Install and Set-Up for a Virtual Server with 3 instances:

- **IP Phone (Wazo)**
- **CRM (SuiteCRM)**
- **Linux (LinuxMint)**

(RC Personal)

Components
Debian
Proxmox VE (virtual environment)
Wazo - and all associated packages
SuiteCRM - and all associated packages
LinuxMint - and all associated packages
Automated Backup
Automated Daily Status Reporting

Instructions current as of:
08 July 2020

Introduction

1) Disclaimer

a) If this document gets distributed (which is fine by me, so long as it is distributed for free) it should be done on the basis of trying to help as opposed to presenting as if this is the ultimate authority.

b) And in this litigious world, allow me now to clearly state that:

- What is described in this document may not be suitable for your individual configuration.
- While I have taken due care to accurately document my actions, and all that I have outlined worked for me, this document may contain errors, typographical mistakes, omissions and even misguidance that may require a lot of extra corrections.
- I do not assume, neither will I accept any responsibility for any losses incurred due to actions or inaction conducted as a result of methods or advice found in this document.

2) Credits

a) Thank you to the entire open source community, including the developers of

- Debian
- Proxmox
- Wazo
- SuiteCRM
- LinuxMint
- And many other packages and utilities used in this installation

b) This is a fun way to learn and to create something truly valuable. The spirit of innovation for the sake of innovation, and collaboration for all the right reasons, tells me the spirit of community, curiosity and creativity is still alive and well.

c) Although all elements shown are available for free download, if you do intend to use this for purposes other than just learning and enjoyment, I do encourage you to make donations to the developers so they can be rewarded for what they have done and encouraged to continue supporting their creations.

3) System Components

a) This is a long, comprehensive document, attempting to make a complicate technology more accessible to a reasonably competent technology person. It contains fullsome explanations of some technologies (ie iptables) that the reader may already understand, in which case, that section can be skimmed/skipped and the specific actions followed.

b) I have set this up as a virtualized server. If you want a dedicated server for Wazo PBX - a good idea if you have any real load on your PBX, but not needed for a small office / home setup - then just start with the Debian installation and add the Wazo installation, starting at heading e). Ignore the Proxmox, SuiteCRM and Linux Mint instructions. In my case, I use Wazo for a home/office PBX so I have a limited number of concurrent users and a virtualized environment works fine for the PBX.

c) See the diagram below in the "System Architecture" section for a graphic representation of the components. The basic components to the system I put in place are:

- Server Hardware
  - ◊ Because I have used a Virtual server setup, more is better. More RAM, more CPU power, more Disc Storage, … . About the only thing you do not really have to invest too much in is the Graphics Processing Unit, since I am using this in a headless (no monitor) setup, with remote access only.
  - ◊ I use Linux Mint as a secondary operating system when I need to test something (hoping to convert to it as my primary system once I find a good replacement for Outlook) or use a particular functionality not available in other OSs (yes, you heard me Windows, Linux can do things you cannot). If you intend to use Linux Mint as a primary desktop, I'd suggest a dedicated or dual-boot setup - with a decent GPU - for speed of user interface.
- Proxmox VE
  - ◊ Operates on Debian Linux Operating System
- Wazo PBX
  - ◊ Operates on Debian Linux Operating System
  - ◊ Installs and configures Nginx Web Server
  - ◊ Installs and configures Postgres database (and others)
  - ◊ Uses Asterisk Soft PBX
  - ◊ I installed Webmin to allow easier administration of the Linux OS
  - ◊ (and many other components used in the system operation)
- SuiteCRM CRM
  - ◊ Operates on Debian Linux Operating System
  - ◊ Built on Apache Web Server
  - ◊ Requires MySQL database (MariaDb used)
- LinuxMint Desktop Operating System
  - ◊ Built on Ubuntu base (which is based on Debian Operating System)

d) The tools used on the Windows machine to remotely configure and manage the system are:

- Putty (v0.673 from Chiark (www.chiark.greenend.org.uk)) to provide SSH connection directly to the Linux OS for Proxmox, Wazo and SuiteCRM - at whatever (root, user) level is used to sign-on. To use the terminal in LinuxMint, use UltraVNC to access LinuxMint then launch the terminal.
- Web Browser (Mozilla Firefox (www.mozilla.com)) to access the Proxmox, SuiteCRM and WebMin interfaces
- UltraVNC client to access the Linux Mint desktop via VNC protocols. With Proxmox, the SPICE protocol is also available for use by the Windows client RemoteViewer as a remote desktop. SPICE does provide faster screen refresh and can pass audio so has advantages over VNC protocols, but each use requires generation of a one-time-authentication ticket, so I use it for extended use of Linux Mint and UltraVNC for quick use of Linux Mint.

e) After the initial installation and configuration, the Server can run without User I/O devices like Keyboard, Mouse and Monitor, since all aspects of System configuration and management can be

accomplished with the remote access tools described above. I put the hardware in a secure location and only access it via remote access tools.

f) The handsets used with the system can be either Soft Phones (Zoiper (www.zoiper.com), 3CXPhone (www.3cx.com), CounterPath X-Lite or Bria (www.counterpath.com)), or IP Phones (Aastra 6753i, 6757i (www.aastra.com)) or Analog Phones hooked up via an ATA (Analog Terminal Adapter) (Cisco SPA2102 (www.cisco.com)).

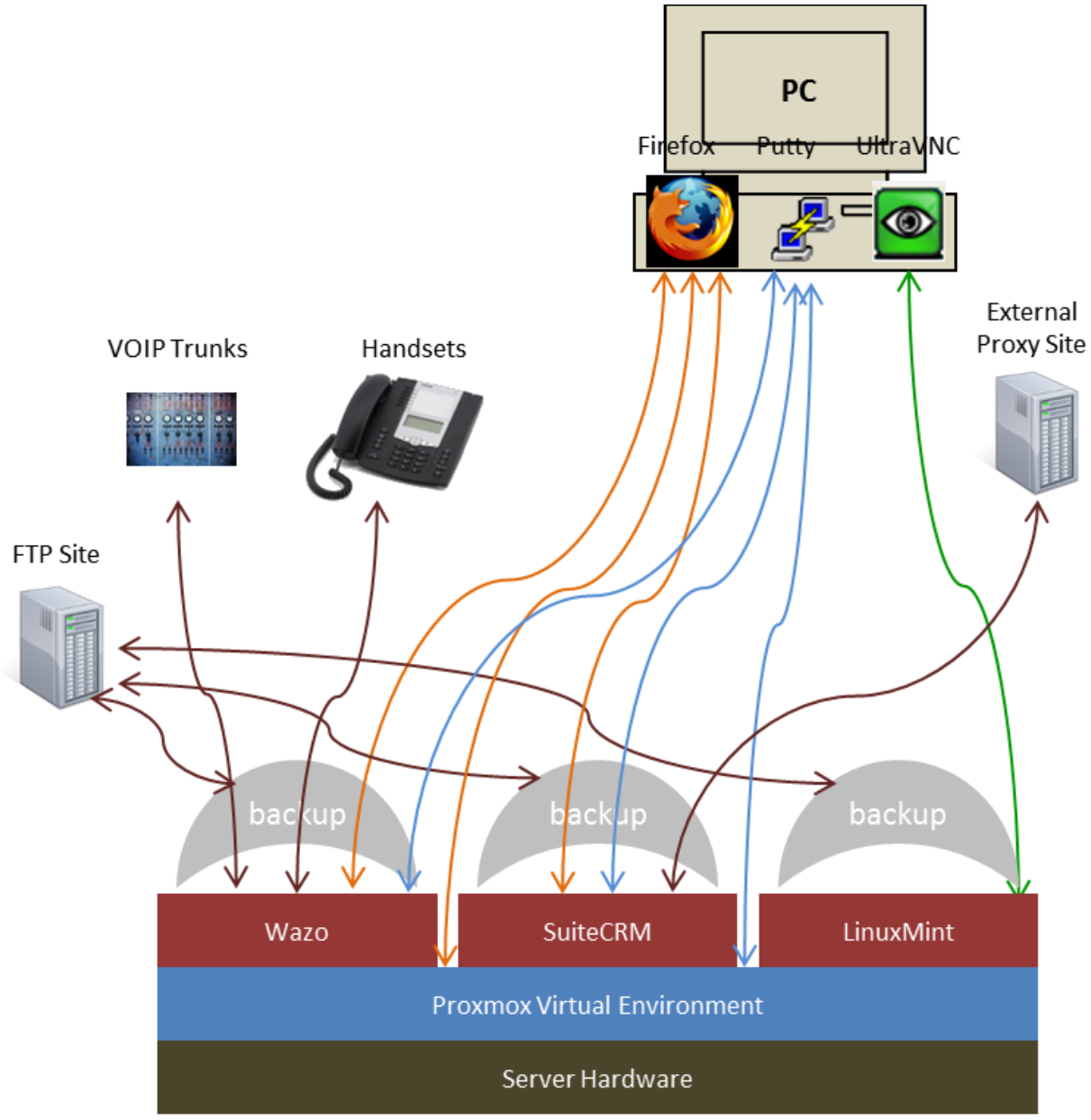
g) The system is connected to two Telecom Service Providers for VOIP trunks (Unlimitel (www.unlimitel.ca) and Vitelity (www.vitelity.com) via an Internet Service Provider (Teksavy (www.teksavvy.com))

h) The system is backed up nightly with a rotating 7-day and 12-month schedule to a remote ftp site.

4) System Architecture

a) This virtualized system is setup with 3 instances : a VOIP Phone system (wazoPBX), a Customer Relationships Management system (SuiteCRM) and a Linux Desktop system (LinuxMint) all operating under a virtual host (Proxmox) on one hardware platform. If you wish to only install one of the instances, just go directly to the "Install …" instructions and you can install just that system on a dedicated hardware platform.

b)


PC
Firefox
Putty
UltraVNC
VOIP Trunks
Handsets
External
Proxy Site
FTP Site
backup
backup
backup
Wazo
SuiteCRM
LinuxMint
Proxmox Virtual Environment
Server Hardware

5) **Formatting proocols for this document**

a) Wherever you see {} , this represents a placeholder

- You are to replace the {} and the contents of the {} with whatever is appropriate for your installation, unless the instructions explicitly call for your to handle this differently

b) Wherever you see [] , this represents commentary/instructions for the command given before the brackets

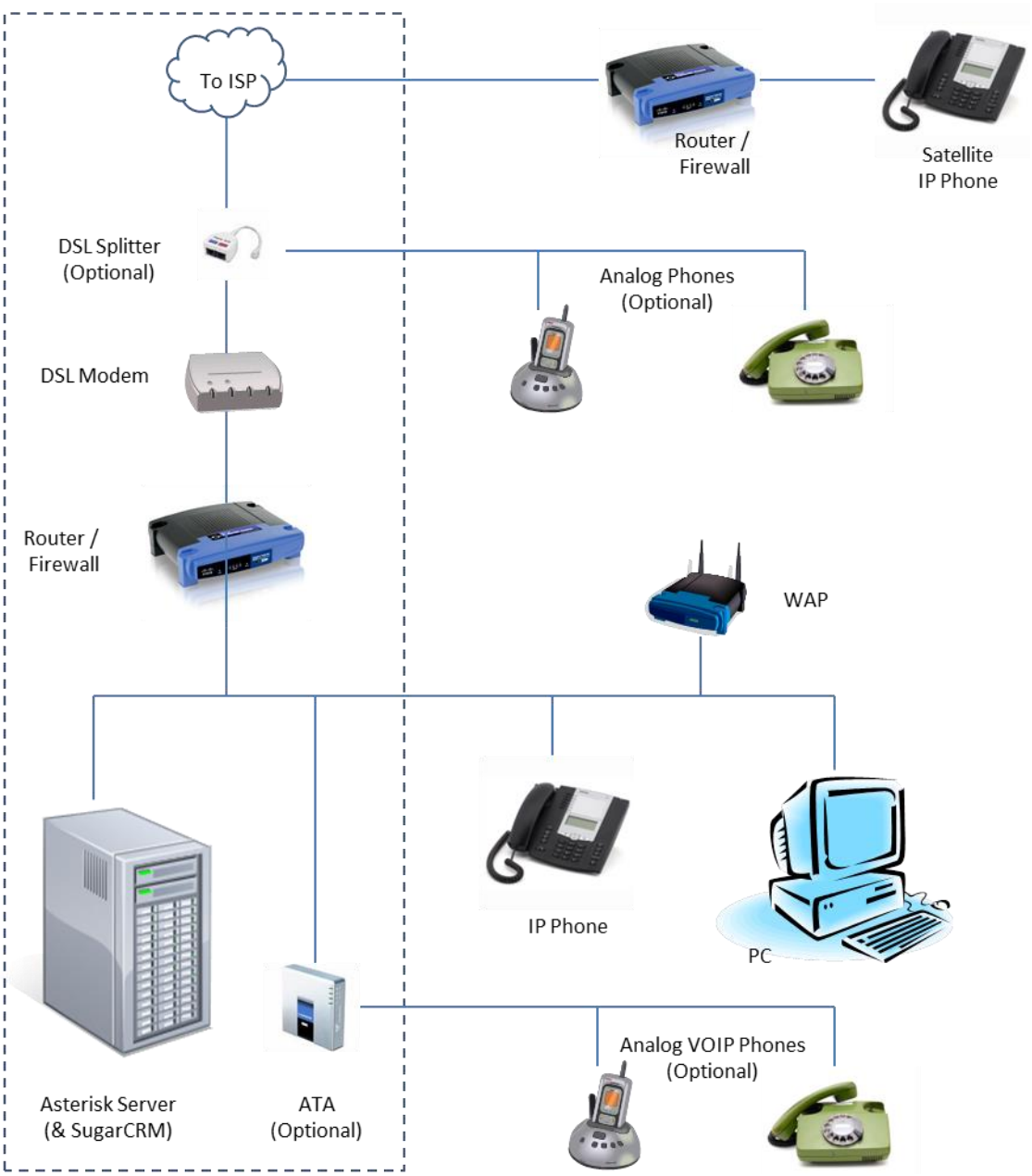
- You are NOT to include the brackets or the content of the brackets in your commands, unless the instructions explicitly call for your to include them

6) **Personalized Settings**

a) I create an Appendix (1) for a place where UserIDs, Passwords, IP_Addresses and other information used in my system can be kept. Appendix 1 is not included in the document I share publicly.

7) **Phone System Hardware Set-Up**

a)



b) Be very careful with a Satellite IP Phone outside your local LAN (like Aastra, but the same caveats apply for other IP phones). Make sure it is behind a Firewall, not directly connected to the internet with a public IP address. It is possible to configure the phone with a direct connection to the internet using a public IP address, but doing so would expose the web interface built into the phone (in my case Aastra) to the outside world. Most IP phones do not have robust security - simply a UserID and Password so security is low and a hacker could gain access with relative ease, thereby having a back-door into your PBX. Never a good thing. Use a more robust firewall/router and put the Satellite IP phone behind that. When you do this, make sure the Primary DNS in the router feeding the server is set to the PBX server IP (even if there is no DNS running on the server) and the Secondary DNS (and Tertiary DNS if you want) are set to the "real" DNS, which in my case was OpenDNS. This way, on a server reboot or other temporary outage, when the connection is restored, the phone comes back from "No Service" without restarting the phone. You could also have the Wazo server be the DNS for the IP phones but I did not.

8) Maintenance Summary

a) Backups

- Proxmox allows you to create backups of the entire VM instance which you can then store offline if desired. I do this once I have a fully-functional system but do not use this to store daily backups since the file transfer each night would store everything, including things that do not change, and be much larger than needed.
- Webmin provides a facility to automate backups of selected directories. For the Proxmox and CRM applications where I installed Webmin, you could do backups directly from Webmin, so long as you know the correct directories and files to backup. I did not use Werbmin for my backups, preferring to use my own scripted backup with added functionality.
- The setup shown here creates a daily backup of selected directories and databases for each instance.
- For WazoPBX, there are options to create images of the installation and you could create a cluster which gives you hot-standby capability. With my minimal needs, I did not use a cluster or image for backup. I created a cold standby server, with the initial configuration matched, to allow a quick swap-out and return to operation in the event of a system crash.
  - ◊ Wazo creates daily backups of the database and data so the Cron hobs and scripts shown just send the backups to a remote ftp server and/or second (archive) hard drive
  - ◊ Restore is a complicated process but described in the Wazo backup section
- SuiteCRM Backup has the web site files and the MySQL database backed up, so recovery for the SuiteCRM installation would only require the admin to use the remote backup to restore the files in both the web directory and the MySQL database.
- Linux Mint has only the data files backed up.

Installation and Set-Up Instructions

1) Configuring the Router

a) I used, for my Home installation, an Asus router with Asus-Merlin firmware

b) This allows QoS setup as needed. In my installation, I just added bandwidth so relieved the need for QoS to manage the phone call quality.

c) Make sure that "UDP timeout" is set high enough to allow network issues to be resolved

- In dd-WRT router it is in Advanced -> Management
- In Asus-Merlin Router it is in Tools -> Other Settings
  - ◊ UDP Timeout: Assured: 180 [which is fine since Asterisk wants at least 60]
  - ◊ UDP Timeout: Unreplied: 30 [which is fines since Asterisk wants at least 10]

d) Equip the router to direct tftpboot requests to the PBX server

- This is only required if you are using the DNS server in the Wazo setup - which I am not - so that you can use autoprovisioning. The option is to manually restart each phone after reconfiguring them - which I do.
- If you do want to use tftpboot, you need to add the following line to the dnsmasq.conf file on the router

dhcp-option=66,{IP_Address_of_server}

- You can do this by manually editing the dnsmasq.conf file of the router but that is dangerous since it may get overwritten and you may mess something else up
- You can do it via a postconf script, which adds the line every reboot, but if for whatever reason the script hangs, the router will never boot.
- We choose to use a conf.add setting so it is added to the dnsmasq configuration automatically on each reboot of the router.
- For the Asus Merlin firmware, you do this in two steps:
  - ◊ Enable the dnsmasq directories in the router
    - ♦ Asus -> Advanced -> Administration -> System -> Persistent JFFS2 partition

      IF you have never created a JFFS directory before

      Format JFFS partition at next boot: Yes

      And then, after the next boot, come in and change this to No

      Enable JFFS custom scripts and configs: Yes
  - ◊ Add the entry to the JFFS directory
    - ♦ SSH into the router using the Router's admin user and password

      find / -iname jffs

      You will get two identified

      /jffs

      /tmp/var/notice/jffs

      The one we are interested in is the one at /jffs

      ls -al /jffs

      One of the directories is configs

      ls -al /jffs/configs

      If a dnsmasq entry has NOT previously been created, this directory will be empty. In this case, you will want to create and populate a file called dnsmasq.conf.add

      nano /jffs/configs/dnsmasq.conf.add

      dhcp-option=66,{IP_Address_of_server}

      Make sure, with chmod and chown that the permissions are set properly on dnsmasq.conf.add

      If you ever want to remove the dnsmaq setting, just delete the dnsmasq.conf.add file and reboot the router
  - ◊ Reboot the router
  - ◊ The router will now add, to the end of the dnsmasq.conf file the line shown above and the VOIP devices that use tftpboot to get their configurations will be directed to the correct IP address (of the server)
  - ◊ If you do want to see the enrire dnsmasq.conf file

    find / -iname dnsmasq.conf

- and for the Asus router, you will see it is stored at

  /tmp/etc/dnsmasq.conf

- with dhcp-option=66,{IP_Address_of_server} as the last line of the file

2) Installing Remote Access Windows Clients/Tools

a) Putty

- Web site http://www.chiark.greenend.org.uk/
- From your PC browser, go to http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html and download the file to your PC
- Double Click on the file and Run (yes, it is that simple)(you may want to change some of the colour configurations, but you can use it as is)
- Enter the <instance_IP_Address> and click Open
- Setup so Numeric Keypad works
  - The default setup has the numeric kepad used to send control characters instead of numbers to the remote server. This can cause (has caused) issues when, for example, editing a file in nano, and without thinking, use the keypad to enter numbers. Some unexpected and frustrating results can occur.
  - To return the keypad to being a numeric entry tool, use the Putty setup menu at the left side when Putty is launched (or left-click top-left icon during a session and select "Change settings")
  - To change the setup for an existing profile, click the Profile name, click Load and:

    Terminal -> Features

    Disable application keypad mode: Check

    Session

    Save

- To enable use of key-pair authentication
  - Identify the User name to be used for login
    - Putty -> Connection -> Data (select Data)
    - In the field "Auto-login username" enter the remote server Username (NOT root)
  - Identify the private key to be used for digital authentication
    - Putty -> Connection -> SSH -> Auth (Select Auth, not click on the + sign)
    - Browse to the location of the private key associated with the public key you will upload to the remote server
- To save changes so they are permanent, not just for the current session, before opening a connection, after making the changes above, click on the Session menu item, select the profile you want to save the settings under (or enter a new Profile name) and click Save.

b) Firefox

- From http://www.mozilla.org/ get Mozilla Firefox onto your PC
- Go to <instance_IP_Address>:{port_number_if_specified} to use the Web GUI for Proxmox, SuiteCRM or WazoPBX. Some require https.

c) UltraVNC

- From https://www.uvnc.com (be careful, there are phishing sites out there like ultravnc.com) download the UltraVNC client/viewer. There are other clients (like x11vnc) but I like the simplicity of UltraVNC.

d) Remote Viewer (SPICE protocols)

- From https://www.spice-space.org download the Windows version of the Remote Viewer client.

3) Patch alert for servers previously loaded with Windows

a) If the server you are planning to use has previously been used by an older Windows (still true to Windows 7) and you are going to load it with Linux, BEFORE you strip Windows off, go to the Network settings and disable the Shutdown Wake-on-LAN setting for the eth0 connector. It turns out, that since Windows does not handle Wake-on-LAN, it turns off the NIC when Windows shuts down to prevent anyone from trying to use Wake-on-LAN. If you leave the NIC disabled like this and load Linux, Linux does not know how to enable the NIC chip so it stays dormant.

4) Installing Proxmox

a) Proxmox is an open source, free (for personal use) virtualization technology allowing you to use one piece of hardware as multiple servers, with each server instance having its own IP address and appearing to be the only thing on its server. To use this, you need a relatively good hardware (RAM, CPU, Storage) but it is less expensive, more convenient and easier to maintain than multiple pieces of hardware. Proxmox has tools for backuping up and restoring images so it can also be useful for trying new things and going back to a working version when you crash the "server" (not that such a thing ever happens, right?)

b) If you want more information than is in this document (optional), see the following pages for varying levels of detail on installing Proxmox

https://pve.proxmox.com/wiki/Installation

https://pve.proxmox.com/pve-docs/chapter-pve-installation.html

https://pve.proxmox.com/pve-docs/pve-admin-guide.html

https://pve.proxmox.com/pve-docs/

https://pve.proxmox.com/pve-docs/chapter-pvesm.html#_storage_features_6

https://forum.proxmox.com/threads/proxmox-beginner-tutorial-how-to-set-up-your-first-virtual-machine-on-a-secondary-hard-disk.59559/

c) Install Proxmox, specifying the types of instances that will be virtualized, then install each instance.

- I created two Proxmox installations, one for development / cold standby and one for use in production.
- Each server and each instance of each server has a admin-level and non-admin-level user defined
  - ◊ Admin
    - ♦ Admin User: root

        - Admin Password: {root-level-password}
    - Non-Admin
        - User Name: {UserName}
        - User Password: {UserPassword}
- The host, WazoPBX and SuiteCRM LVs have only key-pair authentication for remote SSH access
- The LinuxMint VM has remote desktop access via UltraVNC viewer or Remote Viewer
- Development Server
    - Proxmox VE at IP Address : {Proxmox_IP_ Dev}
    - Wazo PBX at IP Address : {Wazo_IP_ Dev}
    - SuiteCRM at IP Address : {SuiteCRM_IP_ Dev }
    - LinuxMint at IP Address : {LinuxMint_IP_ Dev }
- Production Server
    - Proxmox VE at IP Address : {Proxmox_IP_Prod}
    - Wazo PBX at IP Address : {Wazo_IP_Prod }
    - SuiteCRM at IP Address : {SuiteCRM_IP_Prod }
    - LinuxMint at IP Address : {LinuxMint_IP_Prod }

d) Configure the BIOS of the server (press {Del} key while booting on most servers) to allow Proxmox to install

- Boot sequence to Optical drive first (In boot sequence sttings)
- Virtualization Technology (Intel VT) to Enabled (In CPU settings)

e) Hook up a keyboard and mouse to the server for the initial installation/configuration of Proxmox. Thereafter, remote access can handle all that is required.

- You may need to try different (USB or PS/2) keyboards and mouse types to find ones that work with your hardware and the Proxmox install disk drivers.
- Note: It is possible to only use the keyboard to progress through the installation wizard. Use the cursor control keys to move amongst selections shown and the Enter key to choose a selection. Buttons can be pressed by pressing down the ALT key, combined with the underlined character from the respective Button. For example, ALT + N to press a Next button.

f) Download the Proxmox VE ISO, burn it to disc and restart the server with the disc in the optical drive

- Main Proxmox site: https://www.proxmox.com/en/proxmox-ve
- Download site: https://www.proxmox.com/en/downloads/category/iso-images-pve
    - I downloaded the v6.1-1 ISO which was uploaded on 4Dec2019
        - https://www.proxmox.com/de/downloads/item/proxmox-ve-6-1-iso-installer

          Which WILL download 6.1-1 in spite of the URL identifier

OR you culd download from a non-https site with

http://download.proxmox.com/iso/proxmox-ve_6.1-1.iso

- The filename will be proxmox-ve_6.1-1.iso
- The 6.1-1 ISO has a SHA256SUM of 95434b81cf74fdb5f8ac3e341c55293e10dafd1a75d1be45668ccb25f7d3c93c

◊ This will install a customized version of Debian 10 (Buster) - 64bit - with the Proxmox software and other required packages, all properly configured to interact properly

- This will erase any current data on the hard drive onto which you install Proxmox

- When you install Proxmox, you will be asked to specify what kind of instances you want running under Proxmox (Containers, Virtual Environment, …) and, if you want, and are doing multiple repeat installations after the first one, you will be able to use templates to aid in the proper installation and configuration of the instances. I did NOT do use the existing templates since I was only installing two servers (Development/ColdStandby and Production) and wanted to manually configure each one, but am including for reference the instructions for template retrieval.

◊ From the command line at the Proxmox host:

su

[to get to the root-level user]

{root-level-password}

pveam update

[to update the system list of available templates in the repository]

pveam available

[to show the available templates]

pveam download local {name_of_template_you_want_to_download}

pveam download local debian-10.0-standard_10.0-1_amd64.tar.gz

[for server instances]

pveam download local ubuntu-18.04-standard_18.04.1-1_amd64.tar.gz

[for LinuxMint (Ubuntu-based) desktop instances]

pveam list local

[to list the templates now available locally, which you could copy to other servers]

pveam help

[if in doubt as to pveam options available to you]

◊ Like I said, I did not use this, but the instructions are here if you choose to do so

- With the server now booted to the Proxmox Installation Disk GUI, start the installation

◊ If this is your first use of the server, you may want to do a memory test

◊ Select Install Proxmox VE and press the Enter key to start the installation

◊ Agree to the License Agremeent

- ◊ Select the target Hard Disk
  - ♦ I just accepted Default - only had one for test system
  - ♦ If you want to create additional configuration settings for partitions, click Options (I did not)
- ◊ Configuration Options - Page 1
  - ♦ {Country}
  - ♦ {Time Zone}
  - ♦ {Keyboard Layout}
- ◊ Configuration Options - Page 2
  - ♦ {root-level Password}
  - ♦ {admin_email}
- ◊ Configuration Options - Page 3
  - ♦ NIC

    {NIC_Dev}

    {NIC_Prod}
  - ♦ Hostname

    {HostnameDev}

    {HostnameProd}
  - ♦ IP Address of Proxmox System

    {Proxmox_IP_ Dev}

    {Proxmox_IP_ Prod}
  - ♦ {Netmask}
  - ♦ {Gateway}
  - ♦ {DNS_Server1}
- ◊ Confirm all the settings and wait for the installation to complete
- ◊ Reboot the server, removiong the Installation Disc from the Optical Drive so the server will now boot from its own hard drive

- Configuration of the Server in preparation for the installation of the three instances
  - ◊ Web GUI
    - ♦ Once the server has competed its reboot, point your browser to {Proxmox_IP__Dev}:8006 or {Proxmox_IP__Prod}:8006 , login with root and and {root-level Password} and proceed to configuration
    - ♦ You may get a security warning that the certificate is invalid. Ignore this and proceed.
    - ♦ You will get a warning that you do not have a subscription. This is normal. We did not purchase an Enterprise subscription since this is a personal-use installation of Proxmox, something that is fully legal and supported by Proxmox, but they still nag you to get a

subscription every time you login to their GUI.

◊ SSH Client

♦ Test the server for remote login by pointing an SSH client (like Putty) to the server IP address and enter:

User: root

Password: {root-level Password}

The first time you ssh in with the remote client, it will ask you to accept the certificate; Do so.

♦ There is a Console/Shell/Terminal window available from within the Proxmox web GUI and you will need to use it for the initial install of an OS into the LV, but after that, I prefer the user interface and copy/paste capabilities of Putty so I use it for subsequent work. When you want to use the Proxmox terminal:

Click on the triangle tree in the left column to the left of Datacenter to expand the tree

In the left column, click on (select) the {hostname}

In the right column, click on (select) the {name of the LV}

Click on Shell [Menu top right]

◊ Run the hardware capability check from the CLI using command "pveperf" (without the quotes)

♦ Check the server's specs against the hardware recommendations laid out by Proxmox at https://www.proxmox.com/en/proxmox-ve/requirements

♦ For Production systems (evaluation/development systems require less)

Intel EMT64 or AMD64 with Intel VT/AMD-V CPU flag.

Memory, minimum 2 GB for OS and Proxmox VE services. Plus designated memory for guests. For Ceph or ZFS additional memory is required, approximately 1 GB memory for every TB used storage.

Fast and redundant storage, best results with SSD disks.

OS storage: Hardware RAID with batteries protected write cache ("BBU") or non-RAID with ZFS and SSD cache.

VM storage: For local storage use a hardware RAID with battery backed write cache (BBU) or non-RAID for ZFS. Neither ZFS nor Ceph are compatible with a hardware RAID controller. Shared and distributed storage is also possible.

Redundant Gbit NICs, additional NICs depending on the preferred storage technology and cluster setup - 10 Gbit and higher is also supported.

For PCI(e) passthrough a CPU with VT-d/AMD-d CPU flag is needed.

♦ For Development / evaluation systems

CPU: 64bit (Intel EMT64 or AMD64)

Intel VT/AMD-V capable CPU/Mainboard (for KVM Full Virtualization support)

Minimum 1 GB RAM

Hard drive

One NIC

◊ Check the Hard Drive

- To specifically check the health of your hard disk - assuming your hard disk has SMART firmware

  smartctl -a /dev/sdX (where X is the device number (eg a) and sd is the type of device (may be different for you)

- If you get the reply "SMART support is: Disabled", enable it and then run the above command by first running

  smartctl -s on /dev/sdX

◊ Check you have the right OS installation (Debian 10 = Buster) by using the CLI comand "cat /etc/*-release" (without the quotes)

◊ Update the repository list and upgrade the system

- If not purchasing a subscription, you need to change the default Proxmox repository
- The default repository setup assumes a subscription and will not work without one
- Disable the Proxmox enterprise repository (The free Proxmox repository will be enabled below)

  nano /etc/apt/sources.list.d/pve-enterprise.list

  Change

  deb https://enterprise.proxmox.com/debian/pve buster pve-enterprise

  to

  # deb https://enterprise.proxmox.com/debian/pve buster pve-enterprise

- Update the package list

  apt update

  (do NOT upgrade at this point)

- Disable the default debian repositories and enable the free Proxmox repository (assuming you did NOT subscrbe to the support offerings)

  nano /etc/apt/sources.list

  Comment out the existing repositories

  # deb http://ftp.ca.debian.org/debian buster main contrib

  # deb http://ftp.ca.debian.org/debian buster-updates main contrib

  # security updates

  # deb http://security.debian.org buster/updates main contrib

  Add

  # PVE pve-no-subscription repository provided by proxmox.com,

  # NOT recommended for production use

  deb http://download.proxmox.com/debian/pve buster pve-no-subscription

And while in there, change the repository list to a allow non-free libraries

And ONE of the following

Direct from Debian

Standard Debian repositories, including non-free repositories

deb http://deb.debian.org/debian buster main contrib non-free

deb-src http://deb.debian.org/debian buster main contrib non-free

deb http://deb.debian.org/debian-security/ buster/updates main contrib non-free

deb-src http://deb.debian.org/debian-security/ buster/updates main contrib non-free

deb http://deb.debian.org/debian buster-updates main contrib non-free

deb-src http://deb.debian.org/debian buster-updates main contrib non-free

From a local Mirror (I used University of Waterloo; use one close to you)

Standard Debian repositories, including non-free repositories, from the UoW Mirror

deb http://mirror.csclub.uwaterloo.ca/debian/ buster main contrib non-free

deb-src http://mirror.csclub.uwaterloo.ca/debian/ buster main contrib non-free

deb http://mirror.csclub.uwaterloo.ca/debian/ buster-updates main contrib non-free

deb-src http://mirror.csclub.uwaterloo.ca/debian/ buster-updates main contrib non-free

deb http://mirror.csclub.uwaterloo.ca/debian-security/ buster/updates main contrib non-free

deb-src http://mirror.csclub.uwaterloo.ca/debian-security/ buster/updates main contrib non-free

- update and upgrade the host system

  apt update && apt -y dist-upgrade && apt autoremove && apt autoclean

- If you get "packages held back" try (in descending order of preference = safety

  apt-get --with-new-pkgs upgrade

  if that does not work, try

  apt install <name of held-back package>

◊ Change the Proxmox configuration to disable the high-availability (cluster) capability.

- We do not use this and it does impose considerable logging load on the server and the storage - Proxmox writes small updates to the disk every 3-4 seconds - so since we do not use it, we disable it.

- Why would you want to disable these services?

  If you are using SSDs, they write by page (large block of storage) and even if you only intend to put a small amount of data on them, they write a larger amount

This is called write amplification and it causes premature wear on larger sections of the SSD

SSDs also have a write limit (not so much a read limit) so they are better for constant reads, more so than constant writes

Even if you don't use SSDs, constant writing to a HDD will cause it more wear than without constant writing

So if you don't use High-Avaialbilty (Cluster) capability, stop the unneccesarry writing that comes with the capability

- To disable this activity, we disable two services:

  pve-ha-lrm

  pve-ha-crm

  To confirm they are running

  systemctl status pve-ha-crm

  systemctl status pve-ha-lrm

  You first stop the two services manually (but that will only last until the next reboot)

  systemctl stop pve-ha-crm

  systemctl stop pve-ha-lrm

  You now disable them from running in the future after reboots

  systemctl disable pve-ha-crm

  systemctl disable pve-ha-lrm

  If you did ever want to have them running again (I did not)

  systemctl enable pve-ha-crm

  systemctl start pve-ha-crm

  systemctl enable pve-ha-lrm

  systemctl start pve-ha-lrm

◊ Change the Proxmox configuration to disable the logging required for active software replication.

- I do not use hot-standby; I do daily backups for cold standby and do not use this feature so do not need the log entires every minute which are done to support hot standby.
- If you do use Proxmox's software replication, you want the normal replication schedule to continue, but if you do not use Proxmox's software replication (I do not) then you can reduce the entries to the system log which otherwise would get written every minute by the replication schedule and quickly fill the system log.
- To change the repilcation schedule (deleting it would potentially cause erros so why bother)

  systemctl edit --full pvesr.timer

  Change

  OnCalendar=minutely

to

OnCalendar=monthly

systemctl daemon-reload

◊ There are several additions I make to any Debian installation (some of which also apply to LinuxMint as a Debian derivative) which add to the security and/or the interactions with the installation. See the instructions below on "Installing Debian 10", and specifically the section on "Add my standard configurations to Debian" for details.

- ♦ Change to a fixed IP address (done during initial Proxmox installation)
- ♦ Change the repository list to add the non-free Debian repositories and to switch to a local mirror for faster access and downloads (done above)
- ♦ Update the package list and upgrade the system (apt update && apt -y dist-upgrade) (done above)
- ♦ Add a non-root user to facilitate key-pair SSH access instead of user/password authentication (and in this case for using the network file sharing capability (optional) explained later). This is usually part of the Debian installation, but with the Proxmox custom installation, it is not done so must be manually done here.

  adduser {UserName}

  {UserPassword}

  [accept default for remaining questions]

  Check the provisioning of the new user by listing all the users with

  cat /etc/passwd
- ♦ Set so remote access is not allowed with root (leaving only the new, non-root user with ability to remotely access.
- ♦ Configure a mail server to email out only (no inbound or relay)
- ♦ Install unattended-upgrades to have system automatically install security upgrades
- ♦ Install logwatch to have the system email a status report to the administrator every day
- ♦ Install scripts to email the administrator on login for both root and user

◊ If you just wanted to update and upgrade pve

pveupdate && pveupgrade

g) Provision the 3 instances which will be running on the server

- Reconfigure Proxmox to use normal (thick) volumes instead of thin volumes

◊ This is NOT required; you can use the system as is and it will work. However, if one volume goes out of control and starts filling its disk, it will take over space intended for another volume. By reverting to lvm, it restricts each volume to its own max space. In normal, IT-run, virtual servers, the administrator uses monitoring tools to see how full the disks are and manually adjusts allocation and/or adds additional disks to enable continued operation without interference between volumes. However, in my case, I did not want to be worrying about this so re-set the volumes to the old style which locks each volume into a fixed max space on the disk and prevents an issue in one LV from affecting the other LVs.

- ◊ Make sure when creating normal (thick) LVMs, that you leave free space on the disk (minfree) so the system can create snapshots for backing up and duplicating LVs
- ◊ For documentation on LVs see
  - ♦ Proxmox specific documentation

    https://pve.proxmox.com/wiki/Storage

    https://pve.proxmox.com/pve-docs/chapter-pvesm.html

    https://pve.proxmox.com/wiki/LVM2
  - ♦ Generic documentation

    https://www.tecmint.com/setup-thin-provisioning-volumes-in-lvm

    https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/logical_volume_manager_administration/index

    Yes, it is Fedora instructions but 90% is overlap and the explanations are good
  - ♦ Use "man lvmthin" from the Proxmox command-line-interface for details on the commands shown below
- ◊ To see the full disk contents, which include a boot partition called BIOS Boot, a system partition called EFI System and the Logical Volume Group for Proxmox called Linux LVM

  fdisk -l (l is lowercase L)
- ◊ To get a quick summary of the physical devices present and of the partitions on the device

  lsblk (l is lowercase L)

  blkid (l is lowercase L)
- ◊ Using the cli and the Proxmox WebGUI, edit the existing root directory to enable it to hold the items the lvm-thin has held
  - ♦ See the settings for the two volumes

    cat /etc/pve/storage.cfg

    dir: local

    path /var/lib/vz

    content iso,vztmpl,backup

    lvmthin: local-lvm

    thinpool data

    vgname pve

    content rootdir,images
  - ♦ Copy rootdir,images from the "lvmthin: local-data" content definition and paste (with a preceding ,) to the "dir: local" definition (the Proxmox webUI will not let you add them)

    nano /etc/pve/storage.cfg
  - ♦ Now delete the lvm-thin volume but do NOT use the web GUI to ADD the new lvm

Proxmox -> Select Datacenter - Select Storage -> Select local-lvm -> Click Remove

Confirm it is removed (from the Proxmox system) by again using

cat /etc/pve/storage.cfg

Note that by doing this, we have unmounted the LV so do NOT need to use the CLI command that would have otherwise been required before removing an LV

umount /dev/pve/data

However, this has NOT actually removed the LV, it has just removed the pointer to it from the Proxmox web GUI, so we still need to remove the old (thin) LV and create the new (thick) LVs

(Do NOT add a new (thick) LV with the web GUI; this is shown for reference only) If you use the Proxmox web GUI to create a new LV, the first LV you create will take all the available space left, and since we want to create 3 instances /LVs, this is NOT what we want.

Proxmox -> Select Datacenter - Select Storage -> Select local-lvm -> Click on Add -> LVM (not LVM-thin)

ID: local-host-data (free-form-text so use what you want here)

Base storage: laeave as is

Volume group: Select pve from the dropdown memu

Content: Disk image, Container

Click Add

So instead, do from cli

Get a list of the volumes which will include the one called data which is the thin one we will be removing.

lvs (or lvdisplay for a more detailed formatted output)

Get the detailed information on the volume we are about to remove

lvdisplay pve/data

To get a map of the names of the block devices (partitions in our case) on the disk which will include the partition with the LV we are about to remove

lsblk

(NOT used, but included for reference) To use the UUID instad of the name (replace {UUID} with the UUID fetched and {sda?} with the actual partition number number fetched)

Get the system identifier for each mount (dm-1. dm-2, ...)

ls -al /dev/mapper

Get the UUID

blkid [to get all partitions]

blkid /dev/{sda?} [to get UUID for a specific partition]

Get the actual mount path of the partition

findmnt -rn -S {UUID} -o TARGET

Get the mount pont of the LV

blkid -U {UUID}

Remove the old (thin) LV

umount /dev/pve/data

[NOT now required since this was unmounted by Removing the LV with the Proxmox web GUI]

lvchange -an pve/data

lvremove pve/data

Confirm the data LV in the pve VG is gone

pvs && vgs && lvs

or, for more detailed information

pvdisplay && vgdisplay && lvdisplay

or, to get the specifics for an LV, including its mount point

lvs -a -o +devices

- Now we create the new (non-thin) LVs we will be using on this server
  - ◊ For each LV, we will
    - ♦ Create the LV
    - ♦ Format the LV
    - ♦ Mount the LV
    - ♦ Add the LV to the Proxmox web GUI
  - ◊ Make sure to leave some free space on the hard drive (minfree) for the system to create snapshots (backups)
    - ♦ See the formula in Proxmox user manual for sizing of minfree but with our large drives (>128GB), the required minfree is 16GB, so make sure to leave at least that much unallocated on the disk
  - ◊ The sizing used in this document assumes a 1TB drive as the primary drive on which all things are stored. Adjust sizing and setup as required.
  - ◊ The Host, LVs and datashare storage have been allocated a total of 800GB (of the 1,000 GB drive) so we have left plenty of room for minfree and for potentially increasing the size of an LV if required
    - ♦ Proxmox 100GB
    - ♦ WazoPBX 150GB
    - ♦ SuiteCRM 200GB
    - ♦ LinuxMint 300GB
    - ♦ datashare 50GB
    - ♦ minfree (min 16GB)

- Setup the WazoPBX LV (non-thin) (establish the space and parameters)
  - ◊ Create the LV

    lvcreate --name wazopbx --size 150G pve
  - ◊ Format the LV

    mkfs.ext4 /dev/pve/wazopbx
  - ◊ Mount the LV

    mkdir -p /var/lib/vz/wazopbx [may already exist after formatting above]

    nano /etc/fstab

    /dev/pve/wazopbx /var/lib/vz/wazopbx ext4 defaults 0 1

    [place this line above the proc line]

    mount -a [to mount new devices without requiring a reboot]
  - ◊ Add the LV to the Proxmox web GUI

    nano /etc/pve/storage.cfg

    dir: lv-wazopbx

    path /var/lib/vz/wazopbx

    content vztmpl,rootdir,iso,images,backup

    maxfiles 0

    shared 0
- Setup the SuiteCRM LV (non-thin) (establish the space and parameters)
  - ◊ Create the LV

    lvcreate --name suitecrm --size 200G pve
  - ◊ Format the LV

    mkfs.ext4 /dev/pve/suitecrm
  - ◊ Mount the LV

    mkdir -p /var/lib/vz/suitecrm [may already exist after formatting above]

    nano /etc/fstab

    /dev/pve/suitecrm /var/lib/vz/suitecrm ext4 defaults 0 1

    [place this line above the proc line]

    mount -a [to mount new devices without requiring a reboot]
  - ◊ Add the LV to the Proxmox web GUI

    nano /etc/pve/storage.cfg

    dir: lv-suitecrm

    path /var/lib/vz/suitecrm

    content vztmpl,rootdir,iso,images,backup

    maxfiles 0

shared 0

- Setup the LinuxMint LV (non-thin) (establish the space and parameters)
  - ◊ Create the LV

    lvcreate --name linuxmint --size 300G pve
  - ◊ Format the LV

    mkfs.ext4 /dev/pve/linuxmint
  - ◊ Mount the LV

    mkdir -p /var/lib/vz/linuxmint [may already exist after formatting above]

    nano /etc/fstab

    /dev/pve/linuxmint /var/lib/vz/linuxmint ext4 defaults 0 1

    [place this line above the proc line]

    mount -a [to mount new devices without requiring a reboot]
  - ◊ Add the LV to the Proxmox web GUI

    nano /etc/pve/storage.cfg

    dir: lv-linuxmint

    path /var/lib/vz/linuxmint

    content vztmpl,rootdir,iso,images,backup

    maxfiles 0

    shared 0
- (Optional) Setup a (small) DataShare LV (non-thin) in the pve Volume Group which can be used to share data between instances or to locally store backups of some key files for each of the other instances
  - ◊ For example, you may want to create some graphics on the LinuxMint system and use the results on the SuiteCRM installation.
  - ◊ This capability will use the NFS (Network File System) capability enabled by installing Ceph

    nano /etc/apt/sources.list.d/ceph.list

    deb http://download.proxmox.com/debian/ceph-nautilus buster main

    apt update && apt dist-upgrade && apt autoremove && apt autoclean
  - ◊ Create the LV

    lvcreate --name datashare --size 50G pve
  - ◊ Format the LV with ext4

    mkfs.ext4 /dev/pve/datashare
  - ◊ Mount the LV
    - ♦ Because this is a shared LV using NFS to allow access by the other LVs, it requires a special, more complicated configuration
    - ♦ If the nfs volume is on the same disk / physical partition as the host boot volume (the

normal scenario), just follow the mounting instructions for the nfs mounting; do NOT also use linux mount instructions for the nfs drive on the same physical disk or you will generate error messages.

If the nfs volume is on a new partition or a different disk, follow the mounting instructions for linux filesystem mounting (like the examples below) AND the nfs monting instructions.

- This is the setup for the nfs server, which will be hosted on the Proxmox host, so the other LVs can all access it and share / archive data. So we need only follow the nfs mnount instructions, and we do NOT also follow the linux filesystem mount instructions.

  Once the nfs server is configured and running, each LV (client) must configure an nfs client to access the nfs server. These instructions will be shown in the other LVs (clients).

- For this installation, assume a one 1TB disk containing everything.

  mkdir -p /var/lib/vz/datashare

  chmod 755 /var/lib/vz/datashare [to enable all users to access (R&W) this LV]

  apt install nfs-kernel-server nfs-common [common is already installed but is left in for reference]

  rpcinfo -p

  To validate the nfs server is running on the host and accepting calls on port 2049. If not, enter

  modprobe nfs

  and run rpcinfo -p again

  nano /etc/exports [to add eligible clients to the server; change the IP Addressses to match the installation]

  # Enable the clients on this server to access the data Volume on the host in Volume Group pve with read and write access

  # The no-root-squash option allows root users from other VMs to access the host exported volume; default is to block this

  # Leave a space between /var/lib/vz/datashare and what follows and between entries of each SET of IP Address and brackets, but no space between the IP_Address and the left bracket that follows it

  # The following is all on one line

  /var/lib/vz/datashare {Wazo_IP}(rw,sync,no_subtree_check,no_root_squash) {SuiteCRM_IP}(rw,sync,no_subtree_check,no_root_squash) {LinuxMint_IP}(rw,sync,no_subtree_check,no_root_squash)

  exportfs -ra [to refresh the system's export settings]

  chown -R {UserName}:{UserName} /var/lib/vz/datashare

  [to enable remote access to this shared LV by non-admin (and root) users on other LVs ({UserName} will be added to each of those instances as well.)]

- For each LV/VM from which you want to access this datashare LV, you will add a nfs client and mount a nfs directory pointing to this datashare. Instructions for this are in

each LV.

◊ Add the LV to the Proxmox web GUI

♦ Do NOT do this for the shared LV. Doing so will generate error messages every few seconds.

- (Optional, required ONLY if you are using a second drive to hold an LV) Setup an LV on a second drive

◊ Establish the physical disk and volume group on the 2nd disk

fdisk -l [lowercase L - to confirm the 2nd disk is physically connected]

Note: on Proxmox 6.1 (based on Debian 10), the /sbin directory is not on the defined path so you must use the absolute path or will get "fdisk not found", so use

/sbin/fdisk -l

umount /dev/sdb [To unmount the drive (if mounted) and enable commands below]

wipefs -af /dev/sdb [to completely wipe all existing (including boot) partitions from the drive]

sgdisk -N 1 /dev/sdb [Create a single physical (gpt) partition on the disk]

gdisk /dev/sdb [To confirm the gpt partiion was created]

q <enter> [to quit gdisk]

pvcreate --zero y --yes -ff /dev/sdb1 [Create a physcal volume on the disk]

vgcreate disk2 /dev/sdb1 [To create a volume group called disk2]

◊ When this is done, then you can do as before to create the LV:

♦ Create the LV within the Volume Group just created

♦ Format the LV

♦ Mount the LV

This is all you would need to do if you were connecting to a second disk on which you had already created a physical partition, a volume group and logical volume(s)

Using nfs instructions above - for a second drive so you need to do both linux file system and nfs mounts

pvs && vgs && lvs [to confirm the above setup]

linux filesystem mount

mkdir -p /var/lib/vz/disk2-archives

mkdir -p /var/lib/vz/disk2-mintrchome

nano /etc/fstab

/dev/disk2/archives /var/lib/vz/disk2-archives ext4 defaults 0 1

/dev/disk2/mintrchome /var/lib/vz/disk2-mintrchome ext4 defaults 0 1

mount -a

nfs mount

nano /etc/exports [add to the bottom of the file]

Enable the clients on this server to access the logical Volumes on the second disk labelled Volume Group disk2 with read and write access

The no-root-squash option allows root users from other VMs to access the host exported volume; default is to block this

/var/lib/vz/disk2-archives
{Wazo_IP}(rw,sync,no_subtree_check,no_root_squash)
{SuiteCRM_IP}(rw,sync,no_subtree_check,no_root_squash)
{LinuxMint_IP}(rw,sync,no_subtree_check,no_root_squash)

/var/lib/vz/disk2-mintrchome
{Wazo_IP}(rw,sync,no_subtree_check,no_root_squash)
{SuiteCRM_IP}(rw,sync,no_subtree_check,no_root_squash)
{LinuxMint_IP}(rw,sync,no_subtree_check,no_root_squash)

exportfs -ra [to refresh the system's export settings]

chown -R {UserName}:{UserName} /var/lib/vz/disk2-archives

chown -R {UserName}:{UserName} /var/lib/vz/disk2-mintrchome

- Add the LVs to the Proxmox web GUI

  Because this is on a separate physical disk that is mounted in fstab, we can add this to the Proxmox GUI

  nano /etc/pve/storage.cfg [add the following entries below the existing ones]

  dir: disk2-archives

  path /var/lib/vz/disk2-archives

  content vztmpl,rootdir,iso,images,backup

  maxfiles 0

  shared 0

  dir: disk2-mintrchome

  path /var/lib/vz/disk2-mintrchome

  content vztmpl,rootdir,iso,images,backup

  maxfiles 0

  shared 0

- Check the status of the LVs with

  pvesm status

  lvs or lvdisplay

  lvs -a -o +devices

- (For future reference) Resizing a Logical Volume (LV)
  - If, at some point in the future you wish to resize a LV, you can do that so long as you have or can create extra space on the physical disk and the Volume Group.
  - The key to remember is that the actual file system resides inside a LV
    - If you want to increase the size of a LV, the sequence is

1) Increase the size of the LV so it can hold a larger file system

2) Increase the size of the file system to fit inside the LV

- If you want to decrease the size of the LV, it gets a little trickier, and the sequence is

1) Decrease the size of the file system to less than what the LV will be (allow buffer)

2) Decrease the size of the LV

3) Increase the size of the file system to fit inside the LV

◊ Because we are using a Network File System (nfs) which makes calls to the LVs, we need to disable the LVs first and reboot the system so nothing is accessing the LVs - otherwise you will get "LV is busy" error messages and you will not be able to do the resizing

nano /etc/fstab and comment out the lines mounting the LVs you want to resize

reboot the system so it unmounts the LVs and prevents the nfs from connecting them

When you are finished, remember to come back and re-edit /etc/fstab to uncomment the lines and reboot the system

◊ Use the lsblk command to get the mount point(s) of the LV(s) you want to change

lsblk

◊ Make sure there is room in the Volume Group (VG) to make the change(s) you want

vgdisplay

and check the amount available in the "Free PE / Size" value

◊ Make sure the LV is not active

lvchange -an {mount point of LV}

◊ Chaeck the file system to ensure it is clean - a corrupt file system could be irreparably lost with a resizing

e2fsck -fy {mount point of LV}

◊ If all looks good, to resize the LV:

- To increase the size

lvextend -L {# Gigabytes you want the new LV to be}G {mount point of LV}

This increases the size of the LV so it can hold a larger file system

resize2fs {mount point of LV}

The increases the file system to fill the available space in the LV

- To decrease the size

resize2fs {mount point of LV} {# Gigabytes = less than what the final LV will be)G

This reduces the file system to a size smaller than the final LV

Note the inclusion of a size number

Make sure the data currently in the LV does not exceed this number GB

lvreduce -L {# Gigabytes you want the new LV to be}G {mount point of LV}

This reduces the LV to the desired final size

resize2fs {mount point of LV}

The increases the file system to fill the available space in the LV

Note the absence of a size number

- ♦ Edit /etc/fstab to uncomment the mount lines
- ♦ Reboot the system

h) Optional Proxmox configurations

- Change the max number of backups that can be stored in the local storage area - default to 1
  - ◊ Datacenter -> Storage -> local - Edit
    - ♦ Max Backups: 20

i) (Optional) Upload ISOs of operating systems you intend to install on the VMs

- You can install the OS directly from a DVD installer disk or USB installer, but if your server does not have an optical disk or you prefer to not use it for installation of the OS, you can upload the installer ISO to Proxmox and have it use the uploaded ISO for installation on a VM
- Proxmox -> Expand the Datacenter menu (click on arrow to left of Datacenter) -> Click on the Local sub-menu -> in the column to the right, Click on Content -> Click on Upload
- Note that if you do upload an ISO and install from that, it leaves the ISO mounted and every time you restart the VM, it mounts the ISO. It is not necessary but if you want to, you can unmount it with umount (or Eject on Linux Mint desktop).

  blkid to identify mount point of ISO

  [likely /dev/sr0 where 0 in sr0 is number 0 not letter O]

  umount /dev/sr0

  ◊ The other option for having the ISO not appear on the desktop at all is to change the system setting so it does not look for the ISO on boot, but instead just treats the DVD drive as a normal drive with nothing in it (unless you put something there). To do this

  umount /dev/sr0 [0 is number 0 not letter O]

  Proxmox -> Select VM -> Select Hardware -> Double-Click CD/DVD Drive

  Change Radio button selection

  from

  Use CD/DVD disk image file (ISO)

  to

  Do not use any media

  and Click OK

5) Installing Debian 10

a) Proxmox installation includes its own Debian installation. For both WazoPBX and SuiteCRM, you will need to first install Debian 10 and then install the applications on top of Debian 10.Rather than duplicate the instructions in each section, we document the Debian 10 installation process here. Linux Mint is a complete installation on its own.

b) Download Debian installer ISOs from

https://www.debian.org/distrib

- Select the "complete installation image" for the current (10.3 at the time of this document) Debian stable version
- In my case, for Debian 10 (Buster) DVD image for AMD64 I downloaded three ISOs

  https://cdimage.debian.org/debian-cd/current/amd64/iso-dvd/debian-10.3.0-amd64-DVD-1.iso

  https://cdimage.debian.org/debian-cd/current/amd64/iso-dvd/debian-10.3.0-amd64-DVD-2.iso

  https://cdimage.debian.org/debian-cd/current/amd64/iso-dvd/debian-10.3.0-amd64-DVD-3.iso
- with their SHA256SUMs at

  https://cdimage.debian.org/debian-cd/current/amd64/iso-dvd/SHA256SUMS
- If you want detailed installation instructions, use

  https://www.debian.org/releases/stable/installmanual

c) Burn the ISOs to DVDs (3 DVDs required)

d) Reboot the server to boot from the DVD

- This may require reconfiguring the BIOS to set the optical drive as the first boot option

e) Answer the questions posed and allow the system to install Debian

- Screen 1: Select Graphical Install
- Screen 2: Select a language: English
- Screen 3: Select a Country: Canada
- Screen 4: Select a Keyboard: American English
- [Wait for the installer to process the information provided so far]
- Screen 5: Hostname: {hostname}.{domain.tld}
- Screen 6: Non-root user full name: Free-text version of the user's name
- Screen 7: Non-root username: {UserName}
- Screen 8: Non-root user password: {UserPassword}
- Screen 8: Time Zone: Eastern
- Screen 9: Partition Disks: I just accepted the default "Guided - use entire disk"
- Screen 10: Select the target Hard Disk: I just accepted the default
- Screen 11: Partition Options: I just accepted the default - "All files in one partition …"
- Screen 12: Confirm Partition Options: I just accepted the default - "Finish Partitioning …"
- Screen 13: Confirm Partition Options (again): Click the "Yes" Radio Button and Click Continue
- [Wait for the installer to process the information provided so far]

- Screen 14: You will be prompted to scan additional DVD disks for apt: Click No and Continue
- Screen 15: Configure a network mirror: Yes
- Screen 16: Choose the Country in which the mirror you want is located: Canada
- Screen 17: Choose the mirror: mirror.csclub.uwaterloo.ca
- Screen 18: Identify a Proxy Server: leave blank
- Screen 19: Participate in the developer statistics collection: No
  - ◊ This can be changed later by running "dpkg-reconfigure popularity-contest"
- Screen 20: Select Software to be installed
  - ◊ Debian desktop environment: Uncheck
  - ◊ print server: Uncheck
  - ◊ SSH server: Check
  - ◊ Standard system Utilities: Check
  - ◊ All others: Uncheck
- [Wait for the installer to process the information provided so far]
- Screen 21: Install GRUB boot loader: Yes
- Screen 22: Select disk onto which the GRUB boot loader will be installed: Pick the same disk used for the software install (NOT the "Enter device manually" option)
- Screen 23
  - ◊ The optical disk will eject
  - ◊ Remove the disk
  - ◊ Click Continue
- [Wait for the system to boot from the hard disk]
- [You are now presented with the login prompt; proceed as with any other Debian system]
  - ◊ If you would like to switch to a remote access tool like Putty, get the temporary IP address with

    ip addr

f) Add my standard configurations to Debian

- I like to use Putty with its increased functionality over the Proxmox Console so if you want to do the same, get the current IP address of the Wazo server by entering, in the CLI of the Console

  ip addr

- I am not sure why but when you login via non-root user and then su to superuser, the /sbin directory is not included in the path so when you go to use some commands, it says "{application} not found" and you have to enter "/sbin/{application} or /usr/sbin/{application} to make it work.
  - ◊ So to add /sbin back into the path defitnition, add, to the end of the .bash_aliases file in /root/.bash_aliases

- nano /root/.bash_aliases

  # For some reason, when I change from user to superuser

  # the /sbin directory was not included in the PATH variable

  # so when I typed {program_name}, it did not find the progam,

  # and to use the program, I had to enter /sbin/{program}.

  # This adds /sbin to the PATH variable

  # so you can, once again, just enter {program}

  export PATH="/sbin:$PATH"

- You use .bash_aliases instead of .bashrc in case a future upgrade replaces the .bashrc file.

◊ Logout and log back in to your root user and the /sbin directory will now also be included in the search any time you type a command

- Change to a fixed IP address

◊ Unless shown otherwise, leave settings as they are

- Use the {interface descriptor}as shown in the original settings

◊ Comment out the line

iface {interface descriptor} inet dhcp

◊ Add the content (replacing {interface descriptor) with the entry from the above line

Fixed IP Address setup

auto {interface descriptor}

iface {interface descriptor} inet static

address {fixed_IP_address}

netmask 255.255.255.0

gateway 192.168.1.1

bridge_ports {interface descriptor}

bridge_stp off

bridge_fd 0

network 192.168.1.0

broadcast 192.168.1.255

- Change the repository list to include the non-free Debian repositories and use local mirrors

nano /etc/apt/sources.list

◊ Comment out all the existing Repositories

◊ And ONE of the following

- Direct from Debian

  # Standard Debian repositories, including non-free repositories

  deb http://deb.debian.org/debian buster main contrib non-free

deb-src http://deb.debian.org/debian buster main contrib non-free

deb http://deb.debian.org/debian-security/ buster/updates main contrib non-free

deb-src http://deb.debian.org/debian-security/ buster/updates main contrib non-free

deb http://deb.debian.org/debian buster-updates main contrib non-free

deb-src http://deb.debian.org/debian buster-updates main contrib non-free

- ♦ From a local Mirror (I use University of Waterloo; find one close to you)

  # Standard Debian repositories, including non-free repositories, from the UoW Mirror

  deb http://mirror.csclub.uwaterloo.ca/debian/ buster main contrib non-free

  deb-src http://mirror.csclub.uwaterloo.ca/debian/ buster main contrib non-free

  deb http://mirror.csclub.uwaterloo.ca/debian/ buster-updates main contrib non-free

  deb-src http://mirror.csclub.uwaterloo.ca/debian/ buster-updates main contrib non-free

  deb http://mirror.csclub.uwaterloo.ca/debian-security/ buster/updates main contrib non-free

  deb-src http://mirror.csclub.uwaterloo.ca/debian-security/ buster/updates main contrib non-free

- Update the package list and upgrade the system

  apt update && apt -y dist-upgrade && apt autoremove && apt autoclean

  - ◊ If you get "packages held back" try (in descending order of preference = safety

    apt-get --with-new-pkgs upgrade

    apt install <name of held-back package>

  - ◊ Reboot the server
    - ♦ Note that after changing the IP address, when you first ssh in (using Putty) you will again be asked to allow the root certificate to be stored in Putty
- Add a new, non-root-user
  - ◊ Done as part of the initial Debian installation, except for Proxmox, so see the instructions in Proxmox to add a non-root user in Proxmox
- Give the new non-root user sudo access (and test before proceeding to next step)
  - ◊ NOT done in my normal Debian installations to improve security by forcing anyone logging in via SSH as non-admin user to know the root password to upgrade to superuser
  - ◊ Having said that, when Wazo is installed, it installs sudo for Wazo script to use but does NOT add the {user} created during the Debian installation to the sudo file so the {user} cannot use sudo - and we want it kept that way.
  - ◊ If you want to use SPICE as a protocal for remote desktop and you want to use my Perl script on your PC to automate the fetching of a SPICE ticket, you will need to use sudo on the Proxmox host instance.
  - ◊ To install and setup sudo to work with the non-root user
    - ♦ To add a user (not needed in this instance, kept here for reference)

adduser {non-root_username}

Password:

Full Name:

Room Number: <Enter>

Work Phone:

Home Phone: <Enter>

Other <Enter>

Correct?: y

- ♦ To install sudo (if needed; it is needed for the Proxmox host instance) - do this as superuser

  apt install sudo

- ♦ To enable sudo for that user

  usermod -aG sudo {non-root_username}

  You will need to logout (exit) and login for the new group membership to take effect

- ♦ To check that the user is now able to use sudo, check active group memberships with - you will need to be at the {non-root_username} after having loggegd out and back in:

  id or sudo -l

- ♦ From now on, if you installed sudo, to update, from the user prompt, use ONE of:

  To invoke the one-line commend to temporarliy operate as superusr (root) and then return to normal user

  sudo apt update && sudo apt -y upgrade

  sudo -s [enable superuser and stay in normal user home directory]

  sudo -i [enable superuser and change to root home directory]

  To invoke superuser (root) from which you must exit to return to normal user

  su [invoke superuser but stay in normal user home directory]

- Set so remote access is not allowed with root (leaving only the new, non-root user with ability to remotely access)

  ◊ This will now require users to access the system with the non-root user and then upgrade their status to superuser, with an explicit root/password combination, adding an extra layer of authentication to anyone wanting root access.

  ◊ This just disables REMOTE (SSH) access to the server using user/password authentication. A user with physical access/connection to the server can still login with root user and password. Also, root login using the Proxmox Console will continue to work since the Proxmox Console acts as if you were physically connected with a terminal to the LV.

  ◊ Normally the default installation of Debian 10 disables remote access for root user so a new user needs to exist before we can SSH in to the host. This is whyn Debian 10 creates a non-root user during initial installation. So this step is no longer needed for Debian 10, but the instructions are kept here for reference.

- ◊ If you are signed in as root, logout and sign back in as the non-root user established above. Then upgrade the non-root user to superuser with

  su

  and the superuser password. If this does not work, re-do the instructions above and test until it works before disabling direct root-level SSH access. You will be disabling direct root SSH access so do not want to be in root when doing this over SSH.

- ◊ Now disable root-level SSH access

  [in Debian 10, root login is already prevented - to confirm look for #PermitRootLogin prohibit-password]

  [If you did need to disable root login via SSH, you would do the following:]

  nano /etc/ssh/sshd_config

  - ♦ Change

    PermitRootLogin yes

    to

    PermitRootLogin no

- ◊ Reload the ssh configuration

  systemctl restart sshd

- ◊ Test that you can no longer login remotely with ssh at root level, only at user level (and then upgrade to root privileges)

- Set so remote access is only allowed via key-pair, not user/password

  - ◊ This just disables REMOTE (SSH) access to the server using user/password authentication. A user with physical access/connection to the server can still login with root user and password. Also, root login using the Proxmox Console will continue to work since the Proxmox Console acts as if you were physically connected with a terminal to the LV.

  - ◊ Create the public/private key pair to be used for the access

    - ♦ Use PuttyGen on your PC to create 2048-bit, RSA (SSH-2), private and public key pairs on your PC (without a passphrase if on a personal-use PC). Optionally you can use ssh-keygen on your Proxmox host, but I find Puttygen to be very easy to use.

    - ♦ Save the Private key in a folder on the PC to which Putty will point. Save the Private Key without a Passphrase (assuming you are on a personal, not public PC). The Private key will end in .ppk so will be something like

      {PrivateKeyName}.ppk

    - ♦ Save the Public key in a folder on the PC (for future use as required). The Public key will end in .txt so will be something like

      {PublicKeyName}.txt

    - ♦ Copy and paste the text from the text box "Public key for pasting into OpenSSH authorized_keys file" and save it into the same folder as above with a .txt suffix so it will look something like:

      {PublicKeyNameTextForPasting}.txt

◊ Enable the server to use key-pair authentication

- Make sure the server is set to enable authentication via RSA keys (it should be enabled by default on Debian 10-based installs). Check file /etc/ssh/sshd_config if it is not working and adjust settings as required to enable it.

◊ Store the public key on the server

- This will go into the home directory of the non-root user created above, since you will be logging in as the non-root user and then upgrading to superuser once in.
- Check that the following directory and file exist in the server host under the non-soot user. If it does not exist create the directory path and file.

  /home/{UserName}/.ssh/authorized_keys

  If it does not exist (do this as user, NOT superuser):

  cd /home/{UserName}

  mkdir .ssh

  chmod 700 .ssh

  chown {UserName}:{UserName} .ssh

  cd .ssh

  If the authorized_keys file already exists in the /home/{UserName} directory, do NOT replace it; add to it.

  There are many ways to transfer a file from the PC to the linux server, but the easiest way I have found is by using the copy/paste capability in Putty, so I will show this method. If you have a preferred method, feel free to use. Just make sure that if the authorized_keys file already exists, you ADD to the authorized_keys file instead of replacing its contents or you will disable any prior keys that had been put into the file.

  On the PC, using the Putty client connected to the linux server

  nano /home/{UserName}/.ssh/authorized_keys

  Copy the contents from the {PublicKeyNameTextForPasting}.txt file created above

  Paste the contents just copied into the Putty client window (making sure to paste after any existing content in the file (if any))

  Save the /home/{UserName}/.ssh/authorized_keys file

- Restrict permissions to authorized_keys

  chmod 600 /home/{UserName}/.ssh/authorized_keys

  chown {UserName}:{UserName} authorized_keys

- Cleanse the authorized_keys file to ensure it has no Windows control characters. You may have to install the dos2unix package to accomplish this.

  apt install dos2unix [requires superuser authorization]

  dos2unix /home/{UserName}/.ssh/authorized_keys

◊ Equip Putty to login using the key-pair

- On the PC, leave the current Putty session open and open a new Putty session
- In the HostName (or IP address) field, enter the IP address of the server
- In the Connection section:

  In Data

  Auto-login username: {UserName}

  In SSH, click on Auth (do NOT expand with the + sign, click directly on Auth)

  Browse for the Private key file for autherntication
- Scroll back to the top and click on Session

  In the Saved Sessions field enter a free-text descriptor of the server to which you are connecting with the key-pair

  Click Save

  Click Open

  [To test the key-pair functionality before disabling user/password access]

  su

  {root-level-password}

◊ Prevent non-key-pair login

- You have now validated that you can login with a non-root-level user and upgrade to superuser. Now you want to disable remote/SSH user/password access.
- nano /etc/ssh/sshd_config
- Change

  # PasswordAuthentication yes

  ChallengeResponseAuthentication no

  UsePAM yes
- To

  PasswordAuthentication no

  ChallengeResponseAuthentication no

  UsePAM no
- systemctl restart sshd
- Test that password authentication no longer works and key-pair authentication does

- Edit the setup so SSH access has colour-coded prompts to distinguish root and non-root users at the terminal prompt

◊ I like coloured prompts on the terminal since it served as an extra reminder to me when I was working at root level, but the Debian devlopers removed the colour prompts saying they were distracting, not helpful. I disagree and returned the coloured prompts.

◊ You need to change the .bashrc settings for both the user and root

◊ To see the changes take effect, after editing the files, close the terminal and re-open it

◊ For the user (done with user NOT superuser)

- nano /home/{user}/.bashrc
- Change

  #force_color_prompt=yes
- To (ie uncomment it)

  force_color_prompt=yes
- change

  if ["$color_prompt" = yes]; then

  PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '

  else

  PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '

  fi
- to

  if ["$color_prompt" = yes]; then

  #
  PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '

  # Colour Legend: Green 00;32 ; Light (Bold) Green 01;32 ; Red 00;31 ; Light (Bold) Red 01;31 ; Blue 00;34

  PS1='${debian_chroot:+($debian_chroot)}\[\033[00;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '

  else

  PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '

  fi

◊ For root (done as superuser)

- Debian does not add the full .bashrc set to the root .bashrc file so to create the full set, which you will then edit below, look in /etc/skel/.bashrc for the .bashrc template and copy it in its entirety over to the .bashrc files - making sure to retain any custom entries if you previously made any in .bashrc

  cat /root/.bashrc [to check the current contents]

  either nano (to edit existing) or copy new contents (if no existing entries in .bashrc)

  nano /root/.bashrc

  then copy/paste from /etc/skel/.bashrc

  OR

  cp /etc/skel/.bashrc /root/.bashrc

nano /root/.bashrc

- ◊ For root (done as superuser)
  - ♦ Now make the changes to the .bashrc file (as above, but with parameters for root)

    su

    nano /root/.bashrc

    Change

    #force_color_prompt=yes

    to (ie uncomment it)

    force_color_prompt=yes

    nano /root/.bashrc
  - ♦ Change

    if ["$color_prompt" = yes]; then

    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '

    else

    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '

    fi
  - ♦ To

    if ["$color_prompt" = yes]; then

    #
    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;31m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '

    # Colour Legend: Green 00;32 ; Light (Bold) Green 01;32 ; Red 00;31 ; Light (Bold) Red 01;31 ; Blue 00;34

    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;31m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '

    else

    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '

    fi

- Configure a mail server to email out only (no inbound or relay)
  - ◊ The sasl module (for encrypted communication with an external email server) needs to be present. It is present in a standard Debian 10 installation but not in Proxmox, so is left here for reference when installing and configurting Proxmox. For Proxmox, install the library with:

    apt install libsasl2-modules
  - ◊ For WazoPBX, Postfix is installed as part of the Wazo installation, so do NOT do anything

yet for the WazoPBX installation but come back to this set of instructions after installing WazoPBX and then proceed with these instructions.

◊ If Postfix is already installed, use it. To determine if Postfix is installed to handle mail

whereis postfix

◊ If Postfix is installed (like in Proxmox host and WazoPBX) follow the Postfix instructions below

◊ If Postfix is NOT installed

- ♦ You could install the full Postfix system and configure as below OR you can install a simple system, intended only for outbound mail handline (which is all we want anyways).
- ♦ The default simple Mail Transfer Agent (MTA) to use to send emails using an external mail relay has, for years, been ssmtp. However, ssmtp has not been maintained for some time now and it has been deprecated in Debian 10 due to the discovery of some bugs that affect the content and recipient list of emails. There may be a patch coming to allow ssmtp to be used again, but at the time of this document, there was none, so an alternative setup using a package called msmtp is used instead.

  apt install msmtp msmtp-mta

- ♦ make sure to install the above BEFORE installing mailutils or mailutils will install the full Postix application

  apt install mailutils

  touch /var/log/msmtp

  chown {UserName}:root /var/log/msmtp

  chmod 766 /var/log/msmtp

  nano /etc/msmtprc

  # Settings that apply to all accounts

  defaults

  # Lock in default "From" header to avoid aliasing/masquerading issues

  auto_from off

  add_missing_from_header on

  from "{Real Name}"<{admin_email}>

  domain {domain.tld from which you want emails to be sent}

  # Setup for use of TLS

  auth on

  tls on

  # tls_certcheck off

  tls_trust_file /etc/ssl/certs/ca-certificates.crt

  # Do NOT change the logfile directory or name

  # Apparmor will not let you write to another file

  logfile /var/log/msmtp

aliases /etc/aliases

Settings specific to {account}

account {account}

host {mailhub} [without port appended]

port 587

user {AuthUser}

password {AuthPass}

Use {account} as the default account

account default : {account}

nano /etc/aliases

root: {admin_email}

default: {admin_email}

newaliases

nano /etc/mail.rc

set sendmail="/usr/bin/msmtp -t"

- Test the functionality of the outgoing email with

  echo "Someday, this puppy is going to work!" | mail -s "Testing with msmtp" {test@someemail.com}

  tail -n 20 /var/log/msmtp

  and read the 20 most recent entries in the log file for status indication

  or use tail -f /var/log/msmtp to continuously monitor the file and then use <Ctl>-c to exit the tail session

  An alternative test format is to use the native send capabilities of msmtp (not emulating the mail syntax) which will work for test, but system mails are sent with mail so use the above to test the system and if you want, use this to work directly with msmtp options. You could use "echo -e" or printf since the command line needs to recognize "\n" as a newline.

  printf "To: DestinationName<adminemail@domain.tld>\nFrom: SenderName<authorizedemail@domain.tld>\nSubject: This is the subject\n\nAnd this is the body\nwith multiple line possible\nif you want" | msmtp admineamail2@domain.tld

  DestinationName can have a space

  SenderName can have a space

  authorizedemail@domain.tld and admineamail2@domain.tld do NOT have to be the same, but if not, the header will show admineamail@domain.tld as the To: in the mail client and you risk a SPAM check flagging this as a suspicious email

  with msmtp, you could use the command line to compose your email by doing the following (where <Enter> means press the Enter key on your keyboard and <Ctrl-d> means press the Ctrl key and the d key simultaneously (to Deliver the mail)):

msmtp destination@domain.tld <Enter>

To: adminemail@domain.tld

From: authorizedemail@domain.tld

Subject: This is the subject

And now you just enter the body

with as many lines as you want

<Ctrl-d>

◊ If Postfix is installed, or, for some strange reason, you want to go through the complicated scenario of configuring full Postfix instead of MailUtils (hint, hint) (do this OR above, NOT both), enable Postfix mail-out capability for system by

- Install Postfix [if required, NOT if already installed which it is in Proxmox and Wazo]

  apt install postfix

- To see what the current configuration settings are for Postfix, at the CLI, run

  postconf

  or, to get a specific setting

  postconf | grep {some search string}

- In the WazoPBX installation, do this AFTER installing and configuring Wazo (see WazoPBX instructions below) because the process of installing and configuring Wazo reconfigures Postfix and you want to work with the Postfix configuration after Wazo has made its changes to the configuration file or Wazo will override your configurations of Postifx.

  Edit the main Postfix configuration file main.cf

  I have configured this for a Postfix setup that is NOT on a server using a fully qualified domain name and the settings reflect that. If you have a FQDN, the setup is easier; go to http://www.postfix.org for setup instructions.

  For Wazo, the /etc/postfix/main.cf will be overriden after each wazo-upgrade, so do not directly edit the main.cf file at /etc/postfix/main.cf . Instead, AFTER installing and configuring Wazo:

  mkdir -p /etc/xivo/custom-templates/mail/etc/postfix

  cp -a /usr/share/xivo-config/templates/mail/etc/postfix/main.cf /etc/xivo/custom-templates/mail/etc/postfix/main.cf

  And then edit the main.cf file at /etc/xivo/custom-templates/mail/etc/postfix/main.cf just like you would have edited the main.cf file at /etc/postfix/main.cf in a normal Postfix installation (see below)

  nano /etc/xivo/custom-templates/mail/etc/postfix/main.cf

  Update the actual Postfix main.cf by running

  xivo-update-config

- For the full documentation on postfix and postfix configuration, at the CLI enter

  man postfix

man postconf

man 5 postconf (directly to the section summarizing the configuration options and methods)

As always, enter q (quit) to return to the command line

- To confirm that postfix is the email MTA

  sudo netstat -ltnp | grep :25

  If netstat is not installed, install it (with other utilities)

  apt install net-tools

- Confirm that Postfix was configured with SSL and SASL support

  Enter, at the CLI:

  whereis postfix

  You will get back

  postfix: /usr/sbin/postfix /usr/lib/postfix /etc/postfix /usr/share/postfix /usr/share/man/man1/postfix.1.gz

  For Debian, the normal location of the application will be at /usr/sbin/postfix

  Using the /usr/sbin location given, at the CLI enter:

  ldd /usr/sbin/postfix

  You will get a list of modules. You are looking for one starting with libssl.so. and one stating with libsasl2.so

  libssl.so.1.1 => /usr/lib/x86_64-linux-gnu/libssl.so.1.0.0 (0x00007f35d0bfa000)

  libsasl2.so.2 => /usr/lib/x86_64-linux-gnu/libsasl2.so.2 (0x00007f29643ba000)

- Check that the certificates installed and used by Postfix were properly installed when Postfix was installed

  smtpd_tls_key_file = /etc/ssl/private/ssl-cert-snakeoil.key

  smtpd_tls_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem

  smtpd_tls_CAfile = /etc/ssl/certs/ca-certificates.crt

- Interrogate the system intended to be used as a mail relay to see how it authenticates. At the CLI enter:

  telnet {mailhub} 25 [leave the port number off the end (:587) and use 25]

  Trying [IP address of mail server returned]...

  Connected to [name of mail server].

  Escape character is '^]'.

  220 [mail server FQDN] ESMTP [mail server name]

  ehlo {mailhub} [leave the port number off the end]

  250-smtpin.mailjet.com

  250-PIPELINING

250-SIZE 15728640

250-VRFY

250-ETRN

250-STARTTLS

250-AUTH PLAIN LOGIN DIGEST-MD5 CRAM-MD5

250-AUTH=PLAIN LOGIN DIGEST-MD5 CRAM-MD5

250-ENHANCEDSTATUSCODES

250-8BITMIME

250 SMTPUTF8

quit (or bye if quit does not work)

- The above output tells you the mail server you want to use as a relay uses LOGIN and PLAIN authentication and has SSL encryption (STARTTLS) to protect your password when it is sent.
- Redirect internal (system) emails to root to an external administrator account by adding at the end of /etc/aliases the external email address to which the mail directed to root (and other users) should go

  nano /etc/aliases

  root: {admin_email}

  {user}: {admin_email}

  www-data: {admin_email}

  wazo: {admin_email}

  default: {admin_email}

  newaliases

- Create the encrypted username / password file to use for smtp authentication

  Create a file in /etc/postfix/sasl called smtp_sasl_password_map

  nano /etc/postfix/sasl/smtp_sasl_password_map

  {mailhub} {AuthUser}:{AuthPass}

  [{mailhub} is the smtp mail server FQDN with the port appended after a colon (:)]

  Create a hashed .db version of the file with

  postmap hash:/etc/postfix/sasl/smtp_sasl_password_map

  Change the permissions so only root can read this file and the .db version of it

  chmod 600 /etc/postfix/sasl/smtp_sasl_password_map

  chmod 600 /etc/postfix/sasl/smtp_sasl_password_map.db

  Confirm the files are properly set up by entering, at the CLI (all on one line, using the exact same formatting (brackets, ...) as you used above):

  postmap -q {mailhub} /etc/postfix/sasl/smtp_sasl_password_map

If all is working properly, you will have returned your username and password

- Create the file to re-map internal email destinations to FQDN source emails (Only needed when server does NOT have FQDN and instead uses something like thisserver.localdomain)

  Create, in /etc/postfix/ a file called generic with

  nano /etc/postfix/generic

  root {admin_email}

  {user} {admin_email)

  Logwatch {admin_email}

  asterisk {admin_email}

  www-data {admin_email}

  wazo {admin_email}

  Create a hashed .db version of the file with

  postmap hash:/etc/postfix/generic

- Edit a file to re-map internal email sources to FQDN source emails (Only needed when server does NOT have FQDN and instead uses something like thisserver.localdomain)

  Edit, in /etc/postfix/ a file called canonical with

  nano /etc/postfix/canonical

  root@hostname.domain.tld {admin_email}

  root@domain.tld {admin_email}

  root {admin_email}

  root@root {admin_email}

  {user}@hostname.domain.tld {admin_email}

  {user}@domain.tld {admin_email}

  {user} {admin_email}

  Create a hashed .db version of the file with

  postmap /etc/postfix/canonical

- Edit the main Postfix configuration file at /etc/postfix/main.cf

  I have configured this for a Postfix setup that is NOT on a server using a fully qualified domain name and the settings reflect that. If you have a FQDN, the setup is easier; go to http://www.postfix.org for setup instructions.

  cp -a /etc/postfix/main.cf /etc/postfix/main.cf.orig

  [Backup the main.cf to main.cf.orig] [make sure to include the -a flag in the copy command to preserve orginal file permissions]

  nano /etc/postfix/main.cf

  NOTE for WazoPBX this will be overriden after each xivo-upgrade, so instead, as noted above, AFTER configuring Wazo:

mkdir -p /etc/xivo/custom-templates/mail/etc/postfix

cp -a /usr/share/xivo-config/templates/mail/etc/postfix/main.cf /etc/xivo/custom-templates/mail/etc/postfix/main.cf

nano /etc/xivo/custom-templates/mail/etc/postfix/main.cf

Edit existing file to comment out existing duplicate entries and add the following lines to the file (change the hostname to match the {Hostname} of the Proxmox host (for the Proxmox installation) or the LVs (for the other instances)

These are the settings to configure the OUTGOING mail (smtp configurations)

There are no smtpd configurations since we are NOT allowing incoming mail except from localhost

myhostname and mydomain are what you see when you enter hostname -f at the cli ; mydomain is the tld and myhostname is the rest

append_dot_mydomain = yes

append_at_myorigin = no

myhostname = {Hostname}.{mydomain}

mydomain = {mydomain}

The domain name used as the real email address in the generic maps

myorigin = {rewrite_domain}

The file location to remap the From addresses

If also present above, either use it and NOT this or comment out previous entry

canonical_maps = hash:/etc/postfix/canonical

Accept emails only from the localhost

inet_interfaces = loopback-only

this will auto-populate since the ${} serves as a variable lookup; leave as is

mydestination = ${myhostname}, localhost.${mydomain}, ${mydomain}

Note: If you used Square brackets around the server in password map, you must do the same here

relayhost = {mailhub}

smtp_sasl_auth_enable = yes

smtp_sasl_password_maps = hash:/etc/postfix/sasl/smtp_sasl_password_map

This prevents anonymous logins and plaintext - TLS is required and if it is not present, no emails will be sent

smtp_sasl_security_options = noanonymous

smtp_tls_security_level = encrypt

Have the From emails come from the domain only instead of the mydomain

masquerade_domains = ${mydomain}

This prevents a failed authentication by preventing the server from using a mechanism the Cyrus library does not recognize

smtp_sasl_mechanism_filter = plain, login

smtp_cname_overrides_servername = no

This prevents unencrypted and anonymous logins when NOT using TLS

smtp_sasl_security_options = noplaintext, noanonymous

This prevents anonymous logins when using TLS but allows plaintext since the communications itself is encrypted when using TLS

smtp_sasl_tls_security_options = noanonymous

smtp_generic_maps = hash:/etc/postfix/generic

This prevents warning messages from filling logs with backwards-compatibility warnings that are not a concern

compatibility_level=2

Load the new configuration with

xivo-update-config [WazoPBX only]

service postfix reload

systemctl restart postfix

or

postfix stop

postfix start

or

reboot the server [this is what I do usually]

Test the functionality of the outgoing email with

echo "Someday, this puppy is going to work!" | mail -s "Testing with Postfix" {external email address}

tail -n 30 /var/log/mail.log

to display the last 30 lines of the file and then return to the CLI

Or use

tail -f /var/log/mail.log

to read the most recent entries in the log file for status indication and keep monitoring the log file for new entries

If you used tail -f /var/log/mail.log , that command keeps monitoring the file and will continue to output new entries to the file as they occur. To stop the monitoring and return to the CLI, use <Ctl>-c to exit the tail session

- Install apt-listchanges to email admin when package updates are available
  - ♦ apt install apt-listchanges

    [apt-listchanges is installed by default in Debian 10 but is left here to ensure you know the required dependencies]

Also, you need to configure apt-listchanges so it does not try to email root but instead emails the administrator. By running the command shown below, the configuration of apt-listchanges (for the "apt" profile) is stored in /etc/apt/listchanges.conf and the ENVIRONMENT variables for apt-listchanges is stored in /etc/apt/apt.conf.d/20listchanges .

dpkg-reconfigure apt-listchanges

Method used to display changes: mail

Choose which types of changes should be displayed with APT: both

Disable receiving changes over network: no

Email address(es) which will receive change: {admin_email}

Format of email messages: text

Insert headers before changelogs: no

Show changes in reverse order: no

Should apt-listchanges skip changes that have already been seen: yes

The results of this configurator will be stored in /etc/apt/listchanges.conf

- Install unattended-upgrades to have system automatically install selected upgrades
  - ◊ In LinuxMint and Wazo, do NOT install unattended-upgrades
    - ♦ Wazo has an integrated upgrade mechanism that is to be used for upgrades to the OS and the PBX (wazo-upgrade -d and wazo-upgrade)
    - ♦ (Optional) in Linux Mint, install apt-listchages (see below) so you are notified when packages have changed and an upgrade is pending
    - ♦ In LinuxMint, set this up via the Control Center -> Administration -> Update Manager -> Edit -> Preferences -> Auto Upgrade -? Check "Apply Updates Automatically"
    - ♦ You can fine tune these settings. By default ALL (not just security) updates are automatically installed

      /etc/cron.daily/mintupdate

      /etc/mintupdate.blacklist
    - ♦ See the Linux Mint installation instructions for details
  - ◊ Install unattended-upgrades
    - ♦ apt install unattended-upgrades

      dpkg-reconfigure --priority=low unattended-upgrades

      Automatically download and install stable updates: [Prompt]

      Accept the default (Yes) by pressing <Enter>

      [This reconfigures the config file at /etc/apt/apt.conf.d/20auto-upgrades]

      nano /etc/apt/apt.conf.d/50unattended-upgrades

      after the line //Unattended-Upgrade::Mail "root"; add the line

      Unattended-Upgrade::Mail "{admin_email}";

[This set the email to be notified when updates occur]

The above configures unattended upgrades to automatically upgrade security and other Debian buster stable packges. The Wazo repository is NOT included in the default configuration.

To see what the current unattended-upgrades configuration is, look at

/etc/apt/apt.conf.d/50unattended-upgrades

To see what the additional options are for inclusion in unattended-upgrades, run

apt-cache policy

- Install logwatch to have the system email a report to the administrator every day
  - ◊ For SuiteCRM and Linux Mint, make sure to install the msmtp and mailutils BEFORE installing logwatch or logwatch will install Postfix. For WazoPBX, install Logwatch after installing Wazo so Postfix is already installed.

mkdir /var/cache/logwatch

apt install logwatch

cp -a /usr/share/logwatch/default.conf/logwatch.conf /etc/logwatch/conf/logwatch.conf

nano /etc/logwatch/conf/logwatch.conf

change

Output = stdout

to

Output = mail

change

MailTo = root

to

MailTo = {admin_email}

change

MailFrom = Logwatch

to

MailFrom = {MailFrom_email_address}

[use the format for MailFrom
"free text name" <email@domain.tld>

including quotation marks]

Change

Detail = Low

to

Detail = 5 (or Detail = Med)

In previous versions of Logwatch the Detail default setting was 3 which resulted in a Logwatch report every day even when there was not activity

logged. In newer versions the Deatil setting is Low which results in no Logwatch report when there is no activity. I want to use Logwatch as a quick daily report to confirm the server is up and running if so I want a daily report regardless of the activity level so I returned the Detail level to a higher level.

(Optional) Change

#Archives = No

to

Archives = Yes

Note: This is now the default but is set here to be clear of the intention

◊ (Optional) Set a particular service to a higher (or lower) detail of reporting

- I was having issues with invalid emails being sent by the server so I needed to troubleshoot the email service with full detail, but I did not need or want logwatch reporting full detail on all services. So I created a specific override configuration to have the postfix service reported with high detail (when postfix was the email service used) by adding an override.conf file into the /etc/logwatch/conf directory and populating the file with:

  nano

  services/postfix: Detail = High

  (or services/msmtp:Detail = High)

◊ (Optional) Change the time the daily cron job runs to create the logwatch

- The default configuration for logwatch is to run once a day, whenever the cron.daily scripts are executed. This can end up with yesterday's logwatch report not being received until later in the morning and I prefer to see the report first thing in the morning when I start my computer, in case there is something I need to look at before the day gets going.
- To set the time manually, you need to undo the default setup and create a new cron schedule to be run as root.
- Move the current logwatch activation script from /etc/cron.daily/00logwatch to /bin/00logwatch

  mv /etc/cron.daily/00logwatch /bin/00logwatch

- Now edit the root-level crontab file to add a trigger to run logwatch early in the morning by adding the following to the bottom of the file (Note this is set to run at 1:44am; change if you want a different time)

  nano /etc/crontab

  # Manually set the time for the logwatch script to run

  44 1 * * * root /bin/00logwatch 1>/dev/null 2>&1

◊ To test if logwatch is working

For this, the initial test, you need to manually declare the date range to today (since there is no yesterday) and output to be stdout (since you set the default to mail above

logwatch --range=today --output=stdout

After configuring the system if you want to just email yesterday's logwatch to the established admin email

logwatch

◊ To manage msmtp logs by rotating the log file so you do not end up with one massive log file, edit the logratate configuration by adding a new file msmtp to the /etc/logrotate.d directory

nano /etc/logrotate.d/msmtp

```
/var/log/msmtp {
    # set the user:group permissions for this configuration so logrotate can operate
    # this is needed because of the "loose" permissions of the log file
    su root {user}
    # frequency of rotation
    weekly
    # keep 5 old logs
    rotate 5
    # don't do anything if the log is missing
    missingok
    # don't do anything if the log is empty
    notifempty
    # zip the archived logs
    compress
    delaycompress
    # define permissions of newly created (empty) log file
    create 766 {user} root
}
```

♦ To test if the logrotate is working, manually force a logrotate (after finishing the installation and testing msmtp to create the original log file) with

logrotate -vf /etc/logrotate.d/msmtp

◊ To monitor the msmpt logs and include the msmtp logs in the logwatch report (do this AFTER installing logwatch)

♦ Create the following files to add a custom logfile to logwatch reporting

/etc/logwatch/conf/logfiles/{LogFile_Configuration_File_name}.conf

To define a group which will use the location and name of the logfile (and logfile archives if using logrotate) to parse when reporting log entries. The only required declaration in this file is to set the LogFiles variable, but you can set additional ones - for example if you want archives searched as well.

Multiple services can access the same group.

To have logwatch report log activity for msmtp, you need to create a new group since there is no group existing for msmtp.

To add additional reporting on the log activity for postifix, you could edit the postfix script (below - if you are very conversant with Perl, this could be a viable option). If Perl is not your forte, you could create a new, simplified service, like you do for msmtp, to add a new logwatch section for Postfix. This is what I did. If you add a new service for Postfix, you do NOT need to create a new group since the logfile containing the Postfix logs is syslog in /var/log/syslog and the group for that has already been defined in the logwatch default setup at /usr/share/logwatch/default.conf/logfiles/syslog.conf . You will need to create the other files (service and script) in the next sections to make different use of this group.

/etc/logwatch/conf/services/{Service_Configuration_File_name}.conf

Define the name and some paramteres of the new "service" which will access the group defined above by logwatch for parsing the logfile(s). The filename must be the Service Name with the .conf suffix.

/usr/share/logwatch/scripts/shared/applymsmtpdate

Not usually required if the date format used by the custom logfile matches the date format in the syslog file, but for msmtp, the date format includes a leading 0 (zero) in the day when the day number is single digit and syslog uses a leading blank in the day when the day number is single digit, so you need to filter using a date format that is NOT the standard date format.

/etc/logwatch/scripts/services/{Service_Configuration_File_name}

Create a script that determines what entries in the logfile actually get reported with the logwatch report. The filename must be just the service name, without the .conf suffix.

/etc/logwatch/conf/override.conf

(Optional) set the logwatch detail for this logfile to a different (higher or lower) level than the default setting used in the logwatch.conf default configuration file

- Create the logfile group for msmtp by adding a new file msmtp.conf to the /etc/logwatch/conf/logfiles directory

nano /etc/logwatch/conf/logfiles/msmtp.conf

```
##########################################################

# $Id$

##########################################################


##########################################################

# This was written and is maintained by:

#    Richard Cantin <richard.cantin@ayuda.ca>

#    based on the template by

#    Kenneth Porter <shiva@well.com>
```

```
########################################################
# What actual file?  Defaults to LogPath if not absolute path....
LogFile = /var/log/msmtp
# If the archives are searched, here is one or more line
# (optionally containing wildcards) that tell where they are...
# To have logwatch search the archives, add the --archives flag to the logwatch command
#If you use a "-" in naming add that as well -mgt
Archive = /var/log/msmtp.*
Archive = /var/log/msmtp.*.gz
```

- Create a service to parse the msmtp logfiles (defined in the logfile group above) by adding a new file msmtpaudit.conf to the /etc/logwatch/conf/services directory. This filename (minus the suffix) defines the service name which will be accessing the logfile group (defined above). The filename (which does NOT have a suffix) of the script used by this service must match the service name.

  nano /etc/logwatch/conf/services/msmtpaudit.conf

```
# The title shown as header (begin) and footer (end) for the msmtp section in the logwatch report.
Title = "msmtp activity"
# The name of the log file GROUP (file name without suffix).
# This does NOT mean the name of the log that is being monitored is msmtp
# This means the logfile GROUP name (defined above) is msmtp
# The logfile GROUP is often named the same as the log filename
# but this is not a requirement of the naming for logfile group
LogFile = msmtp
# Expand the repeats (actually just removes them now)
*ExpandRepeats
# Keep only the lines in the proper date range...
# Unlike the standard date format for syslog logs,
# msmtp logs include a leading 0 (zero) in the day
# when the day number is single digit (ie June 08)
# Syslog uses a leading blank in the day when the day number is single digit (Jun  8)
# You need to filter using a date format that is NOT the standard date format.
# The upper/lower case used for the name is for readability only;
# the actual script name will be all lowercase
# and the name declared here will be converted to all lowercase
*ApplyMsmtpDate
```

- Create a new filter file that uses a custom date format filter so it will work with msmtp. msmtp date format includes a leading 0 (zero) in the day when the day number is single digit and syslog uses a leading blank in the day when the day number is single digit, so you need to filter using a date format that is NOT the standard date format.

  cp -a /usr/share/logwatch/scripts/shared/applystddate /usr/share/logwatch/scripts/shared/applymsmtpdate

  nano /usr/share/logwatch/scripts/shared/applymsmtpdate

  Change

  ```
  $SearchDate = TimeFilter($ARGV[0] || '%b %e %H:%M:%S ');
  ```

  to

  ```
  $SearchDate = TimeFilter($ARGV[0] || '%b %d %H:%M:%S ');
  ```

- Create a script (I use bash, you can use another language if you prefer), using the same name as the service, which will be used to parse the log file(s) defined by the msmtp group, by adding a new file msmtpaudit to the directory /etc/logwatch/scripts/services (thank you to Simon Oulevay at http://www.alphahydrae.com/2012/08/logwatch-how-to-add-a-service)

  nano /etc/logwatch/scripts/services/msmtpaudit

  ```
  #!/bin/bash
  #
  # Change the line separator to split by new lines.
  OLD_IFS=$IFS
  IFS=$'\n'
  # The contents of the log file are given in stdin.
  for MailLogLine in $( cat /dev/stdin )
  do
    # Only lines matching this regexp will be included.
    # Note use of RegEx comparator "=~" not normal Equal "=="
    TextToFind="recipients="
    if [[ "${MailLogLine}" =~ "${TextToFind}" ]]
      then
        # Every line we echo here will be included in the logwatch report.
        # echo  -e "${MailLogLine}\n"
        printf "%s \n\n" "${MailLogLine}"
    fi
  done
  IFS=$OLD_IFS
  ```

  (The above shows how to use a RegEx to use a filter, in the script, and echo commands to customize the format of the output, to only show lines you want

reported from a logfile. A simpler script which will accomplish the desired result if you want all log entries that match the date would be:)

```
#!/bin/bash
# This is as nice script that will show you the lines you will
# be processing and reporting on. It will first display the
# standard environment variables and then it takes STDIN and
# dump it right back out to STDOUT.

# These are the standard environment variables. You can define
# more in your service config file (see above).
# echo "Date Range: $LOGWATCH_DATE_RANGE"
# echo "Detail Level: $LOGWATCH_DETAIL_LEVEL"
# echo "Temp Dir: $LOGWATCH_TEMP_DIR"
# echo "Debug Level: $LOGWATCH_DEBUG"

# Now take STDIN and dump it to STDOUT
cat
```

Make the file executable

chmod +x /etc/logwatch/scripts/services/msmtpaudit

- (Optional) set the logwatch detail for this logfile to a different (higher or lower) level than the default setting used in the logwatch.conf default configuration file

  nano /etc/logwatch/conf/override.conf

  services/postfix: Detail = High

  (or services/msmtpaudit: Detail = High)

- Test the setup

  logwatch --service msmtpaudit --output stdout --range today --archives

- Install scripts to email the administrator on login for both root and user
  - ◊ Use .bash_aliases (called from .bashrc if if .bash_aliases exists) not .bashrc to avoid the changes being overwritten on an upgrade
  - ◊ Make sure to use who -m (include the -m flag) NOT just who (without the -m flag). For Wazo, and potentially other installations where you may have multiple logins for a user, using just who returns multiple lines and breaks the mail command, causing invalid emails and error messages from the mail relay.
  - ◊ For the root-level email
    - Do this as superuser (root) NOT regular user
    - If /root/.bash_aliases does not exist, create a /root/.bash_aliases file (which is called from /root/.bashrc)

If /root/.bashrc does not contain the call to /root/bash_aliases, (which it normally does, but worth checking) add to the end of the /root/.bashrc file

```
# Include .bash_aliases if it exists
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi
```

- ♦ Leave all () brackets and contents in place and leave ${Source_IP_Address} as shown but replace other {} brackets and the admin_email with an actual administrator email address
- ♦ Do this as superuser (root)

nano /root/.bash_aliases

```
# This will email the administrator whenever someone logs in as root
Source_IP_Address="$(who -m | cut -d'(' -f2 | cut -d')' -f1)"
echo "ALERT - Shell Access to $(hostname) by $(whoami) on $(date) ... Source IP Address: ${Source_IP_Address}" | mail -s "Alert: Shell Access to $(hostname) by $(whoami) from ${Source_IP_Address}" {admin_email}
```

◊ For the user-level email

- ♦ Leave all () brackets and contents in place and leave ${Source_IP_Address} as shown but replace other {} brackets and the admin_email with an actual administrator email address
- ♦ Do this as real user NOT superuser (root)
- ♦ If /home/{UserName}/.bash_aliases does not exist, create a /home/{UserName}/.bash_aliases file (which is called from /home/{UserName}/.bashrc)

If /home/{UserName}/.bashrc does not contain the call to /home/{UserName}/bash_aliases, (which it normally does, but worth checking) add to the end of the .bashrc file

```
# Include .bash_aliases if it exists
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi
```

nano /home/{UserName}/.bash_aliases

```
# This will email the administrator whenever someone logs in as user
Source_IP_Address="$(who -m | cut -d'(' -f2 | cut -d')' -f1)"
echo "ALERT - Shell Access to $(hostname) by $(whoami) on $(date) ... Source IP Address: ${Source_IP_Address}" | mail -s "Alert: Shell Access to $(hostname) by $(whoami) from ${Source_IP_Address}" {admin_email}
```

6) Installing Wazo PBX

a) These instructions assume a server with 16GB RAM, 4-core CPU and a 1TB hard disk. Adjust your

parameters to suit your installation.

b) Install Debian 10 (Buster) with the default local and the UTF-8 charset

- For Wazo, I used the actual Debian installer DVDs for this install. There is a Debian 10 template available as part of the Proxmox system, but I found it did not work well with Wazo, so I installed a fresh Debian 10 from the Debian DVDs I downloaded.
- Put the "Debian 10.3 - Disk 1 of 3" disk into the optical driver of the server
- Using the Proxmox web GUI
  - ◊ Select Datacenter from the Server View column (left column)
  - ◊ In the Menu Bar, Click on Create VM (Virtual Machine)
  - ◊ Check the Advanced checkbox at the bottom to get additional options
  - ◊ General (tab)
    - ♦ Node: proxmox (leave as is - you cannot change this)
    - ♦ VM ID: 100
    - ♦ Name: WazoPBX

      I use the same naming convention as the production server so I can use the development server to re-create a production server (or vice-versa) if required
    - ♦ Resource Pool: (leave as is - you cannot change this)
    - ♦ Advanced Settings

      Start at boot: Check

      Start/Shutdown order: 1

      Startup delay: 30

      Shutdown timeout: 30
  - ◊ OS (tab)
    - ♦ Select: Use Physical CD/DVD Drive

      OR, IF you pre-loaded an ISO earlier:

      Use CD/DVD disc image file (iso)

      Storage: Local

      ISO Image: Click Browse and locate the ISO on your hard drive

      Click Upload

      Guest OS:

      Type: Linux

      Version: 5.x - 2.6 Kernel
  - ◊ System (tab)
    - ♦ Leave all in default mode

You could customize this and have (for example) the LinuxMint installation “own’ the Graphics card by activating PCIe pass-through, but that would prevent the other VMs or host from using the graphics card, so for now, for simplicity, leave this all as default.

◊ Hard Disk (tab)

- ♦ Bus Device: SCSI (Leave as is to get best performance

  [it is NOT actually a SCSI controller, it is an emulated, high-performance controller and is recommended by Proxmox for Linux installs]
- ♦ Storage: lv-wazopbx
- ♦ Disk Size (GiB): Change from 32 to something less than what was alloacted above - you can see avaialable in the drop-down menu

  I used 136 since 139.07 was available
- ♦ Cache: Default (No cache)
- ♦ Discard: Checked
- ♦ Advanced Settings

  (leave rest as is)

◊ CPU (tab)

- ♦ Sockets: 1
- ♦ Cores: Change to 4 (I have a total of 4 so let each container use them all)
- ♦ Type: Default (kvm64)
- ♦ Advanced Settings

  VCPUs: Leave as is (4)

  CPU limit: leave as is (unlimited)

  CPU Units: 2048

  CPU Units is a weight applied to a VM to determine how much of a CPU's cycles a VM can get

  Max value 500000 ; min value 8

  The CPU cycles are apportioned (when there is a potential for conflict) based on the number given here for each VM

  So, if have 4 VMs with CPU Units: 1=2048 2=1024 3=512 4=512

  If the CPU is too busy to hadle everything, it will allocate the following % of its ccles to:

  1=50% 2=25% 3-12.5% 4=12.5%

  Enter pct cpusets at cli to see what is set

  In fact, do a man pct to see all the commands avaialble to the root user

  Enable NUMA:

  To help mitgate both Meltdown and Spectre

  From CLI: numactl --hardware | grep available

If this returns more than one node, then your host system has a NUMA architecture and you can use the Numa architecture which prevents shared memory and negates both bugs. This is not as efficient a use of RAM but it is safer - and enables faster RAM access since the memory is spread into local banks close to each socket. If the NUMA option is used, it is recommended to set the number of sockets to the number of sockets of the host system.

If Numa archtecture is avaialable, Check Enable NUMA

Extra CPU Flags:

There are a series of extra firmware setting that you can activate here to protect against Meltdown and or Spectre. I did not utilize any of them (for simplicity and because this hardware had modern architecture) but this does potentially leave me open to infection if I do not protect the server through other means (firewall, access restrictions).

For Meltdown

From CLI: grep ' pcid ' /proc/cpuinfo

If this does NOT come back empty, your CPU has support for pcid which can mitigate Meltdown

If your CPU supports pcid, check PCID

Enable SPEC_CTRL:

For Spectre

From CLI: grep ' spec_ctrl ' /proc/cpuinfo

If this does NOT come back empty, your CPU has support for spec-ctrl which can mitigate Spectre

If your CPU supports spec-ctrl, check SPEC-CTRL

- Memory (tab)

Memory:

Use 14336 when have 16 GB installed ; 6144 (= 6 GB used * 1024 MB per GB) with 8GB

You could set each to max Avaialable Memory = Free memory (see below), but you always want to leave a little unused for memory buffer.

Calculate the amount of memory available after the Host memory is taken

Use "free -m" from the CLI and read Free (= Total - Used)

Then Enter Free memory (less 10% for buffer) into the Memory field

Advanced Settings

Minimum Memory (MiB):

2048 (use 2048 when 16GB installed ; no need to change even with higher RAM installed)

By setting Minimum Memory to a number lower than Memory, Proxmox will allocate Minimum Memory as dedicated Memory to the VM and, if RAM usage is below 80%, also allocate as much Memory as is available up to Memory, based on

the Memory usage of the other things running on the host.

- Network (tab)

  No Network Device: (leave unchecked)

  Bridge: (leave as is - system generated)

  VLAN Tag: (leave as is - system generated)

  Firewall: (leave unchecked for now at least and use host's firewall setup)

  Model: VirtIO (paravirtualized (leave as is)

  MAC address: (leave as is)

  Advanced Settings

  leave all as is

- Confirm

  Do NOT CLick Start when finished (let the installation finish completely first)

  Validate and click Finish

- Select the newly created VM

  Click on Start - which will initiate the installer disk and start the installation of Debian 10 in this LV

  Wait for a while; it takes some time for the ISO installer to initialize

  The CPU usage (shown in Summary for the VM) will eventually go down

  Click on Console and follow the instructions to install Debian just like you would do on a bare machine

  See the instructions above on Installing Debian

  Note that when the installer asks to write to the entire disk, say yes

  With an LV it means only that portion of the disk assigned to the LV

  Reboot the VM just like you would a normal install on a dedicated server

  It only reboots the VM, not the whole machine

  Make sure to take the install DVD out before reboot progresses or it will attempt another install

c) Do the Debian standard configurations as shown in the Debian install instructions, but do NOT install any Mail utility (Postfix or msmtp) and do NOT install Logwatch. Do these after installing Wazo since Wazo installs and creates a custom configuration for Postfix.

- Do NOT install unattended-upgrades on the Wazo system. Use wazo-upgrade (wazo-upgrade -d and then wazo-upgrade) instead since the OS and PBX applications are closely tied and getting them out of synch can break the system
- Come back after installing Wazo and then
  - Configure Postfix
  - Install and configure Logwatch

d) Add the nfs client and point to the datashare storage, making sure the client IP address matches the

one allowed by the host's export setup

apt install nfs-common [common may be already installed; may not need to install]

- Install and configure nfs client on the client

  cat /proc/filesystems | grep nfs

  To see if the nfs file system has been installed on the system

  On the VM client, using the standard Debian install, it was not, and it is needed, so we run

  modprobe nfs

  and then test again with cat /proc/filesystems | grep nfs

- Create a mount point for the volume in the client that will point to the volume being "exported" from the host via nfs

  mkdir -p /mnt/pve/datashare

  chmod 755 -R /mnt/pve/datashare

  cd /mnt/pve/datashare

  chown {UserName}:{UserName}-R . [do not forget the period at the end]

  - ♦ Repeat above for each shared drive like the disk2 drives (IF they were created; not done in development system):

    mkdir -p /mnt/pve/disk2-archives

    mkdir -p /mnt/pve/disk2-mintrchome

- Mount the directory in this VM pointing to the volume in the Proxmox host at IP address {IP address of host (Proxmox) server}

  - ◊ Manual mount (do NOT do this) would require inclusion of a flag to tell nfs to use a new nfs protocol version (v4) so the manual version would look like

    mount -t nfs -o nfsvers=4 {IP address of host (Proxmox) server}:/var/lib/vz/datashare /mnt/pve/datashare

  - ◊ We want a mount that will survive reboot so we add to the fstab file

    nano /etc/fstab and add, at the end of the file

    # Mount the directory in this VM pointing to the volume in the host at IP address {IP address of host (Proxmox) server}

    {IP address of host (Proxmox) server}:/var/lib/vz/datashare /mnt/pve/datashare nfs4 defaults,sync 0 0

    mount -a

  - ◊ To use this shared drive, access it via /mnt/pve/datashare and treat it like any other directory

    - ♦ Note that it is an nfs drive so items written to it are cached and may not immediately appear as created

  - ◊ In LinuxMint, to add the bookmark for this shared nfs volume in the File Manager

    - ♦ In a Root-level File Manager, it appears under Devices so nothing needing to be done there

- ♦ In a normal (user-level) File Manager:

  Launch File Manager

  To the left of the Location field you will see a pencil icon

  Click on it to toggle between button and text-based location bar

  When in text-based location bar, manually enter /mnt/pve/datashare and <Enter>

  Save as a bookmark by clicking Bookmarks (Menu) -> +Add Bookmark

e) Install WazoPBX

- Reboot the server to get fresh start
- Wazo is an open-source Soft PBX (Private Branch Exchange) system that started life as Xivo but which morphed into Wazo when the owners and developers of Xivo disagreed about the future of the company. The main developers left Xivo, formed Wazo and have since gone through significant evolution of the platform. In my experience with them the Wazo developers are really good people, devoted to making a solid, reliable, open-source PBX, so I support their efforts.
- For details, documentation ,… see

  https://wazo-platform.org

  https://wazo-platform.org/uc-doc

  https://wazo-platform.org/install

  https://wazo-platform.org/documentation

  https://github.com/wazo-platform [for the extra-technical oriented]

  https://github.com/wazo-platform/wazo-ansible/blob/master/README.md#variables

  ◊ Wazo versioning is not like other software packages. With Wazo, the first two digits represent the year and the next digits represent the version of software in that year. So a change from Wazo v19.17 to Wazo v20.01 does NOT mean a major upgrade, it just means we are in a new year.

  - ♦ To show the current wazo version

    cat /usr/share/wazo/WAZO-VERSION

- The instructions here are for a basic system that can handle a home / home-office setup with multiple trunks, multiple devices (phones), conferencing, paging, parking, schedules and other features useful for that setup. It does NOT include cluster setup nor does it use any IVR functionality.
- Wazo is not installed like other applications. There is no ISO to download and no installer disk. Wazo uses GIT, a repository used by developers, to maintain the code and Wazo is installed directly from GIT.
- Install the applications required for installing Wazo from GIT

  cd /

  apt install -yq sudo git ansible

- Fetch and extract the Wazo Platform installer

  cd /

git clone https://github.com/wazo-platform/wazo-ansible.git

This installs a directory in the current directory

wazo-ansible

cd /wazo-ansible

ansible-galaxy install -r requirements-postgresql.yml

This installs a directory and file in /root along with other libraries elsewhere

.ansible_galaxy (directory)

.ansible (file)

- The above installs a version of asterisk that does the basics and may provide all the Asterisk functionality you need (It was all I needed). However, there are numerous asterisk modules that are not installed (see the list at http://documentation.wazo.community/en/stable/upgrade/upgrade_notes.html). If you want the additional modules (I did not), you need to also run

  wazo-asterisk-extra-modules

  - and then manually enable each module you want to utilize by editing

    /etc/asterisk/modules.conf

- Edit the system preferences and passwords to suit your installation
  - ◊ view https://github.com/wazo-platform/wazo-ansible/blob/master/README.md#variables
  - ◊ The default for many variables listed above are shown at (do not change these there)

    /wazo-ansible/roles/wazo-vars/defaults/main.yml
  - ◊ Now make the changes

    cd /wazo-ansible/inventories

    nano uc-engine

    Change (only if your Wazo server is reachable via Public IP; otherwise leave the line as)

    localhost ansible_connection=local

    If your Wazo server is reachable with a public IP address as opposed to a NATed address, you can use the {hostname_FQDN} and leave off the "ansible_connection=local" which lets the ansible playbook run over ssh instead of locally. If your Wazo server is NATed, do as shown above. To get the {hostname_FQDN} at the CLI enter

    hostname -f

    uncomment # [uc-ui:children] [to enable Wazo Web UI ; 1 of 2]

    uncomment # uc-engine-host [to enable Wazo Web UI ; 2 of 2]

    [Add the following two lines in the [uc-engine:vars] section to create a root account at installation. If you do decide to change any other variables, put the entries here.]

    engine_api_configure_wizard = true

engine_api_root_password = {wazo-root-password}

uncomment # wazo_distribution = pelican-buster [to install stable, not dev version]

uncomment # wazo_distribution_upgrade = pelican-buster [to upgrade stable, not dev version]

Uncomment # postgresql_superuser_password = and add a {postgresql_superuser_password} after the = sign (with a space separator)

- Launch the installation

  cd /wazo-ansible

  ansible-playbook -i inventories/uc-engine uc-engine.yml

  ◊ Note that if you ever get to the point where you need to start over, you are supposed to be able to just enter

  wazo-reset

  ♦ and then re-run

  cd /root/wazo-ansible

  ansible-playbook -i inventories/uc-engine uc-engine.yml

  ♦ But I found inconsistent results with this, so instead, as you progress, using the Proxmox GUI, backup the Waxo VM and restore to the last functional one if you need to.

- If you want to configure and manage your Wazo installation with the Web User Interface (UI) - I did - install the web UI modules (newer versions may already have this installed).

  apt update

  apt install wazo-ui

- The above adds a new repository: /etc/apt/sources.list.d/wazo-dist.list

  ◊ If you want to freeze the version of Wazo you have installed, comment out the line

  deb http://mirror.wazo.community/debian/ pelican-buster main

  ♦ in that file. This is NOT recommended, since you want to keep Wazo current with any security and bug fixes published, but some like to lock down a working version to prevent regression issues, so it is documented here.

  ◊ Another way to freeze to a specific version of Wazo (say Wazo 20.02) is to use a specific wazo-upgrade commands to only update the OS and NOT Wazo

  wazo-dist -a wazo-20.02

  wazo-upgrade

  wazo-dist -m pelican-buster

  ◊ You can also customized the configuration of unattended upgrades to ensure Debian is updated but not Wazo, when the unattended-upgrade daemon runs. See the unattended-upgrades section of the Debian install instructions.

- Configure Postfix to only send outgoing mail, not receive mail from outside

  ◊ This may or not already be installed but run this to make sure

apt install libsasl2-modules

- ◊ Postfix must be configured AFTER installing Wazo since Wazo overwites the default Debian mail setup. The configuration described below is set up to enable the system to send out emails (system notices, …) to a designated administrator email account. There is no inbound email processing in this configuration, intentionally so for security purposes. See http://www.zulius.com/how-to/set-up-postfix-with-a-remote-smtp-relay-host for a more detailed explanation of Postfix setup when used to send email with an SMTP authentication. You can also enter the following at the CLI for detailed information about Postfix configuration … enter quit to leave the man pages display

  man postfix

  man postconf

  man 5 postconf

  (go directly to the section summarizing the configuration options and methods)

- ◊ Use the instructions in the "Install Debian" section to configure Postfix under Wazo

- Confirm the Timezone is set properly
  - ◊ At the CLI enter

    date

    - ♦ If the server timezone is not set properly (it should have been on initial install) go back and set it for the timezone you want and restart the server

- Edit the file for CallerID management to add a North American entry for calls that initiate with 1
  - ◊ With my trunk provider, if you leave the initial call with a 1, it sends the call outside the provider's network and costs me more, so I also added a configuration to strip out the initial 0 or 1 (if it was used) on 10-digit calls.

    cp -a /etc/xivo/asterisk/xivo_in_callerid.conf /etc/xivo/asterisk/xivo_in_callerid.conf.orig

    nano /etc/xivo/asterisk/xivo_in_callerid.conf

    - ♦ Add, to the start of the file, after the explanatory comments:

      [northamerican]

      comment = local number within North America - 10 digits, not including leading 1

      callerid = ^[1-9]\d{9}$

      strip =

      add =

      [northamerican2]

      comment = local number within North America - 10 digits, with leading 0 or 1; my provider does not want leading 1 for long distance (yours might)

      callerid = ^1[1-9]\d{9}$

      strip = 1

      add =

- Edit the default ring tones for Aastra phones so we can set the differential ring tones in the Aastra .cfg files
  - ◊ I have left this in the document, even though it makes no sense, since the Change from and Change to are identical. I know I did something with this in a previous installation but I am thinking I did a copy/paste without updating the paste, so I will leave this here in case it comes back to me later. For now, skip this.

    cp -a /etc/xivo/asterisk/xivo_ring.conf /etc/xivo/asterisk/xivo_ring.conf.orig

    nano /etc/xivo/asterisk/xivo_ring.conf

    Change from

    [aastra]

    phonetype = aastra

    intern = <Bellcore-dr1>

    extern = <Bellcore-dr2>

    group = <Bellcore-dr3>

    forward = <Bellcore-dr4>

    to

    [aastra]

    phonetype = aastra

    intern = <Bellcore-dr1>

    extern = <Bellcore-dr2>

    group = <Bellcore-dr3>

    forward = <Bellcore-dr4>

- Wazo uses the nginx web server, not apache. nginx is viewed as better for transaction-oriented applications as opposed to content-oriented applications, so for a phone pbx, Wazo determined that nginx was better. For documentation on nginx go to

  https://www.nginx.com/resources/wiki/start

  https://wiki.debian.org/Nginx/DirectoryStructure

  https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls

- Wazo CLI commands
  - ◊ Here are some of the Wazo CLI commnds you will use

    wazo-service status

    wazo-service is used to control and print the status of the Wazo services.

    cat /usr/share/wazo/WAZO-VERSION

    To show the current wazo version

    wazo-upgrade -d

    To minimize downtime for the Wazo during upgrades, download the upgrade packages first and then run wazo-upgrade. This way, the actual upgrade process is

not keeping the system offline while the packages are being downloaded and you do not risk an interruption to the upgrade mid-upgrade due to a network failure.

wazo-upgrade

when only looking to upgrade wazo packages (NOT Debian) to a later version

When you run wazo-upgrade, it will show you what your current version is and the version to which you would be upgraded if you proceeded - and then give you a y/n choice

wazo-dist-upgrade [when upgrading a Debian system]

*** Must only be run AFTER running wazo-upgrade ***

- (Bug-Fix Patch foor v20.06) When entering the Transport for a Trunk, you are told "Transport was not found"
  - ◊ As reported on Mattermost and registered as a bug at https://wazo-dev.atlassian.net/browse/WAZO-1740 , a regression bug kicked in for Wazo 20.06. This is registered as fixed in v20.07.
  - ◊ You now need to enter transport-udp (all lowercase) and the edit provided below will make things work again.
  - ◊ I am not sure what happens when the bug is fixed. Do we need to go back in and change the workaround fix? Time will tell.
  - ◊ For now, here is how to make it work - by entering transport-udp instead of just udp?
    - ♦ In file /etc/asterisk/pjsip.d/01-wazo.conf
    - ♦ Change

      #exec /usr/bin/wazo-confgen asterisk/pjsip.conf --cached
    - ♦ to

      #exec /usr/bin/wazo-confgen asterisk/pjsip.conf --cached | sed 's/transport-transport-/transport-/'
- (Optional) Use this patch only if you want to use star-codes to transfer calls - may not be needed by the time you read this)
  - ◊ There is an existing (and known) issue with the current (20.06) and previous versions of Wazo not enabling star-codes by default or providing a Web UI way to enable them. See https://wazo-dev.atlassian.net/browse/WAZO-1467 for the bug report. Until the Wazo team has a fix, this is a patch you can implement using the Wazo custom configuration tools which WILL survive an upgrade.
  - ◊ In addition, there is a patch needed to ensure that the handling of calls coming in through a Group maintains the Group settings for these calls, so there is an edit needed to the [usersharedlines] context in /usr/share/xivo-config/dialplan/asterisk/extensions_lib_user.conf which is accomplished by overriding that context with a replacement entry in the file shown below.
  - ◊ There are star-codes (*1 = Blind Transfer and *2 = Attended Transfer) shown in the Wazo Web GUI setup that are supposed to allow a user to use star-codes during a call to transfer the call to another user/extension. Unfortunately, the system has to enable dtmf transfers for the star-codes to work, and the default setting disables dtmf transfers.

◊ The DialPlan options to enable dtmf transfer is the letter t. Adding the lowercase t to the DialPlan enables the person being called (the callee) to transfer the call. Adding the uppercase T to the DialPlan enables the calling party (the caller) to transfer the call. You do NOT want to enable the caller to transfer the call since that would allow hackers to dial into your voice mail and transfer the call to a long-distance call (as an example) so only add the lowercase t. In some special cases, like a remote worker who is called via the pre-mobility subroutine who wants to transfer the call to a co-worker, there may be a reason to enable the calling party to make a transfer so in those cases, enable T but take other precautions in how you enable it (not on a global basis).

◊ There are separate variables and subroutines to use. One enables User-based calls (internal User-to-User and Inbolund calls directly to a User). Another enables Group-based calls (calls to a Group which has members and members of the group pick up the call). If you use Groups, you need to enable both.

◊ The setting to enable dtmf transfers is available if you are using the API but not with the Web GUI. If you use the API to enable dtmf transfers and then make changes with the Web GUI, you may undo the API changes, so that is not a good solution.

◊ You can enable dtmf transfers for all users or selected users. I chose to enable it for all users, but if you want, you can do it for selected users. I will show you the patch that enables dtmf transfer for all users and identify how to change that to enable dtmf transfers for only selected users.

◊ For reference, if you want to debug a Subroutine, add same => n, DumpChan() to a line and it will output all available variables.

- In the Wazo documentaiton (PDF) for Wazo 20.03 under the heading "Dialplan variables" (page 260), you will find a partial list of Channel Variables. Search for one of the variables (eg XIVO_SRCNUM) or the heading "Dialplan variables" if the PDF is updated and you want to find the new location/page number.

◊ In /etc/asterisk/extensions_extra.d add a file named star-codes.conf (you can call it whatever you want as long as it ends in .conf)

◊ Note the underscore ("_") as a prefix to the variable names _XIVO_CALLOPTIONS and _XIVO_GROUPOPTIONS in the Dialplans. It is not strictly needed (but does not hurt) in the [xivo-subrgbl-user] context but it is needed in the [xivo-subrgbl-group] context. By adding an undersacore as a prefix to a variable name when the variable is set, it declares that the variable name and its value is to be maintained beyond the context in which it is declared. For the [xivo-subrgbl-group] context, the XIVO_GROUPOPTIONS variable is to be used in the [usersharedlines] context so the underscore prefix is required for the XIVO_GROUPOPTIONS variable.

◊ nano /etc/asterisk/extensions_extra.d/star-codes.conf

```
[xivo-subrgbl-user]

exten = s,1,NoOp(Enabling users (Callee=t, NOT Caller=T) to transfer calls using star-codes for Blind or Attended transfer)

same  = n,Set(_XIVO_CALLOPTIONS=t${XIVO_CALLOPTIONS})

same  = n,Return()


[xivo-subrgbl-group]
```

```
exten = s,1,NoOp(Enabling members of a Group (Callee=t, NOT Caller=T) to transfer calls using star-codes for Blind or Attended transfer)
same  = n,Set(_XIVO_GROUPOPTIONS=t${XIVO_GROUPOPTIONS})
same  = n,Return()


[usersharedlines]
; 32 [0-9a-f] is too long and - are ignored
exten = _[0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f][0-9a-f].,1,AGI(agi://${XIVO_AGID_IP}/get_user_interfaces,${EXTEN})
same  = n,NoOp(Adding the XIVO_GROUPOPTIONS to the Dial command so star-codes will work for call into a Group)
same  = n,Dial(${WAZO_USER_INTERFACES},,${XIVO_GROUPOPTIONS})
same  = n,Hangup
```

- The only thing to be careful of is that there are no other files in /etc/asterisk/extensions_extra.d that invoke the [xivo-subrgbl-user] global subroutine. There are none in the default Wazo setup so unless you add one elsewhere, you do not have an issue. By using the [xivo-subrgbl-user] global subroutine you do not need to add anything to each user's setting to enable dtmf transfers.
- Observant readers will see that I also added a DialPlan option "r" to force the system to play ring tones to the caller, not MoH while the call is awaiting pickup.

  Edit the default ring that a caller hears if you force the ring (not sound files) with r in the Asterisk DialPlan

  The default ring used when r is added to the DialPlan is the French one. This, for North American callers is close to the Busy tone so I changed it back to the normal NA ring tone.

  nano /etc/asterisk/indications.d/99-general.conf

  Change

  ```
  [general]
  country=fr
  ```

  to

  ```
  [general]
  country=us
  ```

◊ Edit the file permissions to allow Asterisk and the Web GUI to access this new configuration

```
chown asterisk:www-data /etc/asterisk/extensions_extra.d/star-codes.conf
chmod 660 /etc/asterisk/extensions_extra.d/star-codes.conf
```

◊ Reload your dialplan with

asterisk -rx "dialplan reload"

or just reboot your server

◊ All users now have the ability to transfer calls using star-codes when they are the recipient of a call.

*1 = Blind Transfer

*2 = Attended Transfer

◊ If you want only selected users to have the ability to transfer calls using star-codes, change the name of the subroutine from the global subroutine [xivo-subrgbl-user] (which automatically works and does not have to be added to user settings) to something else like [allow-transfer] (a user-defined subroutine that has to be called to be in effect) and then add this subroutine name to the subroutine field in selected user's configuration in:

♦ User -> General (tab) -> Subroutine (field)

Leave the [] brackets out when entering the subroutine name in the Subroutine field.

- (Optional) Use this patch only if you want to use star-codes to initiate recordings of a call while you are in the call (as opposed to enabling call recording before the call is started, which already works) - may not be needed by the time you read this

◊ If you want the allow a line/extension to have ALL calls to/from that line/extension recorded, in Advanced -> Extensions Features

♦ Check the checkbox beside callrecord to enable it

This will activate Call Recording for all calls on the line/extension on which you entered the star-code - use this star-code BEFORE you make a call and thereafter ALL calls in this line will be recorded.

This is done for each line/extension on which you want calls to be recorded.

This is a toggle feature: using it once activates call recording for every call on that line/extension and using it again deactivates call recording on that line/extension.

♦ Recordings are stored in /var/lib/wazo/sounds/tenants/{UserUUID}/monitor

The format of the file name is user-{srcnum}-{dstnum}-{UTC_time}.wav

◊ To use star-codes to initiate recordings of a call during a call, there is an existing (and known) issue with the current (20.06) and previous versions of Wazo. See https://wazo-dev.atlassian.net/browse/WAZO-1467 for the bug report. Until the Wazo team has a fix, this is a patch you can implement using the Wazo custom configuration tools which WILL survive an upgrade.

◊ The DialPlan options to enable online recording of calls is the letter X. Adding the lowercase x to the DialPlan enables the person being called (the callee) to transfer the call. Adding the uppercase X to the DialPlan enables the calling party (the caller) to transfer the call. Since you could be either the called party or the calling party, you want both enabled so use xX to the DialPlan.

◊ There is a star-code (*3 = automixmon) shown in the Wazo Web GUI setup that is supposed to allow a user to use *3 during a call to initiate recording of the call. Unfortunately, the system has to set online_call_record_enabled to true for the star-code to work, and the default setting for this is false.

- ◊ The ability to set online_call_record_enabled to true is available if you are using the API but not with the Web GUI. Both soutions below have been tested and does survive subsequent changes using the WEeb GUI so you can use either one if you want.
- ◊ You can enable online call recording for everyone or user-by-user.
- ◊ The solution to enable online call recording for all users is similar to the one above for enabling dtmf transfer and changes the diaplan settings. It works, but is not as "clean" as the API solution.
- ◊ The solution to enable online call recording user-by-user via APIs is explaind below. It is more complicated but you have more control over where the recordings are stored and what the filename looks like.
- ◊ You would normally use ONE solution OR the other, NOT both. You can use both by using the API for a specific user and the diapplan changes for the general population, but it is likely to cause confusion with different directories for storage of recordings so it is not recommended.
- ◊ To use changes to the dialplan and have the changes work for all users
  - ♦ In /etc/asterisk/extensions_extra.d edit (if you did the changes above for dtmf transfer) or add (if you did not do the changes above for dtmf file transfer) a file named star-codes.conf (you can call it whatever you want as long as it ends in .conf)
  - ♦ nano /etc/asterisk/extensions_extra.d/star-codes.conf

    If you already had created the file for enableing dtmf transfers

    [xivo-subrgbl-user]

    exten = s,1,NoOp(Enabling users (Callee=t, NOT Caller=T) to transfer calls using star-codes for Blind or Attended transfer)

    same = n,Set(_XIVO_CALLOPTIONS=t${XIVO_CALLOPTIONS})

    same = n,NoOp(Forcing the system to generate ringing for calling parties until the call is answered = No MoH while awaiting pickup)

    same = n,Set(_XIVO_CALLOPTIONS=r${XIVO_CALLOPTIONS})

    same = n,NoOp(Enabling online call recording by the so users can initiate call recording during a call)

    same = n,Set(_XIVO_CALLOPTIONS=xX${XIVO_CALLOPTIONS})

    same = n,Set(XIVO_CALLRECORDFILE=)

    same = n,Return()

    [xivo-subrgbl-group]

    exten = s,1,NoOp(Enabling members of a Group (Callee=t, NOT Caller=T) to transfer calls using star-codes for Blind or Attended transfer)

    same = n,Set(_XIVO_GROUPOPTIONS=t${XIVO_GROUPOPTIONS})

    same = n,NoOp(Forcing the system to generate ringing for calling parties until the call is answered = No MoH while awaiting pickup)

    same = n,Set(_XIVO_GROUPOPTIONS=r${XIVO_GROUPOPTIONS})

same = n,NoOp(Enabling online call recording so members of a Group can initiate call recording during a call)

same = n,Set(_XIVO_GROUPOPTIONS=xX${XIVO_GROUPOPTIONS})

same = n,Set(XIVO_CALLRECORDFILE=)

same = n,Return()

[usersharedlines]

; 32 [0-9a-f] is too long and - are ignored

exten = _[0-9a-f].,1,AGI(agi://${XIVO_AGID_IP}/get_user_interfaces,${EXTEN})

same = n,NoOp(Adding the XIVO_GROUPOPTIONS to the Dial command so star-codes will work for call into a Group)

same = n,Dial(${WAZO_USER_INTERFACES},,${XIVO_GROUPOPTIONS})

same = n,Hangup

If you are creating a new file to just handle online recording

[xivo-subrgbl-user]

exten = s,1,NoOp(Enabling online call recording by the so users can initiate call recording during a call)

same = n,Set(_XIVO_CALLOPTIONS=xX${XIVO_CALLOPTIONS})

same = n,Set(_XIVO_CALLRECORDFILE=)

same = n,Return()

[xivo-subrgbl-group]

exten = s,1,NoOp(Enabling online call recording so members of a Group can initiate call recording during a call)

same = n,Set(_XIVO_GROUPOPTIONS=xX${XIVO_GROUPOPTIONS})

same = n,Set(_XIVO_CALLRECORDFILE=)

same = n,Return()

[usersharedlines]

; 32 [0-9a-f] is too long and - are ignored

exten = _[0-9a-f].,1,AGI(agi://${XIVO_AGID_IP}/get_user_interfaces,${EXTEN})

same = n,NoOp(Adding the XIVO_GROUPOPTIONS to the Dial command so star-codes will work for call into a Group)

same = n,Dial(${WAZO_USER_INTERFACES},,${XIVO_GROUPOPTIONS})

same = n,Hangup

- When you are finished, check to ensure the file permissions are as follows

  chown asterisk:www-data /etc/asterisk/extensions_extra.d/star-codes.conf

  chmod 660 /etc/asterisk/extensions_extra.d/star-codes.conf

◊ Reload your dialplan with

  asterisk -rx "dialplan reload"

- or just reboot your server

◊ The files will be stored in

  /var/spool/asterisk/monitor

  and the file name will be

  auto-{timestamp}-(dstnum}-srcnum}

  Note that this means this solution will NOT work well in a multi-tenant installation since all online call recordings are stored in a common space. If it is a simple home office setup like mine with only one "tenant", not a problem.

◊ To use the Wazo API to make this change for user {Username}:

- Using the Wazo Web GUI, in User -> User (tab) -> Enable Authentication (section)

  Check Enable authentication

  Set the Username (probably already set)

  Set the Password

  Update

- Using the Wazo Web GUI, get the User UUID (which we will refer to below as {USER_UUID}with User -> Select the User -> Click the Edit (pencil) icon

  Copy the URL shown in the browser bar. It will contain the {USER_UUID} as shown below:

  https://{server_address}/engine/users/{USER_UUID}?{a_lot_more_stuff}

  Extract the USER_UUID from the full URL

- From the CLI you use, login to the Wazo server as superuser

  Generate a token that will authenticate you, as Username, so you can interact with Wazo and change configurations for that user (the token is temporary, set to last 1 hour from time of creation, so do these steps within one sitting)

  wazo-auth-cli token create --auth-username {Username} --auth-password {Password}

  Copy the token that is returned; we will refer to this below as {TOKEN_UUID}

  If you want to confirm you have the right token for the right user, you can enter:

wazo-auth-cli token show {TOKEN_UUID} | grep "xivo_user_uuid"

Change the setting for online_call_record_enabled to true

Since you are logged in to your Wazo server, for the Wazo server IP address you can use 127.0.0.1

curl -kX PUT "https://127.0.0.1/api/confd/1.1/users/{USER_UUID}" -H "accept: application/json" -H "Content-Type: application/json" -H "X-Auth-Token: {TOKEN_UUID}" -d '{ "online_call_record_enabled": true}'

INCLUDE the {} brackets around the last entry after -d but do NOT include the {} brackets for the {USER_UUID } or the {TOKEN_UUID}

Note: I did NOT use the following to enable the other variables, since I wanted them enabled globally and used the dialplan patch above to accomplish the dtmf transfers, but for reference, you could also set them user-by-user in the above command by replacing the contents of the last {} to:

{ "call_record_enabled": true, "call_transfer_enabled": true, "online_call_record_enabled": true}

◊ Unless you change the default settings, recordings will be stored in

/var/lib/wazo/sounds/tenants/{Tenant_UUID}/monitor

♦ And the file name will be

/user-{{ srcnum }}-{{ dstnum }}-{{ timestamp }

♦ The {Tenant_UUID} can be obtained using the command above

wazo-auth-cli token show {TOKEN_UUID} | grep "tenant_uuid"

♦ If you want to customize the location and/or the file name (I did want to change the file name; I prefer to have the files sorted by creation dates so wanted the timestampt first):

Create a file called call_recording.yml in directory /etc/wazo-agid/conf.d/ which will be upgrade safe and will override the default settings

Into that file put

call_recording:

filename_template: "/var/lib/wazo/sounds/tenants/{{ tenant_uuid }}/monitor/call-{{ timestamp }}-{{ srcnum }}-{{ dstnum }}"

◊ Reboot the server

- (Optional) Use this patch if you have satellite locations for your system where people will be calling from a physical street address other than the one where your main IP trunks are e911 registered. If you have such satellite locations, this is legally required in Canada (not sure about other locales). You can also install this if you just want to have certain extensions use a specified trunk (billing purposes, CallerID, …). - may not be needed by the time you read this

◊ When you register a SIP (or other IP-based) trunk with your provider, you must complete an e911 form identifying the physical street address of the system using the trunk. This is in case a User calls 911. With IP phone systems, the physical street address of the User is not identifyable by the IP Address - the physical address of the ISP is the one that could likely come up for a dynamic IP - so any emergency dispatch would be sent to the wrong address. This has actually happened in Canada and a youg person died waiting for the ambulance. So

in Canada (where I am), it is now law that the e911 registration for the trunk reflect the physical street address of the User who will be using the trunk. If you have a trunk registered for the physical street location of your main office where most of your Users will be located and you have a User who works remotely, if that User called 911, unless the person was capable of redirecting the dispatch operator (not always true) and remembered to do so (easily forgotten at times of stress), emergency dispatch would be sent to your main office location. This patch allows you to specify that a certain User's calls to 911 go out a specific trunk. You would then register that trunk's e911 to that User's physical location and all would work as designed.

◊ You can set this solution up so it works for all users (when the criteria match) by editing the global subroutine [xivo-subrgbl-outcall]. This will not require any edits to the User setup and, if the User setup is already calling a Subroutine via an entry in the User's Subroutine field, this may be the preferred solution. I use this and edit the file shown below to determine what User/Extension is affected and when. This will require edits to the file shown below.

◊ You can set this solution up so it only takes effect for specified Users. This will require an edit to the User setup by changing the Subroutine label in the file shown below and adding this label to the User's Subroutine field in the User configuration. This will also require edits to the file shown below.

◊ Edit the lines following the "actual code" line in the file shown below to suit your situation, and put into your system, with the proper permission the content shown, then reload the dialplan (I usually reboot the system but you can just enter, at the CLI, 'asterisk -rx "dialplan reload"' without the single quotation marks)

◊ The file must be placed into /etc/asterisk/extensions_extra.d/ , it must end in .conf and it must have the correct permissions as shown. You can change the filename (before the .conf) if you want. I called the file force-trunk.conf. The label for the subroutine must be [xivo-subrgbl-outcall] if you want it to apply for all users.

```
touch /etc/asterisk/extensions_extra.d/force-trunk.conf
chmod 660 /etc/asterisk/extensions_extra.d/force-trunk.conf
chown asterisk:www-data /etc/asterisk/extensions_extra.d/force-trunk.conf
nano /etc/asterisk/extensions_extra.d/force-trunk.conf
    [xivo-subrgbl-outcall]
    ;
    ; Put this in a file in /etc/asterisk/extensions_extra.d/
    ; with permissions 660 and asterisk:www-data
    ;
    ; This custom handler is used to force an outgoing call out a specified trunk
    ; when a specified CallerID or Extension is the source of the call
    ; and the call is destined for a specified number (eg. 911)
    ;
    ; One example of this is when users in a satellite location dial 911
    ; In that scenario, to make sure the proper e911 service is called, you want to force a specific trunk,
```

; with that trunk properly registered at e911 with the street address of the satellite location.

; The normal Outgoing Calls setup uses designated trunk(s) for ALL calls from ALL extensions

; and the main trunks will be registered to the main physical street address of the system,

; but if one extension is at a different phsyical location / street address

; then 911 calls from that location must (legally in Canada) identify the correct phsyical address

; so the system must have a trunk registered to that location AND

; that trunk MUST be used for 911 calls from that location.

;

; Caveat: there is no failover with 911 calls when using this setup

; If the specified trunk is not avaialable, the outgoing call to 911 fails

;

; To miniimize the risk of failed calls for normal use,

; we also check (in the subroutinne called) to test what the destination number is

; and only force the specified trunk for specified destination numbers (ie 911 calls),

; leaving the normal outgoing call setup with failover options

;

; Thank you to Sylvain Boily and Jonathon Thomas for their help in getting this working

;

; This method identifies a range of extensions to be checked as the originator of a call

; (so for a satellite office with mutliple employees, set them each up with an extension starting with, say, 25)

; and if the originating extension is from within that range and is going to the specified destination number,

; then send the call out the specified trunk

;

; If we had wanted to specify an individual extension

; we could replace

; same => n(test_extension_1),ExecIf($["${XIVO_PICKUPMARK:0:2}" == "25"]?Goto(force_trunk_1,1))

; with

; same => n(test_extension_1),ExecIf($["${XIVO_PICKUPMARK:0:4}" == "2503"]?Goto(force_trunk_1,1))

;

; If we had wanted to specify an individual CallerID

; we could replace

; same => n(test_callerid_1),ExecIf($["${CallerIDNum::0:3}" == "201"]?Goto(force_trunk_1,1))

; with

; same => n(test_callerid_1),ExecIf($["${CallerIDNum}" == "2015551212"]?Goto(force_trunk_1,1))

;

; You could also specify one or more individual extensions or range of extensions by adding more tests and directives to this file

; eg also Force outgoing calls originating from extension 1504 out trunk 2015551212

; exten => s,1,NoOp(Subroutine to force calls from specified sources out specified trunks if going to specified destinations)

; same => n,NoOp(Test for source equal to CallerID)

; same => n(test_callerid_1),ExecIf($["${CallerIDNum::0:3}" == "201"]?Goto(force_trunk_1,1))

; same => n,NoOp(Test for source equal to Extension / Line)

; same => n(test_extension_1),ExecIf($["${PICKUPMARK:0:2}" == "11"]?Goto(force_trunk_1,1))

; same => n(test_extension_2),ExecIf($["${PICKUPMARK:0:4}" == "1504"]?Goto(force_trunk_2,1))

; same => n,Return()

; exten => force_trunk_1,1,NoOp(If going to destination 911 then send out trunk 5555551212)

; same => n(test_911),ExecIf($["${XIVO_DSTNUM}" == "911"]?Set(TRUNKEXTEN=${XIVO_DSTNUM}@5555551212)

; same => n,Return()

; exten => force_trunk_2,1,NoOp(If going to destination 911 then send out trunk 2015551212)

; same => n(test_911),ExecIf($["${XIVO_DSTNUM}" == "911"]?Set(TRUNKEXTEN=${XIVO_DSTNUM}@2015551212)

; same => n,Return()

;

; Actual code below - edit to suit your situation

;

exten => s,1,NoOp(Subroutine to direct calls out a specified trunk when the call originates from a specified Extension/CallerID or range of Extensions/CallerIDs and is destined for a specified number like 911)

same => n,NoOp(Test for source equal to Extension / Line or range with first 5

digits as specified)

same => n(test_extension_2),ExecIf($["${PICKUPMARK:0:5}" == "12001"]?Goto(force_trunk_1,1))

same => n,NoOp(Test for source equal to CallerID or range with first 3 digits as specified)

same => n(test_callerid_1),ExecIf($["${CallerIDNum::0:3}" == "201"]?Goto(force_trunk_2,1))

same => n,Return()

;

exten => force_trunk_1,1,NoOp(If going to destination 911 then send out trunk 5555551212)

same => n(test_911),ExecIf($["${XIVO_DSTNUM}" == "911"]?Set(TRUNKEXTEN=${XIVO_DSTNUM}@5555551212)

same => n,Return()

;

exten => force_trunk_2,1,NoOp(If going to destination 911 then send out trunk 2015551212)

same => n(test_911),ExecIf($["${XIVO_DSTNUM}" == "911"]?Set(TRUNKEXTEN=${XIVO_DSTNUM}@2015551212)

same => n,Return()

- (Optional) Use this edit only if you plan to use the "Phone mobile" field in Users->General(tab) AND if you did NOT use "default" as the name of the Internal Context (which I did not). Note: We are talking about the "Name", NOT the "Label" or the "Type" of the Internal Context. - likely still needed when you read this
  - ◊ Edit the [pre-mobility] ([] brackets are to be taken literally) subroutine in the example.conf file in the /etc/asterisk/extensions_extra.d directory
  - ◊ nano /etc/asterisk/extensions_extra.d/example.conf
    - ♦ Change

      same = n,Set(XIVO_INTERFACE=${XIVO_INTERFACE}&Local/${XIVO_MOBILEPHONENUMBER}@default)
    - ♦ to

      same = n,Set(XIVO_INTERFACE=${XIVO_INTERFACE}&Local/${XIVO_MOBILEPHONENUMBER}@{Internal_Context_Name})
    - ♦ Do NOT include the {} brackets when making the replacement
- (Optional) Add blacklist via star-codes to numbers you wish to block from calling Users on your system (known telemarketers, …)
  - ◊ Wazo documented a blacklist feature available in a way that allows administrators to store blacklisted numbers using the asterisk CLI commands and have Asterisk check the blacklist

for incoming calls.

- Look up blacklist in the Wazo guide for details. We use modified Wazo code to monitor incoming calls and determine if the incoming call is a blacklisted number.
- The blacklist is stored in an sqlite database in /var/lib/asterisk/ and can be directly viewed/edited/backed up with standard sqlite tools. I do NOT do this.
- To manipulate the database using asterisk commands, at the asterisk CLI (enter with verbosity of at least 3 for full displays) use

  core show help database

  To see commands available to the admin when at the CLI

  These are NOT the same commands used from within the dialplan

  To see applications available from within the dialplan, while a the system CLI (NOT asterisk CLI) enter

  asterisk -rx "core show functions" | grep database

  asterisk -rx "database show"

  To show all entries in database

  asterisk -rx "database show blacklist"

  To show all entries in the blacklist database

  The asterisk database has a "tree" function with family[/subfamily] key value as its hierarchy, so for example to see the whole blacklist database, use the above command and to see one entry in the database, include the key as in

  asterisk -rx "database show blacklist 2015551212"
- In a large system with many Users, limiting this ability to edit a blacklist to a system administrator may be a safer way to control blacklist since it would prevent accidental (or malicious) entries into the blacklist by Users.
- In my case, with a simple home/office system, I wanted to enable Users to blacklist callers using star-codes and I do NOT want to have to login to an asterisk CLI to add a number to a blacklist.

  There is a patch (above) to enable the use of star-codes on Wazo. This must be implemented in Wazo 20.05 for the star-code blacklist solution shown here to work.

◊ You need some extra sound files to implement the blacklist star-code method described below.

- The sound files we need can be downloaded from the Asterisk web site. See below.
- If you really want to create interesting sound files for different user interactions, there are options, NOT used for this installation, but kept here for reference.
- Wazo's guide shows you how to create your own sound files using Google's free tts tools. See the Wazo guide (as of the creatioin of this document in section 1.10.5) for more information on this.
- Additional Allison sound files are available at (NOT used for blacklist)

  http://downloads.asterisk.org/pub/telephony/sounds/releases/

- Additional (replacement) non-Allison sound files are available at (NOT used for blacklist)

  http://www.voicevector.com/Downloads.php

◊ As an aside, if you want to use one of the sound files as your voicemail greeting (Unavailable or Busy), copy the sound file to /var/spool/asterisk/voicemail/{Internal_Conext_Name}/{voicemail_number} and rename it unavail.wav or busy.wav , depending on when you want it to play.

◊ Thank you to Ward Mundy for inspiration of this solution.

◊ Wazo has installed the "core" asterisk sound files so we only need install the "extra" sound files since they are called by this script - and potentially some other custom script you may want to create. The instructions include both "core" and "extra" for reference, but you need only install "extra".

- ulaw codecs use uncompressed source files like (asterisk-compatible) wav files
- Note that wav is higher quality than gsm (gsm is optimized for mobile) but wav has larger files so needs more bandwidth
- gsm files are stored in /usr/share/asterisk/sounds/en and this is what asterisk uses by default
- wav (asterisk-compatible wav for ulaw) files are stored in /usr/share/asterisk/sounds/en_US
- To verify that a sound file on your system is the type you want, at the CLI enter

  file {filename}

  and the file characteristics will be reflected.

- To add gsm files

  Fetch the sound files

  cd /usr/share/asterisk/sounds/en

  wget http://downloads.asterisk.org/pub/telephony/sounds/asterisk-extra-sounds-en-gsm-current.tar.gz

  NOT needed to be added in Wazo but kept for reference

  wget http://downloads.asterisk.org/pub/telephony/sounds/asterisk-core-sounds-en-gsm-current.tar.gz

  tar xzvf asterisk-core-sounds-en-gsm-current.tar.gz

  Install the sound files and remove the tarball

  tar xzvf asterisk-extra-sounds-en-gsm-current.tar.gz

  rm -f *.tar.gz

  find ./ -type d -exec chmod 755 {} \;

  find ./ -type f -exec chmod 644 {} \;

  chown --recursive root:root .

  Note the period (.) at the end of the command above

- To add wav (asterisk compatible wav for ULAW) files

  Fetch the sound files

  cd /usr/share/asterisk/sounds/en_US

  wget http://downloads.asterisk.org/pub/telephony/sounds/asterisk-extra-sounds-en-wav-current.tar.gz

  NOT needed in Wazo but kept for reference

  wget http://downloads.asterisk.org/pub/telephony/sounds/asterisk-core-sounds-en-wav-current.tar.gz

  tar xzvf asterisk-core-sounds-en-wav-current.tar.gz

  Install the sound files and remove the tarball

  tar xzvf asterisk-extra-sounds-en-wav-current.tar.gz

  rm -f *.tar.gz

  find ./ -type d -exec chmod 755 {} \;

  find ./ -type f -exec chmod 644 {} \;

  chown --recursive root:root .

  Note the period (.) at the end of the command above

◊ Add dialplan entries to enable the blacklist feature

- You could add a Preprocess subroutine to the settings for each incall route and then add that soubroutine to the /etc/asterisk/extensions_extra.d/ directory . This would require you to add the Preprocess subroutine to each incall route. The upside of this is that you could control which incall routes are equipped to blacklist callers. The downside is that you have to remember to add the Preprocess subroutine for each incall route.
- This solution uses the global incall subroutine and adds the star-codes to the system features list, making it in effect for ALL incall routes and all users.
- For all files created or edited, make sure that when you are done, the permissions are set at

  chmod 660

  chown asterisk:www-data

- To add the new star-codes to the Wazo system, in the /etc/asterisk/extensions_extra.d/xivo-extrafeatures.conf file after the [xivo-extrafeatures] line (or in other words, in the xivo-extrafeatures context) add

  ;

  ; Added Features to enable management of the blacklist via star-codes

  ; To add the last caller's CallerID Number to the blacklist

  exten => _*70,1,NoOp(Blacklist function for user to automatically add last incall CallerID to blacklist)

  same => n,Goto(blacklist-manage-add-last,blacklist-add-last-number,1)

; To manually add a number to the blacklist, by entering the number through the phone's keypad

exten => _*71,1,NoOp(Blacklist function for user to manually enter a number to blacklist)

same => n,Goto(blacklist-manage-add,blacklist-add-number,1)

; To manually delete a number from the blacklist, by entering the number through the phone's keypad

exten => _*72,1,NoOp(Blacklist function for user to manually enter a number to remove from blacklist)

same => n,Goto(blacklist-manage-delete,blacklist-delete-number,1)

; To read the list of entries in the blacklist to the User on the phone

exten => _*73,1,NoOp(Blacklist function for user to get a list of numbers in the blacklist)

same => n,Goto(blacklist-manage-list,blacklist-list-numbers,1)

- To add the scripts that makes the new star-codes work, add afile /etc/asterisk/extensions_extra.d/blacklist.conf (with permissions as shown above) and into that file put

;

; These contexts work with the new star-codes added in /etc/asterisk/extensions_extra.d/xivo-extrafeatures.conf

; *70 = Add the last caller's CallerID Number to the blacklist

; *71 = Add a number to the blacklist, by entering the number through the phone's keypad

; *72 = Delete a number from the blacklist, by entering the number through the phone's keypad

; *73 = Say over the phone the count of numbers blacklisted and then read each number blacklisted

;

; The additions to the global incall context [xivo-subrgbl-did] will do two things:

; 1) Store the CallerdID Number of the current caller to the system

; (which will be used by the *70 star-code to blacklist that number if *70 is used

; 2) Use a built-in Wazo application to check the current caller's CallerID Number to see if it is in the blacklist

; and if so, play a (silly, zombie apocalypse) message and hangup on the caller

;

; The remaining contexts are to enable the functionality of the new star-codes

;

[xivo-subrgbl-did]

exten => s,1,NoOp(Store the CallerID of the caller - replaced by the next caller - in the Asterisk database)

same => n,Set(DB(LastCaller/CallerIDnum)=${CALLERID(num)})

same => n,NoOp(Check incoming calls to see if the caller is in the blacklist - if so, play nonsense and hangup)

same => n,GotoIf(${BLACKLIST()}?blacklisted)

same => n,Return()

same => n(blacklisted),Playback(zombies)

same => n,Hangup()

;

[blacklist-manage-add-last]

exten => blacklist-add-last-number,1,NoOp(Add the number of the last CallerID to the blacklist)

same => n,Answer()

same => n,Wait(1)

same => n,Set(TIMEOUT(absolute)=60)

same => n,Set(TIMEOUT(response)=10)

same => n,Set(TIMEOUT(digit)=10)

same => n,Set(LastCallerIDnum=${DB(LastCaller/CallerIDnum)})

same => n,GotoIf($[$["${LastCallerIDnum}" = ""] | $["${LastCallerIDnum}" = "unknown"]]?blacklist-error,InvalidNumber,1)

$

same => n,Playback(last-num-to-call)

same => n,SayDigits(${LastCallerIDnum})

same => n,Background(privacy-to-blacklist-this-number&if-correct-press&digits/1)

same => n,WaitExten()

exten => t,1,NoOp(The user did not provide any response before the response timeout set above)

same => n,Goto(blacklist-error,SorryBye,1)

exten => i,1,NoOp(The user either waited longer than the digit timeout between key presses or input an invalid number)

same => n,Goto(blacklist-error,SorryBye,1)

exten => T,1,NoOp(The user did not finish the input - press # - before the absolute timeout)

same => n,Goto(blacklist-error,SorryBye,1)

exten => e,1,NoOp(A Catch-all error handler in case the above 3 conditions did not catch the fail)

same => n,Goto(blacklist-error,InvalidNumber,1)

exten => 1,1,NoOp(The user confirmed the number with an acceptable input within the allowed time limit)

same => n,Set(ReasonForBlacklist=Added by star-codes)

same => n,Set(DB(blacklist/${LastCallerIDnum})=${ReasonForBlacklist})

;same => n,SayDigits(${LastCallerIDnum})

same => n,Playback(num-was-successfully&added)

same => n,Hangup

;

[blacklist-manage-add]

exten => blacklist-add-number,1,NoOp(User manually adds a new number to the blacklist)

same => n,Answer()

same => n,Wait(1)

same => n,Set(TIMEOUT(absolute)=60)

same => n,Set(TIMEOUT(response)=10)

same => n,Set(TIMEOUT(digit)=10)

same => n,Read(IncomingCallerNumber,enter-num-blacklist&vm-then-pound,0,s,1,)

same => n,GotoIf($[${READSTATUS} != OK]?blacklist-error,InvalidNumber,1)

same => n,SayDigits(${IncomingCallerNumber})

same => n,Background(privacy-to-blacklist-this-number&if-correct-press&digits/1)

same => n,WaitExten()

exten => t,1,NoOp(The user did not provide any response before the response timeout set above)

same => n,Goto(blacklist-error,SorryBye,1)

exten => i,1,NoOp(The user either waited longer than the digit timeout between key presses or input an invalid number)

same => n,Goto(blacklist-error,SorryBye,1)

exten => T,1,NoOp(The user did not finish the input - press # - before the absolute timeout)

same => n,Goto(blacklist-error,SorryBye,1)

exten => e,1,NoOp(A Catch-all error handler in case the above 3 conditions did not catch the fail)

same => n,Goto(blacklist-error,InvalidNumber,1)

exten => 1,1,NoOp(The user confirmed the number with an acceptable input within the allowed time limit)

same => n,Set(ReasonForBlacklist=Added by star-codes)

same => n,Set(DB(blacklist/${IncomingCallerNumber})=${ReasonForBlacklist})

;same => n,SayDigits(${IncomingCallerNumber})

same => n,Playback(num-was-successfully&added)

```
same => n,Hangup
;
[blacklist-manage-delete]
exten => blacklist-delete-number,1,NoOp(User manually deletes an existing number from the blacklist)
same => n,Answer()
same => n,Wait(1)
same => n,Set(TIMEOUT(absolute)=60)
same => n,Set(TIMEOUT(response)=10)
same => n,Set(TIMEOUT(digit)=10)
same => n,Read(CallerNumberToDelete,entr-num-rmv-blklist&vm-then-pound,0,s,1,)
same => n,GotoIf($[ ${READSTATUS} != OK ]?blacklist-error,InvalidNumber,1)
same => n,SayDigits(${CallerNumberToDelete})
same => n,GotoIf($[${DB_EXISTS(blacklist/${CallerNumberToDelete})}]?:blacklist-error,InvalidNumber,1)
same => n,Background(if-correct-press&digits/1)
same => n,WaitExten()
exten => t,1,NoOp(The user did not provide any response before the response timeout set above)
same => n,Goto(blacklist-error,SorryBye,1)
exten => i,1,NoOp(The user either waited longer than the digit timeout between key presses or input an invalid number)
same => n,Goto(blacklist-error,SorryBye,1)
exten => T,1,NoOp(The user did not finish the input - press # - before the absolute timeout)
same => n,Goto(blacklist-error,SorryBye,1)
exten => e,1,NoOp(A Catch-all error handler in case the above 3 conditions did not catch the fail)
same => n,Goto(blacklist-error,InvalidNumber,1)
exten => 1,1,NoOp(The user confirmed the number with an acceptable input within the allowed time limit)
same => n,Set(VarValue=${DB_DELETE(blacklist/${CallerNumberToDelete}}))
same => n,GotoIf($[${EmptyCheck} == "" ]?blacklist-error,InvalidNumber,1)
;same => n,SayDigits(${CallerNumberToDelete})
same => n,Playback(num-was-successfully&removed)
same => n,Hangup
;
```

```
[blacklist-manage-list]
exten => blacklist-list-numbers,1,NoOp(Read the numbers stored in the blacklist over the phone)
same => n,Answer()
same => n,NoOp(Get the number of entries in the blacklist)
same => n,Set(BlacklistArray=${DB_KEYS(blacklist)})
same => n,Set(NumberInBlacklist=0)
same => n(CountLoopStart),While( $[ $["${SET(BlacklistEntry=${SHIFT(BlacklistArray)})}" != ""] ] )
same => n,Set(NumberInBlacklist=$[ ${NumberInBlacklist} + 1 ] )
same => n,NoOp(Current count of Number in Blacklist is ${NumberInBlacklist})
same => n(CountLoopEnd),EndWhile()
same => n,Playback(privacy-blacklisted&the-num-i-have-is)
same => n,SayNumber(${NumberInBlacklist})
same => n,GotoIf($[${NumberInBlacklist} == 0 ]?Hangup)
same => n,Wait(.5)
same => n,NoOp(Speak each number in the blacklist)
same => n,Playback(for-a-list-of&privacy-blacklisted)
same => n,Set(BlacklistArray=${DB_KEYS(blacklist)})
same => n,Set(LoopCounter=0)
same => n(LoopStart),While( $[ ${LoopCounter} < ${NumberInBlacklist} ] )
same => n,Set(LoopCounter=$[ ${LoopCounter} + 1 ] )
same => n,Wait(.5)
same => n,SayNumber(${LoopCounter})
same => n,Playback(is)
same => n,Wait(.5)
same => n,SET(BlacklistEntry=${SHIFT(BlacklistArray)})}
same => n,SayDigits(${BlacklistEntry})
same => n(LoopEnd),EndWhile()
same => n(Hangup),Hangup
;
[blacklist-error]
exten => InvalidNumber,1,NoOp(The user tried to add delete an invalid number)
same => n,Playback(pm-invalid-option)
same => n,Goto(SorryBye,1)
exten => SorryBye,1,NoOp(Play apology and hangup)
```

same => n,Playback(sorry-youre-having-problems&goodbye)

same => n,Hangup

- Make sure the permissions are correct with

  chown asterisk:www-data /etc/asterisk/extensions_extra.d/*.conf

  chmod 660 /etc/asterisk/extensions_extra.d/*.conf

- Reload the dialplan

  asterisk -rx "dialplan reload"

◊ Instructions for use of the blacklist feature

- Numbers can be added and removed from the Blacklist by phone or by using commands on the Asterisk CLI.
- Managing the Blacklist by Phone is supported with the following star-codes:

  *70 = Add the last caller's CallerID Number to the blacklist

  *71 = Add a number to the blacklist, by entering the number through the phone's keypad

  *72 = Delete a number from the blacklist, by entering the number through the phone's keypad

  *73 = Have the system speak to you the count of blacklisted numbers and then read back each blacklisted number
- You can also manage the Blacklist using the Asterisk CLI with the following syntax:

  To add a number

  database put blacklist 9999999999 "the reason they are being blacklisted"

  or, from the system CLI with

  asterisk -rx 'database put blacklist 9999999999 "the reason they are being blacklisted"'

  To delete a number

  database del blacklist 9999999999

  or, from the system CLI with

  asterisk -rx 'database del blacklist 9999999999'

  To delete ALL numbers from the blacklist

  database deltree blacklist

  or, from the system CLI with

  asterisk -rx 'database deltree blacklist'

  To show all numbers currently in the blacklist

  database show blacklist

  or, from the system CLI with

  asterisk -rx 'database show blacklist'
- To Block Anonymous and Restricted Calls, you must access the asterisk CLI:

You can log into the asterisk CLI from the superuser CLI using asterisk -r and then logout using quit

Or you can create a series one-line commands, using asterisk -rx " asterisk cli command" which will each

Initiate an asterisk CLI session

Submit an asterisk command

Exit the asterisk session

- ♦ To add anonymous callers to the blacklist (caveat: many cell phones have anonymous by default so I did NOT do this)

asterisk -rvx 'database put blacklist Anonymous "Blocking calls with no identifier"'

asterisk -rvx 'database put blacklist anonymous "Blocking calls with no identifier"'

asterisk -rvx 'database put blacklist Restricted "Blocking calls with no identifier"'

- (Optional) Use this patch to remove an error message - may not be needed by the time you read this
  - ◊ In the version I installed, when you went to Plugins you were presented with an Error message across the top saying 'wazo_ui.user.UserUI object' has no attribute 'get_instance_config'
  - ◊ When I looked this up, it pointed to a fix that was supposed to have been committeed in git but when I cheked my files, they were the pre-commit version so I updated them and the Error message went away. It involved editing 3 files, as described at https://github.com/wazo-platform/wazo-ui/pull/15/commits/9a3752681403b4ca8bd0d7ab9c5991bbbfd39c4b

/usr/lib/python3/dist-packages/wazo_ui/plugins/authentication/form.py

/usr/lib/python3/dist-packages/wazo_ui/plugins/plugin/templates/wazo_engine/plugin/list.html

/usr/lib/python3/dist-packages/wazo_ui/user.py

cp -a /usr/lib/python3/dist-packages/wazo_ui/plugins/authentication/form.py /usr/lib/python3/dist-packages/wazo_ui/plugins/authentication/form.py.orig

cp -a /usr/lib/python3/dist-packages/wazo_ui/plugins/plugin/templates/wazo_engine/plugin/list.html /usr/lib/python3/dist-packages/wazo_ui/plugins/plugin/templates/wazo_engine/plugin/list.html.orig

cp -al /usr/lib/python3/dist-packages/wazo_ui/user.py /usr/lib/python3/dist-packages/wazo_ui/user.py.orig

nano /usr/lib/python3/dist-packages/wazo_ui/plugins/authentication/form.py

add line

self.user.set_instance(app.config['auth'])

between

self.user.set_tenant(response['metadata']['tenant_uuid'])

return True

to end up with

```
self.user.set_tenant(response['metadata']['tenant_uuid'])
self.user.set_instance(app.config['auth'])


return True
```

nano /usr/lib/python3/dist-packages/wazo_ui/plugins/plugin/templates/wazo_engine/plugin/list.html

whre you see

```
<script src="{{ url_for('.static', filename='js/plugin.js') }}"></script>
<script>
 connect("{{ current_user.get_instance_config().host }}",
      "{{ current_user.get_instance_config().port }}",
        "{{ current_user.get_instance_config().token }}");
</script>
{% endblock %}
```

delete

```
 connect("{{ current_user.get_instance_config().host }}",
      "{{ current_user.get_instance_config().port }}",
        "{{ current_user.get_instance_config().token }}");
```

add

```
<script src="{{ url_for('.static', filename='js/plugin.js') }}"></script>
 connect(
  "{{ current_user.get_instance().host }}",
  "{{ current_user.get_instance().port }}",
  "{{ current_user.get_instance().token }}"
 );
```

so you end up with

```
<script src="{{ url_for('.static', filename='js/plugin.js') }}"></script>
<script>
 connect(
  "{{ current_user.get_instance().host }}",
  "{{ current_user.get_instance().port }}",
  "{{ current_user.get_instance().token }}"
 );
</script>
```

{% endblock %}

nano /usr/lib/python3/dist-packages/wazo_ui/user.py

where you see

```
def set_tenant(self, tenant=None):
    session['instance'] = {}
    session['instance'] = {'remote_host': 'localhost'}
    session['instance']['wazo_tenant'] = tenant

def get_instance(self):
    return session['instance']
```

delete

```
    session['instance'] = {'remote_host': 'localhost'}
```

add

```
def set_instance(self, config):
    session['instance']['host'] = config.get('host', 'localhost')
    session['instance']['port'] = config.get('port', 443)
    session['instance']['token'] = self._token
```

so you end up with

```
def set_tenant(self, tenant=None):
    session['instance'] = {}
    session['instance']['wazo_tenant'] = tenant

def set_instance(self, config):
    session['instance']['host'] = config.get('host', 'localhost')
    session['instance']['port'] = config.get('port', 443)
    session['instance']['token'] = self._token

def get_instance(self):
    return session['instance']
```

- Reboot the server

- (Optional) Add a personalized favicon.ico to your site
  - If you are familiar with apache, you know that the root directory for apache is usually /var/www/html
  - To have your personalized facicon.ico used when connecting over http (NOT https), put a 16px by 16px image file, saved with .ico suffix into the http root directory and restart your browser - you may also have to refresh your browser due to caching.

- ◊ For the Wazo installation of nginx, /var/www/html is the normal root directory for the non-secure (http) root directory, so if we put your favicon.ico file into /var/www/html directory with the correct permissions (user:group = root:root and file access = 644) then URLs in the http directory will display with your favicon in the browser tab.
  - ♦ Use the instructions in the Transfer files from PC to Linux server section of this document for transferring your favicon.ico file from your PC to your Linux server.
- ◊ Look in /etc/nginx/sites-available/default to see where the root directories have been defined and you will see that /var/www/html is configured as the directory for http access (port 80). However, https access (port 443) is not defined there - all the https configuration options are commented out.
- ◊ Since the secure (https) root directory is not treated the same way as the http directory, the favicon will not display when accessing URLs with https - which is all of Wazo.
- ◊ To show the configuration options for this nginx installation, at the CLI enter

  nginx -V
  - ♦ This will show you the configuration options. Additinally, as you can see by the --prefix entry, the options are stored in a file location, which, in this case, the --prefix tells us to look in /usr/share/nginx and in that list you will see that the configuration file for nginx is conf-path=/etc/nginx/nginx.conf . If you also look in /etc/nginx you will see another directory /etc/nginx/conf.d where any customizations go.
  - ♦ Look in /etc/nginx/sites-available/wazo to see where wazo sites are defined and you will see that /var/www/html is defined again as root for http but you will also see that it is defined as root for https. Confusing? It was to me.
- ◊ None of this helped, so instead, I just did a search for the other favicon.ico files on the server with

  find / -name favicon.*
  - ♦ and of the results that came back

    /usr/lib/python3/dist-packages/wazo_ui/static/img
  - ♦ turned out to be the location where wazo looked for the favicon.ico image. So I renamed the existing one to favicon.ico.wazo and copied the one I wanted from /var/www/html (put there above) and I now had the favicon I wanted for both http and https
  - ♦ Somehow, Wazo has configured the https site server to be controlled by python scripts, and I did not have the time or inclination to learn more about this so left it at the above.

- You now have a patched, fully functional Wazo PBX system ready for configuration.
- At this point, I recommend:
  - ◊ rebooting your system to get a fresh reload of everything
  - ◊ (If using Proxmox) create a backup of the system.
    - ♦ Just in case something goes wrong during configuration (perish the thought!), with a backup at this point, you can restart configuration with a fully patched system, instead of going through all the above just to re-do the configuration.

f) (Optional) If you want to use the REST APIs

- I do my configuration using the Web UI (below) but include this section for reference in case there is a desire to programmatically get information about or set parameters for Wazo.
- The APIs are still under development, in particular the v0.1 APIs for conf so check with the current documentation when attempting to use APIs
- As an example, the instructions say that to see the list of Wazo APIs available to you, in your browser, after installing Wazo go to {IP_address_of_Wazo}/api , but this no longer works (v 20.06) so use the instructions below.
- Be careful using APIs; you could easily edit/delete something needed for proper operation of the system.
- Documentation for the use of APIs is available at https://wazo-platform.org/documentation through the "API Console" link. The section you will use the most after installing Wazo is the Configuration section in the API Console.
    - ◊ Do NOT enter your information in the top fields; doing so may invalidate your authentication and prevent you from seeing what is on the pages. By NOT entering your information, you will not be able to Execute your API calls from these pages, but you will be able to use curl to Execute your API calls.
    - ◊ On that page and using the menu options, you will be given the curl options you need to accomplish most things you want via the API call using curl
    - ◊ Because Wazo uses a self-signed certificate, when using a curl command, you need to add the "k" flag to the "X" flag, so the curl command starts with curl -kX not just curl -X
    - ◊ See Wazo documentation starting at https://wazo-platform.org/uc-doc/api_sdk/rest_api/conventions for instructions using the REST API
    - ◊ Unfortunately, Wazo 20.06 broke access to the api token process. In Wazo 20.06 the access changed so instead of using the port designation shown in the quickstart documentation which said to go to https://{Wazo_IP_Address}:9497/0.1/api/api.yml and accept the certificate, you are now supposed to go to https://{Wazo_IP_Address}/api/auth/0.1/api/api.yml (note no port designation and different directory) to validate the web-site self-signed Wazo certificate and then return to https://{Wazo_IP_Address}/api to click on wazo-auth to get the token. But, that is broken in 20.06 so for now, use the processes outlined below.
- You can access the APIs by
    - ◊ a) Using the Web UI which will convert the entries to API equivalents
    - ◊ b) From the system CLI, use the Wazo CLI options: wazo-auth-cli, wazo-provd-cli or wazo-agend-cli
    - ◊ c) From the system CLI (or via SSH into the system) use curl commands to emulate web protocol commands
    - ◊ d) (Not working now with the break that occurred in Wazo 20.06) From a browser by going to https://{Wazo_IP_Address}/api
    - ◊ e) Through programming using the protocols shown in the Wazo developer documentation identified above
- API interactions using options c) d) e) are done using a token for authentication of the user making the request. A token is requested with a specific command and, when the user is

authenticated, the token is issued for that user to use for an hour, after which, if the user is still making API calls, they need to fetch another token.

- To see and use many of the configuration options available for REST commands from the system CLI, use the wazo cli options and run
    - ♦ wazo-auth-cli --help
    - ♦ wazo-auth-cli

      to enter and use the wazo cli

      an analog to asterisk -r used to enter and use the asterisk cli

      to exit

      exit()
    - ♦ wazo-auth-cli {command options}

      to run the Wazo cli command and immediately exit the wazo cli

      an analog to asterisk -rx "{asterisk_command}" used to run an asterisk comman and immediately exit the asterisk cli
    - ♦ This is the one you would likely use the most
    - ♦ In the --help option you will see all the options available to you (create, delete, show, list, …) for user, token, policy, goup, … configurations
    - ♦ You can use this cli or the curl cli shown below or the programming options documented in the Wazo online documentation to interact with, fetch data from and configure Wazo
    - ♦ For the most part, this functionality is available via the Web UI, but the CLI option is available if you prefer and is available for bash scripting if you want.
    - ♦ The curl option (shown below) and the direct programming option allow you to create php or other programs to access the Wazo system

  ◊ For accessing provisioning modules

    - ♦ wazo-provd-cli --help

      wazo-provd-cli (to enter and use the cli)

      to exit

      exit()

      is a command-line interface to interact with the REST API of wazo-provd. It provides mainly provisioning-related features.

      has an interactive REPL mode. It should prompt you for a password that is empty by default. Once in the interactive mode, enter help for a list of available operations.

  ◊ For accessing agent modules used by call center agents

    - ♦ wazo-agentd-cli --help

      wazo-agentd-cli (to enter and use the cli)

      to exit

      exit()

is a command-line interface to interact with the REST API of wazo-agentd. It provides mainly agent-related features.

has an interactive REPL mode. It should prompt you for a password that is empty by default. Once in the interactive mode, enter help for a list of available operations.

- To use curl commands
  - ◊ Create and use a (limited permission) API User for API calls
    - ♦ You can create a (limited privileges) user in the Wazo UI in User->General->Enable Authenticate, Check the Autheticate Checkbox, and enter the {API_User_Name} and {API_User_Password}
    - ♦ Due to some recent Wazo changes, the first option, using the Wazo CLI, is the one that works reliably; the option explained below of executing code directly from the web form at https://wazo-platform.org/documentation => Configuration => API Console is not as reliable but the documentation is left here for future reference and code samples.
    - ♦ Use the CLI of the server by entering

      wazo-auth-cli token create --auth-username {API_User_Name} --auth-password {API_User_Password}

      This will return a Token_UUID. Copy this and save it for the remaining curl commands.

      Tokens last 1 hour so if you are working with APIs longer than an hour, fetch a new token for the time past 1 hour.

      See example curl scripts at https://wazo-platform.org/documentation => Configuration => API Console.

      Since we are working with a Wazo server with a self-signed certificate so we need to add the "k" flag (insecure) to the curl command, not just the "X" flag

      You need to also edit the examples shown for the API Console, to use the loopback IP address 127.0.0.1 instead of the Wazo_IP_Address; this is what changed in Wazo 20.06 when using curl commands from the server CLI

      To show what this token will allow you to do, add the Token_UUID to the curl command:

      curl -kX GET "https://127.0.0.1/api/auth/0.1/token/{Token_UUID}" --header "accept: application/json"

      Since you are using the Token for the actual Wazo User, you can get the User_UUID by looking at the entry after xivo-user-uuid in the reponse from the above command.

      To retrieve your User information, use the Token_UUID for authentication and

      curl -kX GET "https://127.0.0.1/api/confd/1.1/users/{User_UUID}" --header "accept: application/json" --header "X-Auth-Token: {Token_UUID}"

      Another way to get the User_UUID (in case you are using an elevated privileges token - see below), use the Wazo Web UI, Click on the User, and copy the URL, extracting the User_UUID from the URL elements.

      To retrieve information on a Group to which a User belongs, look up the group_id from the above result and then enter

      curl -kX GET "https://127.0.0.1/api/confd/1.1/groups/{Group_ID}" --header "accept:

application/json" --header "X-Auth-Token: {Token_UUID}"

To retrieve a list of Users

curl -kX GET "https://127.0.0.1/api/confd/1.1/users" --header "accept: application/json" -H "X-Auth-Token: {Token_UUID}"

You will see only a one-line response reflecting the Token_UUID, since this user does not have Group privileges. See below for elevated privileges.

All the above fetched information using the GET curl command. You can also CREATE, using the POST curl command or EDIT using the PUT curl command. I am NOT recommending this for beginners, simpley showing one example to demonstrate the PUT command. I prefer the Wb UI for it's aimmediate feedback, but this is to show you more is possible with APIs. For example, if you wanted to change the fallback greeting for a Groups (say Group ID 2) voicemail (say VM ID 1) to busy instead of (say) unavailable, you could enter

curl -kX PUT "https://127.0.0.1/api/confd/1.1/groups/{Group_ID}/fallbacks" --header "accept: application/json" --header "Content-Type: application/json" --header "X-Auth-Token: {Token_UUID}" -d '{ "noanswer_destination": { "type": "voicemail" , "voicemail_id": {VM_ID} , "greeting": "unavailable" } }'

- Although you cannot directly execute the commands provided by the API Console at https://wazo-platform.org/documentation => Authentication => API Console (missing the -k flag and not using the loopback IP address), you can still use the sample code provided there and edit it to use in the syntax shown above. Some examples:

  Click on on of the categories you want to use API calls for and click "Try it out"

  For example, click on the Users heading on the page, which will expand it to show options

  Click on the GET /users/{user_uuid_or_me} to see what is required to retrieve information on the User

  Click the "Try it out" button

  Enter the User UUID in the user_uuid_or_me field

  When you click execute, the code will fail (requiring the above changes shown to the -k flag and the IP_Address) but you will see the syntax of the required curl command to accomplish what you want via an API call.

  Although it now seems easier to use the web page, you will quickly find that it is easier and quicker to use the curl commands from the CLI. You can always come back to this page if you are unsure of the format of the curl command.

◊ Use the root-level user for which you created a password when installing Wazo for API calls

- This has elevated privileges compared to the limited permissions available for the user above. This user is likely all you will need to do the API calls on your system
- When using this, make sure you are specifying the User you are asking for information about / setting parameters for, since you do not want to be configuring root options.

◊ Create an all-access admin-level API User for API calls

- NOT recommended for Production systems due its security risk. Useful for learning while on a development system.

- If you do create this, protect the credentials for this user - or delete it and reinstate it when needed - since this has very powerful abilities / dangers.
- Even if you protect user credentials, it is possible someone courl create another user and apply this unlimited policy created (below) to the new user, giving the new user unlimited power on your system Best to delete the User and Policy documented here and re-create if you need them.
- You can see what policy options already exist by using the Master tenant and going to Wazo UI -> Master Tenant -> Credentials -> Policies
- To create the all-access Policy you will use for your all access User, and to assign this User to this Policy for only the intended Tenant, switch to the Tenant you created for this installation. Again, only use this in development systems and delete it after using it. Too powerful.

  Wazo UI -> {Tenant} -> Credentials -> Policies -> Add (+ icon)

  Name: {All_Access_ACL_Name}

  Click Add then come back in and add the Policies you want

  ACL (tab) -> Add (+ icon)

  acl: #

  The # is a wildcard giving full access to all options

  # is a wildcard for entries and periods, so one # gives all access.

  If you wanted full access to one part of an API element, use the * wildcard, which limits the wildcard to within the periods as in element1.*.element3

  Click Update
- Create a new User which has access to the new policy

  Wazo UI -> {Tenant} -> Credentials -> Identities -> Add (+ icon)

  Username: {API_Admin_User_Name}

  Password: {API_Admin_User_Password}

  Firstname:

  Lastname:

  Email:

  Purpose: External API

  Tenant: {Tenant}

  Click Add

  Go back in and Edit the User

  Polices (tab)

  Policies: {All_Access_ACL_Name}
- Use this "user" for API calls and you will have full access to everything

  Get a token with (just like above, using this user's credentials

wazo-auth-cli token create --auth-username {API_Admin_User_Name} --auth-password {API_Admin_User_Password}

Do the curl requests as shown above

g) Launch Web UI to Configure WazoPBX

- Go to https://{wazo-server_ip_address}
  - ◊ make sure to use https, not http
  - ◊ You will get the warning about invalid certificate ; ignore this and advance
- Login with the root user and the password you set above: {wazo-root-password}
- The remainder of this section will use the web UI unless otherwise stated
- Configuration Notes
  - ◊ Any changes to the Wazo configuration must be enabled by reloading Wazo with

    systemctl restart wazo-agid

    Reboot the server

    The instructions imply that doing just "systemctl restart wazo-agid" will update the configuration, but to make the configuration take effect, the system must be restarted.

    This may be due to the translation in the current system from chan_sip settings to chan_pjsip settings
  - ◊ You will be caught in a catch-22 in some confiuguration choices where you need Users defined to fully define Schedules but you need Schedules defined to complete the Users setup. It will sometimes require you to go back in and augment a previously incomplete setup but be patient, it does eventually come together. The sequence shown in this document is intended to minimize the overlap.
- Credentials
  - ◊ Tenants
    - ♦ Add a tenant: {Tenant} (do not use any spaces in the name)
    - ♦ Delete the existing tenant "My-Company"
  - ◊ Select {Tenant} in the top-left drop-down menu
    - ♦ All edits will be made to this Tenant's configuration as long as it is selected
    - ♦ *** MAKE SURE ** that when you do the configuration for the tenant, you ALWAYS select the appropriate tenant, or you will be trying to figure out why your configuration is not working when in fact you have not done the configuration for the tenant.
- Musics
  - ◊ (Optional) Upload MusicOnHold sounds if you want custom sound files for MoH
    - ♦ Name: RC_MoH_Set1
    - ♦ Mode: Files
    - ♦ Application => Changes to Sort when you choose Files above: Random

      Browse: [available using edit icon after saving;can upload multiple times]

- ♦ See section below on locating and formatting MoH files

- Advanced
  - ◊ Extensions Features
    - ♦ (Optional - and likely NOT something you care about, but I do) Change star-codes for CallRecord and BlindTransfer

      I am used to *1 as Call Recording so wanted to switch back to that. Wazo installations use *1 for Blind Transfer (the Asterisk norm is #1) and *26 for activating Call Recording on a line/extension BEFORE initiating a call and *3 for recording a call while the call is in progress (called online call recording). I do not usually set any user/extension to record all calls so I am not worried about the *26 start code. To change online call recording star-code back to *1 :

      In Advanced -> Features General Settings -> Features Map (tab) change the star-code for blindxfer (Blind Transfer) to *55 from *1

      In In Advanced -> Features General Settings -> Features Map (tab) change the star-code for automixmon to *1 from *3

      Note: as of Wazo 20.05 the online_call_record_enabled setting defaults to false and there is no Web GUI setting to change that. As of the creation of this document, the only known way to change that is to use API commands for each user you want to allow to initiate online call recording. See the "patch" above if you want to enable online_call_record.
  - ◊ Sound Files System (Optional)
    - ♦ On a fresh install, only en_US and fr_FR sounds are installed. Canadian French and German are available too.
    - ♦ To install Canadian French sounds you have to execute the following command:

      apt install asterisk-sounds-wav-fr-ca wazo-sounds-fr-ca
  - ◊ SIP General Settings (for my setup)
    - ♦ By running, at the CLI

      wazo-confgen asterisk/sip.conf

      you can get a listing of all the settings used for SIP connections. This is useful as a final check once the settings have been set.
    - ♦ The following are additions to the settings which will apply to all SIP connections unless specifically changed within the Options list for that SIP connection

      The next two replace what was externip / externaddr for situations where the server is NATed behind a router with the public-facing IP address

      external_media_address: {public-facing-IP-address}

      external_signalling_address: {public-facing-IP-address}

      This lets asterisk know which IP addresses are locally connected

      local_net:

      As an example, local_net: 192.168.1.0/24

The default setup of Asterisk is to report, at high verbosity levels, legitimate activities like administrator sign-on and sign-off. This may be a nuisance to you and if so, you can add this parameter to your [General] settings. Sometimes you DO want the connection status to display, so remove the displayconnects=no if you are looking at the connection status as part of your debugging.

displayconnects: no

The next three establish that the server is NATed behind a router.

nat: auto_force_rport

delete; replaced with below. If leave this as is, when using ULAW codec, the dtmf signals could try to go inband

With chan_sip, the NAT setting had two settings in one entry, as shown bleow, but the chan_sip to chan_pjsip translator cannot handle tranlastions where two (or more) settings are included in one line so do NOT use

nat: force_rport,comedia

With chan_pjsip, the following three replace nat=yes or nat=force_rport,comedia

force_rport: yes

rewrite_contact: yes

rtp_symmetric: yes

direct_media: no

Not technically part of NAT settings but is an important part of making NAT work

This is already set in SIP_General settings but left here for reference

(Optional) If you want to have names, not extensions appear as CallerID for internal calls

sendrpid: pai (was no)

trustrpid: yes (was no)

(Optional) If you are having trouble with audio passing cleanly, you may wish to try different jitterbug settings with

jbenable = yes

jbforce = no

jbmaxsize = 200

jbresyncthreshold = 1000

jbimpl = fixed

jblog = no

Unless explicitly identified above, leave the general (default) settings for SIP as they are.

◊ Voicemail General Settings

♦ General (tab)

operator: no

saycid: yes

(change this only if you want the announcement to include the CallerID of the caller; I did)

sendvoicemail: yes

serveremail: {admin_email}

emailsubject (was in French -> Changed to English)

Change

Messagerie Wazo

to

You have received a new voicemail on WazoPBX

emailbody (was in French -> Changed to English)

Change

Bonjour ${VM_NAME},\n\nVous avez reçu un message d'une durée de ${VM_DUR} minute(s), il vous reste actuellement ${VM_MSGNUM} message(s) non lu(s) sur votre messagerie vocale : ${VM_MAILBOX}.\n\nLe dernier a été envoyé par ${VM_CALLERID}, le ${VM_DATE}. Si vous le souhaitez vous pouvez l'écouter ou le consulter en tapant le *98 sur votre téléphone.\n\nMerci.\n\n-- Messagerie Wazo --

to

Hello ${VM_NAME},\n\nYou have received a message lasting ${VM_DUR} minute(s). You currently have a total of ${VM_MSGNUM} unread message(s) in your voicemail box (number ${VM_MAILBOX}).\n\nThe voicemail just left was sent by ${VM_CALLERID}, at ${VM_DATE}. If you wish, you can listen to it by typing *98 on your phone.\n\nThank you.\n\nWazoPBX

- Timezones

  Nothing to edit here

◊ Confbridge General Settings

- With the standard setup for the Conference bridge, users can join the conference and initiate the conference by entering their User PIN - before the leader opens the conference by entering their Admin PIN.
- So unless I change the User PIN every time I host a conference (which I do not want to do), once someone has used my conference bridge as a User, they can use the conference bridge without me as an Admin allowing it - unless I disable the Conference brindge or change the User PIN, neither of which is something I want to do.
- So I made some changes (below) so that now, the Conference bridge will allow Users to join the Conference using the User PIN, but until the Admin User, using the Admin PIN, joins, the other Users cannot talk with each other.
- In addition, with this setup, when the Admin User leaves the Conference, the other Users are disconnected.
- I like this for its increased security

  Advanced -> Confbridge General Settings -> Default User

wait_marked : yes

(Users cannot join until leader joins)

end_marked : yes

Conference is terminated when leader leaves)

Create a new file

In /etc/asterisk/confbridge.d/ add a file (I called it 50-admin-is-marked.conf)

nano /etc/asterisk/confbridge.d/50-admin-is-marked.conf

[xivo-admin-profile-1](+)

marked = yes

Reboot the Server to activate the new configuration

◊ Contexts -> Add (click + icon)

♦ Outcall

General

Name: {OutcallName}

Label: to-extern

Type: Outcall

Description: Calls going out of the LAN

Contexts included: [leave blank]

Only becomes visible after save and return to edit settings

♦ Internal

General

Name: {InternalName}

Label: internal-origin

Type: Internal

Description: Calls originating and, unless captured by the to-extern Context, terminating within the LAN

Contexts included: to-extern

Only becomes visible after save and return to edit settings

User (have to save and then edit to see this option)

Start:{InternalRangeStart}

End: {InternalRangeEnd}

Group

Start:{GroupRangeStart}

End: {GroupRangeEnd}

Queue

Leave blank

Conference

Start: {the internal extension to use for the first (and maybe only) Conference bridge}

End: {the same extension as Start: above if you only want one ; Or (extension + n -1) if you want n Conference bridges available}

- Incall

  General

  Name: {IncallName}

  Label: from-extern

  Type: Incall

  Description: Calls coming in to the LAN

  Contexts included: [leave blank]

  Only becomes visible after save and return to edit settings

  Incall (have to save and then edit to see this option)

  Local calls

  Start: 2010000000

  End: 9999999999

  DID Length: 10

  Long Distance calls

  Start: 12010000000

  End: 19999999999

  DID Length: 11

- Rtp (make no changes here during initial setup and only customize (if want) once get rest working)
  - RTP Start: 10000

    Optional - can change this (I did not) to reduce open ports for security

    If you change these settings, make sure any connected device also changes their Rtp settings, especially the Rtp Start setting

    Start must be an even number ; End must be an odd number

    RTP Start: 10002 [default 10000]

    RTP End: 10203 [default 20000]
  - RTP End: See above
  - RTP Check Sums: no
  - DTMF Timeout
  - RTCP Interval
  - Strict RTP: no

- Probation
- Ice Support: no
- STUN Address
- STUN Blacklist
- TURN Address
- TURN Username
- TURN Password
- Ice Blacklist

- Voicemails
  - You can send to Voicemail at the User, Group or Schedule point. Since the redirect to voicemail could occur from any of the above ((Internal call to User, After-hours call, or normal external call directed by Group), put the redirect to voicemail in all three areas.
  - Voicemail(n)

    Initial Screen

    Name:

    Context: Internal

    Number: [recommend you match the extension number with which this is associated]

    Password:

    Email:

    Language: en_US

    Timezone: America/New_York

    Users: Leave blank for now; add in Users setup

    Save the above then select the item just saved and click the edit (pencil) icon

    Ask for password: Check

    Attach audio: Check

    Delete message after notification: UnCheck

    Activated: Check

    When you have a device/softphone connected, come back and dial the Voicemail extension and create your messages for callers:

    Dial, from any endpoint/extension *99 followed by the mailbox number

    Enter the password followed by #

    press 0 to access the mailbox options

    Press 3 to record your Name

    Press # to end the recording

    Press 2 to listen to what you recorded

    Press 1 to accept the recording and return to the options list

Press 3 to re-record the recording

Press 1 to record your Unavailable Greeting

Press # to end the recording

Press 2 to listen to what you recorded

Press 1 to accept the recording and return to the options list

Press 3 to re-record the recording

Press 2 to record your Busy Greeting

Press # to end the recording

Press 2 to listen to what you recorded

Press 1 to accept the recording and return to the options list

Press 3 to re-record the recording

(Optional; if you record this, callers will hear this until you delete it; this is intended to be used when you are out of the office temporarily and not picking up Voicemails)
Press 4 to record your Temporary Greeting

Press # to end the recording

Press 2 to listen to what you recorded

Press 1 to accept the recording and return to the options list

Press 3 to re-record the recording

- Conference
  - ◊ Conferences
    - ♦ Initial ScreenName: {Free text entry}

      Context: Internal

      Extension: {something in range defined above}

      PIN: {PIN for Users to enter to gain access to Conference}

      Admin PIN: {PIN for Conference owner}

      Announce join leave: Checked (do what you want here)

      Announce user count: Checked (do what you want here)

      This tells the Admin, when they log in, how many are already in the Conference

      Announce only user: Checked

      This tells the first user they are the only/first user in the conference
    - ♦ 2nd Screen - available after saving the initial screen

      Music On Hold: {Select what you want here}

      Subroutine:

      Quiet join/leave: UnChecked
  - ◊ The options available to the users and admin are (press the key to activate the feature):

- 1 Mute/Un-Mute Self Mute/Un-Mute Self
- 2 Lock/Unlock Conference Disabled for non-Admin
- 3 Eject last user who joined conference Disabled for non-Admin
- 4 Decrease Listen Volume of Conference Same for user
- 5 Resets the caller's listening volume to the default level. Same for user
- 6 Increase Listen Volume of Conference Same for user
- 7 Decrease Talk volume Same for user
- 8 Reset speaker's speaking volume to the default level Same for user
- 9 Increase Talk volume Same for user
- * Play menu options Same for user

- Provisioning
  - Plugins - Aastra
    - Once the Plugins are installed, the template files are populated, and the Users/Lines are configured with the Wazo GUI, you generate the Aastra .cfg files for each Aastra Device attached via a Plugin and save them as a file on the Wazo server by running

      wazo-provd-cli -c 'devices.using_plugin("xivo-aastra-3.3.1-SP4").reconfigure()'

      wazo-provd-cli -c 'devices.using_plugin("xivo-aastra-3.3.1-SP4").synchronize()'

      I am using the xivo-aastra-3.3.1-SP4 Plugin; Replace the Plugin with whatever you are using
    - You then load the config files onto each Device by pointing each Aastra to the Wazo server (using http server parameters) and then reboot each phone so each Aastra device can fetch its configuration file . The location the phones will look to for the configuration will be (in my case) /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/tftpboot/Aastra
    - Install the plugins to enable proper configuration of Aastra devices

      I use Aastra phones exclusively for my IP phones and softphones for my remote phones, so I only need to download the Aastra device plugins. Adapt to suit your system.

      In search field enter aastra

      Click the Download icon to the right of the v3.31-SP4 HF9 firmware for Aastra

      xivo-aastra-3.3.1-SP4

      Plugin for Aastra/Mitel 67XXi, 9143i and 9480i in version 3.3.1 SP4 HF9

      Do NOT use xivo-aastra-s.6.0.2019 ; Wazo needs some of the functionality in the v3 firmware

      This is a change from my previous preference for the v2.6 but it is needed with Wazo

      The plugin will expect the firmware to be v3.x for any Aastra connected and configured with these instructions
    - Add the correct firmware versions for the Aastra phones to the server directory so it will check and update the firmware as necessary for any Aastra device connected to the server

For Aastra phones, I use the 6753i and 6757i models so I only needed to make sure those two firmwares were loaded into the correct directory Those files are the v3.3.1_4358 firmware for each phone (make sure to name them as shown so the phones know what to look for on the Wazo server):

53i.st

57i.st

See the instructions below for transferring files from your PC to the Linux server, and put the above two firmware files, with the correct permissions wazo-provd (user and group) into

/var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/tftpboot/Aastra

- Wazo uses template files for the basic configuration of each Aastra device, model-specific template files to adapt the configurations according to each model's capabilities, and (optionally) MAC Address configuration files to adapt the configurations for settings unique to one device. It then creates, from these templates and the settings entered in the web GUI, a configuration file for each phone connected as a device.
- We could just accept the default plugins created by Wazo for the Aastra phones (v3.3.1), but I have some custom setups that I prefer so I will be customizing the core plugin configurations to meet my needs. See the Config Devices section for details.

I edit the aastra.cfg file to make it empty

Do NOT just delete it or the system will replace it. Replace its current contents with a comment line.

echo "#" > /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/tftpboot/Aastra/aastra.cfg

I edit the base.tpl file (see details below in the Config Devices section) to establish settings used for all Aastra devices connected to the Wazo server

I edit the 6753i.tpl and 6757i.tpl files (see details below in the Config Devices section) to:

include the base.tpl settings

establish settings used for all 6753i and 6757i Aastra devices connected to the Wazo server

Where needed, I create and edit {mac_address}.cfg.tpl files (see details below in the Config Devices section) to:

include the {model = 6753i or 6757i}.tpl settings (which includes base.tpl)

create any settings that are unique to one device

◊ Plugins - Cisco SPA2120 ATA

- The methodology for this is the same as the above for Aastra. For detailed explanations, see the Plugins - Aastra section. This section will just provide the bare instructions.
- When finished creating the custom template, to reconfigure the actual configuration files from the custom template, run, from the CLI

wazo-provd-cli -c 'devices.using_plugin("xivo-cisco-spa2102-5.2.12").reconfigure()'

wazo-provd-cli -c 'devices.using_plugin("xivo-cisco-spa2102-5.2.12").synchronize()'

- Using the Wazo GUI, establish Users, with Lines, for 1 or 2 analog Devices. In my case:

  User ata_cordless for the cordless phone

  User ata_fax for the FAX line

  Setup both with incall and outcall routes as with any other UserLine, including adding the cordless phone to a Group if you want.

- Using the Wazo GUI, at Provisioning -> Plugins -> 2120 (search bar) -> Download (icon) the xivo-cisco-spa2102-5.2.12 Plugin

  Make sure your ATA is at the same firmware version as the Plugin specifies

- Using the Wazo GUI, create a Device template for the Cisco SPA2102 ATA with Provisioning -> Config Device -> Add (+ icon) and entering (adjust to suit your preferences)

  General

    Label: Cisco SPA2102 ATA

    Language: en_US

    Timezone: America/NewYork

    Protocol: SIP

    Enabled NTP: Check

    NTP server: ca.pool.ntp.org

    Phonebook server:

    SIP DTMF mode: RTP-out-of-band

    User username: {ATAUserName}

    User password: {ATAUserPassword}

    Admin username: {ATAAdminName}

    Admin password: {ATAAdminPassword}

  Advanced:

    VLAN Enabled: Unchecked

    VLAN ID

    VLAN Priority

    VLAN PC Port ID

    Mac Address of ATA:

    Label: Cisco SPA2102 ATA Template

- Using the Wazo GUI, at Devices -> Add (+ icon),

  Mac Address of ATA:

  Description: Cisco SPA2102 ATA

    Save and go back to add

  Plugin: select xivo-cisco-spa2102-5.2.12

Template: select Cisco SPA2102 ATA Template

- ♦ Using the Wazo GUI, at Users -> (ATA line user) -> Lines

  Connect the Device to the Line

◊ Plugins - Config device

- ♦ Default

  Click on existing entry called "Default config device (defaultconfigdevice)"

  Click the Edit icon (the pencil) and check to make sure

  General (tab)

  Language : en_US

  Timezone : America/Toronto

  Protocol : SIP

  SIP DTMF mode : RTP-out-of-band

  Enabled NTP: Checked

  NTP server: Leave at defaults

  Phonebook server: Leave blank

  SIP DTMF mode: RTP-out-of-band

  User username:

  User password:

  Admin username:

  Admin password:

  Explicit notification of messages:

  Advanced (tab)

  [Nothing to change]

- ♦ Create a "Device" for the Aastra phones. You could create as many as you will have unique devices / unique configurations. If you have multiple devices using the same model number and with the same configuration other than what is entered in the Web GUI, you can use a common Template for them, and do NOT need a unique template for each.

  Add (+ icon)

  General (tab after saving; only screen on initial creation)

  Label

  Language

  Timezone

  Protocol

  Enabled NTP

  NTP server

SIP DTMF mode

User username

User password

Admin username

Admin password

Explicit notification of messages

Save and then click edit (pencil) icon and click on Advanced

Advanced (tab) [no changes being made here]

VLAN Enabled

VLAN ID

VLAN Priority

VLAN PC Port ID

If we also have device-specific configurations we want for one or more Aastra devices, we put a .tpl file into the …/var/templates directory with the devices mac address as the file name so the file name would be

{mac_address}.cfg.tpl

and it would be put into

/var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/templates/{mac_address_without_the_colons }.cfg.tpl

For these custom templates, if they are also using the Wazo Plugin like the others above (which in my case they are), we are creating their complete configuration when we run

wazo-provd-cli -c 'devices.using_plugin("xivo-aastra-3.3.1-SP4").reconfigure()'

However, if the custom configurations are not going to use the Plugin, then we would need to create the final, configuration file with

wazo-provd-cli -c 'devices.using_mac("{mac_address_without_the_colons}").reconfigure()'

so that it would generate

/var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/tftpboot/Aastra/{mac_address_without_the_colons }.cfg

The basic template file for Aastra confuguration is called base.tpl and the template used to create that is stored at

/var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/templates/base.tpl

we will be creating a customized version of that in

/var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/templates/base.tpl

so copy over a starting file and we will customize it below

cp -a /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/templates/base.tpl /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/templates/base.tpl

Wazo will take its base configuration from this file as it creates the final configuration for each Aastra device

The model-specific template file for Aastra confuguration is called {model}.tpl and the default template for these are stored at (I use the 6753 and 6757 Aastra models)

/var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/templates/6753i.tpl

/var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/templates/6757i.tpl

So, as before, copy over starting files which we will customize below

cp -a /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/templates/6753i.tpl /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/templates/6753i.tpl

cp -a /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/templates/6757i.tpl /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/templates/6757i.tpl

Wazo will add to its base configuration for each model from these files as it creates the configuration for each Aastra device

Wazo looks in …/var/templates and if it does not find the file there then it looks in …/templates, so by putting our custom .tpl files into …/var/templates, we ensure the system uses our customized versions.

You can now edit the files created above to create custom Aastra template files and from the custom templates configure devices (below)

nano /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/templates/base.tpl

Make whatever customizations you want (if any)

nano /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/templates/6753i.tpl

Make whatever customizations you want (if any)

nano /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/templates/6757i.tpl

Make whatever customizations you want (if any)

nano /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/templates/{mac_address}.tpl

Make whatever customizations you want (if any)

- Create a "Device" for the Cisco SPA2102 device

  See above Aastra section for extended instructions. This follows similar techniques so will be quick activity list without the extended explanations.

  Note you are NOT replacing the entire contents of the /var/lib/wazo-provd/plugins/xivo-cisco-spa2102-5.2.12/var/templates/base.tpl below, just a section of it so be careful about your replacement selection.

  Create the custom template for the Wazo system

  cp -a /var/lib/wazo-provd/plugins/xivo-cisco-spa2102-5.2.12/templates/base.tpl /var/lib/wazo-provd/plugins/xivo-cisco-spa2102-5.2.12/var/templates/base.tpl

  nano /var/lib/wazo-provd/plugins/xivo-cisco-spa2102-5.2.12//var/templates/base.tpl

  replace everything from

  {{ XX_timezone }}

to

{% block suffix %}{% endblock %}

with (replacing the [] and the items in [] brackets with your information)

{{ XX_timezone }}

<Connection_Type>Static IP</Connection_Type>

<Static_IP>[IP_Address_of_Wazo_Server]</Static_IP>

<NetMask>[Your_NetMask_usually255.255.255.0]</NetMask>

<Gateway>[Your gateway_usually_192.168.1.1]</Gateway>

<HostName>[Your_Hostname]</HostName>

<Domain>[Your_Domain]</Domain>

<Primary_NTP_Server>ca.pool.ntp.org</Primary_NTP_Server>

<Enable_WAN_Web_Server>yes</Enable_WAN_Web_Server>

<Maximum_Uplink_Speed>512</Maximum_Uplink_Speed>

<Networking_Service>Bridge</Networking_Service>

<Enable_DHCP_Server>no</Enable_DHCP_Server>

<RTP-Start-Loopback_Codec>G711u</RTP-Start-Loopback_Codec>

<Resync_Random_Delay>0</Resync_Random_Delay>

<Resync_Periodic>0</Resync_Periodic>

<Upgrade_Enable>No</Upgrade_Enable>

{% for line_no, line in sip_lines.iteritems() %}

<Line_Enable_{{ line_no }}_>Yes</Line_Enable_{{ line_no }}_>

<SIP_Proxy-Require_{{ line_no }}_>{{ XX_proxies[line_no] }}</SIP_Proxy-Require_{{ line_no }}_>

<Network_Jitter_Level_{{ line_no }}_>high</Network_Jitter_Level_{{ line_no }}_>

<Jitter_Buffer_Adjustment_{{ line_no }}_>up and down</Jitter_Buffer_Adjustment_{{ line_no }}_>

<Proxy_{{ line_no }}_>{{ XX_proxies[line_no] }}</Proxy_{{ line_no }}_>

<Use_DNS_SRV_{{ line_no }}_>Yes</Use_DNS_SRV_{{ line_no }}_>

<Proxy_Fallback_Intvl_{{ line_no }}_>120</Proxy_Fallback_Intvl_{{ line_no }}_>

<Display_Name_{{ line_no }}_>{{ line['display_name']|e }}</Display_Name_{{ line_no }}_>

<User_ID_{{ line_no }}_>{{ line['username']|e }}</User_ID_{{ line_no }}_>

<Password_{{ line_no }}_>{{ line['password']|e }}</Password_{{ line_no }}_>

<Auth_ID_{{ line_no }}_>{{ line['auth_username']|e }}</Auth_ID_{{ line_no }}_>

<Preferred_Codec_{{ line_no }}_>G711u</Preferred_Codec_{{ line_no }}_>

<Use_Pref_Codec_Only_{{ line_no }}_>yes</Use_Pref_Codec_Only_{{ line_no }}_>

<Silence_Supp_Enable_{{ line_no }}_>no</Silence_Supp_Enable_{{ line_no }}_>

<Echo_Canc_Enable_{{ line_no }}_>no</Echo_Canc_Enable_{{ line_no }}_>

<Echo_Canc_Adapt_Enable_{{ line_no }}_>no</Echo_Canc_Adapt_Enable_{{ line_no }}_>

<Echo_Supp_Enable_{{ line_no }}_>no</Echo_Supp_Enable_{{ line_no }}_>

<FAX_CED_Detect_Enable_{{ line_no }}_>yes</FAX_CED_Detect_Enable_{{ line_no }}_>

<FAX_CED_Detect_Enable_{{ line_no }}_>yes</FAX_CED_Detect_Enable_{{ line_no }}_>

<FAX_Passthru_Codec_{{ line_no }}_>G711u</FAX_Passthru_Codec_{{ line_no }}_>

<FAX_Passthru_Method_{{ line_no }}_>ReINVITE</FAX_Passthru_Method_{{ line_no }}_>

<DTMF_Tx_Method_{{ line_no }}_>AVT</DTMF_Tx_Method_{{ line_no }}_>

<DTMF_Tx_Mode_{{ line_no }}_>Strict</DTMF_Tx_Mode_{{ line_no }}_>

<FAX_Process_NSE_{{ line_no }}_>yes</FAX_Process_NSE_{{ line_no }}_>

<FAX_Disable_ECAN_{{ line_no }}_>no</FAX_Disable_ECAN_{{ line_no }}_>

<FAX_Enable_T38_{{ line_no }}_>no</FAX_Enable_T38_{{ line_no }}_>

<FAX_Tone_Detect_Mode_{{ line_no }}_>caller or callee</FAX_Tone_Detect_Mode_{{ line_no }}_>

<Dial_Plan_{{ line_no }}_>(*xx?|[3469]11S0|0|00|1xxx[2-9]xxxxxxS0|[2-9]xx[2-9]xxxxxxS0|[2-9]xxxxxx|xxxxxxxxxxxx.)</Dial_Plan_{{ line_no }}_>

{% endfor -%}

{% block suffix %}{% endblock %}

```
wazo-provd-cli -c 'devices.using_plugin("xivo-cisco-spa2102-5.2.12").reconfigure()'
wazo-provd-cli -c 'devices.using_plugin("xivo-cisco-spa2102-5.2.12").synchronize()'
```

Initialize the CISCO SPA2102 ATA for use on the LAN (with just enough information that it will look to the Wazo server for its full configuration)

Reset to Factory Default Settings

Unplug ATA and disconnect Ethernet conections

Hook POTS Analog Phone up to Line 1

Plug in ATA

UnHook Phone

Press: ****

Press 73738 #

Press: 1

The system will hang up on you as it resets

Set the initial conguration parameters

Connect the Ethernet Port (NOT the WAN port) on the ATA to an Ethernet port on a PC that is NOT connected to a router (wired or Wireless)

Launch the browser on the PC to IP address 192.168.0.1 (the default address after factory reset)

With the ATA GUI presented click on "Admin login" then "advanced" then "Router" (main tab) then "WAN Setup" (sub tab)

Connection Type: Static

Static IP: {ATA_WAN_IP_Address}

This is called the WAN IP Address but it is actually the IP address you will enter from inside the LAN to access the ATA GUI once the ATA is connected to the LAN

NetMask: 255.255.255.0 (normal configuration; adapt if required)

Gateway: {ATA_Gateway_IP_Address}

On home systems, this is often 192.168.1.1

Leave the rest on this tab in their factory setting

With the ATA GUI presented click on "Admin login" then "advanced" then "Router" (main tab) then "LAN Setup" (sub tab)

Networking Serve: Bridge

Enable DHCP Server: No

Leave the rest on this tab in their factory setting

Click "Submit All Changes" and wait for the GUI to return

With the ATA GUI presented click on "Admin login" then "advanced" then "Voice" (main tab) then "Provisioning" (sub tab) and edit the "Profile Rule" to now read (replace {} and contents of {} with your data)

http://{IP_Address_of_Wazo_Server}:8667/spa$PSN.cfg

Unplug the ATA, connect the Ethernet (NOT the WAN) port to the LAN and plug the

ATA back in, waiting for it to boot and fetch the configuration settings from the Wazo server

From each analog device (Phone, FAX, …) connected to the ATA, call another extension on the Wazo system to initialize the configuration and test the configuration

You can now connect directly to the ATA at the {ATA_WAN_IP_Address} using the ATAUserName or ATAAdminNAme credentials created above.

◊ Configuration

♦ General (tab)

Nat: Checked

♦ Advanced (tab) viewable after saving and re-entering via edit icon)

Provisioning Host: {server_IP_address}

Provisioning Port: [record the number shown (8667) to use in setting the Configuration Server Port in the devices]

Locale: Set to en_US

- Devices

◊ For each device, you need to add a device, enter the Initial (basic) information, Update, then go back in with the Edit (pencil) icon and finish the configuration

♦ Initial screen:

IP: [leave blank]

MAC:

Switchboard: Unchecked

Description:

♦ Visible after clicking Update and clicking Edit (pencil) icon

Template: Aastra

IP: (leave blank)

MAC: (greyed out - provided in Initial entry)

Plugin: xivo-aastra-3.3.1-SP4

Model: (greyed out - will be populated when synchronized)

Vendor: (greyed out - will be populated when synchronized)

Version: (greyed out - will be populated when synchronized)

Switchboard: (leave unchecked)

Description: (leave as defined in initial entry)

- Schedules

◊ You can activate a Schedule at the DID point, the User point or the Group point. I chose to do it at the User point if the User was NOT associated with Group and at the Group point if the user was part of a group.

◊ You want to specify the Open and Closed options in a Scheule so you can tell Wazo what to

do when a call comes in and the Closed period is activated. I like to declare the Open period during the day and then have two closed periods: One from midnight to before it opens and one after it closes to midnight that night (23:59.

◊ Schedule1

♦ General (first window and then tab when edit)

Name: To Voicemail at Night

Timezone: {your local Timezone}

Destination: Voicemail

Voicemail: leave blank

Greeting: Unavailable

Skip instructions: Unchecked

♦ Exceptional periods

Exception 1

Hour Start: 00:00

Hour End: 07:00

Weekdays: Select All

Monthdays: Select All

Months: Select All

Destination: Voicemail

Voicemail: {your choice}

Greeting: Unavailable

Skip Instructions: Checked

Exception 2

Hour Start: 21:00

Hour End: 23:59:00

Weekdays: Select All

Monthdays: Select All

Months: Select All

Destination: Voicemail

Voicemail: {your choice}

Greeting: Unavailable

Skip Instructions: Checked

♦ Open periods

Hour Start: 07:00

Hour End: 21:00

Weekdays: Select All

Monthdays: Select All

Months: Select All

- Groups
  - ◊ Group 1
    - ♦ First Screen

      Name: {Free_text_name_with_no_spaces}

      Context: Internal

      Members: Will be populated from Users screens
    - ♦ General (tab)

      Name: {entered above in first screen}

      Context: {entered above in first screen}

      Callerid mode: none

      Enabled: Check

      Music On Hold: MoH_Set1 (If you created it)

      Subroutine

      Retry delay: 5

      Ring in use: Check

      Ring strategy: All

      Timeout:

      User timeout: 15
    - ♦ Members (tab)

      Members: Will be populated from Users screens
    - ♦ Fallbacks

      Destination: Voicemail

      Voicemail: {voicemail name}

      Greeting: Unavailable

      Skip instructions: Checked
    - ♦ Schedule

      Schedule: Enter Schedule to use if have one for this group
    - ♦ Call Permissions

      Call Permissions: leave blank
- Users
  - ◊ When you create a user, one of the things you do is add a line to the User. Wazo will automatically create a LineName/LinePassword for that line. I found the LinePassword to be too short and the LineName not descriptive (randomly generated) so I went into the Line settings after creating the User and, in the Line settings edited the LineName/LinePassword,

which is then reflected in the User settings. If your system uses a common CallerID for multiple users, I suggest you do NOT do all the users and then try to edit the Lines password since the Lines are only identified by LineName and CallerID and it is not otherwise clear when you go to the Lines menu which Line is with which User, so doing them one at a time makes the job easier. It does require switching back and forth between Users and Lines, but I think you will find that easy to do.

◊ The "Ring seconds" parameter set in the General tab must be a multiple of 5

◊ User1

♦ General

Subscription Type: Voice

Firstname:

Lastname:

Email:

Password:

Context: Internal

Protocol: SIP

♦ Click Add and then select the User and click the edit (pencil) icon to see the rest

User (tab)

Subscription Type

Firstname

Lastname

Caller ID: [enter using format: "free text" <nnnnnnnnnn>

Outgoing caller ID

Default: (to let the line->trunk set the CallerID)

CallerID is pulled from the first one NOT blank:

Trunk CallerID

Outgoing Call Routing CallerID

If the Internal: Checkbox is checked, the internal CallerID will be used

User CallerID

Line CallerID

Customize: "George" <2015551212> (to have the extension set Caller ID - IF the trunk & line CallerIDs are blank)

Email

Authentication

Enable authentication: Check (Only Check and enter the Password if you want to allow this user to use API calls)

Username

Password

General (tab)

Phone mobile

If you want an outside number to also be called when this extension is called, add the outside number in this field and add "pre-mobility" (without the quotes) in the Subroutine field below.

If you want to have an outside number accessible but not tied to any one User, create a fake User but do not actually connect the fake user to any device or softphone. Add the Phone mobility and Subroutine field to the fake user. When that "User" is accessed, either by someone calling the extension directly or by having the extension added to a group, the external number is called.

Note: If you changed the Name of the internal Context from default, see above for an edit to re-enable this feature.

Ring seconds

Music On Hold

Subroutine

You can add any subroutine you create or one that has been added to the subroutines in /etc/asterisk/extensions_extra.d

See above in the "Phone mobile" field for one such use of this field.

Simultaneous calls

Timezone

User Field

Description

Fallbacks (tab) (only if you have something unique for this user that is NOT handled via a Group or a Schedule)

Busy

Destination

Voicemail

Greeting

Skip instructions

Congestion

Destination

Filename

Skip

No Answer

Fail

Destination

Filename

Skip

No Answer

No answer

Destination

Voicemail

Greeting

Skip instructions

Services (tab)

Do not disturb

Incall filtering

Busy

Destination

Unconditional

Destination

No answer

Destination

Lines (tab)

Protocol: SIP

Name: {Populated by Wazo or from edit in Lines}

Context: Internal

Extension:

Device: {Select device MAC Address if assigned to a Device}

Line (N°): {Select Line on Device if assigned to a Device}

Application

Groups (tab)

{Select the Group to which this user belongs if there is a Group to which this User is assigned}

FuncKeys (tab)

Created in the templates for the Devices so left blank here

Schedule (tab)

{Select Schedule if desired}

Voicemail (tab)

{Select Voicemail if there is one assigned to this User}

Call Permissions (tab)

Left blank

- ◊ Softphones
  - ♦ For Users with no devices (Softphones) , you have no Device to define. You just define a User with a Line, then go to the Line and read/edit the Username/Password, and use the {server IP address}, {LineUser}, {Line Password} as your connection settings on the softphone.

    The one thing to be aware of: For my softphones, the LineName for the User for the softphone connection must be all numeric, NOT alphanumeric.

- Lines
  - ◊ For each line that has been added by creating them in the Users generation above, go in and edit the line settings to give each line a meaningful name, stronger password, correct Max#Calls and restriction to LAN access (permit)
  - ◊ Line (tab)
    - ♦ Username
    - ♦ Secret
    - ♦ Host: Dynamic
    - ♦ Type: friend
    - ♦ Context: internal
  - ◊ Options (tab) (available after you add a line and go back in to edit it)
    - ♦ callerid: {Set in Users}
    - ♦ call-limit: {whatever your provider gives you}
    - ♦ amaflags: {leave as is}
    - ♦ subscribemwi: no [normally no, but if you want mwi on a line, change to yes]
    - ♦ permit: {192.168.1.0/24 or whatever your LAN is}

      This is NOT required, but it adds a little extra security by ensuring that the line/extension can only be accessed from a LAN address - ie another phone inside the LAN or the PBX which is on the LAN

- Linking the devices to the server
  - ◊ Generate and save the device configurations for each device, which, when the "reconfigure" command (see below) is run, will combine the base.tpl template, the {model}.tpl template, the {mac_address} template (if you created one) and the settings input from the web GUI.
    - ♦ To reconfigure the settings or the devices, from the CLI

      wazo-provd-cli -c 'devices.using_plugin("xivo-aastra-3.3.1-SP4").reconfigure()'

      or, for devices with custom configuration for that device

      wazo-provd-cli -c 'devices.using_mac("{mac_address_without_the_colons}").reconfigure()'

      To generate the new configurations and save them on the server for devices that have a {MAC_Address} setup but are NOT using the Aastra plugin

      I did not use this; it is here for reference only

wazo-provd-cli -c 'devices.using_plugin("xivo-aastra-3.3.1-SP4").synchronize()'

To reboot the device(s) so they will now pull in their new configurations

This does not work since it depends on autoprovisioning, which we are not using. So instead, Reboot each phone individually. It is best to not reboot all phones on mass if you have some kind of brute force protection (eg fail2ban) on your active firewall, since the BFD protection may interpret a series of accesses from different IP addresses as a hacking attempt.

◊ After reconfiguring the settings for the devices you will find, in the /var/lib/wazo-provd/plugins/xivo-aastra-3.3.1-SP4/var/tftpboot/Aastra directory a series of files

{mac_address}.cfg files for each device

aastra.cfg file (which we set to empty previously

- ♦ If you created a {mac_address}.cfg.tpl file for a particular device, that {mac_address}.cfg file will contain a complete definition for that Device contining precisely what you specified.
- ♦ For all other devices, the {mac_address}.cfg file will contain only the settings unique to that device, as established in the Wazo GUI and NOT (yet) the information relevant to the Model number of the device.
- ♦ At this point, before you get the devices to fetch their configurations from the Wazo server, if you look in the Devices page, you will see that the Devices do not have a Model Number associated with them. This is because the Devices have not yet communicated with the Wazo server so the Wazo server does not yet know which Model you have, from within the Aastra lineup.
- ♦ When you point the Aastra device to the Wazo server (see below) and restart the Aastra device, it will then communicate with the Wazo server, tell the Wazo server what model number it is and the Wazo Plugin will then combine the base.tpl, the Model.tpl and the {mac_address}.cfg files and generate a complete configuration for the Aastra device, loading that into the Aastra device.
- ♦ When that is complete, if you again look at the Devices page, you will see that the information for the Aastra device now shows the Model number and IP address of the Aastra device.

◊ Reset each Device to Factory Settings

- ♦ Depending on the vendor, you may want to disconnect the device from the network so it will get a complete wipe of all configuration settings. Some vendors reach out to their servers to get the most current factory settings and I do not want that.
- ♦ For Aastra - using the Phone's UI

Tools -> Admin Menu -> Enter Admin Password (default 22222) -> Factory Defaults

Confirm the request and the request to Restart the phone

◊ Point each device to the Wazo server IP so it will fetch the configuration settigns from the server when it is rebooted. You can do this either using the Web interface or the built-in-UI of the phone.

- ♦ Reconnect the phone to the network
- ♦ For Aastra

Using the web interface

First, use the phone UI to get the IP address of the phone

Tools -> Phone Status -> IP&MAC Addresses

Point your browser to the phone IP address

Login credentials (now that you have reset to Factory Defaults)

User Name: admin

Password: 22222

Advanced Settings -> Configuration Server

Download Protocol: HTTP

HTTP Server: {server IP address}

HTTP Path: Aastra

HTTP Port: 8667

Using the phone UI (NOT recommended since it requires considerable dexterity with the phone keypad)

Tools -> Admin Menu -> Enter Admin Password -> Cfg. Svr.

Download Protocol: HTTP

HTTP Settings

IP Address: {Server IP address}

backspace through existing setting and enter IP address, using leading #s to fill spaces for #s less than 3 digits

Press keys repeatedly to get to # instead of letter

use * for period

Path: Aastra

Port: 8667

◊ Reboot each phone

♦ For Aastra - from the Phone UI

Tools -> Restart Phone -> Confirm

- Trunks

◊ When you add a Trunk, you will see three tabs: Trunk, Register, Options

In the Trunk tab, you enter the information that lets your system call out

In the Register tab, you enter the information that lets your system accept incoming calls

Complete any required Options in the Options tab BEFORE checking and filling in the Register tab, so that when Wazo tries to Register, it has the correct configuration for the trunk provider.

If you designate your Type=Friend, then you do NOT need to have the Register tab enabled to make outgoing calls.

If you designate your Type=Peer, then you DO need to have the Register tab enabled to make outgoing calls.

It is marginally faster to have permanent Registration enabled for Outbound calls but it does take more activity to monitor the registration, so you choose your Type accordingly - and according to what the trunk provider uses.

In the Options tab, you enter any specific configuration options that your Trunk Provider says they need for connection to their system. These Options are in addition to the SIP (assuming you are using SIP connection) Options set in Advnaced -> SIP General Settings.

◊ Once configured, you can use, at the cli, the following commands to confirm the trunk settings

asterisk -rx "pjsip list endpoints"

[to see a summarized list of endpoints]

asterisk -rx "pjsip show endpoints"

[to see all of them in detail]

asterisk -rx "pjsip show endpoint {trunk_name}"

[to see the detail of one endpoint]

◊ Trunk(n)

- Protocol: SIP
- Trunk - This is what is used when Wazo uses the OutCall Context; the connection is only created when a call is initiated and the connection is terminated when the call is terminated

  Name: {free-text Label}

  Username: {given to you by your trunk provider}

  Password: {given to you by your trunk provider}

  Host: Static

  {trunk_provider_IP_address_or_FQDN}

  Type: Peer

  (Some trunk providers use Friend ; use the settings provided by your ttrunk provider)

  Context: Incall

  (This uses the label you gave it when setting up the Context; some use from-extern)

- Register - This is what is used when Wazo uses the InCall Context ; When Enabled (waiting for Incalls), the connection is maintained permanently via the ongoing registration

  Enabled: Checked

  If you have a provider, like Vitelity, that uses two trunks, one for InCall and one for OutCall, Do NOT check this for the Outcall trunk, just for the InCall trunk)

  SIP Username:

  Authentication Username: {same as Username in Trunk setting above}

Authentication Password: {same as Username in Trunk setting above}

Remote Host: {trunk_provider_IP_address_or_FQDN}

Remote port: {usually 5060}

Transport: UDP

Callback Extension: {enter DID}

Expiration

- ♦ Options

call-limit: 5

amaflags: default

subscribemwi: no

Plus These two are pretty much needed for any trunk provider

fromdomain: {provided by your trunk provider}

This was thought to be needed but after all the other configurations were properly determined, this could be eliminated and everything still worked

fromuser: {the Username given by your trunk provider - same as entered above}

If you use this, it will override the User (internal) CallerID so if you do not need this and want to pass through the User Caller ID, do NOT use this

- Incalls
  - ◊ Number(n)
    - ♦ Incall

Context: InCall (or whatever you labelled your InCall Context)

Number: {DID number to capture}

Destination: {User} or {Group}

(The above could be User, Group, Voicemail, … and the below is entered accordingly)

User: {User to which calls are directed}

Group: {Group to which calls are directed}

Ring time: 20

Preprocess Subroutine

Caller ID mode

- ♦ Schedule

Choose Schedules here if want to apply in a DID by DID basis, instead of on a User by User basis

- Outcalls
  - ◊ Route

Outcall

Name : {Free-text}

Description: {Free text}

Trunks : {click to choose}

Select, in sequence, the trunks to try so the route has multiple options in case one fails

Preprocess Subroutine:

Internal Caller ID: Check (I want the CallerID used by the User passed through)

Ring Time:

Extensions [can have multiple on each Outcall; just click + icon]

Context: to-extern

Extension: _201XXXXXXX [Area Code 201]

[could have been _201X. but that would have allowed strange entries]

[this is where the pattern goes; must be preceded by _ or is taken literally]

Caller ID: [only enter if want to override Line/User CallerID]

External prefix: [only if you want something prepended when pattern matches and a re dialing externally]

Prefix: [only if you want something prepended when pattern matches]

Strip digits: [only if you want something stripped at the start of the # when pattern matches]

Schedule

Schedule:

Call Permissions

Call Permissions

- Parking
  - ◊ Advanced -> Extensions Features
    - ♦ pickup:

      Enabled: Check

      Extension: (leave as _*8.)
  - ◊ Parking Lots
    - ♦ Name: {Free text}
    - ♦ Extension: {a number, within the range of Internal Conext, that you will use to send Parked calls)
    - ♦ Slots Start: {extension number + 1}
    - ♦ Slots End: {Slots Start + n -1 }

      n is the number of Parking slots you want to enable
    - ♦ Music On Hold:

        - Timeout: 60

          enter what you want; I did not want to make them wait more than 60 seconds

    - Make sure the settings for the softkeys have:
        - To Park the call

          lablel: "Park"

          type: speeddial

          value: "*5599000#"
        - To Pickup the call

          label: "Pickup"

          type: blf

          value: "99001#"

- Paging
    - We are NOT actually setting up Paging; we are setting up Intercom. Pagin is one-way audio only. Intercom is two-way audio. Within Wazo, both are configured via the Paging menu.
    - Paging Set 1
        - Name: {Free-text name for this page set}
        - Number: {an Extension within the Internal Context Range}
        - Members: {Select from the Dropdown menu the Users who will receive the Page)
        - Callers: {Select the Users who are allowed to Initiate a Page}

            And then go back in and Edit the Paging set
        - Announce caller: Checked
        - Announce sound:
        - Play notification to caller: Checked
        - Duplex audio: Checked
        - Enabled: Checked
        - Ignore forward: Checked
        - Announce caller:
    - Setup a softkey on your device to use the paging feature
        - Label: "Page"
        - Type: speeddial
        - Value: "*1198500#"
- To review a summary of the settings for all extensions, go Advanced -> Extension
- Understanding chan_sip vs pjsip
    - Wazo, with Asterisk 17, uses pjsip instead fo chan_sip. For the most part, Wazo has buffered the user from the differences by providing a translator from chan_sip to pjsip, but this is

restricting some functionality only available from pjsip so Wazo will be removing the translation capability in some future release and require the use of pjsip configuration syntax.

◊ Some reference sites for the differences include

https://wiki.asterisk.org/wiki/display/AST/Migrating+from+chan_sip+to+res_pjsip

https://wiki.asterisk.org/wiki/display/AST/PJSIP+Configuration+Wizard

https://wiki.asterisk.org/wiki/display/AST/res_pjsip+Configuration+Examples

https://wiki.asterisk.org/wiki/display/AST/Asterisk+13+Configuration_res_pjsip_endpoint_identifier_ip

- Some asterisk commands to help you troubleshoot the installation - done from the CLI

asterisk -r

[puts you into asterisk monitor mode where calls and other activities / errors are sent to your screen]

enter quit to leave this mode

♦ The following commands are run from the CLI and upon completion, return you to the linux command line.

asterisk -rx "dialplan show {number being dialed (internal or external)}@{context from which call comes (usually internal)}"

asterisk -rx "pjsip list endpoints"

asterisk -rx "pjsip show endpoints"

asterisk -rx "pjsip show endpoint {name of endpoint}"

asterisk -rx "pjsip list contacts"

asterisk -rx "pjsip show contacts"

asterisk -rx "pjsip show contact {name of contact}"

asterisk -rx "pjsip list registrations"

asterisk -rx "pjsip show registrations"

asterisk -rx "pjsip show registration {name of registered trunk}"

asterisk -rx "pjsip list subscriptions {inbound|outbound}"

asterisk -rx "pjsip show subscriptions {inbound|outbound}"

asterisk -rx "pjsip show subsription {inbound|outbound }{name of subscription}"

h) (Optional) Install FOP2 for Monitoring Trunk and Line Status

- FOP2 provides a graphical representation of the status aof lines and trunks. It is NOT required for Wazo, since you can always SSH in to the Wazo server and use Asterisk commands to get whatever status you want (Registration status, In Use status, …). I like FOP2 since it can be accessed via your browser and provides an overview on one page of all your connections.
- You could probably use Sylvain's discovery of the "Node Red" developer toolkit (www.nodered.org) to create your own status dashboard, and maybe, when I have time to learn Node Red and work with it, I will do just that, but in the meantime, FOP2 provides the view I want of the status of Lines and Trunks, and for $40 (the paid version - Basic edition), I will use this for now.

- You can download and use FOP2 for free as long as you use no more than 15 "buttons" so give it a try and if you don't like it, don't buy the paid version which removes the limitation on number of "buttons".
- Go to www.fop2.com to download the Debian version (I used the 64 bit version) or go directly to http://www.fop2.com/download/debian64
- For documentation by the developer
  - ◊ www.fop2.com/docs
  - ◊ https://www.fop2.com/docs/installation.php
  - ◊ https://www.fop2.com/docs/installation.php#ConfigurationServer
  - ◊ https://www.fop2.com/docs/userguide.php
- If you already have FOP2 installed and do not know if you need an update (eg to handle non-FLASH browsers which is fine with FOP2 past 2.28) you can get the existing version by entering, at the CLI

  /usr/local/fop2/fop2_server -v
- You probably could install FOP2 directly onto the Wazo server, but there are enough differnces between what Wazo has and what FOP2 needs that I install FOP2 on a separate server which has what FOP2 needs and point FOP2 to the Wazo instance.
  - ◊ FOP2 uses MySQL as its database and PHP scripts for its operation, neither of which are installed with Wazo, which used PostgreSQL as its database and Python scripts for its operation.
  - ◊ Since this document provides instructions for Proxmox hosting 3 servers: Wazo, a CRM and a Linux Desktop, you already have a virtual machine (the CRM) that has what FOP2 needs, so putting FOP2 on the CRM server / VM is no problem. And since Wazo and the CRM "servers" are actually on one piece of hardware, there is limited concern re lag in signalling between the two "servers".
- You will need to set your firewall on the Wazo system to allow FOP2 access from the CRM VM but you can enable port access specific to the CRM server IP so this is still very secure. This maintains the security and robustness of the WAZO server and provides the FOP2 functionality.
  - ◊ Ports requiring access from a remote server are
    - ♦ TCP / 5038 from the IP address of the CRM "server"
- You will, after following the instructions below, point your browser to the CRM VM to access FOP2 and FOP2 will then access Wazo.
  - ◊ To access FOP2 on the CRM server, point your browser to the CRM server IP address at Port TCP / 4445
  - ◊ http://{CRM_server_IP_address}:4445
- FOP2 has much more functionality than is used in this document. If you want to make more use of FOP2 as a receptionist panel or other functionality, see the documentation shown above.
- The rest of these instructions assume you are installing FOP2 on the CRM VM and configuring Asterisk on the Wazo VM to enable FOP2 to work from the CRM VM.
- On the WAZO VM

◊ Identify an Extention (User/Line in Wazo vocabulary) for FOP to register with when accessing the AMI. If you use a real Line, you have more functionality but if you want restricted (ReadOnly) functionality for a user with FOP2, create a fake User/Line with no device or softphone connected and use that in FOP2 to register with the AMI. I use the real User/Line.

- ♦ {FOP2_Line_Number}
- ♦ {FOP2_Line_UserName}
- ♦ {FOP2_Line_UserPassword}
- ♦ {FOP2_Line Options}

  Permit: {CRM_server_IP_address} or {LAN/24}

◊ Allow the CRM VM to access to Asterisk Manager

- ♦ Edit /etc/asterisk/manager.d/99-general.conf and change the bindaddress to allow the CRM VM access to the Asterisk Manager

  bindaddr = 0.0.0.0

  The bindaddress was set to 127.0.0.1 which prevented anyone other that a user on the localhost to access the Asterisk Manager Interface (AMI) and this is good security.

  By changing bindaddr to 0.0.0.0 we allow ALL IP addresses to access the AMI which is a dangerous thing if your router AND firewall do not adequately protect your Wazo server, so make sure to lock down port 5038 to only allow localhost or the CRM VM to use this port.

  In our case, since the two "servers" (Wazo and the CRM) are on the same physical hardware = on the same LAN, we do not need to enable any port forwarding on the router to allow external, public access to port 5038, so that helps with security, but do put in place strict firewall rules to lock the port down nonetheless.

- ♦ Create a new file /etc/asterisk/manager.d/50-fop2.conf to create a new fop2 user with designated permissions to access the Asterisk Manager

  cd /etc/asterisk/manager.d

  touch 50-fop2.conf

  chown asterisk:www-data 50-fop2.conf

  chmod 660 50-fop2.conf

  nano 50-fop2.conf (include the square brackets [], but NOT the {} brackets

  [{FOP2_AMI_UserName}]

  secret = {FOP2_AMI_UserPassword}

  deny = 0.0.0.0/0.0.0.0

  permit = {IP_address_of_CRM_VM}/255.255.255.255

  read = all

  write = all

  writetimeout = 1000

  eventfilter=!Event: RTCP*

eventfilter=!Event: VarSet

eventfilter=!Event: Cdr

eventfilter=!Event: DTMF

eventfilter=!Event: AGIExec

eventfilter=!Event: ExtensionStatus

eventfilter=!Event: ChannelUpdate

eventfilter=!Event: ChallengeSent

eventfilter=!Event: SuccessfulAuth

eventfilter=!Event: DeviceStateChange

eventfilter=!Event: RequestBadFormat

eventfilter=!Event: MusicOnHoldStart

eventfilter=!Event: MusicOnHoldStop

eventfilter=!Event: NewAccountCode

eventfilter=!Event: DeviceStateChange

- Restart the Wazo server

- On the CRM VM
  - Log into the CRM VM as superuser
    - Install FOP2

      cd /usr/src

      wget http://www.fop2.com/download/debian64 -O fop2.tgz

      tar zxvf fop2.tgz

      cd fop2

      make install

      Unles you have previously installed it, "make" does NOT come with the base Debian install so you will have to install make with

      apt install make

      The installation instructions above will copy the server files under /usr/local/fop2 and the web pages under /var/www/html/fop2 . It will also copy an init script for you and a configuration script to /usr/local/fop2/generate_override_contexts.pl . Do NOT run the configuration script; you do not need its functionality and it may reconfigure "Do Not Disturb" and "Call Forward" Feature Codes on you. I do not use FOP2 for controlling the PBX, just for Monitoring, so I do not want the Feature Codes adapted to FOP2's versions.

      systemctl restart apache2

    - Amongst other things, you will be defining a "user" in the next step that will be used by FOP Panel to login to Asterisk. You can create a "fake" user in Wazo, ie one that is defined just to allow FOP2 to connect but has no actual device connected. This will give you "readonly" status to the FOP Panel. If you want to see the full FOP2 toolbar, use a

real extension, ie one that actually has a softphone or device at the end of it that can make/take calls.

- Edit the configuration files to allow FOP2 to connect to the Wazo Asterisk AMI by changing the lines in the existing file as shown - using the same FOP2_AMI_UserName and FOP2_AMI_UserPassword created above in the Wazo AMI settings and the same {FOP2_Line_Number} and FOP2_Line_Password} created above in the Wazo User/Line setup

  cp -a /usr/local/fop2/fop2.cfg /usr/local/fop2/fop2.cfg.orig

  nano /usr/local/fop2/fop2.cfg

  manager_host={IP_address_of_CRM_VM}

  manager_port=5038

  manager_user={FOP2_AMI_UserName}

  manager_secret={FOP2_AMI_UserPassword}

  ; #exec autoconfig-users.sh

  (comment out the #exec line at the end of the file)

  user={FOP2_Line_Number}:{FOP2_Line_UserPassword}:all

  To enable FOP2 to act as an extension on the Wazo system and initiate calls, …

  buttonfile=buttons_custom.cfg

  To manually configure the button layout in the FOP2 panel layout, NOT using a MySQL database

  Enable the new configuration

  You can just reboot the server and get a fresh start - which I sometimes like to do

  Alternatively, you can reload the Apache (web server) and FOP2 configurations with

  service apache2 restart

  service fop2 restart

  Test if FOP2 can connect to Wazo with the setup created.

  /usr/local/fop2/fop2_server --test

  You should see the following output if the connection works (This is for the installation with a free (Demo) version of FOP2 installed; if you have a paid version, the first two lines will be different. The key is the third line which says "Connection to manager OK"

  Flash Operator Panel 2 - License file /usr/local/fop2/fop2.lic not found.

  Running in Demo Mode

  Connection to manager OK (060000)!

- Establish the credentials for the FOP2 Manager (admin) panel

FOP2 in its current version (2.31.27 comes with a FOP2 Manager that allows you to use a graphical interface, tied to a system database, to configure button and actions for FOP2.

This is different than the FOP2 Panel which is used after configuration to monitor and interact with the Asterisk installation.

You need to make the changes below to disable the FOP2 Manager and prevent FOP2 from trying to pull configuration from a non-existent database. It is non-existent from FOP2's persective, since FOP2 does not recognize Wazo as a known Asterisk package and we do not want to expose the Wazo SQL database to an external server (CRM VM) so we are using manual configuration to setup FOP2 with Wazo without accessing the database.

You can launch the FOP2 Manager to see what it looks like but for this setup, do not try to use it to create configurations.

cp -a /var/www/html/fop2/admin/config.php /var/www/html/fop2/admin/config.php.orig

nano /var/www/html/fop2/admin/config.php

This provides access to the FOP2 Admin Manager which you can normally use to configure FOP2, but we will be manually configuring FOP2 below since Wazo is not a standard installation for FOP2.

Change

```
define('USE_BACKEND_AUTH',true);
```

to

```
define('USE_BACKEND_AUTH',false);
```

Change

```
$ADMINUSER = "fop2admin";
$ADMINPWD  = "fop2admin";
```

to

```
$ADMINUSER = "{FOP2_Web_Admin_UserName}";
$ADMINPWD  = "{FOP2_Web_Admin_UserPassword}";
```

To access the FOP2 Manager you go to

{IP_address_of_CRM_VM}/fop2/admin and enter the {FOP2_Web_Admin_UserName} and {FOP2_Web_Admin_UserPassword} from above

- Customize FOP2 to get the buttons you want displayed on your display page

The UserName and Password for the FOP2 Panel (different than the UserName and Password for the Asterisk AMI and different than the UserName and Password for the FOP Manager) were created above as {FOP2_Line_UserName} and {FOP2_Line_UserPassword}. Again, this is required since you are not using an existing web interface (FreePBX, …) recognized by FOP2 so must create manaual authentication method to get to the FOP2 Panel and access Wazo from this "server"

Make sure to add a button for the "extension" you created with {FOP2_Line_UserName} and {FOP2_Line_UserPassword} and use a "real" extension (as opposed to one created just for FOP2 to connect to the Asterisk server) if you want the FOP2 toolbar to show wth available options. If you leave out this button, FOP2 is effectively put into "ReadOnly" mode (which actually may be what you want for some users)

nano /usr/local/fop2/buttons_custom.cfg

Extensions

[PJSIP/{Line_Name}]

type=extension

extension={Line_Extension}

context={Context_Name_(usually_Internal_Context_Name)}

label={Line_Label}

channel=PJSIP/{Line_Extension}

(Optional)

mailbox={Line_Mailbox}@{Context_Name}

group={Line_Group_Name}

Trunks

[PJSIP/{FOP2_Trunk_Name}]

type=trunk

label=(FOP2_Trunk_Label}

context={FOP2_Trunk_Context}

channel=PJSIP/{FOP2_Trunk_Name}

Conferences

[CONFERENCE/{FOP2_Conference_Number}]

Note: {FOP2_Conference_Number} is NOT the extension number, it is the sequence number so 1,2,3, … depending on how many conference extensions you have setup

type=conference

label={FOP2_Conference_Name}

extension={FOP2_Conference_Extension}

context={FOP2_Conference_Context}

As before, use the Context Name for the Internal context you created

Parking Lot

It is important that you use blind transfer to the {FOP2_ParkingLot_Extension} and do NOT use a bridge/conference/attended-transfer to transfer the call to the parking lot. Using the attended transfer will work to park the call, but FOP2 will see the hang-up after an attended transfer as a disconnect and return the

Parking Lot button to unused status.

[PARK/parkinglot-1]

This is the standard used by Wazo for the first Parking Lot you setup. If you are setting up multiple Parking Lots, activate the Asterisk Monitor with asterisk -rvvv
and connect to the conference so you can see the identifier used for that conference.

type=park

label={FOP2_ParkingLot_Name}

extension={FOP2_ParkingLot_Extension}

This is the extension used to reserve the Parking lot NOT the individual slots in the Parking Lot

context=parkedcalls

nano /usr/local/fop2/autobuttons.cfg

comment out the (only) line in this file with

; #exec autoconfig-buttons.sh $1

Reload the FOP2 configuration

service fop2 restart

View the FOP2 Panel by going to

{IP_address_of_CRM_VM}/fop2

and logging in with the {FOP2_Line_Number} and {FOP2_Line_UserPassword}

If you want to have the browser automatically bring up the FOP2 panel without entering a User/Password combination, you can add the User and Password to the URL as shown below. Make sure if you do this that the computer on which you do this is secure. If you used the real extension to login and are NOT using the Read-Only version of FOP2 (see above), this would give anyone with access to the computer the right to listen in on other's calls and many other capabilities. To enable automatic login to the FOP2 panel, use the following URL

{IP_address_of_CRM_VM}/fop2?user={FOP2_Line_Number}&pass={FOP2_Line_UserPassword}

(Optional, but recommended) The default for FOP2 is to have a fixed size for each "button", with 2 lines lines showing per extension. This leaves a lot of spare real estate not used in the display and reduces the number of buttons than can be shown on a page. To have each button only reserve as much space as needed, set Dynamic Line Display to On.

User (icon top right corner) -> Preferences -> Display -> Dynamic Line Display: On

- If you like what you see, and need more "buttons" for your system, you can now acquire a paid version of FOP2 which removes the restriction on the number of buttons

Activating the license

Go to www.fop2.com/buy.php , select the license that you want, and purchase it. I

use the Basic license for $40.

You will be emailed an activation code.

To activate the license on your server, you need to know the activation code, and you have to chose a registration name. That name will appear in the FOP2 footer for non white labeled versions with the legend 'Licensed to XXXX'. White label versions won't display any footer

Note If you manage several boxes and licenses, it is wise to use a name that will help find or keep track of the activation codes in the future.

The command to activate the license is:

/usr/local/fop2/fop2_server --register

It will prompt for the activation code and then the registration name. If you want to pass that information in one pass, then you can try:

/usr/local/fop2/fop2_server --register --code XXXX --name YYYY

Where XXXX is the activation code and YYYY the name you want to assign to it

The license binds to the hardware MAC address of your eth0 NIC card by default. If your server does not have an eth0 interface but uses a different name for it, like em1, or venet0, then you must have the -i command line parameter to the fop2_server command, like this:

/usr/local/fop2/fop2_server --register -i em1

Revoking the license

Some times you need to upgrade your hardware, change your network configuration, or move your virtual server. In those cases, your license will most probably break after the change. So, before doing anything to your installation, you must revoke your license with the command:

/usr/local/fop2/fop2_server --revoke

You will be prompted for your activation code. After entering it, and if everything works well, your activation code will be released so you can later use it to activate the software again after reinstalling/moving or changing your network configuration

Revokation only works on a valid licensed FOP2 copy, you cannot revoke an unregister/invalid license installation. To verify you have a valid license you can run:

/usr/local/fop2/fop2_server --test

i) Install Backup Scripts to put backup files in an offsite ftp location each night.

- WazoPBX runs its own nightly backup to a local directory /var/backups/xivo and rotates the backups on a 7 day cycle. There are two backup files created each day:
  - ◊ db.tgz
    - ♦ the asterisk PostgreSQL database
  - ◊ data.tgz
    - ♦ the directories and files associated with Wazo
    - ♦ See the Wazo administrators manual - Backup - for the specifics of which files and

driectories are backed up and what conditions could exist to have certain directories or files left out if the file size or number of files per directory get too large.

- The files in /var/backups/xivo are labelled as follows
  - ◊ The most current version of each backup is named
    - ♦ db.tgz
    - ♦ data.tgz
  - ◊ The next most current is renamed by the logrotate function as
    - ♦ db.tgz.1
    - ♦ data.tgz.1
  - ◊ And each of the other 5 days of the backup rotation are labelled 2 through 7 for each backup file
- To restore Wazo, there are several conditions that must be met
  - ◊ You must restore a backup on the same version of Wazo that was backed up
  - ◊ • You must restore a backup on a machine with the same hostname and IP address
- If the above conditions are met, to restore from a backup
  - ◊ Use the best version of the backup files, usually the most current, but possibly an earlier one if there was an undiscovered problem for 2-3 days
    - ♦ The commands shown assume restoring from the most current version; adjust as required
  - ◊ Always synch the restores by backing up the data AND the database with the same version
  - ◊ Restoring the data.tgz file also restores system files such as host hostname, network interfaces, etc.
    - ♦ You will need to reapply the network configuration if you restore the data.tgz file.
  - ◊ Stop the wazo service

    wazo-service stop
  - ◊ Restore the data files as superuser with

    cd /var/backups/xivo

    tar xvfp /var/backups/xivo/data.tgz -C /
    - ♦ Warning: Restoring the data.tgz file also restores system files such as host hostname, network interfaces, etc.
    - ♦ You will need to reapply the network configuration if you restore the data.tgz file.
  - ◊ Restore the datagbase as superuser with

    cd /var/backups/xivo

    tar xvf db.tgz -C /var/tmp

    cd /var/tmp/pg-backup

    sudo -u postgres dropdb asterisk

    sudo -u postgres pg_restore -C -d postgres asterisk-*.dump

- ◊ Restore the server UUID
  - ♦ XIVO_UUID=$(sudo -u postgres psql -d asterisk -tA -c 'select uuid from infos')
  - ♦ echo "export XIVO_UUID=$XIVO_UUID" > /etc/profile.d/xivo_uuid.sh
  - ♦ nano /etc/systemd/system.conf

    to update XIVO_UUID in the DefaultEnvironment variable

    Pull the value from /etc/profile.d/xivo_uuid.sh above
- ◊ Reboot the system

- For transferring files to a remote ftp site, see instructions and script in section below. You do not need to manually generate a database backup since it has already been done. Edit the file to not tar the already tarred files; you just send the backup .tgz files to the ftp.

j) Configure iptables and fail2ban to secure the system

- See the iptables and fail2ban sections below for a full explanation of configuring and using iptables as a firewall and fail2ban as a hacker detection to protect your system.
- I use a whitelist approach with iptables, meaning I block everything and then explicitly identify the traffic I will allow into the server. Although it deos require some configuration work-arounds (shown below), I find this much more effective at creating a secure system, given all the spoofing and hacking going on.
- In addition to the standard iptables setup, the ports to allow in the INPUT chain of the FILTER table for Wazo are (see https://wazo-platform.org/uc-doc/contributors/network for a full listing of Wazo Ports)
  - ◊ 80 and 443 for web access by the Wazo GUI
  - ◊ 8667 for devices to fetch their configuration from Wazo
  - ◊ 5060 and 10000 to 20000 for SIP traffic
- You do NOT need to forward any ports to the Wazo server unless you have satellite locations with phones connected to your Wazo server from outside your LAN. The trunk provider is being accessed by a registration to them from Wazo that originated from the server so this is allowed as is its retrun traffic. Inbound calls to Wazo from the trunk provider use this registration and Outbound calls originate from the Wazo server so again, it is allowed.

7) Installing SuiteCRM

a) These instructions assume a server with 16GB RAM, 4-core CPU and a 1TB hard disk. Adjust your paramaters to suit your installation.

b) Install the Debian (10 = Buster) Operating system

- See the WazoPBX installation instructions for the Proxmox setup process, with any unique settings identified here:
  - ◊ General (tab)
    - ♦ VM ID: 101
    - ♦ Name: SuiteCRM
    - ♦ Start/Shutdown order: 2

- ◊ Hard Disk (tab)
  - ♦ Storage: lv-suitecrm
  - ♦ Disk Size (GiB): 182 (185.79 available)
- ◊ CPU (tab)
  - ♦ CPU Units: 1024
- ◊ Memory
  - ♦ Memory (GiB): 14336
  - ♦ Minimum memory (GiB): 2048

- See the Debian installation instructions for the OS setup process, with any unique settings identified here:
  - ◊ Screen 5: Different {Hostname}
  - ◊ Screen 20
    - ♦ Debian desktop environment: Uncheck
    - ♦ print server: Check
    - ♦ SSH server: Check
    - ♦ Standard system Utilities: Check
    - ♦ All others: Uncheck

c) Update the basic OS install with the following

- ◊ See the Debian installation for generic instructions
- ◊ When instructions/variables are specific to this VM, they will be shown here

- Change the repository list to add the non-free Debian repositories and to switch to a local mirror for faster access and downloads
- Update the package list and upgrade the system (apt update && apt -y dist-upgrade)
- Change to a fixed IP address
  - ◊ IP for SuiteCRM: {server_ip_address}
- Add a non-root user to facilitate key-pair SSH access instead of user/password authentication (and in this case for using the network file sharing capability (optional) explained later)
- (Optional) Add the sudo package and give the new non-root user sudo access (and test before proceedin to next step)
- Set so remote access is not allowed with root (leaving only the new, non-root user with ability to remotely access)
- Add /sbin to PATH vairable
- Set so SSH access is only allowed via key-pair, not user/password
- Edit the setup so SSH access has colour-coded prompts to distinguish root and non-root users at the terminal prompt
- Configure a mail server to email out only (no inbound or relay)

- ◊ Use the non-Postfix installation option for SuiteCRM
- Install unattended-upgrades to have system automatically install security upgrades
- Install logwatch to have the system email a status report to the administrator every day
- Install scripts to email the administrator on login for both root and user

d) Add the nfs client to the OS (AFTER making sure the client IP address matches one allowed by the host's export setup)

- ♦ Create mount points

  mkdir -p /mnt/pve/datashare

  chmod 755 -R /mnt/pve/datashare

  cd /mnt/pve/datashare

  chown {UserName}:{UserName}-R . [do not forget the period at the end]

- ♦ Repeat above for each shared drive like the disk2 drives:

  mkdir -p /mnt/pve/disk2-archives

  mkdir -p /mnt/pve/disk2-mintrchome

- ♦ Now mount the nfs storage

  nano /etc/fstab and add, at the end of the file

  # Mount the directory in this VM pointing to the volume in the host at IP address {IP address of host (Proxmox) server}

  {IP address of host (Proxmox) server}:/var/lib/vz/datashare /mnt/pve/datashare nfs4 defaults,sync 0 0

  {IP address of host (Proxmox) server}:/var/lib/vz/other1 /mnt/pve/other1 nfs4 defaults,sync 0 0

  {IP address of host (Proxmox) server}:/var/lib/vz/other2 /mnt/pve/other2 nfs4 defaults,sync 0 0

  mount -a

e) Add the elements needed to support a web server - do them in the sequence shown

- You have the L (for Linux) portion of the LAMP stack installed
- Now install the other three elements of a LAMP stack: the web server (A for Apache) database (M for MySQL or, in our case, MariaDB) and the programming language (P for PhP)
- To see if a package is already installed

  dpkg-query -l *<package_name_(partial)>*

  [Use * as a widlcard before and/or after name if unsure of full name]

- To see what package(s) are vaialble in the repositories

  apt-cache showpkg <package_name_(partial)>

- Install in the sequence shown so the dependencies are properly configured
- Install Apache

apt install apache2 apache2-utils

- ◊ Not required, but helpful if you want on-screen documentation

  apt install apache2-doc

- ◊ Not required, but install if you want a text-based, command-line browser (used if try to get status of apache2 from command line

  apt install elinks

- ◊ Enable the SSL modfule for apache and enable the ssl-site configuration for apache

  a2enmod ssl

  a2ensite default-ssl.conf

  systemctl reload apache2

- Install MariaDB - a binary-compatile alternative to MySQL (now that Oracle has MySQL)
  - ◊ See the instructions in the MariaDb section below
- Install PHP
  - ◊ Do NOT put both these commands on the same line; do them separately so dependencies are properly installed

    apt install php

    apt install php-mysql libapache2-mod-php
- Install other packages reuired by SuiteCRM
  - ♦ JSON
  - ♦ XML Parsing
  - ♦ MB Strings Module
  - ♦ ZLIB Compression Module
  - ♦ ZIP Handling Module
  - ♦ PCRE Library
  - ♦ IMAP Module
  - ♦ cURL Module
  - ♦ Upload File Size
  - ♦ Sprite Support

    apt install libc-client2007e libzip4 mlock php-imap php-zip php-imap php-zip

    apt install php-gd php-curl php-mbstring php-xml php-dom php-intl

    apt install composer

    a2enmod php7.3
- Test the operation of Apache with php
  - ◊ (if not already done above) Save a file in /var/www/html called php_info.php with contents

    nano /var/www/html/php_info.php

<?php phpinfo(); ?>

- ◊ Point your browswer to <server_ip_address>/php_info.php
- ◊ If all is working properly you will see a long page filled with specifications
  - ♦ Check "APACHE_RUN_USER" and you will see the Apache user is www-data

- Install Webmin
  - ◊ Add the webmin repository

    apt install gnupg1

    nano /etc/apt/sources.list.d/webmin.list

    deb https://download.webmin.com/download/repository sarge contrib

    cd /root

    wget http://www.webmin.com/jcameron-key.asc

    apt-key add jcameron-key.asc

    apt update
  - ◊ Run the Installer

    apt-get install apt-transport-https

    apt install webmin

    apt update && apt upgrade [likely not necessary but just as a final check]
  - ◊ (Optional) Security options with Webmin
    - ♦ As a safety precaution, if you want (NOT considered necessary), you can disable webmin

      systemctl stop webmin

      systemctl disable webmin

      And then re-enable or start it any time you want

      systemctl start webmin

      [If you only want it on for the current sesison - it will not be available after a reboot]

      systemctl enable webmin

      [If you want it available from now on after a reboot]
    - ♦ You culd also ensure that the current Webmin stays installed and not risk issues with regression by sidabling the webmin repository (I did NOT do this)

      Go back with nano /etc/apt/sources.list.d/webmin.list and make sure to comment out the line with deb http: ...

      Save the updated file

      Re-run apt update
    - ♦ Change the Port and require https for security

      Connect to server on port 10000 via https to bring up Webmin window

      https://<server_ip_address>:10000

Accept insecure certificate warning

Login using server root credentials root and {root-level-password}

Webmin -> Webmin Configuration -> Ports and Addresses

Listen on IPs and ports: 42536

Acccept IPv6 connections: No

Listen for broadcasts on UDP port: 42536

Save

From now on, connect to server on port 42536 via https to bring up Webmin window

https://<server_ip_address>:42536

Accept insecure certificate warning

Login using server root credentials root and {root-level-password}

f) Install the SuiteCRM application

- Install Zip packages to allow compression and downloading of directories and their contents
  - ◊ Note that this does NOT retain file permission or ownership settings; you need to tar not zip for that, but SuiteCRM installs better with .zip files

apt install zip unzip

- Change settings to enable two php.ini settings, one for normal operation and one for installs and upgrades of SuiteCRM
  - ◊ Create alternative php.ini files and (edit below)

cp /etc/php/7.3/apache2/php.ini /etc/php/7.3/apache2/php.ini.orig

cp /etc/php/7.3/apache2/php.ini /etc/php/7.3/apache2/php.ini.normal

cp /etc/php/7.3/apache2/php.ini /etc/php/7.3/apache2/php.ini.updates

  - ◊ Normal, operating settings

nano /etc/php/7.3/apache2/php.ini.normal

Change

memory_limit = 128M

to

memory_limit = 256M

Change

upload_max_filesize = 2M

to

upload_max_filesize = 12M

systemctl restart apache2

  - ◊ Settings for when you are upgrading the system
    - ♦ See https://docs.suitecrm.com/admin/installation-guide/upgrading for current instructions

nano /etc/php/7.3/apache2/php.ini.updates

Change

memory_limit = 128M

to

memory_limit = 256M

Change

upload_max_filesize = 2M

to

upload_max_filesize = 100M

Change

post_max_size = 8M

to

post_max_size = 100M

Change

max_input_time = 60

to

max_input_time = 300

Change

max_execution_time = 60

to

max_execution_time = 6000

Change

;opcache.enable=1

to

opcache.enable=0

◊ Now, when you are updating, you can quickly change to the .updates .ini file and when you are done updating, you can quickly change back to the .normal .ini file.

♦ When you want to update

cp -i /etc/php/7.3/apache2/php.ini.updates /etc/php/7.3/apache2/php.ini

[the -i flag makes this command interactive so you have to confirm that you want to overwrite the existing file]

systemctl restart apache2

♦ When you are finished updating and want to return the system to normal operation

cp -i /etc/php/7.3/apache2/php.ini.normal /etc/php/7.3/apache2/php.ini

[the -I flag makes this command interactive so you have to confirm that you want to overwrite the existing file]

systemctl restart apache2

Re-run the file / directory permissions commands

cd /var/www/html/{suitecrm_directory_name}

chown -R www-data:www-data .

chmod -R 755 .

chmod -R 775 cache custom modules themes data upload

chmod 775 config_override.php 2>/dev/null

Repair and Rebuild using SuiteCRM Admin interface

- To Update an already installed system, Run the Update Wizard from Admin -> System -> Upgrade Wizard
  - ◊ After updating, SuiteCRM stores large files in /var/www/html/{suitecrm_directory_name}/upload/upgrades/patch and over time this will significantly increase (in my case triple) the size of the backup file if you backup the entire /var/www/html/{suitecrm_directory_name} directory. By keeping these files,, you enable a rollback of SuiteCRM to a version pre-upgrade. Once you have upgraded SuiteCRM and are comfortable the upgrade is stable, you can clear the files and directories in /var/www/html/{suitecrm_directory_name}/upload/upgrades/patch . After cearing these files and directories, the Upgrade Wizard will no longer show the upgrades that have been applied but you can always just see which version of SuiteCRM ou are on via the UI.
  - ◊ Use the Upgrade Wizard
    - ♦ See https://docs.suitecrm.com/admin/installation-guide/using-the-upgrade-wizard
    - ♦ for current instructions
  - ◊ If you have changed the composer.json file (Not sure how to tell so I always do this), as superuser (root), run the command composer install --no-dev impersonating the www-data user from the CLI by

    cd /var/www/html/{suitecrm_directory_name}

    su - www-data --shell /bin/bash --command 'cd /var/www/html/{suitecrm_directory_name} && composer install --no-dev'

    su - www-data --shell /bin/bash --command 'cd /var/www/html/{suitecrm_directory_name} && composer update'
  - ◊ Reset the php-ini per above
  - ◊ Set file permissions per above
  - ◊ Rebuild and Repair the system with Admin -> System -> Repair -> Quick Repair and Rebuild
  - ◊ Starting with v7.11.3, there is a command-line alternative to doing the upgrade. I do not use this since I like to see what is happening during the upgrade, but it is documented here for reference.
    - ♦ cd {suitecrm_directory_name}
    - ♦ Download the required upgrade pack and place it in the folder with the installed system

wget {URL of {SuiteCRM_upgradeZipFile}}

- From the folder with the installed system, run the command

  ./vendor/bin/robo upgrade:suite {SuiteCRM_upgradeZipFile} {logFile} {pathToSuiteCRMInstance} {adminUser}

  {SuiteCRM_upgradeZipFile} = downloaded upgrade package

  {logFile} = log file name (full or relative path)

  {pathToSuiteCRMInstance} = the full or relative path to the installed SuiteCRM instance (. (period) for relative path since you are in the installation root directory)

  {adminUser} - name of user with administrative rights

  For example

  ./vendor/bin/robo upgrade:suite SuiteCRM-Upgrade-7.11.x-to-7.11.4.zip upgradeLog.log . admin

- Now get and install SuiteCRM

  cd /var/www/html

  wget https://suitecrm.com/files/160/SuiteCRM-7.10/480/SuiteCRM-7.10.22.zip

  [at the time of this document, the most recent LTS version (which I want) was 7.10.22]

  ls -al [to get the filename of the .zip archive]

  unzip SuiteCRM-7.10.22.zip

  ls -al [to get the name of the directory created by extracting the .zip archive]

  mv SuiteCRM-7.10.22 {suitecrm_directory_name}

  ls -al [to confirm the directory name has been changed]

  [set thte directory and file permissions - you may have to re-do this after updates as well]

  cd /var/www/html/{suitecrm_directory_name}

  chown -R www-data:www-data .

  chmod -R 755 .

  chmod -R 775 cache custom modules themes data upload

  chmod 775 config_override.php 2>/dev/null

  - Install Composer locally in the {suitecrm_directory_name} directory within /var/www/html
    - Do this as superuser

      cd /var/www

      mkdir -p /var/www/.composer/cache/repo/https---repo.packagist.org

      cd /var/www/.composer/cache

      mkdir files

      chown www-data:www-data -R files

      chown www-data:www-data -R repo

namei -om /var/www/.composer/cache

[to confirm the permission settings of the whole path to, including final directory]

namei -om /var/www/.composer/cache/files

namei -om /var/www/.composer/cache/repo/https---repo.packagist.org

- Now you need to actually install composer locally to the {suitecrm_directory_name} directory.

  Stay with me here since this is going to sound convoluted but hopefully the explanation will clarify

  Once you have installed the composer application (as superuser, like any other application installation, using apt install composer) you now need to either enable composer globally (which we are NOT doing here) or enable composer per directory, which we ARE doing here for the {suitecrm_directory_name} directory

  You MUST enable composer as user so it puts its configuration files in the correct place

  Problem is, the user for the web directory is the apache user www-data and we have logged in as ourselves {user_name}so to run composer commands as {UserName} when the files are set with www-data permissions, does not work

  I tried adding {user_name} to the www-data group with usermod -a -G www-data {user_name}and, after logging out and back in (required to recognize new group memberships) I confirmed {user_name} was part of the www-data group with "groups" (or "id")

  However, this did not fully solve the issue as I was still getting permission errors

  The next solution would have been to use a sudo command to impersonate the www-data user for one command with sudo --user www-data {command} (example sudo --user www-data composer install) but since we did not install sudo (for security reasons), we cannot use that solution

  We cannot just use the su (switch user) command to impersonate www-data since www-data is not a user with login ability so when the su comand asks for a Password, we won't have one

  So we are left with what seems like a contradictory solution, but it is not

  We are going to login as superuser and from superuser run a command as www-data

  cd /var/www/html/{suitecrm_directory_name}

  su - www-data --shell /bin/bash --command 'cd /var/www/html/suiteayudacrm && composer install'

  su - www-data --shell /bin/bash --command 'cd /var/www/html/suiteayudacrm && composer update'

  The --shell option is required to prevent the request for Password

  And wait; it takes a while

  Re-run the file / directory permissions commands

  cd /var/www/html/{suitecrm_directory_name}

  chown -R www-data:www-data .

chmod -R 755 .

chmod -R 775 cache custom modules themes data upload

chmod 775 config_override.php 2>/dev/null

◊ Patch a bug (required as of v7.11.12 or 7.10.24)

♦ There is a patch for this that will be included in future releases, but for now, you need to add a patch that fixes a bug in the system when 2-factor authentication is enabled for a user. Until this patch is installed, for any user that has had 2-factor authentication enabled, when they login, they will be sent to their Profile page instead of their Home page. The Home page is the desired landing page since that shows users all the current information for them.

cd cd /var/www/html/{suitecrm_directory_name}

cp -a modules/Users/authentication/SugarAuthenticate/FactorAuthEmailCode.tpl modules/Users/authentication/SugarAuthenticate/FactorAuthEmailCode.tpl.orig

nano modules/Users/authentication/SugarAuthenticate/FactorAuthEmailCode.tpl

in the section

```
<body>
   <div class="p_login">
      <div class="p_login_top">
         <a title="SuiteCRM" href="http://www.suitecrm.com">SuiteCRM</a>
      </div>

   <div class="p_login_middle">
      <div id="loginform">
         <div class="error message">{$factor_message}</div>
         <form method="post" action="index.php?module=Users&action=index">
            {$APP.LBL_EMAIL_CODE} <input type="text" name="factor_token">
            <input type="submit" value="{$APP.LBL_VERIFY}">
         </form>
         <a
href="index.php?module=Users&action=Logout">{$APP.LBL_CANCEL}</a>&nbs
p;  
         <a
href="index.php?module=Users&action=Resend">{$APP.LBL_RESEND}</a>
      </div>

      <div class="p_login_bottom">
         <a id="admin_options">© Supercharged by SuiteCRM</a>
         <a id="powered_by">© Powered By SugarCRM</a>
```

```
        </div>
      </div>
    </body>
```

Change

```
<form method="post" action="index.php?module=Users&action=index">
```

to

```
<form method="post" >
```

◊ Go to https://{server_IP_address}/{suitecrm_directory_name}/install.php

- ♦ Go past the certificate warning, if present
- ♦ Welcome Page: Accept license
- ♦ Configuration Check Page: Click Next [assuming all checks come back OK]

  Note the instructions to come back after installation and configure CRON jobs to run SuiteCRM schedulers (do as superuser):

- ♦ Configuration Page

  Database Configuration

  Database Type: MySQL

  [MariaDB is binary compatible so looks to the system like MySQL]

  Database Name:

  Database Name: suiteayudacrm

  Host Name: localhost

  SuiteCRM Database User: Select, from Dropdown menu: Provide existing user

  SuiteCRM Database User: {db_SuiteCRM_user_name}

  SuiteCRM Database User Password: {db_SuiteCRM_user_password}

  Re-enter SuiteCRM Database User Password: {db_SuiteCRM_user_password}

  User Configuration

  SuiteCRM Application Admin Name: {SuiteCRM Application Admin Name}

  SuiteCRM Admin User Password: {SuiteCRM Admin User Password}

  URL of SuiteCRM Instance: {SuiteCRM URL} [likely leave as is]

  Email Address: {SuiteCRM Email Address}

  Choose Demo Data

  No

  Scenario Selection

  Leave all chaecked

  SMTP Server Specification

  "From" Name:

"From" Address:

Choose your Email provider: Other

SMTP Server:

SMTP Port:

Use SMTP Authentication?

Enable SMTP over SSL or TLS?

SMTP Username:

SMTP Password:

Allow users to use this account for outgoing email: Checked

Branding

Name:

Select Logo:

There is a BUG in this version that hangs the system when you try this. Leave it alone for now and upload later from within the Admin -> System settings

The max size is 450px wide x 170px tall

It must be a .jpg or .png file (.png for transparent background)

Make sure the filename has no spaces or special characters in it

System Locale Settings

Date Format: 2012-12-23

Time Format: 11:00pm

Time Zone: America/Toronto

Currecy: Canadian Dollars

Currency Symbol: $

ISO 4217 Currency Code: CAD

Site Security

Leave all as is (unchecked)

◊ Wait for the configurator to finish the installation

◊ Add the CRON entries from the initial screen

crontab -e -u www-data

♦ and add the following lines to the crontab file:

SuiteCRM Schedulers

* * * * * cd /var/www/html/{suitecrm_directory_name}; php -f cron.php 1>/dev/null 2>&1

◊ (Optional) Creating the favicon for your site

♦ If you know what a favicon is, then this is where you can add it to your site. If you do not know what a favicon is, ignore this section or Google favicon and see if you want to

use it. This is completely optional and only cosmetic so feel free to skip if you want. If you want the browser (Firefox yes, MS ie not always so reliable) to display a miniature picture in the tab for your site:

- ♦ Go to /var/www/html/{suitecrm_directory_name}/themes/default/images

  rename sugar_icon.ico to sugar_icon.ico.orig

  upload the favicon you want to show (a 16 pixel x 16 pixel .ico file)

  rename the favicon you uploaded to sugar_icon.ico

  rename sugar_icon.png to sugar_icon.png.orig

  upload the higher-res version of the favicon you want to show (a 57 pixel x 57 pixel .png file with 106 pixel per inch resolution)

  rename the hi-res favicon you uploaded to sugar_icon.png

- ♦ Repeat the above procedure for the /var/ww/html/{suitecrm_directory_name}/themes/SuiteP/images directory
- ♦ To make this effective, you will need to reboot your system, but that can wait until later when you reboot it for another reason

◊ Reboot the server

shutdown -r now

g) Configure the SuiteCRM content

- Point your browswer to http://{server_IP_address}/{suitecrm_directory_name}
- Login with your {SuiteCRM Application Admin Name} and {SuiteCRM Admin User Password}
- Rebuid and Repair:
  - ◊ User Dropdown (top-right) -> Admin -> System -> Repair
  - ◊ Quick Repair and Rebuild
  - ◊ Scroll to bottom to see if any actions are required, and if not Return to Administration Page
- Add a non-root user
  - ◊ Administration (top-right corner) -> Admin
  - ◊ Users -> User Management -> Create New User
    - ♦ User Profile (tab)

      User Profile (section)

      User Name: {CRM_Reg_User_Name}

      Password for TestCRM: {CRM_Reg_User_Password}

      First Name:

      Last Name:

      Status: Active

      User Type: Regular User

Employee Information

... populate as wanted ...

Email Settings

Email Address:

Editor: TinyMCE

- ♦ Password (tab)

  Password:

- ♦ Advanced (tab)

  User Settings:

  ... populate as wanted ...

  Locale Settings

  First day of week: Monday

  ... populate rest as wanted ...

- ♦ Layout Options (tab)

  Layout Options (section)

  Move Comapigns higher in order, right after Contacts, to appear higher in "All" menu

◊ Save

- Do any additional Configurations you want using the Admin functionality

h) Manually backing up and Restoring SuiteCRM from existing SuiteCRM installation

- To ensure database architecture compatibility for a restore, either first update the existing installation to the same version as the new SuiteCRM into which the set is being imported (preferred) or have the new SuiteCRM be installed as the same SuiteCRM version as was used for the existing installation (
- In the existing SuiteCRM, create a directory to hold backups (assuming it does not already exist)

  cd /home/{UserName}

  mkdir backups

- Backups for the existing SuiteCRM have two components:
  - ◊ The database in which SuiteCRM stores variables, data and settings
    - ♦ Do this as user, NOT superuser
    - ♦ Backed up locally by

      mysqldump --user={db_SuiteCRM_user_name} --password={db_SuiteCRM_user_password} --verbose --routines --triggers {old_database_name} > /home/{UserName}/backups/(filename_holding_dump}.sql
  - ◊ The web site structure itself
    - ♦ Do this as user, NOT superuser
    - ♦ Backed up locally by creating a tarball of the full directory holding SuiteCRM

cd /var/www/html

tar -cvzf /home/{UserName}/backups/{filename_holding_tarball}.tar.gz {suitecrm_directory_name}/

- In the new SuiteCRM, do all the prep work shown above for preparing a VM for accepting SuiteCRM, including OS updates, installing the additional packages, …
- In the new SuiteCRM, create a directory to hold backups (assuming it does not already exist)

  cd /home/{UserName}

  mkdir backups

  - ♦ Transfer the two files created above (directory and databasedump) into the backups directory on the new system
- Restoring to a new SuiteCRM has two components:
  - ◊ The database in which SuiteCRM stores variables, data and settings
    - ♦ Do this as superuser, NOT user
    - ♦ Make sure the new system has a MySQL user and password and database identical to the one that was used in the old SuiteCRM.
    - ♦ In the old SuiteCRM, look in

      /var/www/html/{old_SuiteCRM_directory_name}/config.php

      for the values entered for

      'db_user_name'

      'db_password'

      'db_name'
    - ♦ In the new SuiteCRM

      Use the MySQL commands shown earlier in the MySQL section to create identical user, name and (empty) database in the new MySQL database
    - ♦ Restored from a local "dump" file by

      mysql --user={db_SuiteCRM_user_name} --password={db_SuiteCRM_user_password} --verbose {new_database_name} < (filename_holding_dump}.sql
  - ◊ The web site structure itself
    - ♦ Do this as superuser, NOT user
    - ♦ Restored from a local tarball of the full directory of the old SuiteCRM

      cd /var/www/html

      tar -xvzf {path_to_filename_of_tarball}.tar.gz --directory {path_to_directory_to_hold_uncompressed_tarball}

      In /var/www/html/{ SuiteCRM_directory_name}/config.php , change the entries for

      site_url to match the new site url / ip address

      host_name (2nd entry) to match the new site hostname
  - ◊ If not already in place (ie if NOT restoring to a system that already had SuiteSRM rrunning)

crontab -e -u www-data

- and add the following lines to the crontab file:

  # SuiteCRM Schedulers

  * * * * * cd /var/www/html/{suitecrm_directory_name}; php -f cron.php 1>/dev/null 2>&1

◊ Final touches

- See instructions above for a fresh SuiteCRM installation and

  Create Composer cache directories with proper pemissions

  Update Composer

  Run commands to set file/directory permissions

  Rebuild and Repair SuiteCRM

i) Install Backup Scripts

- See instructions and script in section below
- See paramters for SuiteCRM in Appendix 1

8) Installing LinuxMint

a) These instructions assume a server with 16GB RAM, 4-core CPU and a 1TB hard disk. Adjust your paramaters to suit your installation.

b) Install the Linux Mint 19.3 Operating system

- See the WazoPBX installation instructions for the Proxmox setup process, with any unique settings identified here:

◊ General (tab)

- VM ID: 102
- Name: LinuxMint
- Start/Shutdown order: 3

◊ Hard Disk (tab)

- Storage: lv-linuxmint
- Disk Size (GiB): 275 (279.21 available)

◊ CPU (tab)

- CPU Units: 2048

◊ Memory

- Memory (GiB): 14336
- Minimum memory (GiB): 2048

- For additional information on Linux Mint go to

  https://linuxmint.com

  https://linuxmint.com/download.php

- Download the 64 bit, LM 19.3 Mate Interface version from

  https://linuxmint.com/download.php

  which in my case was

  https://linuxmint.com/edition.php?id=276

  - Burn the ISO to disk, insert into the Optical Drive of the host, From the Proxmox web UI, Start the LinuxMint VM created above and click on the Console button to manage the installation

    Note that in the 19.3 Install ISO I had, Linux Mint would not come back after installation of 19.3. I did check the SHA256SUM of the ISO and it matched so it was not a bad ISO. I tried burning from different burners. No difference. It would hang just after the LinuxMint logo appeared. The forums had numerous users talking about this. So I installed 18.3 - which worked fine - and then used the Linux Mint update instructions to update to 19.3 (in two stages)

    If this happens to you, and you want to try a fresh 19.3 installation without the two-stage process identified above, you might want to try using some of the steps in this post to clear past the point where it hangs
    https://community.linuxmint.com/tutorial/view/2416

      If upon reboot, the computer fails to boot and the boot sequence seems stuck, type the left or right arrow to switch from the boot logo to the boot details.

      If the boot is stuck trying to run the /dev/mapper/cryptswap1 job, then do the following:

        Boot the computer with the Shift key pressed to force the Grub menu to show

        Choose "Advanced Options" for the latest kernel entry

        Choose "Recovery mode"

        Once in the recovery menu, choose "fsck" and choose "yes".

        Once fsck is done, press Enter to go back to the menu.

        Choose "root" from the recovery menu and press "Enter" to start the root console.

        Type "nano /etc/fstab" to edit the fstab file.

        Find the line with "/dev/mapper/cryptswap1" and add a # sign in front of "/dev/mapper/cryptswap1"

        Press "Ctrl+O" and then "Enter" to save the file

        Press "Ctrl+X" to exit the nano editor

        Type "reboot" to restart the computer

    If you are going to do the sequential 18.3 -> 19.3 installation, I suggest you load 18.3, and, before running the upgrade to 19.3, as superuser

      apt update && apt dist-upgrade -y && apt autoremove && apt autoclean

      shutdown -r now

    The upgrade from 18.3 to 19.1 is done as superuser, by (see https://blog.linuxmint.com/?p=3615 and

https://community.linuxmint.com/tutorial/view/2416)

(As user) run Timeshift and create a restore point

Give your terminal unlimited scrolling

Open a terminal (resore to normal after)

Click on "Edit"->"Profile Preferences"->"Scrolling".

Check the "unlimited" option and click "OK"

Elevate to superuser with

sudo su

user password

apt install mintupgrade

Revert to normal user (Exit superuser)

mintupgrade check

[This does NOT actually install anything; it shows you what would happen if you DID run the installer (next)]

mintupgrade download

mintupgrade upgrade

[This will take hours, so walk away and do something else for a while]

[You are required to re-authenticate your user password a couple of times so check in once and a while]

sudo shutdown -r now

Now upgrade to 19.3

apt update && apt upgrade && apt autoremove && apt autoclean

Open Update Manager and update to the latest kernel

apt update && apt upgrade && apt autoremove && apt autoclean

Open Update Manager and update to the latest kernel

[No, this is not a typo; do it twice. In fact it may take 3 times]

apt update && apt upgrade && apt autoremove && apt autoclean

Open the Update Manager

Edit -> Upgrade to “Linux Mint 19.3 Tricia”

Confirm the request and wait for this update

apt update && apt upgrade && apt autoremove && apt autoclean

Return your terminal to normal scrolling parameters

Open a terminal (resore to normal after)

Click on "Edit"->"Profile Preferences"->"Scrolling".

Enter 3000 lines

apt install xfce4-xapp-status-plugin xfce4-xapp-status-plugin libxfce4util-bin

Right-click on task Panel and add Applet - Xapp Status Applet (scroll to bottom) - then Right Click on the task Panel again and Reset the Panel to remove the gap in the middle.

Click on the Warning icon that appears bottom-right of task Panel and Click "I understand" for the warning about root password. I do NOT want to add one.

Reboot the computer

c) Add the nfs client and point to the datashare storage (see instructions in WazoPBX section), making sure the client IP address matches the one allowed by the host's export setup

- Add the nfs-common library and activate it

  apt install nfs-common

  modprove nfs

  cat /proc/filesystems | grep nfs

- Create mount points

  mkdir -p /mnt/pve/datashare

  mkdir -p /mnt/pve/disk2-archives

  mkdir -p /mnt/pve/disk2-mintrchome

  cd /mnt/pve

  chown {UserName}:{UserName} -R . [do not forget the period at the end]

  chmod 755 -R .

  ls -al [to confirm settings]

- Mount the nfs storage

  nano /etc/fstab and add, at the end of the file

  # Mount the directory in this VM pointing to the volume in the host at IP address {IP address of host (Proxmox) server}

  {IP address of host (Proxmox) server}:/var/lib/vz/datashare /mnt/pve/datashare nfs4 defaults,sync 0 0

  {IP address of host (Proxmox) server}:/var/lib/vz/datashare /mnt/pve/datashare nfs4 defaults,sync 0 0

  {IP address of host (Proxmox) server}:/var/lib/vz/datashare /mnt/pve/datashare nfs4 defaults,sync 0 0

  mount -a

- Add bookmarks for the above to the File Manager in Linux Mint

  Click on the File Manager icon in the taskbar

  Go: File System -> mnt -> pve

  For each of the above (datashare, disk2-archives, disk2-mintrchome)

  Open the folder

  Click bookmarks -> Add a bookmark

Right-click on the bookmark stored in the left column of the File Manager

Select Rename

Name the Bookmark whatever you want.

Retunr up one level (to /mnt/pve) in the File Manager

d) Refer to the document "Installing Linux Mint 19 … for further installation and configuration instructions"

e) (Optional) Since you will be using a remote desktop client to access Linux Mint, if you use standard VNC connections for remote access, you will not have the ability to hear, on the remote client, any audio generated by the Linux Mint VM. Additionally, the normal VNC-based client / server setup for remote access is relative resource and bandwidth intensive so there is a noticeable lag with remote GUI operation. To change this, use SPICE protocols to do the remote connection. To do this, you need to activate SPICE on the Linux Mint VM and use a SPICE Client on the machine you use to connect to the Linux Mint VM. This does NOT deactivate the VNC client/server capability; it gives you another option.

- See description of SPICE on Proxmox and instructions for use at
  - ◊ https://pve.proxmox.com/wiki/SPICE
  - ◊ https://pve.proxmox.com/wiki/File:Screen-enable-spice.png
- Install a SPICE client on Windows 7
  - ◊ Go to https://virt-manager.org/download
  - ◊ Download and install the Windows client
    - ♦ In my case, for the current 64 bit client it was virt-viewer-x64-8.0.msi at https://virt-manager.org/download/sources/virt-viewer/virt-viewer-x64-8.0.msi
- Enable SPICE on the Linux VM
  - ◊ Install the spice library in the VM

    apt install spice-vdagent
  - ◊ Reboot the VM
  - ◊ Use the Chrome browser, NOT Firefox for this configuration; Firefox handles mime transfers in a unique qway that will not provide the functionality described below
  - ◊ In the Proxmox GUI window, select the Datacentre in the left column and in the Options under Console Viewer select SPICE (remote viewer)
    - ♦ This is required if you want audio to pass from the Linux Mint VM. If you do not want audio to pass to the remote client from the Linux Mint VM, you do not need to set this. I did set this, which then has the Host Shell using SPICE, not VNC as its protocol.
  - ◊ In the Proxmox GUI window, expand the Datacenter and the Host listings in the left column and select the Linux Mint VM.
    - ♦ To change the Display settings: Under Hardware -> Display

      Select SPICE

      If you have a linked graphics card (I do not; I use a virtual graphics driver), and you are using Hi-Resolution display, you should increase the Mb to 32 from the default of 16

You will see Default still showing with SPICE (qxl) whoing under it in red. This means you need to re-boot the Linux Mint VM to activate the SPICE protocol. Do that after setting audio below.

- ♦ To add Audio capability: Under Hardware, click the Dropdown Add menu and select

  To determine what chipset and audio drivers are present on your host system, at the CLI of your host system

  cat /sys/devices/cpu/caps/pmu_name

  lspci | grep -i audio

  Add Audio Device:

  Audio Device: intel-had

  Backend Driver: spice

  You will see the ne settings in red. As above, this means you will have to reboot the Linux Mint VM to activate the audio capability.

  Note that although you have enabled audio, the default setup will launch the SPICE Remote viewer with audio muted. To hear sound, just unmute the audio by clicking on the Sound icon in the taskbar and raising the volume level.

- ♦ Reboot the VM

  When you Reboot with Shutdown -> Reboot, , you will get an error message saying powerdown failed

  Select Shutdown -> Stop

  Select Shutdown -> Start

- ♦ Reboot the Host (which will Reboot the VM again)

- Connect to the Linux Mint VM with the SPICE client (Remote Viewer) on the PC

  - ◊ There is one disadvantage to using SPICE as your connection protocol; each download file is a one-time-use ticket so you will need to get a fresh configuration from the Proxmox GUI for each connection. Given the speed advantages and audio capabiolity of SPICE, if you intend to use the Linux Mint desktop for any length of time during a connection, SPICE could be worth the effort of fetching a fresh configuration file for each connection

  - ◊ The SPICE client creates a window that is resizable with quick refresh and GUI interactions and has audio capabilities.

  - ◊ Note: If you request fullscreen, to stop using full screen, hover your mouse ove the middle of the top of the screen and a pop-down window will spear, allowing you to cancel full screen view.

  - ◊ In the Proxmox GUI window, expand the Datacenter and the Host listings in the left column and select the Linux Mint VM

    - ♦ The first time you initiate the Remote Viewer connection, it may take a few seconds for the screen to properly show while the system sets up for the connection. Once connected, Spice provides a fast GUI.

    - ♦ Use the Console dropdown menu and select SPICE

      using the Firefox browser

Select: Open with … remote-viewer.exe

The remote viewer will launch without any file being saved to your PC

using the Chrome browser

Save the download.vv file to your PC

Double-click the download.vv file.

The remote viewer will launch and the download.vv file will be deleted.

f) Install Backup Scripts

- See instructions and script in section below
- See paramters for Linux Mint in Appendix 1

9) Install MariaDB (MySQL binary compatible)

a) These directions show how to install MariaDb for Suite CRM; adapt for Linux Mint. We use MariaDB in Linux Mint to support the accounting package GnuCash.

◊ apt install mariadb-client mariadb-server

b) Initialize the MariaDB with the wizard to secure the database

◊ mysql_secure_installation

♦ Existing root Password: <Enter>

♦ Change root Password: y

{db-root-Password}

♦ Remove Anaonymous Users: Y

♦ Disallow root login remotely?: y

♦ Remove test database and access to it? : y

♦ Reload privilege tables now? : y

c) Setup the database for use with SuiteCRM

- Login as database root user and create another user to use with database
  - ◊ mysql --user=root --password={db-root-Password}
  - ◊ or
  - ◊ mysql -u root -p '{db-root-Password}'
- Create non-root user for use with SuiteCRM
  - ◊ CREATE USER '{db_SuiteCRM_user_name}'@'localhost' IDENTIFIED BY '{db_SuiteCRM_user_password}';
- Create non-root user for general, local, dbase admin on any dbase
  - ◊ CREATE USER '{db_user_name}'@'localhost' IDENTIFIED BY '{db_user_password}';
- Create non-root user for remote, dbase admin on any dbase from outside the server (see % instead of localhost)
  - ◊ (CREATE USER '{db_user_name_2}'@'%' IDENTIFIED BY '{db_user_password_2}'

- [If using remote access with an IDE (Netbeans) you may need to download and install on the client computer the MySQL connector from https://dev.mysql.com/downloads/connector/j]

- Create dbase for SuiteCRM
  - CREATE DATABASE {SuiteCRM_dbase};
    - This is the database name for the crm database
    - We have to create it now to give the admininstrator user_name access to it
    - We will then delete the database so SuiteCRM can create it later
- Provide new users with desired access rights
  - GRANT ALL PRIVILEGES ON {SuiteCRM_dbase}.* TO '{db_SuiteCRM_user_name}'@'localhost' identified by '{db_SuiteCRM_user_password}';
  - GRANT ALL PRIVILEGES ON *.* TO '{db_user_name}'@'localhost' identified by '{db_user_password}' WITH GRANT OPTION;
  - GRANT ALL PRIVILEGES ON *.* TO '{db_user_name_2}'@'%' identified by '{db_user_password_2}' WITH GRANT OPTION;
- Delete the SuiteCRM database (so SuiteCRM can create it from scratch)
  - DROP DATABASE {SuiteCRM_dbase};
    - This deletes the database so we can have a fresh install with the SuiteCRM installer
- Check to see all users and the permissions for a user
    - SELECT User,Host FROM mysql.user;
    - SHOW GRANTS FOR '{a_dbase_user}'@'localhost';
- Check to see all databases
    - SHOW databases;
- Flush all privileges (updates the settings currently in use to what you just created)
    - FLUSH PRIVILEGES;
- Exit the MySQL prompt and return to the Linux CLI
  - quit
- If you want to access the MariaDB from a remote host (like to use a remote Workbench-like application)
  - nano /etc/mysql/mariadb.conf.d/50-server.cnf
    - either comment out the line bind-address = 127.0.0.1

      # bind-address = 127.0.0.1
    - or change it to enable all IP addresses

      bind-address = 0.0.0.0
    - Restart the MariaDB server or just reboot the whole system

      systemctl restart mariadb.service (must be superuser to use this)

- or

  shutdown -r now

- This does expose the database to external hackers, so unless you really need remote access ongoing, it is recommended that you only enable this feature when needed and then disable it when finished
- Make sure the firewall is configured to allow the IP address through for port 3306

10) Backup Script

a) The parameters required to customize this script for your use, as well as the instructions on how to use this script are explained in the comments section at the start of the script.

- Establish a remote ftp site you can access
- Create a secure key-pair for authentication to the ftp site
- Copy this script to a protected directory (eg /root) on your server (passwords are included in the script)
  - ◊ Set tight permissions with chown root:root and chmod 700
  - ◊ I usually run dos2unix on any file transferred from a PC to a Linux server
- Edit the parameters to suit your installation and to backup the databases and directories you want backed up
  - ◊ Note that his script assumes a MySQL (or MariaDb) as the data base and if you enable the database backup, it uses MySQL commands. Since Wazo uses PostgreSQL, do NOT enable the dabatase backup for Wazo, use the already backed-up and compressed version of the database created by Wazo very day.
- Create the CRON job to run this script as root each night
- Put this script into /root

```
#!/bin/bash
#
# This script is open source, free for use by anyone, subject to GPLv2 licensing
# All caveats regarding use of open-source programs apply;
# I did my best but it is your responsibility to understand the script and check its suitability for you
# Authored by Richard Cantin, Ayuda, www.ayuda.ca
#
# This script assumes it will be used on a linux host with a bash engine, a MTA (Mail Transfer Agent) and sftp installed
#
# This script creates backups of selected files, directories and databases (depending on settings below)
# and puts the backups into a local directory, specified in the configuration below.
# The script creates the $Local_Directory specified below if it does not already exist
```

```
#
# After creating the backup files in the local directory, depending on the configuration settings specified below,
# the script will then:
#  - Do nothing more (Send_to_FTP = 0   and   Copy_To_Archive_Disk = 0)
#  - Send the backup files to a remote FTP site via sftp  (Send_to_FTP = 1)
#  - Copy the backup files to an archive disk            (Copy_To_Archive_Disk = 1)
#
# If the configuration settings below specify it, the script uses tar
# to compress the contents of the backups into a tarball before storing them in the local directory
#
# Because this script uses sftp, not ftp,
# and to enable the script to run unattended and with no interaction,
# this script depends on the user first establishing a key-pair
# with no passphrase for the private key (meaning it must be securely stored on the host system)
# for ssh authentication between the host system and the remote system
# so the sftp program, which calls the ssh protocol to create a secure connection,
# can connect without providing a password or passphrase.
#
# To setup the key-pair authentication (once),
# this script requires that before this script is run, the user (as user, NOT root) must first
# create a private/public key-pair on the local machine and put the matching public key on the remote server
# To do this:
#       - create the key pair using ssh-keygen and put the key pair into /home/{user}/.ssh
#             make sure to check/change permissions so
#                   the public key is readable by anyone (644)
#                   the private key is ONLY readable by user (600)
#       - put the public key into the authorized-keys file on the remote server by either (one or other, NOT both)
#             - using the open-ssh ssh command ssh-copy-id and authenticating (the first time only) with the user/password credentials
#                   to automtically handle all the steps required
#                   to copy the contents of the public key on the host system to the authorized_keys file on the remote server
```

```
#       - manually establishing an ssh connection to the remote server using user/password
#                 and manually doing all the steps required (making sure to set the permissions correctly)
#                 to copy the contents of the public key on the host system to the authorized_keys file on the remote server
#   - do the initial login manually using the key-pair for authentication so your host accepts the connection)
#           ssh ${remote_host_user_name}@${remote_host}      (no password needed now if the key-pair are properly installed and configured)
#   - then configure and run this script
#
# The script creates its backup every night and stores
# - a rolling week's worth of backups (Monday through Sunday each day has its own backup)
#           with each day of the week kept until it is replaced by next week's equivalent daily backup
# - a rolling monthly backup
#           with each month kept until it is replaced by next year's equivalent monthly backup
# so you end up with up to 19 backup instances (7 days and 12 months)
# to choose from if you want to restore your system
#
# If you want the same setup I have, the script will work as is, without any editing, other than:
# - declaring the email parameters to allow error messages to go to the administrator
#   - MailFrom
#   - MailTo
# - declaring the remote host parameters
#   - remote_host
#   - remote_host_user_name
# - declaring the full path (including file name) to the private key in the key pair used for ssh authentication
# - declaring if a MySQL database is to backed up (or not)
#   - mysql_flag
#   (if yes to the MySQL database being backed up : mysql_flag=1)
#   - mysql_dbase_name
#   - mysql_user
#   - mysql_password
```

```
#   - mysql_dbase_backup_file_name
# - declaring the name of the local server (text field for your use only)
# - declaring the Local_Directory into which the files are locally stored before being sent to the remote backup site
# - declaring the directories/files you want backed up
# - declaring if you want the backup files sent to a remote FTP site
# - declaring the directory path where you want the files stored on the remote site (if specified above)
# - declaring if you want the backup files copied to an archice disk
# - declaring the directory path where you want the files stored on the archive disk (if specified above)
#
# When declaring directories, include the full pathname to the directory, but do not include the trailing /
#
# To install and configure this script (assuming you want to stay with my settings (below)
# Put this file, named backup_to_ftp_and_archive_disk.sh, with executable permission,
# into your home directory (outside the web root directory)
# Add to the crontab the schedule for this script
# # Run the script to copy key files and directories to remote storage site using sftp
# 7 3 * * * root /<path_to_script>/backup_to_ftp_and_archive_disk.sh 1>/dev/null 2>&1
# (with these settings, the script runs every night at 3:07AM)
# Leave the server running overnight to backup selected directories and files each evening to a remote site
#
# If there is an error in either the tar process or the FTP file transfer the script sends an email to the root user of the server
# So if you have root aliased to an admin account or if you want to change the MailTo below
# you will get notified of any errors.
#
########################################################


# Parameters to be set for each installation :
#
# The email address from which you wish the notice to come
# Formatted as "Real Name<email@domain.tld>"
```

```
# This specification is required because the script is being called from crontab
# and crontab establishes its own environment, including its mail source
# so you must specify it here to ensure the proper From email address for the MTA
MailFrom="LM20 Backup<pbx.mail@ayuda.ca>"
#
# The email address of the system administrator who will be sent any error messages
# that are generated on failure to transfer files each night
# If your system does not have an MTA (Mail Transfer Agent) installed
# the easiest solution is to install msmtp and then mailutils to allow outbound emails only
# (in that order, NOT together in one line or the full Postfix syste will be installed)
# and then configure per the instructions with ssmtp
MailTo="administrator_email@domain.com"
#
# Declare if you want the backup files sent to a remote FTP site
#    Send_to_FTP = 0 = No
#    Send_to_FTP = 1 = yes
# If you set Send_to_FTP to 0, you do not have to specify remote_host or
remote_host_user_name
Send_to_FTP=1
# Your ssh login credentials to enable sftp (Secure File Transfer Protocol) from this local
machine to your remote site
# (Note: remote_host is the full url or IP address to your remote site)
# The daily backups will overwrite themselves each week
# The monthly backups will overwrite themselves each year
remote_host="IP_Address_of_ftp_site"
remote_host_user_name="ftp_user_name"
#
# Your local user private key location to allow key-pair authentication
# Enter the location of the private key stored on the local machine
# that matches the public key uploaded to the remote machine
# Enter full path name including filename
ssh_private_key_location="/some/place/on/your/server"
#
# The path to the directory on the remote storage site where the backup files will be
stored
```

```
# Use an absolute path (start with /) to define the Remote_Directory, but do NOT include a trailing slash
Remote_Directory="/a/path/to/the/directory/at/the/ftp/site"
#
# Declare if you want the backup files copied to an archive disk
Copy_to_Archive_Disk=1
# The path to the directory on the archive disk into which you want to store the backups
# The archive disk must be mounted and accessible to the script / shell for the user
# The daily backups will overwrite themselves each week
# The monthly backups will overwrite themselves each year
Archive_Disk_Directory="/path/to/archive/disk/directory"
#
# The MySQL database parameters
#
# If a MySQL database is to be backed up (one or more),
#    set mysql_flag to 1 and set other MySQL parameters
#  else
#    set mysql_flag to 0 and leave all other MySQL parameters as empty (ie set = "")
mysql_flag=1
# The MySQL User specified must have read access to the database it is being used to access
# If you only want one database backed up, then just enter one array set:
#       mysql_dbase_name[1]="dbase_1_name"
#   mysql_user[1]="user_1"
#   mysql_password[1]="password_1"
# If you want multiple databases backed up, enter more array sets:
#       mysql_dbase_name[1]="dbase_1_name"
#   mysql_user[1]="user_1"
#   mysql_password[1]="password_1"
#       mysql_dbase_name[2]="dbase_2_name"
#   mysql_user[2]="user_2"
#   mysql_password[2]="password_2"
#       ...
#  (and yes, I know I am not using the first entry in the array [0]; I do this for ease of understanding and coding)
mysql_dbase_name[1]="dbase1"
```

mysql_user[1]="dbase_user_name"

mysql_password[1]="dbase_user_password"

#

The Server Name - text entry to describe which server is being backed up - do NOT include any spaces

(Useful for admins who look after more than one server)

Server_Name="Server-Name-for-identification-of-files"

#

A flag to determine if the files are to be comprressed before beintg transferred

Some systems already create compressed backup files so you do not want to compress them again

If you are NOT compressing files in this script, set File_Suffix to blank ("")

If you are compressing files, include the complete suffic designator, including the leading period (.tar.gz)

Compress_Files=1

File_Suffix=".tar.gz"

#

The name of the local directory into which the database file (if specified) and the data directories and files are stored

after the script labels them with the date and server identification

Use an absolute path (start with /) for the Local_DirectoryLocal_Directory="/home/user/place"

#

The list of Directories to be backed up

You can add to this list of Directories for each additional Directory (or individual file) you want to back up

- add a new Array entry with a number next in sequence (sequence starts at 1)

- You do NOT need to end the Directory name with a /

correct sample: Source_Directory[1]="/var/www/html"

incorrect sample: Source_Directory[1]="/var/www/html/"

#

If the mysql_flag is set to 1, the first directory, Source_Directory[1],

must be the directory that will contain the MySQL_dump backup files generated by this script

If the directory does not already exist, the script will create it.

#

The remaining directories can be any directory you specify

```
# The remaining "Directories" can be either directories or individual files
# If you are not compressing the files (see Compress_Files flag above)
#    the "Directories" MUST be individual files
# Use an absolute path (start with /) to define each Source_Directory
#
# Use Source_Directory[1] as the first directory to be backed up
# Use Source_Directory[2] as the second directory to be backed up
# and continue until you have specified all directories/files to be backed up
# Source_Directory[2]="{directory_2_to_be_backed_up-see_notes_above}"
# Source_Directory[3]="{directory_3_to_be_backed_up-see_notes_above}"
Source_Directory[1]="/a/path/to/the/database/store/directory"
Source_Directory[2]="/a/path/to/a/local/directory"
Source_Directory[3]="/make/sure/the/paths/are/absolute"
Source_Directory[4]="/and/add/as/man/as/you/want"

#########################################################

#
# This defines variables and parameters used in the script
# You should NOT be making changes to these parameters unless you plan to re-write the script
#

date_for_monthly_backup="03"

this_date=$(date +"%d")

this_month=$(date +"%m-%b")
this_weekday=$(date +"%u-%a")

# Sanitize the Server name to replace spaces with hyphens (-)
Server_Name=${Server_Name// /\-}
File_Prefix=${Server_Name}

# Set the path to the mail server
```

```
Mailer="/usr/bin/mail"

# Set the path to the sftp client
sftpProgram="/usr/bin/sftp"

#########################################################
#
# Do NOT edit anything below here - this is the core script
#

# Sanitize the Local_Directory path name to remove spaces and ensure no trailing /
# Strip any spaces
Local_Directory=${Local_Directory// /}
# Strip a trailing / from the path if there is one
Local_Directory=${Local_Directory%/}

# Sanitize the FTP Remote_Directory path name given to remove spaces and ensure no
trailing /
# Strip any spaces
Remote_Directory=${Remote_Directory// /}
# Strip a trailing / from the path if there is one
Remote_Directory=${Remote_Directory%/}

# Sanitize the Archive_Disk_Directory name to remove spaces and ensure no trailing /
# Strip any spaces
Archive_Disk_Directory=${Archive_Disk_Directory// /}
# Strip a trailing / from the path if there is one
Archive_Disk_Directory=${Archive_Disk_Directory%/}



# Sanitize the Source_Directory path names to remove spaces and ensure no trailing /
# Create the names (Daily and Monthly) for each Directory backup file
Num_Backup_Directories=${#Source_Directory[@]}
Dir_Count=1
while [ "${Dir_Count}" -le "${Num_Backup_Directories}" ] ; do
```

```
                # Sanitize the Source_Directory name to remove spaces and ensure no trailing /
                Dir_Name=${Source_Directory[${Dir_Count}]}
                # Strip any spaces
                Dir_Name=${Dir_Name// /}
                # Strip a trailing / from the path if there is one
                Dir_Name=${Dir_Name%/}
                Source_Directory[${Dir_Count}]=${Dir_Name}
                # Create a Name from the Source_Directory path to use in the final backup file name
                # Any / must be eliminated in the name or they wil be interpreted as a directory change
                # Delete the Leading /
                Path_as_Name=${Dir_Name#/}
                # Replace the remaining / with -
                Path_as_Name=${Path_as_Name//\/\-}
        # Create File Names for Daily and Monthly backup files

Backup_Daily_to_Filename[${Dir_Count}]=${this_weekday}"_"${File_Prefix}"_"${Path_as_Name}${File_Suffix}

Backup_Monthly_to_Filename[${Dir_Count}]="m_"${this_month}"_"${File_Prefix}"_"${Path_as_Name}${File_Suffix}
        let Dir_Count=Dir_Count+1
done

# Check if the mysql_flag is set (there is a MySQL database to be backed up) and if so, initialize the MySQL parameters
if [ "${mysql_flag}" -eq "1" ]
        then
                # Check to see if the Directory specified in "Source_Directory[1]" already exists
                # If not, create it
                if [ ! -d "${Source_Directory[1]}" ]
                        then
                                mkdir "${Source_Directory[1]}"
                                echo "New Directory created: "${Source_Directory[1]}
                fi
```

```
        # Generate the (uncompressed) backup of the site MySQL database(s)
        # and put it/them into the Source_Directory[1] directory identified above for backup
        #  Note: to restore the (uncompressed) database, use the CLI command:
        #  mysql --user=${mysql_user} --password=${mysql_password} ${mysql_dbase_name} < ${Source_Directory[1]}/${mysql_dbase_name[${Database_Count}]}.sql
        Num_Databases=${#mysql_dbase_name[@]}
        Database_Count=1
        while [ "${Database_Count}" -le "${Num_Databases}" ] ; do
            mysqldump --user=${mysql_user[${Database_Count}]} --password=${mysql_password[${Database_Count}]} ${mysql_dbase_name[${Database_Count}]} > ${Source_Directory[1]}/${mysql_dbase_name[${Database_Count}]}.sql
            CalledPID=$!          # Get the Process ID (PID) of the process just initiated above
            wait ${CalledPID}     # Wait for the identified Process to finish before proceeding
            let Database_Count=Database_Count+1
        done
fi

# Check to see if the Directory specified in "Local_Directory" already exists
# If not, create it
# Change to the Local_Directory
if [ ! -d "${Local_Directory}" ]
    then
        mkdir "${Local_Directory}"
        echo "New Directory created: "${Local_Directory}
fi

cd ${Local_Directory}

# Check to see if this is the date to do the Monthly backup as well as the Daily backup
# If so, set the counter to enable two backups per File - One for Daily and one for Monthly
if [ "${this_date}" -eq "${date_for_monthly_backup}" ]
    then
```

```
            # It is the date to do the Monthly backup as well as the Daily Backup
            Num_Backups_per_File=2
        else
            # It is NOT the date to do the Monthly backup as well as the Daily Backup
            Num_Backups_per_File=1
fi

# Create Daily (and if the right date, Monthly) backup files
Compression_Error_Flag=0
Dir_Count=1
while [ "${Dir_Count}" -le "${Num_Backup_Directories}" ] ; do
    #
    # Create Daily (and if the right date, Monthly) backup tar.gz files (with error check if tar does not work)
    # Note that the tar command removes the leading / to prevent untarring of files to an absolute directory
    # but to prevent a warning message on each run, we disable that safety precauion with --absolute-names.
    # That would normally leave us at great risk of un-tarring with absolute path and replacing something
    # we did not want replaced.
    # So we use the --directory= option and the parameter parsing functions ${x##*/} and ${x%/*}
    # to explicitly store the specific folder/directory we want, NOT the entire path to the folder/directory
    Full_Path_Defn=${Source_Directory[${Dir_Count}]}
    Path_to_folder=${Full_Path_Defn%/*}
    Folder_Name=${Full_Path_Defn##*/}
        Error_State[${Dir_Count}]=0
        # If the Compress_Files flag is not 1, use the files as they are (likely already compressed)
        if [ "${Compress_Files}" -eq "1" ]
            then
                tar -cvzf ${Backup_Daily_to_Filename[${Dir_Count}]} --absolute-names --directory=${Path_to_folder} ${Folder_Name} 2>error-message-daily.txt
                Error_State[${Dir_Count}]=$?
                wait
                if [ "${Error_State[${Dir_Count}]}" -eq "0" ]
```

```
                then
                        Comp_Result="The compressed file was created for the Daily backup of Directory: ${Source_Directory[${Dir_Count}]} on Server: ${Server_Name}"
                        echo ${Comp_Result}
                else
                        Compression_Error_Flag=1
                        Comp_Result="Error! - the compressed file was NOT properly created for the Daily backup of Directory: ${Source_Directory[${Dir_Count}]} on Server: ${Server_Name}"
                        echo ${Comp_Result}
                        if [ "${Num_Backups_per_File}" -eq "1" ]
                                then
                                        Message="$(cat error-message-daily.txt)"
                                else
                                        Message="The Monthly backup file of of Directory:"${Source_Directory[${Dir_Count}]}" on Server: ""${Server_Name} failed because the Daily backup was never created.  Fix the Daily backup."
                        fi
                fi
        else
                cp ${Path_to_folder}/${Folder_Name} ${Local_Directory}/${Backup_Daily_to_Filename[${Dir_Count}]}
        fi


        if [ "${Num_Backups_per_File}" -eq "2" ]
                then
                        cp ${Backup_Daily_to_Filename[${Dir_Count}]} ${Backup_Monthly_to_Filename[${Dir_Count}]}
                        wait
        fi
    let Dir_Count=Dir_Count+1
done


# If Send_to-FTP is set to 1, copy the backup files to the designated remote FTP site
#       1) Create the set of file transfer commands and put them into a batch file
```

```
#       2) Transfer the files via sftp using batch transfer
#       3) Check if any file transfer created an error and if so, notify the administrator that a backup failed
# Using the above sequence, we only make one ssh connection for all files
# instead of one ssh connection per file
# For multiple file transfers this is much more efficient
# and it also is less likely to raise a security flag at the remote site
# that could occur with multiple, sequential ssh connections from the same client
#
# 1) Create the set of file transfer commands and put them into a batch file
#
if [ "${Send_to_FTP}" -eq "1" ]
      then
            Dir_Count=1
            Num_Files_to_Transfer=0
            while [ "${Dir_Count}" -le "${Num_Backup_Directories}" ] ; do
                  Backup_Count=1
                  while [ "${Backup_Count}" -le "${Num_Backups_per_File}" ] ; do
                        if [ "${Error_State[${Dir_Count}]}" -eq "0" ]
                              then
                                    # Create the sftp command to send the Daily/Monthly backup tar.gz file to the designated remote site
                                    if [ "${Backup_Count}" -gt "1" ]
                                          then
                                                File_to_Transfer=${Backup_Monthly_to_Filename[${Dir_Count}]}
                                          else
                                                File_to_Transfer=${Backup_Daily_to_Filename[${Dir_Count}]}
                                    fi
                                    if [[ "${Dir_Count}" -eq "1" && "${Backup_Count}" -eq "1" ]]
                                          then
                                                echo "put ${Local_Directory}/${File_to_Transfer} ${Remote_Directory}/${File_to_Transfer}" > ${Local_Directory}/sftp_batchfile.txt
                                          else
```

```
                                        echo "put ${Local_Directory}/${File_to_Transfer} ${Remote_Directory}/${File_to_Transfer}" >> ${Local_Directory}/sftp_batchfile.txt
                                    fi
                                    let Num_Files_to_Transfer=Num_Files_to_Transfer+1
                                fi
                                let Backup_Count=Backup_Count+1
                            done
                            let Dir_Count=Dir_Count+1
                        done
                        #
                        # 2) Transfer the files via sftp using batch transfer
                        #
                        # The options shown for the sftp program are ( -o means use a native SSH option):
                        #       -o IdentitiesOnly=yes                   Tell the system that it is to use the private key location given for authentication
                        #                                               (without this, many systems use the default environment and ignore the provided key location)
                        #       -o IdentityFile=${ssh_private_key_location}   Tell the system where the Private key is for this SSH connection
                        # If you were running this script from the command line, the above are likely unneccesary,
                        # but since this script will be called as a CRON job
                        # and since CRON jobs will, if not told otherwise, use their own environment setup
                        # which will prevent the proper key -pair from being used to authenticate
                        # without the above, the script, when called by CRON, would fail
                        Transfer_Error_Flag=9999
                        if [ "${Num_Files_to_Transfer}" -gt "0" ]
                            then
                                ${sftpProgram} -o IdentitiesOnly=yes -o IdentityFile=${ssh_private_key_location} -b ${Local_Directory}/sftp_batchfile.txt ${remote_host_user_name}@${remote_host}
                                Transfer_Error_Flag=$?
                        fi
                        #
```

```
        # 3) Check if any file transfer created an error and if so, notify the administrator that a backup failed
        #
        if [ "${Compression_Error_Flag}" -ne "0" ]
                then
                        Message="Error - Transfer of a file from server ${Server_Name} to host ${remote_host} failed because a file compression failed!"
                else
                        if [ "${Transfer_Error_Flag}" -ne "0" ]
                                then
                                        Message="Error - Transfer of a file from server ${Server_Name} to host ${remote_host} failed with error code ${Transfer_Error_Flag[${Dir_Count}]}!"
                                else
                                        Message="Transfer of files from server ${Server_Name} to host ${remote_host} was successful"
                        fi
        fi
        echo ${Message}
        if [ "${Message:0:5}" = "Error" ]
                then
                        Subject="Error! - Backup Failed for ${Server_Name}"
                        echo "${Message}" | ${Mailer} -s "${Subject}" ${MailTo} --append=From:"${MailFrom}"
                 else
                        Subject="Backup succeeded for ${Server_Name}"
                        echo "${Message}" | ${Mailer} -s "${Subject}" ${MailTo} --append=From:"${MailFrom}"
        fi
fi


# If Copy_to_Archive_Disk is 1 then copy the backup files to the designated archive disk directory
#
if [ "${Copy_to_Archive_Disk}" -eq "1" ]
then
    Dir_Count=1
    Copy_Error_Code=0
```

```
	Error_Flag=0
	while [ "${Dir_Count}" -le "${Num_Backup_Directories}" ] ; do
		Backup_Count=1
		while [ "${Backup_Count}" -le "${Num_Backups_per_File}" ] ; do
			# Copy the Daily/Monthly backup file to the designated archive disk
directory
			if [ "${Backup_Count}" -gt "1" ]
				then
					cp -a
${Local_Directory}/${Backup_Monthly_to_Filename[${Dir_Count}]}
${Archive_Disk_Directory}/
					Copy_Error_Code=$?
				else
					cp -a
${Local_Directory}/${Backup_Daily_to_Filename[${Dir_Count}]}
${Archive_Disk_Directory}/
					Copy_Error_Code=$?
			fi
			if [ "${Copy_Error_Code}" -ne "0" ]
			then
				Error_Flag=1
			fi
			let Backup_Count=Backup_Count+1
		done
		let Dir_Count=Dir_Count+1
	done
	if [ "${Error_Flag}" = "0" ]
		then
			echo "Copy of files to archive disk was completed"
		else
			echo "Copy of files to archive disk had issues; check your archive disk"
	fi
fi

# If get to here, exit with no error flags
exit 0
```

b) For each virtual machine, you can backup the entire instance using the Proxmox user interface. I created a second (Development) server with the same Proxmox setup as the primary (Production) server: 3 VMS for: WazoPBX, SuiteCRM and Linux Mint. I then copied the backup Proxmox file for Wazo from the Production server to the Development server and "restored" the Development server for Wazo to the version from the Production server . This gave me a cold-standby of Wazo, ready to be put into use if the Production server ever crashed. Not as good as a "cluster" setup but for my scenario, more than adequate and a lot simpler to setup and manage. After this, be careful to ensure the Development VM of Wazo is NOT set to boot on startup since the resore creates an exact duplicate of the Production server on the Development server, with the same IP address and Hostname, so only one can be functioning at a time or the router and network will have issues.

c) Note that if you use backup and restore with Wazo, the system has issues with "ACL not found" and will post error messages to the syslog file multiple times per second. If you do a backup and restore with Wazo, then:

- Get the correct token from /var/lib/consul/master_token

  cat /var/lib/consul/master_token

- Compare that with the token shown in the json file /etc/consul.d/wazo-config.json as the variable "acl_master_token"

  cat /etc/consul.d/wazo-config.json | grep acl_master_token

- If they are different,
  - ◊ replace the value assigned to the variable acl_master_token in /etc/consul.d/wazo-config.json with the value in /var/lib/consul/master_token
  - ◊ restart the consul to have it take the correct master_token

    systemctl restart consul

- If they are the same, try resetting consul entirely

  systemctl stop consul

  rm -rf /var/lib/consul/raft/

  rm -rf /var/lib/consul/serf/

  rm -rf /var/lib/consul/services/

  rm -rf /var/lib/consul/tmp/

  rm -rf /var/lib/consul/checks/

  systemctl start consul

11) Configuring iptables - Linux Firewall

a) Make sure the system is NOT exposed to the internet while programming IPtables. While IPtables is being configured, there is no firewall protection on the server, so any Ports forwarded to the server have wide open access. Only forward Ports after IPtables has been configured.

b) Sequence is very important for IPtables, so make sure to enter the rules in the Chains and the actions in the INPUT table as specified below.

c) For this installation I took a different approach to securing the system. I had previousy used a hybrid approach of identifying as many bad guys as possible (DShield, ProjectHoneyPot, Personal Lists, ...) and using a blacklist to prevent their access AND creating a whitelist of special IPs which were

allowed access to the system. With this install, I am taking a fully whitlelist approach: Locking out everyone unless I have included them in a list of allowed IPs (the whitelist). This does require a little more setup for remote phones, people allowed web access or administrators, (especially when they are on a dynamic IP address) but it is not that difficult to do and well worth the improved security and peace of mind.

d) To check if your setup is blocking traffic you want, occasionally look a the log contents in /etc/var/syslog and look for "Denied by iptables". We used that prefix on traffic that was not allowed through so you can search for recent occurances with

```
tail -500 /var/log/syslog | grep "Denied by iptables"
```

- See what traffic is being refused (check port numbers, source addresses, …) and decide if you want to allow it or maybe add it to the "Nuisance_Traffic" chain (see below) to continue blocking it without logging it.

e) Although I like to manually configure iptables using the script shown below, on systems with Webmin Installed (the CRM server) you can use the iptables webmin module (webmin -> Network -> Linux Firewall) to configure IPtables. The steps required to configure IPtables from within Webmin are:

- Backup the existing IPtables file in /etc/sysconfig/iptables to /etc/sysconfig/iptables.orig
- Flush the existing settings from IPtables
- Create custom chains (subroutines in any other langauge)
- Populate the INPUT table, including calling custom chains, in the sequence desired
- Apply the settings
- Refresh the browser view so you can see the new additions to the IPtables configuration by the Dynamic IP script.
- Backup the new iptables configuration file

f) (Optional - only needed if you want to give access to the server from outside the LAN and the place from which the outside user will be coming has a dynamic IP address) Enable the Dynamic IP scripts (see below) which enables specified Dynamic IPs access through the firewall and keeps the IP Address of the Dynamic IP current. In its native state, iptables only converts DNS entries to IP numbers once when loaded, so I could have set a cron job to reload IPtables on a regular basis, but that leaves the system temporarliy exposed (while IPtables reloads) and takes unnecessary processing power while the full configuraton reloads. So instead, I added a script to check the Dynamic IP values and, when changed, to update that entry in the IPtables.

g) iptables is one module of the netfilter suite. iptables is structured hierarchically, with tables holding chains holding rules, with (at its default setup)

- I will show a script, at the end of this section, that does what this section describes. I included in this section a detailed explanation for how iptables works so you (or I at a future date after I have forgotten the intricacies of iptables) can edit to suit your needs.
- Tables
  - ◊ FILTER

    For a system that is an ordinary router and not doing any masquerading, or a system that only needs a firewall to protect itself, this is the only table that rules need to be added to.

    - ♦ Chains

INPUT

OUTPUT

FORWARD

◊ MANGLE

This table is used only for specialized packet alteration. This table is rarely used at all in a typically firewall configuration.

♦ Chains

PREROUTING

OUTPUT

FORWARD

INPUT

POSTROUTING

◊ NAT

Rules are typically added to this table to set up masquerading, transparent proxying or some other kind of address translation. In our setup, the router handles NAT so we do not worry about it in our iptables setup.

♦ Chains

PREROUTING

POSTROUTING

OUTPUT

◊ RAW

Iptable's Raw table is for configuration exceptions and is not normally required in a standard firewall setup.

♦ Chains

PREROUTING

OUTPUT

◊ SECURITY

This table is used for Mandatory Access Control (MAC) networking rules. Not normally used in a standard firewall setup.

♦ Chains

INPUT

OUTPUT

FORWARD

- There are Tables, Chains and Rules for ipV4 and ipV6
- Sequence is important in iptables. Whatever criteria is first matched for a packet is used. If the criteria is met and the sequence is stopped via a jump (see below) to another target (ACCEPT, DROP, QUEUE or RETURN) then later rules will never be considered. When adding rules (--

append or -A option), rules are always appended to the end of the rules list, so plan ahead and add rules in the order you want them to be in the sequence. If you do not want rules added to the end of the Chain, use the --insert or -i option and you can add the rule at the beginning or anywhere specified in the rule set for that Chain.

- The last rule of a table is normally a DROP (all) rule, assuming the intent was that if the packet does not match any recognized traffic, it should not be allowed in to the server. Unfortunately, if you have a final rule as DROP, when you --append a new rule, it goes to the end of the rule set for that Chain and ends up after the DROP, so it is never accessed. POLICY is another way of ensuring that the default for a rule set if no other rule matches is to DROP the packet. By setting the POLICY for the table to DROP as opposed to the default POLICY of ACCEPT. The POLICY defines what happens if the packet gets all the way through the rule set without matching any criteria. Be CAREFUL … If you set the Policy for the FILTER table to DROP and then flush iptables, the POLICY remains, meaning ALL traffic will now be DROPped and you will only be able to access your system with a physically connected terminal (or for Proxmox, via the Console). Make sure, if you are flushing the iptables, and you have set the Policy on the FILTER table to DROP, that you set the POLICY on the FILTER table back to ACCEPT before flushing the iptables. Note: POLICY can only be set for the system Chains like INPUT, OUTPUT, FORWARD, … and not for User-Defined Chains, so for User-defined Chains, you want to make sure they always end with a RETURN.
  - ◊ iptables --table filter --policy INPUT DROP
    - ♦ iptables --table filter --policy INPUT ACCEPT

      to "undo" the POLICY BEFORE flushing the iptables
  - ◊ iptables --table filter --policy OUTPUT ACCEPT
  - ◊ iptables --table filter --policy FORWARD ACCEPT
  - ◊ ~~~~~
- Rules are put into the Chains for each table; if no rules are present, the POLICY will determine what happens to packets
  - ◊ Each Rule
    - ♦ Contains a criteria and a target.
    - ♦ If the criteria is matched, it goes to the rules specified in the target (or) executes the special values mentioned in the target.
    - ♦ If the criteria is not matched, it moves on to the next rule.
    - ♦ For clarity, I prefer to use the long form of the commands below when generating rules with iptables commands. When you run iptables-save, it strips the commands down to their short form and uses the short form in the saved version. This way, netfilter has the least amount of text to read and interpret for each command and can run the quickest.
    - ♦ Has the following possible parameters

      --append or -A (capital A) : Append a rule to the end of the specified Chain in the specified table (if no table specified, the FILTER table is used)

--insert or -I (capital I) : Insert a rule to the start of the specified Chain in the specified table (if no table specified, the FILTER table is used) OR, if an optional [rule_number] parameter is used after the Chain declaration, as in (iptables --table filter --insert INPUT 3 ~~~~ --jump ACCEPT) then the new rule is inserted at the location specified (in this case 3), bumping all subsequent rules down one.

--protocol or -p : Protocols. tcp, udp, icmp, etc.

--source or -s : Source ip-address of the packet

--destination or -d : Desitnation ip-address for the packet

--source-port or --sport : source port for the traffic

--source-ports or --sports if you are using the "--match multiport" option

--destination-port or --dport : destination port for the traffic

--destination-ports or --dports if you are using the "--match multiport" option

--in-interface or -i : The interface through which the incoming packets are coming through the INPUT, FORWARD, and PREROUTING chain

Can be the hardware interfact designation or lo (lower case L and letter O) to indicate loopback source

--out-interface or -o : The interface through which the outgoing packets are sent through the INPUT, FORWARD, and PREROUTING chain

--match or -m : Specifies a match to use, that is, an extension module that tests for a specific property

--match multiport

--match comment

…

--match comment --comment "followed by text in double quotes"

◊ For each Rule, the possible actions associated with the --jump (-j) parameters are

- ACCEPT – Firewall will accept the packet.
- DROP – Firewall will drop the packet.
- QUEUE – Firewall will pass the packet to the userspace.
- RETURN – Firewall will stop executing the next set of rules in the current chain for this packet. The control will be returned to the calling chain

- Check the current status of each table
  - ◊ add --verbose or -v for more detail on each list
  - ◊ add --numeric or -n to have port numbers and ip addresses display in numeric format instead of descriptive (text) format
  - ◊ add --line-numbers to have each line in the rule set show the line number of the rule. This can be handy when deleting a rule.
  - ◊ To see the rules in a formatted output
    - For ipV4

iptables --table filter --list --verbose --numeric --line-numbers

This shows the Chains and Rules associate with the FILTER table

If a TABLE is not specified with -t {table_name}, the default is to show the FILTER talbe, so "iptables --list" is the same as "iptables -t filter -- list"

iptables --table mangle --list --verbose --numeric --line-numbers

iptables --table nat --list --verbose --numeric --line-numbers

iptables --table raw --list --verbose --numeric --line-numbers

iptables --table security --list --verbose --numeric --line-numbers

- ♦ For ipV6

ip6tables --table filter --list --verbose --numeric --line-numbers

This shows the Chains and Rules associate with the FILTER table

If a TABLE is not specified with -t {table_name}, the default is to show the FILTER table, so "ip6tables --list" is the same as "ip6tables --table filter -- list"

ip6tables --table mangle --list --verbose --numeric --line-numbers

ip6tables --table nat --list --verbose --numeric --line-numbers

ip6tables --table raw --list --verbose --numeric --line-numbers

ip6tables --table security --list --verbose --numeric --line-numbers

- ◊ To see the rules as they are stored in the /etc/iptables/rules.v4 and rules.v6 files (--list-rules instead of --list)
  - ♦ For ipV4

iptables --table filter --list-rules

iptables --table mangle --list-rules

iptables --table nat --list-rules

iptables --table raw --list-rules

iptables --table security --list-rules

  - ♦ For ipV6

ip6tables --table filter --list-rules

ip6tables --table mangle --list-rules

ip6tables --table nat --list-rules

ip6tables --table raw --list-rules

ip6tables --table security --list-rules

- Before activating iptables, see what ports are being used and make sure to determine if you want to continue allowing them in. To list ports in use, you can
  - ◊ To see what ports are actively listening on the Wazo server, on another computer, run the nmap command pointing to the Wazo IP address as

nmap -sV {IP_Address_of_Wazo_Server}

◊ To get a list of all processes running, which will show all ports in use on the Wazo server, on the Wazo server run the command

lsof -i

- If there are any rules or chains already established, save them and understand why they are there so you can decide whether or not to incorporate them into your final version of iptables. Wazo has no rules from the initial installation.
- Clear all existing rules and user-defined chains so you can start fresh

```
iptables --table filter --policy INPUT ACCEPT
iptables --table filter --policy OUTPUT ACCEPT
iptables --table filter --policy FORWARD ACCEPT
iptables --table filter --flush
iptables --table filter --delete-chain
iptables --table mangle --policy PREROUTING ACCEPT
iptables --table mangle --policy OUTPUT ACCEPT
iptables --table mangle --policy FORWARD ACCEPT
iptables --table mangle --policy INPUT ACCEPT
iptables --table mangle --policy POSTROUTING ACCEPT
iptables --table mangle --flush
iptables --table mangle --delete-chain
iptables --table nat --policy PREROUTING ACCEPT
iptables --table nat --policy POSTROUTING ACCEPT
iptables --table nat --policy OUTPUT ACCEPT
iptables --table nat --flush
iptables --table nat --delete-chain
iptables --table raw --policy PREROUTING ACCEPT
iptables --table raw --policy OUTPUT ACCEPT
iptables --table raw --flush
iptables --table raw --delete-chain
iptables --table security --policy INPUT ACCEPT
iptables --table security --policy OUTPUT ACCEPT
iptables --table security --policy FORWARD ACCEPT
iptables --table security --flush
iptables --table security --delete-chain

ip6tables --table filter --policy INPUT ACCEPT
ip6tables --table filter --policy OUTPUT ACCEPT
```

ip6tables --table filter --policy FORWARD ACCEPT

ip6tables --table filter --flush

ip6tables --table filter --delete-chain

ip6tables --table mangle --policy PREROUTING ACCEPT

ip6tables --table mangle --policy OUTPUT ACCEPT

ip6tables --table mangle --policy FORWARD ACCEPT

ip6tables --table mangle --policy INPUT ACCEPT

ip6tables --table mangle --policy POSTROUTING ACCEPT

ip6tables --table mangle --flush

ip6tables --table mangle --delete-chain

ip6tables --table nat --policy PREROUTING ACCEPT

ip6tables --table nat --policy POSTROUTING ACCEPT

ip6tables --table nat --policy OUTPUT ACCEPT

ip6tables --table nat --flush

ip6tables --table nat --delete-chain

ip6tables --table raw --policy PREROUTING ACCEPT

ip6tables --table raw --policy OUTPUT ACCEPT

ip6tables --table raw --flush

ip6tables --table raw --delete-chain

ip6tables --table security --policy INPUT ACCEPT

ip6tables --table security --policy OUTPUT ACCEPT

ip6tables --table security --policy FORWARD ACCEPT

ip6tables --table security --flush

ip6tables --table security --delete-chain

- Add rules to Chains in the Table(s) (usually only the FILTER table)
  - ◊ You will add rules via the iptables and ip6tables commands (-A is equivalent to --append ; -I is equivalent to --insert))
    - ♦ iptables --table filter --append INPUT ~~~~~

      or iptables --table filter --insert INPUT 1 ~~~~~
    - ♦ ip6tables --table filter --append INPUT ~~~~~

      or ip6tables --table filter --insert INPUT 1 ~~~~~
  - ◊ and when the rules are stored (see next step with iptables-save and ip6tables-save) they will be put into files as (without the iptables or ip6tables command leading each line and using the short form of each command)
    - ♦ # Generated by ~~~~~
    - ♦ *filter

- :INPUT ~~~~~
- :FORWARD ~~~~~
- :OUTPUT ~~~~~
- -A INPUT ~~~~~
- ~~~~~
- COMMIT

- Once you have the rules the way you want them, save and restore them with (first create iptables directory in /etc)
  - ◊ Save

    iptables-save > /etc/iptables/rules.v4

    ip6tables-save > /etc/iptables/rules.v6
  - ◊ Restore (activate the rules from a saved set)

    iptables-restore < /etc/iptables/rules.v4

    ip6tables-restore < /etc/iptables/rules.v6
  - ◊ Test
    - Temporarily apply the saved iptables rules and if, by applying the rules, you are cutoff from the server, after a timeout period in which you did not reply to the "accept rules" request, iptables will revert to the rules in existence before the iptables-apply was used

      iptables-apply /etc/iptables/rules.v4

      ip6tables-apply /etc/iptables/rules.v6
  - ◊ Temporarily disabling iptables
    - If you need to temporarlity disable the firewall (filter table shown; if there are rules in the other talbes, duplicate for those tables using the CHAINS specific to those tables), you can flush all the rules - and make sure POLICY is ACCEPT - using

      iptables --table filter --policy INPUT ACCEPT

      iptables --table filter --policy OUTPUT ACCEPT

      iptables --table filter --policy FORWARD ACCEPT

      iptables --table filter --delete-chain

      This deletes any user-defined chains in the filter table

      iptables --table filter --flush

      ip6tables --table filter --policy INPUT ACCEPT

      ip6tables --table filter --policy OUTPUT ACCEPT

      ip6tables --table filter --policy FORWARD ACCEPT

      ip6tables --table filter --flush
  - ◊ To make sure the rules are reloaded every time the system reboots, install iptables-persistent

and it will automatically fetch the rules that are stored in the above file locations

apt install iptables-persistent

◊ To see the log of packets DROPped by iptables look at

/var/log/kern.log

♦ and check (grep) for any entries starting with "Denied by iptables"

h) I put the rules into the /root directory in a shell script called iptables-generate.sh , made executable and only accessible to root, with permissions 700 and root:root.

- Populate that file with the code below and run it once to establish the iptables rule set.
- I installed my servers on a NATed private network with NetworkID 192.168.1.0 so adjust your settings if you have a different setup
- The script shows the setup for the Wazo server so includes the SIP ports. For other servers, replace the SIP ports chain with the specifics of the other server (if any).
- This script assumes you are on an internal, private, IPv4 nework with 192.168.1.0/24 as the newtwork ID. If not, adjust the settings in the MANGLE table to suit your situation.
- This script has a safeguard built in so that if, by running this script you are cutoff from the server and cannot acknowledge the proper working of iptables, then after a brief timeout, the script clears iptables and returns you to an open iptables setup.
- If all works as you want, install iptables-persistent and reboot the server.

```
#!/bin/bash
#
# This file generates the iptables rules for the Wazo server
#
# It is very restrictive, allowing in only those packets explictly identified
# and DROPs all other packets attempting to come in to the server
# unless the packets are associated with an outbound connection already setup
#
# Packets going out and packets being forwarded are, by default allowed
#
#
#
# For all tables, flush the existing rules, reset all policy states to ACCEPT
#  and delete existing user-defined chains
function clear_iptables() {
        echo -e "\n\nClearing existing settings from iptables in both ipv4 and ipv6 tables\n"
        iptables --table filter --policy INPUT ACCEPT
        iptables --table filter --policy OUTPUT ACCEPT
```

```
iptables --table filter --policy FORWARD ACCEPT
iptables --table filter --flush
iptables --table filter --delete-chain
iptables --table mangle --policy PREROUTING ACCEPT
iptables --table mangle --policy OUTPUT ACCEPT
iptables --table mangle --policy FORWARD ACCEPT
iptables --table mangle --policy INPUT ACCEPT
iptables --table mangle --policy POSTROUTING ACCEPT
iptables --table mangle --flush
iptables --table mangle --delete-chain
iptables --table nat --policy PREROUTING ACCEPT
iptables --table nat --policy POSTROUTING ACCEPT
iptables --table nat --policy OUTPUT ACCEPT
iptables --table nat --flush
iptables --table nat --delete-chain
iptables --table raw --policy PREROUTING ACCEPT
iptables --table raw --policy OUTPUT ACCEPT
iptables --table raw --flush
iptables --table raw --delete-chain
iptables --table security --policy INPUT ACCEPT
iptables --table security --policy OUTPUT ACCEPT
iptables --table security --policy FORWARD ACCEPT
iptables --table security --flush
iptables --table security --delete-chain
iptables-save > /etc/iptables/rules.v4
#
ip6tables --table filter --policy INPUT ACCEPT
ip6tables --table filter --policy OUTPUT ACCEPT
ip6tables --table filter --policy FORWARD ACCEPT
ip6tables --table filter --flush
ip6tables --table filter --delete-chain
ip6tables --table mangle --policy PREROUTING ACCEPT
ip6tables --table mangle --policy OUTPUT ACCEPT
ip6tables --table mangle --policy FORWARD ACCEPT
ip6tables --table mangle --policy INPUT ACCEPT
```

```
ip6tables --table mangle --policy POSTROUTING ACCEPT
ip6tables --table mangle --flush
ip6tables --table mangle --delete-chain
ip6tables --table nat --policy PREROUTING ACCEPT
ip6tables --table nat --policy POSTROUTING ACCEPT
ip6tables --table nat --policy OUTPUT ACCEPT
ip6tables --table nat --flush
ip6tables --table nat --delete-chain
ip6tables --table raw --policy PREROUTING ACCEPT
ip6tables --table raw --policy OUTPUT ACCEPT
ip6tables --table raw --flush
ip6tables --table raw --delete-chain
ip6tables --table security --policy INPUT ACCEPT
ip6tables --table security --policy OUTPUT ACCEPT
ip6tables --table security --policy FORWARD ACCEPT
ip6tables --table security --flush
ip6tables --table security --delete-chain
ip6tables-save > /etc/iptables/rules.v6
# Show the admin that the flush worked
echo -e "\n\nThe status of the iptables (v4) tables after the initial flush is:"
echo -e "\n\nThe FILTER table"
iptables --table filter --list --verbose --numeric --line-numbers
echo -e "\n\nThe MANGLE table"
iptables --table mangle --list --verbose --numeric --line-numbers
echo -e "\n\nThe NAT table"
iptables --table nat --list --verbose --numeric --line-numbers
echo -e "\n\nThe RAW table"
iptables --table raw --list --verbose --numeric --line-numbers
echo -e "\n\nThe SECURITY table"
iptables --table security --list --verbose --numeric --line-numbers
echo -e "\n\n\n\nThe status of the iptables (v6) tables after the initial flush is:"
echo -e "\n\nThe FILTER table"
ip6tables --table filter --list --verbose --numeric --line-numbers
echo -e "\n\nThe MANGLE table"
ip6tables --table mangle --list --verbose --numeric --line-numbers
```

```
        echo -e "\n\nThe NAT table"
        ip6tables --table nat --list --verbose --numeric --line-numbers
        echo -e "\n\nThe RAW table"
        ip6tables --table raw --list --verbose --numeric --line-numbers
        echo -e "\n\nThe SECURITY table"
        ip6tables --table security --list --verbose --numeric --line-numbers
}
#
#
clear_iptables
#
# Define the FILTER table for ipv4 and ipv6 protocols
echo -e "\n\nGenerating the FILTER table for ipv4 and ipv6"
iptables --table filter --policy INPUT DROP
iptables --table filter --append INPUT --match conntrack --ctstate ESTABLISHED,RELATED --jump ACCEPT
iptables --table filter --append INPUT --in-interface lo --jump ACCEPT --match comment --comment "Enable Loopback traffic on ipv4"
iptables --table filter --new-chain Admin_Auth
iptables --table filter --append INPUT --jump Admin_Auth --match comment --comment "IPs with unfettered server access"
iptables --table filter --new-chain PBX_SIP_Traffic
iptables --table filter --append INPUT --protocol tcp --match multiport --destination-ports 5060,10000:20000 --jump PBX_SIP_Traffic --match comment --comment "WazoPBX Calls - TCP"
iptables --table filter --append INPUT --protocol udp --match multiport --destination-ports 5060,10000:20000 --jump PBX_SIP_Traffic --match comment --comment "WazoPBX Calls - UDP"
iptables --table filter --append INPUT --protocol icmp --source 192.168.1.0/24 --jump ACCEPT --match comment --comment "Allow ICMP from inside the LAN"
iptables --table filter --new-chain LAN_Traffic
iptables --table filter --append INPUT --source 192.168.1.0/24 --jump LAN_Traffic --match comment --comment "Allow Specified packets from inside the LAN"
iptables --table filter --append INPUT --protocol tcp --match multiport --source-ports 53 --match multiport --destination-ports 1024:65535 --source 192.168.1.0/24 --jump ACCEPT --match comment --comment "Evaluate DNS and accept LAN TCP DNS lookups"
```

iptables --table filter --append INPUT --protocol udp --match multiport --source-ports 53 --match multiport --destination-ports 1024:65535 --source 192.168.1.0/24 --jump ACCEPT --match comment --comment "Evaluate DNS and accept LAN UDP DNS lookups"

iptables --table filter --new-chain Nuisance_Traffic

iptables --table filter --append INPUT --jump Nuisance_Traffic

iptables --table filter --append INPUT --match limit --limit 5/min --jump LOG --log-prefix "Denied by iptables ipv4: " --log-level 7

#

iptables --table filter --append Admin_Auth --jump RETURN

#

iptables --table filter --append LAN_Traffic --protocol tcp --destination-port ssh --jump ACCEPT --match comment --comment "SSH"

iptables --table filter --append LAN_Traffic --protocol tcp --destination-port 23 --jump ACCEPT --match comment --comment "Telnet"

iptables --table filter --append LAN_Traffic --protocol tcp --match multiport --destination-ports 80,443 --jump ACCEPT --match comment --comment "Web"

iptables --table filter --append LAN_Traffic --protocol tcp --destination-port 123 --jump ACCEPT --match comment --comment "NTP Time Check"

iptables --table filter --append LAN_Traffic --protocol tcp --destination-port 5038 --jump ACCEPT --match comment --comment "Wazo-AMI for FOP2"

iptables --table filter --append LAN_Traffic --protocol tcp --destination-port 8667 --jump ACCEPT --match comment --comment "Wazo device configuration"

iptables --table filter --append LAN_Traffic --protocol tcp --destination-port 9486 --jump ACCEPT --match comment --comment "Wazo-update"

iptables --table filter --append LAN_Traffic --protocol tcp --destination-port 9497 --jump ACCEPT --match comment --comment "Wazo-auth for APIs"

iptables --table filter --append LAN_Traffic --protocol tcp --destination-port 9498 --jump ACCEPT --match comment --comment "Wazo-phoned for directory"

iptables --table filter --append LAN_Traffic --jump RETURN

#

iptables --table filter --append Nuisance_Traffic --protocol tcp --match multiport --destination-ports 67,68,135:139,111,513,520,445,1433,1434,1234,1524,3127 --jump DROP --match comment --comment "Unneccessary Network Scans - tcp"

iptables --table filter --append Nuisance_Traffic --protocol udp --match multiport --destination-ports 67,68,135:139,111,513,520,445,1433,1434,1234,1524,3127 --jump DROP --match comment --comment "Unneccessary Network Scans - udp"

iptables --table filter --append Nuisance_Traffic --jump RETURN

#

iptables --table filter --append PBX_SIP_Traffic --source 192.168.1.0/24 --jump ACCEPT --match comment --comment "SIP within LAN - TCP or UDP"

```
iptables --table filter --append PBX_SIP_Traffic --jump RETURN

#

#

ip6tables --table filter --policy INPUT DROP

ip6tables --table filter --append INPUT --match conntrack --ctstate
ESTABLISHED,RELATED --jump ACCEPT

ip6tables --table filter --append INPUT --in-interface lo --jump ACCEPT --match
comment --comment "Enable Loopback traffic on ipv6"

ip6tables --table filter --append INPUT --match limit --limit 5/min --jump LOG --log-
prefix "Denied by iptables ipv6: " --log-level 7

# Show the admin what the FILTER table now contains

echo -e "\n\nThe status of the iptables FILTER table is now:"

echo -e "\nFor ipv4"

iptables --table filter --list --verbose --numeric --line-numbers

echo -e "\nFor ipv6"

ip6tables --table filter --list --verbose --numeric --line-numbers

#

#

#

# Define the MANGLE table using the PREROUTING chain to catch bogus packets
early

# From https://javapipe.com/blog/iptables-ddos-protection

# These settings protect against high-volume DOS (Denial of Service) attacks

# by catching the packets early in the packet assessing process,

# (the MANGLE table is evaluated ahead of the FILTER table and the MANGLE table
has a PREROUTING chain)

# before they get to the INPUT Chain of the FILTER table,

# so it avoids using additional processing power to fight the attacks

#

# Drop invalid packets

iptables --table mangle --append PREROUTING --match conntrack --ctstate INVALID --
jump DROP

# Drop TCP packets that are new and are not SYN

iptables --table mangle --append PREROUTING --protocol tcp ! --syn --match conntrack
--ctstate NEW --jump DROP

# Drop SYN packets with suspicious MSS value

iptables --table mangle --append PREROUTING --protocol tcp --match conntrack --
ctstate NEW --match tcpmss ! --mss 536:65535 --jump DROP
```

```
# Block packets with bogus TCP flags

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE --jump DROP

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags FIN,SYN FIN,SYN --jump DROP

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags SYN,RST SYN,RST --jump DROP

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags FIN,RST FIN,RST --jump DROP

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags FIN,ACK FIN --jump DROP

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags ACK,URG URG --jump DROP

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags ACK,FIN FIN --jump DROP

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags ACK,PSH PSH --jump DROP

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags ALL ALL --jump DROP

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags ALL NONE --jump DROP

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags ALL FIN,PSH,URG --jump DROP

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags ALL SYN,FIN,PSH,URG --jump DROP

iptables --table mangle --append PREROUTING --protocol tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG --jump DROP

# Block spoofed packets

iptables --table mangle --append PREROUTING --source 224.0.0.0/3 --jump DROP

iptables --table mangle --append PREROUTING --source 169.254.0.0/16 --jump DROP

iptables --table mangle --append PREROUTING --source 172.16.0.0/12 --jump DROP

iptables --table mangle --append PREROUTING --source 192.0.2.0/24 --jump DROP

#iptables --table mangle --append PREROUTING --source 192.168.0.0/16 --jump DROP

iptables --table mangle --append PREROUTING --source 192.168.0.0/24 --jump DROP

iptables --table mangle --append PREROUTING --source 10.0.0.0/8 --jump DROP

iptables --table mangle --append PREROUTING --source 0.0.0.0/8 --jump DROP

iptables --table mangle --append PREROUTING --source 240.0.0.0/5 --jump DROP

# Drop fragments in all chains (ipv4 only)

iptables --table mangle --append PREROUTING --fragment --jump DROP
```

Limit connections per source IP

iptables --append INPUT --protocol tcp --match connlimit --connlimit-above 111 --jump REJECT --reject-with tcp-reset

Limit RST packets

iptables --append INPUT --protocol tcp --tcp-flags RST RST --match limit --limit 2/s --limit-burst 2 --jump ACCEPT

iptables --append INPUT --protocol tcp --tcp-flags RST RST --jump DROP

Limit new TCP connections per second per source IP

iptables --append INPUT --protocol tcp --match conntrack --ctstate NEW --match limit --limit 60/s --limit-burst 20 --jump ACCEPT

iptables --append INPUT --protocol tcp --match conntrack --ctstate NEW --jump DROP

#

#

ip6tables --table mangle --append PREROUTING --match conntrack --ctstate INVALID --jump DROP

Drop TCP packets that are new and are not SYN

ip6tables --table mangle --append PREROUTING --protocol tcp ! --syn --match conntrack --ctstate NEW --jump DROP

Drop SYN packets with suspicious MSS value

ip6tables --table mangle --append PREROUTING --protocol tcp --match conntrack --ctstate NEW --match tcpmss ! --mss 536:65535 --jump DROP

Block packets with bogus TCP flags

ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE --jump DROP

ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags FIN,SYN FIN,SYN --jump DROP

ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags SYN,RST SYN,RST --jump DROP

ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags FIN,RST FIN,RST --jump DROP

ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags FIN,ACK FIN --jump DROP

ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags ACK,URG URG --jump DROP

ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags ACK,FIN FIN --jump DROP

ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags ACK,PSH PSH --jump DROP

ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags ALL ALL --jump DROP

```
ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags ALL NONE --jump DROP

ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags ALL FIN,PSH,URG --jump DROP

ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags ALL SYN,FIN,PSH,URG --jump DROP

ip6tables --table mangle --append PREROUTING --protocol tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG --jump DROP

# Limit connections per source IP

ip6tables --append INPUT --protocol tcp --match connlimit --connlimit-above 111 --jump REJECT --reject-with tcp-reset

# Limit RST packets

ip6tables --append INPUT --protocol tcp --tcp-flags RST RST --match limit --limit 2/s --limit-burst 2 --jump ACCEPT

ip6tables --append INPUT --protocol tcp --tcp-flags RST RST --jump DROP

# Limit new TCP connections per second per source IP

ip6tables --append INPUT --protocol tcp --match conntrack --ctstate NEW --match limit --limit 60/s --limit-burst 20 --jump ACCEPT

ip6tables --append INPUT --protocol tcp --match conntrack --ctstate NEW --jump DROP

# Show the admin what the MANGLE table now contains

echo -e "\n\nThe status of the iptables MANGLE table is now:"

echo -e "\nFor ipv4"

iptables --table mangle --list --verbose --numeric --line-numbers

echo -e "\nFor ipv6"

ip6tables --table mangle --list --verbose --numeric --line-numbers

#

#

#

# Save the result to the permament files at /etc/iptables/rules.v4 and rules.v6

iptables-save > /etc/iptables/rules.v4

ip6tables-save > /etc/iptables/rules.v6

echo -e "\n\nThe iptables settings have been saved to:"

echo -e "/etc/iptables/rules.v4     for ipv4 rules"

echo -e "/etc/iptables/rules.v6     for ipv6 rules"

#

echo -e "\nFor ipv4, the iptables rules are now:"

echo -e "\nThe FILTER table"
```

```
iptables --table filter --list-rules
echo -e "\nThe MANGLE table"
iptables --table mangle --list-rules
echo -e "\nThe NAT table"
iptables --table nat --list-rules
echo -e "\nThe RAW table"
iptables --table raw --list-rules
echo -e "\nThe SECURITY table"
iptables --table security --list-rules
#
echo -e "\n\nFor ipv6, the iptables rules are now:"
echo -e "\nThe FILTER table"
ip6tables --table filter --list-rules
echo -e "\nThe MANGLE table"
ip6tables --table mangle --list-rules
echo -e "\nThe NAT table"
ip6tables --table nat --list-rules
echo -e "\nThe RAW table"
ip6tables --table raw --list-rules
echo -e "\nThe SECURITY table"
ip6tables --table security --list-rules
#
#
# Ask the admin to confirm that this is what they want
# If the admin does not respond in 5 seconds, assume the iptables settings locked them out
# and flush the rules, allowing the admin to come back in
echo -e "\n\nWithin 5 seconds, confirm, by pressing Enter, that you can read this prompt.\n"
read -t 5 -p "Failure to press Enter in the time allowed will result in all iptables settings being cleared." UserResponse
if [ $? -eq 0 ]
then
    # Enter was pressed so no further action is required
        echo -e "\nYou pressed Enter so the iptables settings will be maintained.\n"
```

```
else
    # Enter was not pressed so clear all the iptables settings
        clear_iptables
        echo "" > /etc/iptables/rules.v4
        echo "" > /etc/iptables/rules.v6
        echo -e "\\nYou did not press Enter in the allowed time."
        echo -e "This has been interpreted to mean you were locked out after creating the iptables rules,"
        echo -e "so the iptables rules have been deleted and the firewall is now open"
fi
```

12) Configuring fail2ban - brute force detection

a) fail2ban comes pretty much configured the way you want it. fail2ban is enabled for ssh, apache and asterisk logon monitoring. That is all I wanted.

b) If you want, edit the config file settings for fail2ban

- Check /etc/fail2ban/jail.d/wazo.conf and /etc/fail2ban/jail.d/defaults-debian.conf to see what setttings have already been customized for Wazo and do not change these
- Do NOT edit the settings in /etc/fail2ban/jail.conf
  - ◊ /etc/fail2ban/jail.conf settings are part of the Wazo install and may be changed with an upgrade so put your changes into a file that will survive upgrades.
  - ◊ You could create a file in /etc/fail2ban/jail.d/ called {whatever}.conf with your customizations and if you are familiar with fail2ban setup go ahead.
  - ◊ You could just create a file /etc/fail2ban/jail.d/jail.local with proper permissions (root:root and 644) with no content and add whatever customizations you want to that file, but you would likely end up jumping back and forth between jail.local and jail.conf to see the proper syntax, so follow the next instructions instead
    - ♦ If you do create a fresh jail.local file and add custom settings, make sure to add the section heading under which that setting resides.
  - ◊ Create a copy of /etc/fail2ban/jail.conf into /etc/cfail2ban/jail.local with all the settings commented out so you have the template in /etc/cfail2ban/jail.local can uncomment and customize only the settings you want customized

    ```
    awk '{ printf "# "; print; }' /etc/fail2ban/jail.conf | sudo tee /etc/fail2ban/jail.local
    ```

    - ♦ If you do incomment a setting, make sure to uncomment the section heading under which that setting resides.
  - ◊ Edit /etc/fail2ban/jail.d/jail.local by uncommenting the settings you want to change and making your changes. Some examples you may want to look at include
    - ♦ ignoreip =

If you want to make sure that your PC or a range of devices (ir everyone on the internal LAN) never get blocked by fail2ban, uncomment this line and put the IPs here

- bantime =

  bantime is the number of seconds that a violator is banned with default at 600 seconds or 10 minutes. You may want to make this longer, say 1 hour or 18000 seconds (I did)

- maxretry =

  maxretry and findtime work in concert, with maxretry being the maximum number of times (within the findtime time period) that a visitor is allowed to fail before they are blocked by fail2ban. maxretry defaults to 3 but Wazo changes that to 5. I changed it back to 3.

- findtime =

  findtime and maxretry work in concert, with findtime being the time period, in seconds, the the counter for failed attempts looks back. This defaults to 600 seconds or 10 minutes.

  findtime = 10m

- dest=root@localhost

  I left this as is and forwarded all root emails to a desired email accout , but if you want, you could edit the destination email address for any of the enable sections of fail2ban

  If you are going to change this, make sure to check all three settings for

  dest={the actual email address to which you want notices to go}

  sendername = Fail2Ban

  sender = {an authorized email address that will be accepted by the MTA}

  mta = sendmail

◊ Restart fail2ban

  systemctl stop fail2ban (or service fail2ban stop)

  systemctl start fail2ban (or service fail2ban start)

  fail2ban-client status

◊ If you have any problems with fail2ban configuration and fail2ban stops working, try running the following commands to help with trouble shooting

  systemctl status fail2ban

  journalctl -xe -u fail2ban

◊ At this point fail2ban sets up to automatically add rules and chains to itables.

- However fail2ban may not do the addition until fail2ban is activated by a violater so you may not see any addtions to the iptables list until and unless fail2ban detects and protects agains a violater.
- So if you want - from a workstation you can afford to have temporarlity banned - try to login 6 times (maxretry = 5) with invalid credentials. On the 6th attempt, the response will change from "permission denied" to "connection failed" and iptables will be updated with

a new chain f2b-sshd

a new entry in the chain contining the banned IP Address

a new entry in the INPUT chain for the filter table, directing traffic to the f2b-ssshd chain

- ♦ If you have violations caught by the other fail2ban jails, those will also be added to iptables.
- ♦ If iptables is restarted or the server is rebooted, fail2ban reverts back to a fresh status, with no bans retained

◊ The log file for fail2ban is at

/var/log/fail2ban.log

◊ To see what IPs are banned

- ♦ List the current iptables and look for the fail2ban chains

iptables --list --numeric

or

iptables --list --numeric | grep -i REJECT

- ♦ The name of the chain is NOT necessarily the same as the name of the jail. The chain name and jail name will likely be close enough that you will recognize the relationship. It is the jail name you will need to unban an ip so to get a list the jail names run

fail2ban-client status

- ♦ The three jails setup by fail2ban in Wazo are

sshd

asterisk-wazo

wazo-provd

- ♦ You can get a list of IP Addresses that are banned in each jail by running

fail2ban-client status sshd

fail2ban-client status asterisk-wazo

fail2ban-client status wazo-provd

◊ If you want to manually unban an IP_Address that was banned and you cannot wait for the bantime to expire, you can use the fail2ban-client command from the system CLI

fail2ban-client set {jailname}unbanip {IP_Address}

◊ If you want to see all the options available to yo with fail2ban-client, run, as a standalone command with no options

fail2ban-client

13) Transferring files between a PC and the Linux Server

a) If you hjave Webmin installed on the Linux server (SuiteCRM), use it by going to Other ->> File Manager and upload/download using the GUI.

b) If you have a desktop interface on the Linux server (Linux Mint), use the File Manager in it. Enable File Sharing on the PC and add a bookmark for the PC Folder in the Linux File Manager and you

can upload/download using that GUI.

c) If you do not have Webmin (or some other GUI File Manager) installed on the server (Proxmox, WazoPBX), use Putty's pscp program to transfer files from the PC to the Linux server.

d) Install pscp on the PC

- You may have already installed this if you chose the full installation option when installing Putty and if so, you can skip this step.
- Go to https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html and download the .msi installer for the full package. At the time of theis document is was Putty 0.73 and the .msi file was at https://the.earth.li/~sgtatham/putty/latest/w64/putty-64bit-0.73-installer.msi
- Use the .msi to install pscp.exe. If you use the installer file, it will properly add pscp.exe file to your PC and include it in the PATH variable. If not, see https://it.cornell.edu/managed-servers/transfer-files-using-putty for manually configuring pscp.exe in the PATH variable so it can be launched from the cmd window in Windows from any directory.
- Had we not configured our servers to prevent UserName/UserPassword access to the server, we would have an easy time transferring files. Simply use

  pscp {path to document on PC - including dodument name} {LinuxUser}@{Linux_server}:{Path to intended file location on Linux server - including filename}

  ◊ The Linux server would have prompted you for the password and your file would be transferred
- However, for good reason, we did disable UserName/UserPassword access to the server, so we need to use the key-pair authentication to access the server.

  ◊ This assumes you have previously created a key-pair, stored the public key on the Linux Server in the authorized_keys file, and connected at least once using SSH so your PC has stored the server as a "known host".

  ◊ Note that Putty does NOT use the same key format for the Private key as the standard used by OPENSSH when an rsa key-pair is created using ssh-keygen, so if you used the Linux ssh-keygen command to create your key-pair, you need to convert the keygen-generated Private key to the Putty compatible version (ending in .ppk) - for the Private key only - and then use the Putty-Compatible Private key with pscp. To do this,

  ♦ Open PuttyGen and from within PuttyGen, load the ssh-keygen-created Private key with Conversions -> Import Key

  ♦ Select the appropriate Private key format : Normally rsa with 2048 bits

  ♦ Click on Save private key

  ◊ Now use the following command to transfer a file from the PC to the server

  pscp -i "(path\of\the\privatekey\privatekey.ppk}" -P {Port number , usually 22} "{C:\temp\example_file.txt}" user@server:"{path/file/to/be/stored}"

  ♦ Note the difference in the directory delimeters: Windows uses \ and Linux uses /

  ◊ If required, login to the server and upgrade the user to superuser and move the file, with appropriate permissions, to the location and permissions you want.
- Remember, we did NOT allow direct access to the root user, so we need to do the transfer as the

non-root user and then, if needed, SSH in to the server and move the file to the location we want it if we need root-level permissions to put the file (with proper permissions) somewhere.

14) Locating and Formatting Sound files

a) If you have already fetched and converted the sound files for use in Wazo, use those

b) If you are getting the sound files for the first time, follow the instructions below

c) The jamendo site gives you a GUI to download the music files. If you use another site that does not give you a GUI to download files and need to use a Windows equivalent of wget on linux, us bitsadmin on Windows

- bitsadmin /transfer myDownloadJob /download /priority normal {URL_to_File_to_be_downloaded} "{path_to_filename_on_Windows}"
  - ◊ in our case as an eg, using asterisk files from the asterisk site:

    bitsadmin /transfer myDownloadJob /download /priority normal http://downloads.asterisk.org/pub/telephony/sounds/asterisk-core-sounds-en-ulaw-current.tar.gz "{path_to_Windows_directory_with_filename_and_suffix}"

d) Go to http://www.jamendo.com/en/album/23661

- This takes you to the Jamendo web site - an online repository of artist-submitted music that is not necessarily designed for but is often appropriate for Musi On Hold sound
- Album 23661 (The Reno Project 1.0) contains some nice instrumental music that can be used for Music On Hold (MOH) sounds
  - ◊ I liked 3 tracks: System, The Field and Friday
- If this is for your personal use, download to your PC the free mp3 file set associated with the Reno Project 1 album. This is for my personal use so I used this.
  - ◊ If this is for commercial use, read and follow the instructions for commercial licensing

e) You now need to convert the selected mp3 files into a sound format that can be played by asterisk

- First you convert the files to the standard (PC compatible, full stereo audio) .wav format
- Then you convert the file to make it mono and a lower bit depth for asterisk

f) You use the linux application sox to do this conversion so :

- If you do not have sox installed or it is installed without the mp3 codec

  apt install sox libsox-fmt-all
- Transfer the mp3 files to a linux box with sox on it (with the mp3 conversion codec)
  - ◊ I put them into a directory on the linux box called /home/{user}/Music/PBXsoundFiles

    05-161544-Reno Project-System.mp3

    09-161545-Reno Project-The Field.mp3

    12-161540-Reno Project-Friday.mp3
  - ◊ Convert the .mp3 files to the PC .wav format (note the -PC in the target file name and replacement of spaces with - in the target file name)

sox "/home/richard/Music/PBXsoundFiles/05-161544-Reno Project-System.mp3" "/home/richard/Music/PBXsoundFiles/$(basename -s .mp3 "05-161544-Reno-Project-System-PC").wav"

sox "/home/richard/Music/PBXsoundFiles/09-161545-Reno Project-The Field.mp3" "/home/richard/Music/PBXsoundFiles/$(basename -s .mp3 "09-161545-Reno-Project-The-Field-PC").wav"

sox "/home/richard/Music/PBXsoundFiles/12-161540-Reno Project-Friday.mp3" "/home/richard/Music/PBXsoundFiles/$(basename -s .mp3 "12-161540-Reno-Project-Friday-PC").wav"

◊ Convert the PC .wav files to the asterisk-compatible .wav format (note the elimination of -PC in the target file name)

sox "/home/richard/Music/PBXsoundFiles/05-161544-Reno-Project-System-PC.wav" -r 8k -c 1 -e signed-integer -b 16 "/home/richard/Music/PBXsoundFiles/05-161544-Reno-Project-System.wav"

sox "/home/richard/Music/PBXsoundFiles/09-161545-Reno-Project-The-Field-PC.wav" -r 8k -c 1 -e signed-integer -b 16 "/home/richard/Music/PBXsoundFiles/09-161545-Reno-Project-The-Field.wav"

sox "/home/richard/Music/PBXsoundFiles/12-161540-Reno-Project-Friday-PC.wav" -r 8k -c 1 -e signed-integer -b 16 "/home/richard/Music/PBXsoundFiles/12-161540-Reno-Project-Friday.wav"

- Transfer the asterisk-compatible wav files to the asterisk server and put them in the proper location.
  - ◊ With Wazo 20.05, there is a GUI for doing this
    - ♦ Sound Files -> Add [and follow the prompts]

      If you already have the .wav as the suffix on the file, you do not need to enter the "Format: wav". If you do, it just adds another .wav suffix to the end of the file, so no harm done, but you do not need to.

    - ♦ On Wazo, the default MoH sound files are stored at /var/lib/asterisk/moh/default

      If manually placing sound files, make sure the permissions are 660 and asterisk:www-data

  - ◊ With non-Wazo, asterisk installs, the sound files can be placed into their own directory in

    /var/lib/asterisk/moh

    /var/lib/asterisk/moh/RC_MoH_Set1

    - ♦ or they can be placed into the default directory which is

      /var/lib/asterisk/moh/default

Appendix 1 - Personalized Settings