

UACM

Universidad Autónoma
de la Ciudad de México

NADA HUMANO ME ES AJENO

COLEGIO DE CIENCIA Y TECNOLOGÍA
LICENCIATURA EN INGENIERÍA EN SISTEMAS
ELECTRÓNICOS Y DE TELECOMUNICACIONES

**Diseño e implementación de servicios de VoIP con Asterisk,
usando el entorno de virtualización Proxmox**

TESIS

QUE PARA OPTAR POR EL TÍTULO DE
**LICENCIADO EN INGENIERÍA EN SISTEMAS
ELECTRÓNICOS Y DE TELECOMUNICACIONES**

PRESENTA

ERIC ALBERTO REYES GUTIÉRREZ

DIRECTOR

M. EN C. JOEL YAZBEK BUENDÍA GÓMEZ

Ciudad de México, septiembre de 2024.

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS[©]

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

Resumen

En este trabajo se presentó la solución a un problema de telecomunicaciones, que se desarrolló con los conocimientos y habilidades obtenidos durante la carrera en la universidad.

Todo surgió tras la necesidad de realizar una actualización de los servicios y en la infraestructura, dado que la empresa donde actualmente trabajo decidió actualizar el sistema operativo de los servicios de voz. Esta actualización es necesaria para después cumplir con la certificación de los servidores y servicios dentro de los equipos, esta certificación logra cumplir con la normativa evitando vulnerabilidades en los equipos y/o servicios.

Esta actualización generó un beneficio a largo plazo para los equipos. Compañeros del trabajo decidieron actualizar los servidores optando por virtualizar los servicios, por lo cual, en el primer paso se llevó a cabo la actualización de los equipos, reemplazando los 4 servidores anteriores que contaban con un PBX instalado en cada uno y se reemplazaron con 2 servidores nuevos donde en cada uno están instalados 2 PBX's virtuales.

Este cambio generó varios beneficios a corto y largo plazo, los cambios inmediatos fueron la reducción de equipos operando, esto benefició en un menor consumo de energía eléctrica, menor gasto en mantenimiento, etc. Por lo anterior, se obtuvo un mayor aprovechamiento de los servidores ya que utilizamos eficientemente todos los recursos del servidor.

El cambio tecnológico realizado impulsó a realizar el proyecto, con pruebas aplicadas en el LRC, cumpliendo con los objetivos definidos en el proyecto.

Para el presente proyecto se planeó implementar PBX virtuales para proporcionar servicios de voz, es decir, dentro de un servidor físico virtualizamos 3 PBX's, todo gracias al software Proxmox.

ProxmoxVE es un software de uso libre que permite crear maquinas virtuales, se utilizó el software en el proyecto ya que resulta útil por la configuración de clúster que más adelante daremos detalles sobre esta funcionalidad en Proxmox.

Se crearon 3 máquinas virtuales, siendo los PBX, donde cada uno tiene el sistema Asterisk instalado.

Asterisk es un software de uso libre, para crear una central telefónica privada (PBX), siendo el sistema más utilizado por las empresas para contar con una o varias centrales telefónicas privadas, ofrece diferentes herramientas con las que todo PBX cuenta, algunos ejemplos son: IVR, contestador automático, grupo de llamada, desvíos, entre otras funciones.

La implementación de los PBX's se realizó en el LRC, contando con 3 equipos telefónicos para conectar por cable de red, además se cuenta con un total de 8 CPU's en el LRC donde se instala Proxmox, creando un clúster para diferentes servicios.

La finalidad fue crear un clúster para poder contar con un sistema distribuido de almacenamiento y otros recursos generando mayor eficiencia.

Se utilizaron para el proyecto solo 3 de los 8 CPU's, para las pruebas creamos 3 PBX's virtuales, después se registraron los teléfonos en cada PBX, para finalmente establecer la comunicación entre extensiones.

Para ello, se realizaron configuraciones dentro de cada PBX, creando las extensiones y registrando cada teléfono, estableciendo una troncal de comunicación, rutas de entrada y rutas de salida por cada PBX. Con las configuraciones necesarias, conectamos los teléfonos por cable de red al switch que utilizamos conectado a la misma red para el registro de las extensiones. Gracias a las troncales y rutas de salida configuradas, la idea fue tener comunicación entre extensiones registradas en cada uno de los PBX.

Logrando así tener el servicio de los PBX virtuales, simulando que cada uno es un PBX de diferente plantel y entre ellos pueda existir comunicación.

Agradecimientos

Para el presente proyecto y trabajo de tesis realizado, quiero dar un emotivo agradecimiento a mi director de tesis, el profesor Joel Yazbek Buendía Gómez. Quien gracias a su conocimiento por aplicar las herramientas de virtualización, aplicando a los servicios de voz, fue que pude desarrollar y aplicar los conocimientos adquiridos en mi formación académica, sumando la experiencia en el campo laboral para que el presente trabajo pudiera realizarse.

Logrando un excelente trabajo, para la obtención de mi título profesional en ingeniería de sistemas electrónicos y de telecomunicaciones.

Agradezco a mi padre, quien me ayudó para que pudiera continuar con mis estudios, a mis familiares, con quienes también tuve la dicha que me extendieron su mano durante mi proceso por terminar la universidad, a todos y cada uno de ellos sea poca o mucha la ayuda que recibí, a todos mis familiares les quiero agradecer.

Agradezco a quienes confiaron en mí, me brindaron su apoyo para que el presente trabajo se pudiera culminar.

Estoy agradecido con la vida por darme la oportunidad de llegar hasta donde me encuentro ahora, confío en que llegaré mas lejos.

Por último, me agradezco a mí mismo, por nunca rendirme, por tan difícil que fue el camino, pero nunca es tarde para lograr las metas o proyectos que uno mismo se propone.

Contenido

Lista de Figuras	7
Lista de Tablas	9
1. Introducción	13
1.1. Planteamiento del problema	13
1.2. Análisis del problema	13
1.3. Objetivos	15
1.3.1. Objetivo General	15
1.3.2. Objetivos Específicos	1
1.4. Justificación del proyecto	1
2. Antecedentes	3
2.1. Historia de la virtualización	3
2.2. ¿Cómo funciona la virtualización?	5
2.3. Núcleo de virtualización KVM	5
2.4. Tipos de virtualización	6
2.4.1. Virtualización de los datos	6
2.4.2. Virtualización de escritorios	7
2.4.3. Virtualización de servidores	7
2.4.4. Virtualización de los sistemas operativos	8
2.4.5. Virtualización de las funciones de red	9
2.5. Características de la virtualización:	9
2.6. Linux containers LXC	10
2.7. Plataforma de virtualización proxmox virtual environment	11
2.8. Clúster	13
2.9. Software asterisk	13
2.10. Central telefónica privada	15
2.11. Troncal telefónica	15
2.12. PSTN - Red telefónica pública conmutada	15
2.13. Enlaces T1/E1	15
2.14. TDM	16

2.15. Protocolo SIP	16
2.16. Protocolo RTP	17
2.17. PoE, Switch PoE y cómo funciona	17
2.18. Protocolo de asignación dinámica de direcciones IP DHCP	17
2.19. Protocolos TCP/UDP	18
2.20. Dirección física MAC address	18
2.21. Protocolo TLS	18
3. Desarrollo del trabajo	21
3.1. Planteamiento del trabajo	21
3.1.1. Requerimientos necesarios para virtualizar un servidor	21
3.1.2. Ventajas de utilizar proxmox	21
3.1.3. Servicio de voz IP asterisk	22
3.2. Materiales y componentes	23
3.2.1. Implementación en la empresa Grupo Salinas	23
3.2.2. Implementación en el laboratorio de redes de computadoras del plantel SLT	23
3.3. Procedimiento para el desarrollo del proyecto	25
3.3.1. Creación de la memoria USB booteable	25
3.3.2. Instalación de proxmox	27
3.3.3. Crear un clúster con proxmox	28
3.3.4. Creación de las VM para los PBX virtuales	38
3.3.5. Instalación del software asterisk	38
3.3.6. Copias de seguridad en proxmox	39
3.3.7. Creación de los PBX virtuales con las copias de seguridad	43
3.3.8. Configurar troncales entre PBX	50
4. Análisis de resultados	59
5. Conclusiones y recomendaciones para trabajo futuro	65
5.1. Recomendaciones para trabajo futuro	67
Referencias	69

Lista de Figuras

2.1. Servidores físicos con aplicaciones.	3
2.2. Virtualización aplicada para aprovechar los recursos del servidor.	4
2.3. Virtualización aplicada para el almacenamiento de datos.	7
2.4. Virtualización aplicada a escritorios virtuales.	7
2.5. Virtualización aplicada a servidores virtuales.	8
2.6. Hipervisor con VM de diferentes sistemas operativos aplicados.	8
2.7. Hipervisor con diferentes enlaces de redes virtuales.	9
2.8. Protocolo SIP aplicado a servicios de voz sobre IP.	16
3.1. Servidores físicos que utilizamos en el LRC del plantel SLT.	24
3.2. Página oficial de ProxmoxVE para descargar el instalador.	25
3.3. Procedimiento para crear la USB booteable.	26
3.4. Instalación de Proxmox en el primer servidor físico.	27
3.5. Datos para configurar el primer hipervisor.	28
3.6. Representación del clúster con los PBX virtuales.	29
3.7. Servidores físicos y su configuración de red.	29
3.8. Clúster entre hipervisores creado con Proxmox.	30
3.9. Archivos para la conexión remota por VNC viewer.	31
3.10. Ejecutable para ingresar por VNC viewer.	31
3.11. Conexión por VNC viewer a la PC D-19 del LRC.	32
3.12. Desde Virtualbox hay 3 VM con Proxmox para realizar un clúster.	33
3.13. Configuraciones donde se habilita la virtualización anidada.	33
3.14. CLI del hipervisor donde se creó el clúster.	34
3.15. CLI del hipervisor agregado al clúster.	35
3.16. CLI del segundo hipervisor agregado al clúster.	35
3.17. Muestra el estado del clúster.	36
3.18. Muestra el número de equipos que hay en el clúster.	36
3.19. Clúster con los 3 hipervisores en Virtualbox.	37
3.20. Clúster con los equipos del LRC del plantel SLT.	37
3.21. Porcentajes de todo el hardware en el clúster.	38
3.22. Archivo de instalación Asterisk en el hipervisor.	39
3.23. Ruta para guardar el archivo de respaldo del PBX.	40

3.24. Se selecciona detener la VM.	40
3.25. Tipo de compresión para la copia de seguridad.	41
3.26. Proceso del backup.	41
3.27. Archivo backup para restaurar.	42
3.28. Opción de almacenamiento para restaurar el PBX.	42
3.29. Ventana para restaurar la copia del PBX.	43
3.30. Interfaz de red con MAC duplicada.	43
3.31. Después de borrar se agrega una nueva interfaz de red.	44
3.32. Comprobando diferencia de MAC address entre PBX01 y PBX02.	44
3.33. Dirección IP que tiene el PBX en el puerto eth0.	45
3.34. Pasos para modificar la interfaz de red eth0.	45
3.35. Archivo de la interfaz de red del PBX 2.	46
3.36. Se comprueba la nueva dirección IP configurada al PBX.	47
3.37. Resultado del cambio de hostname.	48
3.38. PBX 1 y PBX 2 con su propia dirección IP y MAC.	48
3.39. Interfaz web FreePBX, crear una cuenta de administrador.	49
3.40. Configuración de un teléfono, muestra extensión y SIP server.	50
3.41. Opción Trunks.	51
3.42. Agregar una nueva troncal.	51
3.43. Agregar nombre a la troncal.	52
3.44. Datos del enlace para la troncal hacia el PBX02.	53
3.45. Guardar cambios y aplicar configuraciones en el PBX.	53
3.46. Estatus de las extensiones registradas y troncales a los otros PBX.	54
3.47. Creando una ruta de salida.	54
3.48. Parámetros que se configuran en “Outbound Routes”.	55
3.49. Dial Patterns para especificar las marcaciones.	55
3.50. “Outbound Routes” configuradas en el PBX01.	56
3.51. Configuraciones SIP.	56
3.52. Cambio de puerto 5060 entre chan SIP y chan PJSIP.	57
3.53. Número de extensión y troncales en el PBX02 plantel Cuauhtémoc.	57
3.54. Número de extensión y troncales en el PBX03 Plantel Centro Histórico.	57
3.55. Primer prueba realizada con 3 teléfonos IP, un switch con puertos PoE y los 3 PBX virtuales con sus troncales configuradas.	58
4.1. Diagrama de la topología de red del proyecto en operación.	61
4.2. Segunda fase de pruebas en el LRC.	62
4.3. Herramienta sngrep analizando todas las llamadas que se realizan dentro del PBX.	63
4.4. Información de los equipos que establecieron una llamada.	63
4.5. Traza de una llamada desde el inicio hasta terminar la comunicación.	64

Lista de Tablas

2.1.	Tabla comparativa para los diferentes sistemas de virtualización.	12
3.1.	Componentes de los CPU del LRC.	24
4.1.	Lista de todas las pruebas realizadas en cada PBX.	60

Acrónimos

San Lorenzo Tezonco (SLT)
Private Branch Exchange (PBX)
Proxmox Virtual Environment (ProxmoxVE)
Interactive Voice Response (IVR)
Central Process Unit (CPU)
Voice over Internet Protocol (VoIP)
Internet Protocol (IP)
Virtual Machine (VM)
International Business Machines (IBM)
Technologies Information (TI)
Kernel-based Virtual Machine (KVM)
Linux Containers (LXC)
Open source container-based virtualization in Linux (OpenVZ)
Interfaz de Usuario Gráfica (GUI)
Sistema operativo libre en desarrollo compatible con Linux (GNU)
Protocolo de Iniciación de Sesión (SIP)
Protocolo de Transporte en tiempo real (RTP)
Red Telefónica Pública Conmutada (PSTN)
Power over Internet (PoE)
Protocolo de configuración dinámica de host (DHCP)
Protocolo de Control de Transmisión (TCP)
User Datagram Protocol (UDP)
Media Access Control (MAC)
Controlador de Interfaz de Red (NIC)
Controlador de Interfaz de Red Virtual (vNIC)
Transport Layer Security (TLS)
Random Access Memory (RAM)
Universal Serial Bus (USB)
Virtual Network Computing (VNC)
Secure Shell (SSH)
Solid State Drive (SSD)
Command Line Interface (CLI)

Capítulo 1

Introducción

Surge la problemática para certificar los servidores y servicios de voz, por lo que es necesario actualizar el sistema operativo, por esa razón se exponen en el siguiente documento los puntos clave para realizar una actualización innovadora para los servicios de voz.

1.1. Planteamiento del problema

Tras el constante cambio, actualización e innovación tecnológica en las comunicaciones, surge la necesidad de innovar en el campo de las telecomunicaciones, más enfocada en los servicios de voz sobre IP.

Por normativas de la organización, específicamente el departamento de seguridad de la información de Grupo Salinas, es necesario certificar los servidores que alojan los servicios de voz, cumpliendo con las especificaciones para evitar vulnerabilidades en los equipos y servicios, uno de los principales requisitos que solicitan es tener el sistema operativo de los PBX actualizado.

Se planteó la actualización de PBX en un entorno virtual, donde estarán implementados los servicios de VoIP en un hipervisor, de esta manera contamos con la actualización del sistema operativo y llevamos los servicios a una novedosa tecnología que ofrece grandes beneficios a largo plazo.

Este cambio constituye un gran beneficio para la organización, cumpliendo con certificar los servidores, lograr la normativa en seguridad de los servicios y no tener vulnerabilidades.

1.2. Análisis del problema

Para llevar acabo esta actualización, la empresa requiere implementar la virtualización en 4 servidores, teniendo dos centrales telefónicas por cada servidor físico, para dar servicio a un total de 4,050 usuarios distribuidos en todo el corporativo.

Originalmente los servicios de voz están distribuidos en 4 PBX, catalogados de la siguiente manera:

PBX 0 - Son todas aquellas extensiones que se encuentran en los sótanos, niveles bajo las torres y otros sitios fuera del corporativo Torres Esmeralda.

PBX 1 - Son todas las extensiones que ofrecen servicio a todos los pisos de la Torre 1 en Torres Esmeralda.

PBX 2 - Son todas las extensiones que ofrecen el servicio a todos los pisos de la Torre 2 en Torres Esmeralda.

PBX 3 - Son todas las extensiones que ofrecen el servicio a todos los pisos de la Torre 3 en Torres Esmeralda.

Después de realizar el análisis de requerimientos, existen 8 servidores físicos en operación, son 4 PBX principales y 4 de respaldo, representando un alto costo para la organización, ya que se deben cubrir los costos de operación y mantenimiento de cada equipo, sumando los costos en suministro de energía eléctrica, en un principio hablamos de actualizar 4 conmutadores telefónicos, pero se necesita tener redundancia en los servicios, teniendo 4 PBX principales y otros 4 PBX que estarán como equipos de respaldo.

Sin contar con la virtualización, se tienen 4 servidores físicos en el site principal y otros 4 servidores físicos en el site secundario como respaldo de los equipos principales, ya que en caso de que algún PBX falle, estará operando otro PBX de respaldo.

Debido a los altos costos que dicha implementación conlleva, se planteó una solución tecnológica más económica, ofreciendo beneficios para la operación, desempeño e infraestructura, obteniendo una mayor disponibilidad, mantenimiento y facilidad de acceso, la cual cuenta con la misma capacidad de recursos que se requiere para este caso, con la ventaja de que en el momento que se requiera crecer los servicios, esta solución tiene la facilidad de poder incrementar los servicios sin ningún costo adicional, hablando de infraestructura.

Obtenemos un ahorro en costos que se detallan en:

- Operativos
- Energéticos
- Instalaciones
- Mantenimiento
- Equipos físicos
- Cableado
- Ahorro de espacios

Si la empresa requiere de alguna aplicación adicional del servicio de VoIP, con esta solución se pueden implementar tantos servicios como sea la capacidad del servidor, por lo tanto, se propone instalar PBX virtuales.

La finalidad es disminuir el uso de servidores físicos alojando nuestros PBX virtuales de manera distribuida en los servidores, para un balanceo de carga y un mayor aprovechamiento de los recursos de cada equipo, de esta manera mantenemos actualizados los servicios, aprovechamos recursos y mejoramos la infraestructura ahorrando recursos físicos a largo plazo.

Teniendo nuestras centrales telefónicas virtuales, cubriremos la cantidad de usuarios que requieren, y así, dentro de cada PBX poder crear, registrar y administrar extensiones para servicios de voz sobre IP.

Para ello se necesita instalar 4 máquinas virtuales, alojando en cada una de ellas un PBX, en cuanto al número de extensiones, se requiere de los 4 PBX para tener una cantidad de entre 1,000 a 1,100 extensiones por cada PBX, cubriendo así el total de extensiones que son alrededor de 4,050. La implementación de PBX virtuales serán los que darán pie al desarrollo del proyecto, logrando reducir costos significativos en el caso de llevar el proyecto a otras organizaciones.

Al utilizar entornos virtuales obtenemos ventajas significativas, como:

- Tener un manejo y control de la operación.
- Un respaldo de servidores en caso de algún fallo o error.
- Instalación compatible con diferentes sistemas operativos.
- Se tiene la ventaja para en un futuro aumentar la capacidad según se requiera en su momento.
- Aislamiento total con otras máquinas virtuales.

También cuenta con algunas desventajas, como:

- Costos elevados en recursos para el caso de empresas pequeñas.
- Se requiere gran cantidad en espacio de almacenamiento en los servidores.
- Puede existir una sobre carga en el rendimiento del equipo físico.
- Si falla el equipo físico fallarán las VM en él.

1.3. Objetivos

1.3.1. Objetivo General

Tenemos la finalidad de implementar servicios de voz utilizando el entorno virtual con Proxmox, como una novedosa tecnología, con el fin de ofrecer mejoras en cuanto a la implementación, operación y servicio de la comunicación por voz a través de internet, creando PBX con el software Asterisk, para configurar y crear troncales SIP en cada PBX para lograr la comunicación entre PBX virtuales.

1.3.2. Objetivos Específicos

1. Aprender sobre el software de virtualización llamado ProxmoxVE y utilizar las herramientas que nos ofrece para crear VM.
2. Instalar ProxmoxVE en cada uno de los CPU's del LRC del plantel SLT.
3. Crear un clúster con todos los CPU's del LRC.
4. Utilizar el software Asterisk para crear PBX virtuales, configurar y aprender a desplegar servicios de VoIP.
5. Aprender sobre el uso y finalidad de las copias de seguridad en Proxmox.
6. Habilitar 3 PBX virtuales y comprobar configuraciones de cada PBX.
7. Validar servicios correctos de cada PBX antes de comenzar con las pruebas de comunicación.
8. Realizar pruebas de comunicación entre extensiones de diferentes PBX y comprobar la comunicación correcta entre PBX.
9. Cumplir con los resultados al contar con una lista de revisión de servicios y pruebas.

1.4. Justificación del proyecto

Tomando el panorama de la organización se opta por virtualizar los servicios de voz, para el presente proyecto, se va a realizar una implementación de servicios de VoIP en un entorno virtualizado, se requiere de 3 PBX, uno en cada plantel con la finalidad de tener comunicación entre PBX de cada plantel.

Esto representa a comenzar adquiriendo experiencia con servicios de voz, conociendo mejor nuestro entorno de virtualización a utilizar, desplegar configuraciones de servicios y realizar pruebas o resultados de los servicios para los PBX.

Decidimos utilizar el software Proxmox como solución de virtualización, que nos permite crear un clúster, mejorando la administración en nuestro servicio de voz sobre IP.

Otro beneficio es la fácil configuración de las VM, ya que con una VM creada, podemos generar un backup y posteriormente replicarlo o simplemente tener el respaldo de la VM, nos permite realizar copias de seguridad en vivo sin interrumpir operaciones, entre otras funciones, logrando generar tantos servicios como sean necesarios.

Tener un panorama mucho más amplio sobre los servicios de VoIP, conociendo el cómo funciona, cómo se transmite, protocolos que utiliza, la infraestructura que se emplea y cómo se logra realizar una implementación de comunicación VoIP utilizando el entorno virtual.

A continuación en el siguiente capítulo describimos algunos de los conceptos necesarios para entender más a detalle el presente proyecto.

Capítulo 2

Antecedentes

En este capítulo se presentan los conceptos y la historia de la virtualización, los servicios de red, redes de computadoras, los sistemas de telefonía y todo concepto necesario para entender el proyecto realizado.

2.1. Historia de la virtualización

En el año 1960 se comienza a realizar la virtualización, la compañía International Business Machines, por sus siglas IBM dió comienzo a lo que hoy conocemos como la tecnología de virtualizar servicios. Realizando un intento por mejorar la utilización del CPU, creando una segmentación de las computadoras centrales llamadas “mainframes”, permitiendo lograr ejecutar múltiples tareas y procesos al mismo tiempo.

En aquella época los equipos eran costosos y solo se podían usar para una sola aplicación o servicio, tal como se muestra en la figura 2.1.[1 p. 62, 63]

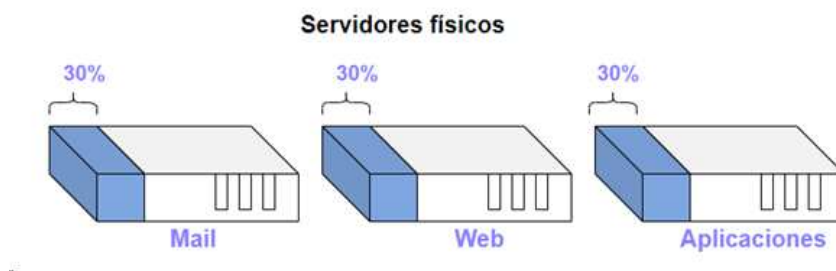


Figura 2.1: Servidores físicos con aplicaciones.

Particionando un “mainframe” podían aprovechar los recursos del equipo y la inversión al máximo, capaz de realizar varias operaciones en simultáneo. Eso solamente se logra creando particiones lógicas, (hoy en día son las VM) logrando destinar los recursos del mainframe a cada aplicación o servicios particionados, en la figura 2.2 vemos la partición de los recursos de un servidor permitiendo ahorrar en equipos.[1 p. 62, 63]

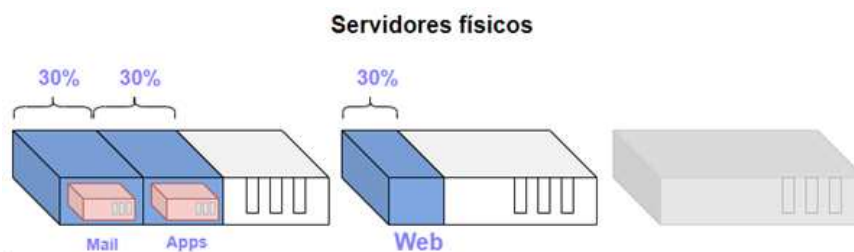


Figura 2.2: Virtualización aplicada para aprovechar los recursos del servidor.

En la década de 1980, la arquitectura x86 significó un avance tecnológico, ya que las máquinas tienen una mayor memoria física de lo que le era posible a sus predecesores, permitiendo así que los programas puedan almacenar mayor cantidad de datos en la memoria.

Tal popularidad causó que rápidamente abandonaran la virtualización, fue reemplazada por un modelo de informática distribuida, con aplicaciones cliente-servidor y escritorios de arquitectura x86 más económicos.[1 p. 62, 63]

Para la década de los 90 se adoptaron los sistemas operativos Windows y Linux para servidores x86, convirtiéndose en el estándar del sector. Con el avance en el desarrollo del hardware, surgió la misma problemática, similar que los años 60, teniendo equipos altamente eficientes, pero no se estaban aprovechando todos los recursos con que contaban los equipos, por lo cual utilizar un solo “equipo” para cada aplicación conlleva un mayor gasto de espacio, energía y desperdicio de recursos.[1 p.62, 63]

Nuevamente surge la idea de segmentar el hardware, para que funcione como múltiples equipos, independientes uno del otro pero compartiendo los mismos recursos físicos. La virtualización creó nuevos problemas de infraestructura y operaciones en TI, por su gran crecimiento e implementación de servicios.[1 p. 69, 70]

Algunos de los principales problemas fueron:

- Utilización de infraestructura demasiado baja: La mayoría de organizaciones tienen ejecutándose una sola aplicación por cada servidor, evitando vulnerabilidades o afectaciones por tener más de una aplicación en un solo servidor.
- Incremento de costos en cuanto a mantenimiento y operación: cada vez van en aumento los costos del crecimiento de infraestructuras físicas.
- Las organizaciones gastan cantidades exponenciales: para dar mantenimiento a los servidores y gastando también en personal para dicho mantenimiento.
- Se generaron procedimientos y reglas intensivas para asegurar la continuidad del negocio, para evitar vulnerabilidades ante desastres naturales, terrorismo y ataques a la seguridad, sea para escritorios como para servidores.
- Actualizaciones, seguridad, mantenimiento y escritorios para el usuario final conllevan el reto de aplicar reglas de seguridad, políticas para la gestión y acceso sin entorpecer

el trabajo de los usuarios, teniendo que aplicar parches de seguridad y actualizaciones generando mayores costos para evitar brechas en la seguridad.[1 p. 69, 70]

En 1998 VMware fue fundada por un grupo de investigadores de la Universidad de California, tratando de resolver varias deficiencias de la arquitectura x86. Para 1999 se creó la virtualización para los sistemas de arquitectura x86, ofreciendo aislamiento completo, para uso general y pudiendo elegir el sistema operativo para cada aplicación.[1 p. 69, 70]

El resultado es una VM de alto rendimiento, que se adapta y mantiene una total compatibilidad de software, VMware fue pionero en esta técnica.[1 p. 69, 70]

A principios del año 2000 comenzó a utilizarse la virtualización de servidores, dando pie al desarrollo de nuevas tecnologías que favorecieron la virtualización, como por ejemplo los hipervisores.[1 p. 69, 70]

2.2. ¿Cómo funciona la virtualización?

Utilizando una interfaz de usuario conocida como “hipervisor”, facilita la administración de los recursos físicos para los equipos virtuales alojados en el servidor físico.[3]

Los hipervisores al encontrarse instalados directamente en el hardware (sea una PC, servidor, laptop, etc.), pueden manipular el sistema operativo del equipo, comúnmente se instala un hipervisor de esa manera y se utiliza a menudo por cualquiera que desee implementar virtualización de sus servicios. Gracias al hipervisor podemos dividir los recursos físicos para los servicios virtuales que se requieran. Para finalmente poder hacer funcionar los entornos virtuales con los recursos destinados por el hipervisor, a esto se le conoce como máquina virtual.[3]

“La máquina virtual funciona como un archivo de datos único; por eso, tal como ocurre con cualquier archivo digital, es posible trasladarla de una computadora a otra, abrirla en cualquiera de ellas, y tener la tranquilidad que funcionará de la misma forma”.[4]

Cuando algún usuario ejecuta una instrucción mientras se utiliza el entorno virtual, esa instrucción la transmite el hipervisor al sistema físico con la finalidad de solicitar recursos físicos, almacenando los cambios en la memoria caché.[4]

2.3. Núcleo de virtualización KVM

En el presente proyecto los servidores que se virtualizaron están basados en KVM, Kernel Virtual Machine, es decir, es una solución para poder realizar una virtualización completa

utilizando Linux, este tipo de virtualización nos ofrece todos los componentes virtualizados: tarjeta de red, discos duros, tarjeta gráfica, etc.[5]

Cuando contamos con un servicio dentro de nuestra VM basado en KVM y necesitamos procesar información, esas instrucciones y cambios se realizan con la misma rapidez, es decir, como si se estuvieran realizando de manera física en el servidor, esto ocurre si nuestro hipervisor es bare metal o tipo 1 y la máquina virtual utilizada está basada en el Kernel KVM.[5]

Las KVM pueden realizar migración activa, es decir, puede pasar una máquina virtual de un entorno a otro sin que sea afectada su operación o sus datos, sin interrupciones o pérdida de datos. Permitiendo tener las conexiones de red activas y la operación de la máquina virtual sin interrupciones, logrando un proceso de migración sin verse afectadas las aplicaciones de la máquina virtual. Otro beneficio es que las KVM guarda el estado actual de la VM, para que pueda almacenarla y reanudarla más adelante.[5]

2.4. Tipos de virtualización

La virtualización es una representación de un equipo, datos o programa, que imita las características del hardware, ya sea de aplicaciones, almacenamiento, servidores o redes.[5]

2.4.1. Virtualización de los datos

Se pueden crear bases de datos de manera virtualizada, con la finalidad de procesar varios datos de diferentes fuentes, agregar y transformar el contenido dependiendo las necesidades del usuario, logrando obtener la información cuando se requiera.[1 p. 63-68][5]

Podemos tener una base de datos dentro de una VM, tal como si fuera un servidor dedicado para recopilar la información dentro de ese entorno virtual, la siguiente imagen es una representación de una base de datos en un entorno virtual.

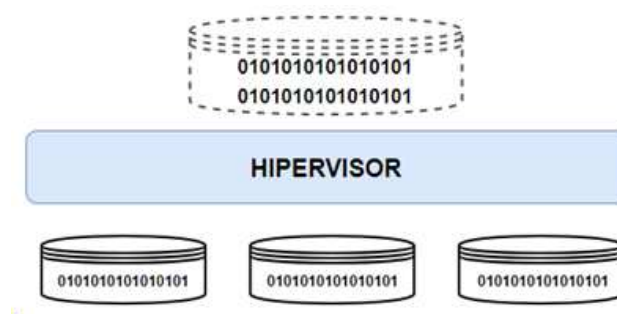


Figura 2.3: Virtualización aplicada para el almacenamiento de datos.

2.4.2. Virtualización de escritorios

Se pueden tener varios escritorios en una sola máquina, permitiendo que un administrador o una herramienta de administración pueda crear varios escritorios en diferentes máquinas físicas al mismo tiempo, tal como se muestra en la figura 2.4. De esta manera cualquier administrador de los escritorios virtuales pueden aplicar actualizaciones, configuraciones de seguridad y control de servicios para todos los escritorios virtuales.[1 p. 63-68][5]

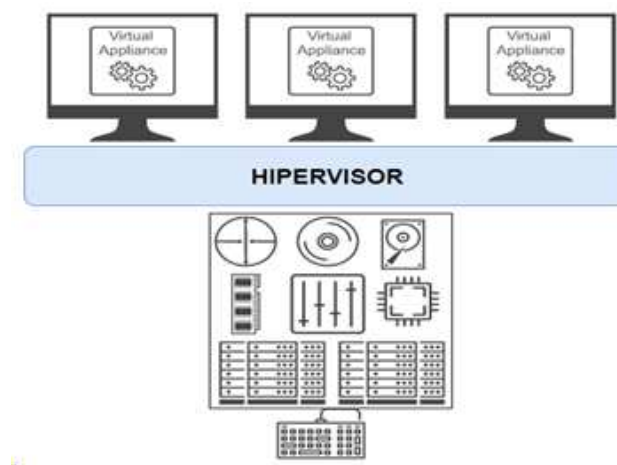


Figura 2.4: Virtualización aplicada a escritorios virtuales.

2.4.3. Virtualización de servidores

Los servidores están diseñados para almacenamiento y/o procesamiento de datos, para ofrecer servicio a otros equipos (sea una PC) y ejecuten procesos con esos datos, como se mencionó anteriormente, virtualizar un servidor es para dividir los recursos del equipo y

puedan ser utilizados para varias tareas, logrando aplicar diversas funciones específicas.[1 p. 63-68][5]

En la figura 2.5 vemos una representación de como contamos con servidores virtuales habitando dentro de un servidor físico, a los servidores virtuales se les puede instalar o almacenar diferentes servicios y/o aplicaciones.

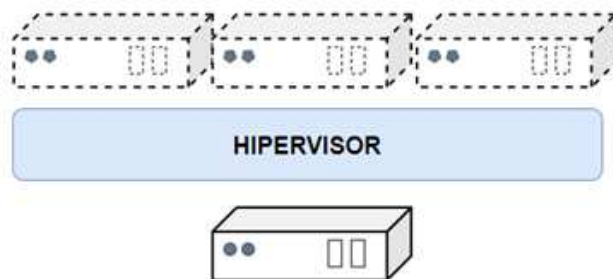


Figura 2.5: Virtualización aplicada a servidores virtuales.

2.4.4. Virtualización de los sistemas operativos

Los sistemas operativos también se virtualizan, es una forma útil de ejecutar entornos de Linux y Windows de manera paralela, reduciendo grandes gastos en hardware, aumenta la seguridad ya que todos los sistemas virtuales se pueden supervisar y aislar. Este tipo de virtualización es útil para realizar pruebas con software que solo sea compatible con un sistema operativo específico.[1 p. 63-68][5]

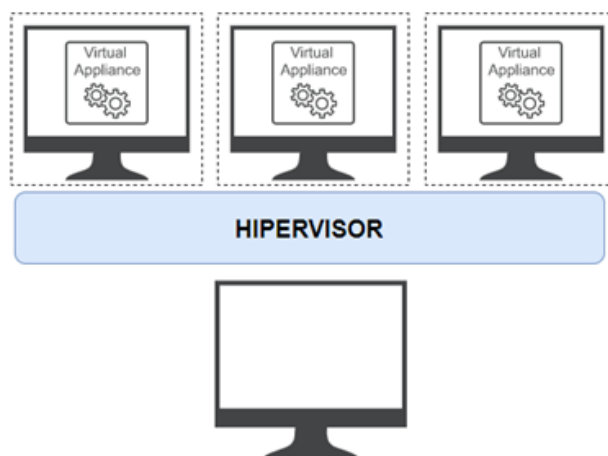


Figura 2.6: Hipervisor con VM de diferentes sistemas operativos aplicados.

2.4.5. Virtualización de las funciones de red

La virtualización de redes es empleada ampliamente para las telecomunicaciones, emplear una virtualización de redes reduce considerablemente el uso de equipos dedicados a la interconexión en una red (conmutadores, enrutadores, servidores, cables ethernet, etc.), necesarios para crear varias redes independientes.[1 p. 63-68][5]

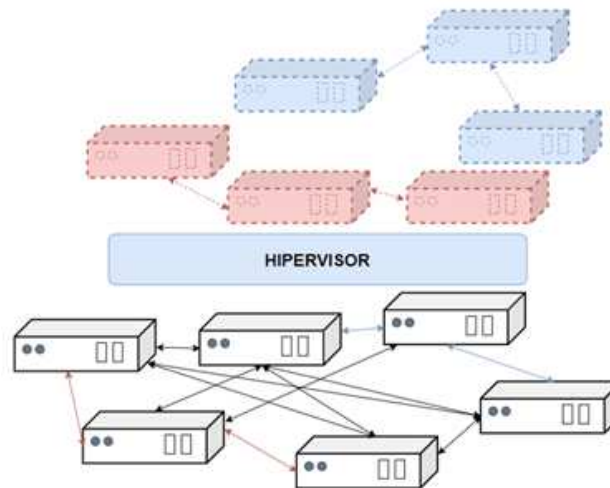


Figura 2.7: Hipervisor con diferentes enlaces de redes virtuales.

2.5. Características de la virtualización:

1. Crea particiones en el equipo físico
 - Puedes tener y ejecutar varios sistemas operativos dentro de una sola máquina física.
 - Administra todos los recursos físicos del sistema entre las máquinas virtuales.
2. Posibilidad de aislamiento
 - Permite dividir la seguridad y los fallos a nivel de hardware.
 - Controles avanzados de los recursos para un mejor rendimiento.
3. Encapsulación
 - Guarda una máquina virtual en almacenamiento local o como archivo.
 - Realiza copias de seguridad para las máquinas virtuales y se pueden migrar con facilidad de un equipo a otro.

4. Hardware independiente

- Capaz de migrar una máquina virtual a un servidor físico.

2.6. Linux containers LXC

Los contenedores Linux son un sistema aislado del equipo físico anfitrión, un contenedor funciona directamente con el kernel del host o servidor físico, es un sistema virtual pero hay diferencias entre un contenedor y una máquina virtual. El contenedor solo utiliza de los componentes necesarios para poder ejecutar la aplicación instalada, mientras que una VM necesita mas recursos ya que dentro de la VM se instala un nuevo sistema operativo para hacer funcionar dicha aplicación, en otras palabras el contenedor es mas ligero que una VM.[7][8][9]

La diferencia consiste en que para una VM hay que destinar recursos para que pueda operar, mientras que un contenedor cuenta con los mismos recursos del servidor físico anfitrión, por esta diferencia un contenedor tiene la ventaja al ser ligero y que al utilizar una aplicación o servicio los procesos que realiza los toma directamente del servidor físico, mientras que para una VM estará limitada la aplicación con los recursos que previamente destinamos al crear una VM.

Ventajas de un contenedor:

- Es más ligero ya que utiliza solo los recursos necesarios para hacer funcionar la aplicación dentro del contenedor.
- Fácil operación, en cuanto se crea y se instala dicha aplicación puede operar inmediatamente.
- Ya existen contenedores base creados, o puedes crear una plantilla de tu contenedor para crear contenedores de manera rápida y sencilla.

Desventajas de un contenedor:

- No se recomiendan para servicios en producción, ya que puede presentar fallos e inestabilidad.
 - Solo se recomienda para experimentar, realizar test y/o utilizar temporalmente para reemplazar una aplicación en producción.
-

2.7. Plataforma de virtualización proxmox virtual environment

Proxmox VE es un programa con entorno para crear virtualización de código abierto. Es una distribución de Linux basada en Debian, permite el uso y la gestión de VM y contenedores. Proxmox VE cuenta con una interfaz de usuario web y herramientas por línea de comandos.[8 p. 7-10]

Tiene dos tipos de virtualización compatibles: los contenedores basados en LXC y la virtualización con KVM.[8 p. 7-10]

El desarrollo de Proxmox VE comenzó cuando dos desarrolladores Linux, Martin Maurer y Dietmar utilizando OpenVZ para virtualizar, encontraron que no contaba con alguna herramienta capaz de poder realizar copias de seguridad, y mucho menos una GUI (interfaz de usuario gráfica) de administración.[8 p. 7-10]

KVM fue apareciendo para Linux y se añadió poco después, para el año 2008 fue el primer lanzamiento de la plataforma siendo atractivo para su uso en la virtualización.[8 p. 7-10]

La gran ventaja es que fue una de las pocas plataformas en contar con un entorno web de administración gráfica, proporcionando una virtualización completa y el soporte necesario para contenedores.[8 p. 7-10]

Características de Proxmox

De cada máquina virtual que se crea puede guardar una copia de la configuración. Es posible administrar los recursos para darle a cada servidor virtual lo que necesite para operar de manera correcta y sin limitaciones. Pueden moverse de un nodo(host) a otro para un balanceo de carga, fácil de administrar, con una interfaz amigable y de fácil operación.[8 p. 7-10]

Se pueden virtualizar sistemas operativos como: Microsoft (32/64 bits), Windows 2000, Windows XP, Windows Vista / 2008, Windows 7, Windows 8 y Windows 10 (se está probando). Sistema Operativo Linux / UNIX (32/64 bits), Solaris, Linux 3.X, Linux 2.6, Linux 2.4, entre otros. Fácil administración para dispositivos de red (bonding/bridging/vlan/ovswitch).[8 p. 7-10]

Realizamos una tabla comparativa de los diferentes sistemas de virtualización comparados con Proxmox donde nos indica las diferencias y los apartados significativos por el cual elegimos Proxmox como nuestra herramienta de virtualización.

Diferencia frente a otros sistemas de virtualización

Producto	Proxmox	vSphere	XEN	Hyper-V	VirtualBox
Soporte a sistemas operativos	Todas las versiones de Windows y Linux, BSD y otros (No oficialmente soportados)	Windows, Linux y Unix	Algunas versiones de Linux, últimas versiones de Windows	Últimas versiones de Windows, Linux muy pobre	Todas las versiones de Windows y Linux
Open Source	Si	No	No	No	Es gratuito pero no es Open Source
Soporte OpenVZ	Si, Linux	No	No	No	No
Control unificado	Si, web	Si, requiere software adicional	No, interfaz poco amigable	Si, requiere software adicional	Si
Soporte	Si, de pago	No	No	No	No
Alta Disponibilidad	Si	Si	Si	Si, requiere producto adicional	No
Snapshots (backup rápido)	Si	Si	Si	Limitado	Si
Migración rápida	Si	Si	Si	Si	No
Monitoreo fácilmente configurable	Si	No	No	No	No

Cuadro 2.1: Tabla comparativa para los diferentes sistemas de virtualización.

Como nos muestra en la tabla anterior algunas características son similares entre sistemas de virtualización pero Proxmox cuenta con la ventaja de ser un sistema de código abierto a diferencia de las otras soluciones.

Algunos beneficios que ofrece Proxmox son:

- Es una solución para el manejo de máquinas virtuales.

- Permite el manejo de sistemas operativos distintos como: Windows, Linux entre otros sistemas no oficialmente soportados.
- Permite la creación de clústeres y tiene una alta disponibilidad.
- Se puede acceder vía web a la mayoría de las funciones.
- Bajos requerimientos de hardware para utilizarlo.
- Compatible con otras soluciones de virtualización (VirtualBox, Hyper-V, vSphere).
- Virtualización utilizando KVM y OpenVZ.
- Software de uso libre.

2.8. Clúster

Se llama clúster al conjunto de servidores que de manera coordinada operan los recursos de cada servidor como si se tratase de uno solo, en otras palabras se describe como una sola unidad de varios equipos, al estar dichos equipos unidos en un clúster se puede representar como un solo equipo o un solo servidor que cuenta con componentes de gran capacidad, estos componentes son hardware (disco duro, memoria RAM, núcleos de procesador, interfaces de red, etc.) y al emplearlos como uno solo, puede soportar gran cantidad de aplicaciones y/o servicios, tantos como sea la capacidad de estos componentes hardware.

Una ventaja de utilizar clústers para servicios es que entre los mismos equipos que forman el clúster las cargas de trabajo de servicios se pueden balancear aprovechando los recursos de manera eficiente.[2 p. 87]

2.9. Software asterisk

Se trata de una solución para telefonía IP de código abierto, proporciona funcionalidades de una central telefónica privada (PBX). Como cualquier PBX podemos llegar a configurar y conectar n-cantidad de teléfonos, dependiendo de la capacidad del equipo host, para hacer llamadas entre sí dentro de una misma red u organización, e incluso realizar la conectividad para hacer llamadas dentro de la organización hacia el exterior a través de la PSTN, configurando una troncal de salida respectivamente.[11]

Mark Spencer, fundador de Digium, fue el creador de asterisk, actualmente es su principal desarrollo en conjunto con más desarrolladores, con el tiempo han podido agregar funcionalidades y corrección de algunos errores para mejorar asterisk.[11 p. 18,19]

Asterisk se distribuye para diferentes sistemas operativos (Windows, Solaris, BSD), aunque la principal plataforma nativa y con mayor desarrollo es para GNU/Linux.[11]

Se puede utilizar bajo una interfaz web llamada freepbx, para un mejor manejo y gestión de las funcionalidades, teniendo una mejor administración y control de equipos configurados.[11]

La principal función es la escalabilidad, puede funcionar para empresas pequeñas, medianas y de gran tamaño, una gran cantidad de usuarios pueden estar distribuidos en varias sedes alrededor del mundo.[12]

Las principales funcionalidades de Asterisk son:

- Escalabilidad total de líneas y extensiones
- Uso de teléfonos convencionales o de VoIP
- Integración de telefonía fija y móvil
- Integración de la telefonía con la red de datos y las aplicaciones informáticas
- Distribución flexible de llamadas entrantes y salientes
- Operadora automática IVR
- Buzón de voz personalizado con opción a envío de correo electrónico
- Conferencias
- Grupos de timbrado
- Desvío de extensiones

También asterisk cuenta con una interfaz web que lo hace un sistema de fácil manejo y operación para desplegar servicios, tanto puedes ejecutar configuraciones a nivel interfaz por línea de comandos como también se pueden aplicar cambios por la interfaz web de una manera mas amigable y de fácil manejo. Actualmente conocemos las interfaces web de Elastix y freePBX, siendo freePBX la interfaz web que utilizamos en el presente proyecto, debido a que es una interfaz mas actual, con mayor funcionalidades y que sigue en constantes actualizaciones, mientras que Elastix desde el año 2016 ya no recibió actualizaciones, pasando a ser un proyecto de código cerrado, fue comprado por 3CX.[11 p. 18,19]

Lo interesante de asterisk es que reconoce diferentes protocolos de comunicación VoIP (SIP, H.323, IAX y MGCP) y permite también la unificación de tecnologías (VoIP, GSM y PSTN).[12]

Para el presente proyecto utilizamos la comunicación de los servicios sobre el protocolo SIP, ya que SIP es el estándar para establecer una comunicación de servicios de VoIP y vídeo, entre otros detalles que más adelante mencionamos.

2.10. Central telefónica privada

Conocida como PBX ó central telefónica privada, un PBX puede utilizar múltiples líneas troncales de servicio, con la finalidad de lograr comunicación con otros PBX e incluso con algún proveedor de servicios para la comunicación fuera de la organización hacia la PSTN. Generalmente las líneas troncales de los PBX son enlaces digitales E1 y T1 que soportan hasta 30 canales (líneas) de voz por enlace.[11 p. 12,13]

2.11. Troncal telefónica

Una línea troncal o un enlace troncal es una intersección de comunicación entre dos puntos, interconecta la transmisión de datos, específicamente de manera digital. Una central telefónica privada o conocida como PBX, puede utilizar múltiples líneas troncales para tener comunicación de manera interna entre otros PBX o una troncal de servicios hacia la PSTN, sería una conexión de servicios con otros proveedores de servicios de voz.[12 p. 165]

El presente proyecto se necesita de la configuración de troncales para la comunicación, específicamente troncales SIP.

Una troncal SIP es como lo mencionamos anteriormente, una interconexión entre dos puntos pero la conexión es de manera virtual, es decir, realizamos una conexión de servicios de voz a través de internet entre PBX, más adelante veremos como se realiza la configuración.[15 p. 125]

2.12. PSTN - Red telefónica pública conmutada

Son empresas que ofrecen el servicio de telefonía a nivel internacional, nacional o regional, con la infraestructura necesaria para ofrecer los servicios de comunicación telefónica, para llamadas locales, nacionales e internacionales, dicho servicios conectados a través de una red de interconexiones llamada “nodos”.[12 p. 14, 136]

2.13. Enlaces T1/E1

“Las conexiones T1 son paquetes compuestos por 24 canales de multiplexado por división de tiempo (TDM) de 64 kbps a través de un circuito de cobre de cuatro hilos, creando un ancho de banda total de 1.544 Mbps. Para una conexión E1 es un paquete compuesto por 32 canales de 64 Kbps, con un total de 2.048 Mbps”.[13]

2.14. TDM

Es un método para transmitir múltiples señales ya sean digitales o analógicas a través de un solo canal, este tipo de señales pueden ser de voz, datos y vídeo, al manejar múltiples señales sin tener interferencias hace que podamos maximizar el uso del canal y lograr una mayor eficiencia al transferir datos, esto se logra ya que puede dividir la transmisión de las señales por intervalos de tiempo.[14 p.708][17 p. 248]

2.15. Protocolo SIP

SIP es uno de los protocolos de señalización para voz sobre IP. Protocolo de inicio de sesión: es un protocolo desarrollado con la finalidad de ser el estándar para iniciar, modificar y finalizar sesiones interactivas de usuario, para elementos multimedia como el video, voz, mensajería instantánea, juegos en línea y realidad virtual. [24 p. 204-206][11 p. 32-47][15 p. 63][22 p. 389]

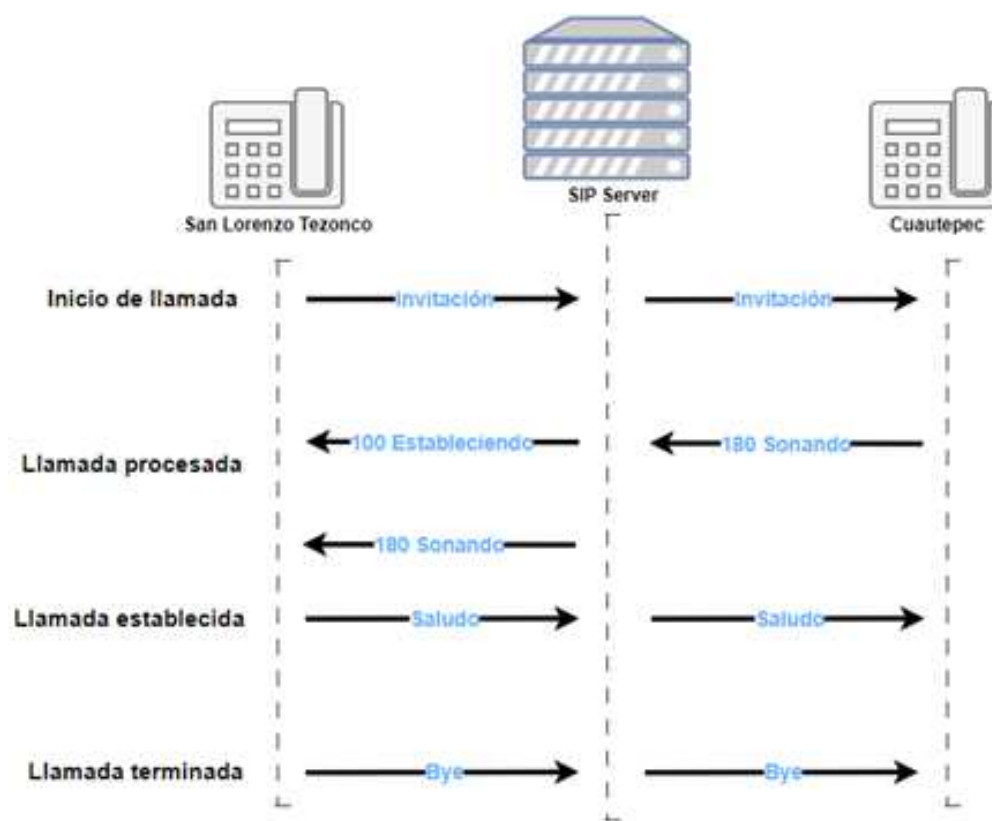


Figura 2.8: Protocolo SIP aplicado a servicios de voz sobre IP.

Como vemos en la imagen anterior, es un ejemplo de cómo se establece una comunicación SIP entre extensiones de diferentes planteles, vemos una extensión de SLT que se quiere comunicar con otra extensión de Cuauhtepc, al comienzo de establecer la llamada el protocolo SIP empieza la negociación, comenzando por una invitación de SLT a Cuauhtepc, al comprobar la disponibilidad nos regresa la respuesta de “sonando”, es cuando en el teléfono escuchamos un tono que está llamando, hasta el momento que Cuauhtepc contesta se nos asigna un canal proporcionado por RTP (Real Time Protocol) y ese canal es exclusivo de la llamada mientras duren en línea, en el momento que se decide finalizar la llamada se manda una respuesta de “bye” donde corta la comunicación y se libera ese canal para ser utilizado por otra llamada.

2.16. Protocolo RTP

Protocolo de transporte en tiempo real (RTP), es un protocolo a nivel aplicación, se emplea para la transmisión de datos en tiempo real, comúnmente utilizado en servicios de voz y vídeo, este protocolo es considerado una de las bases para los servicios de VoIP y trabaja en conjunto con el protocolo SIP, usando primordialmente UDP para la transmisión de los datos.[11 p. 50-53][22 p. 367]

2.17. PoE, Switch PoE y cómo funciona

Primero vamos a definir PoE, el significado se refiere a “Power over Ethernet”, no es más que la energización de cualquier equipo que requiera de energía eléctrica para funcionar pero a través de una conexión por cable Ethernet. Ahora existen switches PoE y sin PoE, la diferencia está en que los puertos de los primeros proporcionan energía eléctrica. La ventaja de un switch PoE es que genera suministro de datos y energía a través de un único cable Ethernet de manera simultánea.[16 p. 88][17]

Para entornos de oficinas, comúnmente se utilizan teléfonos VoIP y de vídeo, cámaras IP, puntos de acceso inalámbricos, dispositivos de audio, etc. Todos los equipos mencionados necesitan de una alimentación eléctrica.[16 p. 88][17]

2.18. Protocolo de asignación dinámica de direcciones IP DHCP

Es un protocolo cliente/servidor capaz de permitir el establecer una conexión al dispositivo que requiera conectarse a dicha red. Al conectar un equipo a la red, DHCP nos asigna una

dirección IP del segmento de la red a la que nos conectamos, esta información de red nos asigna una dirección IP, máscara de subred y la puerta de enlace predeterminada. [18][24 p. 7-9][22 p. 189]

2.19. Protocolos TCP/UDP

TCP (Protocolo de control de transmisión), es uno de los protocolos principales en redes TCP/IP, orientado a conexión por lo cual en la transmisión de datos necesita de una respuesta confirmando que los datos se recibieron de manera correcta. Este protocolo es muy utilizado para transferencia de archivos y correos electrónicos.[15 p. 61]

El protocolo UDP (User Datagram Protocol), es un protocolo de transmisión de datos no orientado a conexión, es decir, la entrega de paquetes no está garantizada al destino final, aunque se caracteriza por el envío de datos más rápido y ligero. Comúnmente el protocolo UDP se utiliza para las comunicaciones en tiempo real por voz sobre IP, videoconferencias y mensajería instantánea.[15 p. 61]

2.20. Dirección física MAC address

Se conoce como dirección física del equipo, es un identificador o clave para cada dispositivo y es único para cada equipo. Por medio de la MAC se puede identificar cada dispositivo en la red y es útil en redes como medida de seguridad para permitir el acceso o denegar por medio de su MAC address.[19 p. 201]

vNIC (Controlador de interfaz de red virtual) En el presente proyecto la dirección MAC y el vNIC se usan para que nuestras VM sean equipos independientes una de la otra, la vNIC es un controlador de interfaz de red virtual, al crear VM utilizados como nuestros PBX deben llevar una tarjeta de red virtual (vNIC) y cada una de nuestras VM cuenta con su propia MAC virtual asignada. [20][25 p. 310, 311]

2.21. Protocolo TLS

TLS (Transport Layer Security), es un protocolo diseñado para proporcionar seguridad en las comunicaciones, a través de una red informática. El protocolo es utilizado en aplicaciones como correo electrónico, voz sobre IP y mensajería instantánea.[21][22 p. 291]

El objetivo principal es proporcionar seguridad, con el uso de certificados entre dos o más aplicaciones que se comunican, logrando privacidad, integridad y autenticidad en la

comunicación. Se ejecuta en la capa de presentación y se compone de dos capas: el registro TLS y los protocolos de protocolo de enlace TLS. [21][22 p. 291]

Teniendo un panorama más amplio sobre los conceptos que se necesitaron para entender el presente proyecto, a continuación se da pie al siguiente capítulo que expone todo el desarrollo realizado para los servicios de voz usando el entorno de virtualización.

Desarrollo del trabajo

3.1. Planteamiento del trabajo

En este capítulo se expone el procedimiento realizado para cumplir los objetivos, implementar servidores virtuales para la creación de PBX, dando solución a la conexión de 3 PBX para la organización, cubriendo las necesidades de la empresa y que todos los empleados estén en comunicación.

3.1.1. Requerimientos necesarios para virtualizar un servidor

Se requiere de un servidor con especificaciones adecuadas para el uso que se va a destinar. El software que utilizaremos es Proxmox Virtual Environment.

3.1.2. Ventajas de utilizar proxmox

Características de Proxmox:

- Funciones avanzadas para la administración.
- Bajos requerimientos para la instalación y uso del software (se necesita de poco procesador, memoria RAM, almacenamiento).
- Se necesita de muy poco hardware (se puede instalar en un servidor o hasta una PC).
- Gratuito.
- Compatible con otros entornos de virtualización.
- Compatible con varios sistemas operativos.

Ventajas de crear un entorno virtualización de servidores:

- Reducir costos operativos y de mantenimiento.
- Utilizar y aprovechar los componentes del servidor.
- Alta disponibilidad de aplicaciones.
- Menor consumo de energía.
- Aumento en productividad y administración de las operaciones.
- Mejor capacidad de respuesta.
- Operación continua, sin pérdida del servicio.
- Mayor seguridad de la información.
- Utilizar diferentes sistemas operativos.
- Implementación rápida y segura.
- Puede ser aplicada para diferentes finalidades (Redes, Datos, Aplicaciones, etc.).

3.1.3. Servicio de voz IP asterisk

- Es un software de uso libre que proporciona servicio de voz sobre IP.
 - Provee servicios similares a una central telefónica.
 - Tiene un costo para funciones especiales o algún soporte por parte de los desarrolladores de asterisk, pero en general el software es gratuito.
 - Tiene las funciones que toda empresa pequeña, mediana o grande necesita para una comunicación entre miembros de la empresa.
 - Funciones como: conferencias, IVR, grabadora de llamadas, grupos de timbrado, entre otras funcionalidades.
 - Se puede implementar tanto para equipos físicos de telefonía, y con software softphone para utilizar en PC o dispositivos móviles.
-

3.2. Materiales y componentes

Se desarrolló en dos lugares, el primero en la empresa Grupo Salinas en la que laboro y el segundo en el laboratorio de redes de computadoras del plantel SLT. En las siguientes secciones se describen las características.

3.2.1. Implementación en la empresa Grupo Salinas

Primero que nada, nuestra infraestructura consta de 2 sites en las instalaciones de la empresa GS, el primer site tendrá 2 servidores físicos, los cuales se usarán como equipos principales donde vamos a proporcionar el servicio de los 4 PBX.

El site secundario, está localizado en otro lugar y cuenta con diferente suministro de electricidad, es el site con equipos de respaldo para garantizar la redundancia en los servicios.

En el site secundario se contará con otros 2 servidores físicos instalados, estarán configurados con los 4 PBX y proveerán el servicio de voz como respaldo, en caso de que los principales lleguen a fallar.

3.2.2. Implementación en el laboratorio de redes de computadoras del plantel SLT

Para las prácticas se realizó la implementación y pruebas en el LRC del plantel SLT, con servidores para crear nuestra red clúster de ProxmoxVE.

Para los equipos físicos que serán los servidores, se requieren de las siguientes especificaciones:

- Procesador
- Memoria RAM
- Tarjeta gráfica
- Infraestructura de red
- Software de virtualización
- Software asterisk

Para los servidores físicos del LRC se tienen las siguientes características como se muestran en la tabla 3.1:

Computadora Ensamblada LUFAC	
Procesador	INTEL Core i7-4790 4-CORE HT 3.6 GHz 8 MB LGA1150
MotherBoard	ASUS H97-PLUS, ATX From Factor 12 inch x 8.6 inch
Memoria Ram	16 GB DIMM DDR3 KINGSTON
Disco Duro	1 TB 3.5 SATA 6.0 7200 RPM 64 MB
Tarjeta de Vídeo	ASUS NVIDIA GT 730, "GB DDR3, 96 Núcleos, 64 BIT

Cuadro 3.1: Componentes de los CPU del LRC.

La siguiente figura 3.1 son los equipos con las características mencionadas anteriormente, dentro de los 8 CPU's tenemos un clúster configurado con Proxmox.

Utilizando el clúster nos da la ventaja de poder sumar los recursos con los que cuenta cada CPU, es decir, si cada CPU cuenta con 1 TB de almacenamiento y una RAM de 16 GB, al agrupar 8 CPU's logramos tener disponibles 8 TB de almacenamiento con 128 GB en memoria RAM, todo distribuido para poder generar tantos servicios como recursos tengamos disponibles dentro del clúster.



Figura 3.1: Servidores físicos que utilizamos en el LRC del plantel SLT.

Con estos equipos se configuran 3 PBX, uno por cada servidor físico, para que funcionen de manera óptima, sin exigir demasiado a un solo servidor físico y que pueda aprovechar los recursos del clúster eficientemente.

3.3. Procedimiento para el desarrollo del proyecto

En cada uno de los servidores físicos se instala el sistema Proxmox Virtual Environment con la versión 7.3.

La liga oficial del sistema Proxmox es la siguiente:

- <https://www.proxmox.com/en/downloads/category/iso-images-pve>

La versión es: 7.3, como vemos en la figura 3.2, accediendo a la liga anteriormente mencionada.



Figura 3.2: Página oficial de ProxmoxVE para descargar el instalador.

3.3.1. Creación de la memoria USB booteable

Se utiliza una memoria USB booteable creada con el archivo .iso de Proxmox, que previamente se descarga de la página oficial.

Para crear la memoria USB booteable se utiliza el software Balena Etcher, esto para poder instalar Proxmox desde el arranque del servidor.

Balena Etcher es un software de uso libre y código abierto, utilizado para escribir archivos de imagen con extensión .iso ó .img, está desarrollado por Balena y con licencia de Apache 2.0.[28]

Este software se descarga desde la siguiente liga:

- <https://www.balena.io/etcher/>
- Se descargó la versión para Linux x64 (AppImage).

Se utiliza una memoria USB con capacidad de 8 GB como mínimo, se carga el archivo con extensión .iso del programa Proxmox v7.3.1 en la memoria USB, el mismo programa al conectar el USB a la PC reconoce el dispositivo de almacenamiento, se selecciona la memoria USB para comenzar a grabar la imagen de Proxmox, tal como se muestra en la figura 3.3.

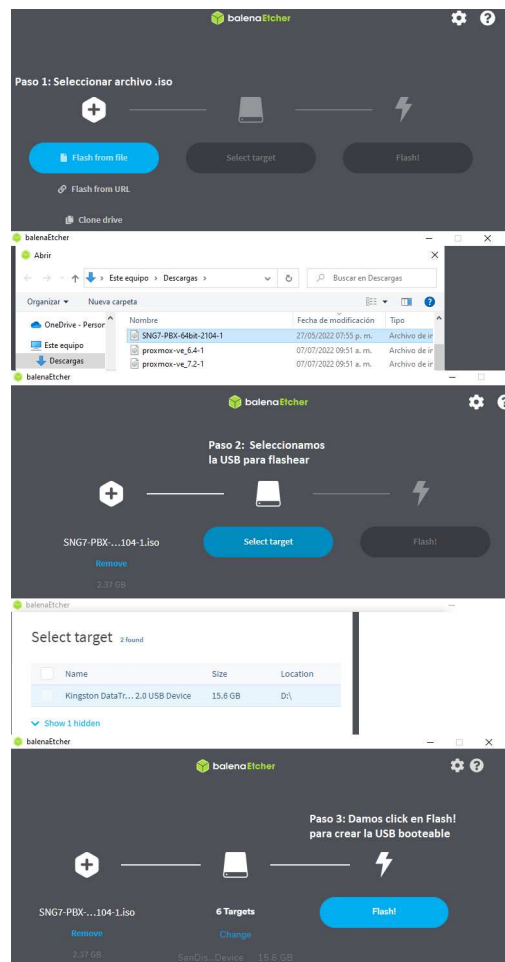


Figura 3.3: Procedimiento para crear la USB booteable.

3.3.2. Instalación de proxmox

Posterior a la creación de la USB booteable se sigue con la instalación de ProxmoxVE dentro del primer servidor físico.

Es importante introducir la USB booteable mientras el servidor se reinicia, para comenzar la instalación de Proxmox v7.3, en la figura 3.4 se observa como comienza el arranque de la instalación conectando la USB booteable a un servidor.



Figura 3.4: Instalación de Proxmox en el primer servidor físico.

Comenzando nuestro proceso de instalación haremos lo siguiente:

La información que el proceso solicita como primer paso es identificar la interfaz de red donde se encuentra conectado el servidor, siempre selecciona el puerto 0 ó 1, solo hay que verificar que sea el mismo puerto donde tenemos la conexión a la red, ese dato se muestra en la figura 3.5. Como sugerencia utilizar el puerto de red eth0 que será la principal interfaz de red.

El siguiente dato es el nombre de host que llevará el hipervisor, será donde se visualizarán las máquinas virtuales, tal como se muestra en la figura 3.5.

Después se configuran los datos de red que tendrá el hipervisor, estos datos son por donde

se accederá a la interfaz web de Proxmox, también se muestran en la figura 3.5.

Management Network Configuration

Please verify the displayed network configuration. You will need a valid network configuration to access the management interface after installing.

After you have finished, press the Next button. You will be shown a list of the options that you chose during the previous steps.

- **IP address (CIDR):** Set the main IP address and netmask for your server in CIDR notation.
- **Gateway:** IP address of your gateway or firewall.
- **DNS Server:** IP address of your DNS server.

Management Interface:	enp0s3 - 08:00:27:2b:58:0b (e1000) ▼	
Hostname (FQDN):	Proxmox00.com	
IP Address (CIDR):	192.168.200.41	/ 24
Gateway:	192.168.200.254	
DNS Server:	172.17.102.14	

Figura 3.5: Datos para configurar el primer hipervisor.

Comenzará la instalación del hipervisor proxmox que demora algunos minutos, posterior se reiniciará el servidor para volver a cargar configuraciones y ya podrá iniciar el hipervisor.

La nota al final de la instalación indica que se puede acceder con la dirección IP configurada y el puerto designado 8006.

Esto significa que para la conexión con el hipervisor, se esta en comunicación sobre la capa 4 de transporte, significa que se tiene la interacción y comunicación entre dos hosts, sobre esta capa se utiliza mayormente los protocolos TCP y UDP. Para el acceso al hipervisor se hace con el puerto 8006, utilizando el protocolo TCP, que esta orientado a conexión, es decir, para garantizar la transmisión de los datos debe existir un aviso que los paquetes se entregaron correctamente, esos datos también se pueden mandar de manera bidireccional.

3.3.3. Crear un clúster con proxmox

En la figura 3.6 se observa un diagrama de red de como esta planteado crear un clúster con proxmox, para los servicios de voz sobre IP, se tiene del lado izquierdo los servidores con proxmox representando un clúster, después las VM con el sistema asterisk, son los PBX para finalmente conectar los equipos telefónicos para el servicio de VoIP.

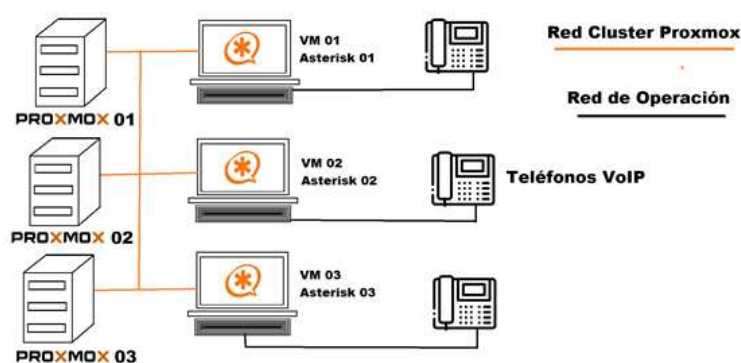


Figura 3.6: Representación del clúster con los PBX virtuales.

En la figura 3.7 tenemos los servidores por separado, cada uno con el servicio Proxmox instalado, se observa que cada equipo tiene su configuración de red y desde una PC de administración se puede acceder a cada uno de ellos desde su propia dirección IP.

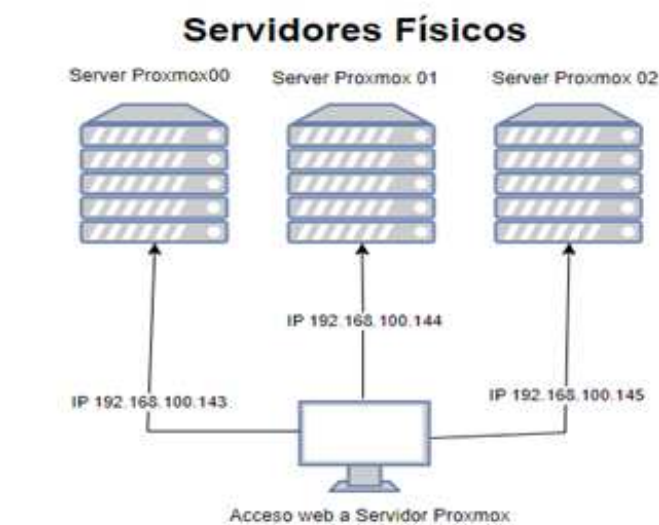


Figura 3.7: Servidores físicos y su configuración de red.

Ahora bien, para poder realizar mejoras en cuanto al servicio, tener una mejor gestión y respuesta en caso de fallas, con las ventajas que nos ofrece la virtualización se implementarán algunas características adicionales, para tener un servicio con alta redundancia y prevenir la caída del servicio de voz.

Proxmox ofrece algunas características para implementar, una de ellas es poder realizar un clúster entre los servidores físicos, de esta forma accediendo al hipervisor se visualiza,

opera y modifica cualquiera de los PBX virtuales sin la necesidad de acceder a cada uno por su propia dirección IP.

Un ejemplo sobre los servicios a implementar sería el siguiente:

En el site principal se tienen 2 servidores físicos y en cada uno de ellos se encontrarán 2 PBX virtuales, es decir, existen 4 PBX operando. Ahora si se realiza un cluster entre ambos hipervisores de los dos servidores físicos, accediendo a cualquier servidor físico se pueden administrar los 4 PBX.

Un gran beneficio de tener la configuración de esa manera, resulta que si por algún motivo llegase a fallar un servidor físico, los PBX virtuales que habitan en el servidor que falla, automáticamente migrarían al otro servidor, de esa manera los servicios seguirán activos, para los usuarios la falla o el cambio sería transparente, sin afectación. Garantiza tener la operación de servicios funcionando 24/7 sin interrupciones.

El primer paso a realizar será la creación del clúster en un hipervisor, previamente ya se cuenta con cada hipervisor configurado con su dirección IP propia. Accediendo al hipervisor utilizado como el principal, posterior los demás hipervisores se deberán registrar al principal compartiendo una llave que permitirá unirse al clúster.

En la figura 3.8 hay 3 CPU's en clúster, cada uno con su propio nombre de host y su configuración de red, la ventaja que al estar en clúster pueden distribuir la carga de los servicios alojados y cuenta con todos los recursos de manera distribuida, administrando todos los equipos dentro del clúster utilizando el hipervisor de Proxmox.

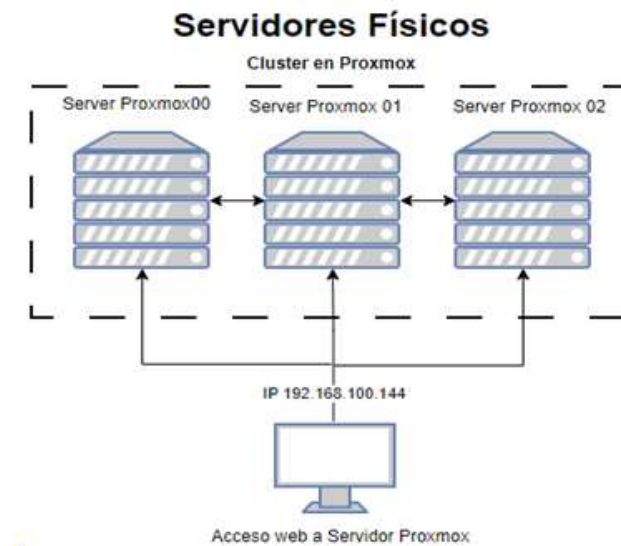


Figura 3.8: Clúster entre hipervisores creado con Proxmox.

Antes de utilizar los equipos en operación se realizó un clúster desde un equipo de pruebas,

antes de implementarlo en los equipos del LRC del plantel SLT.

En el procedimiento se utilizó Virtualbox como prueba para crear un clúster, pero en general es el mismo procedimiento para crear un clúster con Proxmox.

Para crear un clúster de prueba utilizando Virtualbox se utiliza VNC viewer para lograr una conexión remota al LRC, específicamente tener acceso a una computadora dentro del LRC, como estar trabajando presencialmente en el LRC.

En la figura 3.9 se muestran los archivos necesarios para establecer la conexión remota con VNC viewer desde un equipo remoto.

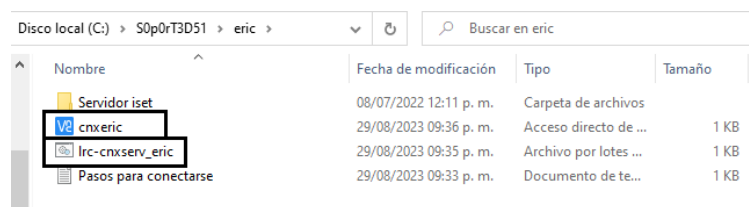


Figura 3.9: Archivos para la conexión remota por VNC viewer.

VNC Viewer sirve para lograr tener una conexión cliente-servidor, logrando visualizar la pantalla del servidor o computadora a la que se requiere conectar, es de gran ayuda para acceder a cualquier máquina desde diferentes equipos remotos. Para el presente proyecto se usó una laptop con sistema operativo Windows y al equipo que se conecta es un sistema operativo Linux, exactamente Ubuntu 20.04.

En la figura 3.10 al ejecutar el archivo “lrc-cnxserv_eric” (ver figura 3.9), este archivo es un ejecutable que automatiza la conexión por VNC viewer generando un túnel con el protocolo SSH, para después ejecutar el programa VNC viewer donde se accede con las credenciales correspondientes.

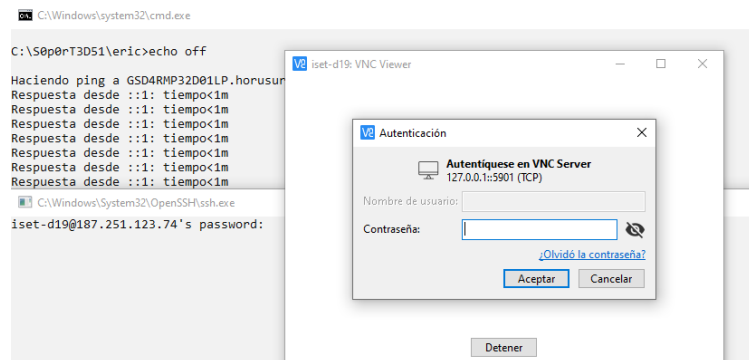


Figura 3.10: Ejecutable para ingresar por VNC viewer.

Teniendo la conexión remota a través de VNC viewer como se muestra en la figura 3.11, se utiliza Virtualbox para crear 3 VM e instalar Proxmox en cada una de ellas.

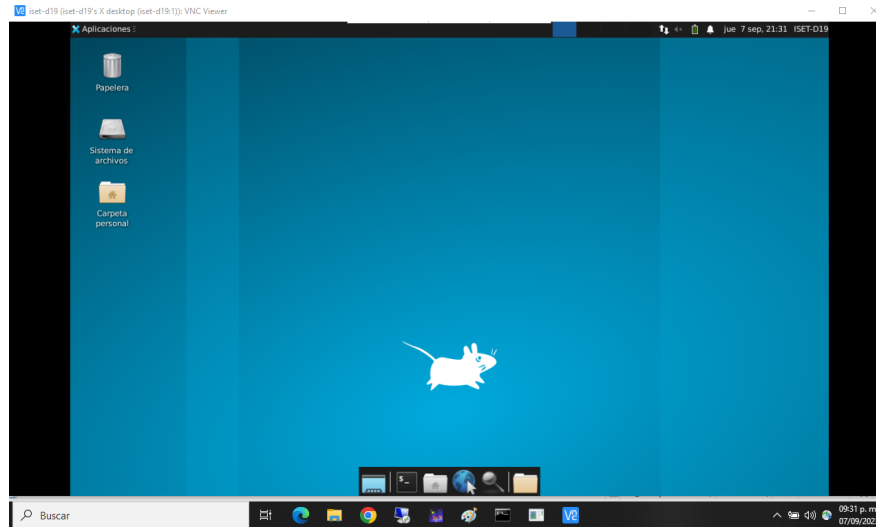


Figura 3.11: Conexión por VNC viewer a la PC D-19 del LRC.

Ahora bien, para instalar Proxmox en cada una de las VM de Virtualbox, se toman en cuenta los recursos mínimos necesarios para que puedan funcionar correctamente los hipervisores para crear el clúster.

Los siguientes datos son los requisitos mínimos necesarios para instalar el servicio ProxmoxVE, datos obtenidos del sitio oficial.

- **Procesador de 64 Bits, con múltiples núcleos.**
- **Tarjeta madre con soporte para virtualización.**
- **2 GB en RAM, solo para servicios ProxmoxVE.**
- **Disco duro (SAS, SSD).**
- **Tarjetas de red Gbit (Cantidad según la configuración de almacenamiento externo y/o clúster).**

En la figura 3.12 hay 3 VM, cada una de ellas con Proxmox instalado, se visualizan los componentes que tienen las VM, para las pruebas se asignan 4 GB en memoria RAM, un procesador de 4 núcleos, almacenamiento de 32 GB y una interfaz de red virtual.



Figura 3.12: Desde Virtualbox hay 3 VM con Proxmox para realizar un clúster.

Dato importante, antes de iniciar las VM con Proxmox, en las configuraciones de cada VM se habilita la virtualización anidada, como se muestra en la figura 3.13. La virtualización anidada sirve para crear un entorno de virtualización dentro de una VM, la finalidad es probar el clúster con Proxmox.

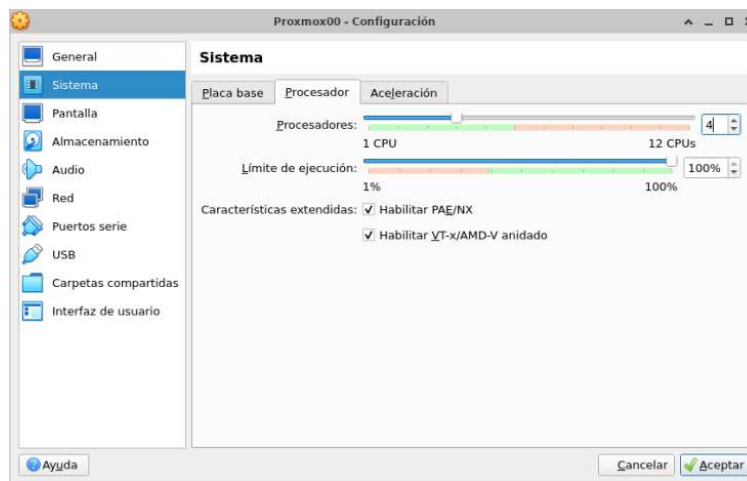


Figura 3.13: Configuraciones donde se habilita la virtualización anidada.

Teniendo las VM de Proxmox y el acceso web a cada una de ellas con la dirección IP que se les asignó, se selecciona una VM como la principal para crear el clúster, ingresando al CLI de la VM nombrada Proxmox00.

Dentro del CLI se crea el clúster con los siguientes comandos:

- **pvecm create proxmox-cluster --link0 192.168.200.41**

Siendo 192.168.200.41 la dirección IP del hipervisor seleccionado para el cluster.

Posterior a crear el clúster, el siguiente comando es para verificar el estado del clúster con el servicio corosync:

- **systemctl status corosync**

En la figura 3.14 muestra el resultado de ejecutar los dos comandos anteriormente mencionados.

```

iset-d19@iset-d19: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@Proxmox00:~# pvecm create proxmox-cluster --link0 192.168.200.41
Corosync Cluster Engine Authentication key generator.
Gathering 2048 bits for key from /dev/urandom.
Writing corosync key to /etc/corosync/authkey.
Writing corosync config to /etc/pve/corosync.conf
Restart corosync and cluster filesystem
root@Proxmox00:~# systemctl status corosync
● corosync.service - Corosync Cluster Engine
   Loaded: loaded (/lib/systemd/system/corosync.service; enabled; preset: ena
   Active: active (running) since Mon 2023-09-04 22:10:20 CST; 17s ago
     Docs: man:corosync
           man:corosync.conf
           man:corosync_overview
   Main PID: 2122 (corosync)
    Tasks: 9 (limit: 4607)
   Memory: 129.5M
      CPU: 250ms
   CGroup: /system.slice/corosync.service
           └─2122 /usr/sbin/corosync -f
  
```

Figura 3.14: CLI del hipervisor donde se creó el clúster.

A continuación se agregan los otros dos hipervisores dentro del clúster, entrando al CLI desde Proxmox01 y Proxmox02 se ejecuta el siguiente comando:

- **pvecm add “192.168.200.41” –link0 “192.168.200.42”**

En el comando anterior primero se ingresa la dirección IP del hipervisor donde está creado el clúster (Proxmox00), después la dirección IP del equipo que se desea agregar al clúster, es decir, el hipervisor Proxmox01, ambas direcciones IP sin comillas como se muestra en la figura 3.15.

Al ejecutar el comando pide la contraseña del clúster (es la contraseña de administrador, es decir, la contraseña root) para agregar el hipervisor al clúster, ingresando los datos se establece la conexión al clúster, como se muestra en la figura 3.15.

Aparecerá una advertencia para establecer la conexión, solo pide escribir “yes” ó “no”. Finalmente se agrega el hipervisor al clúster estableciendo una conexión entre hosts, como se muestra en la figura 3.15.



```

iset-d19@iset-d19: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@Proxmox01:~# pvecm add 192.168.200.41 --link0 192.168.200.42
Please enter superuser (root) password for '192.168.200.41': *****
Establishing API connection with host '192.168.200.41'
The authenticity of host '192.168.200.41' can't be established.
X509 SHA256 key fingerprint is 9C:BC:C6:74:7A:40:AD:75:49:06:42:C9:B7:2C:9B:98:E4:FA:12:8B:07:69:7A
:06:A7:63:AD:14:D4:BA:F1.
Are you sure you want to continue connecting (yes/no)? yes
Login succeeded.
check cluster join API version
Request addition of this node
Join request OK, finishing setup locally
stopping pve-cluster service
backup old database to '/var/lib/pve-cluster/backup/config-1693887654.sql.gz'
waiting for quorum...OK
(re)generate node files
generate new node certificate
merge authorized SSH keys and known hosts
generated new node certificate, restart pveproxy and pvedaemon services
successfully added node 'Proxmox01' to cluster.
root@Proxmox01:~#

```

Figura 3.15: CLI del hipervisor agregado al cl ster.

Se realiza el mismo procedimiento con el segundo hipervisor.

■ **pvecm add 192.168.200.41--link0 192.168.200.43**

En el comando anterior es la direcci n IP del hipervisor donde est  creado el cl ster y la direcci n IP del hipervisor Proxmox02. Se ingresa la contrase a root del hipervisor que tiene el cl ster y damos enter.

Con los comandos previos se agrega el tercer hipervisor al cl ster como se muestra en la figura 3.16.



```

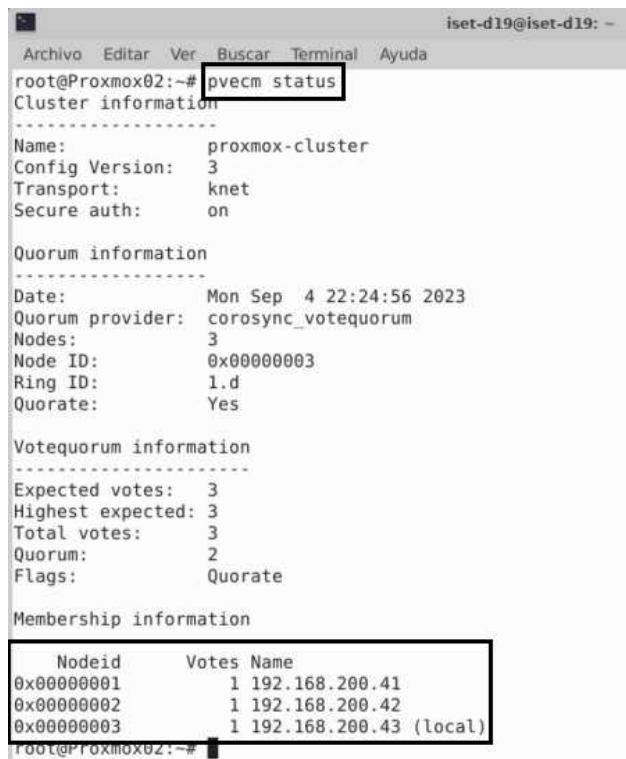
iset-d19@iset-d19: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@Proxmox02:~# pvecm add 192.168.200.41 --link0 192.168.200.43
Please enter superuser (root) password for '192.168.200.41': *****
Establishing API connection with host '192.168.200.41'
The authenticity of host '192.168.200.41' can't be established.
X509 SHA256 key fingerprint is 9C:BC:C6:74:7A:40:AD:75:49:06:42:C9:B7:2C:9B:98:E4:FA:12:8B:07:69:7A
:06:A7:63:AD:14:D4:BA:F1.
Are you sure you want to continue connecting (yes/no)? yes
Login succeeded.
check cluster join API version
Request addition of this node
Join request OK, finishing setup locally
stopping pve-cluster service
backup old database to '/var/lib/pve-cluster/backup/config-1693887800.sql.gz'
waiting for quorum...OK
(re)generate node files
generate new node certificate
merge authorized SSH keys and known hosts
generated new node certificate, restart pveproxy and pvedaemon services
successfully added node 'Proxmox02' to cluster.
root@Proxmox02:~#

```

Figura 3.16: CLI del segundo hipervisor agregado al cl ster.

Una vez agregados los hipervisores al cl ster, se puede visualizar el estado o n mero de equipos dentro del cl ster, ejecutando los siguientes comandos desde cualquier hipervisor que se encuentra dentro del cl ster.

- *pvecm status*



```
iset-d19@iset-d19: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@Proxmox02:~# pvecm status
Cluster information
-----
Name: proxmox-cluster
Config Version: 3
Transport: knet
Secure auth: on

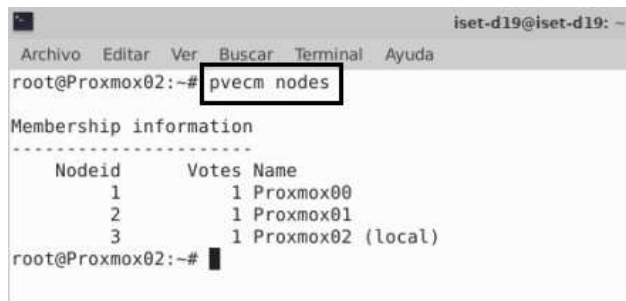
Quorum information
-----
Date: Mon Sep  4 22:24:56 2023
Quorum provider: corosync_votequorum
Nodes: 3
Node ID: 0x00000003
Ring ID: 1.d
Quorate: Yes

Votequorum information
-----
Expected votes: 3
Highest expected: 3
Total votes: 3
Quorum: 2
Flags: Quorate

Membership information
-----
Nodeid  Votes  Name
0x00000001  1  192.168.200.41
0x00000002  1  192.168.200.42
0x00000003  1  192.168.200.43 (local)
root@Proxmox02:~#
```

Figura 3.17: Muestra el estado del clúster.

- *pvecm nodes*



```
iset-d19@iset-d19: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@Proxmox02:~# pvecm nodes
Membership information
-----
Nodeid  Votes  Name
1       1      Proxmox00
2       1      Proxmox01
3       1      Proxmox02 (local)
root@Proxmox02:~#
```

Figura 3.18: Muestra el número de equipos que hay en el clúster.

Como se muestra en la figura 3.17 y 3.18, arroja información de los nodos y equipos dentro del clúster.

Finalmente accediendo a cualquier hipervisor se visualiza el clúster entre los equipos, de preferencia hay que utilizar la dirección IP del hipervisor donde está creado el clúster. La figura 3.19 es la interfaz web de un hipervisor para observar el arreglo clúster entre Proxmox00, Proxmox01 y Proxmox02.

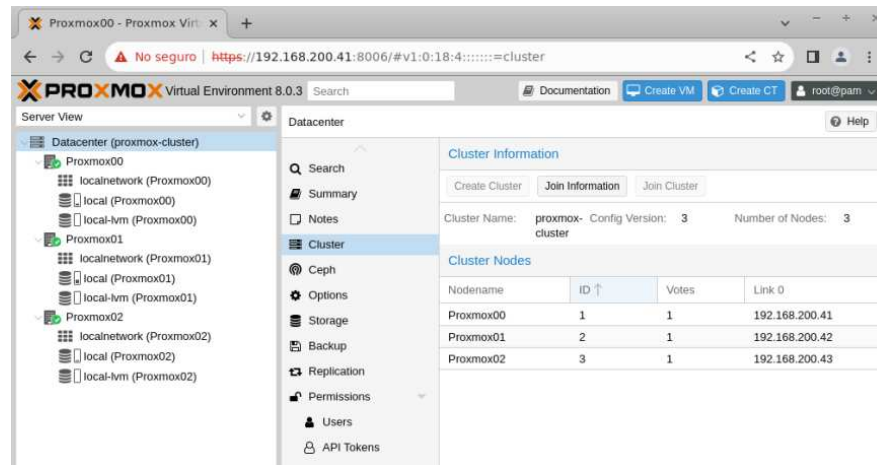


Figura 3.19: Clúster con los 3 hipervisores en Virtualbox.

Para la figura 3.20 muestra el arreglo clúster en el LRC del plantel SLT, un clúster de 8 equipos CPU's con datos sobre el estado de cada equipo, indica el porcentaje del uso de cada disco duro, memoria RAM, procesador y el tiempo de actividad para cada equipo, entre otros detalles.

The screenshot shows the Proxmox Virtual Environment 7.2.3 web interface. The left sidebar displays a tree view of the datacenter hierarchy, including Xexelo-L01 through Xexelo-L08. The main panel shows the 'Cluster Information' section, which includes a 'Cluster Nodes' table with the following data:

Type	Description	Disk usage...	Memory us...	CPU usage...	Uptime	Host CPU...	Host Mem...
lxc	100 (Ubuntu20)	2.6 %	3.5 %	0.0 % of 2...	18 days 20:4...	0.0 % of 8C...	0.5 %
node	Xexelo-L01	6.6 %	31.5 %	0.5 % of 8...	6 days 17:08...		
node	Xexelo-L02	3.8 %	21.3 %	0.4 % of 8...	24 days 02:4...		
node	Xexelo-L03	3.5 %	16.2 %	0.4 % of 8...	6 days 17:08...		
node	Xexelo-L04	6.8 %	40.6 %	0.6 % of 8...	6 days 17:08...		
node	Xexelo-L05	17.9 %	16.4 %	0.4 % of 8...	6 days 17:08...		
node	Xexelo-L06	5.8 %	19.8 %	0.5 % of 8...	19 days 19:2...		
node	Xexelo-L07	5.0 %	20.3 %	0.4 % of 8...	24 days 02:4...		
node	Xexelo-L08	3.5 %	16.6 %	0.4 % of 8...	6 days 17:08...		
qemu	103 (ipfsense)						
qemu	102 (Ub-MD)	0.0 %	28.7 %	0.5 % of 8...	4 days 17:57...	0.4 % of 8C...	22.3 %

Figura 3.20: Clúster con los equipos del LRC del plantel SLT.

Para la figura 3.21 seleccionando el “Datacenter (Xexelo-PVE)”, que es todo el arreglo clúster y seleccionamos la opción “summary”, muestra de manera detallada todo el conjunto

de recursos ocupados y disponibles en el clúster, el total de procesador, memoria RAM y almacenamiento, con detalles sobre el consumo por cada equipo CPU que habita dentro del clúster. Con el clúster existe una mejor administración y acceso más rápido a los servicios.



Figura 3.21: Porcentajes de todo el hardware en el clúster.

3.3.4. Creación de las VM para los PBX virtuales

El siguiente paso a realizar será la creación de las VM para tener los PBX virtuales. Cada una de las VM llevan la configuración necesaria para cubrir la cantidad de usuarios que se requiere para cada PBX virtual.

Los servidores virtuales tendrán las siguientes características:

- Cada VM cuenta con un disco duro virtual de 8 GB en almacenamiento (son 8 GB mínimo, dependiendo la cantidad de usuarios la capacidad del disco será mayor).
- 4 GB en memoria RAM. (dependiendo la cantidad de usuarios será mayor RAM).
- Interfaces de red virtuales para la operación de los servicios.

3.3.5. Instalación del software asterisk

Se descarga la versión más reciente del software asterisk, para crear un conmutador para servicios de voz por internet, en la siguiente liga se accede para descargar el software asterisk.

- <https://www.asterisk.org/downloads/>
- Versión más reciente de Asterisk: 20.0.1

Descargado en el servidor, se accede al almacenamiento local del hipervisor, para subir el archivo con extensión .iso del software asterisk, como se muestra en la figura 3.22, finalizando la carga en el almacenamiento del hipervisor, ya se puede utilizar como imagen de instalación en las VM.

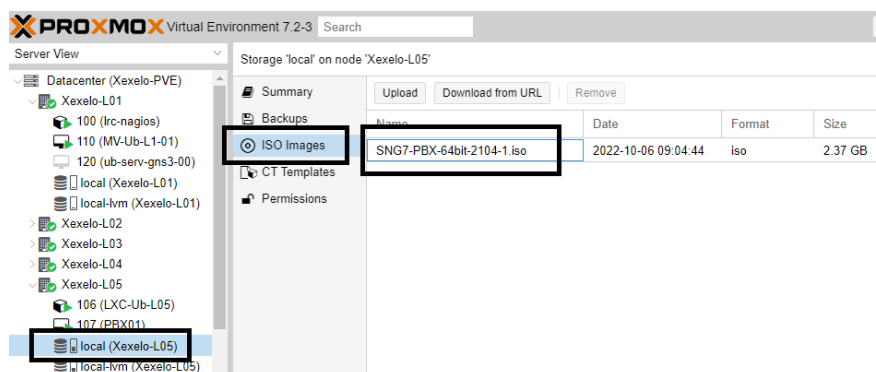


Figura 3.22: Archivo de instalación Asterisk en el hipervisor.

Al momento de crear la VM existe la opción de insertar una imagen de instalación y al arrancar la VM empezar con la instalación.

Con la primer VM se coloca el archivo con extensión .iso del software Asterisk para comenzar la instalación. Iniciando la consola de la VM, comienza el proceso de instalación, seguidas las instrucciones para configurar el PBX, se espera el proceso de instalación con los requerimientos básicos (configuración de red, zona horaria, disco de almacenamiento para la instalación, idioma, configuración de teclado, usuario root y contraseña, dirección IP, etc.).

Al finalizar la instalación el sistema en automático pedirá el reinicio, para comenzar a cargar las configuraciones y poder visualizar el inicio del PBX. Ya iniciado el sistema, nos dará la bienvenida por CLI de Asterisk, accediendo con el usuario y contraseña root se accede al sistema del PBX. Con el acceso se pueden hacer modificaciones avanzadas al PBX, como es un sistema de código abierto, se puede modificar a la necesidad para el servicio de los usuarios.

3.3.6. Copias de seguridad en proxmox

Ahora se utiliza otra de las funcionalidades que ofrece Proxmox y ahorrar tiempo de instalación, las copias de seguridad.

Utilizando el PBX virtual instalado y en operación, se realiza una copia de seguridad del mismo, es una forma para poder replicar la configuración básica del sistema Asterisk, para generar los demás PBX virtuales.

La copia de seguridad también es útil ya que si el PBX cuenta con una configuración específica y deseamos hacer cambios, en caso de presentar problemas en la configuración o algún error durante los cambios, se puede restaurar el PBX a los valores de la copia anteriormente creada.

Ya iniciando el primer PBX y con la configuración básica, desde el hipervisor en las opciones de la VM, se selecciona la opción “Backup” para crear una copia de seguridad del PBX.

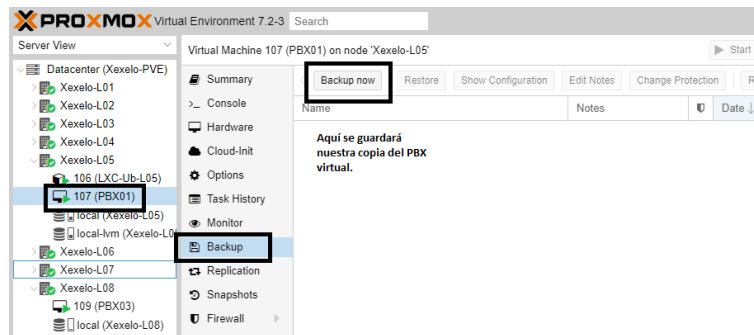


Figura 3.23: Ruta para guardar el archivo de respaldo del PBX.

Seleccionar “Backup now” como en la figura 3.23 para continuar el proceso de respaldo, existen diferentes formas de crear una copia de seguridad, incluso permite crear una copia mientras el PBX está funcionando sin interrumpir la operación, ese tipo de copia es con el modo “Snapshot”. Al elegir el modo “Suspend” se detiene el servicio de la VM y al finalizar la copia volverá a reactivarse, por último, con el modo “Stop” se apagará la VM para posteriormente realizar la copia. Para la copia del proyecto se decide apagar la VM como vemos en la figura 3.24.

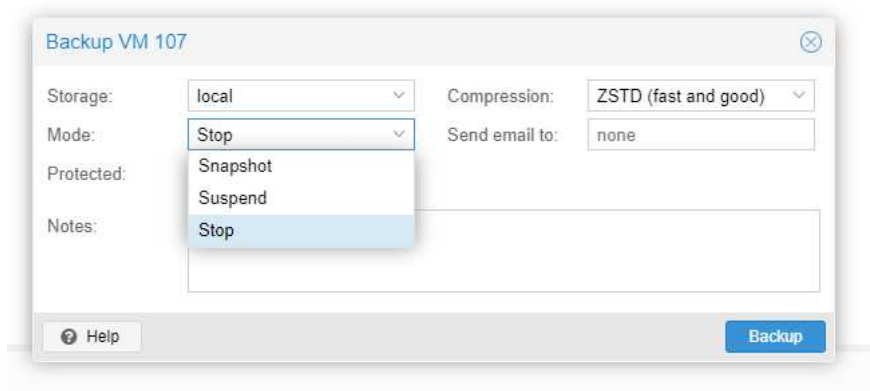


Figura 3.24: Se selecciona detener la VM.

Otra opción a elegir es la forma de compresión o rapidez para crear la copia de seguridad,

catalogado en tres opciones:

- LZO (fast)
- GZIP (good)
- ZSTD (fast and good)

Se utiliza el modo de compresión GZIP, como se muestra en la figura 3.25.

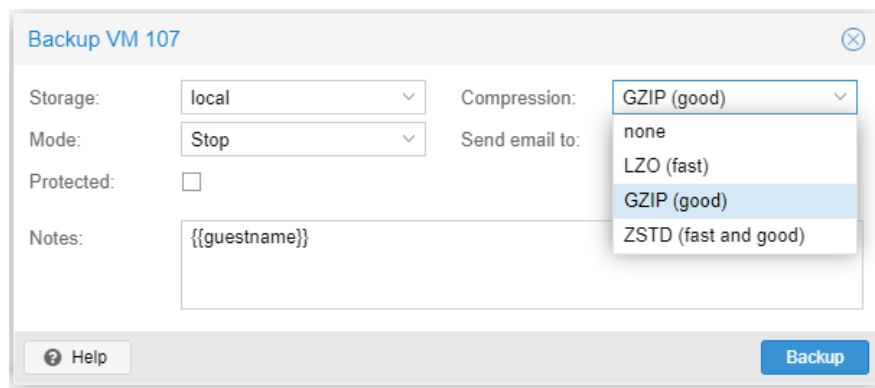


Figura 3.25: Tipo de compresión para la copia de seguridad.

Con las opciones GZIP y apagar el PBX virtual, se evita generar algún error que pueda afectar la VM, de momento el PBX solo tiene las configuraciones básicas para operar, sin extensiones, troncales u otros servicios aplicados.

Una vez seleccionado que hacer con la VM, mientras se realiza la copia y haber elegido el tipo de compresión, se muestra el proceso de la copia, ver figura 3.26. Al momento de llegar al 100 % se cierra la ventana y la copia de seguridad estará creada.

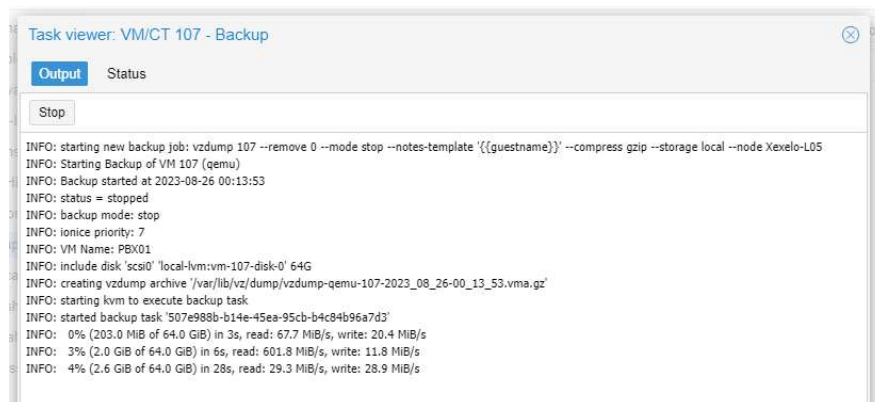


Figura 3.26: Proceso del backup.

Al finalizar la copia de seguridad, se van a replicar los otros 2 PBX restantes con el backup, se selecciona el disco local del hipervisor, después el apartado “Backup”. Dentro del apartado se encuentra la copia de seguridad, se selecciona el backup para después hacer click en la opción “Restaurar”, tal como se muestra en la figura 3.27.

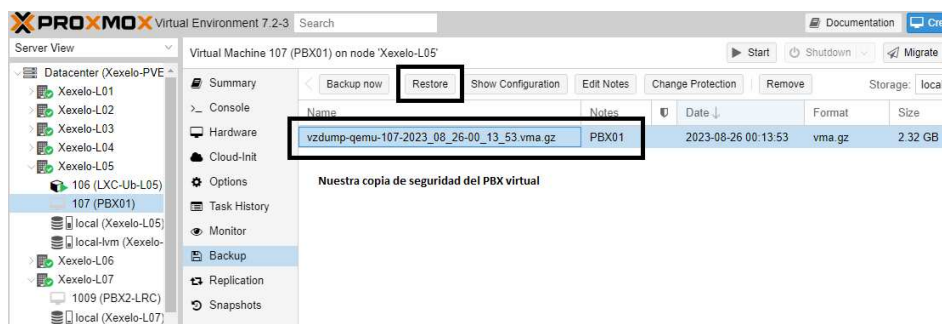


Figura 3.27: Archivo backup para restaurar.

Posterior nos muestra mas información para restaurar: Solicita seleccionar el almacenamiento local del hipervisor para restablecer la copia de seguridad, ver figura 3.28.

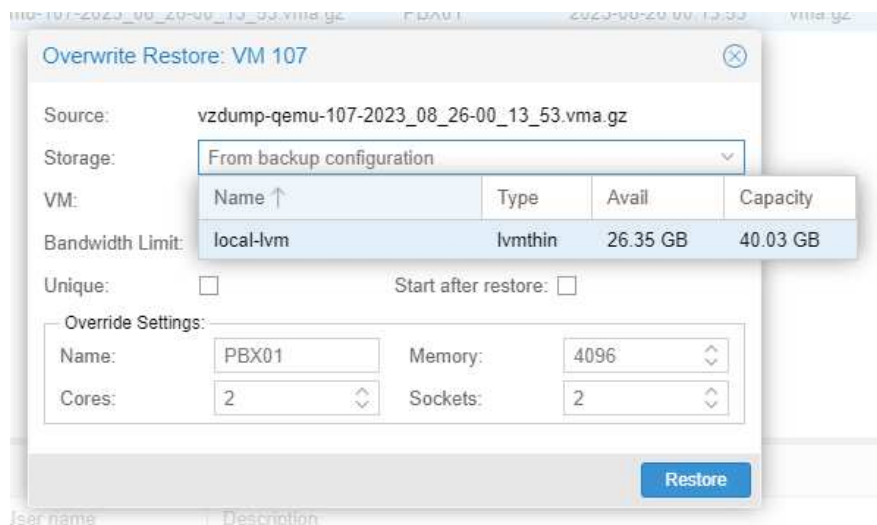


Figura 3.28: Opción de almacenamiento para restaurar el PBX.

Al hacerlo solicita un nombre a la VM, núcleos de procesador, memoria RAM para restaurarla, los datos son opcionales, finalmente comienza la restauración seleccionando el botón “Restore”, ver figura 3.29.

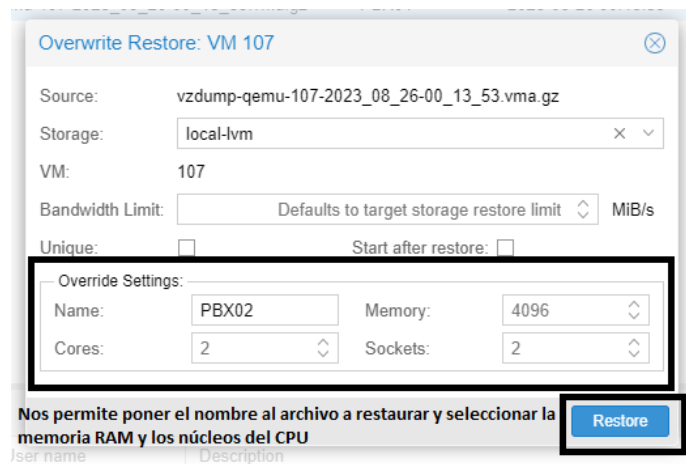


Figura 3.29: Ventana para restaurar la copia del PBX.

3.3.7. Creación de los PBX virtuales con las copias de seguridad

Al restaurar la VM se crea el segundo PBX virtual.

- **NOTA:** Antes de encender los PBX se deben realizar cambios al PBX, ya que la copia de seguridad tiene los mismos componentes virtuales que el primer PBX, creando una nueva interfaz de red virtual, la MAC Address cambiará, tal como se muestra en la figura 3.30. Para esto, en la opción “Hardware” de la VM, se selecciona la interfaz de red para eliminarla con la opción “Remove”.

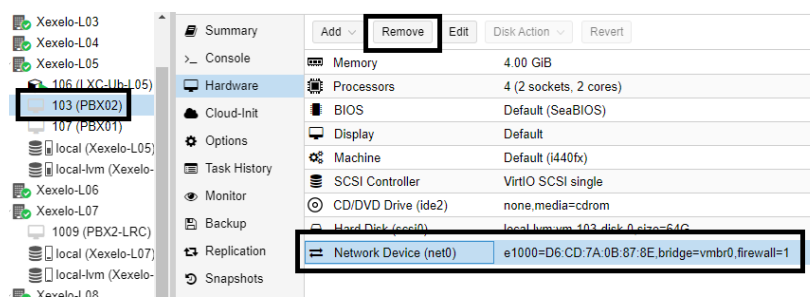


Figura 3.30: Interfaz de red con MAC duplicada.

En la figura 3.31, después de eliminar la interfaz de red, en el apartado “Add” elegir “Network Device”, con esta opción se crea una nueva interfaz de red para la VM y también cambiará la MAC Address.

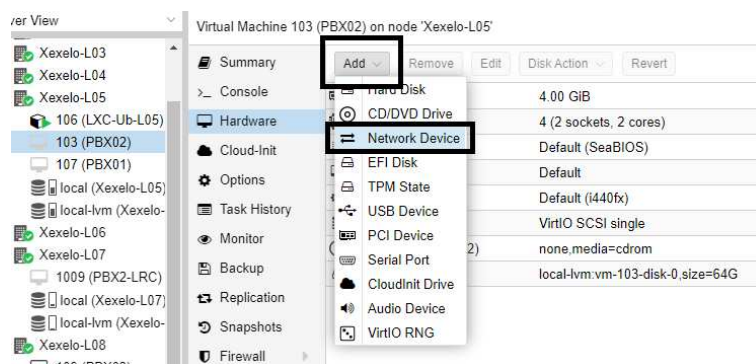


Figura 3.31: Después de borrar se agrega una nueva interfaz de red.

Finalmente se comprueba que las interfaces de red entre el PBX01 y PBX02 sean diferentes y cada una con su propia MAC address, tal como se muestra en la figura 3.32.



Figura 3.32: Comprobando diferencia de MAC address entre PBX01 y PBX02.

Realizando el cambio previo, se activa la VM del PBX para realizar la siguiente configuración dentro del CLI en el PBX.

- **NOTA:** por el momento es necesario apagar el primer PBX, ya que el PBX 1 y 2 cuentan con la misma dirección IP asignada, generando problemas al tener duplicada la dirección IP.

Ahora realizando el cambio en la configuración de la red del PBX 2, se configuran los datos de red para el PBX, se asigna una dirección IP disponible del mismo segmento, iniciando la VM con el PBX virtual, se accede con las credenciales del usuario root para hacer cambios con los siguientes comandos:

- **ip a**

El comando previo muestra la configuración de red del PBX 2, comprobando que tiene la misma dirección IP que el PBX 1, la figura 3.33 muestra la información de red del PBX al ejecutar el comando.

```

root@SLT-PBX:~
[root@SLT-PBX ~]# ip a
1: lo: <LOOPBACK,UP,LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:cd:7a:00:87:8e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.91/24 brd 192.168.1.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::d4cd:7aff:fe0b:878e/64 scope link
        valid_lft forever preferred_lft forever
[root@SLT-PBX ~]#
  
```

Figura 3.33: Dirección IP que tiene el PBX en el puerto eth0.

Con el siguiente comando accedamos a la ruta indicada para cambiar la configuración de red del PBX, `cd` “indicar la ruta donde se encuentra el archivo”. Ahora con el comando `cd` se elige la ruta para las configuraciones de red del PBX, esa ruta es la siguiente:

- **cd /etc/sysconfig/network-scripts**

El siguiente comando muestra la lista de los archivos en la carpeta donde nos encontramos. Situados dentro de las configuraciones de red con el comando `ls` se busca el archivo `ifcfg-eth0`.

- **ls**

Este último comando sirve para modificar los datos de cualquier archivo, “vim” seguido del nombre del archivo, el archivo es “`ifcfg-eth0`”.

- **vim ifcfg-eth0**

En la figura 3.34 muestra como se ejecutan los tres comandos anteriormente mencionados.

```

root@SLT-PBX:/etc/sysconfig/network-scripts
[root@SLT-PBX ~]# cd /etc/sysconfig/network-scripts/
[root@SLT-PBX network-scripts]#
[root@SLT-PBX network-scripts]# ls
ifcfg-eth0  ifdown-ippv6  ifdown-routes  ifup-bnep  ifup-isdn  ifup-routes  network-functions
ifcfg-lo    ifdown-ipv6  ifdown-sit     ifup-eth   ifup-plip  ifup-sit     network-functions-ipv6
ifdown     ifdown-isdn  ifdown-tunnel  ifup-hdlc  ifup-plusb ifup-tunnel
ifdown-bnep ifdown-post  ifup           ifup-ippv6 ifup-post  ifup-wireless
ifdown-eth  ifdown-ppp   ifup-aliases   ifup-ipv6  ifup-ppp   init.ipv6-global
[root@SLT-PBX network-scripts]#
[root@SLT-PBX network-scripts]# vim ifcfg-eth0
  
```

Figura 3.34: Pasos para modificar la interfaz de red eth0.

A continuación dentro de la configuración de red del puerto ifcfg-eth0, solo se edita el archivo cambiando la dirección IP por la que se asigna al PBX 2 y finalmente se guardan los cambios. En la figura 3.35 muestra las configuraciones del puerto de red, dentro se modifica la nueva dirección IP.

```

root@freepbx:/etc/sysconfig/network-scripts
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="none"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="eth0"
DEVICE="eth0"
ONBOOT="yes"
IPADDR="192.168.1.91"
PREFIX="24"
GATEWAY="192.168.1.254"
IPV6_PRIVACY="no"
~
~

```



Cambio de dirección IP

Figura 3.35: Archivo de la interfaz de red del PBX 2.

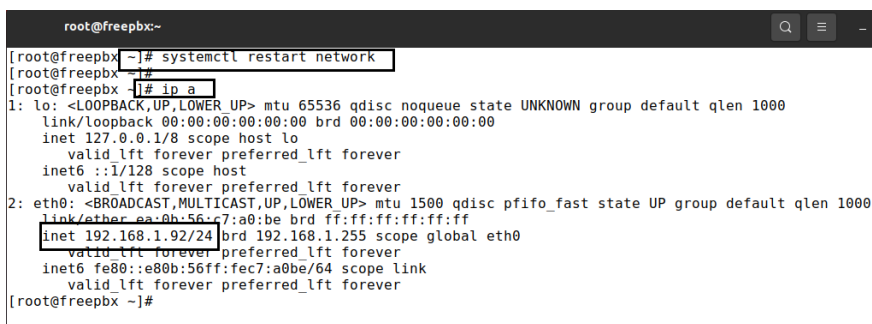
- **type=ethernet:** Indica el tipo de interfaz a configurar.
- **proxy_method=none:** Necesario para configurar un proxy.
- **browser_only=no:** Solo sirve en caso que se tenga configurado un proxy.
- **bootproto=none:** Indica el protocolo que queremos utilizar al iniciar la interfaz.
- **defroute=yes:** Define si la interfaz es la ruta por defecto.
- **ipv4_failure_fatal=no:** La interfaz no se deshabilita si la configuración falla.
- **ipv6init=yes:** Habilita la configuración IPv6 en esta interfaz, se agregan datos de IPv6 si los hay.
- **ipv6_autoconf=yes:** Configuración automática de la red.
- **ipv6_defroute=yes:** Define la ruta por defecto para IPv6.
- **ipv6_failure_fatal=no:** La interfaz no se deshabilita si la configuración falla en IPv6.
- **ipv6_addr_gen_mode=stable-privacy:** Genera una dirección IP para la conexión estable.

- **name=eth0:** Establece el nombre del perfil de conexión.
- **device=eth0:** Establece la interfaz de red.
- **onboot=yes:** Define iniciar la conexión al arrancar el sistema.
- **ipaddr=XXX:** Dirección IP fija.
- **prefix=XXX:** Máscara de red.
- **gateway=XXX:** Gateway de la configuración de red.
- **ipv6_privacy=no:** Establece si la conexión es segura.

Por último y para aplicar los cambios a la configuración de red, se aplica un reinicio a la interfaz de red del PBX para que pueda tomar los nuevos cambios con el siguiente comando:

- **systemctl restart network**

En la figura 3.36 aplicado el comando anterior para reiniciar la interfaz de red, se vuelve a comprobar si ha cambiado la dirección IP del PBX con el comando “ip a”.



```
root@freepbx:~  
[root@freepbx ~]# systemctl restart network  
[root@freepbx ~]#  
[root@freepbx ~]# ip a  
1: lo: <LOOPBACK,UP,LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000  
    link/ether ea:0b:56:c7:a0:be brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.92/24 brd 192.168.1.255 scope global eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::e80b:56ff:fec7:a0be/64 scope link  
        valid_lft forever preferred_lft forever  
[root@freepbx ~]#
```

Figura 3.36: Se comprueba la nueva dirección IP configurada al PBX.

Ahora se realiza el cambio de hostname del PBX para poder identificar cada uno de los diferentes PBX, como fue una instalación principal todos llevan el nombre de freepbx, a continuación para el cambio de hostname se ejecuta el siguiente comando:

- **hostnamectl set-hostname SLT-PBX**

la frase “SLT-PBX” será el nombre de host del PBX, tal como vemos en la figura 3.37. Se debe finalizar la sesión para validar el cambio de nombre en el CLI del PBX.

```

root@cuautepecpbx:~
[root@cuautepecpbx ~]# hostnamectl set-hostname CuautepecPBX
[root@cuautepecpbx ~]#
[root@cuautepecpbx ~]# hostnamectl
  Static hostname: cuautepecpbx
    Pretty hostname: CuautepecPBX
      Icon name: computer-vm
    Chassis: vm
      Machine ID: cb0dd509c35c40fa92d5c69c363a1d33
      Boot ID: 8e5a68cf346844808e8a04e78a68c276
  Virtualization: kvm
  Operating System: Sangoma Linux 7 (Core)
    CPE OS Name: cpe:/o:sangoma:sng:7::server:utf8
      Kernel: Linux 3.10.0-1127.19.1.el7.x86_64
  Architecture: x86-64
[root@cuautepecpbx ~]#

```

Figura 3.37: Resultado del cambio de hostname.

Se realizaron los cambios necesarios en el PBX al restaurar la copia de seguridad, para que funcionen correctamente y que cada uno tenga su respectiva interfaz de red, dirección IP asignada y nombre de host. En la figura 3.38 ingresando al CLI de cada PBX y al iniciar sesión con el usuario root nos muestra la interfaz de red configurada, la MAC address de la interfaz y la dirección IP asignada al PBX, comprobando como los dos PBX cuentan con su propia MAC address y dirección IP.

Por último se repite el mismo procedimiento descrito anteriormente para restaurar otra VM y cambiar la interfaz de red, dirección IP y hostname para contar con los 3 PBX del proyecto.

```

root@cuautepecpbx:~
Last login: Thu Aug 31 01:37:16 2023 from 192.168.1.102
FreePBX
NOTICE! You have 3 notifications! Please log into the UI to see them!
Current Network Configuration
+-----+-----+-----+
| Interface | MAC Address | IP Addresses |
+-----+-----+-----+
| eth0      | EA:0B:56:C7:A0:BE | 192.168.1.92 |
|           |               | fe80::e80b:56ff:fec7:a0be |
+-----+-----+-----+

root@SLT-PBX:~
Last login: Thu Aug 31 01:28:30 2023 from 192.168.1.102
FreePBX
NOTICE! You have 3 notifications! Please log into the UI to see them!
Current Network Configuration
+-----+-----+-----+
| Interface | MAC Address | IP Addresses |
+-----+-----+-----+
| eth0      | D6:CD:7A:0B:07:8E | 192.168.1.91 |
|           |               | fe80::d4cd:7aff:fe0b:878e |
+-----+-----+-----+

```

Figura 3.38: PBX 1 y PBX 2 con su propia dirección IP y MAC.

Validando el acceso desde la GUI por medio de freePBX en cada VM, se crean algunas extensiones de prueba para verificar que se tiene comunicación.

Primero realizando pruebas individuales con cada PBX, para comprobar que se tiene comunicación interna con el mismo PBX, usando 2 extensiones telefónicas, cada una conectada a su nodo de red que van al switch situado en el LRC, dentro de la misma red donde están conectados los servidores físicos.

Otro paso importante por mencionar es que para la operación de los servicios de telefonía, se configuran dos interfaces de red que cada servidor físico tiene, cada PBX virtual tiene dos interfaces de red virtuales, esto con la finalidad de ofrecer el servicio de voz a través de una interfaz que llamaremos “red de operación”, la otra interfaz de red la llamaremos “red de administración”, esta última interfaz de red se utiliza para poder acceder a nuestro PBX virtual, permitiendo administrar y configurar cuando sea necesario.

Al momento de acceder desde la GUI con la dirección IP del PBX virtual, dará una bienvenida, donde con perfil de administrador del servicio PBX, al ser el primer ingreso, pide crear el usuario y contraseña de administrador, como podemos verlo en la figura 3.39, después de una breve configuración de la interfaz, ya se cuenta con acceso al PBX. Asterisk proporciona una gran variedad de configuraciones necesarias para el entorno empresarial, dependiendo de las necesidades de cada organización, como primer paso, se crean extensiones de prueba para establecer la comunicación dentro del PBX.

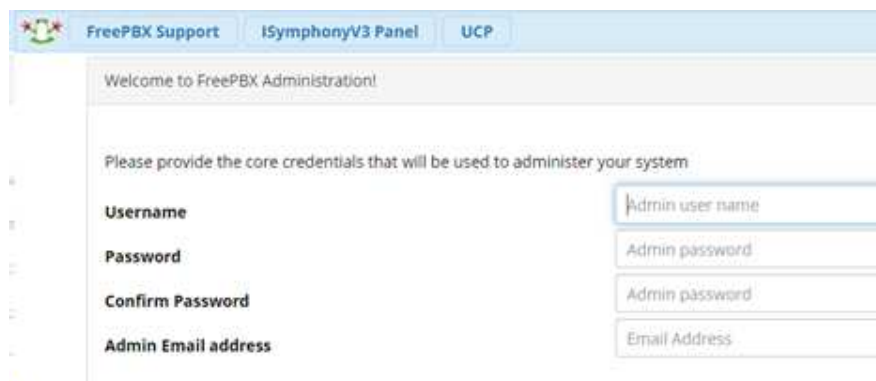
The image shows a web browser window displaying the FreePBX Administration interface. At the top, there are three tabs: 'FreePBX Support', 'iSymphonyV3 Panel', and 'UCP'. Below the tabs, a message reads 'Welcome to FreePBX Administration!'. A prompt asks the user to 'Please provide the core credentials that will be used to administer your system'. There are four input fields: 'Username' with the placeholder 'Admin user name', 'Password' with 'Admin password', 'Confirm Password' with 'Admin password', and 'Admin Email address' with 'Email Address'.

Figura 3.39: Interfaz web FreePBX, crear una cuenta de administrador.

Una vez configuradas como mínimo dos extensiones para una prueba de comunicación, es importante en la configuración de los teléfonos poner el SIP server, que se trata de la “dirección IP de operación”, para que los equipos telefónicos se puedan registrar y operar a través de la dirección IP.

Para esta implementación de servicios VoIP, se utilizan 3 teléfonos Yealink T42G con conexión por cable de red, ya que el switch para las conexiones cuenta con puertos PoE, ya no es necesario que los teléfonos cuenten con un eliminador conectado a la corriente eléctrica, la misma configuración del puerto ofrece energía a través del cable de red.

Ya conectados los teléfonos, el switch a su vez está conectado a la red principal del LRC

del plantel SLT, al momento de encender, con la dirección IP que toman los teléfonos, se accede a la interfaz de configuración de cada uno. En la figura 3.40 muestra el acceso web en uno de los teléfonos, accediendo al apartado de cuenta donde se registra la extensión creada en el PBX. Ingresando el número de extensión, la contraseña de registro y muy importante el servidor SIP al cual se va a registrar, es la dirección IP del PBX, finalmente se aplican los cambios, habilitando la extensión con el registro.

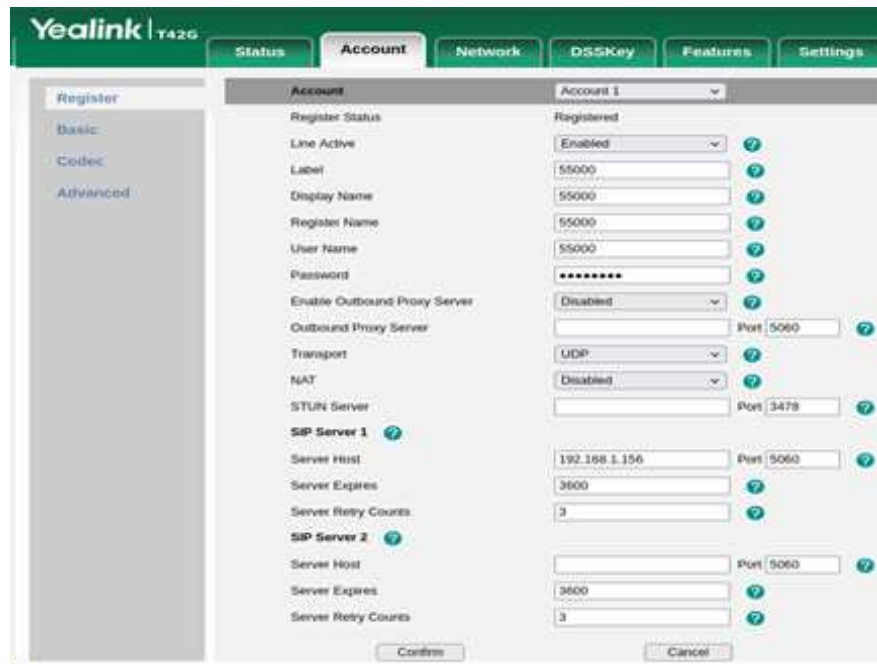


Figura 3.40: Configuración de un teléfono, muestra extensión y SIP server.

El siguiente paso es tener cada teléfono con un número de extensión de cada PBX. Se habló anteriormente de implementar 3 PBX virtuales, cada uno etiquetado como PBX01, 02 y 03, por lo tanto, cada equipo telefónico llevará un número de extensión que se encontrará configurado en cada uno de los PBX.

3.3.8. Configurar troncales entre PBX

Se realiza la configuración de troncales SIP dentro de cada PBX, para establecer comunicación entre extensiones de diferentes PBX.

Para ello se deben configurar cada una de las troncales en los PBX, es decir, para el PBX01 se crearán las troncales hacia el PBX02 y 03. Para el PBX02 se crean las troncales hacia el PBX01 y 03, finalmente para el PBX03 troncales al PBX01 y 02.

Para esta implementación se realizan las troncales para cada uno de los PBX virtuales,

primero dando nombre a cada troncal creada. Para identificar cada una de las troncales que dará comunicación entre PBX, se crean 2 troncales en cada uno, desde el PBX01 tiene las troncales PBX02 y PBX03, sucesivamente haciendo lo mismo para el PBX02 y PBX03. Esto con la finalidad que exista comunicación entre los diferentes PBX.

Se decide crear números de extensión diferentes como si se tratasen de números en cada plantel, para el primer PBX01 tenemos la extensión 55000 que pertenece al plantel San Lorenzo Tezonco, para PBX02 el número 66000 que es del plantel Cuauhtepac, y la extensión 77000 que es del plantel Centro Histórico pertenece al PBX03.

Pasos para crear las troncales SIP en cada PBX.

Dentro del primer PBX en a la sección “Conectividad”, seleccionamos la opción “Troncales”, ver figura 3.41.



Figura 3.41: Opción Trunks.

Para agregar una nueva troncal, es seleccionando “Add Trunk”, después seleccionar “Add SIP (chan sip) Trunk”, ver figura 3.42.

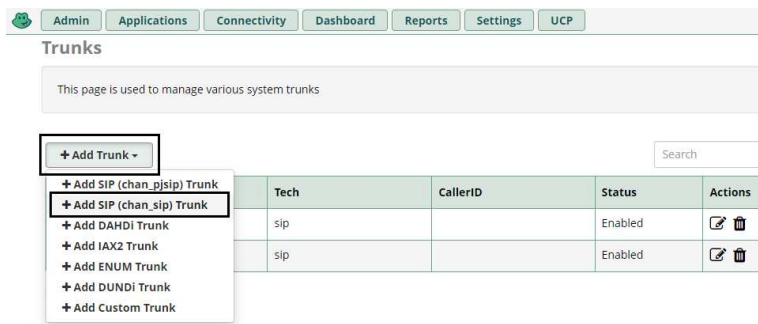


Figura 3.42: Agregar una nueva troncal.

Primero dando nombre a la troncal para identificarla, en caso que se desee agregar más troncales de servicio.

Seleccionamos la opción “*sip settings*”, para crear el enlace a donde se quiere tener comunicación, ver figura 3.43.

The screenshot shows the 'Add Trunk' configuration page. At the top, there are three tabs: 'General', 'Dialed Number Manipulation Rules', and 'sip Settings'. The 'sip Settings' tab is active. Below the tabs, there are several fields and options:

- Trunk Name**: A text input field, highlighted with a red box.
- Nombre de la troncal**: A text input field.
- Hide CallerID**: A toggle switch with 'Yes' and 'No' buttons.
- Outbound CallerID**: A text input field.
- CID Options**: A group of four buttons: 'Allow Any CID', 'Block Foreign CIDs', 'Remove CNAM', and 'Force Trunk CID'.
- Maximum Channels**: A text input field.

Figura 3.43: Agregar nombre a la troncal.

En las opciones SIP se van a registrar las siguientes reglas:

Outgoing Peer Details:

- **host=** (Será la dirección IP del PBX al que se desea establecer comunicación).
- **type=peer** (Es el tipo de enlace con el otro PBX, en este caso será peer ya que es una conexión entre direcciones IP fijas de cada PBX).
- **port=5060** (Puerto utilizado para la comunicación VoIP).
- **qualify=yes** (Verifica la conexión con la cuenta SIP registrada periódicamente)
- **transport=udp** (Protocolo usado para voz sobre IP).
- **tlsenable=no** (Define si se habilita el protocolo TLS en la comunicación entre enlaces).
- **sendrpid=yes** (Define si establece el envío de datos para el enlace).
- **trustrpid=yes** (Define si es de confiar en el enlace con quien se establece la troncal).

Las reglas anteriores se observan en la figura 3.44.

Edit Trunk

In use by 1 route

General | Dialed Number Manipulation Rules | **sip Settings**

Outgoing | Incoming

Trunk Name ⓘ PBX02

PEER Details ⓘ

host=192.168.1.157
type=peer
port=5060
qualify=yes
transport=udp
tlsenable=no
sendrpid=yes
trustpid=yes

Figura 3.44: Datos del enlace para la troncal hacia el PBX02.

Después de configurar las reglas de la troncal, guardamos los cambios y aplicamos configuraciones realizadas en el PBX, ver figura 3.45.

hboard | Reports | Settings | UCP | **Apply Config** ⓘ 🔍 ⚙️

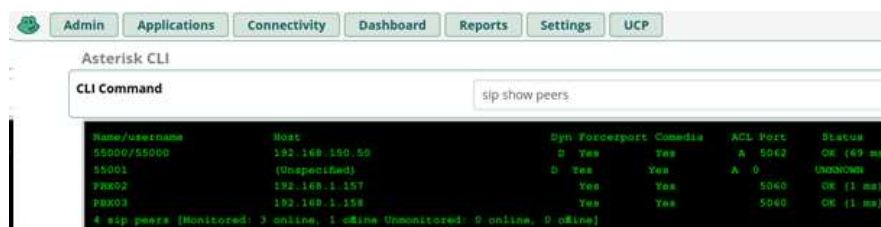
sip Settings

Submit Reset

Figura 3.45: Guardar cambios y aplicar configuraciones en el PBX.

Este procedimiento se realiza 2 veces en cada PBX. Ejemplo: para el PBX 2 se crea troncal PBX01 y PBX03 donde en “host=” va la dirección IP de cada PBX, los demás parámetros permanecen igual, en el PBX 1 creando troncal PBX02 y PBX03, finalmente en PBX 3 creando troncales PBX01 y PBX02.

Terminando de configurar las troncales, al dirigirnos a la opción “Admin”, seleccionar “Asterisk CLI” y ejecutar el siguiente comando: ***“sip show peers”***, el comando anterior es para que muestre el estatus de los números de extensión y troncales previamente configuradas, ver figura 3.46.



Name/Username	Host	Syn	Forceport	Comedia	ACL	Port	Status
55000/55000	192.168.150.50	0	Yes	Yes	A	5062	OK (49 ms)
55001	(Unspecified)	0	Yes	Yes	A	0	UNKNOWN
PBX02	192.168.1.157		Yes	Yes		5060	OK (1 ms)
PBX03	192.168.1.158		Yes	Yes		5060	OK (1 ms)

4 sip peers (Monitored: 3 online, 1 offline Unmonitored: 0 online, 0 offline)

Figura 3.46: Estatus de las extensiones registradas y troncales a los otros PBX.

Ahora se deben crear las “Outbound Routes”, para que pueda operar de manera correcta la comunicación entre troncales.

La finalidad de configurar los “Outbound Routes ” es:

Poder elegir un patrón de marcación específico y en base al tipo de marcación que se realice el PBX elegirá la troncal configurada para ese patrón de marcación. Si en el PBX01 tiene extensiones de la serie 55000 a 55999, al momento de marcar del PBX02 un número de extensión de esa serie, la ruta de salida reconoce el tipo de marcación y como resultado elige la troncal hacia el PBX01, para establecer la llamada por su troncal dedicada.

La configuración que se hizo para los PBX del LRC en SLT fue:

En la opción “Connectivity” se selecciona “Outbound Routes”, para crear una nueva ruta de salida, ver figura 3.47.



Figura 3.47: Creando una ruta de salida.

Para el PBX01 se crean dos Outbound Routes (PBX02 y PBX03).

Ingresando el nombre para identificar cada ruta que queremos configurar.

En la ruta de nombre PBX02 configuramos lo siguiente:

En la opción “*Trunk sequence for matched routes*”, Se elige la troncal configurada para el PBX02, incluso permite elegir una segunda troncal, esto en caso que tener alguna ruta de respaldo, por si se presenta algún fallo con la troncal principal, ver figura 3.48.

Admin Applications Connectivity Dashboard Reports Settings UCP

Outbound Routes

Edit Route: PBX02: PBX02

Route Settings Dial Patterns Import/Export Patterns Notifications Additional Settings

Route Name PBX02

Route CID

Override Extension Yes No

Route Password

Route Type Emergency Intra-Company

Music On Hold? default

Time Match Time Zone Use System Timezone

Time Match Time Group Permanent Route

Trunk Sequence for Matched Routes PBX02

Optional Destination on Congestion Trunks PBX02 (slp)

Note: Extension Routes is not registered

Figura 3.48: Parámetros que se configuran en “Outbound Routes”.

Ahora en la siguiente pestaña “Dial Patterns”, se indican las reglas sobre que números pueden pasar sobre la troncal configurada, ver figura 3.49.

Al estar configurando la troncal hacia el PBX02, sabemos que es plantel Cuauhtémoc y tiene extensiones de la serie 66XXX, entonces el “Dial Patterns” será todas las extensiones que empiecen con 66XXX, se asigna “66XXX” para especificar, así cualquier extensión de la serie 66000 a la 66999 establece comunicación por la troncal PBX02, que son todos los números que existen en plantel Cuauhtémoc.

Route Settings Dial Patterns Import/Export Patterns Additional Settings

Dial Patterns that will use this Route

Pattern Help

Dial patterns wizards

prepend prefix 66000

prepend prefix match pattern

Figura 3.49: Dial Patterns para especificar las marcaciones.

Posteriormente se crea otra ruta de salida para el PBX03, seleccionando la troncal del

PBX03 y en *“dial patterns”* estableciendo la regla ahora con los números 77XXX, guardando la configuración y aplicando cambios en el PBX01, con esto ya están configuradas las dos rutas de salida hacia el PBX02 y PBX03, ver figura 3.50.



Figura 3.50: “Outbound Routes” configuradas en el PBX01.

Por último y una configuración importante, accedemos a las configuraciones SIP de Asterisk, para elegir el número de puerto que utilizamos en la comunicación de VoIP.

En la opción Settings - Asterisk SIP Settings (ver fig. 3.51) se despliegan tres pestañas, se selecciona la pestaña Chan SIP Settings, y después el apartado Advance General Settings que incluye la opción Bind Port. La opción por default tiene configurado el puerto 5160, pero debe cambiarse al puerto 5060. Lo mismo debe hacerse en la opción Chan PJSIP Settings. Para aplicar los cambios en el PBX se selecciona la opción Submit.

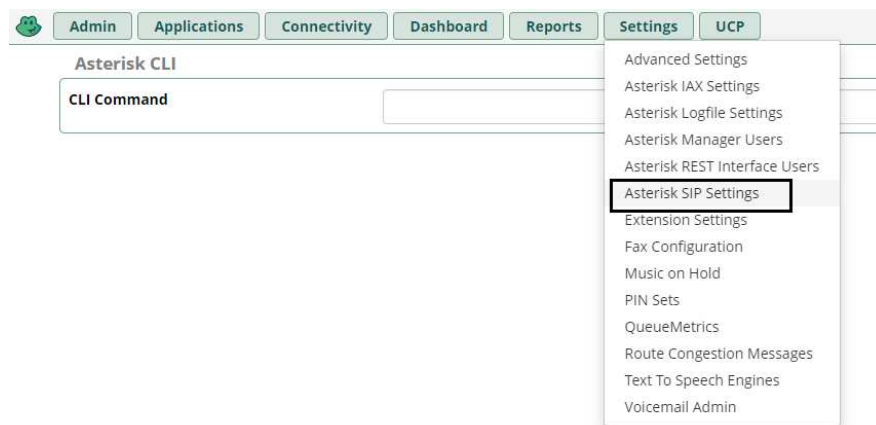


Figura 3.51: Configuraciones SIP.

NOTA: Para la comunicación a través de servicios por SIP normalmente se utiliza el puerto definido 5060 UDP y TCP, es por esto que para la comunicación entre extensiones

hay que definir ese puerto, garantizando la comunicación entre equipos usando el protocolo SIP. Las extensiones creadas utilizan el driver Chan SIP, por tal motivo en la configuración Chan SIP definir el puerto 5060, se debe intercambiar el número de puerto entre Chan SIP y Chan PJSIP. Las configuraciones no permitirán hacer el cambio entre ambos, se debe ingresar un número de puerto y guardar cambios para la primer opción (Chan PJSIP) y guardar cambios, posteriormente se asigna el puerto deseado 5060 en la segunda opción (Chan SIP) y aplicamos cambios, por último guardaremos la configuración del PBX. Como se muestra en la figura 3.52, se realiza el cambio de puerto entre drivers y se aplican los cambios en el PBX.

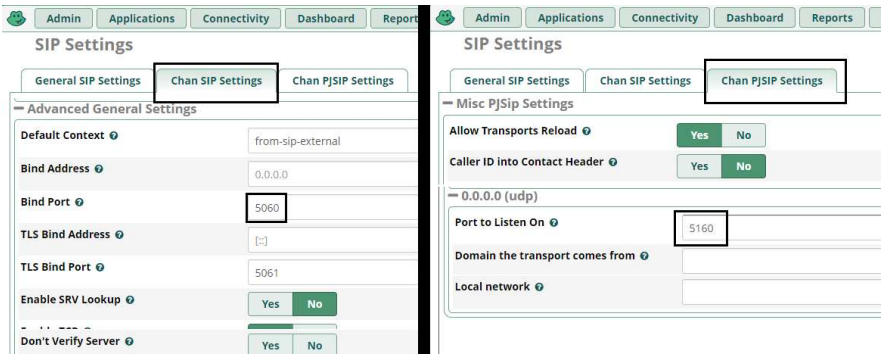


Figura 3.52: Cambio de puerto 5060 entre chan SIP y chan PJSIP.

Finalmente se tiene la configuración de cada PBX, con sus respectivas extensiones y las troncales configuradas, como en las figuras 3.53 y 3.54 al ejecutar el comando “sip show peers” se observa el estado de las extensiones y troncales.

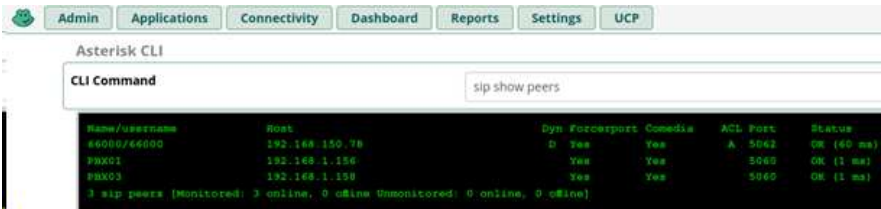


Figura 3.53: Número de extensión y troncales en el PBX02 plantel Cuauhtec.

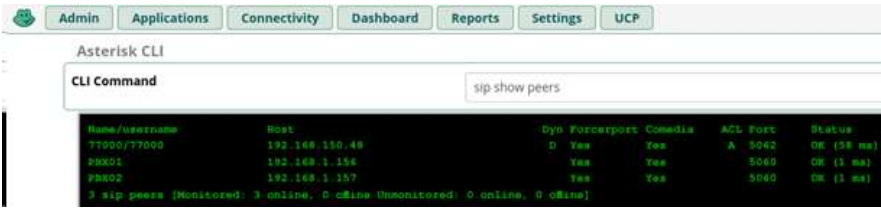


Figura 3.54: Número de extensión y troncales en el PBX03 Plantel Centro Histórico.

En la figura 3.55 se realizó la primer prueba de los servicios de manera presencial. Para las pruebas en el LRC del plantel SLT se cuenta con 3 teléfonos IP, a cada teléfono se le configuró una extensión de cada PBX.

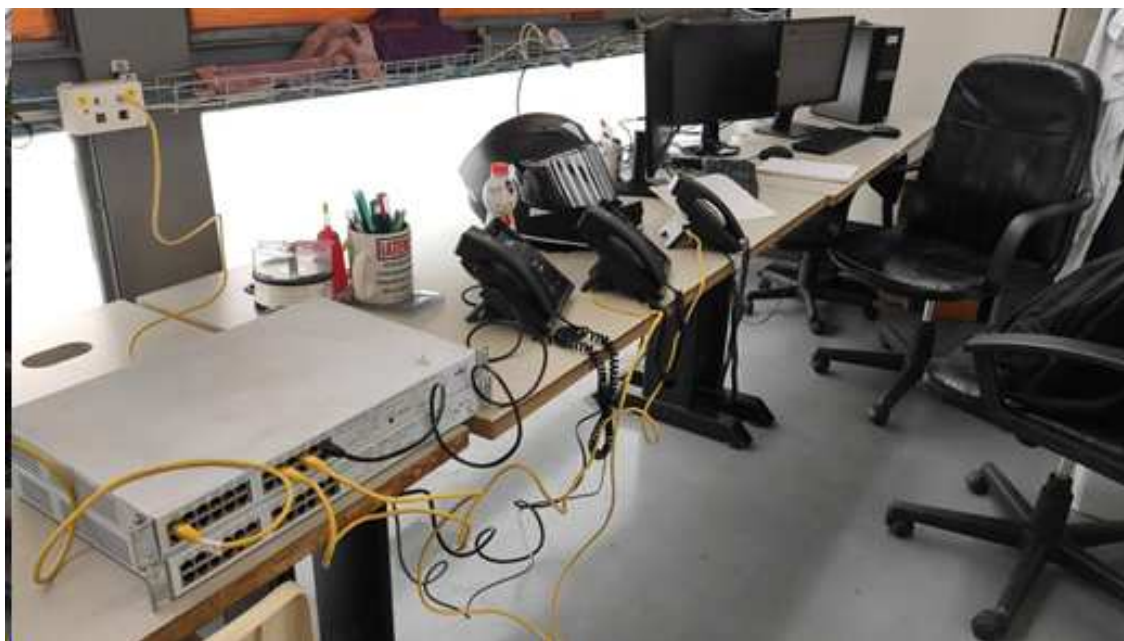


Figura 3.55: Primer prueba realizada con 3 teléfonos IP, un switch con puertos PoE y los 3 PBX virtuales con sus troncales configuradas.

Con esto se llega a la culminación del proyecto realizado, logrando tener comunicación entre los 3 PBX configurados, cada uno en su propio servidor virtual, lo que vendría después ya es agregar más extensiones, las que los planteles necesiten para tener los servicios de VoIP operando correctamente.

A continuación sigue un breve resumen sobre las pruebas y el análisis de resultados que se obtuvieron, así como todo lo aprendido sobre la implementación de servicios de voz utilizando el entorno de virtualización.

Análisis de resultados

Para comenzar con el análisis de resultados generamos una tabla que especifica todos los puntos a revisar de los servicios (hipervisor, clúster, VM, PBX, troncales, extensiones, servicios) y el tipo de pruebas realizadas para concluir con los resultados obtenidos. Para esta tabla se tomaron específicamente los siguientes puntos a considerar ya que son algunos de los objetivos específicos y otros son pruebas del servicio, funcionalidad y resultados esperados. Esta lista sirve para llevar un control del estatus de los PBX, desde validar los sistemas activos, hasta una serie de pruebas sobre las funcionalidades y servicios de VoIP.

En la tabla 4.1 marcamos como validaciones cada una de las configuraciones a lo largo del desarrollo del trabajo, validando servicios activos y correctos, para finalmente pruebas de comunicación y resultados obtenidos para cada PBX. Una lista que sirve para obtener un panorama general sobre los objetivos y resultados esperados, como se muestra en la siguiente tabla validamos el hipervisor, el clúster, las VM, el acceso web a cada PBX, acceso por CLI a los PBX, comprobando la correcta configuración de cada PBX, puerto SIP, extensiones creadas y activas en los teléfonos, las troncales SIP correctamente configuradas en cada PBX y las rutas de salida para cada PBX.

Finalmente al termino de la lista validamos la funcionalidad de los servicios, marcaciones entre extensiones para probar la comunicación dentro y fuera de cada PBX y por último probar la herramienta “sngrep” para visualizar el tráfico SIP de los PBX.

Lista de pruebas para los PBX

Prueba Realizada	PBX Plantel SLT	PBX Plantel Cuau-tepec	PBX Plantel Centro Histórico
Hipervisor activo	Con acceso	Con acceso	Con acceso
Clúster activo y funcionando	Con acceso	Con acceso	Con acceso
Máquinas virtuales funcionando	Con acceso	Con acceso	Con acceso
Backup realizado en cada VM	Correcto	Correcto	Correcto
Validar acceso web a freePBX	Con acceso	Con acceso	Con acceso
Validar acceso por SSH a los PBX	Con acceso	Con acceso	Con acceso
Creación de extensiones en cada PBX	Correcto	Correcto	Correcto
Configuración de puertos para protocolo SIP	Puerto 5060 asignado	Puerto 5060 asignado	Puerto 5060 asignado
Configuración de troncales	Correcto	Correcto	Correcto
Configuración de Rutas de salida	Correcto	Correcto	Correcto
Comprobar servicios activos en CLI	Correcto	Correcto	Correcto
Prueba de llamadas entre extensiones del mismo PBX	Correcto	Correcto	Correcto
Prueba de llamadas entre extensiones de diferentes PBX	Correcto	Correcto	Correcto
Análisis del tráfico SIP con sngrep	Correcto	Correcto	Correcto

Cuadro 4.1: Lista de todas las pruebas realizadas en cada PBX.

La tabla 4.1 muestra el diagrama de red que originalmente tiene la finalidad de demostrar como realizar las conexiones y operar los servicios en el entorno de virtualización para servicios de VoIP, se cuenta con 3 servidores en los que instalamos Proxmox, el entorno de virtualización y donde habitan cada uno de los PBX, todos unificados en un clúster creado por los servidores físicos y operando sobre la red denominada “red de administración”. En el

misimo segmento de red contamos con una PC de administración para llevar la gestión de los servicios de voz. Sobre esa misma infraestructura se tiene otro segmento de red denominado “red de operación”, ese segmento es exclusivo para la conexión de los teléfonos IP, para que específicamente los servicios VoIP funcionen sobre esa red de operación.

Lo anterior resuelve una de las problemáticas en los servicios cotidianos para las empresas u organizaciones que desean implementar funcionalidades compatibles y tener la infraestructura lo más actual posible.

Con la implementación en el LRC del plantel SLT se realizaron pruebas de comunicación, logrando validar el servicio estable y funcional.

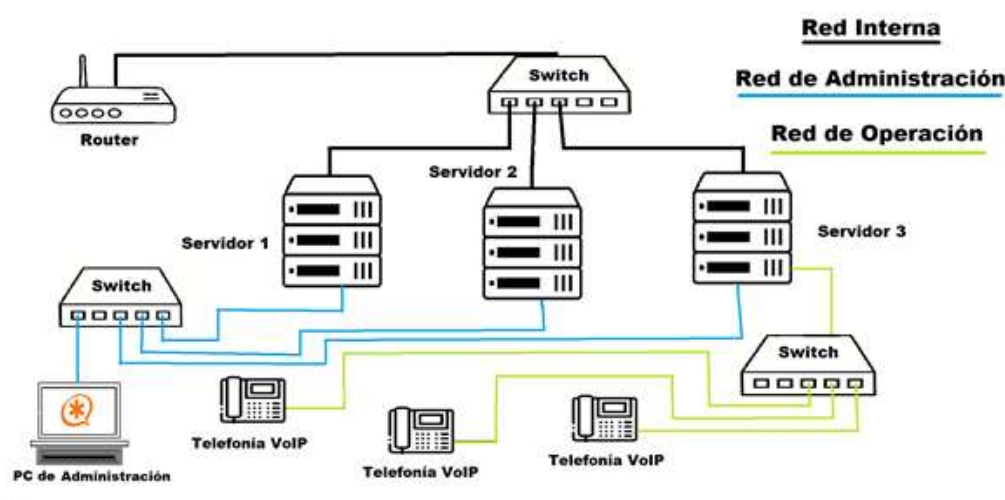


Figura 4.1: Diagrama de la topología de red del proyecto en operación.

Se usaron 3 PBX virtuales capaces de establecer comunicación entre sí. Realizando una lista para validar que todas las funcionalidades de cada PBX operen de manera correcta, adicional con pruebas de llamadas para comprobar servicios, logrando la comunicación y el funcionamiento de los enlaces troncales entre cada PBX.

En la figura 4.2 se muestra una segunda prueba de servicios de manera presencial en el LRC, para esta ocasión con los equipos telefónicos cerca de los servidores que cuentan con el clúster de Proxmox, desde una laptop conectada a la red interna del laboratorio, entrando al GUI del hipervisor dentro del clúster, para después acceder a un PBX y verificar servicios. Comprobando que cada teléfono tenga extensión correctamente configurada y habilitada, para después realizar pruebas de llamadas con cada teléfono, validando nuevamente el correcto funcionamiento de troncales SIP entre cada PBX.

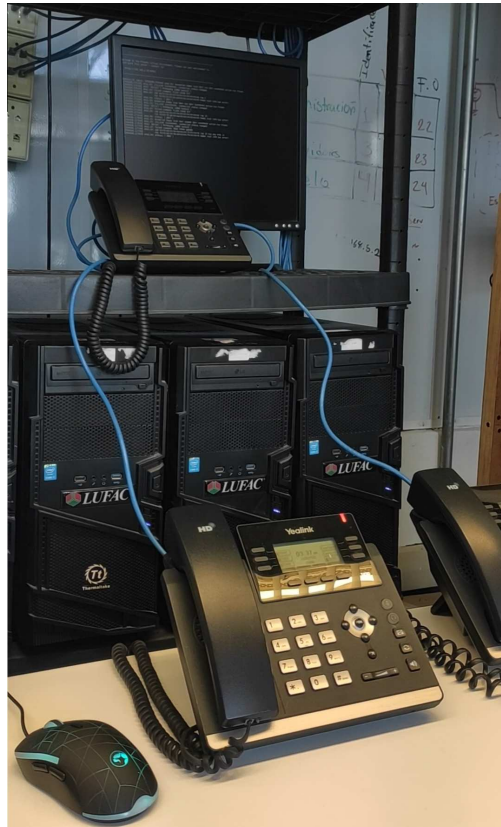


Figura 4.2: Segunda fase de pruebas en el LRC.

Ahora para otro tipo de pruebas, se necesita acceder a la consola del PBX por medio de SSH, donde se puede visualizar el tráfico de llamadas entrantes o salientes.

Una de las pruebas es lograr demostrar la comunicación que se tiene entre extensiones a través de las troncales SIP configuradas para cada PBX, estableciendo la comunicación a través del protocolo SIP con ayuda de una herramienta llamada “**sngrep**”, esta herramienta nos ayuda a comprobar como se establece una comunicación SIP, para este proyecto fue a través de la comunicación VoIP entre extensiones.

La herramienta viene integrada dentro de cada PBX virtual, como una paquetería adicional para Asterisk, con la herramienta instalada simplemente debemos desplegarla ejecutando el siguiente comando:

- **sudo sngrep**

Una vez abierta la herramienta “sngrep” podemos comenzar con el análisis del tráfico SIP, realizando una marcación, ya sea entre extensiones del mismo PBX como también recibir

llamadas de otro PBX siempre y cuando la extensión origen o destino habite dentro del PBX en el que desplegamos la herramienta “sngrep”, ver figura 4.3.

sngrep - SIP messages flow viewer						
Current Mode: Online [any]		Dialogs: 40 (10 displayed)				
Match Expression:		BPF Filter:				
Display Filter:						
Idx	Method	SIP From	SIP To	Msgs	Source	Destination
[] 1	OPTIONS	Unknown0192.168.1.91	550000192.168.1.121:5060	2	192.168.1.91:5060	192.168.1.121:5060
[] 6	OPTIONS	Unknown0192.168.1.91	550000192.168.1.121:5060	2	192.168.1.91:5060	192.168.1.121:5060
[] 11	OPTIONS	Unknown0192.168.1.91	550000192.168.1.121:5060	2	192.168.1.91:5060	192.168.1.121:5060
[] 18	OPTIONS	Unknown0192.168.1.91	550000192.168.1.121:5060	2	192.168.1.91:5060	192.168.1.121:5060
[] 23	OPTIONS	Unknown0192.168.1.91	550000192.168.1.121:5060	2	192.168.1.91:5060	192.168.1.121:5060
[] 28	OPTIONS	Unknown0192.168.1.91	550000192.168.1.121:5060	2	192.168.1.91:5060	192.168.1.121:5060
[] 33	OPTIONS	Unknown0192.168.1.91	550000192.168.1.121:5060	2	192.168.1.91:5060	192.168.1.121:5060
[] 34	INVITE	770000192.168.1.93	550000192.168.1.91:5060	12	192.168.1.93:5060	192.168.1.91:5060
[] 35	INVITE	770000192.168.1.91	550000192.168.1.121:5060	11	192.168.1.91:5060	192.168.1.121:5060
[] 40	OPTIONS	Unknown0192.168.1.91	550000192.168.1.121:5060	2	192.168.1.91:5060	192.168.1.121:5060

Figura 4.3: Herramienta sngrep analizando todas las llamadas que se realizan dentro del PBX.

Proporciona un desglose de información muy completa, como se muestra en la figura 4.4, del lado izquierdo de la ventana se muestra el establecimiento de la comunicación, muestra el flujo de cómo se realiza una llamada entre dos servicios VoIP, van desde la invitación para establecer la comunicación, regresando un trying (intentando establecer la comunicación) y después un ringing (dando un tono de llamada esperando que el destino tome la llamada), para finalmente establecer un canal de comunicación entre ambas extensiones proporcionado por RTP y finalmente al terminar la comunicación libera el canal asignado para posterior ser utilizado en otra comunicación SIP.

Del lado derecho observamos toda la información que se tiene en dicha comunicación SIP, desde los datos de la extensión con la dirección IP del servidor, el protocolo usado para la transmisión de los datos, información del destino que es el otro número de extensión con su respectiva dirección IP del servidor y datos sobre el protocolo RTP.

Call flow for 18ec57e80f407cf96e15cace47bade120192.168.1.93:5060 (Color by Request/Response)		
192.168.1.93:5060	192.168.1.91:5060	
15:14:47.998929	INV (192.168.1.93) audio 13494 (g711u)	INVITE sip:550000192.168.1.91:5060 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.93:5060;branch=z9hG4kK2475b97b;port Max-Forwards: 70 From: "Centro Historico" <sip:770000192.168.1.93>;tag=as68d22c46 To: <sip:550000192.168.1.91:5060> Contact: <sip:770000192.168.1.93:5060> Call-ID: 18ec57e80f407cf96e15cace47bade120192.168.1.93:5060 CSeq: 102 INVITE User-Agent: FFMPEG-15.0.17.34(16.17.0) Date: Thu, 19 Oct 2023 20:14:47 GMT Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY INFO, PUBLISH, MESSAGE Supported: replaces, timer Remote-Party-ID: "Centro Historico" <sip:770000192.168.1.93>;party alling;privacy=off;screen=no Content-Type: application/sdp Content-Length: 354
+0.003246	100 Trying	
15:14:47.994175	100 Ringing	
+0.123418	100 Ringing	
15:14:48.117595	200 (192.168.1.91) audio 16474 (g711u)	
+0.065284	ACK	
15:14:48.182869	RTP (g711u) 1534	
+9.433252	RTP (g711u) 1525	
15:14:57.616121	BYE	
+0.000646	200 OK	
15:14:57.616767		
15:14:57.713186		
15:14:57.988747		
+30.774537		
15:15:28.391364		
+0.000767		
15:15:28.392131		

Figura 4.4: Información de los equipos que establecieron una llamada.

Sngrep es una herramienta muy útil para los servicios de telefonía, capaz de proporcionar información de una comunicación SIP entre equipos, muy útil si se presenta algún problema de comunicación, con los datos que proporciona se puede identificar si existe algún tipo de falla en la comunicación.

Desde sngrep se visualiza como se realiza la negociación del protocolo SIP y se logra la comunicación, sngrep tiene la opción de habilitar el protocolo RTP para visualizar a detalle el canal que se asigna en la comunicación, esta información la podemos ver en la figura 4.5.

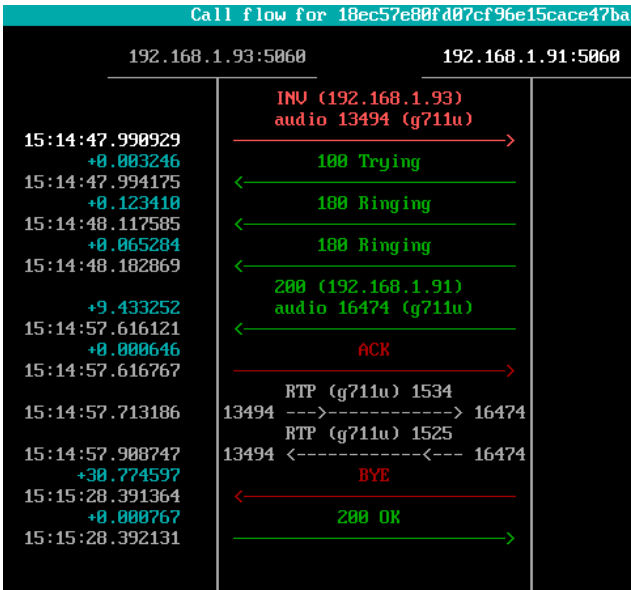


Figura 4.5: Traza de una llamada desde el inicio hasta terminar la comunicación.

Logrando tener comunicación entre extensiones, se confirma que los PBX virtuales funcionan de manera correcta. Este modelo de servicios VoIP virtualizados ofrece la ventaja de lograr una mayor escalabilidad de servicios, lo que se traduce como un crecimiento en la demanda de usuarios, utilizando los servicios de VoIP, sin la necesidad de expandir el hardware para los servicios.

Con la práctica realizada en este proyecto se aprendió todo lo relacionado en configuración e implementación sobre la plataforma Proxmox para crear un ambiente virtual, dando solución a la problemática de contar con el sistema operativo actualizado en los PBX para poder certificar los servidores en la empresa Grupo Salinas, pudiendo eliminar brechas de seguridad en los equipos, un requisito prioritario para la empresa, es un punto de seguridad con el que todo equipo y servicios deben contar.

Para finalizar el trabajo escrito, en el siguiente capítulo cerramos el proyecto con las conclusiones sobre la implementación, desarrollo, configuraciones, pruebas y todo el conocimiento adquirido.

Conclusiones y recomendaciones para trabajo futuro

Utilizando un entorno virtual se implementaron servicios de voz, aprendimos a utilizar el software de virtualización, aplicando las configuraciones necesarias y empleando algunas funcionalidades con las que cuenta Proxmox; logramos emplear servicios de VoIP aplicados al entorno virtual. Tales configuraciones son de gran ayuda para tener una mejor gestión de los servicios, el crear un clúster con los servidores de LRC sirve para muchas aplicaciones pero las conclusiones del presente trabajo están basados en los servicios de VoIP virtualizando PBX.

Empleamos la funcionalidad de crear copias de seguridad dentro de Proxmox para las VM, con la finalidad de tener un respaldo de información en los servidores. Son de gran utilidad en caso de tener algún problema o algún error en la configuración de las VM o de los PBX.

Se sabe que todo sistema esta expuesto a presentar fallas, por este motivo tener un respaldo de la información, a modo de prevenir errores o fallas en los PBX y gracias a las copias de seguridad, se pueden restaurar los servicios de manera rápida, eficaz y lo más actual posible para el caso de presentar algún fallo.

Logramos tener un buen manejo del software Proxmox para crear VM, para elegir la capacidad del equipo con base en las características de los CPU físicos que se tienen en el LRC, con base en las especificaciones comenzamos a crear los PBX con cierta capacidad en disco duro, memoria RAM, interfaces de red y núcleos necesarios en el procesador para el PBX.

Gracias al manejo del hipervisor con el que cuenta Proxmox fue como se emplean las herramientas necesarias, logrando crear un clúster dentro del LRC del plantel SLT.

Teniendo esos recursos, dimos comienzo a crear tres VM que utilizamos para instalar el software Asterisk, creando 3 PBX para los servicios de voz.

Aplicando el conocimiento de las configuraciones de Proxmox es como se crean las VM,

aprendiendo a crear VM, asignando los recursos necesarios para el funcionamiento de los PBX. Con el clúster en operación contamos con los 3 PBX en carga balanceada, para garantizar el buen funcionamiento de los servicios, aprendiendo a crear un PBX desde las configuraciones iniciales, hasta poner en funcionamiento números de extensión y troncales SIP de servicio.

Asterisk es la herramienta crucial para ofrecer una comunicación de servicios VoIP dentro del LRC del plantel SLT, usando teléfonos de tecnología VoIP logrando activar servicios en ellos y realizando pruebas de comunicación para garantizar el funcionamiento tanto de troncales como de los PBX.

Para la parte de pruebas realizadas, se concluye que la herramienta sngrep es una forma de fácil acceso para el análisis de los datos que se transmiten en las pruebas de comunicación, tenemos todos los datos disponibles en una comunicación SIP logrando analizar el tráfico, ahora para que una comunicación SIP exista entre origen y destino debe de cumplirse una sesión entre equipos que necesitan realizar la comunicación por VoIP. Comparando la información obtenida en el capítulo de análisis de resultados por la herramienta sngrep, se comprueba que estableciendo una troncal SIP entre PBX logramos establecer un canal por medio del protocolo SIP para el intercambio de información entre equipos VoIP.

Los resultados por sngrep mostraron una correcta comunicación, primero el origen solicitando una sesión con el destino, para después recibir una respuesta a establecer comunicación, para finalmente SIP proporcionar un canal y por este mismo canal comenzar con el flujo de datos a través de internet, ese mismo canal proporcionado por RTP.

Cuando finalmente se concluye la comunicación se libera ese canal y tanto origen como destino reciben una última respuesta de que finalizó la comunicación.

Este resultado comparado con la teoría del protocolo SIP indica que se cumplieron los pasos y el procedimiento de manera correcta, concluyendo que las pruebas de comunicación por SIP resultaron exitosas.

Estas configuraciones y funcionalidades nos abren un panorama mucho más amplio para visualizar el alcance que puede llegar a tener el proyecto. Para el presente trabajo se aplicaron servicios en 3 PBX para la comunicación entre sí, viendo que se pueden aplicar más funcionalidades o proponer mejoras en el desarrollo del trabajo para llevar el proyecto fuera del banco de pruebas, validando que las configuraciones y aplicaciones usadas funcionan a la perfección para diferentes entornos.

Todas las pruebas y resultados concluyeron como se esperaba, el proyecto fue capaz de lograr los objetivos definidos al inicio del trabajo, también mencionar que todo el proceso del proyecto es con la finalidad de mantener el servicio actualizado, cumpliendo con las normas de seguridad y evitar vulnerabilidades en los servicios. Contando con el entorno virtual, en caso de requerir actualización de servicios, empleando una copia de seguridad respaldando las configuraciones actuales para después iniciar una actualización, y en caso de presentar algún problema de actualización podemos replicar servicios sin perder información o algún

otro problema con los equipos.

Adquirimos gran experiencia en configuraciones de servicios, implementación de servicios virtualizados, procedimientos de instalación, configuración y aplicación de pruebas. Se adquirió un amplio conocimiento sobre herramientas de virtualización, software para servicios de voz, conceptos relacionados con la comunicación a través de VoIP, logrando analizar el tráfico de comunicación SIP al emplear una herramienta especializada para analizar todo el tráfico de voz, dando un panorama más amplio para saber cómo se establece la comunicación y entender cómo están operando los servicios que se implementaron en el presente proyecto.

5.1. Recomendaciones para trabajo futuro

Como recomendaciones sobre el presente proyecto podemos agregar que se puede crear un mayor alcance en los servicios que se implementaron, es decir, se pueden hacer pruebas de los servicios desde otros segmentos de red, como la posibilidad de utilizar la red de otro laboratorio en SLT para configurar un PBX que funcione a través de una VPN y lograr tener comunicación con los otros 3 PBX del presente proyecto. Dicha configuración abre un panorama para utilizar más recursos y conceptos sobre el entorno de telecomunicaciones y redes, logrando agregar otro tipo de tecnologías e implementación de los servicios de redes dentro del plantel SLT.

Como siguiente recomendación para trabajo futuro no solo podemos utilizar VM. Proxmox es una herramienta muy completa capaz de brindarnos el manejo de contenedores Linux, por tanto al utilizar LXC para instalar el software Asterisk, se puede crear un PBX.

La diferencia entre utilizar una VM y un contenedor está en el uso de los recursos del servidor físico, es decir, para una VM es necesario destinar y configurar los recursos con los que opera la VM (hay que destinar cierta cantidad de RAM, disco duro, núcleos para el CPU, entre otros componentes virtuales), mientras que un LXC utiliza directamente los recursos físicos del servidor, por tal motivo el contenedor es más ligero y no le exige demasiado al servidor físico, en otras palabras una VM tendrá un mayor consumo de recursos ya que hacen una virtualización del hardware, mientras que un LXC solo virtualiza el sistema operativo, los componentes hardware los utiliza como si fuera el propio servidor físico.

Referencias

- [1] HN Datacenter “*Historia de la Virtualización — HN Datacenter en Chile*”. Hosting Rápido con cPanel y certificado SSL en Chile — HN Datacenter. Disponible: <https://www.hn.cl/blog/hablemos-de-virtualizacion/>.
- [2] I. Lemus “Historia de la virtualización - Conocimiento Libre”. Conocimiento Libre. [En línea]. Disponible: <https://conocimientolibre.mx/historia-de-la-virtualizacion/>
- [3] E. Limones. “Virtualización: Qué es, para qué sirve y ventaja”. OpenWebinars.net. [En línea]. Disponible: <https://openwebinars.net/blog/virtualizacion-que-es-para-que-sirve-y-ventajas/>
- [4] IBM. “¿Qué es la virtualización? — IBM”. IBM in Deutschland, Österreich und der Schweiz — IBM. [En línea]. Disponible: <https://www.ibm.com/mx-es/topics/virtualization>
- [5] Red Hat. “¿Qué es y para qué sirve la virtualización?” Red Hat - We make open source technologies for the enterprise. [En línea]. Disponible: <https://www.redhat.com/es/topics/virtualization/what-is-virtualization>
- [6] KVM “¿Qué es KVM?”. Red Hat - We make open source technologies for the enterprise. <https://www.redhat.com/es/topics/virtualization/what-is-KVM>.
- [7] Amazon AWS. “¿Qué es la virtualización? - Explicación de la virtualización de la computación en la nube - AWS”. Amazon Web Services, Inc. [En línea]. Disponible: [https://aws.amazon.com/es/what-is/virtualization/#:~:text=La %20virtualización %20es %20una %20tecnología,en %20una %20única %20máquina %20física.](https://aws.amazon.com/es/what-is/virtualization/#:~:text=La%20virtualización%20es%20una%20tecnología,en%20una%20única%20máquina%20física.)
- [8] ProxmoxVE “*Proxmox VE - Virtualization Management Platform*”. Proxmox - Powerful open-source server solutions. <https://www.proxmox.com/en/proxmox-ve>.
- [9] Asterisk “*Home * Asterisk*”. Asterisk. <https://www.asterisk.org/>.
- [10] Colaboradores de los proyectos Wikimedia. “*Proxmox Virtual Environment - Wikipedia, la enciclopedia libre*”. Wikipedia, la enciclopedia libre. [En línea]. Disponible: https://es.wikipedia.org/wiki/Proxmox_Virtual_Environment.

-
- [11] Genos Cloud Services. “¿Qué es Proxmox VE? Soporte Proxmox VE - Genos”. Genos - Servicios Cloud para empresas. [https://genos.es/proxmox-ve/#:~:text=Proxmox %20Virtual %20Environment %20es %20una,hardware %20 y %20el %20tiempo %20empleado](https://genos.es/proxmox-ve/#:~:text=Proxmox%20Virtual%20Environment%20es%20una,hardware%20y%20el%20tiempo%20empleado).
 - [12] Colaboradores de los proyectos Wikimedia. “Asterisk - Wikipedia, la enciclopedia libre”. Wikipedia, la enciclopedia libre. [En línea]. Disponible: <https://es.wikipedia.org/wiki/Asterisk>
 - [13] Colaboradores de los proyectos Wikimedia. “PBX - Wikipedia, la enciclopedia libre”. Wikipedia, la enciclopedia libre. [En línea]. Disponible: <https://es.wikipedia.org/wiki/PBX>
 - [14] Colaboradores de los proyectos Wikimedia. “Central telefónica - Wikipedia, la enciclopedia libre”. Wikipedia, la enciclopedia libre. [En línea]. Disponible: https://es.wikipedia.org/wiki/Central_telefónica
 - [15] Colaboradores de los proyectos Wikimedia. “Línea troncal - Wikipedia, la enciclopedia libre”. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Línea_troncal.
 - [16] “¿Qué es PSTN?” 3CX.es [En línea]. Disponible: [https://www.3cx.es/voip-sip/pstn/#:~:text=PSTN %20son %20las %20siglas %20para,de %20datos %20entre %20los %20usuarios](https://www.3cx.es/voip-sip/pstn/#:~:text=PSTN%20son%20las%20siglas%20para,de%20datos%20entre%20los%20usuarios).
 - [17] IBM. “IBM Documentation”. IBM in Deutschland, Österreich und der Schweiz — IBM. [En línea]. Disponible: <https://www.ibm.com/docs/es/i/7.1?topic=alternatives-t1e1-fractional-t1-connections>
 - [18] “¿Qué es la multiplexación por división de tiempo (TDM)? — phoenixNAP Glosario de TI”. [En línea]. Disponible: [https://phoenixnap.mx/glosario/multiplexación-por-división-de-tiempo#:~:text=La %20multiplexación %20por %20división %20de %20tiempo %20\(TDM\) %20es %20un %20método,intervalos %20de %20tiempo %20de %20transmisión](https://phoenixnap.mx/glosario/multiplexación-por-división-de-tiempo#:~:text=La%20multiplexación%20por%20división%20de%20tiempo%20(TDM)%20es%20un%20método,intervalos%20de%20tiempo%20de%20transmisión).
 - [19] Colaboradores de los proyectos Wikimedia. “Protocolo de iniciación de sesión - Wikipedia, la enciclopedia libre”. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Protocolo_de_iniciación_de_sesión.
 - [20] Contributors to Wikimedia projects. “Power over Ethernet - Wikipedia”. Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Power_over_Ethernet.
 - [21] Charlene. “¿Qué es un Switch PoE y cómo funciona? — Comunidad F”. Knowledge. <https://community.fs.com/es/blog/what-is-a-poe-switch-and-how-does-it-work.html>.
 - [22] J. Gerend. “Protocolo de configuración dinámica de host (DHCP)”. Microsoft Learn: Build skills that open doors in your career. <https://learn.microsoft.com/es-es/windows-server/networking/technologies/dhcp/dhcp-top>.
-

-
- [23] S. de Luz. “¿Qué protocolo es mejor?: TCP vs UDP, descubre cuándo usar cada uno”. RedesZone. <https://www.redeszone.net/tutoriales/internet/tcp-udp-caracteristicas-uso-diferencias/>.
- [24] Colaboradores de los proyectos Wikimedia. “Dirección MAC - Wikipedia, la enciclopedia libre”. Wikipedia, la enciclopedia libre. [En línea]. Disponible: https://es.wikipedia.org/wiki/Dirección_MAC
- [25] Oracle. “Tarjetas de interfaz de red virtual (VNIC)”. Moved. [En línea]. Disponible: <https://docs.oracle.com/es-ww/iaas/Content/Network/Tasks/managingVNICs.htm>
- [26] IBM. “IBM Documentation”. IBM in Deutschland, Österreich und der Schweiz — IBM. [En línea]. Disponible: <https://www.ibm.com/docs/es/ibm-mq/9.1?topic=mechanisms-cryptographic-security-protocols-tls>
- [27] Colaboradores de los proyectos Wikimedia. “Seguridad de la capa de transporte - Wikipedia, la enciclopedia libre”. Wikipedia, la enciclopedia libre. [En línea]. Disponible: https://es.wikipedia.org/wiki/Seguridad_de_la_capa_de_transporte
- [28] BalenaEtcher “*balenaEtcher - Flash OS images to SD cards & USB drives*”. balena - The complete IoT fleet management platform. <https://www.balena.io/etcher/>.
-