help you get this high-availability.

SECTION 14 - SECURITY

Introduction

VitalPBX possesses an inherent toolset for the best security practices on a PBX system. In this section, we will take a look at how you can take advantage of these tools to make your system more secure.

We will see the best practices we recommend, how to secure your calls with encryption, and some commercial add-ons to increment the security of your VitalPBX installation.

Section 14, Lesson 1 - Best Security Practices

First, let's look at various recommendations we present to make your VitalPBX installation more secure.

This section can work as a checklist of various configurations you can follow when setting up your VitalPBX installation.

1. **Never use the same username and password on your extensions.**
    - It is quite common to see instances where the username and password for extensions are the same as the extension number. This might make remembering the usernames and passwords easier when setting up your devices, but it is the easiest way to get bad actors registering on your PBX system. The bad actors can then start placing a massive number of phone calls.
    - What we recommend, and taking advantage of VitalPBX's separation of the Extension number and devices, is to use unique usernames and strong passwords for your extensions. You can make the device's user anything you like, instead of the extension number. VitalPBX also generates a random strong password automatically, so we recommend using this instead of a repeating password for all your devices.

2. **Use the "Permit" and "Deny" options for your devices.**
    - If possible, you should limit the networks that can reach the registration for your devices. In the case you know that a device will only register from a specific network address, you can use the Permit and Deny options when configuring your devices. Permit will only allow devices from the defined network address or segment to register. Deny will disallow devices from the defined segment to register.

3. **Limit extension registration using a Bind Address.**
    - The Bind Address option will also limit who can register to your extensions. With this option, you can limit the network addresses or segments that can register to your extension devices.

4. **Change the default ports for the services you are using.**
    - Default ports are one of the most common ways to have your system attacked by online scanners. By changing these ports to another value, bot scanners will have a harder time detecting open spots in your VitalPBX Server. You can change these ports on the VitalPBX firewall. Remember that you also need to change the ports on the Technology Settings module for PJSIP and IAX2. The most common ports to change are PJSIP, IAX2, and SSH.

5. **Disable the ports you are not using.**
    - Speaking of ports, if you are not using a service, disabling the port is a better option than changing it to something else. For example, if you are not using IAX2, disable the port on the VitalPBX firewall. This is one less way to detect an open spot by bot scanners.

6. **Don't route inbound calls to very permissive contexts.**
    - When routing incoming calls make sure that you are limiting the incoming calls to only the intended destinations. Using the right Class of Service can help you limit the destination options that someone can reach in a context. For example, don't have an IVR with a permissive Class of Service. Create one that limits the options to the destinations you intend. If you are using a Custom Context, only allow dialing to a specific destination.

7. **Always have the Firewall active and try to place your PBX behind a Firewall and/or SBC.**
    - The firewall included in VitalPBX is set to block unwanted access to your PBX. Having it enabled at all times will deter attempts to breach the server. Having an external firewall is another good way to manage the network routes and permissions at a network level to limit access to the VitalPBX server. Finally, an SBC or Session Border Controller is a good way to externally filter registration and other type of events from reaching your VitalPBX server.

8. **Use Fail2Ban to automatically detect malicious attempts to enter your PBX.**
    - Using the Fail2Ban application allows you to easily jail malicious attempts towards your VitalPBX server. Fail2Ban will block the connection from an IP Address after multiple failed attempts to access the server. This can be through SSH, PJSIP/IAX2 Registration, or web login. You can set the number of failed attempts and for how long the IP address will be blocked.

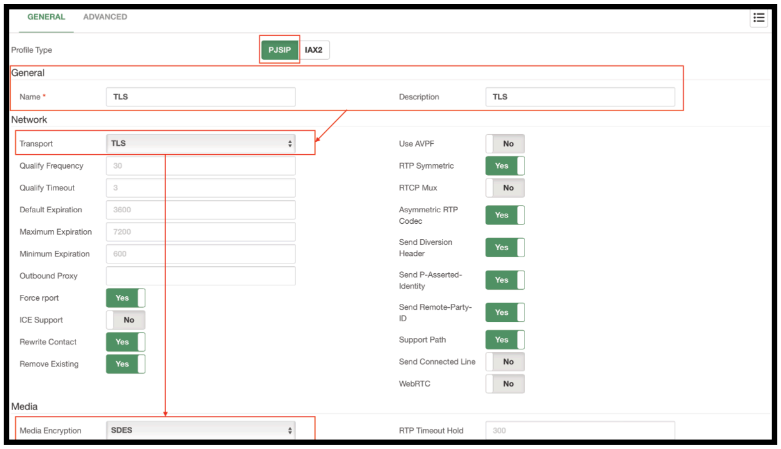
Following these suggestions will allow you to have a more secure server and keep your data and work safely. These are ways that you can secure your server out of the box. In the following lessons, we will look into more ways to make your server even safer.

Section 14, Lesson 2 - Secure Calls (TLS)

Keeping your conversations private is key for secure communications. With VitalPBX you can configure your calls to be encrypted, so they are secure from end to end. For this, we are going to be using TLS or Transported Layer Security.

The first thing we need to do is to create a new **Device Profile for PJSIP**. Go to **Settings > Technology Settings > Device Profiles.**

Here, we will select the **PJSIP Profile Type**. Then, enter a **Name** and **Description** to identify this device profile. Under **Network**, we will set the **Transport** to **TLS**. And under **Media**, we will set the **Media Encryption** to **SDES**.



Profile Type: PJSIP | IAX2

General

Name *: TLS
Description: TLS

Network

Transport: TLS
Qualify Frequency: 30
Qualify Timeout: 3
Default Expiration: 3600
Maximum Expiration: 7200
Minimum Expiration: 600
Outbound Proxy:
Force rport: Yes
ICE Support: No
Rewrite Contact: Yes
Remove Existing: Yes
Use AVPF: No
RTP Symmetric: Yes
RTCP Mux: No
Asymmetric RTP Codec: Yes
Send Diversion Header: Yes
Send P-Asserted-Identity: Yes
Send Remote-Party-ID: Yes
Support Path: Yes
Send Connected Line: No
WebRTC: No

Media

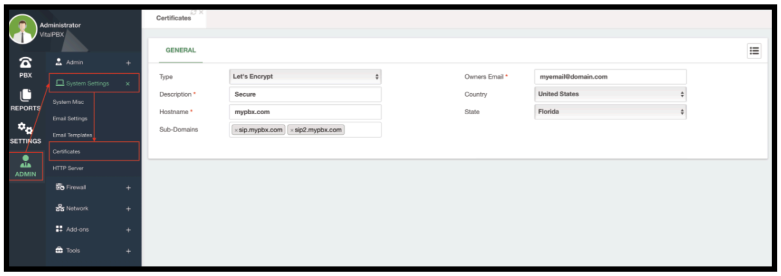
Media Encryption: SDES
RTP Timeout Hold: 300

Image 14.1.1 - TLS PJSIP device profile.

Then click on **Save**, and then **Apply Changes**.

Next, we will create a new **SSL Certificate**. Go to **Admin > System Settings > Certificates**. In this example, we will be creating a **Let's Encrypt** certificate. In this module, you can create **self-signed** certificates and **custom** SSL certificates you may acquire with an SSL Certificate vendor. Self-signed may be used in local network environments, but they are not recommended as many browsers consider sites using self-signed certificates risky.

For the Let's Encrypt certificate, we need to enter a **Description** to identify the certificate, enter the **Hostname** for the VitalPBX server, and enter the **Owner's Email** address.



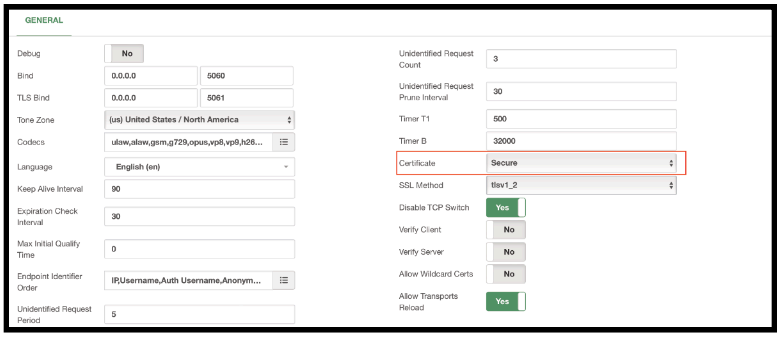
Type: Let's Encrypt
Description *: Secure
Hostname *: mypbx.com
Sub-Domains: sip.mypbx.com, sip2.mypbx.com
Owners Email *: myemail@domain.com
Country: United States
State: Florida

Image 14.1.2 - Let's Encrypt certificate in the Certificates module.

When creating a Let's Encrypt certificate, we support **Sub-Domains**. This means that under hostname, you can enter the main valid FQDN. Then, you can enter a sub-domain, **i.e. sip.mydomain.com**, that you can use to access the VitalPBX installation if you use this certificate for an HTTPS connection. You can enter multiple sub-domains in this section. This is helpful if you want to separate different tenants by sub-domain, and still use a single VitalPBX instance.

With the fields configured, click **Save**.

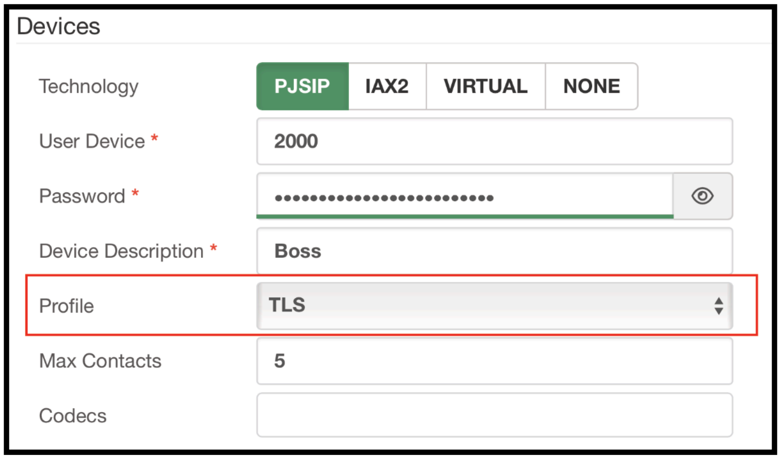
Now, go to **Settings > Technology Settings > PJSIP Settings**. Under **Certificate** select the certificate we just created. If you are using multiple sub-domains, you must also enable the **Allow Wildcard Certs** option. Then, click on **Save** and then **Apply Changes**.


GENERAL

Debug	No		Unidentified Request Count	3
Bind	0.0.0.0	5060	Unidentified Request Prune Interval	30
TLS Bind	0.0.0.0	5061	Timer T1	500
Tone Zone	(us) United States / North America		Timer B	32000
Codecs	ulaw,alaw,gsm,g729,opus,vp8,vp9,h26...		Certificate	Secure
Language	English (en)		SSL Method	tlsv1_2
Keep Alive Interval	90		Disable TCP Switch	Yes
Expiration Check Interval	30		Verify Client	No
Max Initial Qualify Time	0		Verify Server	No
Endpoint Identifier Order	IP,Username,Auth Username,Anonym...		Allow Wildcard Certs	No
Unidentified Request Period	5		Allow Transports Reload	Yes

Image 14.1.3 - Setting the Certificate on PJSIP settings for TLS.

Afterward, we will need to assign the device profile we created to the devices we want to use TLS encryption with. Go to **PBX > Extensions > Extensions**, and under the device section on the **Profile** field, select the device profile we created.


Devices

Technology	PJSIP / IAX2 / VIRTUAL / NONE
User Device *	2000
Password *	•••••••••••••••••••••••••
Device Description *	Boss
Profile	TLS
Max Contacts	5
Codecs	

Image 14.1.4 - Device profile assigned to an extension device.

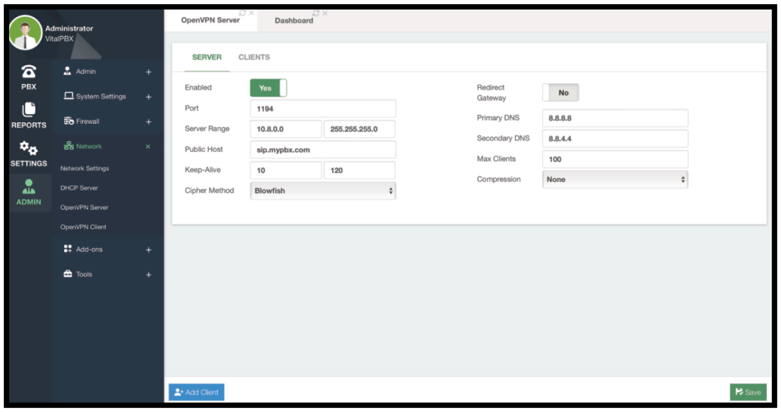
All that is left is to register the extensions using the TLS port instead of the default port for PJSIP. By default, this is port 5061. Some devices will require you to enable encrypted calls so you are able to use TLS for the voice packets and change the signaling from UDP to TLS. With this, your devices will have their calls encrypted, making their conversations even more private and secure.

Section 14, Lesson 3 - OpenVPN

Sometimes, you might need to have remote access and registration for your VitalPBX installation. This might be a local installation that you don't want to fully expose to the internet. For this, we have the **OpenVPN** commercial add-on that allows you to turn your VitalPBX server into a VPN server.

The OpenVPN add-on is a commercial add-on module, meaning that you will need to purchase an individual license for this module, or any of our licensing plans.

First, install the add-on from the add-ons module at **Admin > Add-Ons > Add-Ons**. Once installed, refresh your browser and go to **Admin > Network > OpenVPN Server**.



Enabled: Yes
Port: 1194
Server Range: 10.8.0.0 255.255.255.0
Public Host: sip.mypbx.com
Keep-Alive: 10 120
Cipher Method: Blowfish
Redirect Gateway: No
Primary DNS: 8.8.8.8
Secondary DNS: 8.8.4.4
Max Clients: 100
Compression: None

Image 14.3.1 - OpenVPN Server module.

Next, you must **Enable** the service. By default, the module comes with **Port 1194** configured. You can change this to anything of your liking. Remember, this needs to be changed on the Firewall services as well under **Admin > Firewall > Services**. You can set a **range of IP addresses** to be assigned to the clients by entering a network with the appropriate netmask for the number of clients you wish to create.

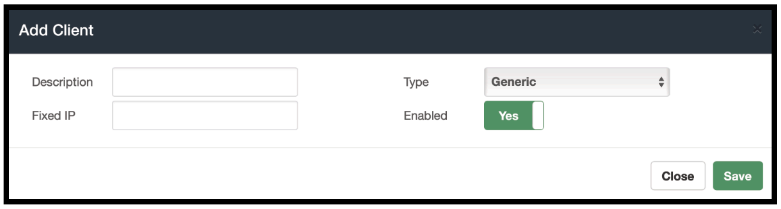
Afterward, you must enter the **Public Host**. This will be the Public IP Address to reach your VitalPBX installation or an FQDN that points to this server.

Then, set the **DNS servers** to use. By default, Google's© DNS servers are used. Lastly, we recommend setting the **compression to comp-lzo**. Click **Save** and then **Apply Changes**.

Optionally, you can set the following options.

- **Keep Alive -** This option sends a ping-like message through the connection to see if the connection with the client is still active. You can set the minimum and maximum times for this ping in seconds.
- **Cipher Method -** This is the cipher algorithm to encrypt the data.
- **Redirect Gateway -** If this option is enabled, all network traffic will go through the OpenVPN server. This is good if you want to use the outbound connection and public IP address of the VitalPBX Server. Keep this in mind based on your server's bandwidth's limits.
- **Max Clients -** This is the maximum number of concurrently connected clients.

With the service enabled, and reachable through the Public Host, you can start creating the certificates for your clients. For this, click the **Add Client** button in the lower left-hand corner.



Add Client

Description

Fixed IP

Type: Generic

Enabled: Yes

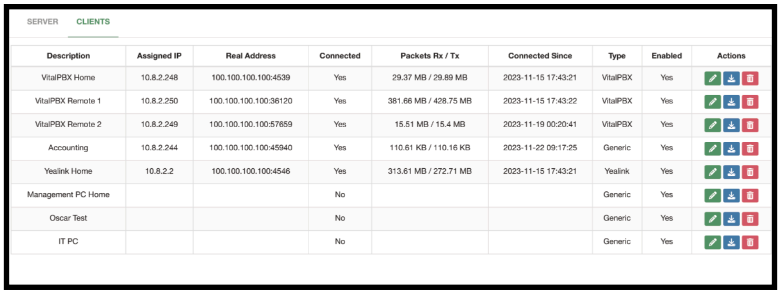
Image 14.3.2 - Adding an OpenVPN client.

Here, you must enter a **Description** to identify this certificate for the client. Optionally, you can enter a **Fixed IP address** that will be assigned to the client. You can **Enable or Disable** this client at any time. Finally, you must select the type of certificate you want to create for this client. You can chose between the following options.

- **Generic -** This is best for devices like computers or mobile phones. You can use the OpenVPN Connect application to upload the certificate and connect to this VPN server.
- **Grandstream -** This is a special certificate created for Grandstream devices.
- **Fanvil -** This is a special certificate created for Fanvil devices.
- **Yealink -** This is a special certificate created for Yealink devices.
- **VitalPBX -** This is a special certificate for other VitalPBX installations you want to connect to this OpenVPN server.

Once you have configured these fields, click **Save** and **Apply Changes**.

Under the **Clients tab**, you will see the different Clients you have created.



Description	Assigned IP	Real Address	Connected	Packets Rx / Tx	Connected Since	Type	Enabled	Actions
VitalPBX Home	10.8.2.248	100.100.100.100:4539	Yes	29.37 MB / 29.89 MB	2023-11-15 17:43:21	VitalPBX	Yes	✎ ⬇ 🗑
VitalPBX Remote 1	10.8.2.250	100.100.100.100:36120	Yes	381.66 MB / 428.75 MB	2023-11-15 17:43:22	VitalPBX	Yes	✎ ⬇ 🗑
VitalPBX Remote 2	10.8.2.249	100.100.100.100:57659	Yes	15.51 MB / 15.4 MB	2023-11-19 00:20:41	VitalPBX	Yes	✎ ⬇ 🗑
Accounting	10.8.2.244	100.100.100.100:45940	Yes	110.61 KB / 110.16 KB	2023-11-22 09:17:25	Generic	Yes	✎ ⬇ 🗑
Yealink Home	10.8.2.2	100.100.100.100:4546	Yes	313.61 MB / 272.71 MB	2023-11-15 17:43:21	Yealink	Yes	✎ ⬇ 🗑
Management PC Home			No			Generic	Yes	✎ ⬇ 🗑
Oscar Test			No			Generic	Yes	✎ ⬇ 🗑
IT PC			No			Generic	Yes	✎ ⬇ 🗑

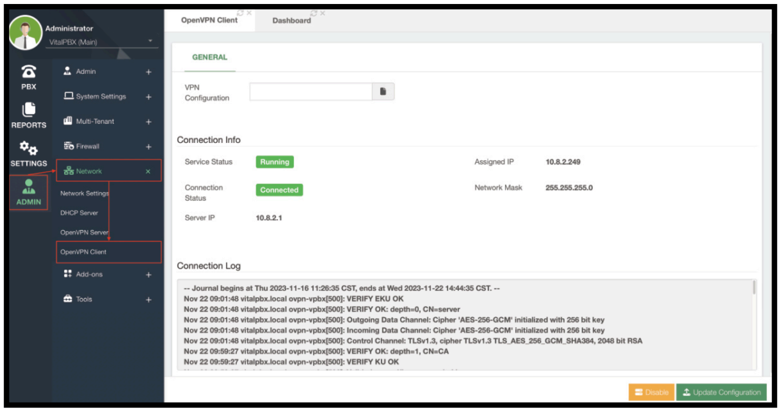
14.3.3 - OpenVPN Server Clients.

Here you can see relevant information about the **client's connection**, as well as their **type**, and whether or not they are **enabled**. In the last column, you will see some actions you can take over the clients. You can **delete** or **edit** the client, as well as **download the certificate files**.

The files will differ based on the type of client you selected. You can then **upload** the certificate to the appropriate device based on the client type.

The device will then connect to the OpenVPN server we have just configured.

One of the client types we saw at the OpenVPN server module, was the VitalPBX client type. This can be uploaded on other VitalPBX installations with internet access to connect to this VPN. To use this, on the remote VitalPBX server, we must go to **Admin > Network > OpenVPN Client**. Here, you can upload the **VPN Configuration**, which is the certificate we created at the OpenVPN Server. Once selected, click the green **Upload Configuration** button in the lower right-hand corner.



VPN Configuration

Connection Info

Service Status: Running
Connection Status: Connected
Server IP: 10.8.2.1
Assigned IP: 10.8.2.249
Network Mask: 255.255.255.0

Connection Log

```
-- Journal begins at Thu 2023-11-16 11:26:35 CST, ends at Wed 2023-11-22 14:44:35 CST. --
Nov 22 09:01:48 vitalpbx.local ovpn-vpbx[500]: VERIFY EKU OK
Nov 22 09:01:48 vitalpbx.local ovpn-vpbx[500]: VERIFY OK: depth=0, CN=server
Nov 22 09:01:48 vitalpbx.local ovpn-vpbx[500]: Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
Nov 22 09:01:48 vitalpbx.local ovpn-vpbx[500]: Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
Nov 22 09:01:48 vitalpbx.local ovpn-vpbx[500]: Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, 2048 bit RSA
Nov 22 09:59:27 vitalpbx.local ovpn-vpbx[500]: VERIFY OK: depth=1, CN=CA
Nov 22 09:59:27 vitalpbx.local ovpn-vpbx[500]: VERIFY KU OK
```

Image 14.3.4 - OpenVPN client module connected.

Once uploaded, the remote VitalPBX installation will connect to the OpenVPN server and have an IP address assigned. This is very useful when you have a VitalPBX server in a remote location with no public IP address available. You will now have a secure encrypted tunnel between the remote VitalPBX server and the one with the OpenVPN server. Through this tunnel, you can create trunks to connect both systems in a secure manner.

Section 14, Lesson 4 - Geo-Firewall

When you set up your VitalPBX installation publicly available through the internet, we have seen various ways to deter bot scanners from trying to attack your VitalPBX. Additionally to this, we can apply the **Geo-Firewall** commercial add-on. This is a simple add-on you can install that blocks incoming connection attempts from entire countries. As this is a commercial add-on, you need to purchase individual licensing or any of our licensing plans. Once the add-on is installed, refresh your browser and go to **Admin > Firewall > Geo-Firewall**.

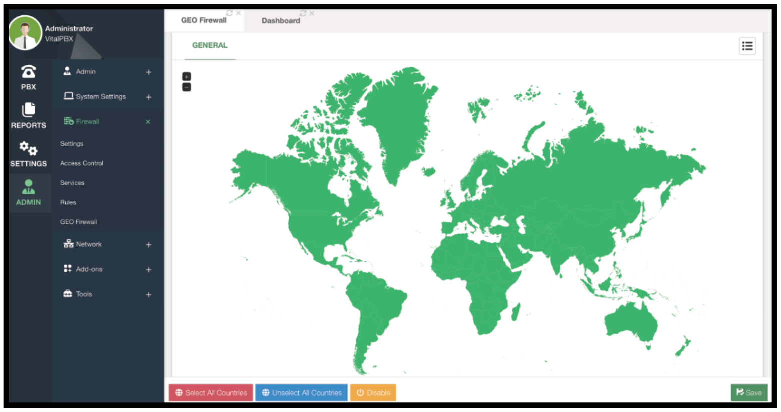


Image 14.4.1 - Geo-Firewall add-on module.

By default, all countries are unblocked. You can block a country by clicking it on the map, or by going to the list of countries in the upper right-hand corner. When you click the country name, it will turn from green to red on the map.

You can also **Select or Unselect all countries** from the buttons in the lower left-hand corner. The Geo-Firewall can be **enabled or disabled** as a whole from here as well.

Once you have the countries you want blocked, click **Save** and then **Apply Changes**. With this, you have now set up the Geo-Firewall.

Conclusion

As you can see, we take the security of your system very seriously. Using the recommendations provided, you can rest assured your communications remain private and safe. Your communications will remain encrypted, and you will deter any possibility of an attack on your VitalPBX installation.

SECTION 15 - ADD-ONS

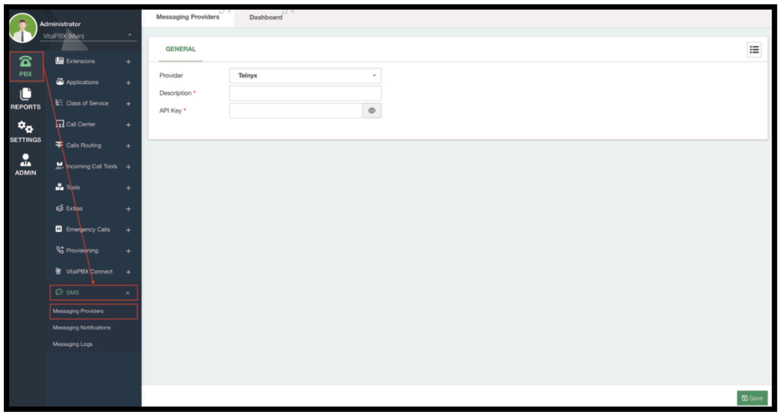
Introduction

Throughout this section, we will cover the rest of the add-ons we have yet to see. This will cover add-on modules like Custom Contexts for higher control over the dial plan, Maintenance for your storage, Branding to make VitalPBX yours, SMS, CRM, and Virtual Faxes.

Section 15, Lesson 1 - SMS

With VitalPBX 4 we have introduced the SMS add-on module. This module allows you to connect with homologated vendors to be able to send and receive SMS messages. To start using the module you need to install it under the add-ons module at **Admin > Add-Ons > Add-Ons**. The SMS add-on module is a commercial module, meaning you need a license to use it. The SMS module is only available through any of our licensing plans.

Once the add-on is installed, refresh your browser and you will find a new section in the PBX navigation menu.

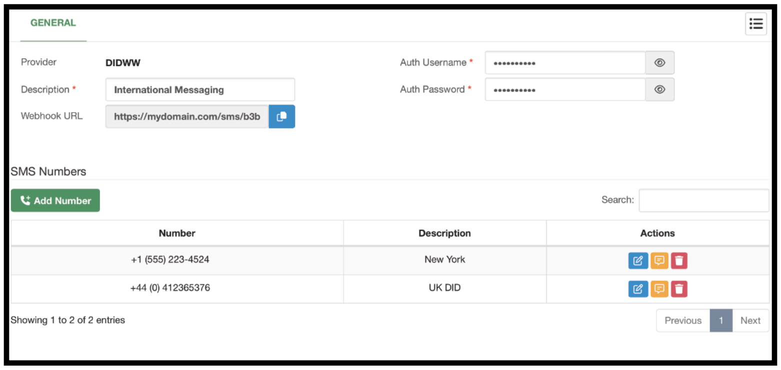


Provider: Telnyx
Description *
API Key *

Image 15.1.1 - SMS module Messaging Providers

For this module, we have homologated multiple VoIP vendors that provide an SMS service. The first step to using SMS with VitalPBX is to create a connection with one of the homologated vendors with your VitalPBX installation. The number of vendors is constantly growing and we can't cover all of them in this guide. For detailed articles covering each vendor, you can head over to our blog at our website, https://vitalpbx.com/?s=SMS.

Depending on the vendor, the fields you will find under **PBX > SMS > Messaging Providers** will vary.


GENERAL

Provider DIDWW
Description * International Messaging
Webhook URL https://mydomain.com/sms/b3b
Auth Username * ••••••••••
Auth Password * ••••••••••

SMS Numbers

Search:

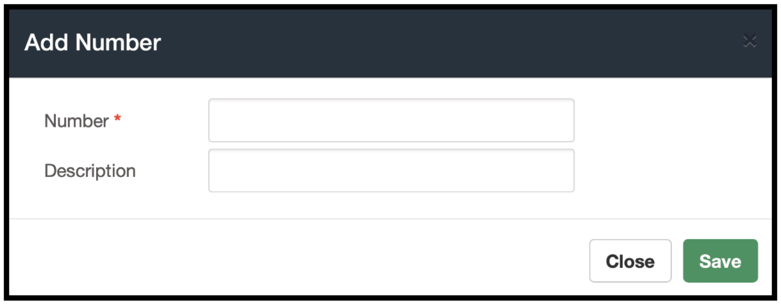
Number	Description	Actions
+1 (555) 223-4524	New York	
+44 (0) 412365376	UK DID	

Showing 1 to 2 of 2 entries

Image 15.1.2 - SMS Messaging Provider set.

When you establish a connection, a **Web-hook URL** is generated. This can be used with the vendor to receive messages and get information on the status of sent messages.

Under **SMS Numbers**, you will see the numbers you have acquired with the vendor to send and receive SMS messages. You can add the numbers manually by clicking **Add Number**.

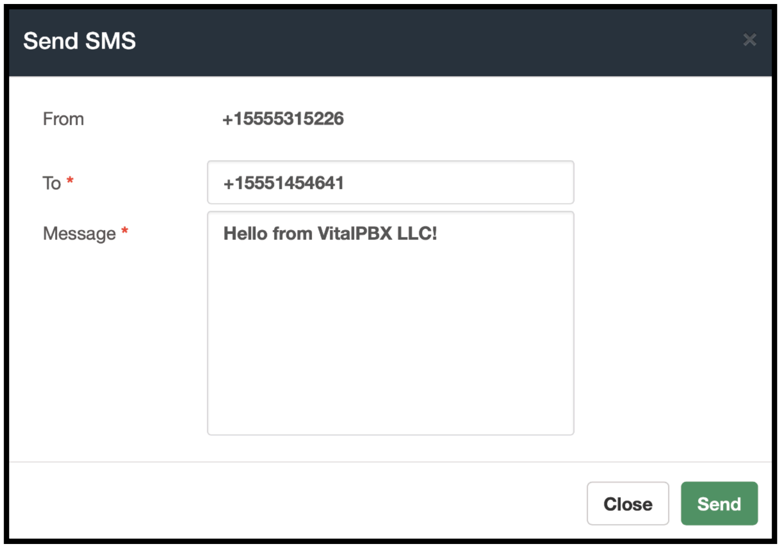

Add Number

Number *
Description

Image 15.1.3 - Adding an SMS number modal.

To add a number enter the **number** with the appropriate format for your vendor, and a **description** to identify this number. Some vendors allow you to retrieve the numbers through the API, so you can use the **Get Numbers from API** button.

When sending an SMS message, you will need to enter the recipient in the **To** field, and the message in the **Message** field. The format of the number is validated, so you need to enter the recipient with the correct format supported by the vendor. Usually, you need to use E.164 format.



From +15555315226

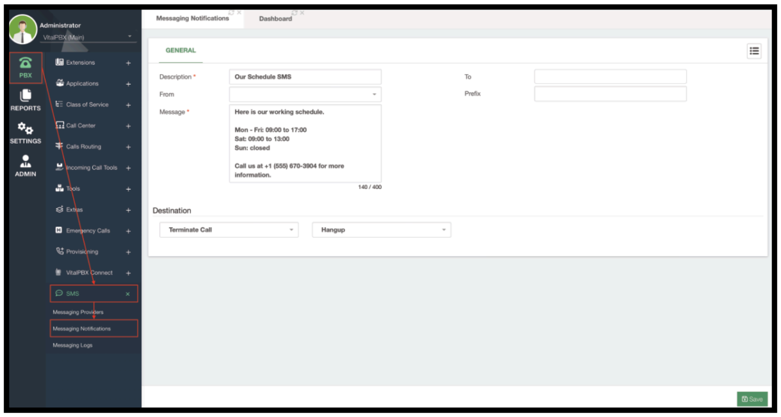
To * +15551454641

Message * Hello from VitalPBX LLC!

Image 15.1.4 - Sending a test SMS from the Messaging Providers module.

When you have the numbers available, you can **edit or delete** them from the **Actions** column. You also have the ability to **send** an SMS message from a number on the list. This allows you to test that you can send SMS messages from VitalPBX.

Once you have your connection with your preferred vendor, you can now use SMS messaging with VitalPBX in multiple ways. The first one is the Messaging Notifications feature. Go to **PBX > SMS > Messaging Notifications**.



Description * Our Schedule SMS

From

To

Prefix

Message * Here is our working schedule.

Mon - Fri: 09:00 to 17:00
Sat: 09:00 to 13:00
Sun: closed

Call us at +1 (555) 670-3904 for more information.

140 / 400

Destination

Terminate Call | Hangup

Image 15.1.4 - SMS Messaging Notifications module.

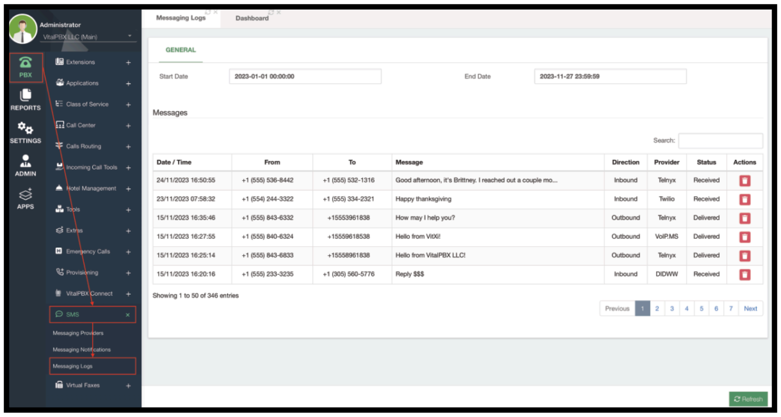
This module allows you to create an automatic SMS message as a destination in your VitalPBX. Whenever a caller is calling from a mobile device or from a phone number that supports SMS, an SMS message is automatically sent.

To configure a Messaging Notification, you need to enter a **Description** to identify the notification. You can then select the **phone number from** which the SMS is sent.

Then, set the **Message** you wish to send. You can then establish a **Final Destination** to send the caller to after the SMS message is sent.

Click **Save** and then **Apply Changes**.

Optionally, you can define a static recipient for the SMS message in the **To** field. This way, when a caller reaches this notification, the SMS message is sent to this phone number. This is useful if you want to send an SMS message to a particular user or service as part of your incoming call chain. You can add a **Prefix** if this is needed as part of your Dial Plan.



Start Date: 2023-01-01 00:00:00
End Date: 2023-11-27 23:59:59

Messages

Search:

Date / Time	From	To	Message	Direction	Provider	Status	Actions
24/11/2023 16:50:55	+1 (555) 536-8442	+1 (555) 532-1316	Good afternoon, it's Brittney. I reached out a couple mo...	Inbound	Telnyx	Received	🗑
23/11/2023 07:58:32	+1 (554) 244-3322	+1 (555) 334-2321	Happy thanksgiving	Inbound	Twilio	Received	🗑
15/11/2023 16:35:46	+1 (555) 843-6332	+15553961838	How may I help you?	Outbound	Telnyx	Delivered	🗑
15/11/2023 16:27:55	+1 (555) 840-6324	+15559618538	Hello from VitXi!	Outbound	VoIP.MS	Delivered	🗑
15/11/2023 16:25:14	+1 (555) 843-6833	+15558961838	Hello from VitalPBX LLC!	Outbound	Telnyx	Delivered	🗑
15/11/2023 16:20:16	+1 (555) 233-3235	+1 (305) 560-5776	Reply $$$	Inbound	DIDWW	Received	🗑

Showing 1 to 50 of 346 entries

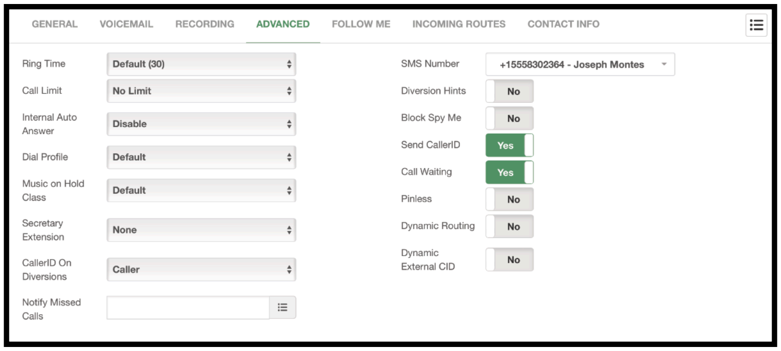
Previous 1 2 3 4 5 6 7 Next

Image 15.1.5 - SMS Messaging Logs.

Next, you can see your **Messaging Logs**. This is like a CDR for SMS messages. Here, you can see the history of all the SMS messages sent and received from your VitalPBX. This allows you to keep track of all messages sent and received.

You can set a date range to filter for specific messages, search by number or message content, and delete message entries.

With the SMS module configured, you can now assign the numbers directly to any extension. Go to **PBX > Extensions > Extensions**. Here, select any extension and go to the **Advanced Tab**.



Ring Time: Default (30)
Call Limit: No Limit
Internal Auto Answer: Disable
Dial Profile: Default
Music on Hold Class: Default
Secretary Extension: None
CallerID On Diversions: Caller
Notify Missed Calls:

SMS Number: +15558302364 - Joseph Montes
Diversion Hints: No
Block Spy Me: No
Send CallerID: Yes
Call Waiting: Yes
Pinless: No
Dynamic Routing: No
Dynamic External CID: No

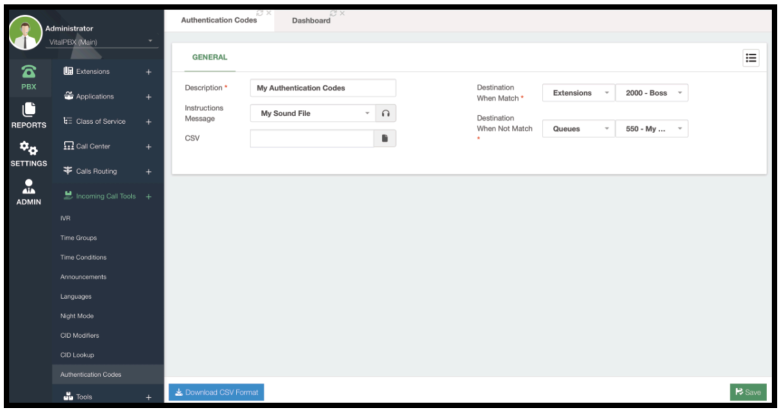
Image 15.1.6 - Setting an SMS number to an extension.

Here, you can set an SMS Number assigned to this extension. This way, any device with this extension registered that supports PJSIP messaging can send and receive messages as a PJSIP message. Whenever you send a PJSIP message to an E.164 number this will be sent as an SMS message. When an SMS message is sent to the number assigned to the extension, it will be converted to a PJSIP message, and forwarded to the registered devices for the extension.

With this set, you can now use SMS with VitalPBX.

Section 15, Lesson 2 - Authentication Codes

Now, let’s look into the **Authentication Codes** add-on module. This free add-on allows you to route an incoming call based on a valid code entered by the calling party. This is perfect if you need to route calls based on customer IDs, reference numbers, and more. To use the authentication codes add-on, you need to install it through the add-ons module at **Admin > Add-Ons > Add-Ons**. Once the add-on is installed, you can refresh your browser and go to **PBX > Incoming Call Tools > Authentication Codes**.



Description * My Authentication Codes
Instructions Message My Sound File
CSV
Destination When Match * Extensions 2000 - Boss
Destination When Not Match * Queues 550 - My ...

Image 15.2.1 - Authentication codes module.

To add authentication codes, you will first need to add a **Description** to identify the codes. Afterward, you can select an **Instruction Message** you have uploaded to **Settings > Voice Prompts > Recordings Management**.

Afterward, you must upload a **CSV** file containing the codes to consider. The format can be downloaded by clicking the **Download CSV Format** button in the lower left-hand corner. You can edit this CSV file in any number processing application.

Once you have the CSV file with the codes selected, you can define the **Destination When the codes Match and when they Don’t**. This way, whenever the calling party enters a code that exists in your codes list, they will be taken to a specific destination.

Finally, you can click on **Save** and then **Apply Changes**.

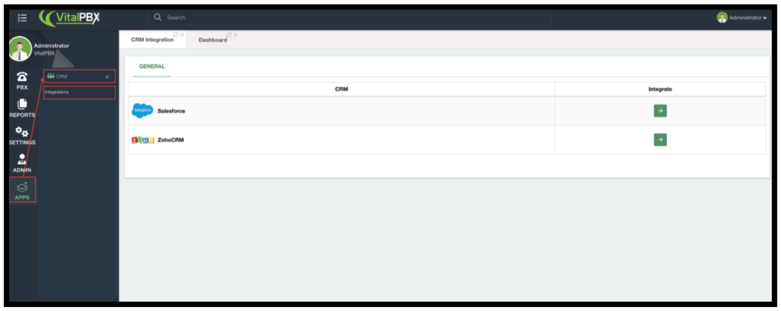
The Authentication Codes can be set as a destination at any part of your call flow pipeline. This way, you have a more dynamic way to route your incoming calls.

Section 15, Lesson 3 - CRM Connector

With VitalPBX 4 we have introduced the **CRM Connector** add-on. This allows you to integrate VitalPBX with a CRM application. As there is a growing list of CRM applications that we are integrating with this add-on, you can see specific guides for the CRM applications through this link, https://vitalpbx.com/?s=CRM.

The CRM Connector add-on is a commercial add-on module available with any of our licensing plans. You will need to have a Licensing Plan subscription to be able to use this commercial add-on.

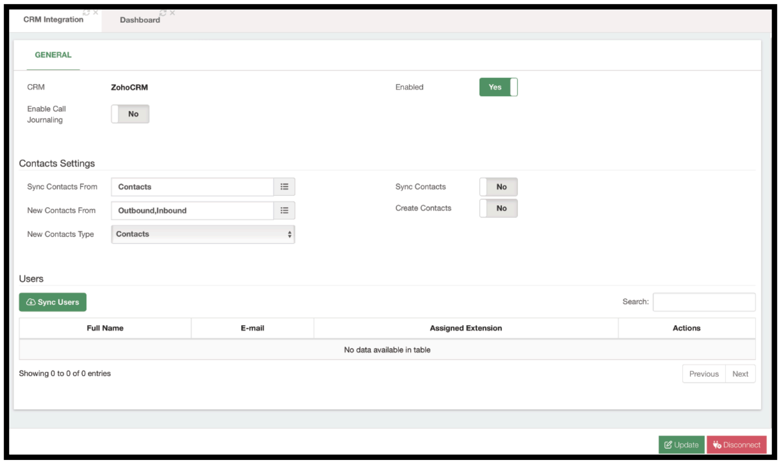
To start using the CRM Connector add-on, you will need to install it through the add-ons module at **Admin > Add-Ons > Add-Ons**. Once the add-on is installed, refresh your browser. You will now see a new section in the navigation menu to the left called Apps. **Go to Apps > CRM > Integrations**.



CRM	Integrate
Salesforce	➔
ZohoCRM	➔

Image 15.3.1 - CRM Integrations module.

At the moment of writing, only Zoho and Salesforce CRM integration is available. Once again, you can go to the following link to find articles with more recent CRM integration guides, https://vitalpbx.com/?s=CRM. Depending on the CRM you choose to use, the connection instructions may vary, follow the instructions from the guides for your specific CRM to continue.



CRM: ZohoCRM

Enabled: Yes

Enable Call Journaling: No

Contacts Settings

Sync Contacts From: Contacts

New Contacts From: Outbound,Inbound

New Contacts Type: Contacts

Sync Contacts: No

Create Contacts: No

Users

Full Name	E-mail	Assigned Extension	Actions
No data available in table			

Showing 0 to 0 of 0 entries

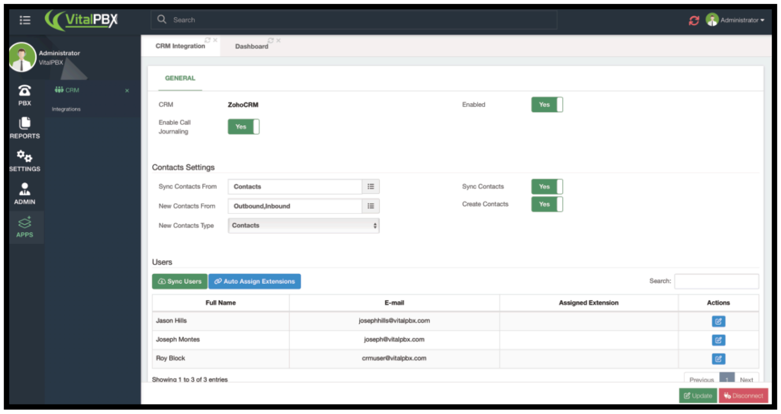
Image 15.3.2 - CRM Connector module with an authenticated connection.

With the CRM Connector now connected you can now **Enable or Disable** the connection. You will also have the ability to ***Enable Call Journaling***. With this, calls will start to get logged in your CRM.

Under the **Contacts Settings**, we define how the calls get logged in your CRM. You can choose whether to **Sync your Contacts** from your CRM **Contacts, Leads, or Accounts**. And whether or not you wish to enable this feature. This will create a **Phonebook** with our Phonebooks add-on module that will contain your contacts. This can later be used with our CID Lookup feature to apply the CID Name to incoming calls and be able to identify the callers.

Next, you can select whether you create **New Contacts From Inbound or Outbound** calls. This feature can also be **Enabled** or **Disabled**. With this, whenever your synced users place or receive a call, a new contact will be created. Finally, you can select whether this new contact is created as a **Contact, Lead, or Account**. In this case, we will leave the default options on, so new calls are created as contacts.

Lastly, you will have the **Users** section. Here you will be able to synchronize the different users that you have created in your CRM. To synchronize them, click on the green **Sync Users** button.



CRM: ZohoCRM
Enabled: Yes
Enable Call Journaling: Yes

Contacts Settings

Sync Contacts From: Contacts
Sync Contacts: Yes
New Contacts From: Outbound,Inbound
Create Contacts: Yes
New Contacts Type: Contacts

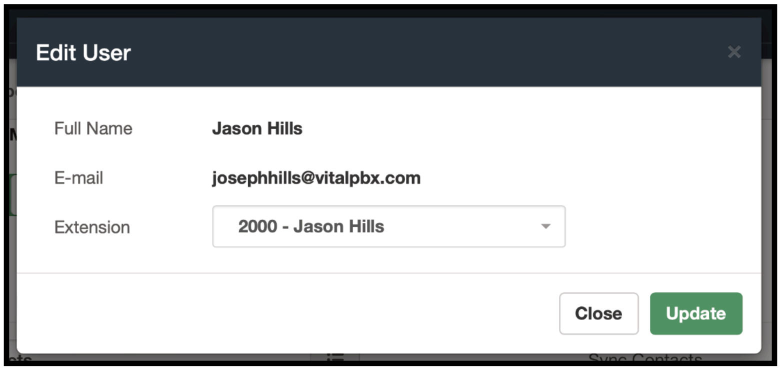
Users

Full Name	E-mail	Assigned Extension	Actions
Jason Hills	josephhills@vitalpbx.com		
Joseph Montes	joseph@vitalpbx.com		
Roy Block	crmuser@vitalpbx.com		

Showing 1 to 3 of 3 entries

Image 15.3.3 - Synchronized users in the CRM Connector module.

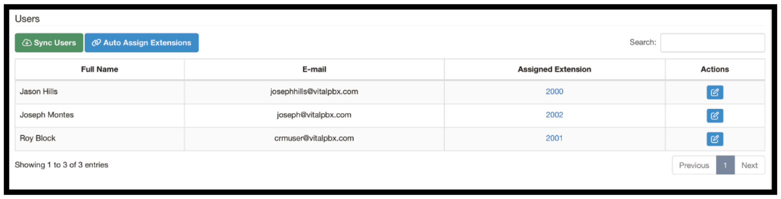
With the users synchronized, you will now have a button to **Auto Assign Extensions**. For this option to work, you must make sure that the extensions have the same email addresses assigned to your CRM users. Click on this button to assign the extensions. Otherwise, you can always edit the user by clicking the blue **Edit** button next to them.


Edit User

Full Name: Jason Hills
E-mail: josephhills@vitalpbx.com
Extension: 2000 - Jason Hills

Image 15.3.4 - Editing User in the CRM Connector Module.

Here you can assign the **Extension** number that belongs to the user. Afterward, you can click on **Update**.

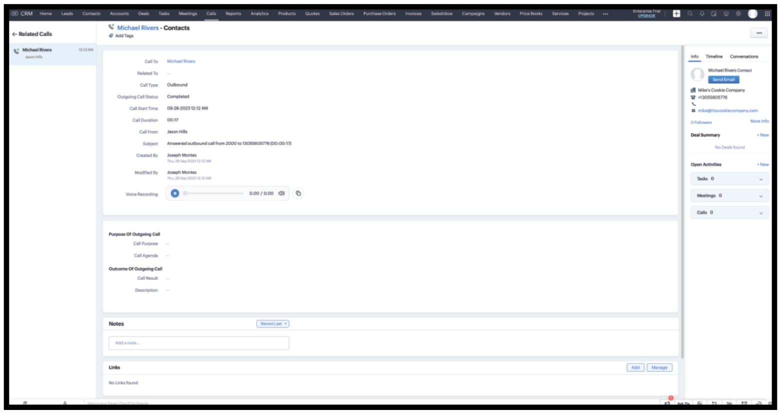


Full Name	E-mail	Assigned Extension	Actions
Jason Hills	josephhills@vitalpbx.com	2000	
Joseph Montes	joseph@vitalpbx.com	2002	
Roy Block	crmuser@vitalpbx.com	2001	

Showing 1 to 3 of 3 entries

Image 15.3.5 - Users with their extensions assigned.

With your users and their extensions assigned, you can start placing or receiving calls. These will now appear in your CRM. For this article, since our contacts are created and synced as Contacts, you will find the activities inside the contacts.



Call To: Michael Rivers
Related To: —
Call Type: Outbound
Outgoing Call Status: Completed
Call Start Time: 09-28-2023 12:12 AM
Call Duration: 00:17
Call From: Jason Hills
Subject: Answered outbound call from 2000 to 13055605776 (00:00:17)
Created By: Joseph Montes, Thu, 28 Sep 2023 12:12 AM
Modified By: Joseph Montes, Thu, 28 Sep 2023 12:12 AM
Voice Recording: 0.00 / 0.00

Purpose Of Outgoing Call
Call Purpose: —
Call Agenda: —

Outcome Of Outgoing Call
Call Result: —
Description: —

Michael Rivers Contact
Mike's Cookie Company
+13055605776
mike@hiscookiecompany.com
0 Followers

Deal Summary
No Deals found

Open Activities
Tasks 0
Meetings 0
Calls 0

No Links found

Image 15.3.6 - Call Activity in the CRM contact.

Here, you can see information on the call itself. When it was created, its duration, the user that is involved in this call, as well as the **Call Recording**. It is important to note that the call recording option is only available with ZOHO CRM at the moment.

For this feature to work, your VitalPBX installation needs to be accessible via the internet with either a Public IP Address or ideally an FQDN. This will allow ZOHO CRM with your permission to point toward the URL of the call recording.

And with this, you now have a full connection between your VitalPBX installation and ZOHO CRM. The process was straightforward, and you can now log your calls with your Contacts, Leads, or Accounts.

This makes the tracking of the activity with your customers easier and more manageable. Resulting in a better management experience and customer relationships.

Section 15, Lesson 4 - Microsoft© Teams™ Connector

One of the most popular tools in Enterprise businesses nowadays is **Microsoft© Teams™**. Since this is a popular collaborative tool, we have created an add-on that integrates your Teams users with VitalPBX. This brings multiple benefits, including the ability to place outbound calls from Teams to a VoIP provider, have Teams users call regular Extension devices and more.

The **MS Teams Connector add-on** is a commercial add-on that is licensed separately from any VitalPBX Licensing Plan. You will need to purchase a separate license to be able to declare more channels with MS Teams extensions. Without any licensing, you can declare 2 MS Teams extensions that can be called through VitalPBX.

Here are some recommendations before we start configuring the connection with **Microsoft© Teams™**.

- In this manual, we are going to create a Trial account to have access to MS Teams with Voice plan.
- We use the latest version available of VitalPBX.
- We must have a valid domain (FQDN) that we have access to configure at a DNS level.
- We must work from **Windows™** since we are going to use **PowerShell™** for some configurations and module installations. Although there are also versions of **PowerShell™** for **MacOS™**.

When using your own SIP Trunk (BYOT) in combination with a VitalPBX to set up **Direct Routing** to **Microsoft© Teams™** you will need to make some firewall rules.

> ***Note:*** *This applies to Microsoft 365, Office 365, and Office 365 GCC tenants.*

When configuring **Direct Routing** you will need to setup 3 FQDNs for signaling.

- **sip.pstnhub.microsoft.com**
- **sip2.pstnhub.microsoft.com**
- **sip3.pstnhub.microsoft.com**

The first one will be located by location and performance metrics. If your location is EMEA, your 2nd location will be the US and the 3rd ASIA.

Primary DC	sip.pstnhub.microsoft.com	**EMEA**	**NOAM**	**ASIA**
Secondary DC	sip2.pstnhub.microsoft.com	US	EU	US
Tertiary DC	sip3.pstnhub.microsoft.com	ASIA	ASIA	EU

These FQDNs resolve to the following IP Address ranges.

- **52.112.0.0/14**
- **52.120.0.0/14**

In the list below you will find the extracted IP Address Ranges.

- **52.112.0.0/16**
- **52.113.0.0/16**
- **52.114.0.0/16**
- **52.115.0.0/16**
- **52.120.0.0/16**
- **52.121.0.0/16**
- **52.122.0.0/16**
- **52.123.0.0/16**

When we talk about media ports (not in media Bypass mode 2) the following ranges are used. Media is using a separate service in the Microsoft© cloud, these are called Media Processors.

Locations where both SIP proxy and media processor components are deployed.

- **US (two in US West and US East data centers)**
- **Europe (Amsterdam and Dublin data centers)**
- **Asia (Singapore and Hong Kong data centers)**

Locations where only media processors are deployed (SIP flows via the closest data center listed above).

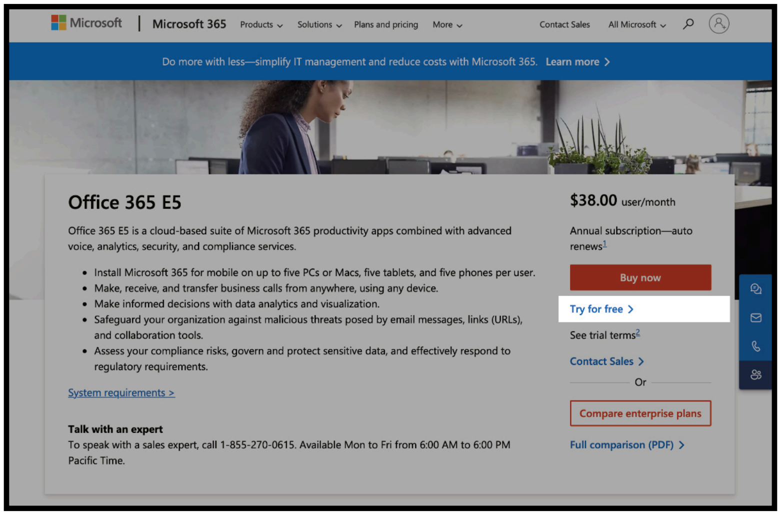
- **Japan (JP East and West data centers)**
- **Australia (AU East and Southeast data centers)**

The following ranges are being used.

- **52.112.0.0/14**
- **52.122.0.0/15**

This information will help you configure your Firewall to prevent having these IP Addresses blocked.

With all these recommendations, we can proceed to configure VitalPBX with MS Teams.



Do more with less—simplify IT management and reduce costs with Microsoft 365. Learn more >

Office 365 E5

Office 365 E5 is a cloud-based suite of Microsoft 365 productivity apps combined with advanced voice, analytics, security, and compliance services.

- Install Microsoft 365 for mobile on up to five PCs or Macs, five tablets, and five phones per user.
- Make, receive, and transfer business calls from anywhere, using any device.
- Make informed decisions with data analytics and visualization.
- Safeguard your organization against malicious threats posed by email messages, links (URLs), and collaboration tools.
- Assess your compliance risks, govern and protect sensitive data, and effectively respond to regulatory requirements.

Talk with an expert
To speak with a sales expert, call 1-855-270-0615. Available Mon to Fri from 6:00 AM to 6:00 PM Pacific Time.

$38.00 user/month

Annual subscription—auto renews[1]

See trial terms[2]

Image 15.4.1 - Setting up an Office 365™ trial account.

First, we go to the Microsoft© page and create an **Office 365 E5 Trial** account which already comes with MS Teams and the Phone System plan enabled. As Microsoft© constantly changes the links, we recommend going to microsoft.com and searching for the appropriate location for the Office 365™ plans.

> ***Note:*** *If in your country this type of plan is not available yet, and you want to choose the US. You must make sure that you have a cell phone number from the US so you can receive the codes they will send you by SMS.*

It is also possible to use the **Office 365™ E1 Plan** or **Microsoft 365™ Business Basic**, but we would have to purchase the **Phone System plugin** separately.

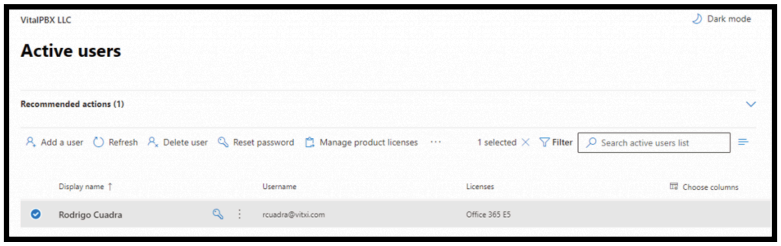
During the account creation, we select the domain that contains **mydomain.onmicrosoft.com**. In this example, we will be using **vitxi.onmicrosoft.com**. Later, we will create a new domain that is the one that will be configured in our VitalPBX Server.

Once the account is created, we must take into consideration the following.

- **Microsoft© 365 admin center -** This is where we create users, domains, groups, etc.
  - https://admin.microsoft.com/
- **Microsoft© Teams™ administration -** This is where we are going to configure the interconnection with VitalPBX.
  - https://admin.teams.microsoft.com/
- **Microsoft© Teams™ -** This is where we will be able to make calls, conferences, chat, etc.
  - https://teams.microsoft.com/

Now, we must go to the Microsoft© 365 admin center. Enter the username and password you used at the time of registration.

Go to **User > Active Users**. Here, you can manage the users you have created or create a new one. Here, we verify that our users have the appropriate licensing.



Display name ↑	Username	Licenses
Rodrigo Cuadra	rcuadra@vitxi.com	Office 365 E5

Image 15.4.2 - Active users in the Microsoft© 365 admin center.

Now, go to **Settings > Domains** on the left-hand side navigation menu. You might have to click on **Show All** to see the settings option. Here, you will see the domain that was created during the initial configuration of your account creation. We can also create a new domain from here.

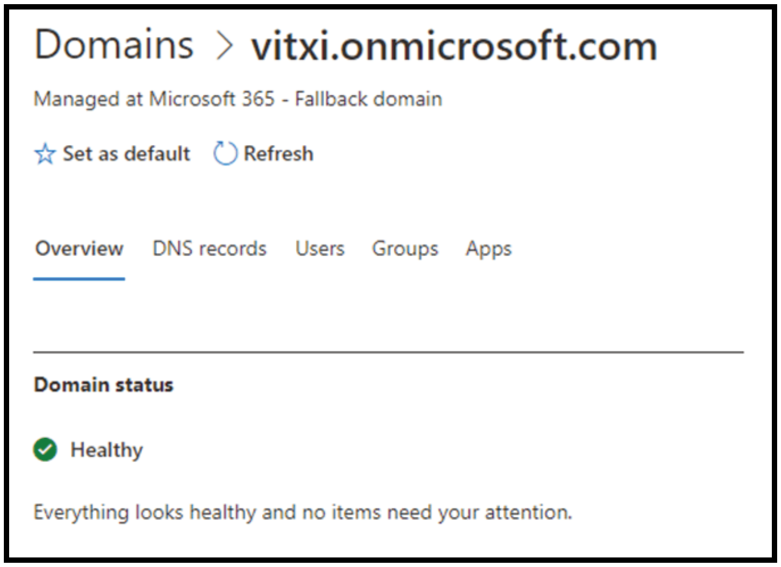
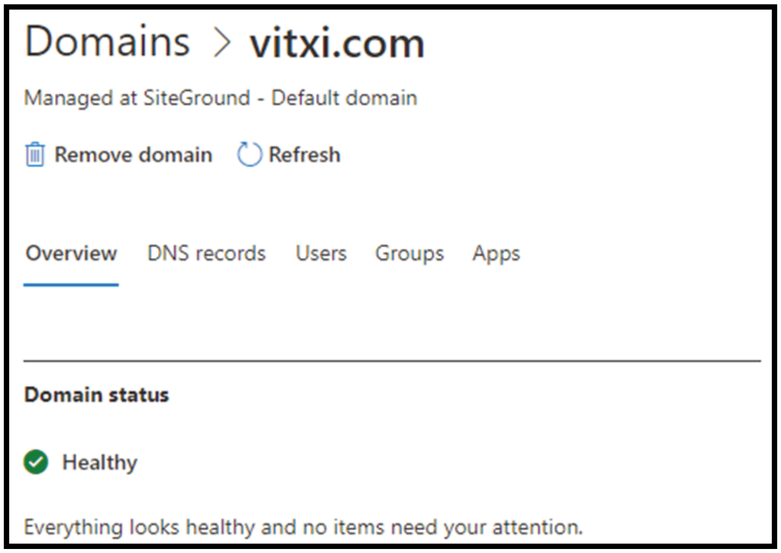

Domains > vitxi.onmicrosoft.com
Managed at Microsoft 365 - Fallback domain
Set as default
Refresh
Overview
DNS records
Users
Groups
Apps
Domain status
Healthy
Everything looks healthy and no items need your attention.

Image 15.4.3 - Domains module in Microsoft© 365 admin center.

Next, we **create a domain** that matches the domain that we are going to connect our trunk to through Direct Routing of MS Teams. For this, click the **+ Add Domain** button. This must be a domain that we have full control of, as we will be asked to make some changes to our **DNS configurations**, to prove our ownership of the domain. Once the domain is created, you will see it as follows.



Domains > vitxi.com

Managed at SiteGround - Default domain

Domain status

Healthy

Everything looks healthy and no items need your attention.

Image 15.4.4 - New domain added to the Microsoft© 365 admin center.

In this example, we are using **vitxi.com**. Take into consideration that our VitalPBX does not necessarily have to be in the domain itself, but it can be in a sub-domain of it, i.e. **msteams.vitxi.com**.

> ***Note:*** *The domain on the user's email must match the domain for the VitalPBX installation we are connecting with.*

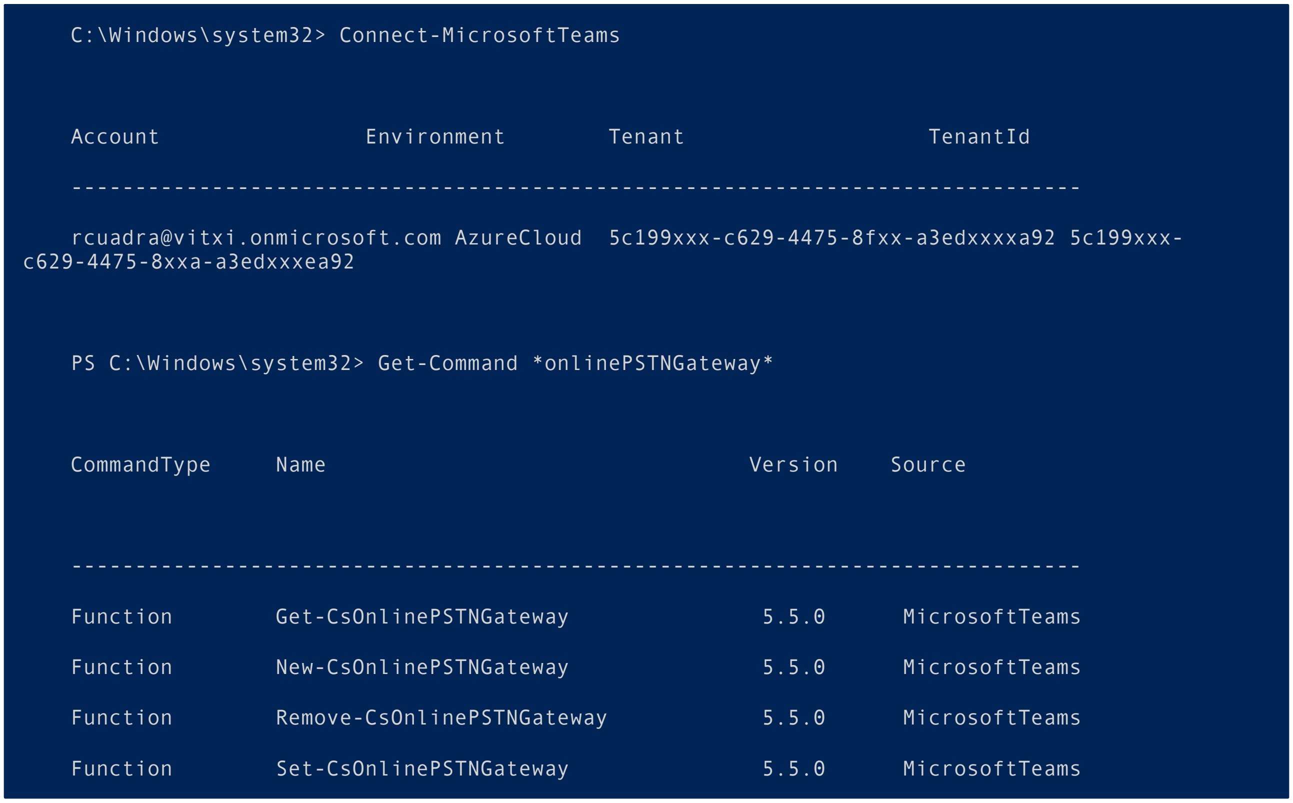
Afterward, we will need to connect to Microsoft© Teams™ using **PowerShell™**. **Run** PowerShell™ as an **Administrator**.

Then, run the following commands from PowerShell™.

```
PS C:\Windows\system32> Install-Module MicrosoftTeams

PS C:\Windows\system32> Import-Module MicrosoftTeams
```

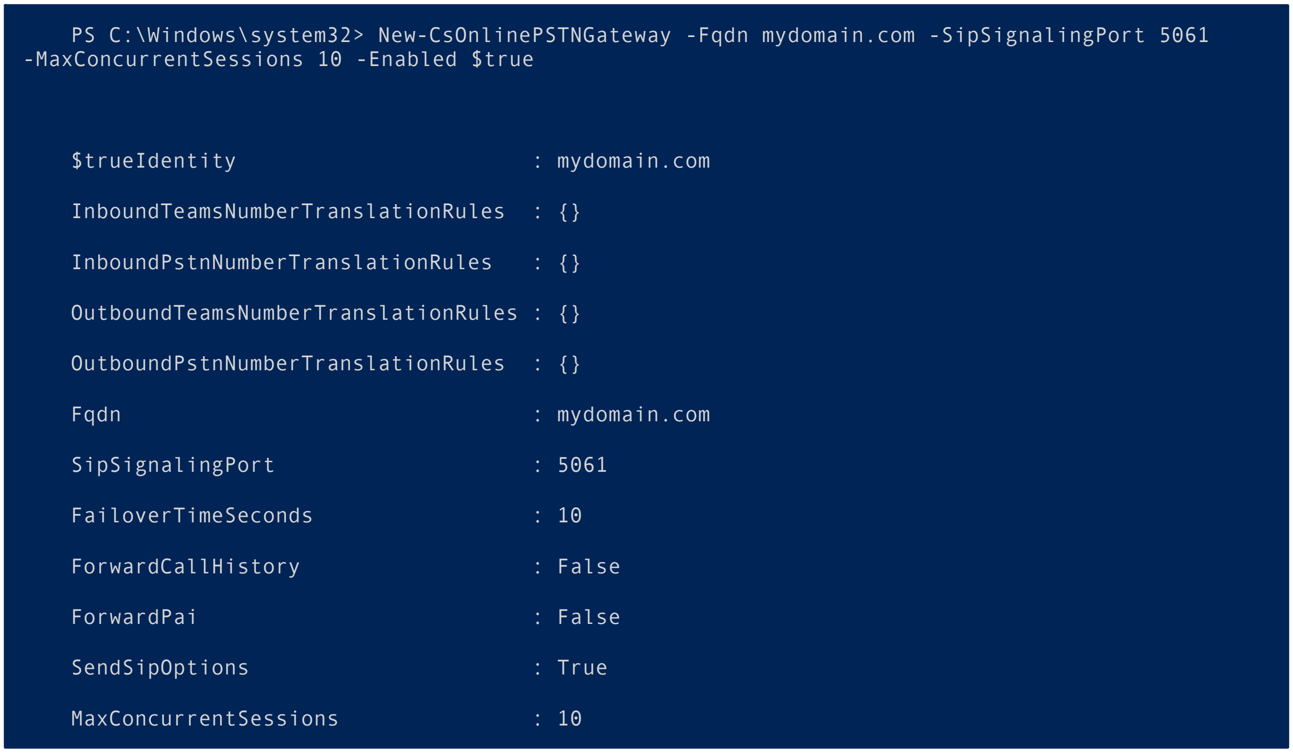
Next, run the following commands which will prompt for the Office 365™ administrator credentials. Enter the user and password you created previously.

```
PS C:\Windows\system32> $credential = Get-Credential


cmdlet Get-Credential at command pipeline position 1

Supply values for the following parameters:

CredentialPS
```



```
    C:\Windows\system32> Connect-MicrosoftTeams


    Account                 Environment       Tenant                    TenantId

    ------------------------------------------------------------------------------

    rcuadra@vitxi.onmicrosoft.com AzureCloud  5c199xxx-c629-4475-8fxx-a3edxxxxa92 5c199xxx-
c629-4475-8xxa-a3edxxxea92


    PS C:\Windows\system32> Get-Command *onlinePSTNGateway*


    CommandType     Name                                 Version    Source



    ------------------------------------------------------------------------------

    Function        Get-CsOnlinePSTNGateway               5.5.0      MicrosoftTeams

    Function        New-CsOnlinePSTNGateway               5.5.0      MicrosoftTeams

    Function        Remove-CsOnlinePSTNGateway            5.5.0      MicrosoftTeams

    Function        Set-CsOnlinePSTNGateway               5.5.0      MicrosoftTeams
```

This shows that the connection was successful and the commands are now available. Now, we will proceed to create the **Direct Routing**. We run the following command on **PowerShell™** to do so. Remember to change the domain to the domain you have created previously.



```
    PS C:\Windows\system32> New-CsOnlinePSTNGateway -Fqdn mydomain.com -SipSignalingPort 5061
-MaxConcurrentSessions 10 -Enabled $true


    $trueIdentity                        : mydomain.com

    InboundTeamsNumberTranslationRules   : {}

    InboundPstnNumberTranslationRules    : {}

    OutboundTeamsNumberTranslationRules : {}

    OutboundPstnNumberTranslationRules   : {}

    Fqdn                                 : mydomain.com

    SipSignalingPort                     : 5061

    FailoverTimeSeconds                  : 10

    ForwardCallHistory                   : False

    ForwardPai                           : False

    SendSipOptions                       : True

    MaxConcurrentSessions                : 10
    Enabled                           : True

    MediaBypass                       : False

    GatewaySiteId                     :

    GatewaySiteLbrEnabled             : False

    GatewayLbrEnabledUserOverride     : False

    FailoverResponseCodes             : 408,503,504

    PidfLoSupported                   : False

    MediaRelayRoutingLocationOverride :

    ProxySbc                          :

    BypassMode                        : None

    Description                       :
```

Afterward, we create the **Incoming Call Routing**. To configure the routing of incoming calls within MS Teams, you must configure in MS Teams which phone number will be associated with it. To do this, execute the following commands on **PowerShell™**. Remember to change the user to the user you created previously.

```
    PS C:\Windows\system32> Set-CsPhoneNumberAssignment -Identity myemail@mydomain.com
-EnterpriseVoiceEnabled $true

    PS C:\Windows\system32> Set-CsPhoneNumberAssignment -Identity myemail@mydomain.com
-PhoneNumber “3000” -PhoneNumberType DirectRouting
```

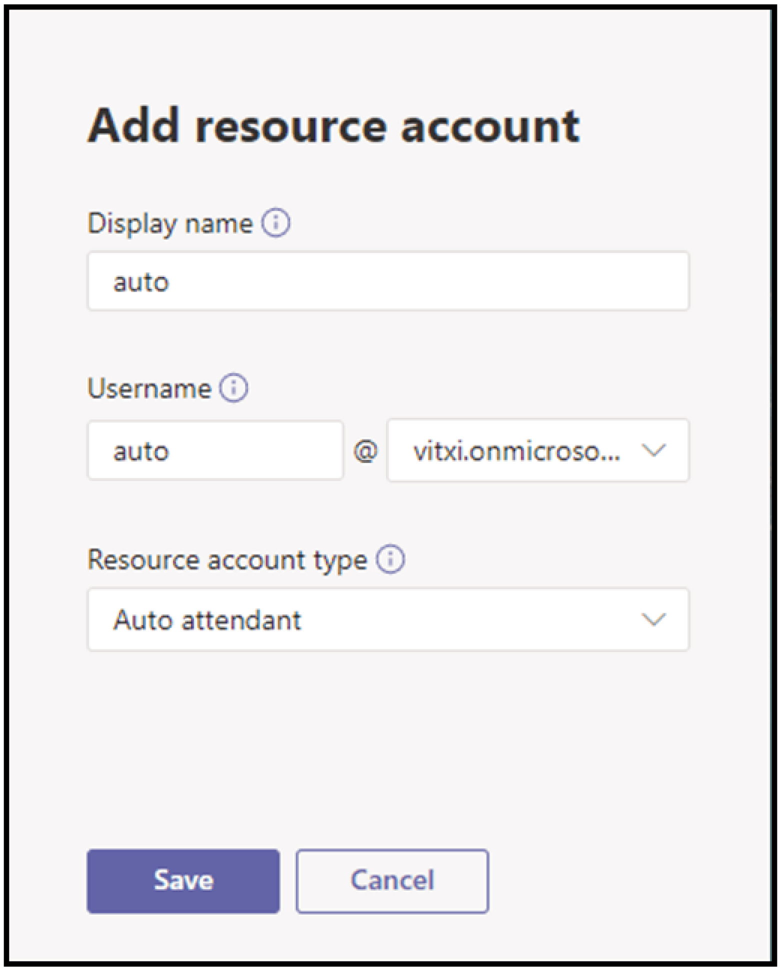
With these commands, we have assigned the number **3000** to our user. This would be similar to assigning an extension number. This must be different from the numbering plan on your VitalPBX installation, so we can have easier routing when configuring our VitalPBX.

Now, we must configure the **Outgoing Call Routing**. For this run the following command on **PowerShell™**. Remember to change the domain to the domain you have added previously.

```
    PS C:\Windows\system32> Set-CsOnlinePstnUsage -Identity Global -Usage @{Add=”Default”}

    PS C:\Windows\system32> New-CsOnlineVoiceRoute -Identity “All” -NumberPattern “.*”
-OnlinePstnGatewayList mydomain.com -Priority 0 -OnlinePstnUsages “Default”

    Identity                : All

    Priority                : 0

    Description             :

    NumberPattern           : .*

    OnlinePstnUsages        : {Default}

    OnlinePstnGatewayList   : {mydomain.com}

    BridgeSourcePhoneNumber :

    Name                    : All
```

Now, we will create an **Auto-Attendant** so we can later test our calling.

First, we will create a **Resource account**. Go to the **Teams Admin Center**, and go to **Voice > Resource Accounts**. We must create this account with our original domain using the **onmicrosoft.com** domain. Click on **+ Add,** then add a **Display Name** to identify the resource, a **Username**, and set the **Resource Account Type** to **Auto Attendant**. Then click on **Save**.



Add resource account

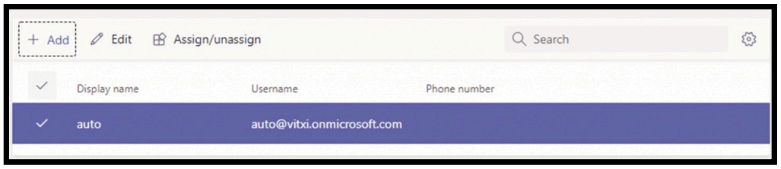
Display name: auto

Username: auto @ vitxi.onmicroso...

Resource account type: Auto attendant

Image 15.4.5 - Create a Resource Account in Microsoft© Teams™.

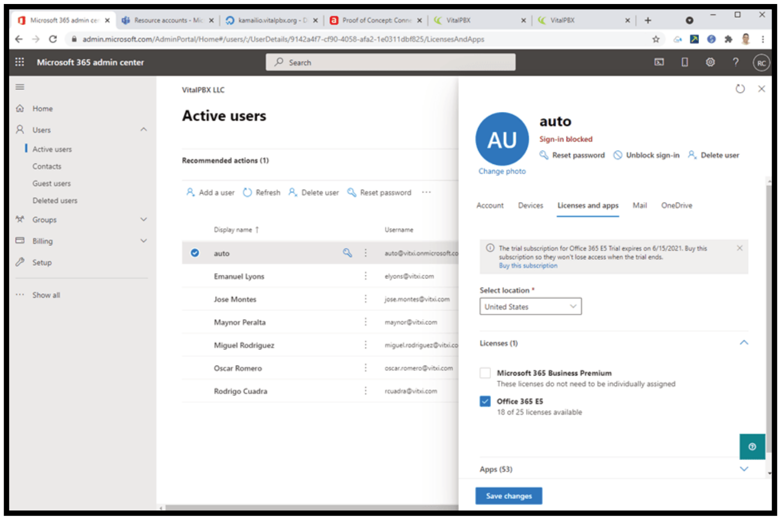
Once the resource is created it will look like this.



✓	Display name	Username	Phone number
✓	auto	auto@vitxi.onmicrosoft.com	

Image 15.4.6 - Resource account created in Microsoft© Teams™ admin center.

Now we go to the Microsoft© 365™ admin center and assign a license that contains Voice. For this, we go to **Users > Active users**, select the resource account user we just created, and edit it to assign the Office 365 E5 license.



admin.microsoft.com/AdminPortal/Home#/users/:/UserDetails/9142a4f7-cf90-4058-afa2-1e0311dbf825/LicensesAndApps

VitalPBX LLC

Active users

Recommended actions (1)

Display name ↑	Username
auto	auto@vitxi.onmicrosoft.co
Emanuel Lyons	elyons@vitxi.com
Jose Montes	jose.montes@vitxi.com
Maynor Peralta	maynor@vitxi.com
Miguel Rodriguez	miguel.rodriguez@vitxi.co
Oscar Romero	oscar.romero@vitxi.com
Rodrigo Cuadra	rcuadra@vitxi.com

auto
Sign-in blocked

The trial subscription for Office 365 E5 Trial expires on 6/15/2021. Buy this subscription so they won't lose access when the trial ends.
Buy this subscription

Select location *
United States

Licenses (1)

☐ Microsoft 365 Business Premium
These licenses do not need to be individually assigned

☑ Office 365 E5
18 of 25 licenses available

Apps (53)

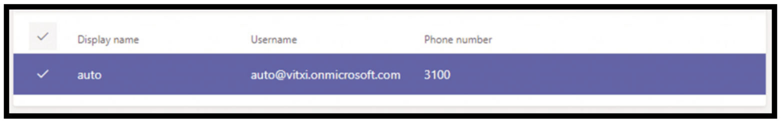
Image 15.4.7 - Assigning an E5 license to the Resource Account in the MS admin center.

Since we have created the new resource with the voice permissions, we must go to PowerShell™ and execute the following command, where +3100 is the number that we are going to dial to access the Auto Attendant.

> ***Note:*** *It may take up to 30 minutes for the new resource to recognize the license, so if you get a license error, wait for a moment.*

```
    PS C:\Windows\system32> Set-CsOnlineApplicationInstance -Identity
“auto@vitxi.onmicrosoft.com” -OnpremPhoneNumber +3100

    RunspaceId        : 738753bc-e7e9-4948-9d56-6a29b633920e

    ObjectId          : 9142a4f7-cf90-4058-afa2-1e0311dbf825

    TenantId          : 5c199f18-c629-4475-8f1a-a3ed1644ea92

    UserPrincipalName : auto@mydomain.com

    ApplicationId     : ce933385-9390-45d1-9512-c8d228074e07

    DisplayName       : auto

    PhoneNumber       :
```

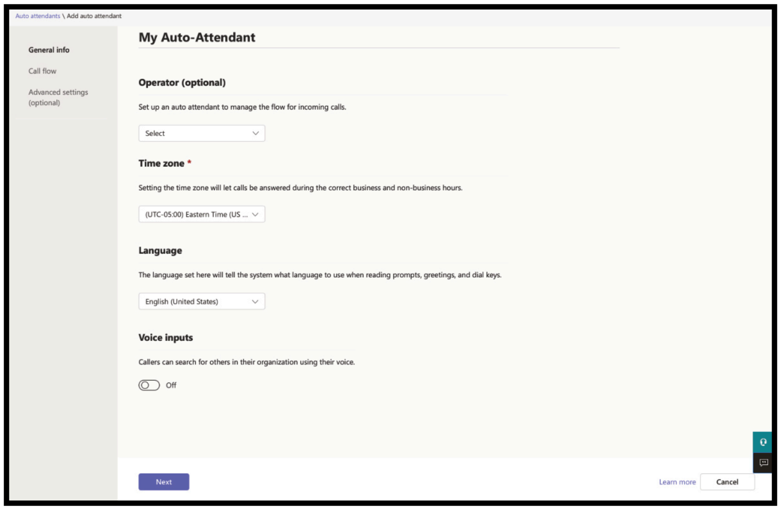
Then, on the MS Teams admin center go to **Voice > Auto Attendants**, and follow the instructions. Verify that the extension was configured by entering the resource, which should look like this.



✓	Display name	Username	Phone number
✓	auto	auto@vitxi.onmicrosoft.com	3100

Image 15.4.7 - Resource account with a Phone Number assigned.

Now we are going to create our Auto Attendant, for which we go to the MS Teams admin center, then to **Voice > Auto Attendants**. Click on the **+ Add** button.



Auto attendants \ Add auto attendant

My Auto-Attendant

Operator (optional)

Set up an auto attendant to manage the flow for incoming calls.

Select

Time zone *

Setting the time zone will let calls be answered during the correct business and non-business hours.

(UTC-05:00) Eastern Time (US ...

Language

The language set here will tell the system what language to use when reading prompts, greetings, and dial keys.

English (United States)

Voice inputs

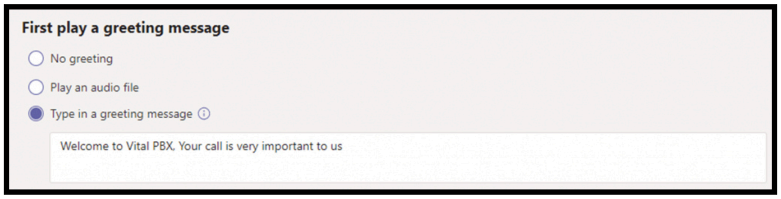
Callers can search for others in their organization using their voice.

Off

Image 15.4.8 - General Info tab creating an Auto-Attendant in MS Teams admin center.

Under the **General Info tab**, give the auto-attendant a name to identify it. Then, leave the **Operator** option as is, set the **time zone** to your time zone, and select the **language** you are going to use. Click on **Next**.

Under the **Call Flow tab**, for the **First Play a Greeting Message** option, you can type in a greeting message if you don't have an audio file.



First play a greeting message

○ No greeting

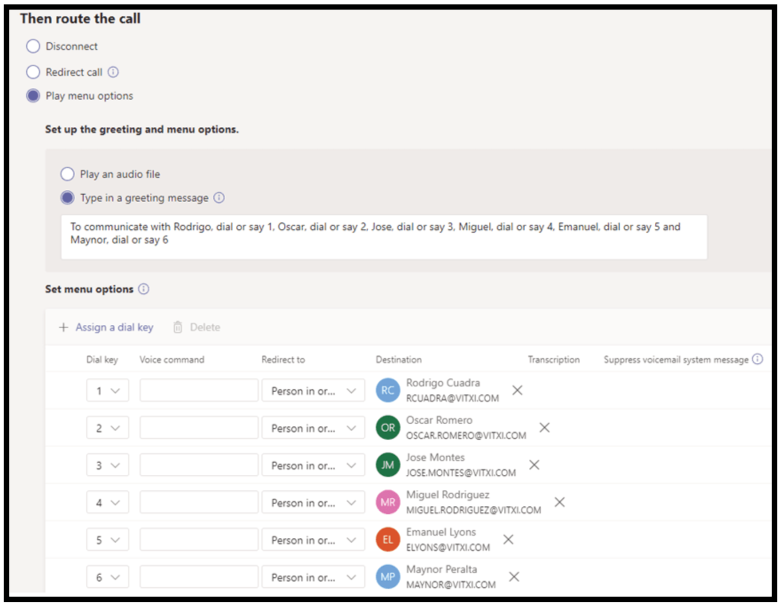
○ Play an audio file

● Type in a greeting message ⓘ

Welcome to Vital PBX. Your call is very important to us

Image 15.4.8 - Play a greeting message in Call Flow for MS Teams Auto-Attendant.

Next, under the **Then Route The Call** option, we will configure the dialing instructions so the caller can reach the desired destination.



Then route the call

○ Disconnect
○ Redirect call
● Play menu options

Set up the greeting and menu options.

○ Play an audio file
● Type in a greeting message

To communicate with Rodrigo, dial or say 1, Oscar, dial or say 2, Jose, dial or say 3, Miguel, dial or say 4, Emanuel, dial or say 5 and Maynor, dial or say 6

Set menu options

Dial key	Voice command	Redirect to	Destination	Transcription	Suppress voicemail system message
1		Person in or...	RC Rodrigo Cuadra RCUADRA@VITXI.COM		
2		Person in or...	OR Oscar Romero OSCAR.ROMERO@VITXI.COM		
3		Person in or...	JM Jose Montes JOSE.MONTES@VITXI.COM		
4		Person in or...	MR Miguel Rodriguez MIGUEL.RODRIGUEZ@VITXI.COM		
5		Person in or...	EL Emanuel Lyons ELYONS@VITXI.COM		
6		Person in or...	MP Maynor Peralta MAYNOR@VITXI.COM		

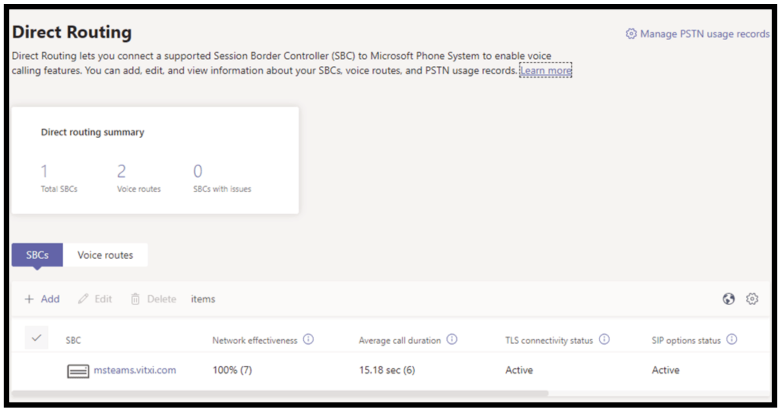
15.4.9 - Auto Attendant menu in MS Teams admin center.

In our example, we have other users with the telephony plugin added to their accounts. If you only have one account, you will only be able to add that account as a menu option.

Afterward, there is the option to add a schedule, but we will leave it as is.

With this, we now have an auto-attendant resource in Microsoft© Teams™ that we can dial from any MS Teams user.

Now, we must verify the configurations in the Microsoft© Teams™ admin center. Go to Voice > Direct Routing, and verify that the trunk towards our VitalPBX has been created. An error might appear at the beginning since we have not configured VitalPBX's side, but after configuring VitalPBX, we will see the following screen.



Direct Routing

Direct Routing lets you connect a supported Session Border Controller (SBC) to Microsoft Phone System to enable voice calling features. You can add, edit, and view information about your SBCs, voice routes, and PSTN usage records. Learn more

Direct routing summary

1 Total SBCs
2 Voice routes
0 SBCs with issues

SBC	Network effectiveness	Average call duration	TLS connectivity status	SIP options status
msteams.vitxi.com	100% (7)	15.18 sec (6)	Active	Active

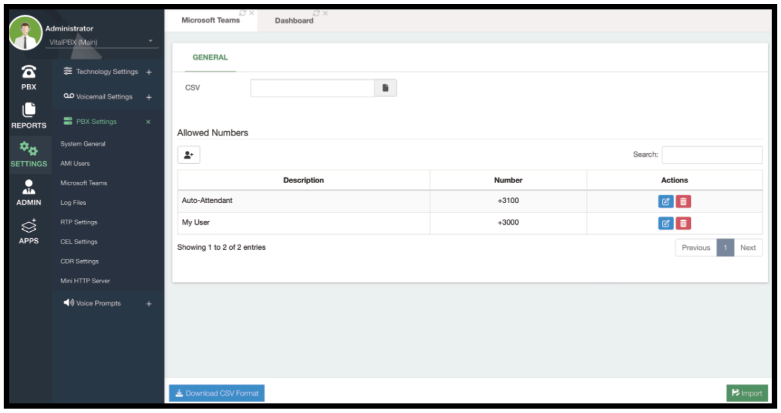
Image 15.4.10 - Direct Routing configuration in MS Teams admin center.

Now, we can go ahead and configure VitalPBX for the MS Teams connection.

Before we proceed, remember that your VitalPBX installation must have a valid FQDN pointing to it using the same domain that you added to the Microsoft© admin center. This FQDN domain must have a valid **SSL certificate**, which can be created under **Admin > System Settings > Certificates** using **Let's Encrypt** or a **Custom Certificate**. This will **not** work with **self-signed certificates.**

The certificate will then need to be assigned to the VitalPBX installation under **Admin > System Settings > HTTP Server** under the **Certificate** field, and **Force HTTPS** must be **enabled**.

Now, install the Microsoft© Teams™ Connector add-on under **Admin > Add-Ons > Add-Ons**. Once the add-on has been installed, refresh your browser and go to **Settings > PBX Settings > Microsoft© Teams™.**



CSV

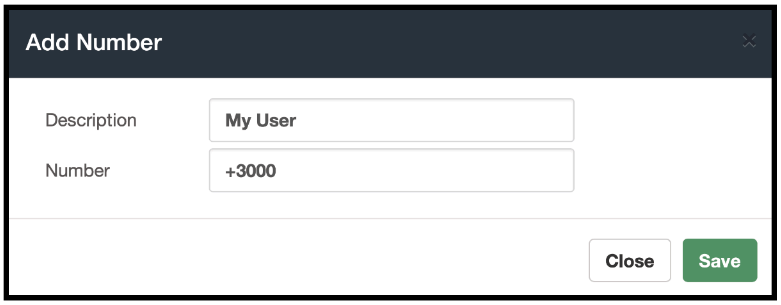
Allowed Numbers

Description	Number	Actions
Auto-Attendant	+3100	
My User	+3000	

Showing 1 to 2 of 2 entries

Image 15.4.11 - Microsoft© Teams™ add-on module in VitalPBX.

Here, you can declare the users and phone numbers we created in Microsoft© Teams™. This way, we can call them and receive their calls on VitalPBX. To add the users, you can click the **Add** button under **Allowed Numbers**.



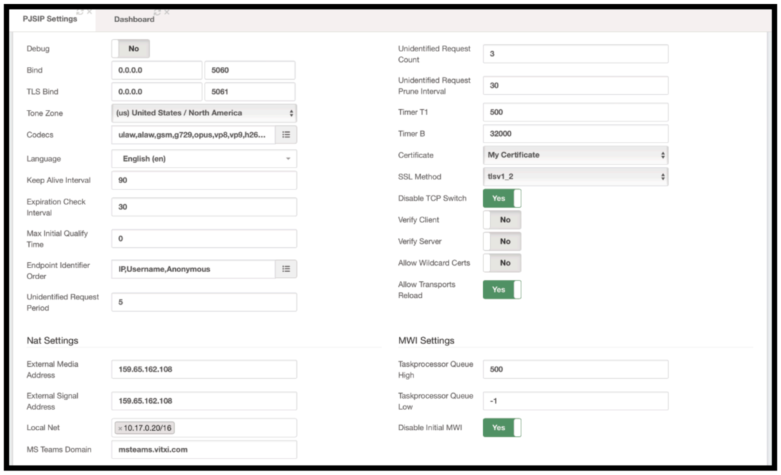
Description: My User

Number: +3000

Image 15.4.12 - Adding a Microsoft© Teams™ user to the MS Teams add-on.

You can enter a **description** to identify the user and the **number** associated with this user. Remember to add the plus (+) sign at the beginning of the number. Then click on **Save**. If you have multiple users, you can import a **CSV** file with all the user data. A template is available under the **Download CSV Format** button in the lower left-hand corner. Once the file is added to the CSV field, you can click on **Import** in the lower right-hand corner. With the users created or imported, you can see them listed under Allowed Numbers, and **edit or delete** them.

The next step will be to configure the PJSIP settings. Go to **Settings > Technology Settings > PJSIP Settings**. You will now see a new field for MS Teams under **NAT Settings**, where you must declare the **MS Teams Domain**.



PJSIP Settings | Dashboard

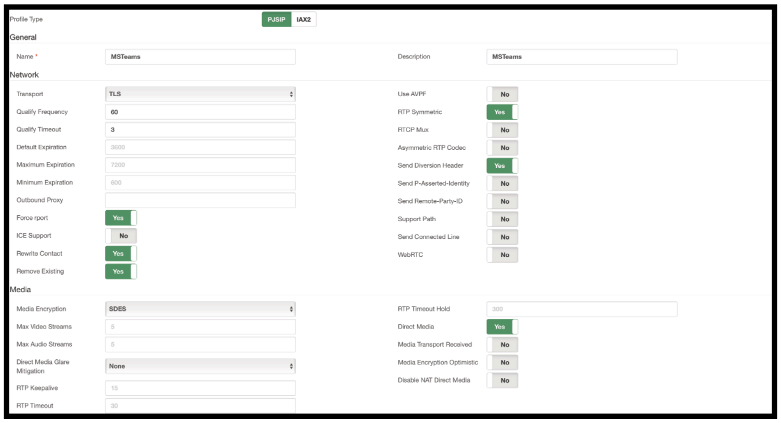
Debug	No	Unidentified Request Count	3
Bind	0.0.0.0 · 5060	Unidentified Request Prune Interval	30
TLS Bind	0.0.0.0 · 5061	Timer T1	500
Tone Zone	(us) United States / North America	Timer B	32000
Codecs	ulaw,alaw,gsm,g729,opus,vp8,vp9,h26...	Certificate	My Certificate
Language	English (en)	SSL Method	tlsv1_2
Keep Alive Interval	90	Disable TCP Switch	Yes
Expiration Check Interval	30	Verify Client	No
Max Initial Qualify Time	0	Verify Server	No
Endpoint Identifier Order	IP,Username,Anonymous	Allow Wildcard Certs	No
Unidentified Request Period	5	Allow Transports Reload	Yes

Nat Settings		MWI Settings	
External Media Address	159.65.162.108	Taskprocessor Queue High	500
External Signal Address	159.65.162.108	Taskprocessor Queue Low	-1
Local Net	× 10.17.0.20/16	Disable Initial MWI	Yes
MS Teams Domain	msteams.vitxi.com		

Image 15.4.13 - PJSIP Settings for MS Teams connection.

Additionally, we must set the **Certificate** we have created previously, set the **SSL Method** to **tlsv1_2**, add the **External Media and Signal Addresses**, and add the **Local Net** (optional, if available). Click on **Save** and then **Apply Changes**.

Now, we must create a new **PJSIP Device Profile**. Go to **Settings > Technology Settings > Device Profiles**.



Profile Type: PJSIP | IAX2

General			
Name *	MSTeams	Description	MSTeams

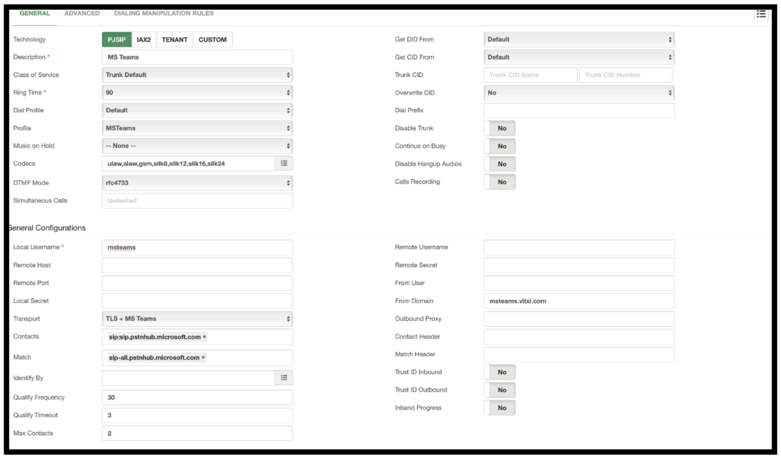
Network			
Transport	TLS	Use AVPF	No
Qualify Frequency	60	RTP Symmetric	Yes
Qualify Timeout	3	RTCP Mux	No
Default Expiration	3600	Asymmetric RTP Codec	No
Maximum Expiration	7200	Send Diversion Header	Yes
Minimum Expiration	600	Send P-Asserted-Identity	No
Outbound Proxy		Send Remote-Party-ID	No
Force rport	Yes	Support Path	No
ICE Support	No	Send Connected Line	No
Rewrite Contact	Yes	WebRTC	No
Remove Existing	Yes		

Media			
Media Encryption	SDES	RTP Timeout Hold	300
Max Video Streams	5	Direct Media	Yes
Max Audio Streams	5	Media Transport Received	No
Direct Media Glare Mitigation	None	Media Encryption Optimistic	No
RTP Keepalive	15	Disable NAT Direct Media	No
RTP Timeout	30		

Image 15.4.14 - PJSIP Device profile for MSTeams connection.

Here, add a **Name and Description** to identify the profile. Under **Network**, change the **Transport** to **TLS**, set the **Qualify Frequency** to **60**, and the **Qualify Timeout** to **3**. Then, make sure that **Force rport, Rewrite Contact, Remove Existing, RTP Symmetric, and Send Diversion Header** are **enabled**. The rest of the options will be left as default and the yes/no options will be disabled. Under **Media**, make sure **Media Encryption** is set to **SDES**. Then, only **Direct Media** is **enabled**, and the other yes/no options are disabled. You can leave the rest of the options with the default options. Click on **Save** and then **Apply Changes**.

Now, we will configure the **PJSIP Trunk** between VitalPBX and Microsoft© Teams™. Go to **PBX > Call Routing > Trunks.**



Technology	PJSIP | IAX2 | TENANT | CUSTOM	Get DID From	Default
Description *	MS Teams	Get CID From	Default
Class of Service	Trunk Default	Trunk CID	Trunk CID Name Trunk CID Number
Ring Time *	90	Overwrite CID	No
Dial Profile	Default	Dial Prefix	
Profile	MSTeams	Disable Trunk	No
Music on Hold	-- None --	Continue on Busy	No
Codecs	ulaw,alaw,gsm,silk8,silk12,silk16,silk24	Disable Hangup Audios	No
DTMF Mode	rfc4733	Calls Recording	No
Simultaneous Calls	Unlimited		

General Configurations

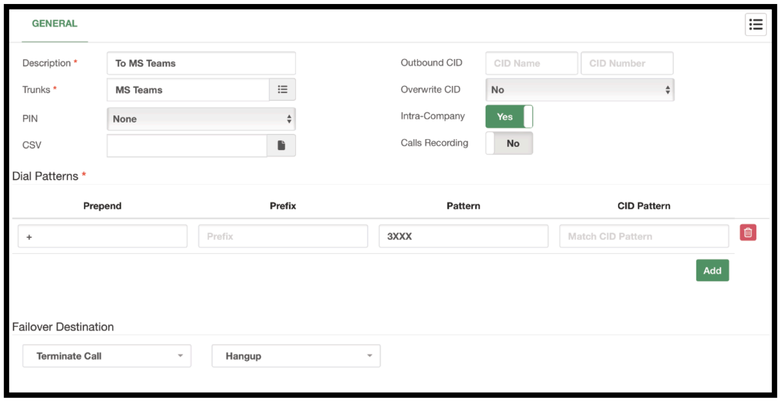
Local Username *	msteams	Remote Username	
Remote Host		Remote Secret	
Remote Port		From User	
Local Secret		From Domain	msteams.vitxi.com
Transport	TLS + MS Teams	Outbound Proxy	
Contacts	sip:sip.pstnhub.microsoft.com ×	Contact Header	
Match	sip-all.pstnhub.microsoft.com ×	Match Header	
Identify By		Trust ID Inbound	No
Qualify Frequency	30	Trust ID Outbound	No
Qualify Timeout	3	Inband Progress	No
Max Contacts	2		

15.4.15 - MS Teams PJSIP Trunk in VitalPBX.

Here, we select **PJSIP** as the **technology**. Then, set the **Profile** to the device profile we just created. Set the **Codecs** to **ulaw, alaw, gsm, and the silk codecs**. Under **General Configurations**, add any name without spaces to the **Local Username**. This will be used only to identify this trunk. Set the **Transport** to **TLS + MS Teams**. This transport is created when installing the MS Teams Connector add-on. For the **Contacts** enter **sip:sip.pstnhub.microsoft.com**, this is provided by Microsoft©. Under **Match** enter **sip-all.pstnhub.microsoft.com**, this is also provided by Microsoft©. Finally, under **From Domain**, enter the domain for your VitalPBX installation.

Then click on **Save** and then **Apply Changes**.

Next, we must configure our **Outbound Routes**. For this, go to **PBX > Calls Routing > Outbound Routes**.



Description *	To MS Teams	Outbound CID	CID Name CID Number
Trunks *	MS Teams	Overwrite CID	No
PIN	None	Intra-Company	Yes
CSV		Calls Recording	No

Dial Patterns *

Prepend	Prefix	Pattern	CID Pattern
+	Prefix	3XXX	Match CID Pattern

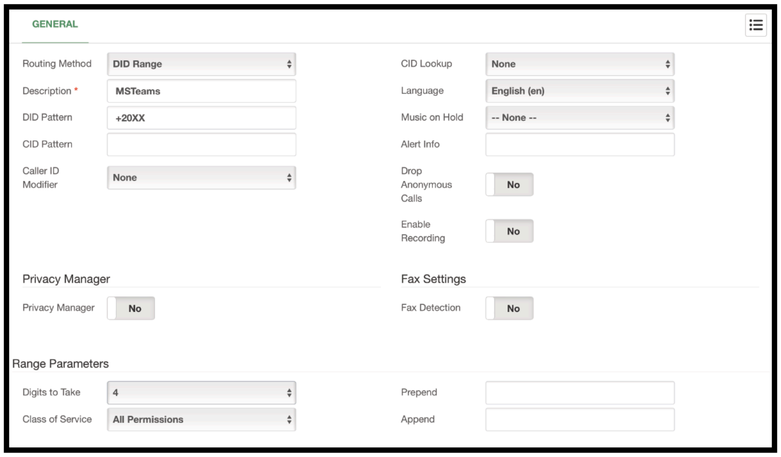
Failover Destination

Terminate Call	Hangup

Image 15.4.16 - Outbound Route using the MS Teams PJSIP Trunk.

Here, add a **Description** to identify the outbound route. Enable **Intra-Company** to use the internal CID information of extensions. Under Dial Patterns, we must **Prepend** a plus (**+**) sign to our **Pattern** since MS Teams only sends and receives calls with E.164 formatting. Since our MS Teams extensions start with a 3, we add the **3XXX** pattern, but this may vary based on your configurations. Click on **Save** and then **Apply Changes**.

Now, we must configure our **Inbound Routes**. For this, go to **PBX > Calls Routing > Inbound Routes**.



GENERAL

Routing Method	DID Range	CID Lookup	None
Description *	MSTeams	Language	English (en)
DID Pattern	+20XX	Music on Hold	-- None --
CID Pattern		Alert Info	
Caller ID Modifier	None	Drop Anonymous Calls	No
		Enable Recording	No

Privacy Manager

Privacy Manager: No

Fax Settings

Fax Detection: No

Range Parameters

Digits to Take	4	Prepend	
Class of Service	All Permissions	Append	

Image 15.4.17 - Inbound Route for calls coming from MS Teams.

For this, we will create an inbound route using a DID Range as the **Routing Method**. We then add a **description** to identify this inbound route. Under the **DID Pattern** we enter **+20XX**, as MS Teams will send the DID information with a plus (**+**) sign in front, and our extensions go from 2000 to 2099, in our example. This may vary based on your configurations. Under **Range Parameters**, we set the **Digits to Take** to **4**, as we are interested in the number dialed without the plus sign. This will essentially, remove the plus sign to find the appropriate extension. We can leave the rest of the options as default. Click on **Save** and then **Apply Changes**.

With this, you can now try calling a Microsoft© Teams™ user from a VitalPBX extension or calling the Auto-Attendant we have configured.

You have now successfully connected Microsoft© Teams™ with your VitalPBX installation.

Section 15, Lesson 5 - Custom Contexts

Now, let's explore how you can make customizations to the Asterisk dial plan, and set them as destinations in VitalPBX with the **Custom Contexts** add-on module. This is a free add-on module you can use to declare a custom Asterisk Context you create as a destination within VitalPBX.

With a custom context, you have the freedom to integrate any external service with VitalPBX at a deep level. Custom Contexts will allow you to interface between the Asterisk dial plan, VitalPBX, and any external service or application. You are expected to have Asterisk knowledge to use this feature within VitalPBX. Throughout this lesson, we will be showing one example of using Custom Contexts, but your imagination and Asterisk knowledge are your only limits.

Before using the Custom Contexts add-on, we need to have a **context** created. In this example, will be creating a service to playback the public IP Address of our server to callers that dial a special code. For this, we will be using the services of ipinfo.io and the CURL application for Asterisk.

Now, log in via **SSH** to your VitalPBX server, ideally as the root user. Use the following command to go to where the .conf files are located in VitalPBX.

```
root@debian:~# cd /etc/asterisk/vitalpbx
```

You will notice that all of the .conf files here have a number. This is used to maintain a sequence order of how the files are read. You can create a new .conf file using the following structure: **extensions__{ANYNUMBER}-{DESCRIPTION}.conf**, where **{ANYNUMBER}** is a consecutive number that represents the order that Asterisk will read the files, and the **{DESCRIPTION}** is any string used to identify what the file contains. So for our example, we will run the following command to create a new .conf file for our custom context using nano.

```
root@debian:/etc/asterisk/vitalpbx# nano extensions__90-custom.conf
```

With this, we make sure that our custom context is considered after the extensions's contexts.

Inside the **extensions__90-custom.conf** file, we add the following content as our context.

```
[say-public-ip]
exten => s,1,NoOp(Say Public IP Address)
 same => n,Answer()
 same => n,Set(PUBLIC_IP=${CURL(http://ipinfo.io/ip)})
 same => n,Playback(your-public-ip-is)
 same => n,SayDigits(${CUT(PUBLIC_IP,.,1)})
 same => n,Playback(letters/dot)
 same => n,SayDigits(${CUT(PUBLIC_IP,.,2)})
 same => n,Playback(letters/dot)
 same => n,SayDigits(${CUT(PUBLIC_IP,.,3)})
 same => n,Playback(letters/dot)
 same => n,SayDigits(${CUT(PUBLIC_IP,.,4)})
 same => n,PlayBack(silence/1&vm-goodbye)
 same => n,Hangup()
```

Then we **Save** and **Exit** nano. Then, we reload the Asterisk Dial Plan using the following command.

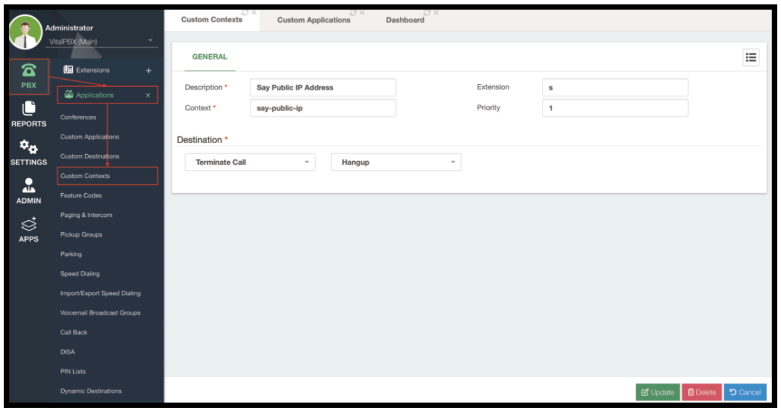
```
root@debian:/etc/asterisk/vitalpbx# asterisk -rx "dialplan reload"
```

> ***Note:*** *When creating the new .conf file, notice that there are 2 underscores(_) after extensions. So it is **extensions__90-custom.conf**, not **extensions_90-custom.conf**.*

You may notice the **your-public-ip-is** sound file that does not come with VitalPBX by default. This sound can be downloaded and added to your system using the following command.

```
   root@debian:~# wget -P /var/lib/asterisk/sounds/en/ https://github.com/VitalPBX/custom-
contexts/raw/master/sounds/your-public-ip-is.wav
```

With the context created, you must install the add-on from **Admin > Add-Ons > Add-Ons**. Once the add-on has been installed, refresh your browser, and go to **PBX > Applications > Custom Contexts**.



Description * Say Public IP Address

Context * say-public-ip

Extension s

Priority 1

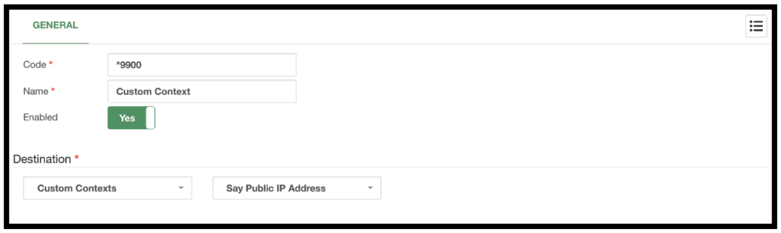
Destination *

Terminate Call | Hangup

Image 15.5.1 - Custom Context add-on module.

Here, you must add a **Description** to identify this context. Then under **Context**, enter the name of the context without the square brackets. Optionally, you can specify the **Extension** and **Priority** of the context. Under **Destination**, you can set a destination where you will take the user after they reach the custom context. Click on **Save** and then **Apply Changes**.

To test the custom context, we will create a **custom application** so we can have a code to dial the custom context. Go to **PBX > Applications > Custom Applications**.



Code * *9900

Name * Custom Context

Enabled Yes

Destination *

Custom Contexts | Say Public IP Address

Image 15.5.2 - Custom Application using a Custom Context as a destination.

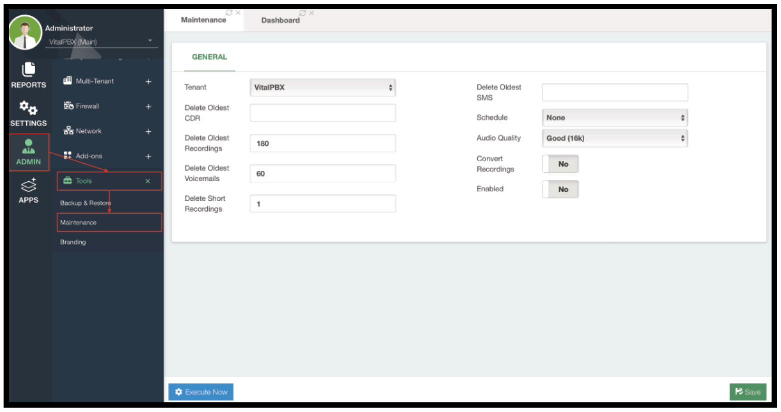
Add a **Code** to dial, set a **Name** to identify this custom application, and add the custom context as the **destination**. Then, **Save** and then **Apply Changes**. Now when you dial this code, you will go to your custom context.

Section 15, Lesson 6 - Maintenance

Now let's look at the **Maintenance** add-on. The Maintenance add-on is designed to save space on your VitalPBX server storage. With this add-on you control how long you keep voice recordings and voicemails, as well as converting the audio files from WAV to MP3 files.

The Maintenance add-on is a commercial add-on that requires a license to work to its full potential. You can acquire a license separately just for the Maintenance add-on, or it also comes included with any of our licensing plan subscriptions.

To start using the Maintenance add-on, you must install it through the add-ons module under **Admin > Add-Ons > Add-Ons**. Once the add-on is installed, refresh your browser and go to **Admin > Tools > Maintenance.** If you installed the Multi-Tenant add-on module, the Maintenance add-on is automatically installed.



Tenant: VitalPBX
Delete Oldest CDR:
Delete Oldest Recordings: 180
Delete Oldest Voicemails: 60
Delete Short Recordings: 1
Delete Oldest SMS:
Schedule: None
Audio Quality: Good (16k)
Convert Recordings: No
Enabled: No

Image 15.6.1 - Maintenance add-on module.

Here you can **Enable** the Maintenance for specific **Tenants** and the Main System. You can configure the time in days to **Delete the Oldest CDR, Recordings, Voicemails, and SMS messages**. So all call recordings, voicemails, CDR records, and SMS messages will be deleted if they are equal to or older than the set amount of days. You can set the time in seconds to consider the **deletion of short call recordings**. So call recordings shorter or equal to the specified time in seconds will be deleted. You can change the **Audio Quality** and **Convert Recordings** from WAV to MP3 files.

The maintenance can be run manually by clicking the **Execute Now** button in the lower left-hand corner, or you can **Schedule** the maintenance to run automatically using a **Cron Profile**. Once again, Cron Profiles can be created under **PBX > Tools > Cron Profiles**.

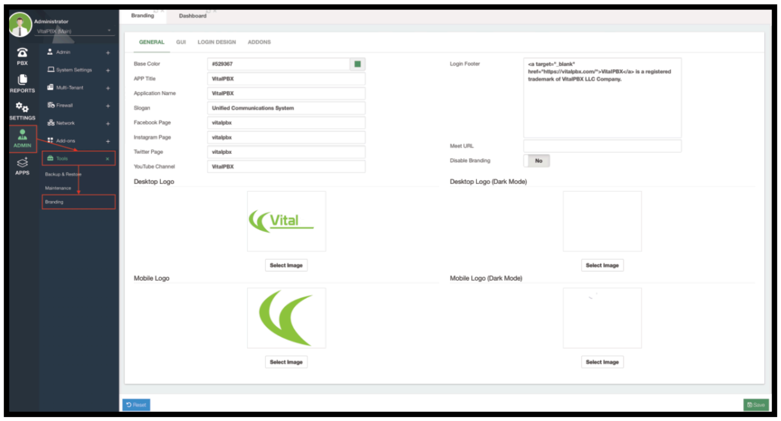
Once you have the maintenance set as you would like, click on **Save** and then **Apply Changes**.

Section 15, Lesson 7 - Branding

You can make your VitalPBX installation your own with the **Branding** add-on module. This add-on module will allow you to upload your logos, change the application name, change the base color, login design, and much more.

The Branding add-on is a commercial add-on that requires a license to work to its full potential. You can acquire a license separately just for the Branding add-on, or it also comes included with any of our licensing plan subscriptions.

To start using the Branding add-on, you must install it through the add-ons module under **Admin > Add-Ons > Add-Ons**. Once the add-on is installed, refresh your browser and go to **Admin > Tools > Branding**.



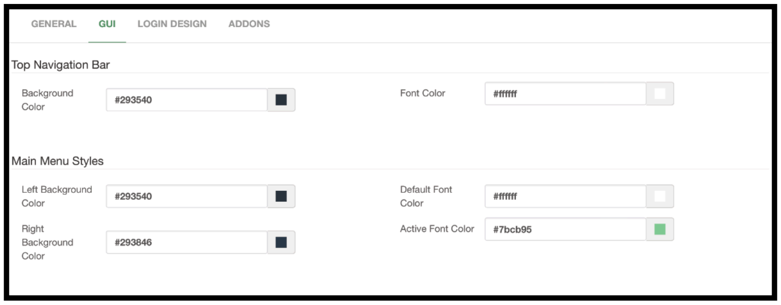
Base Color: #529367
APP Title: VitalPBX
Application Name: VitalPBX
Slogan: Unified Communications System
Facebook Page: vitalpbx
Instagram Page: vitalpbx
Twitter Page: vitalpbx
YouTube Channel: VitalPBX
Login Footer: <a target="_blank" href="https://vitalpbx.com/">VitalPBX</a> is a registered trademark of VitalPBX LLC Company.
Meet URL:
Disable Branding: No
Desktop Logo
Desktop Logo (Dark Mode)
Mobile Logo
Mobile Logo (Dark Mode)

Image 15.7.1 - Branding add-on module general tab.

On the **General tab**, you can change the **Base Color** for the whole interface, so lettering, highlights, and accented items will use this color instead of the base green. It is also possible to change the **APP Title and Name**, so there is no mention of VitalPBX on your application. This is perfect if you are using the Multi-Tenant add-on to provide a PBX on the cloud service to your customers. You can change the **Slogan and Social media links** that are displayed on the login screen, as well as the **Login Footer**. You can upload your **Logos for Desktop and Mobile view for Light and Dark Mode**. The mobile logo is also used for the Favicon. You can quickly **disable the branding** to revert to the default VitalPBX branding at any time. This is useful to check the differences or if an uploaded logo does not fit on the first try.

There is also a **Meet URL** option. This option will change the base URL for the video conferencing link under **PBX > Extras > Video Conferencing**. By default, we offer a courtesy server we host based on Jitsi™. You can learn how to set up a Jitsi™-based server in this article we made at https://vitalpbx.com/blog/jitsi-videoconference-with-vitalpbx/. With the Meet URL option, you can change the base URL to your meeting server.

Next, we have the **GUI tab**.



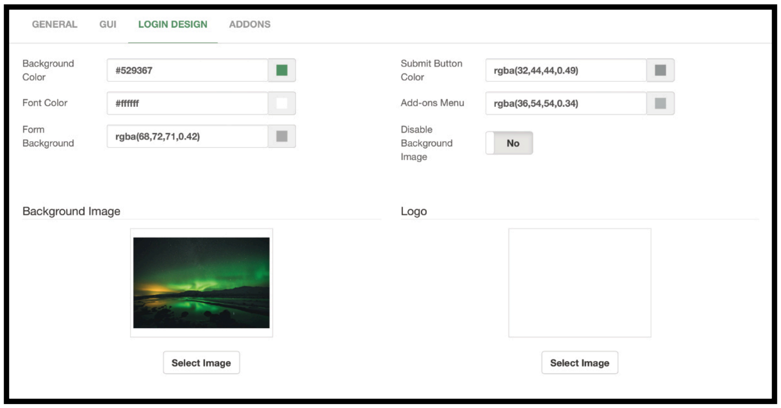
Top Navigation Bar
Background Color: #293540
Font Color: #ffffff

Main Menu Styles
Left Background Color: #293540
Default Font Color: #ffffff
Right Background Color: #293846
Active Font Color: #7bcb95

Image 15.7.2 - GUI tab for the Branding add-on module.

Here, you can change multiple colors found on the VitalPBX Web UI. Giving you more customization options for the **Top Navigation Bar** and **Main Menu** on the left-hand side.

Afterward, we have the **Login Design** tab.



Background Color: #529367
Font Color: #ffffff
Form Background: rgba(68,72,71,0.42)
Submit Button Color: rgba(32,44,44,0.49)
Add-ons Menu: rgba(36,54,54,0.34)
Disable Background Image: No

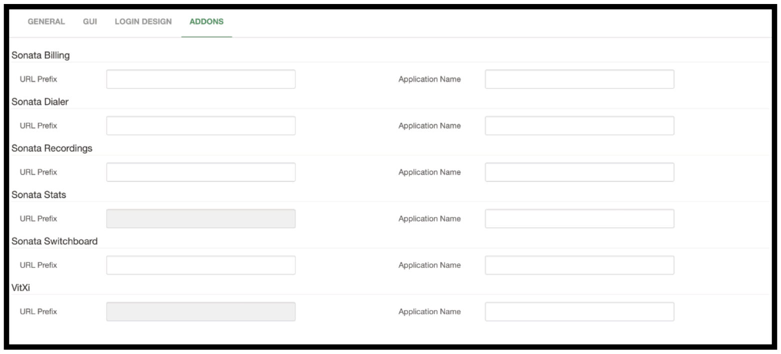
Background Image

Logo

Image 15.7.3 - Login Design tab for the Branding add-on module.

Here, you can change **various colors** for elements on the Login Screen. As well as the **Background Image and Logo** used on the Login Screen. You can also disable the background image if you want to use a solid color instead.

Finally, if you have a **Carrier Plus** Licensing Plan Subscription, you get the **Branding Plus** extended feature. This will add the **AddOns tab**. In this tab, you can change the **Application Name and URL Prefix** for the Sonata Suite and VitXi applications. This will also allow Sonata Suite and VitXi to use the same logos you uploaded on the general tab.



Sonata Billing
URL Prefix Application Name
Sonata Dialer
URL Prefix Application Name
Sonata Recordings
URL Prefix Application Name
Sonata Stats
URL Prefix Application Name
Sonata Switchboard
URL Prefix Application Name
VitXi
URL Prefix Application Name

Image 15.7.4 - AddOns tab for the Branding Plus add-on module.

Once you are happy with your changes, you can click on **Save** in the lower right-hand corner. If you are not happy with the styles you have given to your VitalPBX, you can always click on **Reset** in the lower left-hand corner, reset all the values to their default, and start over.

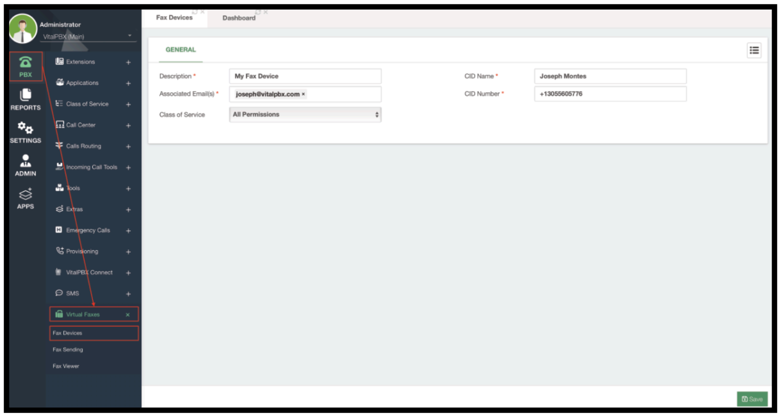
With the Branding module, you can give your VitalPBX your personal touch and have it align with your company's brand. Giving your customers a more professional look to the services you provide.

Section 15, Lesson 8 - Virtual Faxes

Now let's go through how you can send and receive Faxes from your VitalPBX, using the **Virtual Faxes** add-on module. Even though faxing may sound like a very old way of communication (and it is), it is still widely used in various fields like the law or medical fields.

The Virtual Faxes add-on is a commercial add-on that requires a license to work to its full potential. You can acquire a license separately just for the Virtual Faxes add-on, or it also comes included with any of our licensing plan subscriptions.

To start using the Virtual Faxes add-on, you must install it through the add-ons module under **Admin > Add-Ons > Add-Ons**. Once the add-on is installed, refresh your browser and go to **PBX > Virtual Faxes > Fax Devices**.



Description * My Fax Device

Associated Email(s) * joseph@vitalpbx.com

Class of Service All Permissions

CID Name * Joseph Montes

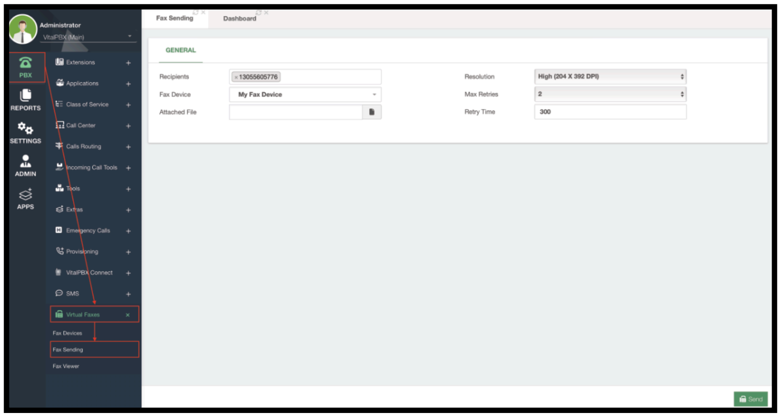
CID Number * +13055605776

Image 15.8.1 - Virtual Faxes's Fax Devices module.

First, you must create a **Fax Device**. This is like having a fax machine inside of VitalPBX. This is a device we will use to send and receive faxes. You will need to add a **Description** to identify the device. Then, add the **Associated Email(s)**, as these will be used when receiving a fax, so it is sent to the email addresses set here. You can enter as many email addresses as you need. You must have the **Email Client** set up to use the **Fax to Email** feature. You must set a **Caller ID Name and Number**, as this is the CID information that will be used when sending a Fax. Finally, set an appropriate **Class of Service**, so the device can send faxes to their destinations.

With the fields configured, you can **Save** and then **Apply Changes**.

To **Send** a Fax from the VitalPBX Web Interface, you must go to **PBX > Virtual Faxes > Fax Sending**.

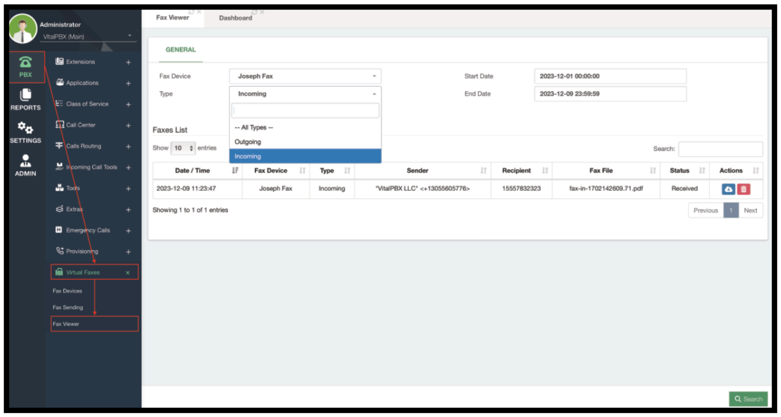


Recipients	× 13055605776	Resolution	High (204 X 392 DPI)
Fax Device	My Fax Device	Max Retries	2
Attached File		Retry Time	300

Image 15.8.2 - Virtual Faxes's Fax Sending module.

Here, you must enter the **Recipients** to whom you are sending the fax. You can enter multiple phone numbers in this field. Make sure that you enter the number exactly as you would dial it, including any prefixes if needed. Then, select the **Fax Device** you wish to use. Afterward, you will need to **attach the file** you wish to send. This must be a **TIFF, TXT, or PDF** file. You can select the **Resolution** in which you wish to send it. You can set the number of **Max Retries** to try and send the fax, as well as the **Retry Time** between each try. Once you have filled out the fields, click on **Send** in the bottom right-hand corner.

When you **receive** a fax, you will have it sent to the email addresses set on the Fax Device, but you can also view faxes from the VitalPBX Web Interface. Go to **PBX > Virtual Faxes > Fax Viewer**.



Fax Device	Joseph Fax	Start Date	2023-12-01 00:00:00
Type	Incoming	End Date	2023-12-09 23:59:59

Faxes List

Show 10 entries

Date / Time	Fax Device	Type	Sender	Recipient	Fax File	Status	Actions
2023-12-09 11:23:47	Joseph Fax	Incoming	"VitalPBX LLC" <+13055605776>	15557832323	fax-in-1702142609.71.pdf	Received	

Showing 1 to 1 of 1 entries

Image 15.8.3 - Virtual Faxes's Fax Viewer module.

Here, you can see the faxes for a specific **Fax Device**, set the **type** to **All Types, Incoming, or Outgoing**, and set a **Start and End Date** to filter the faxes you wish to see. You can click the **Search** button in the lower right-hand corner to apply your filters.

Your faxes will be listed on the Faxes List, and you will see information on when they were sent or received, by whom, and the fax content information. Under the **Actions** column, you can download the Fax Content, and delete any faxes.

With this, you can now send and receive as many faxes as you need directly from your VitalPBX. Allowing you to digitize your faxing abilities.

Conclusion

With this, we have now covered all of the add-ons available at this time. As you can see, you can expand your VitalPBX installation's features with various powerful tools to make the most of your VitalPBX. We have covered other add-ons throughout this guide, so you can find them in their respective lessons. Once again, all of the commercial add-ons can also be installed on the Community version of VitalPBX, allowing you to test all of the features without worrying about any time or feature limits. The only limitation without a license, is the number of items you can create or monitor in the commercial add-on.

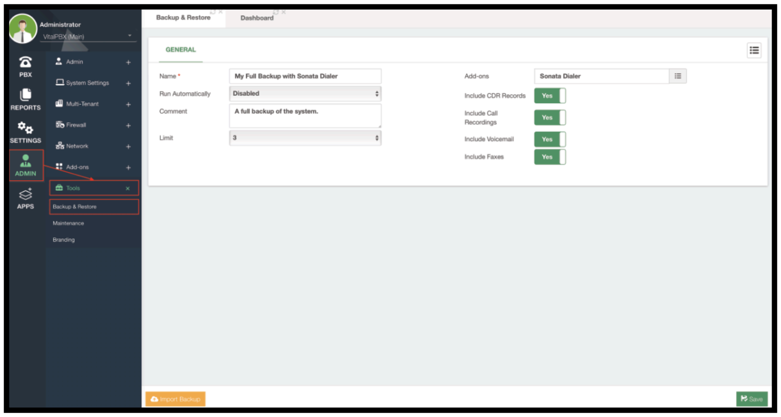
SECTION 16 - TOOLS

Introduction

In this section, we will be taking a look into some additional tools that will aid you with your VitalPBX installation. You will be able to create a backup you can restore in case you reinstall VitalPBX in a separate server or on the same one if anything goes wrong. You will learn how to find weak passwords, read the logs from the WebUI, and add numbers to a blacklist so they don't reach your main routes.

Section 16, Lesson 1 - Backup and Restore

With all of the configurations you have made to your VitalPBX, we must be able to save them in case we want to migrate to another server or restore our current server to an older set of configurations. For this, we have a **Backup and Restore** module in VitalPBX. Go to **Admin > Tools > Backup and Restore**.

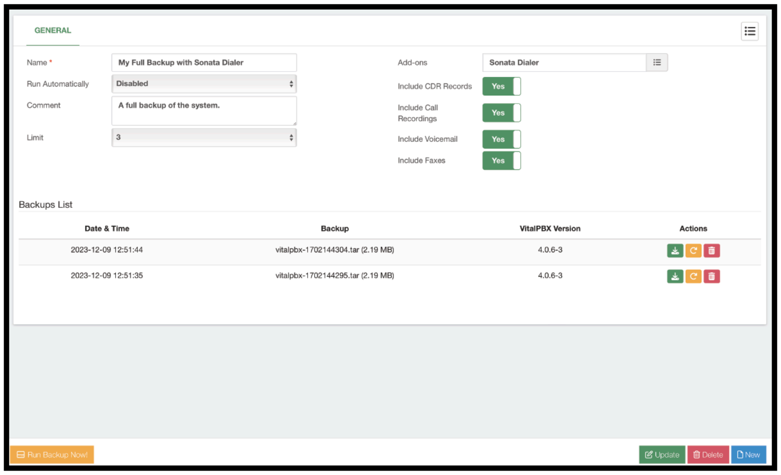


Name * My Full Backup with Sonata Dialer
Run Automatically Disabled
Comment A full backup of the system.
Limit 3
Add-ons Sonata Dialer
Include CDR Records Yes
Include Call Recordings Yes
Include Voicemail Yes
Include Faxes Yes

Image 16.1.1 - Backup and Restore module.

Here, you can create multiple **Backup Profiles**, that will generate backup files you can export and use on your VitalPBX installation. For this, enter a **Name** to identify the Backup Profile. You can add a brief **comment** to describe what the backup includes. You can select if you want to back up the configurations for any **Sonata Suite and VitXi add-on application**. Then, you can select to include the **CDR Records, Call Recordings, Voicemails, and/or Faxes**. Since this is a profile you can run multiple times, you can set a **limit** on how many backups you wish to keep stored on your server at any time. You can automate the backup creation process by setting a **Cron Profile** on the **Run Automatically** option. With your settings configured, click on **Save**.

You will now find the **Backup Profile** on the list in the upper right-hand corner. If you enter your Backup Profile again, you can edit what the profile contains, and run the backup manually by clicking on **Run Backup Now!** in the lower left-hand corner.



Name * My Full Backup with Sonata Dialer
Run Automatically Disabled
Comment A full backup of the system.
Limit 3
Add-ons Sonata Dialer
Include CDR Records Yes
Include Call Recordings Yes
Include Voicemail Yes
Include Faxes Yes

Backups List

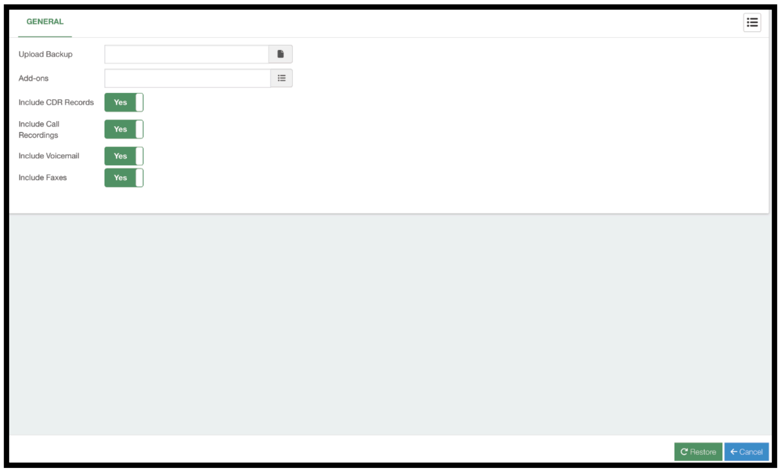
Date & Time	Backup	VitalPBX Version	Actions
2023-12-09 12:51:44	vitalpbx-1702144304.tar (2.19 MB)	4.0.6-3	
2023-12-09 12:51:35	vitalpbx-1702144295.tar (2.19 MB)	4.0.6-3	

Image 16.1.2 - Backup Profile with backup files.

Whenever you run a backup, you will see the backup files listed on the **Backups List**. Here, you will see information on **when the backup was created**, the **backup file name**, the **VitalPBX version** of the backup, and some **actions**. From the actions column, you can **download** the backup, **delete** it, or **restore** from the backup.

> ***Note:*** *You must make sure your server has enough storage to store the backup files. Especially, if you are backing up your call recordings. As call recordings can take a lot of space, we recommend you back them up in a separate server, by either mounting a NAS or SAN server to the call recording directory at **/var/spool/asterisk/monitor** or using an SCP or SFTP application to download the call recordings directly. You can then upload them to the VitalPBX server after you restore from your backup.*

To restore from a backup that you created in a separate server, you must go back to **Admin > Tools > Backup and Restore**, and click on **Import Backup** in the lower left-hand corner. This will allow you to select a backup file you have on your system, and select what you want to import from the backup file.



Upload Backup

Add-ons

Include CDR Records: Yes

Include Call Recordings: Yes

Include Voicemail: Yes

Include Faxes: Yes

Image 16.1.3 - Import Backup feature from the Backup and Restore module.

Once you have selected the backup file, and what you wish to import from the backup file, click on the **Restore** button in the lower right-hand corner.

With this, you are now able to backup and restore your VitalPBX configurations.

Section 16, Lesson 2 - Asterisk CLI (Web)

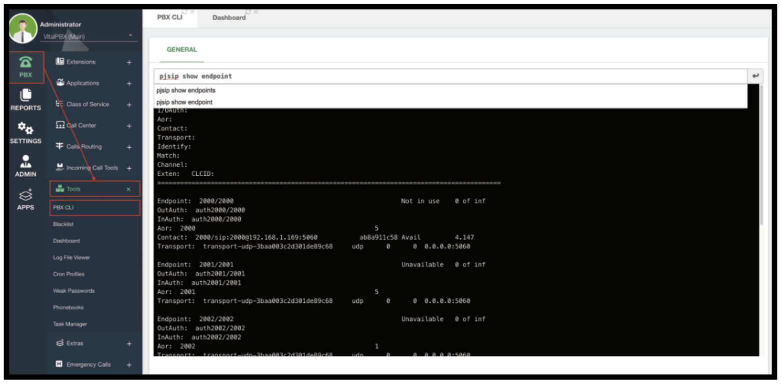
Part of allowing you to monitor your VitalPBX installation is the ability to view the Asterisk CLI. Even though we recommend doing this through an SSH connection to your server and running the following command.

```
root@debian:~# asterisk -rvvvvvvv
```

Or simply

```
root@debian:~# a
```

With VitalPBX you can view the Asterisk CLI from the Web UI, allowing for a quick look. Go to **PBX > Tools > PBX CLI**.



```
pjsip show endpoint
```

pjsip show endpoints
pjsip show endpoint

```
I/OAuth:
Aor:
Contact:
Transport:
Identify:
Match:
Channel:
Exten:   CLCID:
==========================================================================================

Endpoint:  2000/2000                                      Not in use    0 of inf
OutAuth:  auth2000/2000
InAuth:  auth2000/2000
Aor:  2000                                         5
Contact:  2000/sip:2000@192.168.1.169:5060          ab8a911c58 Avail         4.147
Transport:  transport-udp-3baa003c2d301de89c68     udp      0      0  0.0.0.0:5060

Endpoint:  2001/2001                                      Unavailable   0 of inf
OutAuth:  auth2001/2001
InAuth:  auth2001/2001
Aor:  2001                                         5
Transport:  transport-udp-3baa003c2d301de89c68     udp      0      0  0.0.0.0:5060

Endpoint:  2002/2002                                      Unavailable   0 of inf
OutAuth:  auth2002/2002
InAuth:  auth2002/2002
Aor:  2002                                         1
Transport:  transport-udp-3baa003c2d301de89c68     udp      0      0  0.0.0.0:5060
```

Image 16.2.1 - PBX CLI module.

Here you can run any Asterisk command from the Web UI. This will allow you to monitor your VitalPBX's Asterisk layer easily from the web.

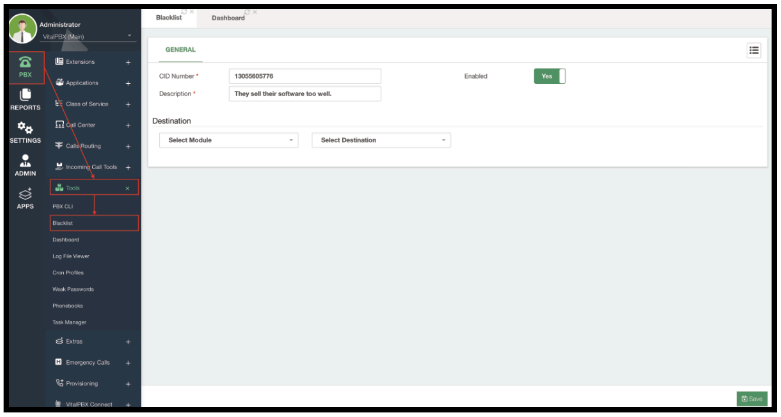
Commands can be entered from the text field on the top. As you type them, suggestions for the command string will appear. To run the command, click on the **Enter Arrow** at the end of the text field.

The outcome of the command will be displayed in the black box below the text field.

With this, you can run any Asterisk command from the web interface.

Section 16, Lesson 3 - Blacklist

Sometimes, you may want to block entire callers from dialing your VitalPBX. As an additional security layer, you can block numbers using our **Blacklist** module. Go to **PBX > Tools > Blacklist**.



CID Number * 13055605776

Description * They sell their software too well.

Enabled Yes

Images 16.3.1 - Blacklist module.

This is a simple module to use. To blacklist a number, enter their **Caller ID number** and add a **description** to identify this blacklisted number. You can specify a destination to take these blacklisted numbers if you want. This is optional as the default action is to playback the Blacklist sound file set under **Settings > PBX Settings > System General > System Prompts**. If you don't want the caller to know they're blacklisted, you can simply set the Destination to terminate the call and hang up.

Once you set the blacklisted number, click on **Save** in the bottom right-hand corner. You can see a list of numbers that have been blacklisted in the upper right-hand corner. You can **Enable and Disable** a blacklisting at any time you like without having to delete the entry, only saving the change.

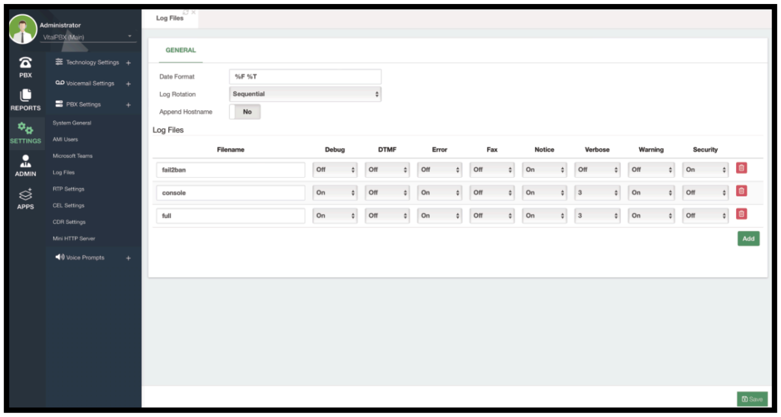
Another way to blacklist a number is by using the feature code ***30** and following the prompts to add a number you wish to blacklist, or ***32** which will blacklist the last caller on your extension's call history. You can use the code ***31** to remove a number from the blacklist.

Now, you can use this feature to make your VitalPBX more secure and block any unnecessary or disturbing callers from your VitalPBX installation.

Section 16, Lesson 4 - Log Files and Log File Viewer

Another part of allowing you to monitor your VitalPBX is to be able to see the log files for different aspects of your system. For this, we have the **Log Files** and **Log File Viewer** modules in VitalPBX.

First, let's take a look at the **Log Files** module. Go to **Settings > PBX Settings > Log Files**.



Date Format: %F %T

Log Rotation: Sequential

Append Hostname: No

Log Files

Filename	Debug	DTMF	Error	Fax	Notice	Verbose	Warning	Security
fail2ban	Off	Off	Off	Off	On	Off	Off	On
console	On	Off	On	Off	On	3	On	Off
full	On	Off	On	Off	On	3	On	Off

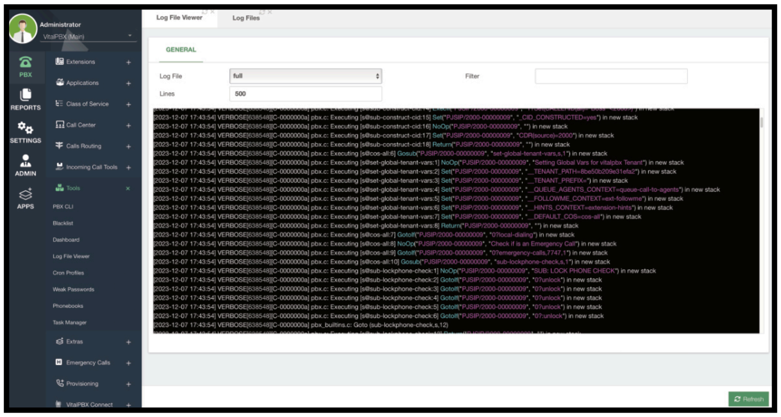
Image 16.4.1 - Log Files module.

Here, you can define the log files that can be viewed from the **Log File Viewer**. You can change the Date Format using the **standard strftime format string**. On **Log Rotation** you can choose between the following.

- **Sequential -** Renames the log files in order. This means that the file that has the highest number is the most recent one.
- **Timestamp -** This uses a timestamp rather than a sequential number when the logger rotate task is executed.
- **Rotate -** This rotates all the log files. This means that the log file with the lowest number is the most recent one. This is the expected behavior for Unix users.

You can choose to **append the hostname** as well. This will append the hostname to the log files. Below, you will see the list of **Log Files** and the type of events you want to include. You can add additional log files by specifying the path and log file name. Once done, you can click on **Save**.

To view the Log Files, we must go to **PBX > Tools > Log File Viewer**.



Log File: full

Lines: 500

Filter:

```
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@sub-construct-cid:15] Set("PJSIP/2000-00000009", "_CID_CONSTRUCTED=yes") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@sub-construct-cid:16] NoOp("PJSIP/2000-00000009", "") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@sub-construct-cid:17] Set("PJSIP/2000-00000009", "CDR(source)=2000") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@sub-construct-cid:18] Return("PJSIP/2000-00000009", "") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@cos-all:6] Gosub("PJSIP/2000-00000009", "set-global-tenant-vars,s,1") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@set-global-tenant-vars:1] NoOp("PJSIP/2000-00000009", "Setting Global Vars for vitalpbx Tenant") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@set-global-tenant-vars:2] Set("PJSIP/2000-00000009", "__TENANT_PATH=8be50b209e31efa2") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@set-global-tenant-vars:3] Set("PJSIP/2000-00000009", "__TENANT_PREFIX=") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@set-global-tenant-vars:4] Set("PJSIP/2000-00000009", "__QUEUE_AGENTS_CONTEXT=queue-call-to-agents") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@set-global-tenant-vars:5] Set("PJSIP/2000-00000009", "__FOLLOWME_CONTEXT=ext-followme") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@set-global-tenant-vars:6] Set("PJSIP/2000-00000009", "__HINTS_CONTEXT=extension-hints") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@set-global-tenant-vars:7] Set("PJSIP/2000-00000009", "__DEFAULT_COS=cos-all") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@set-global-tenant-vars:8] Return("PJSIP/2000-00000009", "") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@cos-all:7] GotoIf("PJSIP/2000-00000009", "0?local-dialing") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@cos-all:8] NoOp("PJSIP/2000-00000009", "Check if is an Emergency Call") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@cos-all:9] GotoIf("PJSIP/2000-00000009", "0?emergency-calls,7747,1") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@cos-all:10] Gosub("PJSIP/2000-00000009", "sub-lockphone-check,s,1") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@sub-lockphone-check:1] NoOp("PJSIP/2000-00000009", "SUB: LOCK PHONE CHECK") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@sub-lockphone-check:2] GotoIf("PJSIP/2000-00000009", "0?unlock") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@sub-lockphone-check:3] GotoIf("PJSIP/2000-00000009", "0?unlock") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@sub-lockphone-check:4] GotoIf("PJSIP/2000-00000009", "0?unlock") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@sub-lockphone-check:5] GotoIf("PJSIP/2000-00000009", "0?unlock") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx.c: Executing [s@sub-lockphone-check:6] GotoIf("PJSIP/2000-00000009", "0?:unlock") in new stack
[2023-12-07 17:43:54] VERBOSE[638548][C-0000000a] pbx_builtins.c: Goto (sub-lockphone-check,s,12)
```

Image 16.4.2 - Log File Viewer module.

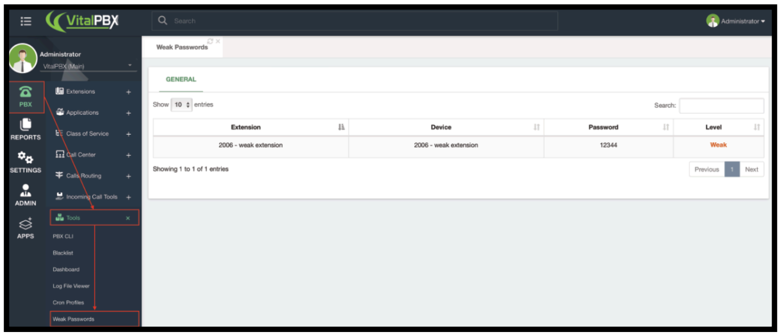
Here, you can select the **Log File** from the ones we have available, enter the **number of lines** you want to see, and you can **filter** for a specific keyword or phrase.

The File will be output in the black box below with color coding for easy viewing. You can click on the **Refresh** button in the lower right-hand corner to view the most recent information based on the filters you apply.

All of the log files we have by default are located under the **/var/log** directory. So you can view them from the Linux CLI as well. You can apply **CAT** filters to them from the CLI to make the search as custom as you need.

Section 16, Lesson 5 - Weak Passwords

As you create multiple users and passwords for device registration, you want to make sure that you don't have any weak passwords on your system. For this, we have the **Weak Passwords** module. Go to **PBX > Tools > Weak Passwords**.



Show 10 entries

Search:

Extension	Device	Password	Level
2006 - weak extension	2006 - weak extension	12344	Weak

Showing 1 to 1 of 1 entries

Image 16.5.1 - Weak Passwords module.

This is a simple module that will list all devices that have a weak password. You can see the **Extension number**, the **Device** for this extension, the **Password**, and the **Password Level**. This will help you identify the devices on your system that are weak so you can address them. We recommend using auto-generated passwords for devices to keep them secure.

Conclusion

With these additional tools, you can monitor your VitalPBX to see that everything is set up correctly and securely. Allowing you to monitor various aspects of your VitalPBX from the Web UI, for a quick and easy checkup.

SECTION 17 - API AND AMI

Introduction

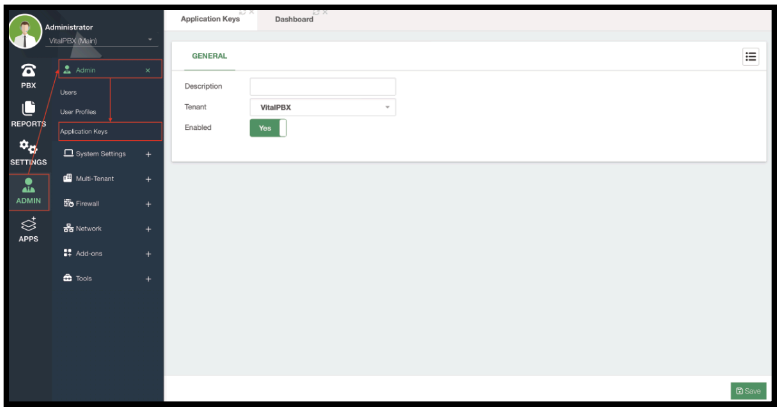
You can integrate third-party applications with VitalPBX. For this, we have our API and AMI (Asterisk Management Interface) capabilities. Here, we will show you how to create the users and keys so you can integrate third-party applications with VitalPBX.

Section 17, Lesson 1 - API

VitalPBX has an API that allows you to retrieve information from VitalPBX and any of its tenants. The VitalPBX is constantly growing and more features are added constantly. You can see our full API documentation in the following link.

https://documenter.getpostman.com/view/5481262/2s935hQmgP

To create your API Key to use with third-party applications, go to Admin > Admin > Application Keys.



Description:
Tenant: VitalPBX
Enabled: Yes

Image 17.1.1 - VitalPBX Application Keys module.

Here, you just need to enter a **description** to identify the key and select which tenant you wish to have the API connection with. You can choose a specific **tenant**, or select **Any Tenant** to be able to get API calls from any tenant.

You can **enable or disable** the API Key at any moment. Once done, click on **Save**. This will generate the API Key. If you go back to the Key you created, in the list button in the upper right-hand corner, you will see the API key you can copy and use with your third-party application.

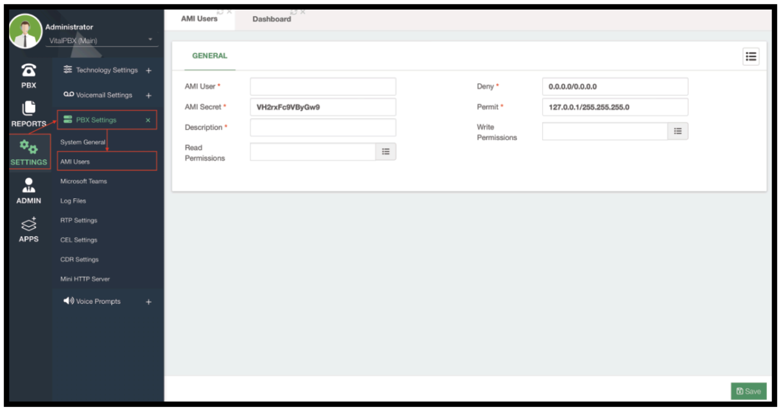
Section 17, Lesson 2 - AMI (Asterisk Management Interface)

Another way to integrate third-party applications is using the **AMI** or **Asterisk Management Interface**. This way, you can connect directly to the Asterisk layer VitalPBX uses and monitor and manage Asterisk directly. This is useful if you require real-time information to trigger events or monitor information on a third-party application.

You can see the full AMI documentation on the following link.

https://docs.asterisk.org/Configuration/Interfaces/Asterisk-Manager-Interface-AMI/

To create an AMI user in VitalPBX, go to **Settings > PBX Settings > AMI Users**. Here, you will create the AMI users that will connect directly with Asterisk.



AMI User *
AMI Secret * VH2rxFc9VByGw9
Description *
Read Permissions
Deny * 0.0.0.0/0.0.0.0
Permit * 127.0.0.1/255.255.255.0
Write Permissions

Image 17.2.1 - VitalPBX AMI users module.

You will need to define an **AMI User and Secret**. We generate a random password, and we recommend you use this randomly generated password for the best security. You will also need to enter a **description** to identify this user.

We also require you to use the **Deny and Permit** options for this user. With these fields, you specify the IP address or Network segment that can access Asterisk using this AMI user. It is important that you only allow the specific IP addresses that can connect to Asterisk using this AMI user as this is an extremely permissive user. An AMI user can affect your Asterisk directly, so we need to take all the precautions necessary.

You can also limit the read and write permissions, so the AMI users are only able to perform the tasks you need for your third-party application integration.

With everything set, you can **Save** and then **Apply Changes**.

With this, you are now able to connect directly with Asterisk and monitor and manage the Asterisk layer for VitalPBX from a third-party application.

SECTION 18 - TROUBLESHOOTING

Introduction

Up to now, we have covered every single module VitalPBX offers. We don't always have smooth sailing as we can make mistakes during the configuration process. Here are some of the most common and useful ways to troubleshoot your VitalPBX installation.

Section 18, Lesson 1 - Fail2Ban Commands

Sometimes you might get blocked by the Intrusion Detection system. This can be due to a device using the wrong password to register, and this blocks you out. Or, the wrong password is entered too many times through the Web UI. We recommend you whitelist your known IP addresses through the Firewall Whitelist. But in the case, you have been blocked and you have access to the Linux terminal, here are some commands for **Fail2Ban** that you can use to manage the intrusion detection system from a command line level.

To see the possible Fail2Ban commands, run the following command from the Linux Command Line.

```
root@debian:~# man fail2ban-client
```

To see the clients that have been banned due to too many failed attempts through SSH, run the following command.

```
root@debian:~# fail2ban-client status sshd
```

To see the clients that have been banned due to too many failed attempts to register through Asterisk, run the following command.

```
root@debian:~# fail2ban-client status asterisk
```

To unban an IP Address, run the following command.

```
root@debian:~# fail2ban-client set sshd unbanip {IP_ADDRESS}
```

Where **{IP_ADDRESS}** is the exact IP address to unban. Do not include the curly braces.

To ban an IP address, run the following command.

```
root@debian:~# fail2ban-client set sshd banip {IP_ADDRESS}
```

Where **{IP_ADDRESS}** is the exact IP address to ban. Do not include the curly braces.

To block an IP address using IPTables, run the following command.

```
root@debian:~# iptables -I INPUT 1 -s IPAddress -j DROP
```

To unblock an IP address using IPTables, run the following command.

```
root@debian:~# iptables -D INPUT -s IPAddress -j DROP
```

Using these commands you can manage the Intrusion Detection system directly through the Linux Command Line.

Section 18, Lesson 2 - Call Filter

When troubleshooting a call process we always go to the Asterisk CLI first to see if we are receiving the calls and trace the call to see what is the outcome that we get. We use the following command to get to the Asterisk CLI.

```
root@debian:~# asterisk -rvvvvvvvvv
```

Or simply

```
root@debian:~# a
```

However, monitoring the Asterisk CLI with high verbosity while the system is under production can show us a lot of information at a very fast pace. For this, we can filter out the Full log for Asterisk to trace actions that affect a particular call only.

We start by running the following command.

```
root@debian:~# tail -f /var/log/asterisk/full | grep {PHONE_NUMBER}
```

Where **{PHONE_NUMBER}** is a particular phone number you want to trace within the full log. Remember that you don't have to include the curly braces. For example, you will enter something like this.

```
root@debian:~# tail -f /var/log/asterisk/full | grep 15554846868
```

In the beginning, you may see nothing if information including this number has not been set. Once any activity starts that includes this phone number you will start to see anything related to it.

This phone number you monitor can be an external number calling into your VitalPBX and you want to see the route it uses. Or it can be a DID on your system and you want to see that it is used appropriately for your inbound routes. This can help you monitor how your VoIP provider sends your DID information so you can enter it correctly on your Inbound Routes. It can also be a specific Extension Number you want to monitor their current call. Use this option to make your call troubleshooting easier during active hours on your VitalPBX.

Section 18, Lesson 3 - SNGrep

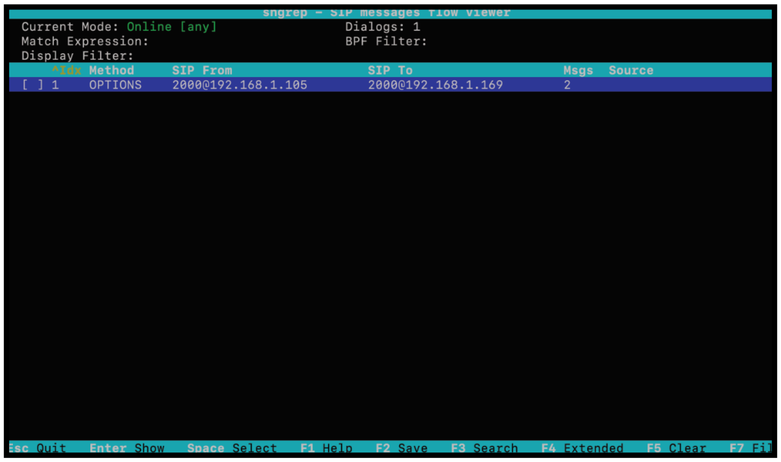
A more advanced way to trace events on your VitalPBX is using **SNGrep**. SNGrep is a tool that allows you to monitor SIP/PJSIP events on your VitalPBX. Here, you can see detailed information for your registrations, RTP traffic, and notices of devices and trunks connected to your VitalPBX. SNGrep is installed as one of the dependencies for VitalPBX. You can run the following command to see the available attributes on the application.

```
root@debian:~# sngrep --help
```

To run SNGrep simply enter the following command.

```
root@debian:~# sngrep
```

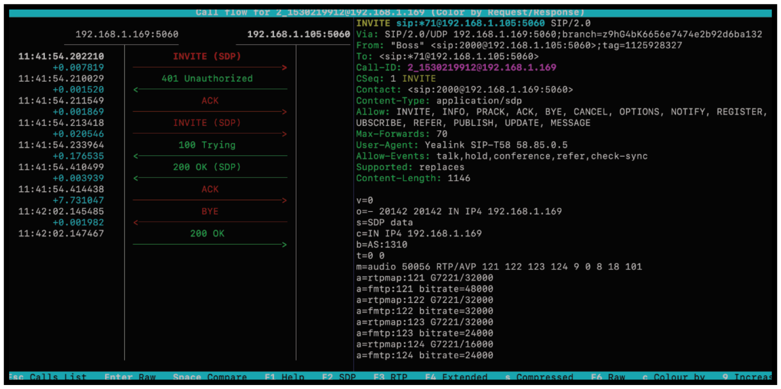
This will show you the SNGrep interface.



```
                     sngrep - SIP messages flow viewer
 Current Mode: Online [any]                  Dialogs: 1
 Match Expression:                           BPF Filter:
 Display Filter:
    ^Idx Method     SIP From                   SIP To                     Msgs  Source
 [ ] 1   OPTIONS    2000@192.168.1.105         2000@192.168.1.169         2
```

Image 18.3.1 - SNGrep Interface.

Here, you will see a list of events occurring on your VitalPBX. You can navigate the list of events using the arrow keys. You can enter an event by pressing the **Enter** key on your keyboard. This will show you all the details of the event. You can navigate the stages using the arrow keys.



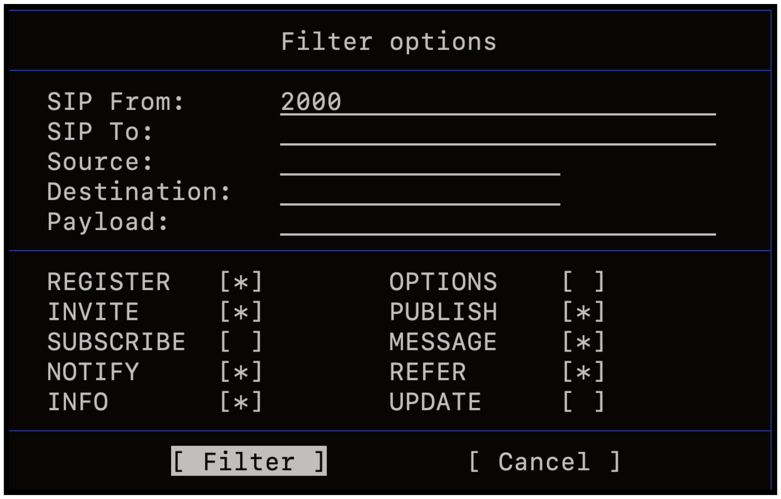
```
              Call flow for 2_1530219912@192.168.1.169 (Color by Request/Response)
      192.168.1.169:5060          192.168.1.105:5060
11:41:54.202210        INVIITE (SDP)
       +0.007819   ────────────────────>
11:41:54.210029     401 Unauthorized
       +0.001520   <────────────────────
11:41:54.211549          ACK
       +0.001869   ────────────────────>
11:41:54.213418        INVITE (SDP)
       +0.020546   ────────────────────>
11:41:54.233964        100 Trying
       +0.176535   <────────────────────
11:41:54.410499        200 OK (SDP)
       +0.003939   <────────────────────
11:41:54.414438          ACK
       +7.731047   ────────────────────>
11:42:02.145485          BYE
       +0.001982   <────────────────────
11:42:02.147467         200 OK
                   ────────────────────>
```

```
INVITE sip:*71@192.168.1.105:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.169:5060;branch=z9hG4bK6656e7474e2b92d6ba132
From: "Boss" <sip:2000@192.168.1.105:5060>;tag=1125928327
To: <sip:*71@192.168.1.105:5060>
Call-ID: 2_1530219912@192.168.1.169
CSeq: 1 INVITE
Contact: <sip:2000@192.168.1.169:5060>
Content-Type: application/sdp
Allow: INVITE, INFO, PRACK, ACK, BYE, CANCEL, OPTIONS, NOTIFY, REGISTER,
UBSCRIBE, REFER, PUBLISH, UPDATE, MESSAGE
Max-Forwards: 70
User-Agent: Yealink SIP-T58 58.85.0.5
Allow-Events: talk,hold,conference,refer,check-sync
Supported: replaces
Content-Length: 1146

v=0
o=- 20142 20142 IN IP4 192.168.1.169
s=SDP data
c=IN IP4 192.168.1.169
b=AS:1310
t=0 0
m=audio 50056 RTP/AVP 121 122 123 124 9 0 8 18 101
a=rtpmap:121 G7221/32000
a=fmtp:121 bitrate=48000
a=rtpmap:122 G7221/32000
a=fmtp:122 bitrate=32000
a=rtpmap:123 G7221/32000
a=fmtp:123 bitrate=24000
a=rtpmap:124 G7221/16000
a=fmtp:124 bitrate=24000
```

Image 18.3.2 - Example of an event in SNGrep.

This will help you get direct information on the current status of your devices, calls, and registration.

You will see that we will have a lot of information that is constantly being sent on the main list of events. For this, we can filter our list to see only the events we are interested in. So, on the main page for SNGrep, press the **F7** key on your keyboard. This will show you a prompt where you can filter the events by the **SIP From and To**, **Source and Destination**, and **Payload**. You can also apply a filter by the **type** of event you want to monitor.

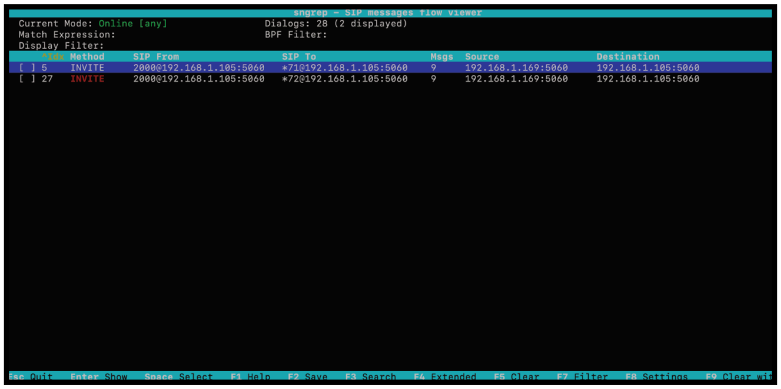


```
                 Filter options

SIP From:       2000
SIP To:
Source:
Destination:
Payload:

REGISTER   [*]         OPTIONS    [ ]
INVITE     [*]         PUBLISH    [*]
SUBSCRIBE  [ ]         MESSAGE    [*]
NOTIFY     [*]         REFER      [*]
INFO       [*]         UPDATE     [ ]

           [ Filter ]          [ Cancel ]
```

Image 18.3.3 - SNGrep filter options.

With your filters set, navigate with the arrow keys to **Filter** and press **Enter** on your keyboard.

You will see that the list of events will now show the events related to your filter.



```
                              sngrep - SIP messages flow viewer
Current Mode: Online [any]                 Dialogs: 28 (2 displayed)
Match Expression:                          BPF Filter:
Display Filter:
     ^Idx Method     SIP From                SIP To                  Msgs  Source               Destination
[ ] 5    INVITE     2000@192.168.1.105:5060 *71@192.168.1.105:5060  9     192.168.1.169:5060   192.168.1.105:5060
[ ] 27   INVITE     2000@192.168.1.105:5060 *72@192.168.1.105:5060  9     192.168.1.169:5060   192.168.1.105:5060

Esc Quit   Enter Show   Space Select   F1 Help   F2 Save   F3 Search   F4 Extended   F5 Clear   F7 Filter   F8 Settings   F9 Clear wit
```

Image 18.3.4 - SNGrep SIP messages flow viewer with a filter applied.

This is a great tool for you to troubleshoot the SIP/PJSIP events on your VitalPBX. To quit the SNGrep interface press the **Escape** key on your keyboard and then select **Exit** by pressing the **Enter** key on your keyboard.

Section 18, Lesson 4 - VitalPBX Commands

With VitalPBX, we have created multiple commands you can run from the Linux CLI, that can help you manage and stabilize your VitalPBX installation. Here are all the VitalPBX Commands you can run.

To see the list of available commands, run the following command.

```
root@debian:~# vitalpbx --help
```

To reset the password for any Web UI user, run the following command.

```
root@debian:~# vitalpbx reset-pwd [username]
```

Where [username] is the user you wish to change the password.

To rebuild the VitalPBX database, run the following command.

```
root@debian:~# vitalpbx build-db
```

This will execute a series of scripts to build the VitalPBX database.

To generate the Asterisk configurations and rebuild the Asterisk database for the main tenant only, we run the following command.

```
root@debian:~# vitalpbx gen-conf
```

To generate the Asterisk configurations and rebuild the Asterisk database for all tenants, we run the following command.

```
root@debian:~# vitalpbx fully-gen-conf
```

To check for any issues regarding directories/files and their permissions and owners, we run the following command.

```
root@debian:~# vitalpbx check-integrity
```

To apply the changes made on the Firewall module on VitalPBX, we run the following command.

```
root@debian:~# vitalpbx apply-firewall
```

You can manually restore from a backup file you have uploaded manually by running the following command.

```
root@debian:~# vitalpbx restore-backup path/to/file.tar
```

This is especially useful when the backup file is too big to upload from the web interface. You can upload it manually using an SCP or SFTP application to your VitalPBX, then run the command and specify the path to the backup file to restore the system from it.

To reset the SSL certificates and disable the force HTTPS option, run the following command.

```
root@debian:~# vitalpbx reset-apache-conf
```

This will help you in case your SSL certificate expires or you don't have access to your server via HTTPS.

To optimize the performance of MariaDB/MySQL, you can run the following command.

```
root@debian:~# vitalpbx optimize-mariadb
```

Keep in mind that this will restart MariaDB.

To optimize the Apache configurations on your VitalPBX, run the following command.

```
root@debian:~# vitalpbx optimize-apache
```

To disable 2FA for a user, run the following command.

```
root@debian:~# vitalpbx disable-2FA [username or email address]
```

With this, you can disable 2FA for a specific user in case they lose access to their 2FA application. You can use either their username or email address.

To update VitalPBX and all its dependencies, run the following command.

```
root@debian:~# vitalpbx update
```

With this, you can control your VitalPBX installation from the command line and troubleshoot any issues you may have.

Conclusion

With these tips and tricks, you can monitor and manage your VitalPBX instance and troubleshoot for any errors. By following this manual, you know where to tackle any issue if you find what is happening during the situation.

GRAND CONCLUSION

Congratulations! With this, we come to an end with the VitalPBX Complete Guide and Manual. The manual covered everything you need to know about VitalPBX, how to configure it, and have your VitalPBX completely set up. We appreciate you choosing VitalPBX for your business endeavors, and we hope to hear more about you.

If you ever have any comments or questions, send us an email to sales@vitalpbx.com and we will gladly guide you to the right place.

APPENDIX

Resources

- **VitalPBX Wiki -** https://wiki.vitalpbx.com/
- **VoIP Info -** https://voip-info.org
- **Asterisk -** https://asterisk.org
- **Asterisk AMI -** https://docs.asterisk.org/Configuration/Interfaces/Asterisk-Manager-Interface-AMI/
- **LINBIT creators of DRBD -** linbit.com/en/
- **Pacemaker -** clusterlabs.org/pacemaker
- **VitalPBX Github -** https://github.com/VitalPBX
- **strftime documentation -** https://man7.org/linux/man-pages/man3/strftime.3.html
- **VitalPBX Commercial FAQ -** https://wiki.vitalpbx.com/wiki-category/commercial-faq/
- **VitalPBX Technical FAQ -** https://wiki.vitalpbx.com/wiki-category/faq/