UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

Programa de Apoyo al Egreso de Profesionales en Actividad

TÍTULO: Infraestructura Hiperconvergente

AUTOR: Moglia Matías David

DIRECTOR ACADÉMICO: Mg. Paula Venosa

DIRECTOR PROFESIONAL: Diego Alejandro Del Roio

CARRERA: Licenciatura en Informática

Resumen

El concepto de infraestructura hiperconvergente se refiere al conjunto de tecnologías utilizadas para generar una infraestructura definida por software, eliminando la separación entre los recursos computacionales, redes y almacenamiento y generando la capacidad de hacer crecer de forma horizontal el centro de datos, utilizando servidores estándar en la industria.

Palabras Clave

Infraestructura Hiperconvergente – Cluster – Alta Disponibilidad – Almacenamiento Definido por Software – Redes Definidas por Software - Virtualización

Conclusiones

Este proyecto comenzó por medio de la implementación de la nueva red provincial, la misma incluye, en uno de sus ítems, la compra de equipamiento de cómputo con el fin de dar soporte a servicios básicos y centrales que la dirección presta. Para ello se decidió implementar el cluster de HA con Proxmox VE, ya que permite concentrar, por medio de las tecnologías presentadas en el proyecto, la administración del cómputo, red y almacenamiento.

Trabajos Realizados

Resumen – Marco Conceptual – Objetivos – Descripción de las herramientas – Equipamiento utilizado – Instalación de la plataforma – Configuración de la plataforma – Funcionamiento – Conclusiones - Trabajos Futuros

Trabajos Futuros

- Disaster Recovery Site.
- Automatización de operaciones.
- Observabilidad.

Fecha de la presentación: 12/2019

Infraestructura Hiperconvergente

Tesina de Grado

Licenciatura en Informática – Plan 2015

Matias D. Moglia

Directora Académica: Mg. Paula Venosa

Director Profesional: Diego Alejandro Del Roio

Facultad de Informática

Universidad Nacional de La Plata

Resumen

El concepto de infraestructura hiperconvergente se refiere al conjunto de tecnologías utilizadas para generar una infraestructura definida por software, eliminando la separación entre los recursos computacionales, redes y almacenamiento y generando la capacidad de hacer crecer de forma horizontal el centro de datos, utilizando servidores estándar en la industria.

El crecimiento de los requerimientos informáticos, la convergencia entre Internet y las comunicaciones móviles, así como también el número creciente de sistemas IOT, conllevan el surgimiento de nuevos requerimientos para el desarrollo de aplicaciones sociales que puedan ser implementadas sobre estos sistemas. Ante esta necesidad, la posibilidad de generar una infraestructura fácil de mantener y de crecer es una prioridad para la Dirección Provincial de Telecomunicaciones.

El objetivo general de este proyecto, comprende la migración a la nueva plataforma de los servicios básicos y generales que presta la Dirección: DNS centrales, Centrales telefónicas IP, Concentradores VPN, Servidores de autenticación y otros servicios prestados por la misma.

La plataforma permite a los desarrolladores y administradores concentrarse en el desarrollo de nuevas aplicaciones, abstrayéndolos de los procesos de actualización de sistemas y otras tareas que pueden generar una interrupción del servicio y otorgando a éstos las características más importantes y útiles que provee la virtualización de servidores.

Índice General

Resumen **2**

Índice General **3**

1. Introducción **5**

1.1. Marco conceptual 6
1.2. Objetivos de este trabajo 8
1.3. Motivación 9
1.4. Resultados esperados 10
1.5. Aportes de este trabajo 11

2. Descripción de las herramientas **13**

2.1. Proxmox VE 13
2.1.1. Inicios y descripción 13
2.1.2. Características 15
2.1.3. Utilidad 17
2.2. KVM/QEMU 17
2.2.1. Inicios y descripción 17
2.2.2. Características de KVM 18
2.2.3. Características de QEMU 20
2.2.4. Utilidad 20
2.3. LXC 20
2.3.1. Inicios y descripción 20
2.3.2. Características 22
2.3.3. Utilidad 23
2.4. ZFS 23
2.4.1. Inicios y descripción 23
2.4.2. Características 24
2.4.3. Utilidad 26
2.5. CEPH 27
2.5.1. Inicios y descripción 27
2.5.2. Características 28
2.5.3. Utilidad 29
2.6 Corosync Cluster Engine 30
2.6.1. Inicios y descripción 30
2.6.2. Características 30

2.6.3. Utilidad 31

3. Equipamiento 31
3.1. Introducción 31
3.2. Características del equipamiento 32
3.3. Arquitectura 35

4. Instalación de la plataforma 40
4.1. Configuración de placa RAID 40
4.2. Instalación de PVE 41
4.3. Licencia de la plataforma 41
4.4. ZFS 42
4.5. Detalles finales de la instalación 44
4.5.1. Zona horaria 44
4.5.2. Definición de password de administrador 45
4.5.3. Direccionamiento de Management 45

5. Configuración de la plataforma 47
5.1. Planteo general del esquema 47
5.2. Configuración inicial 48
5.2.1. Configuración del Cluster Corosync 49
5.2.2. Configuración del Cluster CEPH 52

6. Funcionamiento 61
6.1. Migraciones 63
6.2. Nuevos proyectos 64
6.2.1. Mail Gateway 64
6.2.2. Cluster Kubernetes 65
6.3. Nuevas funcionalidades 66

7. Conclusiones y trabajos futuros 66
7.1. Conclusiones 66
7.2. Trabajos Futuros 69

8. Bibliografía básica Relacionada 71

1. Introducción

El presente trabajo hace referencia a la elaboración de una infraestructura de virtualización utilizando como backend de almacenamiento compartido a los mismos equipos que luego virtualizarán servicios por medio de Máquinas Virtuales y Contenedores, generando así un cluster de equipos hiperconvergentes. Dicho cluster será utilizado para migrar la infraestructura de servicios existentes, que actualmente se encuentra en producción en equipos físicos y en su gran mayoría obsoletos.

La característica principal de este tipo de infraestructura es la posibilidad de prescindir de costosos equipos de storage, los cuales generalmente, son equipamientos de disponibilidad acotada en referencia a los utilizados, que son servidores de fácil acceso y económicos. Por otra parte, la capacidad de crecer en forma horizontal con equipamiento heterogéneo, permitirá a futuro una expansión de la infraestructura física permitiendo agregar más poder de almacenamiento y procesamiento en el cluster.

El objetivo general de este proyecto, comprende la migración a la nueva plataforma de los servicios básicos y generales que presta la Dirección Provincial de Telecomunicaciones, los mismos comprenden DNS centrales, Centrales telefónicas IP, Concentradores VPN, Servidores de autenticación y otros servicios prestados por la misma. Estos servicios tal como se describió anteriormente, están siendo ejecutados en equipos obsoletos, por lo cual se hace necesaria la migración inmediata de los mismos.

1.1. Marco conceptual

1. Generalidades

El marco conceptual “está compuesto de referencias a sucesos y situaciones pertinentes a resultados de investigación, incluye por tanto, un marco de antecedentes, definiciones, supuestos, etc” [1].

El marco conceptual es “un conjunto de definiciones, teorías, conceptos, sobre los temas que estructuran el desarrollo de la investigación y que sirven para interpretar los resultados que se obtengan del trabajo realizado en campo” [2].

En consecuencia, el marco conceptual está constituido por las definiciones de algunos conceptos que permiten ubicar su investigación en un campo específico. Sin embargo, no consiste en solamente una lista de definiciones o glosario, se supone que éstas hacen parte de una trama teórica, es decir, de un marco que las una, que establezca relaciones. El marco conceptual permitirá de esta manera identificar las palabras clave de la investigación.

2. Definición de conceptos

1. Infraestructura Hiperconvergente

El concepto de infraestructura hiperconvergente fue introducido por Arun Taneja en 2012 para describir la plataforma HC3 de la empresa Scale Computing. En esta definición dada por el autor, hizo hincapié en la diferencia respecto a una infraestructura convergente, ésta radica en el hecho de que una infraestructura hiperconvergente genera un enfoque centralizado donde ofrecen virtualización de máquinas, SAN virtual y redes virtualizadas, todo desde una interfaz de administración centralizada y unificada.

Si bien el esquema HCI puede contener otras características avanzadas, como ser un sistema de Backups y Disaster Recovery, este proyecto se enfocó sólo en las características mínimas dadas en el párrafo anterior, éstas son: Virtualización de Máquinas, SAN virtual y Redes Virtualizadas.

Desde esta perspectiva, la virtualización “crea un entorno informático simulado, o virtual, en lugar de un entorno físico. A menudo, incluye versiones de hardware, sistemas operativos, dispositivos de almacenamiento, etc. Ésto permite a las organizaciones particionar un equipo o servidor físico en varias máquinas virtuales” [3].

Una SAN es definida como “cualquier red de alta performance cuyo propósito primario es habilitar a los dispositivos de almacenamiento a comunicarse con servidores, computadoras y también entre sí” [4].

El concepto de redes virtualizadas es definido como “un método para combinar los recursos disponibles en una red para consolidar múltiples redes físicas, dividir una red en segmentos o crear redes de software entre máquinas virtuales” [5].

Por último, la infraestructura fue definida como “el conjunto de software y hardware sobre el que se soportan los servicios de una organización para responder eficientemente a las necesidades de los consumidores, actualizar los planes de control o supervisión y optimizar la cooperación con proveedores y clientes” [6].

2. Cluster

La definición que se utilizará sobre cluster está definida como “cualquier conjunto de elementos operacionales independientes, integrados por algún medio de comportamiento coordinado y cooperativo” [7].

3. Alta Disponibilidad

Primeramente se definirá qué es la disponibilidad. La definición dada por la RAE es “Cualidad o condición de disponible”. Teniendo en cuenta la definición anterior, el concepto de alta disponibilidad en IT es definido como “un sistema que está continuamente operativo y disponible para la prestación de servicios que brinda a los usuarios finales” [8].

4. Almacenamiento Definido por Software

Se comenzará definiendo el concepto de Almacenamiento, donde se darán dos definiciones para complementarlas. Según un portal de internet llamado Techopedia el almacenamiento es “un proceso mediante el cual los datos digitales se guardan dentro de un dispositivo de almacenamiento de datos mediante tecnología informática”, por otra parte, el mismo sitio dá una definición de la siguiente forma, “El almacenamiento es un mecanismo que permite que una computadora retenga datos, ya sea temporal o permanentemente” [9].

Para terminar de definir el concepto general de almacenamiento definido por Software se utilizará la definición de la empresa RedHat que la define como “una arquitectura de almacenamiento que separa el software de almacenamiento de su hardware. A diferencia del almacenamiento conectado a la red (NAS) tradicional o de los sistemas de red de área de almacenamiento (SAN), el Almacenamiento definido por Software por lo general está diseñado para ejecutarse en cualquier sistema x86 o estándar del sector, y de esa manera el software no depende del hardware propietario” [10].

1.2. Objetivos de este trabajo

El objetivo general de este proyecto, comprende la migración a la nueva plataforma de los servicios básicos y generales que presta la dirección: DNS centrales,

Centrales telefónicas IP, Concentradores VPN, Servidores de autenticación y otros servicios prestados por la misma. Asimismo, se desprenden los siguientes objetivos específicos que serán descritos a continuación:

a. Proveer la disponibilidad de recursos a los distintos departamentos, los cuales tienen necesidades de recursos informáticos.

b. Generar las pruebas y configuraciones pertinentes para otorgar alta disponibilidad a las máquinas virtuales y contenedores de los servicios prestados por la dirección.

c. Lograr otorgar a la Dirección Provincial de Telecomunicaciones la autonomía de los recursos.

d. Disponibilizar todas las bondades de la virtualización a los administradores de los servicios que van a utilizar la nueva infraestructura.

e. Otorgar al administrador la posibilidad de actualizar el software en forma transparente al usuario, utilizando la tecnología de Snapshots y Clones prestadas por la nueva infraestructura.

1.3. Motivación

Debido a los cambios que se están presentando en relación a varios de los servicios prestados por la Dirección Provincial y considerando que el equipamiento actual donde dichos servicios están funcionando ya se encuentra obsoleto, el área de servicios de red propone realizar un proyecto en el cual se utilice el hardware obtenido por la licitación de la nueva red, creando un Cluster con alta disponibilidad, tomando como premisa el uso de software libre.

Dicho Cluster servirá como plataforma de virtualización para dar servicio no solo a la dirección sino también, y de manera general, a toda la red provincial, siendo lo más urgente la migración de los DNS centrales, tanto internos como externos. Por otro lado el sistema de relay de correos, que si bien tiene disponibilidad desde varios equipos, están sujetos a que los mismos no puedan ser actualizados fácilmente, ocupando a los administradores en tareas arduas que al día de hoy con la virtualización se han mitigado.

A su vez, la Red Provincial se encuentra en fase de migración hacia la nueva red, la cual actualmente convive con la red legacy en forma aislada y donde aún se encuentran los servicios centrales y transversales. Contar con dicho cluster permitirá fácilmente tener funcionando varios servicios en ambas redes y que su administración sea sencilla, pudiendo acompañar el proceso de transición hacia la nueva red.

1.4. Resultados esperados

a- Migración total de los servicios centrales: DNS, relays de correo, proxies centrales, centrales telefónicas y otros servicios prestados por la Dirección Provincial de Telecomunicaciones.

b- Alta disponibilidad de servicios.

c- Beneficios de la virtualización en general, snapshots, backups, migración en vivo, etc.

d- Administración centralizada de recursos.

e- Crecimiento a futuro en forma horizontal y hardware heterogéneo.

1.5. Aportes de este trabajo

En esta sección del trabajo, se intentará plasmar, de forma concisa, un procedimiento general para quien desee implementar y migrar hacia una infraestructura hiperconvergente basado en las herramientas Open Source, intentando también dar sugerencias, las cuales fueron estudiadas y aprendidas en la elaboración de este proyecto.

Un punto fundamental es la necesidad de poseer al menos tres equipos del tipo servidor, esto es una necesidad limitante a la hora de diseñar este tipo de clusters. Cada nodo es parte de un conjunto donde por medio de votos se mantiene el quórum, por lo cual es necesario poseer un mínimo de 3 nodos para que éste sea conciso.

En caso de utilizar discos del tipo enterprise y sobre todo si son de tecnología SSD o NVMe, es necesario que la interfaz de comunicación de CEPH sea por un medio físico de fibra a una velocidad mínima de 10 Gbps. Si se realiza una conectividad por debajo de ésta, la infraestructura puede verse saturada en performance con posibilidad de tener fallos. Ésto siempre es una recomendación si el cluster vá a ser utilizado en producción.

Si bien parece algo obvio, es recomendable tener una capacidad de memoria principal con grandes márgenes de sobra. Proxmox VE, en conjunto con las tecnologías que utiliza, es muy dependiente de la memoria principal y su performance se basa en que ésta sea de gran capacidad. Tecnologías como ZFS y CEPH utilizan muchísimos recursos de memoria para realizar caché de los datos, cálculos de integridad, entre otras características que si el sistema no dispone la performance caerá drásticamente.

Con respecto a las migraciones de los servicios hacia la infraestructura hiperconvergente, es importante saber que Proxmox VE soporta archivos del tipo VMDK, es decir que si se posee una infraestructura basada en VMWare se podrá fácilmente realizar una migración del disco y posteriormente encender la Máquina Virtual sobre la nueva infraestructura. Con respecto a los equipos físicos, es posible utilizar herramientas del estilo CloneZilla [11], y pasar los datos por red a la nueva Máquina Virtual. En este caso, se ha podido realizar con centrales telefónicas basadas en Issabel [12].

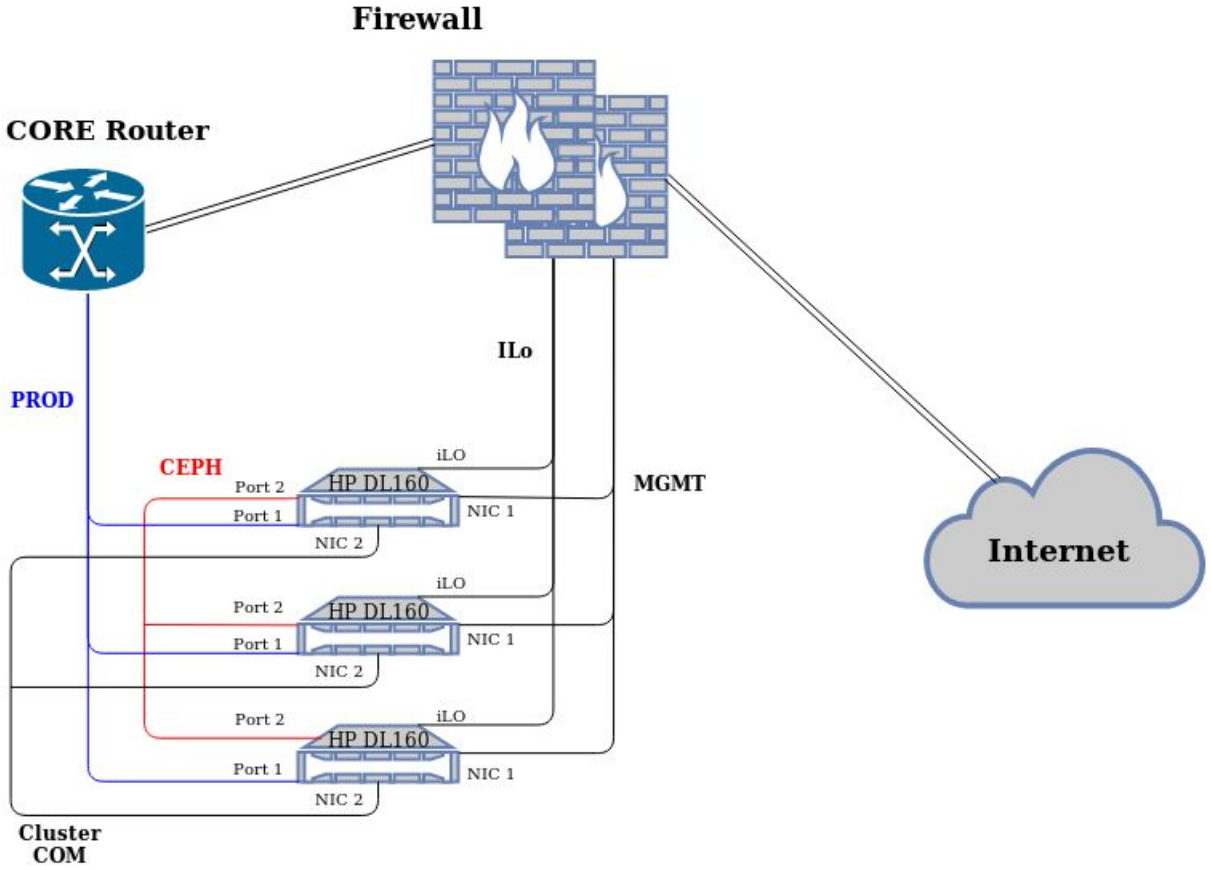

Figura 1 - Estado Actual de la red y cluster

2. Descripción de las herramientas

2.1. Proxmox VE

2.1.1. Inicios y descripción

Proxmox Virtual Environment (Proxmox VE) es la herramienta central y la más importante de este proyecto, por ende el punto de partida del mismo. En primer lugar, y en referencia a su historia, se pudo ver que la empresa detrás de la plataforma de virtualización Proxmox VE, nació en el año 2005; su nombre: Proxmox Server Solutions GmbH. Sus fundadores, Martin y Dietman Maurer, lanzaron la primera versión estable en el año 2008, y a lo largo de los años fueron incorporando nuevas características al entorno, como ser, en el año 2012 se incorporó el cluster de alta disponibilidad (HA), basándose en tecnología de Red Hat y Corosync [13].

La licencia con la cual se publica PVE es GNU Affero General Public License (V3), la misma otorga la libertad de distribuir el código sin restricciones, por lo cual se podrá cambiar el código para obtener un rendimiento específico a nuestras necesidades, distribuirlo, y comercializarlo [14].

Proxmox VE es una plataforma que contiene un conjunto de herramientas Open Source totalmente integradas, para lograr un sistema completo de virtualización con características empresariales. A la fecha, Proxmox VE se encuentra en su versión 6, la cual ha logrado, en base al tiempo, una madurez y estabilidad muy alta; así como también se han agregado muchas características que hacen de este software una poderosa herramienta para virtualizar.

Su empresa nos sirve de esta plataforma a través de un instalador (ISO), el cual puede instalarse en hardware del tipo servidor, hardware hogareño o en máquinas virtuales, en éste último, no se podrá aprovechar la virtualización por hardware (Intel V-TX y AMD SVM). La instalación de la plataforma es muy sencilla, con un formato gráfico del tipo wizard, donde a través de los menús, se pueden seleccionar opciones, como ser, nombre de host, contraseña del usuario root, tipo de almacenamiento; es muy similar a una instalación de Sistema Operativo GNU/Linux. Es importante destacar que Proxmox VE es un hipervisor de tipo 2, instalado sobre un Sistema Operativo GNU/Linux Debian, que brinda la posibilidad de tener la flexibilidad del SO GNU/Linux y la potencia de esta plataforma.

La finalidad de este tipo de sistema es aprovechar el hardware con el software, generalmente los servidores físicos pueden generar subutilización de recursos, en muchos casos no superando el 20% del mismo. La virtualización logra reducir costos en energía, y compras de nuevo hardware, aprovechando al 100% los equipos existentes. A su vez, simplifica todo tipo de operaciones, como ser, reparaciones, actualizaciones, y otras tareas que pueden generar incluso una interrupción del servicio [15]. Cabe destacar, que las herramientas Opensource, que esta plataforma integra, son muchas y variadas, cada una otorga a Proxmox VE una característica para hacer de ella un sistema capaz de realizar muchas tareas del tipo empresarial. Se pasará a nombrar cada herramienta y luego se desarrollará con más detalle:

- KVM/QEMU: Virtualización y Emulación.
- LXC: Contenedores.
- ZFS: Storage no compartido.
- CEPH: Storage compartido.
- Corosync: Comunicación de cluster HA.

En este proyecto particularmente se utilizaron, para la tarea de networking, los linux bridges. Si bien Proxmox VE soporta también otra plataforma, como ser Open VSwitch (OVS), se eligió por sobre éste debido a su facilidad de uso y a su vez

porque no se necesitaron las características que sí provee OVS, como ser, RSTP, VXLAN, Openflow entre otras [16].

Más allá de las características ofrecidas en base a la integración de las herramientas mencionadas, la plataforma en sí provee una interfaz de comunicación para el usuario, de modo que la administración de la herramienta sea visual y sencilla. Lo importante es que si se necesita realizar un chequeo más profundo, también provee del acceso a la consola donde un administrador experimentado puede sacar lo mejor del SO Debian GNU/Linux.

2.1.2. Características

a- CLI y consola Web: Proxmox VE otorga al administrador accesos vía Web y CLI. Particularmente la Web se ha ido puliendo con el tiempo y en la actualidad casi todas las operaciones se pueden realizar desde este sector, donde se acota al usuario a tareas específicas y se reducen los riesgos de cometer errores. Por su parte la CLI, claramente es más dinámica y no está circunscripta a la herramienta en si, es por ello que el administrador deberá hacerlo con precaución en servidores en producción.

b- Stand-alone o cluster: La plataforma permite administrar servidores en formato stand-alone, donde cada uno tiene su propia Web GUI y cada operación se circunscribe a ese equipo. Por otro lado, también brinda la posibilidad de generar un Cluster entre 2 o más equipos, donde luego de generarlo se podrá ingresar a una Web central de administración.

c- Migraciones en vivo: Proxmox VE permite hacer migraciones en vivo, tanto en almacenamiento compartido como en almacenamiento local, es decir, el disco puede estar local en uno de los nodos del cluster y Proxmox VE permite hacer esta migración a otro nodo en vivo; ya no es necesario tener un almacenamiento compartido para realizar esta operación.

d- Alta disponibilidad: Proxmox VE permite realizar una configuración de alta disponibilidad de los recursos, sean máquinas virtuales o contendores, para lograr una disponibilidad de los servicios en caso de una catástrofe. Para ésto es necesario anteriormente generar un cluster.

e- Storage auto-administrado: La plataforma permite la administración del storage, tanto desde la Web GUI como de la consola. En ella se pueden generar volúmenes LVM, LVM-Thin, ZFS Pools, CEPH pools, entre otros tipos de storage.

f- Backups: Proxmox VE otorga la posibilidad de realizar full backups de las Máquinas Virtuales y Contenedores, ya sea en forma manual como generando un cronograma. Ésto se realiza tanto desde la Web GUI como desde CLI. El archivo que produce es autocontenido, donde luego puede ser restaurado en otro equipo si es necesario.

g- Snapshots: esta característica es muy importante a la hora de tener dinamismo para hacer cambios complejos dentro de una máquina virtual o contenedor. Aprovecha las características de varios File System que actualmente pueden generar snapshots para que el administrador desde la Web GUI las utilice fácilmente y pueda volver al punto anterior en caso de ser necesario.

h- API REST: Debido al auge de la automatización por medio de herramientas como por ejemplo Ansible, Chef u otras que pueden ser desarrolladas por uno mismo, el uso de APIs se ha extendido ampliamente. Proxmox VE otorga la posibilidad de poder comunicarnos con su potente API, donde se puede lograr desde obtener información de una máquina virtual hasta apagar o encender un contenedor.

i- Soporte y capacitación: La empresa Proxmox Server Solutions GmbH otorga la posibilidad de contratar un servicio de soporte y a su vez, puede realizar capacitaciones de la administración de la herramienta [17].

2.1.3. Utilidad

La utilidad central de la plataforma Proxmox VE, es la posibilidad de tener trabajando mancomunadamente a todos los proyectos Open Source, para lograr entregar un producto fácil de administrar y con características avanzadas, que en la actualidad son necesarias para el crecimiento de cualquier organismo, sea público o privado.

2.2. KVM/QEMU

2.2.1. Inicios y descripción

En primer lugar, un breve análisis de la historia de ambas herramientas. KVM (Kernel-Based Virtual Machine) fue creado en 2006 por un programador llamado Avi Kivity, como un módulo de kernel Linux [18]. En principio KVM solo soportaba las instrucciones V-TX de Intel, utilizada para generar virtualización por hardware. Con el tiempo se adoptaron las instrucciones de AMD SVM. Si bien el patch al kernel se lanzó en 2006, fue adoptado en la versión 2.6.20 por el año 2007. QEMU (Quick Emulator) también fue desarrollado en el año 2006 por Fabrice Bellard; donde posteriormente se creó la comunidad de desarrolladores de QEMU [19].

La licencia de estos software vá en conjunto con la licencia del código Linux, ambas utilizan la licencia GPL, donde el usuario es libre de hacer del código lo que desee, ésto es, comercializarlo, utilizarlo, distribuirlo y modificarlo.

La sigla KVM, significa Kernel-based Virtual Machine, como se mencionó anteriormente fue desarrollado como módulo de Kernel. Linux adoptó este modo

no-monolítico donde los desarrolladores pueden generar módulos que son cargados dinámicamente al kernel gracias a la nueva estructura que nos provee.

KVM es uno de los proyectos de virtualización Open Source más utilizados. Gran parte de los proyectos más grandes open source que realizan virtualización, cómo puede ser Openstack o cloudstack, utilizan a KVM como su estándar de virtualización. Proxmox VE también hace uso de este virtualizador como estándar de facto, no pudiendo utilizar otro. Openstack en cambio tiene la posibilidad de utilizar otros, sean o no Open Source, como puede ser el virtualizador de la empresa VMWare, entre otros.

Proxmox VE utiliza la dupla KVM/QEMU para lograr la virtualización de servidores, con estas dos herramientas convierte al sistema operativo Debian GNU/Linux en un hipervisor capaz de generar máquinas virtuales con paravirtualización y virtualización completa. KVM particularmente se basa en tres módulos: kvm.ko, kvm-intel.ko y kvm-amd.ko. La función de estos módulos consiste en aportar virtualización por hardware aprovechando las extensiones de los procesadores actuales de la empresa Intel y AMD.

La plataforma también hace uso de QEMU, que funciona como interfaz entre KVM y el kernel con el espacio de usuario. QEMU es un emulador y virtualizador de hardware que puede o no apoyarse en un virtualizador como puede ser KVM o XEN para lograr un rendimiento casi similar a un hardware para máquinas virtuales. QEMU logra presentarle toda la emulacion del hardware a KVM para lograr así la virtualización tal cual hoy en día se la conoce. No solo emula CPU, sino también periféricos, discos, placas de red, PCI, entre otros.

2.2.2. Características de KVM

a- KSM (Kernel Same Page Merging): Es una herramienta capaz de reutilizar páginas de memoria entre las máquinas virtuales. Es conveniente su utilización

cuando se poseen máquinas virtuales donde sus aplicaciones generan muchas instancias del mismo dato. Ésto provoca una reducción en el uso de memoria principal, aprovechado mejor sus recursos que generalmente son limitados.

b- CPU Hot-Plug: Es la capacidad de poder agregar más vCPU (virtual CPU) a una máquina virtual sin necesidad de apagarla. Para esta función es necesario que la máquina virtual tenga habilitada esta característica.

c- PCI Hot-Plug: Similarmente al punto anterior, también es posible agregar dispositivos PCI sin necesidad de apagar la máquina virtual.

d- Migración en vivo: KVM soporta migración de máquinas virtuales, sea en línea o fuera de línea. Lo importante a saber es que KVM es independiente del Sistema Operativo que esté funcionando en la máquina virtual, así como también es independiente del hardware destino, es decir, KVM podrá hacer la migración de un host físico con hardware Intel a uno AMD.

e- VirtIO: Driver de altas prestaciones que emulan dispositivos IO para KVM. Este driver elimina la emulación de hardware para máquinas virtuales y en su lugar genera una API, donde la performance aumenta considerablemente.

f- Periféricos: En algunos casos es conveniente la utilidad de poder emular dispositivos de sonido, USB, CDROM, entre otros. KVM actualmente permite cada uno de éstos.

g- PCI Passthrough: Permite realizar una conexión directa de una máquina virtual con el hardware PCI conectado al servidor físico. De esta forma se podrá conectar por ejemplo una placa FXO/FXS para que la central telefónica virtual pueda hacer uso de este hardware. Cabe destacar que una vez realizado, la máquina física no podrá hacer uso de este dispositivo [20].

2.2.3. Características de QEMU

a- Modo Emulación/Virtualización: QEMU puede funcionar como un emulador total del sistema, éste no aprovecha la virtualización por hardware. Por otro lado, puede funcionar en conjunto con KVM para lograr una virtualización completa.

b- Multiplataforma: Soporta un gran abanico de sistemas operativos, como ser, GNU/Linux, *BSD, MAC OS y Windows.

c- Emulación de dispositivos: Una gran variedad de dispositivos pueden ser emulados: Dispositivos seriales, paralelos, USB, discos, etc.

d- Soporte para SMP (symmetric multiprocessing): En conjunto con KVM se podrá utilizar más de un procesador en la máquina virtual [21].

2.2.4. Utilidad

KVM/QEMU es utilizado para convertir a GNU/Linux en un hipervisor. Éste se beneficia de las características del sistema Linux para aprovechar sus mejoras de performance, el scheduler, el manejo de memoria, entre otras funciones existentes; para lograr así una plataforma de virtualización segura, estable y escalable.

2.3. LXC

2.3.1. Inicios y descripción

Para comenzar es menester definir qué son los contenedores. Éstos son procesos que parten de una imagen de sistema operativo, donde agrupan un conjunto de

aplicaciones y recursos, separados del host físico, logrando aislarse del mismo y generando la sensación que se está ejecutando otra máquina. Lo interesante de ésto, es que partiendo de la misma base de sistema operativo se pueden correr muchos contenedores diferentes en el mismo host físico [22].

LXC (LinuX Containers) nació en el año 2008, él mismo pudo tomar características existentes en el kernel de linux y sin necesidad de un hipervisor, otorga al administrador lo necesario para generar contenedores. La licencia con la cual LXC es lanzado es LGPL, que si bien es similar a GPL permite, a diferencia de ésta, la asociación con otros programas que pueden ser no libres.

Como se mencionó en el párrafo anterior, LXC utilizó características existentes en el kernel para lograr que sea factible la utilización de contenedores en el sistema GNU/Linux, para ésto se valió de las siguientes herramientas [23]:

1- CGroup (Control Groups): son utilizados para realizar un control de los recursos consumidos por procesos ejecutados en una máquina; tal como se explicó anteriormente, un contenedor es un proceso, por lo cual desde el kernel se podrán asignar recursos como ser, cantidad de memoria, número de procesadores, tamaño de disco, entre otras. Ésto genera algo similar a lo que se puede hacer con una máquina virtual, donde los recursos son seleccionados por el administrador.

2- Namespace: éstos son utilizados para aislar el proceso en sí, es decir, proveen al proceso de su propia vista del sistema. Son los encargados de limitar lo que el proceso (contenedor) puede ver, a diferencia de los CGroups que son los encargados de limitar los recursos al proceso. LXC utiliza varios namespaces para generar un contenedor:

- pid Namespace: Aísla los pid de los contenedores.
- net Namespace: Genera un networking propio para cada contenedor.

- mnt Namespace: Genera puntos de montajes propios al contendor.
- uts Namespace: Relacionado al nombre de host del contenedor.
- user Namespace: Mapea los UID/GID del contendor con el host físico.

3- File System root o rootfs: Apoyándose en el mnt Namespace, LXC provee un rootfs sin el kernel, ya que el contenedor utiliza el kernel del host físico. Ésto dá la sensación de poseer un file system propio con la estructura convencional de un File System POSIX de Linux.

2.3.2. Características

a- Livianos: Comparados con una máquina virtual, los contenedores son más livianos, es decir, utilizan menos recursos debido a que no deben implementar ningún tipo de emulación de hardware y aprovechan las características que el kernel de Linux del host físico ya posee.

b- Sin Hipervisor: Los contenedores no necesitan de un hipervisor para ejecutar, ya que éstos son procesos como cualquier otro que el sistema anfitrión esté corriendo.

c- Maximización de recursos: Dado a su naturaleza más liviana que las máquinas virtuales, se podrán ejecutar más contenedores que máquinas virtuales en un mismo host.

d- Inicio y apagado: Los contenedores tienen tiempos de inicio y apagado instantáneos, ya que no necesitan corroborar nada de hardware para iniciar y apagarse.

e- No instalación: Los contendores en sí, no necesitan ser instalados, ya que los mismos parten de una imagen base, por ejemplo, un ubuntu server, donde a partir de ella luego se podrán iniciar muchos contenedores.

f- API: Tiene una potente API, donde se pueden utilizar librerías en lenguajes de programación de alto nivel, como por ejemplo Python, y a través de código de programación administrar la herramienta LXC.

2.3.3. Utilidad

LXC brinda la posibilidad de iniciar contenedores de forma efectiva, estable, transparente y de fácil administración. Particularmente la plataforma Proxmox VE, posee una interfaz amigable, donde los contendores base (templates) son descargados, ejecutados y administrados tanto desde la Web GUI, como desde consola.

2.4. ZFS

2.4.1. Inicios y descripción

Con respecto a sus inicios, ZFS (Z File System) fue lanzado en el año 2005 por la empresa Sun Microsystems para su sistema Operativo Solaris. El mismo posee una licencia de software libre CDDL (Common Development and Distribution License), la cual es considerada como tal por la Free Software Foundation. Si bien este software no funciona en Linux, la comunidad comenzó a realizar una portación de código para que funcione sobre la plataforma GNU/Linux; el proyecto fue llamado OpenZFS, y su primera versión data del 2013 [24].

En el presente apartado se mencionan las características que posee un File System, éste se describe como el proceso de cómo y dónde la información es guardada. Básicamente es un software que hace de interfaz de comunicación entre el sistema operativo y los discos duros.

ZFS es un sistema de archivos moderno, el mismo fue creado para asegurar la integridad de datos donde una vez almacenados, por medio de checksums, los mismos sean íntegros y no se corrompan. Ésto lo puede conseguir debido a que el hardware actual logra desempeños altos, ya que muchas de las tareas son realizadas por el procesador y un alto uso de la memoria principal.

Si bien su nombre indica que es un File System, en realidad no solo es eso ya que también es un administrador de volúmenes, o sea es un sistema de archivos basado en pools al igual que LVM, el manejador de volúmenes por defecto de linux, que también es basado en Pools. Lo interesante es que estos pools pueden estar diseñados con discos funcionando como un RAID por software, ésto es, ZFS permite generar RAID del tipo 0, 1, 10, RAIDZ1, RAIDZ2 y RAIDZ3. Con ésto logra 3 características interesantes y todas funcionando mancomunadamente para dar integridad y disponibilidad de la información.

2.4.2. Características

a- Checksums: ZFS realiza un checksum de cada dato guardado en el disco, ésto es utilizado para corroborar si un dato se corrompió y es realizado en tiempo real por el sistema.

b- COW (copy on-write): El sistema fue pensado para utilizar de base el esquema COW, que si bien tiene algunas desventajas, aporta mucho más en lo relacionado a la administración y flexibilidad. En el esquema COW, cuando un dato necesita ser sobreescrito, el mismo no sobreescribe el dato en sí, sino que asigna un nuevo bloque de disco, y por medio de un sistema de punteros, asigna la posición al bloque nuevo. Ésta es la base central para el uso de snapshots.

c- Snapshots: ZFS permite en forma por defecto el uso de snapshots sobre sus dos tipos de storage, datasets (File device) y zvols (Block Device). Ésto es gracias al esquema COW del cual se habló en el punto anterior.

d- Pools: ZFS es una sistema basado en pool (Pooled storage). Similarmente a LVM, se generan pools, donde el administrador podrá agregar más discos físicos al mismo, permitiendo hacer crecer dinámicamente el espacio del storage.

e- RAID: ZFS permite generar dentro de los pools dispositivos virtuales (vdevs), los cuales pueden funcionar como RAID por software. Los tipos de RAID son algunos de los convencionales (0, 1 y 10), por otro lado, también ofrece la posibilidad de generar los RAIDZ:

- RAIDZ1: con un disco de paridad.
- RAIDZ2: con dos discos de paridad.
- RAIDZ3: con tres discos de paridad.

f- Clones: Los clones de ZFS están basados en los snapshots, éstos permiten clonar por completo, ya sea un dataset o un zvol, generando información nueva y clonada. Ésto es muy útil en máquinas virtuales para realizar una clonación completa de una Máquina Virtual.

g- Replicación: Permite replicar información de un host a otro utilizando snapshots, ya sean full o incrementales. Proxmox VE utiliza esta tecnología para replicar los discos de máquinas virtuales a otro nodo del cluster en formato incremental, por medio de una tarea de replicación donde se puede indicar cada cuanto tiempo queremos hacerlo.

h- Caché: ZFS otorga la posibilidad de generar dos tipos de caché, la caché de lectura y la caché de escritura. Si bien es una característica que puede otorgar

beneficios de rendimiento, es necesario primero obtener métricas para determinar si es necesario hacerlo. El sistema fue pensado para trabajar con la memoria principal y la mejora en rendimiento es notoria cuando se aumenta la misma,.

i- Scrubbing: Como se mencionó anteriormente, ZFS genera checksums de la información guardada. La tarea de scrub, es básicamente un escaneo completo de los volúmenes buscando si un dato se ha corrompido, es decir, si un checksums es diferente sobre el mismo dato. Esta tarea es necesaria y costosa a nivel recursos computacionales, por lo cual es conveniente realizarlo cuando la demanda de servicios es baja. En caso de encontrar un dato corrupto intentará repararlo en tiempo real.

j- Compresión y deduplicación: La última característica reúne dos puntos, uno es la compresión de datos y el otro la deduplicación. La compresión significa poder comprimir los datos, ésto hoy en día prácticamente no agrega overhead en el procesamiento así que es una característica casi estándar en las instalaciones de ZFS. Por su parte, la deduplicación intenta aprovechar datos duplicados y solo ocupar el espacio en disco de ese bloque solo una vez. Es un esquema costoso, ya que utiliza grandes cantidades de memoria principal para garantizar tiempos de acceso óptimos a los datos [25].

2.4.3. Utilidad

La misma otorga al administrador, la posibilidad de obtener variadas características antes mencionadas para tener un sistema completo, donde el File System, el RAID y los volúmenes son administrados por una sola herramienta, la cual es capaz de repararse a sí misma en tiempo real y sin caídas.

Proxmox VE justamente aprovecha toda la potencia de ZFS para usarlo en replicaciones, snapshots, storage del tipo archivo y bloque.

2.5. CEPH

2.5.1. Inicios y descripción

CEPH nació en la Universidad de California como tesis doctoral de Sage Weil en el año 2003, para posteriormente en el año 2006, publicar el código bajo la licencia LGPL. En el año 2012, Sage Weil funda la empresa Inktank donde genera grandes mejoras sobre el código inicial de CEPH y debido a su notoriedad en el año 2014 fue adquirida por la empresa RedHat [26].

CEPH fue creado con el principal objetivo de lograr la consistencia y la protección de los datos por sobre la disponibilidad de los mismos; con disponibilidad se hace referencia al acceso a los datos de los usuarios que utilizan el sistema. Por lo cual el mismo fue diseñado como un sistema distribuido, donde los metadatos y los datos están replicados en varios nodos del cluster, de modo que ante una eventual catástrofe, CEPH podrá recuperarse por medio del algoritmo RADOS. Dicho algoritmo se encarga de distribuir los datos dentro del sistema y éste es almacenado en una tabla de búsqueda (CRUSH Maps), la misma es replicada dentro de los nodos monitor del cluster, eliminando así el punto de falla.

Una de las facultades que tiene CEPH, es la de proveer todos los tipos posibles de almacenamientos existentes, como ser, almacenamiento por objetos, almacenamiento por bloques y almacenamiento de archivo. Otra característica interesante de CEPH es la posibilidad de que, por medio de APIs, permite comunicarse y realizar operaciones con el cluster CEPH, directamente con las aplicaciones; ésto elimina otras capas que pueden generar overhead, ya que directamente se están comunicando con la plataforma de almacenamiento [27].

Desde la aparición de Openstack como proyecto Open Source de nube pública y privada, CEPH ha logrado más popularidad debido a su robusto sistema de replicación de datos. Distintas encuestas muestran el crecimiento en la adopción de CEPH, posicionándolo como líder en la elección como plataforma de almacenamiento para Openstack, debido a su escalabilidad y fuerte integración con los 3 proyectos de almacenamiento: Cinder (block storage), Swift (Object Storage) y Manila (File Storage).

2.5.2. Características

a- Hardware: Un punto fuerte de esta plataforma es la posibilidad de utilizar hardware asequible y económico, ya que CEPH puede ser ejecutado en cualquier servidor e incluso en PC. Por otra parte, se puede utilizar el sistema por medio de máquinas virtuales, este tipo de pruebas con equipamientos dedicados de marcas reconocidas es casi imposible de lograr.

b- Almacenamiento Definido por Software: CEPH fue pensado para ser de este tipo desde sus inicios. Ésto permite realizar muchas tareas de automatización e integración con otras herramientas. Tanto Proxmox VE como Openstack hacen uso de esta característica para generar la comunicación con las instancias de máquinas virtuales.

c- Tipos de almacenamiento: Como ya se mencionó, CEPH permite generar los tres tipos existentes de almacenamiento en el mercado, éstos son: Almacenamiento de objetos, de bloque y de archivo.

d- No RAID: CEPH fue pensado desde otro modelo, por lo cual no es necesario que los discos formen parte de un sistema RAID. Los datos están replicados en varios sitios, quitando la necesidad de un sistema de paridad para la recuperación.

e- Conexión: Como se mencionó anteriormente, CEPH otorga la posibilidad de conexión directa contra la plataforma por medio de la librería libRADOS. La misma soporta varios lenguajes de programación, donde se pueden realizar distintas tareas, directamente con el cluster, sin necesidad de ejecutar la aplicación sobre una máquina virtual donde el almacenamiento final está sobre CEPH.

f- Tolerancia a fallas: CEPH está preparado para la caída completa de uno o más servidores, siempre y cuando el diseñador del storage haya tenido en cuenta esta posibilidad.

g- Auto curación: Por medio del algoritmo RADOS de CEPH, éste logra ante una caída la posibilidad de, sin intervención del administrador, recuperar los datos de un componente dañado; genera nuevas réplicas dentro del cluster asegurando así la disponibilidad y durabilidad de los datos.

h- Hardware heterogéneo: Debido a que CEPH es un tipo de almacenamiento definido por software, ésto no genera una dependencia con una marca. La plataforma permite operar sobre distintos tipos de servidores, tarjetas de red, discos duros, sean mecánicos o SSD.

i- API REST: Posee una API bien definida por medio de RADOS Gateway, donde es posible utilizarla por medio de aplicaciones como almacenamiento de objetos. Así mismo podemos integrarla al proyecto de Openstack Swift y a S3 de amazon.

2.5.3. Utilidad

CEPH permite generar, a través de software, un storage distribuido y heterogéneo que logra replicar la información y a su vez brinda una gran variedad de herramientas para accederla. Proxmox VE, por su parte, en las últimas versiones, lo

ha incorporado a su Web GUI, por ende instalarlo ahora es mucho más sencillo y ponerlo en producción también.

2.6 Corosync Cluster Engine

2.6.1. Inicios y descripción

A modo de introducción histórica Corosync nació en 2008 basándose en el proyecto OpenAIS. Este código se mantiene por la comunidad Open Source y está establecido bajo la licencia BSD 2.0. La misma es aprobada por la Free Software Foundation como licencia de código abierto y permite la incorporación de software no libre.

Corosync Cluster engine es un protocolo para comunicación de Clusters. Utiliza el protocolo de TÓTEM, el cual genera un anillo entre los nodos y permite la comunicación efectiva entre ellos. El diseño del protocolo TÓTEM se basaba en broadcast. Corosync en cambio, utiliza multicast para la comunicación sobre el protocolo UDP, donde hace la difusión de tokens entre los nodos [28].

Corosync genera, de acuerdo a los miembros del cluster, un quórum. Éste se logra basándose en la cantidad de nodos que se están comunicando (votando); lo cual permite saber el estado de salud del cluster, para determinar qué hacer en caso de que éste se pierda.

2.6.2. Características

a- Confiabilidad: Genera un mecanismo de comunicación efectiva, otorgando robustez al sistema de envío y recepción de mensajes de los nodos.

b- Ordenamiento de mensajes: El protocolo permite realizar, en caso de sufrir un desorden de la llegada de mensajes, un ordenamiento de los mismos en tiempo real.

c- Multicast o Unicast: Permite al administrador la elección del protocolo de difusión de mensajes. Corosync igualmente recomienda el uso de multicast.

d- Simple o Redundante: El protocolo de Corosync permite tener redundancia en el vínculo de conexión, ésto es muy útil para justamente obtener la redundancia necesaria evitando puntos de fallas que puedan producirse en uno de los vínculos.

f- Detección de fallas: El protocolo es capaz, de acuerdo al quórum, de determinar qué nodo está fallando.

2.6.3. Utilidad

Es un esquema ampliamente utilizado en la generación de clusters en GNU/Linux. Su sencillez de configuración hace que sea atractiva para la adopción de base para la comunicación de Clusters. Proxmox VE se basa en esta herramienta para generar su cluster de virtualización de alta disponibilidad, poniendo a Corosync como base en la comunicación del mismo.

3. Equipamiento

3.1. Introducción

Este punto del proyecto, estará diagramado de la siguiente forma: primero se realizará una introducción con respecto a la arquitectura general elegida y a su vez

el equipamiento adquirido, donde por medio de gráficos se mostrará un panorama de cómo se construyó el cluster. Luego se desarrollará la forma en la cual se instaló y configuró toda la plataforma, comentando también algunos pormenores ocurridos y detectados antes de salir a producción (migraciones sobre placa de mangement, disco fallado). Por último se presentarán estadísticas y funciones, como por ejemplo: migraciones en vivo, backups, snapshots, replicación, etc.

3.2. Características del equipamiento

La Dirección Provincial de Telecomunicaciones, adquirió en una licitación pública para la nueva Red Única Provincial de Comunicación de Datos (RUPCD), servidores de la plataforma Proliant de la marca Hewlett Packard (HP). Dichos servidores fueron licitados con el fin de prestar servicios de red, como ser DNS, SMTP relay, telefonía IP, entre otros. Los servicios estaban funcionando en servidores físicos ya obsoletos y sin soporte de la empresa HP, otros en cambio, se encontraban virtualizados en un espacio de recursos asignados a la dirección por un proveedor externo.

Figura 2- Imágen ilustrativa del servidor HP DL160 G9

Los servidores adquiridos son de la línea Proliant DL160 G9, los mismos son rackeables y de una unidad de rack. Si bien se compraron más equipos en la licitación, los mismos serán distribuidos por los distintos centros de datos que serán instalados en la Provincia de Buenos Aires.

Para este proyecto se utilizaron tres de ellos y los mismos disponen de las siguientes características:

- Procesador: Cada servidor está equipado con dos procesadores de la marca Intel(R) Xeon(R) CPU E5-2650 v4. Sus especificaciones son las siguientes [29]:

Velocidad del procesador	2200 MHz
Cantidad de núcleos - subprocesos	12 Cores - 24 Threads
Memoria Caché	30 MB
Cantidad de QPI (QuickPath Interconnect)	2
Virtualización Intel VT-x	Sí

- Memoria Principal (RAM): El servidor está equipado con un total de 512 GB; cada banco de memoria tiene una capacidad de 64 GB direccionando la suma de 4 bancos por procesador. Las memorias poseen la siguiente descripción:

Tipo de memoria	DIMM DDR4
Capacidad	65536 MB
Tecnología	LRDIMM
Frecuencia Máxima	2400 MHz

- Energía: El mismo viene equipado con doble fuente y una batería de respaldo que realiza un backup de la caché de escritura en una memoria flash, si es que se pierde la energía.

- Almacenamiento: Los servidores están equipados con una controladora RAID, modelo Smart Array P440, la misma tiene las siguientes características [30]:

Tipo de Conectividad	SAS
Caché	2 GB
Respaldo en flash	Sí
Niveles RAID	0, 1, 10, 5, 50, 6, 60, 10

- Discos Duros: Cada uno de los servidores posee en total, cuatro discos duros de estado sólido de clase empresarial, los cuales se detallan a continuación:

Capacidad	3.84 TB / 3840 GB
Tipo de carga de trabajo	Lectura intensiva
Tipo de interfaz de conexión	SAS
Tipo de enchufe	Hot-Plug

- Administración Remota: Se contrató para cada uno de los equipos, licencia de administración remota de la marca HP iLO (Integrated Lights-Out) Advanced, que incluye muchas características para que su administración sea fácil, segura y fuera de carga.

- Conectividad: Este ítem en particular se dejó para el final, debido a que tiene una relevancia muy importante en la arquitectura elegida, donde las tareas que el hipervisor realiza fueron divididas físicamente en cada placa de red. Los equipos poseen las siguientes tarjetas de red:

1- HPE Ethernet 1 GB de dos puertos: uno fue utilizado para realizar el management del hipervisor Proxmox VE, y la otra para la comunicación del Corosync Cluster Engine.

2- HPE Ethernet 10 GB de dos puertos SFP: Uno de los puertos fue utilizado para la comunicación del cluster CEPH, y la otra por medio de VLANs fue utilizada como interfaz de producción.

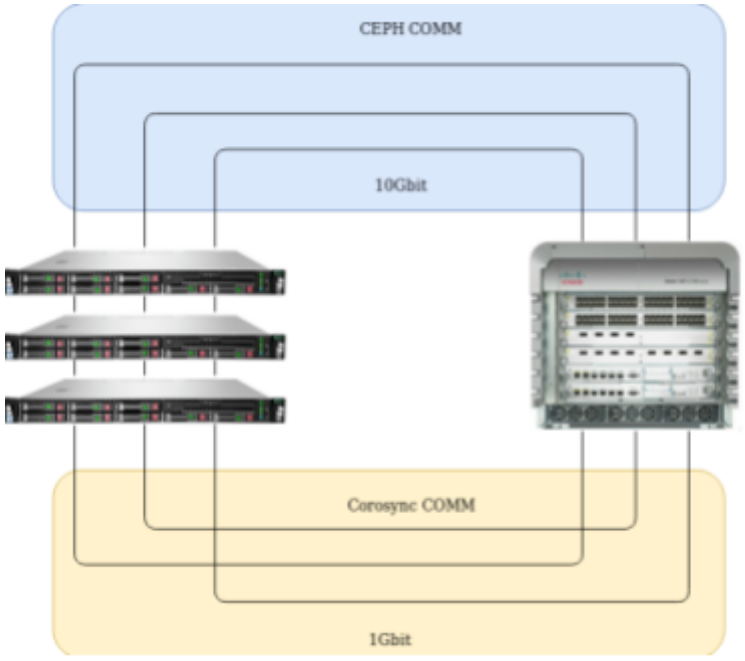

Figura 3- Corosync y CEPH clúster

3.3. Arquitectura

Para comenzar se describirán cada uno de los distintos tipos de almacenamientos que soporta la plataforma de virtualización Proxmox VE. Los mismos serán presentados en la siguiente tabla donde, por medio de características, se podrán comparar uno a uno; cabe destacar que la plataforma soporta almacenamiento de archivos (File Storage) y almacenamiento de bloques (Block Device) [31]:

Storage	Tipo	Compartido	Snapshots

Directorio	Archivo	No	No
NFS	Archivo	Sí	No
CIFS	Archivo	Sí	No
GlusterFS	Archivo	Sí	No
ZFS	Archivo	No	Sí
CephFS (CEPH)	Archivo	Sí	Sí
LVM	Bloque	No	No
LVM-Thin	Bloque	No	Sí
iSCSI	Bloque	Sí	No
ZFS over iSCSI	Bloque	Sí	Sí
RDB (CEPH)	Bloque	Sí	Sí

Luego de analizar la tabla, se puede notar que CEPH es el único sistema de almacenamiento distribuido que provee los dos tipos de almacenamiento, se está hablando de almacenamiento de archivo y de bloque; se puede ver que también permite, en ambos casos, realizar snapshots (sin utilizar un medio de almacenamiento con formato archivo, por ejemplo QCOW2). Éste permite tener todo lo necesario para obtener el rendimiento y las características adecuadas, pudiendo aprovechar toda la potencia de cómputo.

Si bien la elección de CEPH resolvió la cuestión del almacenamiento compartido, la plataforma debe instalarse sobre un sistema de archivos local. Para ello se optó por el File System ZFS, donde dos discos de los cuatro fueron utilizados para crear un RAID 1, y ahí mismo se instaló la plataforma Proxmox VE. Siguiendo las recomendaciones, se configuró la placa RAID para que funcione en formato HBA, ésto es lo recomendable tanto para ZFS como para CEPH (aunque CEPH puede ser configurado con un volumen RAID), ya que en ambos casos es conveniente que sean los mismos sistemas los responsables de comunicarse directamente con el hardware sin ninguna capa extra que no puedan visualizar ni manejar.

Por otra parte ZFS, en el sistema ROOT de la plataforma Proxmox VE, otorga la flexibilidad de realizar un snapshot previo a una tarea de mantenimiento, como puede ser una actualización del sistema Debian GNU/Linux y sus dependencias de PVE. Por lo cual se obtiene una característica que supera a la de tener un RAID por hardware, pudiendo volver a un snapshot anterior si algo sale mal en el proceso de actualización.

```
root@pve1:~# zpool status
  pool: rpool
 state: ONLINE
  scan: scrub repaired 0B in 0h0m with 0 errors on Sun Sep  8 00:24:41 2019
config:

        NAME        STATE     READ WRITE CKSUM
        rpool       ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            sda3    ONLINE       0     0     0
            sdb3    ONLINE       0     0     0

errors: No known data errors
```

Figura 4- Imagen del pool raíz en el nodo pve1 de ZFS en RAID1

Volviendo al tema de la elección de CEPH, por sobre otro sistema de almacenamiento compartido, cabe destacar que para el proyecto se asignaron tres equipos, éstos con la misma capacidad, cómputo y memoria, por lo cual se determinó que otorgar el rol de equipo de almacenamiento a uno de ellos produce un derrochamiento en áreas como el cómputo y la memoria. CEPH justamente resuelve esta situación, ya que los tres equipos realizan las mismas operaciones y pueden así repartir cargas. Por otra parte, la plataforma Proxmox VE, permite fácilmente, a futuro, la adopción de nuevos nodos para el cluster, que pueden o no ser parte del cluster CEPH.

Con respecto a la arquitectura de red, se comenzará con las interfaces de management. Estas serán utilizadas para realizar tareas de administración tanto sobre la plataforma Proxmox VE como para la administración remota del hardware.

Ambas fueron conectadas a un Switch de marca Cisco Catalyst por capa dos plana (Flat) a 100 Mbps de velocidad, y la misma por medio de capa 3, quedó lógicamente ubicada detrás de los Firewalls, de esta forma se logró dar seguridad a esta red, ya que la misma debe tener conectividad con la RUPCD pero con accesos limitados dentro la misma. El esquema de red para el management e iLO es el siguiente:

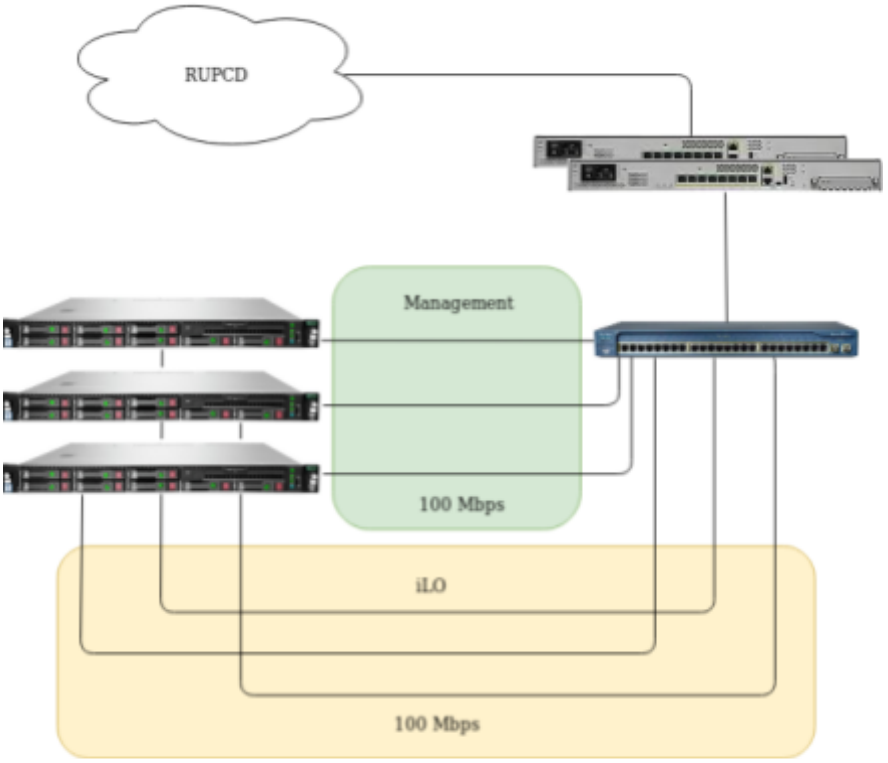

Figura 5- Conexión de Red de Management e iLO

Siguiendo con el esquema de red, se continuó con el armado del cluster, es decir, con la comunicación del cluster a través de Corosync Cluster Engine. Para ésto se utilizó una interfaz de cobre de 1 Gbps, la red no tiene acceso a la RUPCD, la misma está aislada para que no ingrese ruido a esta red sensible y de esta forma, el cluster funcione correctamente. A su vez, Proxmox VE recomienda el uso de una interfaz física dedicada para esta comunicación, separando así todo lo relacionado a management y almacenamiento, evitando saturaciones sobre la interfaz de comunicación del cluster.

Por su parte CEPH, también requiere de una interfaz física aislada al igual que el protocolo de cluster; solo que ésta se recomienda por un tema de performance que sea de 10 Gbps [32], para conseguir ésto, se utilizó uno de los dos puertos que el servidor trae por medio del adaptador de la marca HPE, modelo Ethernet 10G 2-port 546SFP+ Adapter. Esta interfaz es utilizada también por Proxmox VE para la comunicación entre la infraestructura con el cluster CEPH, por lo cual, si bien éste recomienda aislarlo también recomienda que sean dos interfaces de 10 Gbps en modo “bonding”, ésto no se logró conseguir por el límite en el hardware disponible; por lo cual se utilizó solo una interfaz, ya que la otra fue utilizada para la conectividad de las Máquinas virtuales y Contenedores con la RUPCD.

Por último, se utilizó la única interfaz restante para producción, la misma es una interfaz de 10 Gbps, la cual fue configurada como trunk para poder así dar conectividad a los equipos en distintos segmentos de red, donde algunos servicios están conectados lógicamente detrás del Firewall, en redes seguras, otros en DMZ internas y externas y otros en la MAN de la RUPCD. Se puede apreciar en la siguiente imagen, la conexión física de todas las interfaces descritas de la plataforma Proxmox VE:

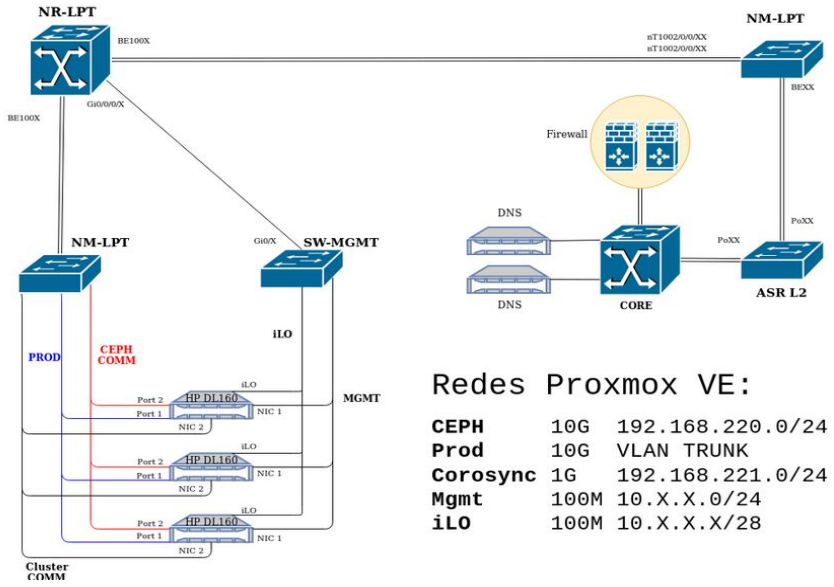

Figura 6- Diagrama físico general de la red de conexión

4. Instalación de la plataforma

Como se mencionó anteriormente, para la instalación de la plataforma Proxmox VE, la empresa provee de un instalador tipo ISO, donde la misma es desarrollada como una instalación gráfica, donde se seleccionaron opciones básicas para su instalación. A continuación cada una de las etapas de instalación de la plataforma Proxmox VE.

4.1. Configuración de placa RAID

Antes de comenzar dicha instalación procedimos a cambiar, tal como recomiendan tanto CEPH como ZFS, el modo en que la placa RAID funciona. Ésta, por defecto, viene con la utilidad para generar RAID por hardware, pero tal como dice la documentación de ZFS, éste necesita que los discos estén íntegramente controlados sin tener ninguna capa media de software entre la herramienta y el hardware. Es por ésto que se tomó la decisión de ingresar en la configuración de la placa RAID y se seteó el modo HBA (Host Bus Adapter):

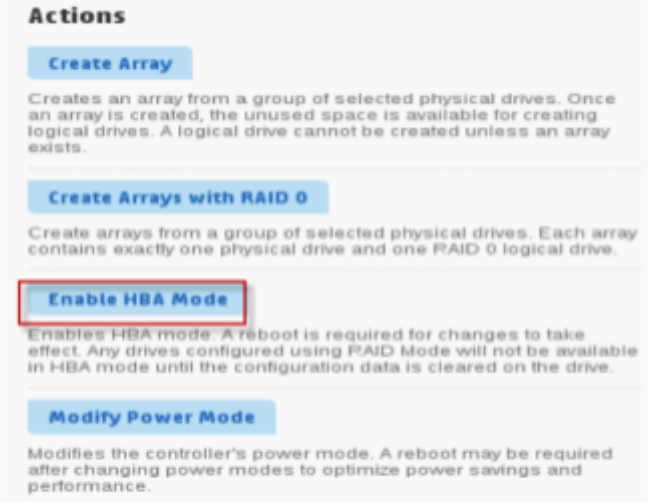

Figura 7- Seteamos el modo HBA en la placa RAID

4.2. Instalación de PVE

La siguiente etapa, fue la instalación en sí del sistema operativo, para ésto se seleccionó el arranque por USB/CDROM. La primer pantalla se observa el inicio de la instalación donde la misma es gráfica. Se seleccionó la primera opción y se comenzó con la instalación:

Figura 8- Opciones de instalación de la plataforma Proxmox VE

4.3. Licencia de la plataforma

En la siguiente etapa, se puede apreciar la licencia para el usuario final, es decir, detalla la licencia GNU Affero General Public License (V3), antes mencionada en el presente documento. Por otro lado, en uno de los puntos mostrados, se puede ver

un ítem, donde la empresa no se hace cargo bajo ningún punto de dar garantía del perfecto funcionamiento de la herramienta:

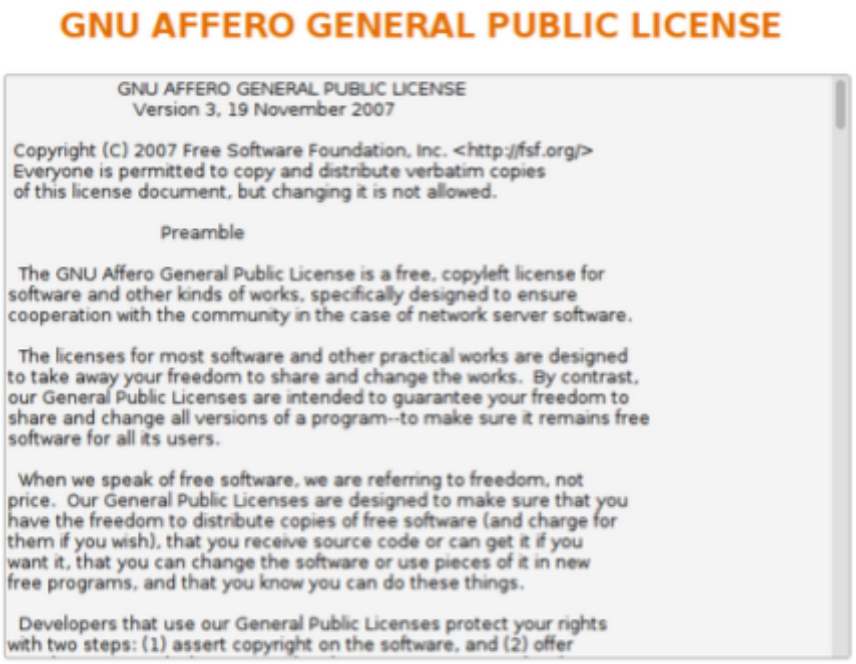

Figura 9- Licencia de usuario Final

4.4. ZFS

En la siguiente etapa de instalación, se generó el RAID ZFS, el mismo es un RAID tipo 1 (espejo). Para ésto, en las opciones botón “Options” se puede elegir que raid tomar:

Figura 10- ZFS con opciones por defecto

En el siguiente apartado se explicarán cada una de las variables, que si bien se dejaron por defecto en la generación del pool ZFS, éstas son las mismas que se hubiesen elegido para esta instalación [33]:

a- ASHIFT: ésta controla el tamaño del sector del pool creado, ashift con valor 12, hace referencia a 4096 bytes, ésto es por 2**12 = 4096.

b- Compress: Este flag en “ON”, produce que todo dato almacenado en el pool ZFS, se almacene comprimido, generando así una mejor utilización del espacio total del pool.

c- Checksum [34]: Si bien puede setearse en off, el mismo no es recomendado, ya que quita a ZFS la potencia de la recuperación de datos corrompidos. Las opciones son las siguientes:

Checksum	OK para deduplicación	Compatible con otras implementaciones de ZFS	Notas
on	ver notas	si	“on” es fletcher4 para datasets no de-duplicados y sha256 para datasets de-duplicados
off	no	si	No usar esta opción
fletcher2	no	si	Implementación obsoleta del checksum fletcher, usar fletcher4
fletcher4	no	si	algoritmo de fletcher

sha256	si	si	opción por defecto
noparity	no	si	no usar esta opción
sha256	si	si	no es soportado aun por todos los Filesystems
skein	si	si	no es soportado aun por todos los Filesystems
edonr	si	si	no es soportado aun por todos los Filesystems

d- copies: esta opción permite generar copias de los datos de usuario, es decir, si se setea por ejemplo el número de copias a dos, éste duplicará cada bloque guardado, duplicando la información de usuario.

4.5. Detalles finales de la instalación

4.5.1. Zona horaria

En la siguiente etapa, se seleccionó el país junto con la zona horaria y la distribución del teclado:

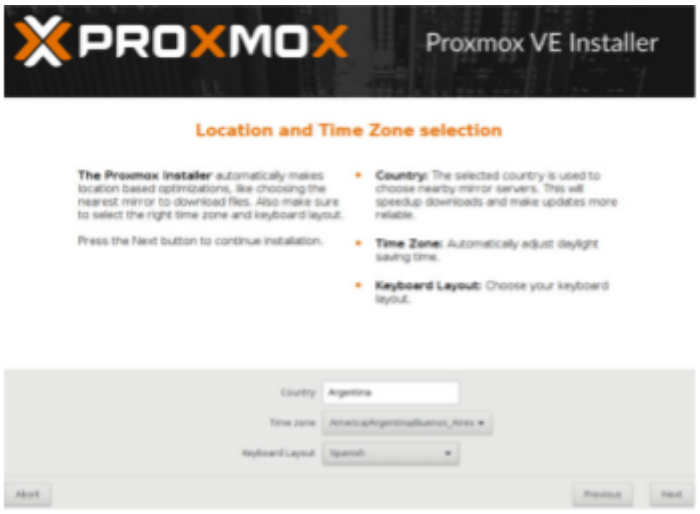

Figura 11- Localización y distribución del teclado

4.5.2. Definición de password de administrador

Siguiendo con la instalación, se configuró la password administración, tanto para user root del Sistema Operativo Debian GNU/Linux como el user para la web GUI, que también es el mismo usuario:

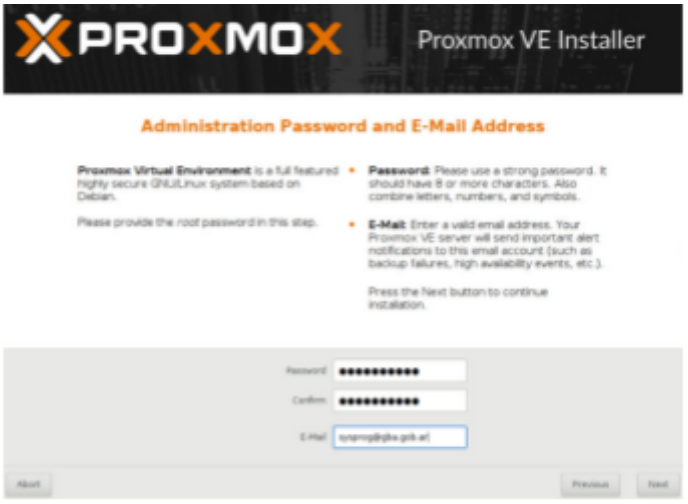

Figura 12- Configuración de las password de administrador de la plataforma

4.5.3. Direccionamiento de Management

En esta etapa, se seleccionó cual será el direccionamiento IP para la interfaz de management, dicha interfaz ha sido seleccionada en la definición de la arquitectura de red:

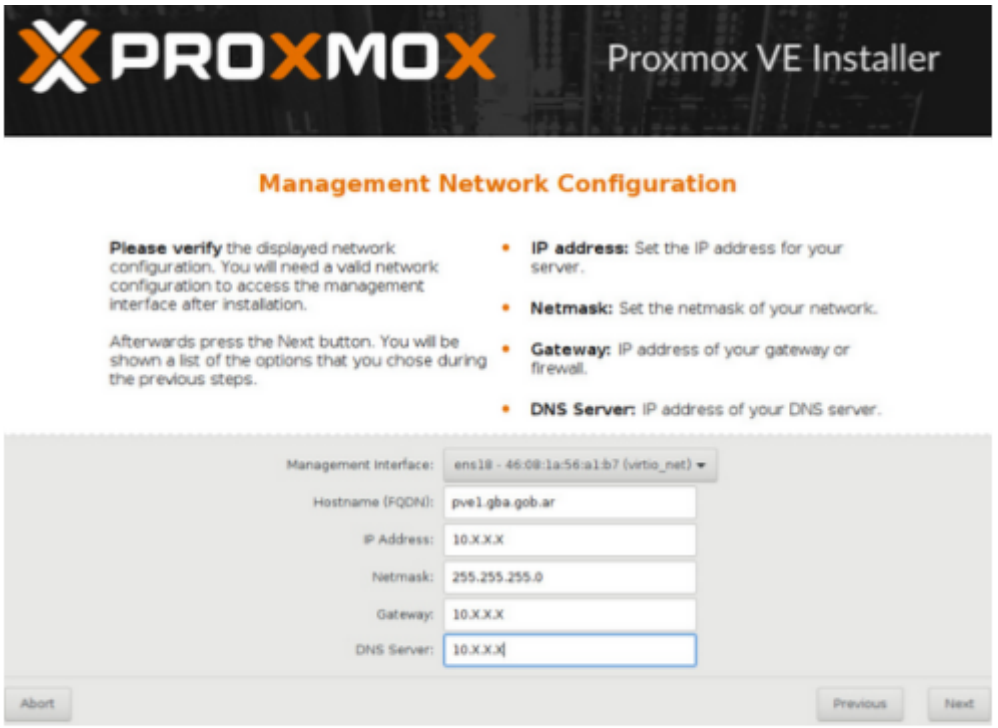

Figura 13- Configuración de red de administración de Proxmox VE

Terminada esta etapa comenzó la instalación del mismo, que fue muy veloz debido al recorte de muchas herramientas que suelen traer los sistemas operativos convencionales.

Figura 14- Instalación en proceso

Con ésto quedó instalada la plataforma en formato stand-alone, donde luego se procedió a la configuración del cluster de virtualización y el cluster CEPH, logrando convertir a Proxmox VE en una plataforma integral e hiperconvergente.

5. Configuración de la plataforma

5.1. Planteo general del esquema

El objetivo era lograr generar un cluster hiperconvergente, teniendo en cuenta las recomendaciones y exigencias propuestas por la misma plataforma para la construcción del sistema [35]:

1- Interfaces Aisladas: Es recomendable que la interfaz de management esté aislada del resto de las interfaces, lo mismo para la interfaz de CEPH, Corosync Cluster Engine y producción.

2- Sincronización de tiempo: Este punto es una exigencia dentro de la creación de clusters en general, ya que los mismos deberán estar sincronizados para funcionar correctamente; CEPH por ejemplo no funcionará si ésto no está correctamente definido.

3- CEPH: Para el caso de la comunicación de CEPH es necesario, para que no se produzcan cuellos de botella, conectar los mismos por interfaces con velocidades mayores a 1 Gbps, en este caso, se conectó por 10 Gbps en fibra como medio físico.

4- Red de migración de VMs: Para realizar la migración de máquinas virtuales, se puede definir una interfaz particular para hacerlo. Según la documentación, la plataforma utilizará la interfaz definida para la comunicación del

cluster y recomienda cambiarla. Por lo cual, se generó una VLAN dentro de la interfaz de producción para realizar esta tarea.

5.2. Configuración inicial

En esta etapa del proyecto, se disponían instalados en forma base cada uno de los sistemas operativos PVE en los tres servidores físicos. A su vez, los mismos se encontraban conectados todos siguiendo la arquitectura de red definida en puntos anteriores, de modo que la misma permite empezar la configuración de cada una de las partes del cluster.

A grandes rasgos, la configuración realizada consta de tres pasos, en el primer paso se describieron las configuraciones iniciales, como ser, la definición del servidor NTP y los nombres de host, con sus respectivas direcciones IPs. En el segundo paso, se definió la configuración de Cluster con Corosync y la herramienta que provee la plataforma para realizarlo. Y en el último paso se definió el cluster CEPH.

En el siguiente paso, se detallarán cada una de las configuraciones realizadas:

En primer lugar, se mostrarán los nombres asociados a las direcciones IPs en el archivo /etc/hosts de cada uno de los equipos. En este caso, se presentará como ejemplo el del nodo1, llamado pve1:

```
# cat /etc/hosts
127.0.0.1 localhost.localdomain localhost
10.1.222.11 pve1.gba.gob.ar pve1

# other pve host
10.1.222.12 pve2.gba.gob.ar pve2
10.1.222.13 pve3.gba.gob.ar pve3
```

corosync network hosts - Cluster COM
192.168.221.11 pve1-corosync.gba.gob.ar pve1-corosync
192.168.221.12 pve2-corosync.gba.gob.ar pve2-corosync
192.168.221.13 pve3-corosync.gba.gob.ar pve3-corosync

Para el caso de la sincronización del tiempo, la plataforma Proxmox VE, utiliza un cliente NTP llamado systemd-timesyncd [36] y su configuración reside en el archivo /etc/systemd/timesyncd.conf, donde se cambiaron los servidores remotos NTP, que este cliente trae configurado por defecto, por uno local a la RUPCD.

5.2.1. Configuración del Cluster Corosync

En la segunda parte de la configuración se realizó lo necesario para crear el cluster, ya que los tres nodos no estaban aún interactuando entre sí, y para ello se utilizó la utilidad de consola que provee la plataforma llamada “pvecm” (Proxmox Virtual Environment Cluster Manager).

A través de la consola de cada uno de los nodos, sobre la definición de las interfaces de GNU/Linux Debian, se configuró y luego se dió de alta la interfaz necesaria para la comunicación del protocolo generado por Corosync Cluster Engine:

Nodo1 (pve1):

Archivo /etc/network/interfaces:
auto eno2
iface eno2 inet static
    address 192.168.221.11

```
    netmask  255.255.255.0
```

Nodo2 (pve2):

Archivo /etc/network/interfaces:

```
auto eno2
iface eno2 inet static
    address  192.168.221.12
    netmask  255.255.255.0
#Cluster COM
```

Nodo3 (pve3):

Archivo /etc/network/interfaces:

```
auto eno2
iface eno2 inet static
    address  192.168.221.13
    netmask  255.255.255.0
#Cluster COM
```

Terminada la configuración de la interfaz para la comunicación, se levantó la interfaz para luego poder verificar si entre cada uno de los nodos existe conectividad entre sí. El resultado se puede observar en la siguiente salida del comando ping:

```
root@pve1:~# ping 192.168.221.13 -c4
PING 192.168.221.13 (192.168.221.13) 56(84) bytes of data.
64 bytes from 192.168.221.13: icmp_seq=1 ttl=64 time=0.171 ms
64 bytes from 192.168.221.13: icmp_seq=2 ttl=64 time=0.110 ms
64 bytes from 192.168.221.13: icmp_seq=3 ttl=64 time=0.149 ms
64 bytes from 192.168.221.13: icmp_seq=4 ttl=64 time=0.149 ms

--- 192.168.221.13 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3059ms
```

rtt min/avg/max/mdev = 0.110/0.144/0.171/0.026 ms

Asegurada la conectividad y la velocidad de la interfaz, se procedió a la creación del cluster. Ésto, como ya se mencionó anteriormente, se realiza con la utilidad “pvecm” de línea de comandos, aunque es bueno mencionar que actualmente Proxmox VE permite realizar ésto desde la Web GUI. De todas formas, hacerlo desde la línea de comandos también es muy sencillo. Se pasará a mostrar paso a paso cómo se realiza:

Para comenzar, desde el nodo 1, llamado pve1, se generó, por medio de la herramienta “pvecm,” la creación del cluster:

pvecm create cluster-dpt -bindnet0_addr 192.168.221.11 -ring0_addr pve1-corosync

El cluster fue llamado “cluster-dpt”, este nombre luego es visualizado desde la web GUI, y como puede verse se utilizó la red 192.168.221.0/24, donde cada uno de los nodos tiene una dirección IP definida sobre la interfaz física de 1 Gbps llamada “eno2”.

Ahora, solo restaba agregar los demás nodos al cluster definido en el nodo 1 (pve1). Para efectuar ésto se realizó la conexión a las consolas de los nodos pve2 y pve3. En ellos se ejecutaron los siguientes comandos:

nodo 2 (pve2):
pvecm add 192.168.221.11 -ring0_addr pve2-corosync

nodo 3 (pve3):
pvecm add 192.168.221.11 -ring0_addr pve3-corosync

Terminada esta configuración se apreciará en el menú de la web GUI como quedó armado el cluster y en qué estado está. Para ello hay que ingresar a la plataforma por un navegador WEB y desde la barra de herramientas seleccionar la opción Datacenter -> Cluster:

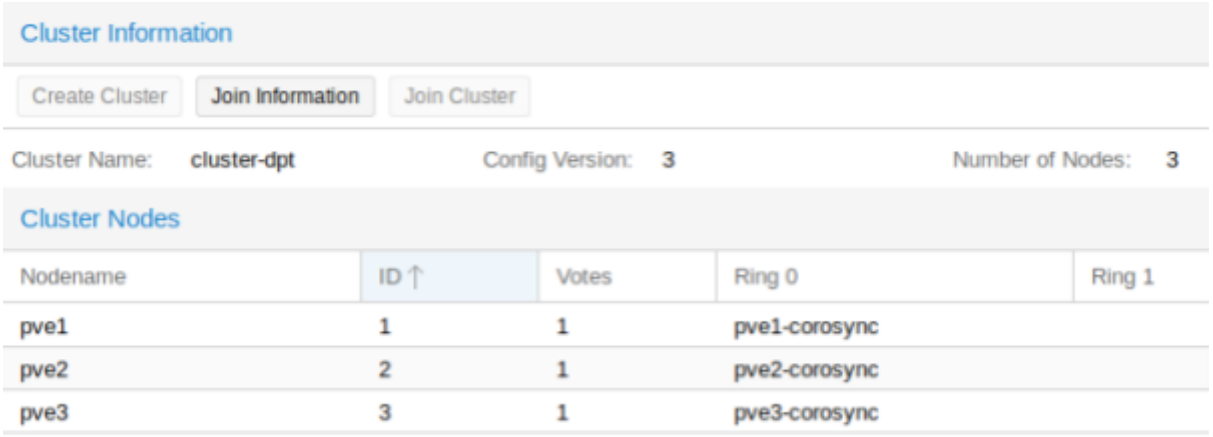

Figura 15- Visualización del estado del cluster PVE

Por otro lado se pueden visualizar en la consola web los tres servidores conectados al cluster donde luego se podrán administrar en forma centralizada:

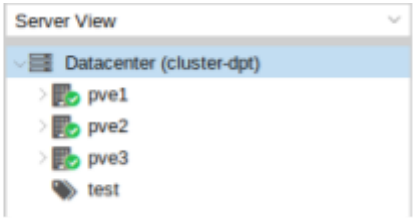

Figura 16- Vista de los servidores en forma centralizada

5.2.2. Configuración del Cluster CEPH

Para este punto, el cluster de Proxmox VE ya se encontraba definido, ahora solo falta la última etapa del proyecto donde se definió el cluster CEPH. Para ésto se comenzó definiendo la interfaz de comunicación así como también se llevó a cabo

para el punto anterior. Luego de configurarla, la misma fue activada y se realizó una prueba de conectividad y performance.

Antes de comenzar a mostrar cómo se realizó la instalación de CEPH, es necesario aclarar algunos puntos importantes del mismo, ya que esta tecnología utiliza muchos conceptos nuevos y es necesario entenderlos para luego introducirlos en las configuraciones que se realizaron luego de la instalación de la herramienta [37]:

1- RADOS (Reliable Autonomic Distributed Object Store): Es la base de CEPH, la cual se encarga de manejar los datos a bajo nivel. A su vez, es el encargado de definir zonas, donde las réplicas están distribuidas y permite que ante la caída de discos, servidores o racks enteros, los datos puedan seguir disponibles y replicados.

2- Algoritmo CRUSH (Controlled Replication Under Scalable Hashing): Tal como ya se mencionó, CEPH es un almacenamiento distribuido. Para lograr ésto, los clientes son quienes saben calcular a donde tienen que buscar los datos en los discos del clúster. Para ello, utilizan el algoritmo y los mapas de CRUSH, y con ello los clientes son capaces de resolverlo de forma muy veloz y sin puntos de fallas.

3- OSD (Object Storage Daemon): Éstos son los encargados de generar y proveer al usuario el almacenamiento. A su vez, se encargan de manejar todo lo referente a operaciones con discos, como ser, lecturas, escrituras e integridad de datos.

4- Monitor (MONs): Ellos son los encargados de mantener y verificar el estado del cluster. También realizan y mantienen el mapa de OSD y otras características más para mantener el correcto funcionamiento de CEPH.

5- RADOS GateWay (RGW): Provee almacenamiento por objetos, esta instalación no cubre esta característica.

6- MetaData Servers (MDS): Servicios utilizados para proveer almacenamiento del tipo archivo, en él reside todo lo relacionado a los metadatos de un File System, como los permisos, jerarquía, nombres, dueños, entre otras. Solo contiene metadatos, no los datos en sí.

7- CephFS: Es el sistema de archivos de CEPH, éste se apoya en el MDS para lograr convertir a CEPH en un tipo de almacenamiento de archivos.

8- RADOS Block Device (RBD): Similar a una arquitectura SAN (Storage Area Networks) con protocolos como iSCSI o Fibre Channel, CEPH permite presentar al cliente un almacenamiento por bloques. Es una de las utilidades más usadas en las primeras instalaciones de CEPH.

9- Placement Groups: CEPH almacena los datos en conjuntos llamados Placement Groups. Cada placement group es guardado en un OSD y replicado en el cluster. La forma de calcular en que PG es guardado el dato, es a través del algoritmo CRUSH.

Presentados los componentes más importantes para la instalación de este cluster, se puede ahora continuar con la descripción de la instalación de CEPH. Para comenzar, desde la consola de los nodos y sobre el archivo de definición de red de Debian GNU/Linux, se configuró la interfaz para la comunicación de CEPH:

Nodo1 (pve1):
Archivo /etc/network/interfaces:

```
auto ens1d1
iface ens1d1 inet static
        address  192.168.220.11
        netmask  255.255.255.0
```

#CEPH

Nodo2 (pve2):
Archivo /etc/network/interfaces:

```
auto ens1d1
iface ens1d1 inet static
        address  192.168.220.12
        netmask  255.255.255.0
#CEPH
```

Nodo3 (pve3):
Archivo /etc/network/interfaces:

```
auto ens1d1
iface ens1d1 inet static
        address  192.168.220.13
        netmask  255.255.255.0
#CEPH
```

Para terminar (y al igual que la configuración de Corosync), se realizó la prueba de conectividad a través del comando ping y la misma respondió positivamente.

Ahora se procederá a explicar cómo fue la creación del cluster CEPH. Para realizar ésto, en primer lugar fue necesario instalar el software sobre los nodos. Proxmox VE, en versiones actuales permite la instalación del cluster por Web GUI, en el momento de la instalación esta utilidad no estaba desarrollada y por lo tanto se realizó por consola CLI. Se describen a continuación cada una de las etapas donde se puede apreciar que a través de la utilidad proveída por la plataforma Proxmox VE, resulta sencillo desarrollarlo:

Por medio de la consola, en los tres nodos, se ejecutó el siguiente comando de instalación a través de la utilidad “pveceph”. La misma, como se mencionó anteriormente, es proveída por Proxmox VE:

```
# En todos los nodos ejecutar:
$ pveceph install --version luminous
```

Como puede verse, se instaló la versión “luminous” de CEPH, donde la misma era hasta el momento recomendada por la documentación de Proxmox VE. Terminada la instalación, se generó desde el nodo 1 (pve1), la configuración inicial de CEPH a través del comando pveceph:

```
# En nodo 1 (pve1) ejecutamos:
pveceph init --network 192.168.220.0/24
```

De esta forma se generó la inicialización del cluster en la red definida anteriormente para esta función. Luego, se procedió a generar los nodos monitor (MONs). Para realizar ésto, se ejecutó el siguiente comando por medio de la consola de los tres nodos:

```
# Ejecutamos la siguiente instrucción en todos los nodos:
$ pveceph createmon
```

Terminada esta etapa, y ya desde la Web GUI, se puede ver desplegado CEPH en la plataforma Proxmox VE, donde en este caso se ven los tres servidores definidos como nodos monitor:

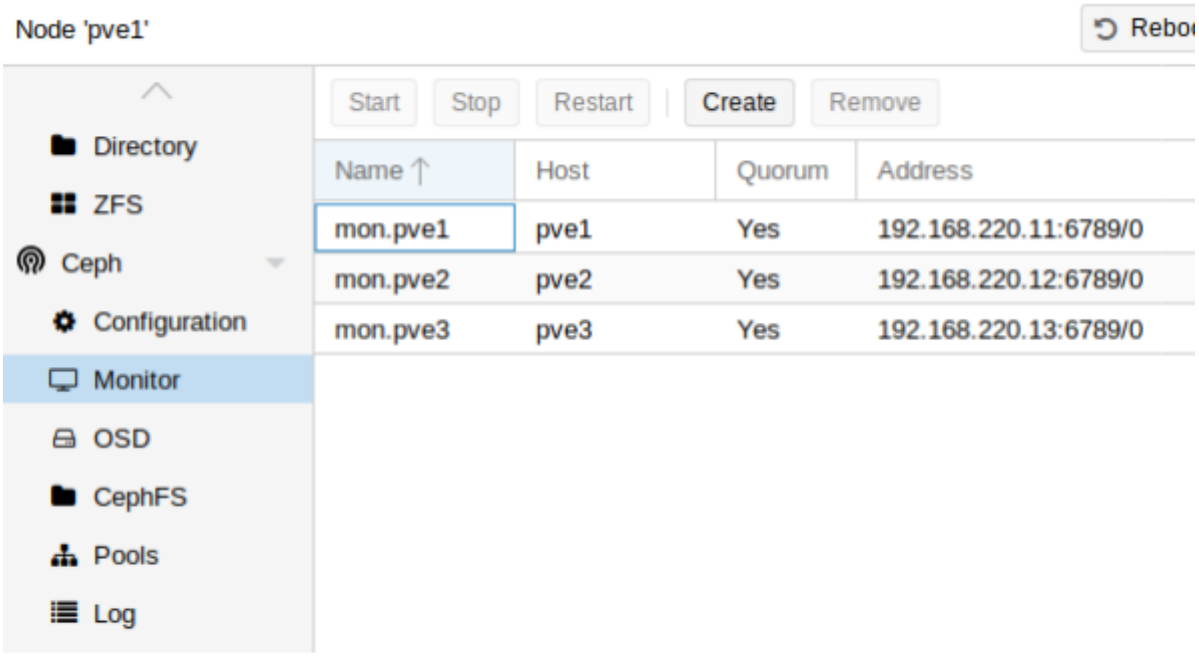

Figura 17- Visualización de los monitores de CEPH

En este paso solo restaba agregar los OSD, para esta tarea se utilizó la WEB GUI, donde en la ya mencionada interfaz se pudieron agregar los OSD de forma sencilla:

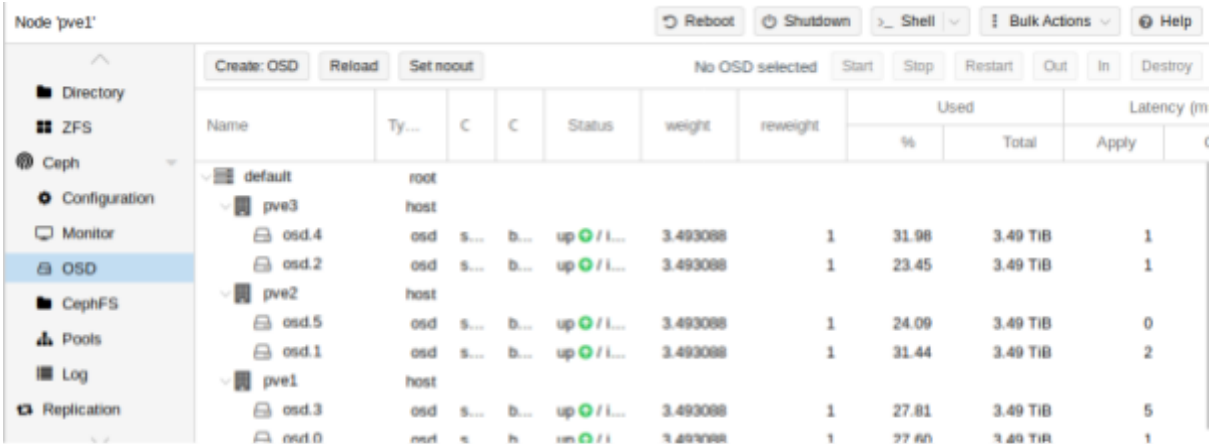

Figura 18- OSD creados en el cluster CEPH

Para continuar con la configuración de la infraestructura hiperconvergente, aún faltaban generar los almacenamientos para que la plataforma Proxmox VE pueda hacer uso de ellos. Dicha plataforma, como se ha mencionado anteriormente, soporta dos tipos de almacenamiento, por bloque y por archivos. En la primera

parte, se mostrará a modo ejemplo, cómo generar el almacenamiento por bloque de CEPH, donde el mismo se realiza desde la interfaz web:

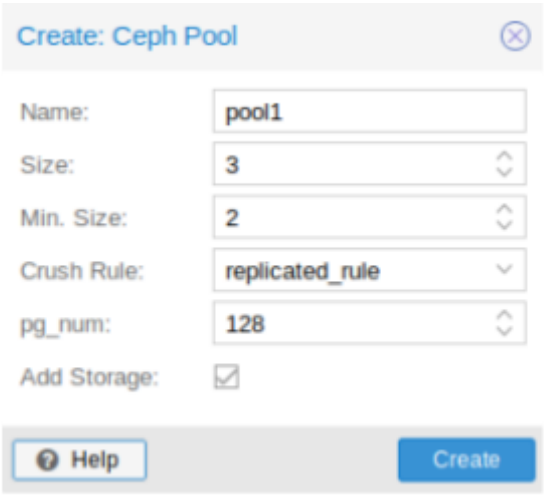

Figura 19- Creamos el pool de bloque

Antes de continuar con el siguiente almacenamiento, se procederá a explicar cada una de las opciones ofrecidas en la generación del pool [38]:

a- Size: Número mínimo de réplicas, es decir, cuántas veces será replicado el dato dentro del cluster CEPH.

b- Min. Size: Número mínimo de réplicas para considerar el cluster CEPH como degradado.

c- CRUSH Rule: Proxmox provee de una regla CRUSH llamada replicated_rule, ésta viene por defecto, y básicamente configurada para que realice la replicación de los datos.

d- pg_num: Número que define cuantos Placement Groups va a tener el Pool. CEPH no puede calcular esto automaticamente, de modo que deja una lista simple en su documentación para saber qué número es recomendable escoger de acuerdo a la instalación.

Con ésto se dio por terminada la configuración del pool para almacenamiento del tipo bloque de la plataforma, en la misma se puede visualizar, desde la web GUI, que ya quedó configurada y lista para usarse en la definición de Máquinas Virtuales y Contenedores.

Para finalizar con la configuración de CEPH, se ha creado también un almacenamiento del tipo archivo, éste permite almacenar archivos del tipo ISOs, backups, templates, máquinas virtuales y contenedores. Para realizar ésto, primero se deben definir los MDS (MetaData Servers), ésto se realizó desde la WEB GUI y su configuración es sencilla:

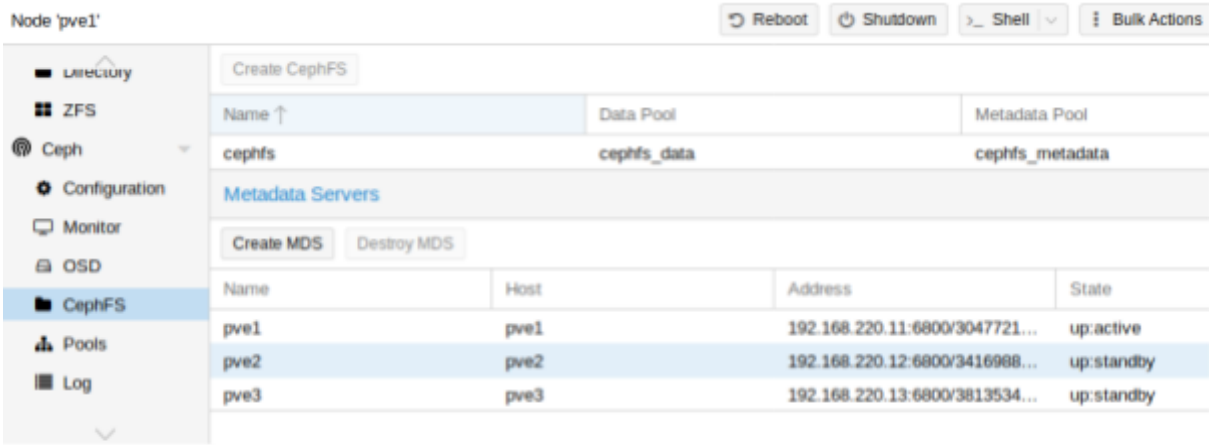

Figura 20- Servidores de MetaDatos y CephFS

Terminada esta acción, se creó también el pool CephFS [39], éste también es muy sencillo crearlo desde la web GUI. Con ésto quedó concluida la etapa de configuración de CEPH y la plataforma Proxmox VE se convirtió en una infraestructura hiperconvergente, soportando virtualización, networking y storage con servidores convencionales.

Antes de comenzar con las pruebas de funcionamiento y la muestra de los servicios migrados, se pasarán a comentar algunos de los detalles, sobre todo en la migración de Máquinas Virtuales y Contenedores. En una de las pruebas se pudo

apreciar que la migración se realizaba muy lenta, con tiempos muy altos considerando que los discos residen sobre el almacenamiento compartido. Por lo tanto al verificar ésto, se notó que por defecto la plataforma estaba utilizando la interfaz de management para realizar esta acción. La misma estaba conectada a una velocidad de 100 Mbps, ya que para la administración de la plataforma no es necesario utilizar un puerto con más velocidad. Es por ello que se realizó la siguiente acción:

Se generó una nueva VLAN sobre la interfaz de producción, ya que la misma es de 10 Gbps y tanto las acciones de migración como replicación de la plataforma se utilizan en casos puntuales. Por otro lado, no es conveniente utilizar ni la interfaz de Corosync Cluster Engine, ni tampoco la interfaz de comunicación de CEPH. Se mostrará a continuación la configuración del nodo 1, donde fue definida la VLAN 4000 con la dirección IP en la red 192.1688.222.0/24; lo mismo fue realizado en los demás nodos:

```
auto ens1.4000
iface ens1.4000 inet manual
        address  192.168.222.11
        netmask  255.255.255.0
        vlan-raw-device ens1
#REP - No Usar en PROD
```

Cuando la conectividad estaba lograda, se configuró sobre las opciones de la plataforma que la red de migración/replicación, sea la red definida en el punto anterior (192.168.222.0/24). De esta forma se pudo notar que las migraciones bajaron de varios minutos a segundos.

Por otro lado, siguiendo la recomendación de la plataforma para nuestra arquitectura [40], se puede aumentar la velocidad de transferencia configurando sobre la red de migración, la opción de hacerlo en forma insegura, es decir, no

utilizando un tunnel SSH sino que realizando una conexión sin encriptación. Ésto logró reducir a la mitad los tiempos de migración y replicación.

Figura 21- Utilizando insecure en migración

Sin esta opción la misma migración se demoraba unos 25 segundos.

6. Funcionamiento

Actualmente la plataforma de virtualización Proxmox VE, está funcionado con varios servicios que presta la dirección, los cuales ya se encuentran migrados. Algunos de ellos estaban prestando servicio en servidores físicos ya obsoletos, por lo cual éstos fueron virtualizados. En cambio, otros ya estaban funcionando en un entorno virtual sobre la plataforma VMWare con su formato de disco tipo archivo VMDK (Virtual Machine DisK).

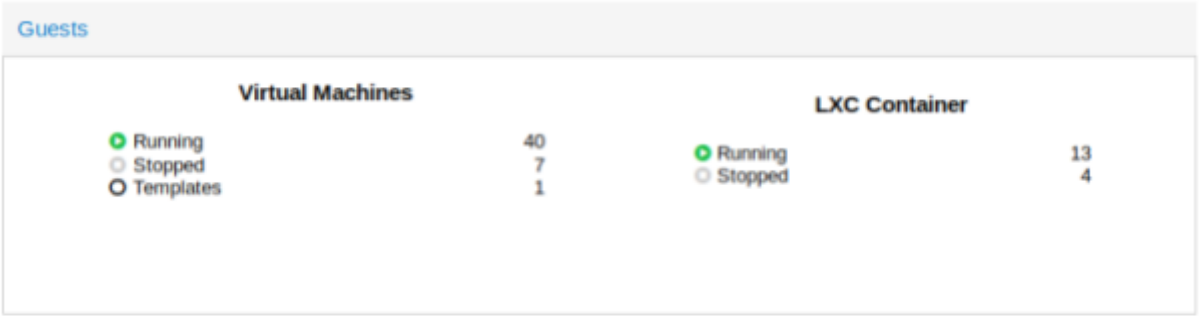

Figura 22- Máquinas virtuales y contenedores LXC sobre la plataforma

En la imagen se pueden notar todas las máquinas que están ejecutándose sobre la plataforma, las mismas son DNS centrales (internos y externos), pasarelas de correos para los dominios, servidores de monitoreo, Servidores LDAP, servicios IPAM (IP Administration), Git, entre otros.

Los recursos están siendo administrados desde el sector, intentando mantener la distribución de las cargas, de modo de ir utilizando los recursos equitativamente en el cluster:

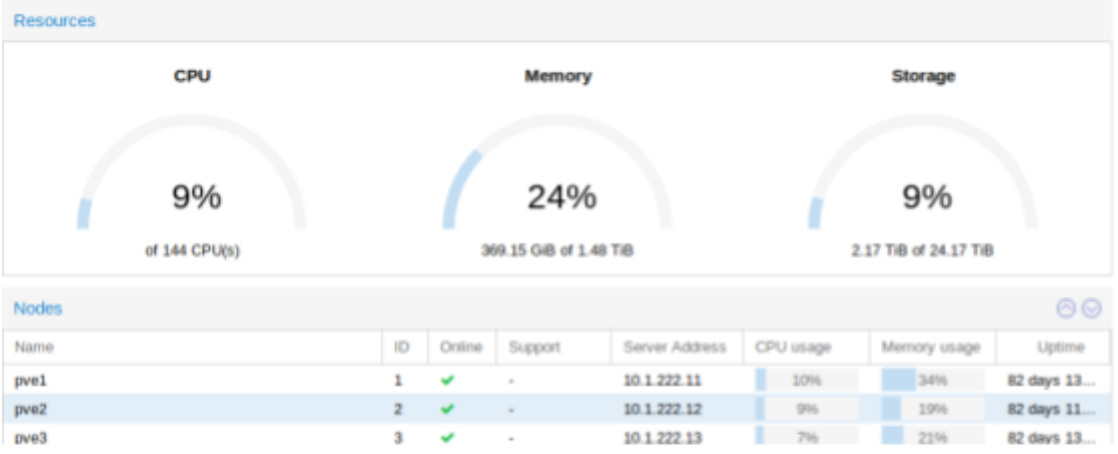

Figura 23- Dashboard de visualización centralizada de utilización de recursos

6.1. Migraciones

Las migraciones de servicios existentes fueron desarrolladas posteriormente a las pruebas de funcionamiento y performance sobre el cluster y el sistema de almacenamiento. Éstas fueron organizadas teniendo en cuenta su nivel de importancia y el tipo de migración que se iba a realizar. Las virtuales por ejemplo, se pudieron realizar de forma más sencilla, ya que Proxmox VE soporta el tipo de archivo VMDK [41], que hoy día es un formato libre.

Con respecto a los servicios que se estaban ejecutando sobre servidores físicos, éstos se encontraban desactualizados, en versiones muy anticuadas y fuera de soporte, tanto en hardware como en software. Por lo cual, dicha migración fue llevada a cabo tal como se realiza una convencional, donde se instaló un servidor virtual con un Sistema Operativo moderno y luego se migró la configuración de cada uno de los servicios. En su gran mayoría las configuraciones no sufrieron muchos cambios por lo cual fue sencillo realizar la migración teniendo en cuenta las recomendaciones de cada uno.

Volviendo al tema de los servidores virtuales, los discos de extensión VMDK, fueron enviados con el comando rsync y calculados los MD5 para corroborar que el pasaje se haya realizado correctamente. Una vez realizada esta acción, se generó una máquina virtual de similar característica sobre la plataforma de virtualización Proxmox VE y se asignó como disco al archivo VMDK, antes mencionado. Algunas distribuciones generaron problemas de arranque, como por ejemplo una central telefónica IP basada en ISSABEL, la misma está basada en el sistema operativo GNU/Linux CentOS 7.

Los problemas fueron asociados a los identificadores de los discos virtuales que genera una nueva máquina virtual; éstos cambiaron su ID y por lo tanto fue necesario ingresar a la configuración de arranque y cambiarlos a los valores nuevos.

Con ésto resuelto se pudieron migrar, eventualmente, todos los sistemas virtuales fácilmente.

Actualmente hay casi un 80% de los servicios migrados a la nueva plataforma, éstos están funcionando en producción y los resultados de la performance son satisfactorios considerando que muchos estaban con mucha demanda y el hardware no era conveniente ni suficiente para cubrirlo.

6.2. Nuevos proyectos

Por otro lado, la instalación de la plataforma Proxmox VE, permitió elaborar una serie de nuevos proyectos aprovechando los recursos disponibles. Se pasarán a detallar algunos de ellos:

6.2.1. Mail Gateway

Se implementó, y aún está en etapa de pruebas, otra herramienta de la misma empresa que distribuye Proxmox VE, la cual provee una plataforma para realizar control de correos, Antispam/Virus, definición de reglas, entre otras características. Dicha plataforma se llama Proxmox Mail Gateway y se distribuye bajo licencia GPL [42].

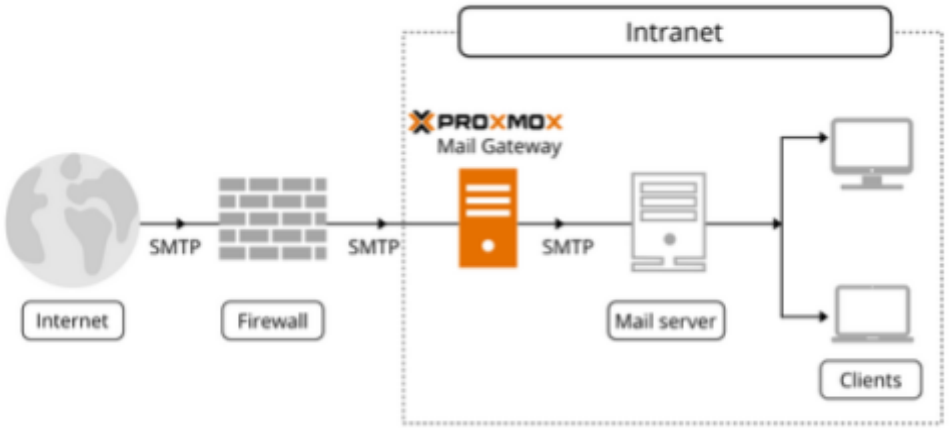

Figura 24- Proxmox Mail Gateway

Este proyecto será utilizado como contención a los correos entrantes de muchos organismos que necesitan de este servicio para protegerse de los ataques externos por este medio; delegando en la dirección la responsabilidad de contenerlos y responder en caso que algo sucediera.

6.2.2. Cluster Kubernetes

Así mismo se está desarrollando un cluster de Kubernetes [43] utilizando a Rancher [44] para administrarlo, éste permitirá experimentar la potencia de ambas herramientas y migrar algunos servicios hacia esta plataforma cuando se tome la decisión de ponerlo en producción. Para realizar ésto, se instalaron tres máquinas virtuales con Ubuntu Server 16.04 y la versión de docker soportada. En uno de ellos se instaló el controlador (Control plane y Etcd) y en los dos host restantes, los nodos de cómputos (Workers):

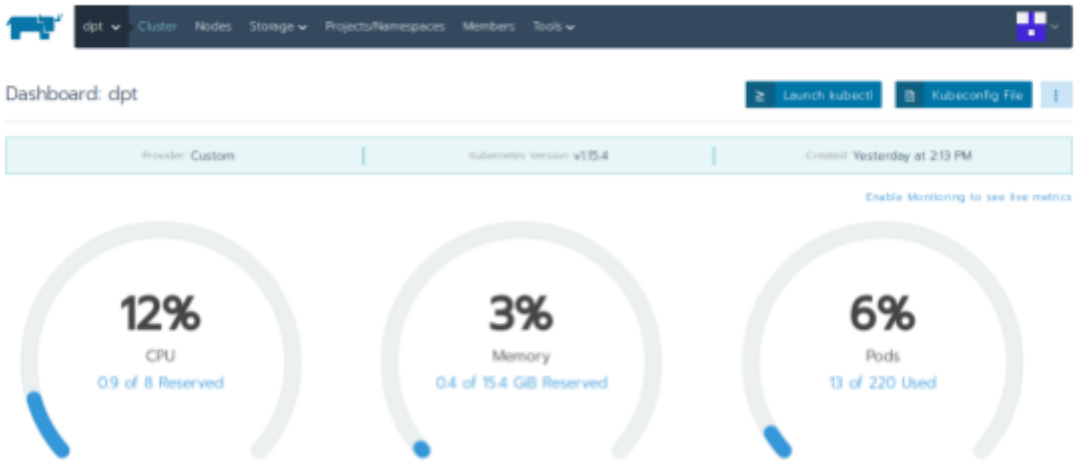

Figura 25- Rancher GUI: Kubernetes Cluster On-Premise

La idea final de este proyecto será otorgar a los desarrolladores de la dirección, la posibilidad de tener un entorno Docker en producción con Kubernetes utilizando las nuevas costumbres denominadas DEVOPS.

6.3. Nuevas funcionalidades

Cuando se hace referencia a las nuevas funcionalidades, se habla específicamente de las características que la virtualización otorga a los administradores de sistemas e infraestructuras. Antes de tener esta plataforma, las actualizaciones de los sistemas operativos y herramientas deben ser estudiados muy minuciosamente, teniendo en cuenta también los recaudos de poder recuperarnos ante una posible falla de la misma. Actualmente, con el uso de snapshots y backups de Proxmox VE, permite hacer ésto en mucho menos tiempo, y en algunos casos sin la necesidad de hacer un estudio de la actualización.

Otra característica interesante es que se puede, en forma manual, realizar un balanceo de carga en la utilización de recursos de los nodos del cluster. En el pasado, había equipos subutilizados y otros sobre-utilizados, y corregir ésto no era tan trivial como ahora, donde el administrador puede migrar máquinas virtuales en vivo sin tener caídas de los sistemas.

7. Conclusiones y trabajos futuros

7.1. Conclusiones

Este proyecto comenzó por medio de la implementación de la nueva red, la misma incluye, en uno de sus ítems, la compra de equipamiento de cómputo con el fin de dar soporte a servicios básicos y centrales que la dirección presta. Para ello se decidió implementar el cluster de HA con Proxmox VE, ya que permite concentrar, por medio de las tecnologías presentadas en el proyecto, la administración del cómputo, red y almacenamiento.

Lo fundamental que este proyecto aborda, más allá del cluster hiperconvergente, es la migración de los servicios. Tal como se mencionó anteriormente, el mismo se encuentra en una etapa avanzada, donde ya se lograron migrar servicios importantes que brinda la dirección y a su vez también se generó en ellos una mejor redundancia, ya que en algunos casos se disponía sólo de 2 equipos físicos sin virtualizar. A su vez, algunos presentaban problemas de performance, muchos de ellos ejecutándose en equipos con una cantidad insuficiente de memoria principal, generando en ellos el uso de “swap” que termina impactando negativamente en la performance general.

Otra de las ventajas asociadas fue la posibilidad de generar alta disponibilidad sobre algunos recursos. Un ejemplo de esto fue el servicio de VPN que la dirección ofrece. Este servicio es una máquina virtual (antes se ejecutaba sobre un servidor físico) basado en tecnología Open Source donde no había implementada ningún tipo de redundancia. Por medio de la plataforma Proxmox VE y CEPH, se logró generar un recurso en HA para esta máquina virtual, y así ante un eventual problema en algunos de los nodos, la misma pueda ser migrada sin intervención de los administradores.

Así mismo, se generó un ambiente de administración centralizado y homogéneo, donde el administrador queda acotado a la utilización de un solo sistema para realizar tareas rutinarias, como pueden ser snapshots, upgrades, backups, entre otras. Ésto generó una agilidad adicional, ya que anteriormente se presentaban limitaciones por la existencia de tecnología obsoleta, tanto de hardware como de software.

Para finalizar con los beneficios que otorga el cluster HA hiperconvergente, cabe también destacar la posibilidad de escalar este cluster existente con tres nodos. Proxmox VE, permite agregar nuevos nodos en forma sencilla y segura, teniendo en cuenta los requerimientos de la arquitectura existente, sin limitarnos en el uso de

distintas marcas vigentes en el mercado. Es necesario tener en cuenta algunos detalles a la hora de agregar nuevos equipos, como por ejemplo el tipo de disco. En este caso el pool CEPH fue creado con discos SSD SAS, por lo cual para agregar más OSD a este pool se debe tener el mismo tipo de disco para no limitar la performance.

Otro detalle a tener en cuenta es que la red de CEPH tiene una velocidad de 10 Gbps, por lo cual los nodos futuros deberán también cumplir con esta característica. Si bien existen más limitantes, como por ejemplo la cantidad de interfaces que requiere el nuevo equipamiento con respecto a la arquitectura existente, lo importantes es que se puede seguir aumentando el cómputo y almacenamiento en forma horizontal y heterogénea.

Uno de los puntos en contra, a la hora de implementar un cluster Open Source, es su posterior administración y mantenimiento. El equipo de IT encargado del mismo, deberá estudiar cada una de las tecnologías con las cuales el cluster fue instalado, de las cuales algunas llevan mucho tiempo en el mercado y con una excelente documentación, y otras en cambio son tecnologías modernas y su documentación y aportes a la comunidad son más escasos.

A su vez y para finalizar, se destacarán algunos eventos ocurridos en la etapa posterior a la etapa de instalación y migración de los servicios. En la mencionada etapa ocurrieron dos caídas generales del centro de datos, donde los equipos se reiniciaron y tomaron la configuración en forma excelente, no sufriendo ningún daño en los datos. Otro evento, posterior a estos cortes de energía, fue la rotura de uno de los discos SSD, el mismo era parte del cluster CEPH. La dirección tiene un contrato de misión crítica con la empresa HPE, y por ello se pudo realizar el cambio del OSD para el cluster CEPH, tomando los recaudos y buenas prácticas que la documentación de la tecnología provee.

7.2. Trabajos Futuros

1- Disaster Recovery Site: Si bien este proyecto se basó en la tecnología CEPH para la creación del cluster; en la etapa de evaluación de otro tipo de arquitecturas se trabajó con la posibilidad de realizar réplicas remotas de las máquinas virtuales utilizando ZFS como backend de storage. Esta tecnología permite realizar snapshots sobre los discos de las máquinas virtuales y enviarlos a sitios remotos para su importación. Lo interesante de esta herramienta es la posibilidad de hacerlo en forma incremental, logrando así que las copias sean más pequeñas.

A futuro, los equipos que fueron adquiridos por la dirección serán distribuidos en forma individual por distintos datacenters ubicados a distancias remotas unos de otros, no pudiendo lograr el mínimo de nodos necesarios para generar un cluster como en el que en este proyecto se desarrolló, es por ello que una solución con replicación en sitios remotos puede ser una buena posibilidad.

El siguiente script bash subido a la plataforma github [45] fue realizado como una pequeña prueba de la potencia de ZFS. El mismo puede aplicarse en forma de tarea de cron y se podrán enviar copias, en este caso, a un sitio remoto, pero puede adaptarse para hacerlo a más de uno si se quisiera. Para seguir adelante con esto, puede ser interesante cambiar la arquitectura y dar una solución más robusta y centralizada, donde desde una plataforma web se podrán realizar las copias de las Máquinas Virtuales seleccionadas en los sitios remotos que se desee.

2- Automatización de Operaciones: Otra línea de trabajo a seguir, puede ser la automatización de tareas de operaciones utilizando alguna de las existentes en el mercado. En el sector se estuvieron desarrollado varias automatizaciones con la plataforma Rundeck en conjunto con Ansible. Por medio de estas dos herramientas

se puede automatizar la generación de contenedores y máquinas virtuales otorgando a los usuarios la posibilidad de autogestión de los mismos.

Proxmox VE en sus versiones más actuales soporta la herramienta de bootstrapping Cloud-Init la cual permite, al momento de crear la Máquina virtual, que la misma se aprovisione de muchas características para tener un sistema base pero adaptado a la infraestructura existente. Ésto, en combinación con lo anterior, puede ser excelente para la administración de los servidores, máquinas virtuales y contenedores existentes en la infraestructura.

3- Observabilidad: En los últimos tiempos el kernel Linux está generando muchos cambios positivos sobre la inserción de herramientas para llevar a cabo la tarea de observación (trace) de varias capas del sistema operativo, como ser la CPU y sus caches, el uso de la memoria y swap, los File Systems, entre otros aspectos importantes para medir la performance. La inserción de eBPF (Enhanced Berkeley Packet Filter) al Kernel, ha logrado generar esta revolución y puede ser un interesante trabajo futuro poder medir variables para lograr una mejor performance en los sistemas.

8. Bibliografía básica Relacionada

[1] Ortiz - 2011 - P.4

[2] Aulafacil - 2014 - P.1

[3] Microsoft Azure

[4] Storage Area Essentials - Barker/Massiglia - 2001 - P.3

[5] TechTarget - Margaret Rouse

[6] RSG Comunicaciones

[7] Beowulf cluster computing with Linux - Gropp/Lusk/Sterling/Hall - 2003 - P.1

[8] CentOS High Availability - Resman - 2015 - P.2

[9] https://www.techopedia.com/definition/1115/storage

[10] https://www.redhat.com/en/topics/data-storage/software-defined-storage

[11] https://www.clonezilla.org/

[12] https://www.issabel.org/

[13] https://www.proxmox.com/en/news/press-releases/10-years-of-proxmox

[14] http://www.gnu.org/licenses/agpl-3.0.html

[15] Proxmox High Availability - Simon Cheng - pag. 7 - 2014

[16] https://pve.proxmox.com/wiki/Open_vSwitch

[17] https://www.proxmox.com/en/proxmox-ve/features

[18] https://lwn.net/Articles/705160/

[19] Mastering KVM Virtualization - pag 94 - Anil Vettathu, Prasad Mukhedkar, Humble Devassy Chirammal

[20] https://www.linux-kvm.org/page/KVM_Features

[21] http://people.redhat.com

[22] Practical LXC and LXD - pag 2- Senthil Kumaran

[23] Practical LXC and LXD - pag 4 - Senthil Kumaran

[24] FreeBSD Mastery: ZFS - Pag 3 - Michael Lucas - Allan Jude

[25] https://www.enterprisestorageforum.com

[26] Learning CEPH - pag 44 - Anthony D’atri, Karan Singh, Vaibhav Bhembre, Anthony D'Atri

[27] Mastering CEPH - pag 39 - Nick Fish
[28] https://en.wikipedia.org/wiki/Corosync_Cluster_Engine
[29] https://ark.intel.com
[30] https://www.hpe.com
[31] https://pve.proxmox.com/wiki/Storage
[32] https://www.proxmox.com/en/downloads/item/proxmox-ve-ceph-benchmark
[33] FreeBSD Mastery: ZFS - pag 76, 100 147 - Michael Lucas & Allan Jude
[34] https://github.com/zfsonlinux/zfs/wiki/Checksums
[35] https://pve.proxmox.com/wiki/Cluster_Manager
[36] https://pve.proxmox.com/wiki/Time_Synchronization
[37] Learning CEPH - Karan Singh, Vaibhav Bhembre, Anthony D'Atri
[38] https://pve.proxmox.com/pve-docs/chapter-pveceph.html#pve_ceph_pools
[39] Learning CEPH - pag 121 - Karan Singh, Vaibhav Bhembre, Anthony D'Atri
[40] https://pve.proxmox.com/wiki/Manual:_datacenter.cfg
[41] http://www.vmware.com/app/vmdk/?src=vmdk
[42] https://www.proxmox.com/en/proxmox-mail-gateway
[43] https://kubernetes.io/es/
[44] https://rancher.com/
[45] https://github.com/Moglius/zfs_replicator/blob/master/zfs_replicator.sh