

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA**

CARRERA DE INGENIERÍA DE SISTEMAS

*Trabajo de titulación previo
a la obtención del título
de Ingeniero de Sistemas*

PROYECTO TÉCNICO:

**“METODOLOGÍA PARA SISTEMAS DE
REPLICACIÓN MULTICLOUD HÍBRIDAS”**

AUTORES:

ÁNGEL GEOVANNY DUCHITANGA ABAD
ITALO FERNANDO MENDIETA SÁNCHEZ

TUTOR:

ING. ERWIN JAIRO SACOTO CABRERA, Ph.D.

CUENCA - ECUADOR
2022

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Ángel Geovanny Duchitanga Abad con documento de identificación N° 0104649652 e Italo Fernando Mendieta Sánchez con documento de identificación N° 0105998793, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos los autores del trabajo de titulación: **“METODOLOGÍA PARA SISTEMAS DE REPLICACIÓN MULTICLOUD HÍBRIDAS”**, mismo que ha sido desarrollado para optar por el título de: *Ingeniero de Sistemas*, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, febrero de 2022.

Ángel Geovanny Duchitanga Abad

C.I. 0104649652

Italo Fernando Mendieta Sánchez


C.I. 0105998793

CERTIFICACIÓN

Yo, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **“METODOLOGÍA PARA SISTEMAS DE REPLICACIÓN MULTICLOUD HÍBRIDAS”**, realizado por Ángel Geovanny Duchitanga Abad e Italo Fernando Mendieta Sánchez, obteniendo el *Proyecto Técnico* que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana.

Cuenca, febrero de 2022.

ERWIN JAIRO
SACOTO
CABRERA



Firmado digitalmente
por ERWIN JAIRO
SACOTO CABRERA
Fecha: 2022.01.27
12:25:21 -05'00'

Erwin Jairo Sacoto Cabrera, PhD
C.I. 0301185229

DECLARATORIA DE RESPONSABILIDAD

Nosotros, Ángel Geovanny Duchitanga Abad con documento de identificación N° 0104649652 e Italo Fernando Mendieta Sánchez con documento de identificación N° 0105998793, autores del trabajo de titulación: **“METODOLOGÍA PARA SISTEMAS DE REPLICACIÓN MULTICLOUD HÍBRIDAS”**, certificamos que el total contenido del *Proyecto Técnico*, es de nuestra exclusiva responsabilidad y autoría.

Cuenca, febrero de 2022.

Ángel Geovanny Duchitanga Abad

C.I. 0104649652

Italo Fernando Mendieta Sánchez

C.I. 0105998793

Dedicatorias

A mi madre Por todo su apoyo de manera constante en el aspecto moral, por su amor incondicional, por su trabajo, su esfuerzo y el sacrificio que ha realizado para permitirme llegar a este momento tan especial en mi vida, ese sacrificio que valore día tras día, gracias por inculcar en mí, la valentía y el coraje para poder superar cualquier dificultad y hacer posible que pueda culminar mi carrera profesional, gracias por ser mi ejemplo y hacerme sentir afortunado de tener una madre tan especial y ejemplar como ella, sin ella no hubiese sido posible llegar a cumplir este gran sueño.

A mi hermano por ser el motor principal de mi lucha y sacrificio día a día, el cariño que me ha brindado, por estar siempre conmigo y animarme a alcanzar esta meta anhelada. A mis amigos y a todas aquellas personas que me han permitido ser parte de su vida. Por todos los momentos buenos y malos que hemos pasado juntos, por confiar en mí, por ser parte de este proceso y por ser una fuente de ayuda e inspiración para alcanzar todos mis objetivos.

Angel Geovanny Duchitanga Abad

A mis abuelos Telmo Norman Sánchez y Rosa María Orellana Narváez por haber dedicado todos sus esfuerzos ya que son los principales promotores de mis sueños, por confiar y creer en mis expectativas.

A mis Padres Susy Yanira Sanchez Orellana y Italo Napoleon Mendieta Mendieta quienes a pesar de todas las dificultades y esfuerzos realizados, me ayudaron a hacer realidad mi sueño.

A mi querida amiga Pamela Mayte Villarroel Tigrero por tu gran apoyo y creer en mí y siempre guiar mi camino.

A mis amigos por toda la ayuda y el apoyo durante este período tan importante de mi educación académica.

Italo Fernando Mendieta Sánchez

Agradecimientos

En primer lugar, quiero agradecer a Dios por darme la sabiduría y la fortaleza para poder superar obstáculos a lo largo de mi vida y de mi carrera. A la Universidad Politécnica Salesiana por haberme abierto sus puertas para educarme, formarme y convertirme en una profesional siempre al servicio del prójimo. Al PhD Pablo Leonidas Gallegos Segovia por apoyarme en todo momento para realizar el proyecto de titulación, por compartir sus conocimientos, por ser un guía en el desarrollo del proyecto de titulación, por el tiempo, la dedicación y la predisposición para ayudar en la elaboración de la presente tesis a pesar de los duros momentos que se vive como consecuencia de la pandemia del COVID-19. Al Magister Jairo Sacoto por apoyarnos en el grupo de investigación GIHP4C, también expresar mi agradecimiento, enorme gratitud y cariño a mi madre Rosa Etelvina Abad Tacuri por darme la vida, por convertirse en el pilar fundamental de mi vida, por el apoyo incondicional que me ha brindado en cada instante, por los valores que me ha inculcado, los mismos que me han permitido ser una persona de bien, por el sacrificio que ha hecho para brindarme una excelente educación durante toda mi vida. A mi hermano Jonnathan Salazar por cada uno de sus momentos, felicidad, cariño, amor y apoyo incondicional. Por último, quiero agradecer de manera atenta a mis docentes por impartir sus valiosos conocimientos en las clases dictadas a lo largo de mi formación, a mis amigos, a mis compañeros y compañeras por cada momento de felicidad y experiencias que hemos vivido a lo largo de nuestra carrera universitaria, pero en especial a dos de ellos Italo Mendieta y Diego Rodriguez por ser unos excelentes amigos, agradezco el esfuerzo, la dedicación y la perseverancia puesta para poder culminar nuestro proyecto.

Angel Geovanny Duchitanga Abad

Yo Italo Fernando Mendieta Sanchez quiero agradecer a Dios, quien con su bendicion llena siempre mi vida de fortaleza para poder llegar siempre mas alla de lo esperado. De igual manera mis agradecimientos a la universidad politecnica salesiana, Al PhD. Pablo Leonidas Gallegos Segovia quien nos apoyo en los momentos mas complejos de nuestro proyecto de titulación y con la enseñanza de sus valiosos conocimientos hicieron que pueda crecer día a día como profesional. Al Magister Erwin Jairo Sacoto Cabrera quien estuvo pendiente gracias por su paciencia, dedicación, apoyo incondicional. A mis padres Susy Sanchez y Italo Mendieta quienes con su amor, paciencia y esfuerzo me han permitido llegar a cumplir hoy un sueño más, gracias por inculcar en mi el ejemplo de esfuerzo y valentía, de no temer las adversidades porque Dios esta conmigo siempre. A mis abuelos Telmo y Rosa por su cariño y apoyo incondicional ya que ellos son la base fundamental durante todo este proceso, por estar conmigo en todo momento desde mis inicios de mi educacion hasta este punto que veran convertirme en un profesional. Finalmente quiero agradecer a mis docentes de mi carrera por impartirnos sus conocimientos y experiencias. A mis compañeros Geovanny Duchitanga y Diego Rodríguez les agradezco por nunca rendirnos por siempre luchar ante cualquier situación y gracias por la amistad que nos hemos brindado.

Italo Fernando Mendieta Sánchez

Contenido

Glosario de términos	1
Resumen	2
Abstract	3
Introducción	4
Justificación	6
Objetivos	7
Objetivos Generales	7
Objetivos Especificos	7
1 Estado del Arte y Fundamentación Teórica.	8
1.1 Cloud Computing	8
1.1.1 Definición de Computación en la Nube	8
1.1.2 Nube Pública	9
1.1.3 Nube Privada	10
1.1.4 El modelo de nube híbrida y multicloud	11
1.1.5 Características	11
1.1.6 Arquitectura de la Computación en la Nube	12
1.2 Plataformas Open Source	13
1.2.1 OpenNebula	13
1.2.2 Eucalyptus	15
1.2.3 OpenStack	16
1.3 Plataformas de Cloud Propietarias	17
1.3.1 Google Cloud	17
1.3.2 Microsoft Azure	18
1.3.3 Oracle Cloud	18
1.3.4 Amazon Web Services	20
1.4 Servicios Web	21

1.5	Servicios de Base de datos	22
1.5.1	Características de MySQL	22
2	Topología Propuesta	24
2.1	Factores funcionales	24
2.1.1	Redes Virtuales	24
2.2	Topología	24
2.2.1	Recursos de software y hardware utilizados	27
2.2.2	Recursos de hardware utilizados para el despliegue de OpenNebula	27
2.2.3	Recursos utilizados para la nube pública en AWS	27
2.2.4	Recursos utilizados para la nube pública en Azure	28
3	Metodología.	29
3.1	Metodología y configuración de las plataformas	29
3.2	Descripción general de la metodología de enrutamiento VPN sitio a sitio	29
3.3	Flujo de tráfico	30
4	Análisis y Presentación de Resultados	40
4.1	Directorio Activo	40
4.2	Base de datos - Postgres	45
5	Conclusiones y Recomendaciones	53
5.1	Conclusiones	53
5.2	Recomendaciones	54
6	Anexos	55
6.1	Instalación y configuración de OpenNebula	55
6.2	Configuración para la replicación	66
6.2.1	Configuración para permitir la salida a internet	66
6.3	Configuración de instancias AWS y Azure	67
6.4	Instalación y configuración de OpenVPN	75
6.4.1	Instalación de OpenVPN Server	76
6.4.2	Instalación de OpenVPN Client	78
6.4.3	Configuración de OpenVPN	80
6.5	Azure	104
	Bibliografía	107

Lista de Figuras

2-1	Topología Propuesta. Fuente Autor	25
2-2	Topología Implementada. Fuente Autor	26
3-1	Diagrama de VPN sitio a sitio. Fuente [1]	30
3-2	Diagrama de VPN sitio a sitio. Fuente [1]	31
3-3	Diagrama de VPN sitio a sitio [1]	32
3-4	Diagrama de VPN sitio a sitio. Fuente [1]	33
3-5	Diagrama de VPN sitio a sitio. Fuente [1]	34
3-6	Diagrama de VPN sitio a sitio. Fuente [1]	35
3-7	Diagrama de VPN sitio a sitio. Fuente [1]	36
3-8	Diagrama de VPN sitio a sitio. Fuente [1]	37
3-9	Diagrama de VPN sitio a sitio. Fuente [1]	38
4-1	Se comprueba que el servicio este activo. Fuente Autor	40
4-2	Lista de usuarios en archivo csv. Fuente Autor	41
4-3	Salida que se muestra al ejecutar el comando. Fuente Autor	41
4-4	Script usado para la importación de archivos. Fuente Autor	42
4-5	Salida al ejecutar el script. Fuente Autor	42
4-6	Usuarios en la nube privada. Fuente Autor	43
4-7	Usuarios en la nube pública. Fuente Autor	43
4-8	Analizando el estado de la replicación. Fuente Autor	44
4-9	Inicio del proceso en OpenNebula. Fuente Autor	45
4-10	Inicio del proceso en AWS. Fuente Autor	45
4-11	Conexión a la base de OpenNebula. Fuente Autor	46
4-12	Conexión a la base de AWS. Fuente Autor	46
4-13	Inserción de datos en la base de OpenNebula. Fuente Autor	47
4-14	Datos replicados de manera correcta. Fuente Autor	47
4-15	Consulta SQL en OpenNebula. Fuente Autor	48
4-16	Resultado de la consulta en OpenNebula. Fuente Autor	48
4-17	Consulta SQL en AWS. Fuente Autor	49
4-18	Resultado de la consulta en AWS. Fuente Autor	49
4-19	Consulta en OpenNebula. Fuente Autor	50
4-20	Resultado de la consulta en OpenNebula. Fuente Autor	50
4-21	Consulta en AWS. Fuente Autor	51

4-22 Resultado de la consulta en AWS. Fuente Autor	51
4-23 Replicación usando DBeaver. Fuente Autor	52
6-1 Se deshabilita SELINUX. Fuente Autor	56
6-2 Se agrega los repositorios. Fuente Autor	56
6-3 Se deshabilita SELINUX. Fuente Autor	57
6-4 Se agrega los repositorios. Fuente Autor	57
6-5 Instalación de las gemas. Fuente Autor	58
6-6 Se cambia la contraseña por defecto. Fuente Autor	58
6-7 Instalación de paquetes de virtualización. Fuente Autor	59
6-8 Eliminando de todos los archivos dentro de esta ruta. Fuente Autor	60
6-9 Instalación de las llaves. Fuente Autor	60
6-10 Instalación de las llaves. Fuente Autor	61
6-11 Cambio de clave en el nodo Maestro. Fuente Autor	61
6-12 Cambio de clave en el nodo Esclavo. Fuente Autor	62
6-13 Se copia la llave pública al nodo esclavo. Fuente Autor	62
6-14 Se copia la llave pública al nodo maestro. Fuente Autor	62
6-15 Creación de la carpeta que contendrá las llaves autorizadas. Fuente Autor . .	63
6-16 Creación de la carpeta que contendrá las llaves autorizadas. Fuente Autor . .	64
6-17 Se edita el archivo de configuración de SSH. Fuente Autor	64
6-18 Creación del nodo para despliegue de las maquinas virtuales. Fuente Autor .	65
6-19 Despliegue de OpenNebula. Fuente Autor	65
6-20 Configuración de Port Forwarding para la salida a internet. Fuente Autor . .	66
6-21 Dashboard de Opennebula y EC2. Fuente Autor	67
6-22 Instancia AWS. Fuente Autor	68
6-23 Instancia Azure. Fuente Autor	69
6-24 Instalación Postgres. Fuente Autor	70
6-25 Cambio en la dirección de escucha, nodo maestro. Fuente Autor	71
6-26 Cambio en la dirección de escucha, nodo secundario. Fuente Autor	72
6-27 Configuración nodo maestro. Fuente Autor	73
6-28 Configuración del archivo de postgres. Fuente Autor	74
6-29 Configuración del archivo de postgres. Fuente Autor	75
6-30 Creación del archivo recovery. Fuente Autor	75
6-31 Iniciamos Postgres. Fuente Autor	75
6-32 Instalación OpenVpn. Fuente Autor	76
6-33 Tap de OpenVpn. Fuente Autor	77
6-34 Finalización de instalación OpenVpn. Fuente Autor	77
6-35 Instalación OpenVpn Client. Fuente Autor	78
6-36 Tap de OpenVpn Client. Fuente Autor	79
6-37 Finalización de instalacion OpenVpn Client. Fuente Autor	79

6-38	Creación de llave maestra. Fuente Autor	82
6-39	Creación de llaves en el servidor. Fuente Autor	84
6-40	Generación de la encriptación. Fuente Autor	85
6-41	Copia de archivos. Fuente Autor	86
6-42	Copia de archivos. Fuente Autor	87
6-43	Copia de archivos. Fuente Autor	88
6-44	Conexiones de red. Fuente Autor	89
6-45	Configuración del Firewall. Fuente Autor	93
6-46	Selección de puerto. Fuente Autor	94
6-47	Se permite la conexión. Fuente Autor	95
6-48	Se escoge los perfiles. Fuente Autor	96
6-49	Se escoge un nombre. Fuente Autor	96
6-50	Certificados. Fuente Autor	98
6-51	Configuración de usuarios. Fuente Autor	99
6-52	Configuración de OpenVPN Client. Fuente Autor	100
6-53	OpenVpn site to site. Fuente Autor	102
6-54	Máquinas virtuales de Azure. Fuente Autor	104
6-55	Listado de máquinas virtuales. Fuente Autor	105
6-56	Apartado de redes. Fuente Autor	105
6-57	Regla de entrada. Fuente Autor	106

Lista de Tablas

2-1	Recursos Utilizados en Hardware y Software. Fuente Autor	27
2-2	Recursos Utilizados en Hardware. Fuente Autor	27
2-3	Recursos Utilizados en Hardware. Fuente Autor	27
2-4	Recursos Utilizados en Hardware. Fuente Autor	28

Glosario de términos

Multicloud: En [2] indica que es un enfoque compuesto por más de un servicio de nube o de nubes, que proporcionan por lo menos dos proveedores de nube pública o privada .

Replicación: En [3] explica que la replicación permite proporcionar alta disponibilidad y tolerancia a fallos en cuanto a sistemas distribuidos .

Cloud Computing: En [4] el autor explica que es un modelo que permite el acceso bajo demanda y a través de la red a un conjunto de recursos compartidos y configurables que pueden ser rápidamente asignados y liberados con una mínima gestión .

Active Directory: El autor en [5] indica que es una manera de organizar y gestionar todos los elementos de una red informática: ordenadores, grupos, usuarios, dominios, políticas de seguridad, y cualquier tipo de objetos definidos para el usuario .

Google Cloud Platform: En [6] explica que es conjunto de servicios de nube pública que se basa en una enorme y sólida infraestructura. GCP permite a los desarrolladores utilizar estos servicios accediendo a la API RESTful de GCP que se describe a través de páginas HTML en su sitio web.

Amazon Web Services: El autor en [7] detalla que es un conjunto de servicios de computación a distancia que, en conjunto, conforman una plataforma de computación en la nube, ofrecida a través de Internet por amazon.com.

PYMEs: En [8] se dice que se conoce como PYMES al conjunto de pequeñas y medianas empresas.

Snapshot: En [9] se da un concepto donde explica que es un elemento de seguridad informática complementarias a las copias de seguridad o backups.

TI:El término tecnologías de la información en [10] explica que hace referencia al uso de equipos de telecomunicaciones y computadores usados para fines como la transmisión, el procesamiento y el almacenamiento de datos.

Resumen

Hace ya casi de una década en las áreas informáticas de todo el mundo, apareció un nuevo término que desde aquel entonces se lo conoce como Cloud Computing. El uso de Cloud Computing durante estos años ha ido en aumento y está relacionado de manera directa con los beneficios en los centros de datos. El desarrollo de la computación en la nube ha tenido un gran impacto en la industria de TI. Si se habla en términos de energía y hardware, se hace referencia a que se necesita tiempo de procesamiento relevante que va de la mano con un costo limitado para la escalabilidad. La mayor parte del desarrollo en el ámbito del cloud está a cargo de gigantes de TI como lo son IBM, Amazon y Google, que hacen cada día grandes esfuerzos para proporcionar y brindar soluciones en la nube que sean rentables.

Los autores en [2] manifiestan que un usuario de la nube no necesita desarrollar o mantener un gran edificio de infraestructura o personal de TI, sino que le es mucho más fácil pagar solo en tiempo real por los recursos que usa en el ámbito de la nube. Los servicios de computación en la nube son escalables y se adaptan de acuerdo a las necesidades de los usuarios.

En [3] explica que durante estos últimos años, la evolución y la innovación en la computación en la nube ha permitido convertirse en uno de los campos con mayor crecimiento. La investigación en el campo ha permitido ofrecer servicios operativos con mucho menos personal de TI, un menor mantenimiento y una gran implementación de manera más rápida. Lo cual ha significado un impacto positivo en los entornos de aprendizaje y enseñanza debido al auge de la computación en la nube.

Los servicios en la nube se basan en los requisitos que presenta el usuario, el usuario puede requerir de calidad de servicio (QoS), usabilidad, almacenamiento de datos, la complejidad de las aplicaciones, middleware, escalabilidad, lenguaje de programación. Además en [4] se explica que una sola nube no puede cumplir con todos los requisitos, y comenta que el término multi-nube, es similar a un cielo nublado cuando posee diferentes colores y formas, la computación en la nube no debería terminarse a través de una sola nube que lleva a diferentes implementaciones y dominios administrativos. La computación en múltiples nubes (multi-cloud) aprovisiona el uso de múltiples entornos de nube independientes.

“La mayoría de las organizaciones adoptan una estrategia multinube con el deseo de evitar el bloqueo de proveedores o de aprovechar las mejores soluciones”.

Palabras clave: (Multicloud, Cloud Computing, Replicación, Google Cloud Platform, Amazon Web Services, Pyme..

Abstract

Almost a decade ago in computer science areas around the world, a new term appeared that since then has been known as Cloud Computing. The use of Cloud Computing during these years has been increasing and is directly related to the benefits in data centers. The development of cloud computing has had a great impact on the IT industry. If you talk in terms of power and hardware, you are referring to the need for relevant processing time that goes hand in hand with limited cost for scalability. Most of the development in the cloud space is carried out by IT giants such as IBM, Amazon and Google, who go to great lengths every day to provide and deliver cost-effective cloud solutions.

The authors in [2] state that a cloud user does not need to develop or maintain a large infrastructure building or IT staff, but it is much easier for him to pay only in real time for the resources he uses in the field of Cloud. Cloud computing services are scalable and adapt according to user needs.

In [3] he explains that in recent years, evolution and innovation in cloud computing has made it one of the fastest growing fields. Research in the field has made it possible to offer operational services with much fewer IT staff, less maintenance, and a large deployment faster. This has had a positive impact on learning and teaching environments due to the rise of cloud computing.

Cloud services are based on the requirements presented by the user, the user may require quality of service (QoS), usability, data storage, the complexity of the applications, middleware, scalability, programming language. Also in [4] explains that a single cloud cannot meet all the requirements, and comments that the term multi-cloud is similar to a cloudy sky when it has different colors and shapes, cloud computing should not be terminated through of a single cloud that leads to different implementations and administrative domains. Multi-cloud computing provisions the use of multiple independent cloud environments.

”Most organizations adopt a multi-cloud strategy with the desire to avoid vendor lock-in or take advantage of the best solutions.”

Keywords: Multicloud, Cloud Computing, Replication, Google Cloud Platform, Amazon Web Services, Pyme.

Introducción

El Instituto Nacional de Estándares y Tecnología (NIST) define la computación en la nube como un “modelo para otorgar a los usuarios acceso a un grupo compartido de recursos computacionales configurables que incluyen redes, servidores, almacenamiento, aplicaciones y servicios que se pueden aprovisionar de manera rápida y de esta manera adaptarse a nuevos servicios pero con una poca interacción entre la administración o el proveedor de servicios”. En [11] explica en pocas palabras que la computación en la nube es simplemente una forma de acceder y gestionar de manera remota los recursos informáticos.

En [8] el autor indica que esta tecnología en la actualidad esta presente e integrada en muchas empresas, ya que las redes en la nube de terceros permiten a los usuarios de las empresas, centrarse en los elementos principales del negocio, sin existir la necesidad de tener que preocuparse por temas como el mantenimiento o la infraestructura informática. Además en [8] indica que una ventaja importante de este tipo de tecnología es que, a todos los recursos computacionales se los puede utilizar de una manera mucho más eficiente, lo que brinda resultados mucho mas rápidos.

El autor en [6] da un concepto de que la nube múltiple tiene como enfoque principal, el evitar la dependencia de un único proveedor de nube, y presenta algunas características como lo son:

- Diversidad: La cual brinda una opción adicional de múltiples entornos, ademas brinda flexibilidad a las empresas y evita la dependencia de los agentes.
- Tolerancia a fallas y proximidad: Aunque el tema principal de todos los proveedores de nube es evitar los desastres, la ejecución en múltiples nubes permite al cliente una mayor garantía de alta disponibilidad.
- Cumplimiento privado: La nube múltiple brinda beneficios y ayuda a las empresas a lograr sus objetivos específicos, al cumplir con los parámetros de gobernanza y la gestión de riesgos.
- Modelo de precios flexible: Si se da el caso de que una empresa no se comprometa con un proveedor de nube, dicha empresa es libre de seleccionar el proveedor de nube más asequible o personalizar sus modelos de precios según sus especificaciones y requerimientos.

En [11] se indica que en la actualidad existen tres modelos de servicios de computación en la nube que dominan la industria:

- Software como servicio (SaaS): Donde el cliente puede operar un software en el sistema, en la nube del proveedor, por lo general a través de Internet .
- Plataforma como servicio (PaaS): Aqui la prestación permite al cliente utilizar la red en la nube como plataforma, para sus propios programas de desarrollo o adquiridos.
- Infraestructura como servicio (IaaS): Aqui el cliente cuenta con máquinas y almacenamiento pero de manera virtual.

Por otro lado en [12] el autor detalla de que la computación en la nube se está fortaleciendo para ser mucha más eficiente y está ganando espacio entre las necesidades actuales de las empresas, ya que se basa en la gestión virtual dentro y fuera de las instalaciones de las empresas. Esto genera una complejidad significativa en la gobernanza, el cumplimiento, el autoservicio, la gestión de recursos, la planificación de la capacidad y los controles financieros. Cada empresa, desde la más pequeña hasta la más grande, poseen una variada gama de soporte y necesidades que requieren de los proveedores actuales de la nube.

Además en [7] se detalla que la nube múltiple se presenta a las empresas y sus usuarios como una gran solución, ya que permite la gestión operativa, la optimización de costes, las fortalezas de este tipo de estrategia de múltiples nubes ayudan a las empresas a mantener la flexibilidad y tomar la decisión correcta al elegir las herramientas de administración, el cumplimiento, la seguridad y la arquitectura.

Justificación

La Nube se ha vuelto el modelo computacional por excelencia, las necesidades de la transformación digital y la globalización de los negocios marcan la pauta del uso de esta tecnología. Hoy en día, las organizaciones están implementando esquemas de multicloud, las empresas capturan los beneficios de cada plataforma y mitigan así los inconvenientes que una u otra infraestructura puede presentar.

Tanto las grandes, como las pequeñas y medianas empresas están migrando de manera progresiva a la nube. En [10] se indica que muchos sectores de la población usan la nube, ya sea cuando se envía un correo electrónico, cuando se escucha música en Spotify (servicio de streaming audio), cuando se consulta la posición geográfica actual en Google Maps, o simplemente cuando se ve una fotografía o imagen en Google, ya sea que se lo haga en una Pc de escritorio o en un smartphone. Todo esto unido al uso de almacenamiento masivo en la red de datos cada vez que se hace uso de la información en dichos servicios .

Cuando se usa el servicio de multicloud, se tiene grandes beneficios a favor de la empresa, por eso en [13] se mencionan los siguientes beneficios:

- Redundancia: Significa que la empresa va a tener sus datos duplicados en las nubes de las cuales disponen. Así de esta manera, el tener más de una nube blindará a la empresa ante problemas y mantiene asegurada la operación en todo momento .
- Escalabilidad: Brinda el acceso a más recursos debido a la alta demanda, lo cual significa que se puede aumentar o disminuir los servicios según la empresa lo vaya necesitando.
- Competitividad: Hoy en día la amplia oferta de proveedores propicia una mayor competitividad y esto conlleva a más posibilidades de negociar los términos contractuales y de esta manera una mayor generación de modelos de servicios.
- Servicios: Entre los diferentes proveedores, algunos tienen puntos fuertes en cuanto a la rapidez y otros en temas de seguridad. Bajo este esquema las empresas pueden aprovechar las características de cada proveedor y así, elegir un proveedor más seguro para proteger sus datos más confidenciales, y hacer uso de un proveedor más económico para una aplicación con menos requisitos de seguridad.

Objetivos

Objetivos Generales

Generar e implementar una metodología para el correcto uso de una aplicación de software que nos permitirá la migración de los datos desde una nube privada hacia una nube pública.

Objetivos Especificos

- Recolectar información que atraen a las empresas a migrar hacia la nube.
- Recolectar información sobre ventajas y desventajas que genera la migración hacia los diferentes tipos de nube.
- Establecer una propuesta metodología para migrar hacia la nube, dependiendo del tipo de empresa.
- Implementar la metodología escogida y establecer los resultados.

1 Estado del Arte y Fundamentación Teórica.

En esta sección se realizará la fundamentación teórica que se ha realizado para fundamentar el desarrollo de este proyecto.

1.1. Cloud Computing

En este capítulo, se explica que es Cloud Computing o Computación en la nube, su arquitectura, características y plataformas de código abierto.

1.1.1. Definición de Computación en la Nube

El autor en [9] explica que durante los últimos años, la computación en la nube ha transformado el panorama de lo que es TI, en el ámbito de los usuarios y de igual manera para las empresas, desde la forma en que se accede, se almacena y comparte información hasta la manera en como se comunica, colabora y procesa los datos. Esto ha llevado a niveles sin precedentes de I + D y ha dado lugar a numerosas conferencias académicas y de la industria. A medida que ha evolucionado la computación en la nube, ha ido creciendo su expansión para incluir prácticamente todos los aspectos de los sistemas distribuidos modernos, incluyendo el desarrollo, la administración, la implementación y la orquestación de los recursos en el tema de redes, almacenamiento y computación.

En [5] se indica que la Computación en la nube esta compuesta de un conjunto de servicios que están listos y presentes en Internet. En dichos servicios, se proporcionan disposiciones fundamentales como lo son computación y almacenamiento de datos. A la computación en la nube se la puede definir como un modelo informático en el que se presenta un nuevo ámbito empresarial, que proporciona una nueva visión de los sistemas informáticos de alto rendimiento y basados en Internet en la que los recursos relacionados con la informática se proporcionan en forma de servicio .

1.1.2. Nube Pública

Los autores de [14] indican que la nube pública es un conjunto de hardware, redes, almacenamiento, servicios, aplicaciones e interfaces que pertenecen y son operados por un tercero, para poder ser utilizados por otras empresas o usuarios. Estos proveedores comerciales crean un centro de datos que es altamente escalable. Las nubes públicas son viables porque ofrecen muchos recursos que están siempre disponibles, los usuarios o empresas que contratan este tipo de nubes públicas tienen la opción de seleccionar, optimizar y utilizar de forma rápida aquellos recursos que se ajustan a las necesidades de las aplicaciones que van a ejecutar en la nube pública.

En una nube pública, el proveedor de servicios de la nube, es el dueño de la infraestructura física, administra y gestiona los servicios y recursos.

Además en [15] se indica que los proveedores de nubes públicas ofrecen cada vez más instancias dedicadas, sin multiarrendamiento, dentro de su centro de datos. En este tipo de casos se asigna sus propias máquinas y almacenamiento dentro del centro de datos del proveedor de la nube. En este caso las cargas de trabajo están físicamente aisladas, pero en algún punto compartirán algunas de las redes con otros clientes de la nube.

En [16] indica que entre las ventajas de las nubes públicas están:

- Ahorro de costos: Al usar una nube pública las empresas pueden reducir los costos de operaciones de TI. Ya que subcontratan estos costos a un tercero (proveedor), que los puede manejar de manera más eficiente. Este tipo de nubes suelen costar menos que las privadas, porque el proveedor de servicio en la nube puede maximizar el uso del hardware .
- Menos administración del servidor: Si la empresa usa una nube pública, los equipos internos no pierden tiempo en la administración de servidores, como lo hacen con los centros de datos locales o con las nubes privadas internas.
- Seguridad: Es muy probable que las pequeñas y medianas empresas no tengan los recursos suficientes para implementar fuertes medidas de seguridad. Al hacer uso de un servicio de nube pública, pueden externalizar algunos aspectos de la seguridad informática a un proveedor más grande y que posee muchos más recursos.

La nube pública posee algunas desventajas e inconvenientes, y en [17] enumera las siguientes:

- La nube pública obliga a compartir los equipos, servidores y red con otros clientes que también están almacenando sus datos en dicha estructura.
- En la nube pública el cliente no tiene el control de la seguridad de sus datos.

- El funcionamiento adecuado de los servidores, así como las copias de seguridad, dependen del proveedor del servicio, por lo cual no se garantiza que no existan períodos de inactividad.

1.1.3. Nube Privada

En [18], el autor describe a la nube privada como el funcionamiento de varias nubes pero de un modo similar a una red o un centro de datos privado. Aquí la infraestructura de la nube es gestionada por una organización pero de manera única, ya sea esta de manera directa o por terceros.

Según el autor [18], en una nube privada, la empresa que hace de cliente establece un entorno de virtualización pero lo hace en sus propios servidores, es decir en sus propios centros de datos o en los de un proveedor de servicios. Las empresas deben adquirir, construir y administrar la nube para de esta manera poder tener el control de la misma, y como consecuencia de esto, sus costes y condiciones de gestión serán obviamente más altos. Los clientes de organizaciones y empresas que optan por una nube privada deben asumir las responsabilidades del funcionamiento de esta y además con esto logran tener el control sobre su infraestructura y se benefician con todas las ventajas de la virtualización.

En [19] el autor indica que la nube privada está muy automatizada y se centra en temas de la seguridad, donde la automatización reemplaza a los procesos más manuales de gestión de servicios de TI para dar soporte a sus clientes. Así, de esta manera, los procesos empresariales se pueden implementar dentro del software para que el entorno sea más predecible y a su vez manejable.

Así mismo, en [19] se indica que las empresas o usuarios que optan por una nube privada poseen algunas ventajas como:

- Necesidades específicas de seguridad: En el caso de las empresas que están sujetas a requisitos de cumplimiento normativo, puede que se necesite una nube privada para poder lograr la conformidad, lo cual la lleva a optar por usar una nube privada para poder almacenar datos confidenciales y tener un mayor control de la seguridad.
- Conocimientos técnicos: El manejo de una nube privada exige más inversión y conocimientos técnicos para poder gestionar los recursos y problemas técnicos, que en un modelo de nube pública, lo haría el proveedor de nube.
- Necesidades predecibles de recursos: Una de las mayores ventajas de una nube pública es la elasticidad, o sea su capacidad para ampliar o reducir recursos de forma rápida según sean las necesidades del cliente o empresa. Sin embargo, algunas empresas no

requieren esta elasticidad ya que hacen uso de los recursos de manera relativamente uniforme.

1.1.4. El modelo de nube híbrida y multicloud

El autor en [20], explica que una nube híbrida es una combinación de nube privada pero que esta combinada con el uso de servicios de nube pública, donde ambos entornos de nube trabajan de manera conjunta para resolver los problemas de la empresa. Y cuyo objetivo es crear un entorno de nube híbrida, la cual combina servicios y datos de diversos modelos de nube para poder crear un entorno informático pero unificado, automatizado y gestionado de manera correcta. En dicho entorno de nube híbrida, los usuarios finales no diferenciarán si están utilizando servicios locales o de la nube.

Además de la nube híbrida, el autor en [20] manifiesta que el término multicloud hace referencia a cuando se hace uso de dos o más nubes públicas dentro de una empresa. A medida que las empresas se encontraban utilizando varias nubes, los equipos de finanzas, operaciones y TI necesitaban una manera de obtener visibilidad, control y elección entre las nubes. Por lo tanto, la gestión de múltiples nubes está se esta priorizando como una consideración importante.

En [18] se describe que la infraestructura de una nube híbrida es una combinación de las nubes públicas y privadas. Un entorno de nube privada consta de múltiples proveedores internos o externos y es un despliegue posible para organizaciones. Con una nube híbrida las empresas pueden ejecutar sus aplicaciones no fundamentales en una nube pública, mientras mantienen las aplicaciones que sean fundamentales y los datos sensibles internos para la empresa, en una nube privada.

1.1.5. Características

En [21] se indica que un sistema de Cloud Computing tiene algunas características que son muy importantes y que a su vez prometen futuras aplicaciones, entre estas características esenciales están:

- Autoservicio: Donde el usuario está permitido para poder hacer uso de más capacidades de procesamiento o almacenamiento de la información, sin pedir de manera expresa a su proveedor de servicio.
- Amplio acceso a la red: Se puede acceder a la red desde diferentes redes y dispositivos.

- Agrupación y reserva de recursos: Aquí existe un conjunto de recursos que son compartidos por todos los usuarios, de acuerdo a sus necesidades, y esto implica que, en cada momento los recursos reservados puedan ser diferentes.
- Rapidez y elasticidad: Esto permite acceder a los nuevos recursos de forma ágil y de una manera aparentemente ilimitada.
- Servicio medible y supervisado: Se controla el uso y en todo momento se puede conocer, de forma transparente, el nivel de los recursos que están siendo utilizados.

1.1.6. Arquitectura de la Computación en la Nube

La arquitectura de la computación en la nube está basada en hardware la misma que engloba la plataforma y aplicaciones que traen consigo los siguientes tipos.

Tipos de Servicios

En [22] se indica que para poder entender el funcionamiento del cloud computing es primordial conocer los tres servicios en la nube y cada uno de estos con sus necesidades, entre los cuales se pueden clasificar en: servicios de software e infraestructura o servicios de hardware.

Software as a Service (SaaS): En [22] se indica que SaaS es un modelo de distribución de software, el cual proporciona a los clientes el acceso a éste, pero a través de la red (generalmente Internet). De esta manera, los usuarios no tienen que preocuparse de la configuración, implementación o mantenimiento de las aplicaciones, ya que todas estas labores pasan a ser responsabilidad del proveedor. Las aplicaciones distribuidas a través de este modelo de Software como Servicio pueden llegar a cualquier empresa sin importar su tamaño o ubicación geográfica.

Por otra parte el autor de [18] define a SaaS como un modelo donde el usuario no tiene que preocuparse de saber donde está alojado el software, o que tipo de sistema operativo se utiliza o en que lenguaje está escrito. Ya que el usuario no tiene que instalar ningún programa de software como se lo hace en el modelo tradicional, tampoco gestiona ni controla la infraestructura fundamental de la nube, ni la red, sistemas operativos, servidores, mucho menos las características o funcionalidades de las aplicaciones individuales.

Platform as a Service (PaaS): En [22] se define PaaS como un entorno de desarrollo e implementación completo en la nube, y que cuenta con recursos que permiten hacer uso desde aplicaciones sencillas, basadas en la nube hasta aplicaciones empresariales muy complejas habilitadas para la nube. Aquí el usuario únicamente compra los recursos que necesita a un proveedor de servicios en la nube, a los que accede a través de una conexión segura a

Internet, pero con la finalidad de que unicamente paga por el uso que hace de ellos.

Desde otro concepto, en [18] el autor explica que en este modelo, los sistemas PaaS son muy útiles ya que facilitan de gran manera a los desarrolladores y empresas pequeñas desplegar aplicaciones basadas en la web, sin el coste y la complejidad que conllevaria la compra de servidores, configuraciones y la puesta en funcionamiento. En [18] se da detalles de los beneficios de PaaS residen en el número creciente de personas que lo pueden desarrollar, mantener y desplegar aplicaciones web, PaaS ofrece democratizar el desarrollo de aplicaciones. En el modelo de plataforma como servicio, el proveedor ofrece un entorno de desarrollo a los desarrolladores de aplicaciones, quienes desarrollan aplicaciones y ofrecen sus servicios a través de la plataforma del proveedor.

Infrastructure as a Service (IaaS): En [22] se define a la IaaS acceso por parte del cliente a servocops como almacenamiento, sistema operativo, red, entre otras características con el fin de que sus clientes puedan empezar un nuevo proyecto con la infraestructura que sea necesaria, los proveedores permiten a sus clientes tener escalabilidad y elasticidad ya que pueden solicitar los recursos que sean necesarios. Como nota importante se debe decir que el usuario no puede administrar o controlar la infraestructura de la nube.

En [18] los autores presenta otra definición de IaaS y detalla que este modelo ofrece espacio de almacenamiento, capacidad de proceso, servidores, en pago por uso.

Sin dejar a un lado que también se puede incluir, la entrega de sistemas operativos y tecnología de virtualización para poder gestionar los recursos.

Al usuario se le brinda y se le facilita la capacidad de almacenamiento, procesamiento, redes y otros recursos informáticos los cuales son fundamentales, ya que el consumidor es capaz de desplegar y así poder ejecutar software específico, dentro de los cuales se puede incluir sistemas operativos y aplicaciones.

El estudio de la computación en la nube brinda soporte a varias tecnologías y redes como 5G tal como se describe en [23, 24, 25, 24, 26], así como permiten del desarrollo y soporte de proyectos basados en Internet de las Cosas (IoT), como se describe por los autores en [27, 28, 29, 30, 31].

1.2. Plataformas Open Source

1.2.1. OpenNebula

En [32] se describe a OpenNebula como una plataforma flexible y de código abierto potente, muy fácil de usar, que posee recursos compartidos, actualizaciones constantes, libertad al momento de alguna modificación. Además dispone de actualizaciones en el ámbito de la virtualización de data centers para un desarrollo básico, esto permite al usuario o empresas el

poder desplegarlo en una infraestructura, y capacidades pre-existentes, además no depende de hypervisores, el autor en [32] también menciona que OpenNebula facilita al usuario la creación de cualquier tipo de nubes: públicas, privadas o híbridas.

En [33] se indica que OpenNebula es una plataforma adaptable e interoperable para la virtualización de data centers, permite la gestión de la nube empresarial y gracias a esto proporciona una solución más simple y flexible para la gestión integral de la infraestructura virtualizada en el centro de datos para habilitar las nubes IaaS locales. OpenNebula también es compatible con Hybrid Cloud y gracias a esto permite combinar la infraestructura local con infraestructura pública basada en la nube, lo que hace posible entornos de alojamiento que son altamente escalables, y adicional a esto es compatible con nubes públicas, al proporcionar interfaces en la nube lo cual permite exponer su funcionalidad en máquinas virtuales (VM), almacenamiento y administración de redes.

Además en [32], se indica que OpenNebula permite al usuario el poder gestionar:

- Cualquier aplicación: Esto permite combinar aplicaciones en contenedores de Kubernetes y Docker Hub, con cargas de trabajo de máquinas virtuales en un entorno compartido y común, gracia a esto ofrece lo mejor de ambos como la tecnología de virtualización robusta y la orquestación de contenedores de aplicaciones.
- Cualquier infraestructura: Ya que desbloquea el poder de una plataforma híbrida, perimetral y de múltiples nubes, y combina la expansión de nube privada con recursos de infraestructura de otros proveedores de nube pública y bare-metal como AWS.
- Cualquier virtualización: Permite al usuario el poder integrar varios tipos de tecnologías de virtualización para poder satisfacer las necesidades de carga de trabajo, incluyendo máquinas virtuales VMware y KVM para nubes totalmente virtualizadas, contenedores del sistema LXC para nubes de contenedores y microVM Firecracker para implementaciones sin servidor.
- En cualquier momento: Brinda la facilidad de agregar y eliminar automáticamente nuevos recursos para satisfacer la alta demanda o para poder implementar estrategias tolerantes a fallas o requisitos de latencia.

El modelo OpenNebula para usuarios de la nube

En [33] se explica que Opennebula ha sido diseñado teniendo en cuenta la flexibilidad para poder adaptarse a las necesidades reales que tienen las empresas, y detallan algunos de los casos que son de uso básicos y modelos de aplicaciones que admite OpenNebula.

- Permite organizar máquinas virtuales , pero dependiendo del tipo de carga de trabajo, para esto se puede usar diferentes tipos de hipervisores. OpenNebula se puede implementar sobre una infraestructura VMware vCentre, pero también permite administrar cargas de trabajo basadas en KVM.
- Proporciona multi-tenancy por diseño, esto significa que ofrece diferentes tipos de interfaces para los usuarios, pero según sus roles dentro de la empresa o el nivel de experiencia o funcionalidad requerida.
- Permite administrar desde VM individuales hasta servicios que son mucho mas complejos de varios niveles, y que estan compuestos por varias VM que requieren reglas de elasticidad sofisticadas y adaptabilidad dinámica.
- Las aplicaciones basadas en VM se crean a partir de imágenes y plantillas que están disponibles en OpenNebula Public Marketplace, pero que también los propios usuarios las pueden crear y ser compartidas por el administrador de la nube mediante un mercado corporativo privado.
- Este modelo permite a los usuarios la creación rápida de instancias de aplicaciones y servicios complejos, por ejemplo, la implementación de clústeres de Kubernetes..

1.2.2. Eucalyptus

En [34] el autor brinda detalles, así como, se indica que Eucalyptus ofrece varias formas de implementar, administrar y mantener una propia colección de recursos virtuales como máquinas, red y almacenamiento. Eucalyptus brinda compatibilidad de API con Amazon Web Services, para permitir hacer uso de las herramientas y comandos familiares para aprovisionar nuestra nube.

Entre las características de Eucalyptus que en [34] se menciona, se lista las siguientes:

- Abstracciones de almacenamiento basadas en bloques y depósitos
- Capacidades de autoservicio
- Interfaz basada en web
- Administración de recursos
- Gestión de claves SSH

- Gestión de imágenes
- Compatibilidad con el sistema operativo invitado Linux
- Gestión de direcciones IP
- Gestión de grupos de seguridad
- Gestión de instantáneas y volúmenes

1.2.3. OpenStack

En [35] se indica que OpenStack es un sistema operativo en la nube, la cual controla grandes grupos de recursos informáticos, de almacenamiento y de redes en un centro de datos, todos estos administrados y aprovisionados a través de API con mecanismos de autenticación comunes.

Por otro lado en [35], se dice que OpenStack brinda y dispone de un panel de control, el cual brinda a los administradores el control y al mismo tiempo permite a los usuarios suministrar recursos a través de una interfaz web.

Ademas los componentes adicionales brindan orquestación, administración de fallas y administración de servicios, entre otros servicios, para poder garantizar una alta disponibilidad de las aplicaciones de usuario.

En [36] se menciona que OpenStack posee algunas características y entre ellas tenemos las siguientes:

- Gestión básica del centro de datos físicos: Esto proporciona una abstracción que permite albergar servicios más especializados, tanto de OpenStack como de terceros.
- Virtualización de hardware: Para cualquier servicio que normalmente es proporcionado por una pieza especializada de hardware, OpenStack tiene como objetivo, el proporcionar una API que brinda al usuario un control definido por software para asignar el recurso en un entorno de múltiples inquilinos.
- Escalado continuo e infinito: OpenStack proporcionar a los desarrolladores de aplicaciones interfaces que permiten en un principio, escalar de manera eficiente desde cargas de trabajo muy pequeñas hasta muy grandes, sin la necesidad de tener que rediseñar sus aplicaciones.
- Integración personalizable: OpenStack no impone ningún modelo o arquitectura de implementación de manera particular en las aplicaciones. Cada aplicación posee requisitos únicos, y OpenStack satisface esto al permitir que los servicios se conecten de manera

conjunto en el 'espacio de usuario', a través de API públicas, sin necesidad de acciones programadas tomadas detrás de escena que solo admiten modelos de implementación predefinidos.

- Interfaz gráfica de usuario: Una GUI y la presentación de opciones y flujos de trabajo de manera gráfica permite descubrir capacidades de una manera que la lectura a través de la documentación de API o CLI no puede.

1.3. Plataformas de Cloud Propietarias

1.3.1. Google Cloud

Por otra parte en [37], el autor explica que Google Cloud es la suite de infraestructuras y servicios que Google utiliza pero a nivel interno y que actualmente se encuentra disponible para cualquier empresa, de tal manera que es aplicable a una gran multitud de procesos empresariales.

Si se habla de Google Cloud Platform (GCP), se hace referencia a todas las herramientas disponibles de Google en la nube que hasta ahora se las ofrecía de manera separada. Este conjunto de servicios ofrecen prestaciones muy dispares; desde machine learning hasta Inteligencia artificial pasando por el big data, todo englobado bajo el entorno del cloud computing.

En [38] se argumenta que Google Cloud Platform proporciona servicios de Big Data en forma de BigQuery, Cloud Dataflow, Google Cloud Dataproc, Cloud Datalab, Cloud Pub / Sub y Google Genomics. Y que además proporciona una potente plataforma de análisis de Big Data en la nube.

Así mismo, en [38] indica que Google Cloud ofrece ventajas para las empresas y pymes, entre las cuales están:

- Open Source: GCP es una tecnología flexible y escalable puesto que es de código abierto. Esto brinda la posibilidad de personalizar las herramientas al máximo, según las necesidades de cada empresa.
- Seguro: Google Cloud Platform cuenta con infraestructura de Google, la cual brinda casi un 100 por ciento de fiabilidad y disponibilidad. Sin existir ningún período de inactividad programado. Además usa sus redes privadas, por lo que los ciber-ataques son prácticamente imposibles.

- **Innovador:** Brinda herramientas modernas, lo cual permite a las empresas estar a la vanguardia de la transformación digital en ámbitos como el machine learning, big data o inteligencia artificial.
- **Asequible:** Algunas de las herramientas son gratis. Y las que son de pago, tienen precios muy competitivos con diferentes planes, para que de esta manera cualquier empresa, grande o pequeña tengan la opción de beneficiarse de todos los beneficios que ofrece Google cloud.

1.3.2. Microsoft Azure

El autor en [39] expresa que Azure ofrece un conjunto de servicios en la nube, perteneciente a la empresa Microsoft. Con Azure es posible almacenar información y crear, administrar e implementar aplicaciones en cloud. Para utilizar Azure es necesario el pago de una cuota que recoge los servicios contratados.

Ventajas de trabajar en la nube con Azure

De igual manera en [39] se detalla algunos beneficios que Azure ofrece a la empresa que decide contratar alguno de sus planes para trabajar en la nube, entre los cuales tenemos:

- **Almacenamiento y backup seguro en la nube:** Con Azure se dispone de un sistema de almacenamiento de datos en la nube, de esta manera garantiza la privacidad e integridad de la información. Lo cual significa una herramienta ideal para guardar la información y poder disponer de copias de seguridad externas a la empresa.
- **Reducción de costes:** Azure es un modelo de pago por uso, por lo que permite optimizar los servicios contratados y así pagar solo por aquello que realmente se va a utilizar.
- **Escalable y accesible:** Es posible cambiar las condiciones de Microsoft Azure de manera fácil y confiable, ya sea para aumentar o disminuir los servicios contratados, en esto Microsoft Azure proporciona toda la accesibilidad de un entorno cloud a una empresa.
- **Permite el trabajo híbrido:** Con Azure se puede trabajar de forma conjunta en un entorno local y en la nube, y de esta manera facilitar la integración de ambos ambientes para un trabajo mucho más efectivo y seguro.

1.3.3. Oracle Cloud

En [40], explica que es un conjunto de servicios en la nube complementarios, el cual permite al usuario o la empresa crear y ejecutar una amplia variedad de aplicaciones y servicios en un

entorno alojado de alta disponibilidad. En [40] se describe que Oracle Cloud ofrece funciones informáticas de alto rendimiento como lo son instancias de hardware físico y capacidad de almacenamiento en una red virtual de superposición flexible accesible de forma segura desde su red local.

Los autores de [41] indican que Oracle Cloud es una oferta de infraestructura/plataforma/software como servicio (IaaS/PaaS/SaaS) de 2da generación, y la misma esta diseñada con principios de seguridad.

En [41] dan detalles de estos principios y comentan que entre ellos se incluye la virtualización de red aislada y un entorno físico de implementación de hosts, el cual proporciona un aislamiento superior de los clientes en comparación con los diseños de nube pública de 1ra generación.

Entre los servicios que ofrece Oracle Cloud, en [40] se menciona los siguientes:

- **Analytics Cloud:** Ofrece a sus clientes y analistas empresariales funciones modernas, entre las cuales están, el análisis de autoservicio que funcionan con IA para la preparación de datos, la visualización, la generación de informes empresariales, los análisis aumentados y el procesamiento del lenguaje natural.
- **El gateway de API:** El mismo que permite crear interfaces HTTP/S, gestionadas para otros servicios entre los cuales se incluye Oracle Functions, Oracle Cloud Infrastructure Container Engine for Kubernetes y Oracle Cloud Infrastructure Registry.
- **Auditoría:** Ofrece una visibilidad de las actividades que están relacionadas con los recursos y el arrendamiento de Oracle Cloud Infrastructure. Los eventos del log de auditoría se los puede utilizar para poder realizar auditorías de seguridad, un seguimiento del uso y los cambios realizados en los recursos de Oracle Cloud Infrastructure.
- **Big Data:** Proporciona una estructura de software de código abierto para almacenar datos y ejecutar aplicaciones en clústeres que son de hardware comercial, como un servicio de nivel empresarial con seguridad integral, alto rendimiento, facilidad de gestión y capacidad de actualización.
- **Blockchain Platform Cloud:** Permite la creación de redes de cadena de bloques gestionadas y bajo la autorización para compartir datos de forma segura y en tiempo real y realizar transacciones de confianza entre socios comerciales.
- **Cloud Guard:** Es un servicio en la nube que sirve de ayuda a los clientes a supervisar, identificar, lograr y mantener una estrategia de seguridad sólida en Oracle Cloud. Si se detecta alguna deficiencia, Cloud Guard realiza sugerencias, ayudar a llevar a cabo acciones correctivas, según sea la configuración de la empresa.

- Cloud Shell: Es una terminal basada en un explorador gratuito al que pueden acceder desde la consola de Oracle Cloud y proporciona acceso a un shell de Linux con la CLI de Oracle Cloud Infrastructure, a través de una autenticación previa y otras herramientas de desarrollador útiles.
- Data Flow: Es un servicio gestionado en su totalidad con una interfaz de usuario que permite a los desarrolladores y científicos de datos crear, editar y ejecutar aplicaciones de Apache Spark a cualquier escala sin necesidad de clusters.
- Data Integration: Es un servicio totalmente gestionado que ayuda a ingenieros de datos y desarrolladores de ETL con tareas comunes de extracción, transformación y carga (ETL), como la incorporación de datos desde una variedad de activos de datos, la limpieza, transformación y remodelación de esos datos y, a continuación, su carga de una forma eficaz en los activos de datos de destino.
- Seguridad de los datos: Es un servicio en la nube totalmente integrado que se centra en la seguridad de los datos, y proporciona un conjunto de funciones completas e integradas para proteger los datos confidenciales y regulados de las bases de datos de Oracle Cloud. Dentro de las funciones se incluye la evaluación de seguridad, evaluación de usuarios, detección de datos, enmascaramiento de datos y auditoría de actividad.
- Data Science: Es una plataforma que permite a los científicos de datos crear, entrenar y gestionar modelos de aprendizaje automático en Oracle Cloud Infrastructure con Python y bibliotecas de aprendizaje automático de código abierto.

1.3.4. Amazon Web Services

El autor de [42] detalla que es un conjunto de herramientas y servicios de cloud computing de Amazon. El cual se lanzó oficialmente en 2006 y en junio de 2007 AWS ya contaba con una base de usuarios de aproximadamente 180 mil. Esto se debe gracias a la madurez del servicio frente a otros similares y las diversas herramientas disponibles.

En [42] se explica que Amazon Web Services ofrece los siguientes beneficios:

- Bajo coste: El cual ofrece precios bajos según su uso, sin ningún tipo de gasto anticipado, ni compromisos a largo plazo. Las empresas pueden construir y administrar una infraestructura global a escala y haciéndoles partícipes del ahorro de costes con precios inferiores.
- Agilidad y elasticidad instantánea: Brinda una infraestructura global y masiva en la cloud el cual permite innovar, experimentar e iterar de manera rápida. Las empresas pueden implementar de manera inmediata nuevas aplicaciones y nuevos servidores en segundos, aumentar su escala al instante cuando crezca su carga de trabajo o reducirla

en función de algunos parámetros, los cuales pueden ser por el número de usuarios conectados, carga de CPU, uso de memoria.

- **Accesibilidad y flexibilidad:** Es una plataforma independiente del lenguaje y del sistema operativo. Solo se determina la plataforma de desarrollo o el modelo de programación que tenga más sentido para el negocio de la empresa.
- **Capacidades híbridas:** La elección para las empresas esta entre una inversión existente en infraestructura y trasladarse a la nube. AWS ayuda a implementar una estrategia híbrida bien diseñada, sin necesidad de nuevas y costosas inversiones en hardware y software bajo demanda, para simplificar las operaciones y alcanzar los objetivos empresariales con mayor facilidad.

1.4. Servicios Web

El autor de [43], explica que un Servicio Web es un componente de software el cual se comunica con varias aplicaciones pero codificando los mensajes en formato XML y luego los envia a través de protocolos estándares de Internet, entre los cuales estan Hypertext Transfer Protocol (HTTP).

En [44], se indica que los sistemas interactúan con el servicios web de una manera prescrita por su descripción usando mensajes SOAP, que son enviados usando HTTP con una serialización XML en relación con otros estándares relacionados con la web. También en [44] se describe que un servicio web se lo puede definir de manera más sencilla como un conjunto de tecnologías estándares de software para el intercambio de datos entre aplicaciones tales como SOAP, WDSL y UDDI.

De igual manera en [44] se expresa que los componentes clave de los servicios web son:

- **Web Services Description Language (WSDL):** WSDL es el archivo basado en XML que describe el servicio web.
- **SOAP:** Es el protocolo basado en XML que utiliza la solicitud de servicio web para invocar el servicio.
- **Protocolo UDDI (Universal Description, Discovery and Integration):** UDDI es el registro que alberga el intermediario del servicio. UDDI es parecido a las páginas amarillas de teléfonos.

1.5. Servicios de Base de datos

El autor en [45] brinda detalles y dice que MySQL es un sistema de administración de base de datos relacional (RDBMS), lo cual significa que se trata de un software capaz de almacenar una enorme cantidad de datos de gran variedad y distribuirlos para cubrir la necesidad de cualquier tipo de empresa, desde pequeñas a grandes empresas y organismos administrativos.

En [46], se indica que MySQL es mayormente utilizado en conjunción con servidores Web donde se lo encuentra relacionado a aplicaciones Web o CMS para sitios online, como Wordpress, Joomla, Drupal, entre otros. Está muy ligado a PHP en lo que se refiere a este tipo de desarrollos.

Es una base de datos que muestra rapidez en lectura de datos, sobre todo cuando se utilizan ciertos motores como lo son MyISAM o InnoDB.

1.5.1. Características de MySQL

En [47] el autor indica que, MySQL es un sistema de base de datos relacional, lo que quiere decir que archiva datos en tablas separadas en lugar de guardar todos los datos en un gran archivo, gracias a esto permite tener mayor velocidad y flexibilidad. Dichas tablas están relacionadas de formas definidas, por lo que es posible combinar distintos datos en varias tablas y conectarlos.

Algunas características de MySQL se detallan [46]:

- Permite escoger múltiples motores de almacenamiento para cada tabla.
- Agrupación de transacciones, pudiendo reunirlos de forma múltiple desde varias conexiones con el fin de incrementar el número de transacciones por segundo.
- Conectividad segura.
- Ejecución de transacciones y uso de claves foráneas.
- Presenta un amplio subconjunto del lenguaje SQL.
- Replicación
- Disponible en casi todas las plataformas o sistemas.
- Búsqueda e indexación de campos de texto.
- Utiliza varias herramientas para portabilidad.

- Tablas hash en memorias temporales
- Uso de tablas en disco b-tree para búsquedas rápidas con compresión de índice.
- Ofrece un sistema de contraseñas y privilegios seguros de verificación basada en el host y tráfico de contraseñas encriptado al conectarse a un servidor.
- Uso de multi-hilos mediante hilos de kernel.
- Soporta gran cantidad de datos, incluso con más de 50 millones de registros.

2 Topología Propuesta

En este capítulo vamos a detallar la topología propuesta, de igual manera los recursos necesarios en hardware y software para la topología propuesta.

2.1. Factores funcionales

2.1.1. Redes Virtuales

El autor en [48] explica que gracias a las redes virtuales es posible la comunicación entre varios computadores, máquinas virtuales, servidores virtuales y cualquier otro dispositivos que este ubicado en diversos centros de datos y oficinas sin importar la distancia o ubicación geográfica. Mientras que las redes físicas hacen uso del cableado y otros elementos de hardware para poder conectar los ordenadores, las redes virtuales amplían estas funciones al hacer eso de la gestión de software para poder conectar dichos servidores y computadores a través de Internet.

En otro contexto en [48] se comenta que las redes virtuales permiten que los dispositivos ubicados en distintos puntos geográficos funcionen con las mismas prestaciones como si estuviesen conectados a través de redes físicas convencionales. Gracias a esto los centros de datos abarcan diferentes ubicaciones físicas y permite brindar opciones más eficientes a los administradores de red, entre las cuales están, la capacidad de modificar la red de manera sencilla según las necesidades, y sin tener que adquirir más hardware o renovarlo, una mayor flexibilidad en el aprovisionamiento de la red conforme a necesidades y aplicaciones específicas.

2.2. Topología

En este apartado se va a detallar la topología propuesta para poder desplegar el esquema de nube híbrida para entornos de desarrollo usando la replicación de la base en el caso de base de datos, en las plataformas OpenNebula y AWS.

Para poder desplegar esta nube híbrida se utilizara tres herramientas como infraestructura, tal y como se indica en la figura **2-1**, una de ellas es la plataforma de OpenNebula, la otra

es la plataforma de Amazon (AWS) y por último la plataforma de Azure, así mismo dentro de estas plataformas se instalará y configurará la base de datos Postgres y un Directorio Activo.

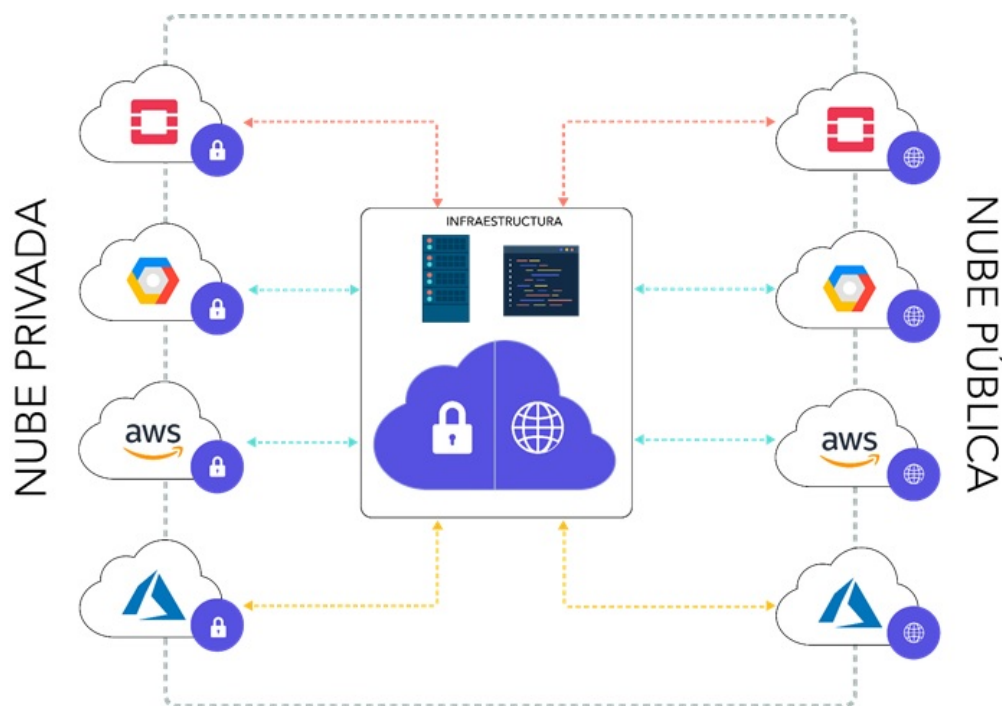


Figura 2-1: Topología Propuesta. Fuente Autor

En esta topología se demostrará cómo interactúan las infraestructuras y la posterior integración de la nube privada con la nube pública mediante previa instalación de gemas desde la parte de AWS, para la plataforma de OpenNebula se ha desplegado una máquina virtual con Ubuntu, la cual servirá para la administración, la parte de cómputo, la red, y el almacenamiento, dentro de OpenNebula, esta instancia estará corriendo una imagen de Ubuntu donde se configurará y estará corriendo el servidor de base de datos, el cual se configurará para su posterior replicación, así mismo en la plataforma de AWS, se creará una instancia, dentro del menú principal de la consola AWS, donde permitirá iniciar una nueva instancia con la guía rápida de AWS, esta guía permite crear una instancia de manera sencilla y en un tiempo corto, el tiempo de inicio dependerá del tipo de máquina creada, para esta topología propuesta, se hará uso del tipo t2.micro con una imagen de Ubuntu 16.04 que se encuentra disponible en Amazon Machine (AMI).

A continuación en la figura 2-2, se muestra la topología diseñada en ambas nubes, para esto utilizamos una red interna y una red externa, donde en la red externa nos permitirá tener una conexión a la red local física, así mismo, la red interna mediante un router nos permitirá conectarnos hacia la red externa creada en OpenNebula y a partir de esta tener una salida

a internet, pero para comunicarnos desde y hacia afuera de una instancia de OpenNebula necesitamos IPs publicas exclusivamente para las instancias las mismas en las que estarán desplegados los servidores.

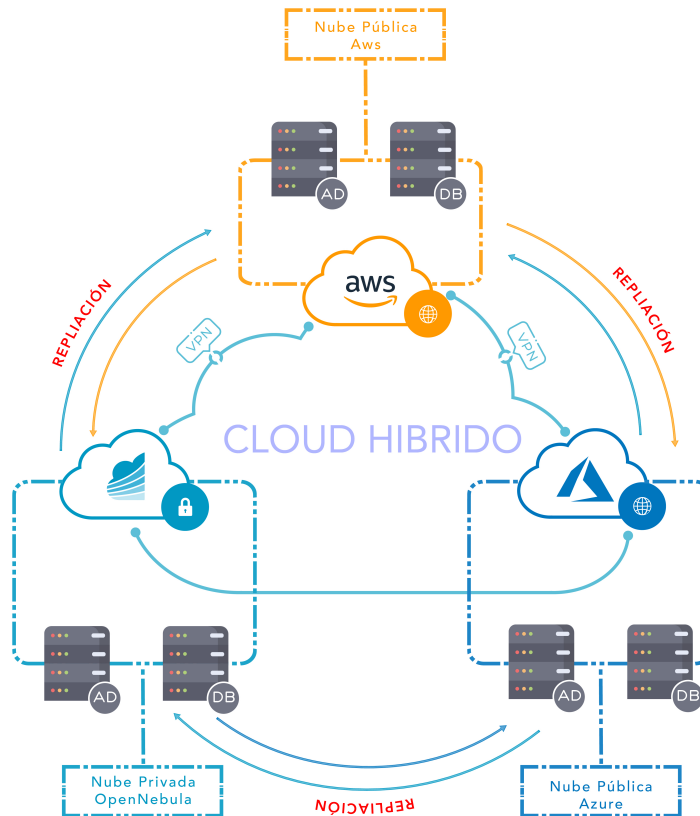


Figura 2-2: Topología Implementada. Fuente Autor

A partir de la topología presentada, se detallará los recursos necesarios tanto en hardware y software que se utilizará para la implementación de la arquitectura antes mencionada.

2.2.1. Recursos de software y hardware utilizados

En la tabla 2-1 se detalla los recursos utilizados.

Marca	Modelo	RAM	Disco Duro	Procesador
MSI	Leopard	16 Gb	1TB	i7
Apple	Macbook Pro	16 Gb	500GB	i7

Tabla 2-1: Recursos Utilizados en Hardware y Software. Fuente Autor

2.2.2. Recursos de hardware utilizados para el despliegue de OpenNebula

En la tabla 2-2 se detalla los recursos utilizados para el despliegue de OpenNebula.

Sistema Operativo	RAM	Procesador	Disco Duro	Detalle
Centos 7	2GB	4 Cores	60 GB	VM Front
Centos 7	6GB	4 Cores	90 GB	VM Nodo

Tabla 2-2: Recursos Utilizados en Hardware. Fuente Autor

2.2.3. Recursos utilizados para la nube pública en AWS

En la tabla 2-3 se detalla los recursos utilizados para el despliegue de AWS.

Instancia	vCPU	Memoria	Storage	Network Performance
t2.micro	1	1 GB	EBS Only	Low to moderate
t2.micro	1	1 GB	EBS Only	Low to moderate

Tabla 2-3: Recursos Utilizados en Hardware. Fuente Autor

2.2.4. Recursos utilizados para la nube pública en Azure

En la tabla **2-4** se detalla los recursos utilizados para el despliegue de Azure.

Instancia	vCPU	Memoria	Almacenamiento Temporal
DS1	1	3.5 GB	7 GB
DS!	1	3.5 GB	7 GB

Tabla 2-4: Recursos Utilizados en Hardware. Fuente Autor

3 Metodología.

3.1. Metodología y configuración de las plataformas

La metodología empleada es la metodología de enrutamiento VPN sitio a sitio ya que esta permite conectar dos o mas redes diferentes usando un túnel OpenVPN. En esta conexión los dispositivos de una red pueden llegar a los dispositivos de la otra red y viceversa.

Para poder implementar y así realizar la replicación de los datos de manera exitosa desde la nube privada OpenNebula hacia las nubes públicas de AWS y Azure con éxito, se utilizó una red privada virtual (VPN), la cuál proporciona una conexión cifrada de forma segura a nuestra red a través de la internet, la misma que incorpora partes importantes de seguridad en capas, y que es esencial para la protección de los datos. El uso de una VPN brinda la posibilidad de acceder a importantes recursos de la red y también poder conectar varios servidores.

Después de implementar la infraestructura se debe configurar cada uno de los servicios que se desea replicar, para el caso se uso una base de datos y un directorio activo, los cuales están detallados en el capítulo 6 del presente trabajo.

3.2. Descripción general de la metodología de enrutamiento VPN sitio a sitio

En el sitio oficial de OpenVpn [1], se describe un ejemplo de como es la configuración correcta de una Vpn sitio a sitio, la figura **3-1**, indica la sede de una empresa la cual está a la derecha, donde existen computadoras y servidores los cuales están situados en la plataforma de AWS y Azure. Uno de los servidores tiene instalado el producto OpenVPN Access Server, además de una base de datos y un Directorio Activo. Todas las computadoras y servidores de esa red están conectados a un enrutador (la pequeña caja cuadrada plana con un ícono de engranaje) la cual también brinda acceso a Internet. A la izquierda se encuentra la plataforma de OpenNebula donde hay una oficina que también posee servidores, al igual que una base de datos y un Directorio Activo, todos ellos conectados a su propio enrutador que también brinda acceso a Internet.

Los servidores en las oficinas tienen un programa cliente OpenVPN instalado en un sistema operativo Linux y Windows Server, que tiene una conexión de túnel OpenVPN activa al servidor de acceso OpenVPN en la sede (AWS), como se indica en la figura 3-1 . En este ejemplo de configuración de sitio a sitio, se ha abierto un acceso completo entre computadoras y servidores en la sede y las oficinas cercanas. En otras palabras, un usuario sentado en una computadora en la oficina subsidiaria puede acceder a los servidores en la sede como si estuviera allí, gracias a una conexión de túnel OpenVPN entre las dos redes. Este túnel OpenVPN viaja a través de Internet y su contenido está encriptado de forma segura.

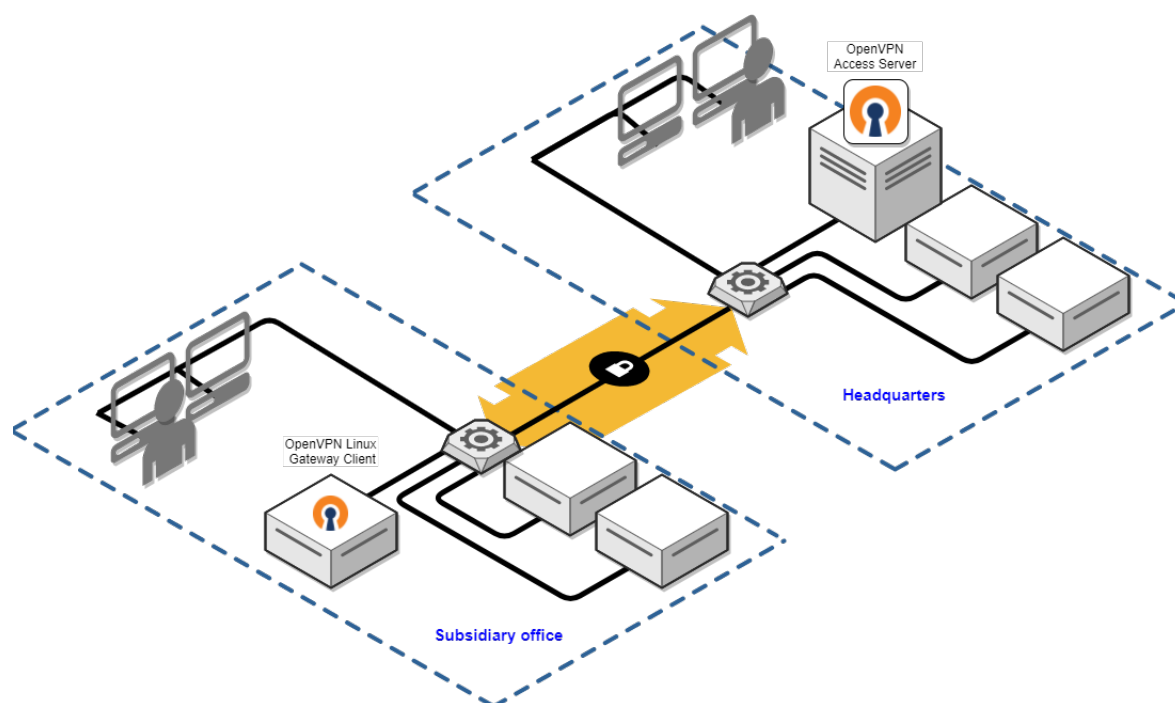


Figura 3-1: Diagrama de VPN sitio a sitio. Fuente [1]

3.3. Flujo de tráfico

Se ha creado una serie de figuras que indican paso a paso cómo una solicitud de una computadora cliente en la oficina cercana llega a un servidor de aplicaciones en la oficina central y cómo se envía una respuesta. Cada paso del proceso se muestra de manera clara con información de red relevante. cabe recalcar que este tipo de configuración permite que otros clientes VPN puedan iniciar sesión en el servidor de acceso OpenVPN y obtengan acceso a cualquiera de los dispositivos en estas 3 redes. Además, una configuración de sitio a sitio no necesita limitarse a una red cercana, puede ser múltiple con la misma facilidad.

En la figura 3-2, el usuario de la izquierda (Azul) quiere contactarse con el servidor de la derecha (Purpura), la solicitud va de su computadora a la red y al enrutador local (Verde).

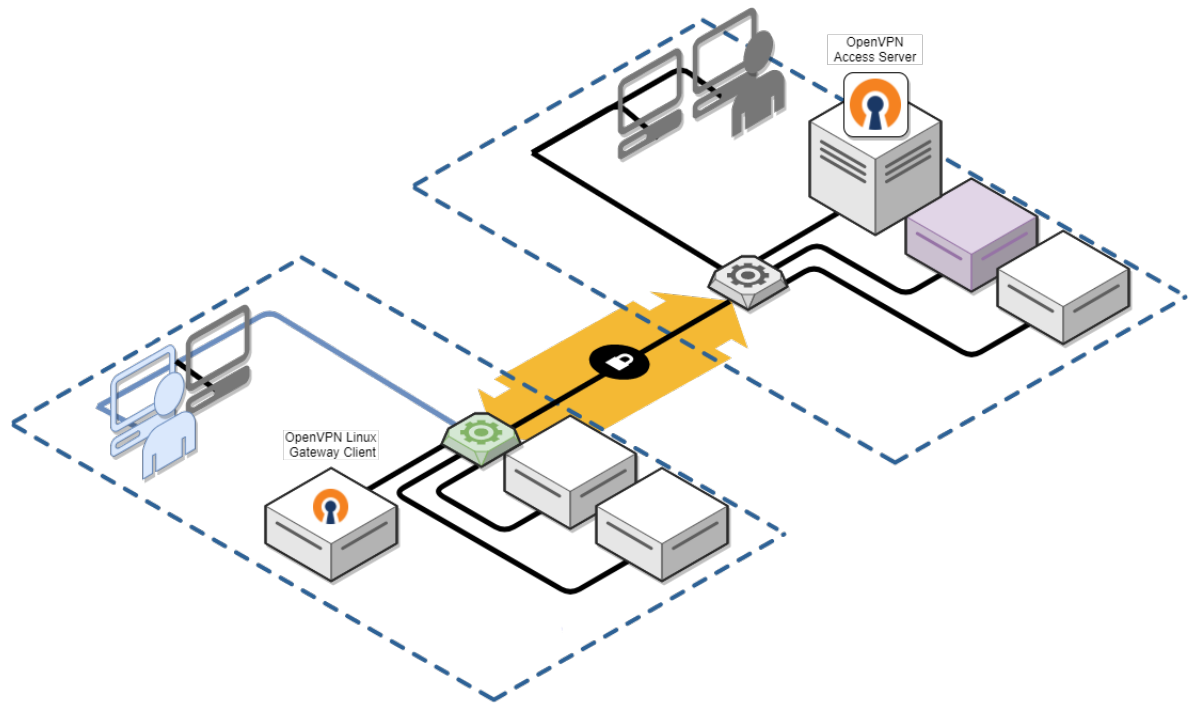


Figura 3-2: Diagrama de VPN sitio a sitio. Fuente [1]

En la figura **3-3** la tabla de enrutamiento en el enrutador local (verde) redirige la solicitud al cliente OpenVpn (naranja) que funciona como una puerta de entrada a la sede

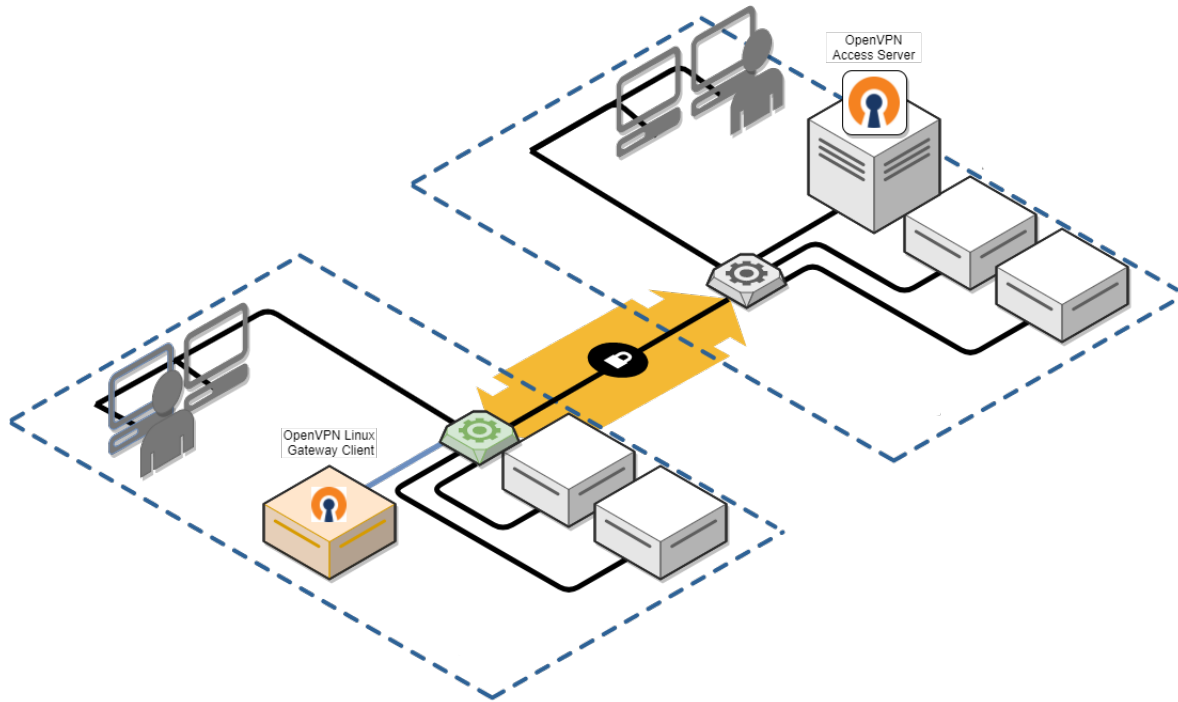


Figura 3-3: Diagrama de VPN sitio a sitio [1]

El cliente OpenVpn que se encuentra a la izquierda (naranja) envía la solicitud a través de un tunel OpenVpn cifrado al servidor de acceso OpenVpn de la derecha, tal y como se muestra en la figura 3-4.

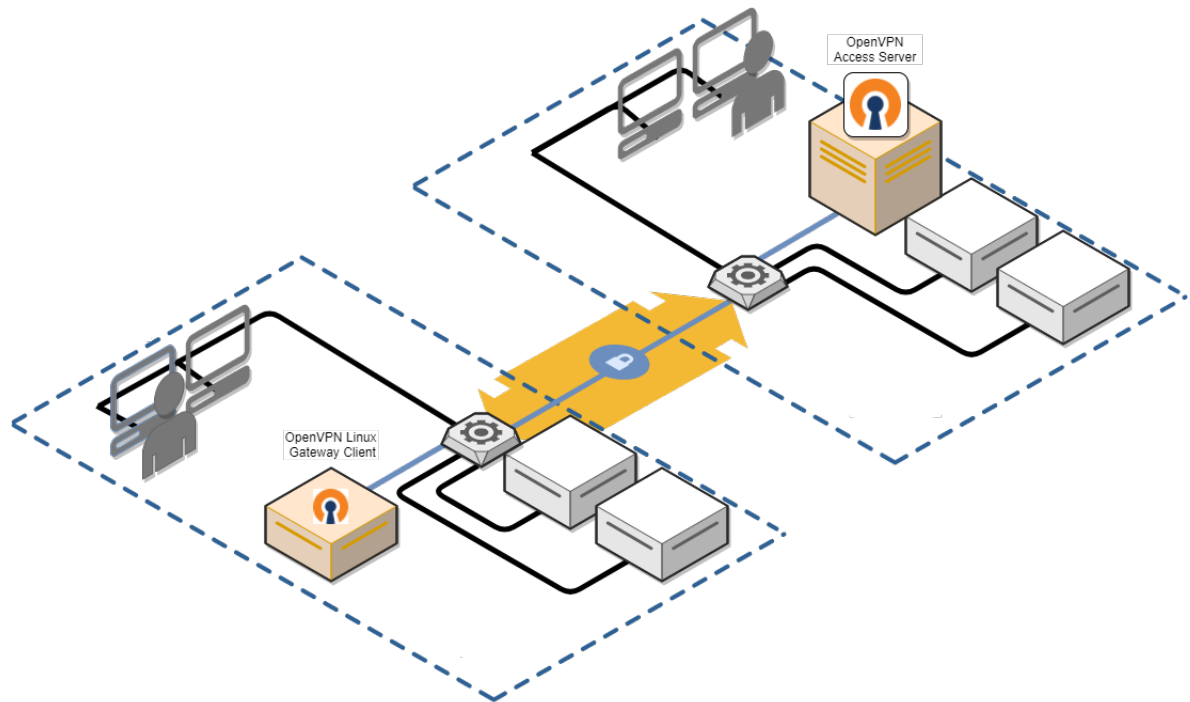


Figura 3-4: Diagrama de VPN sitio a sitio. Fuente [1]

El servidor de acceso OpenVpn descifra el tráfico de la vpn y lo solicita en red, para volver a acceder al servidor de destino a través de mi red como lo indica la figura **3-5**.

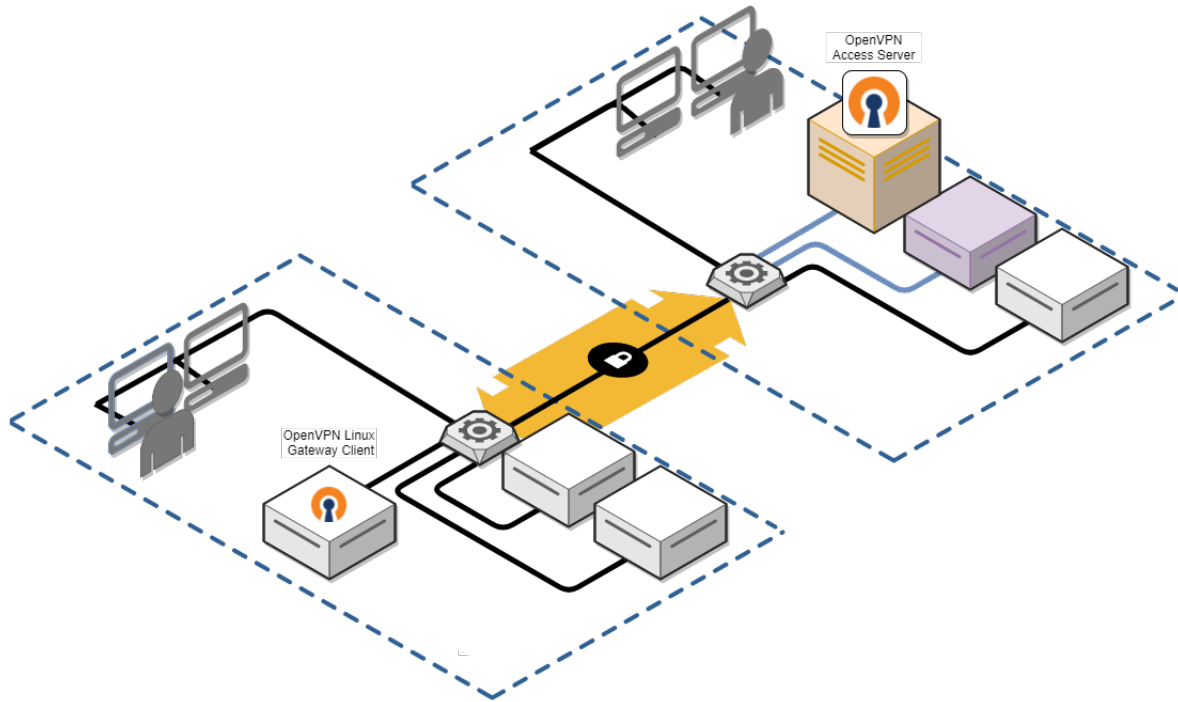


Figura 3-5: Diagrama de VPN sitio a sitio. Fuente [1]

En la figura **3-6** el servidor de la derecha quiere enviar la respuesta al cliente, entonces envía la respuesta a la red para así llegar al enrutador local.

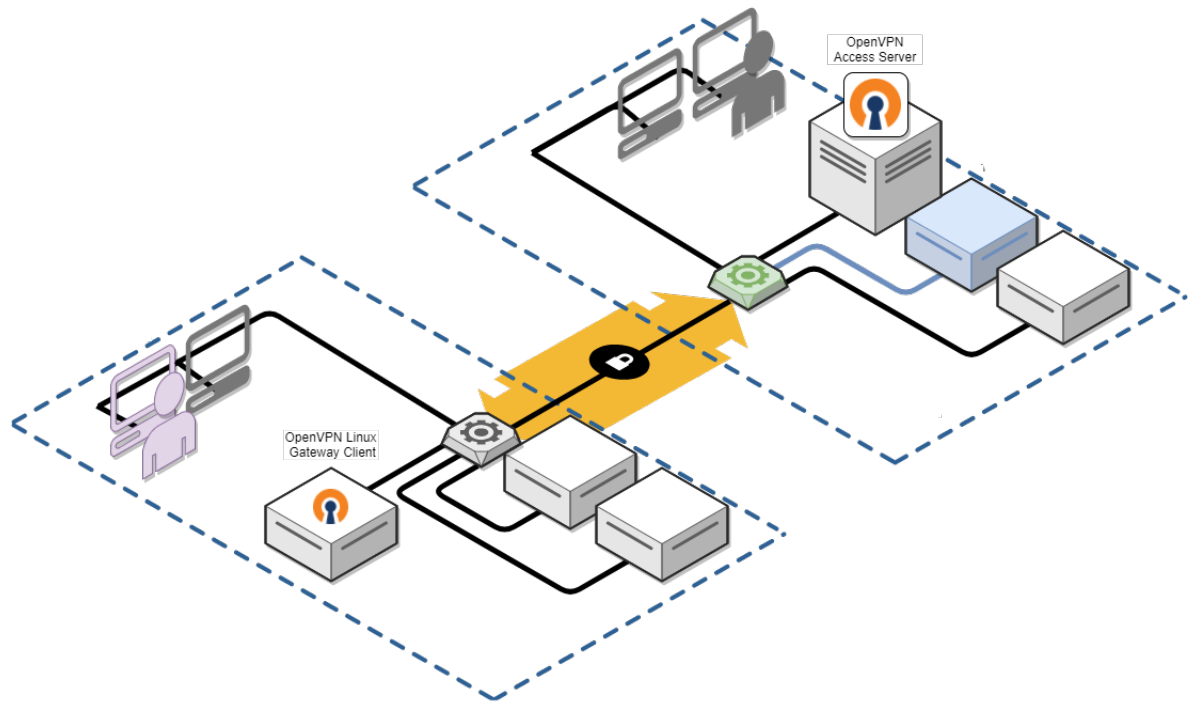


Figura 3-6: Diagrama de VPN sitio a sitio. Fuente [1]

La tabla de enrutamiento del enrutador en color (verde) redirige las respuestas al servidor de acceso OpenVpn que funciona como la puerta de enlace a la oficina subsidiaria tal y como se muestra en la figura 3-7.

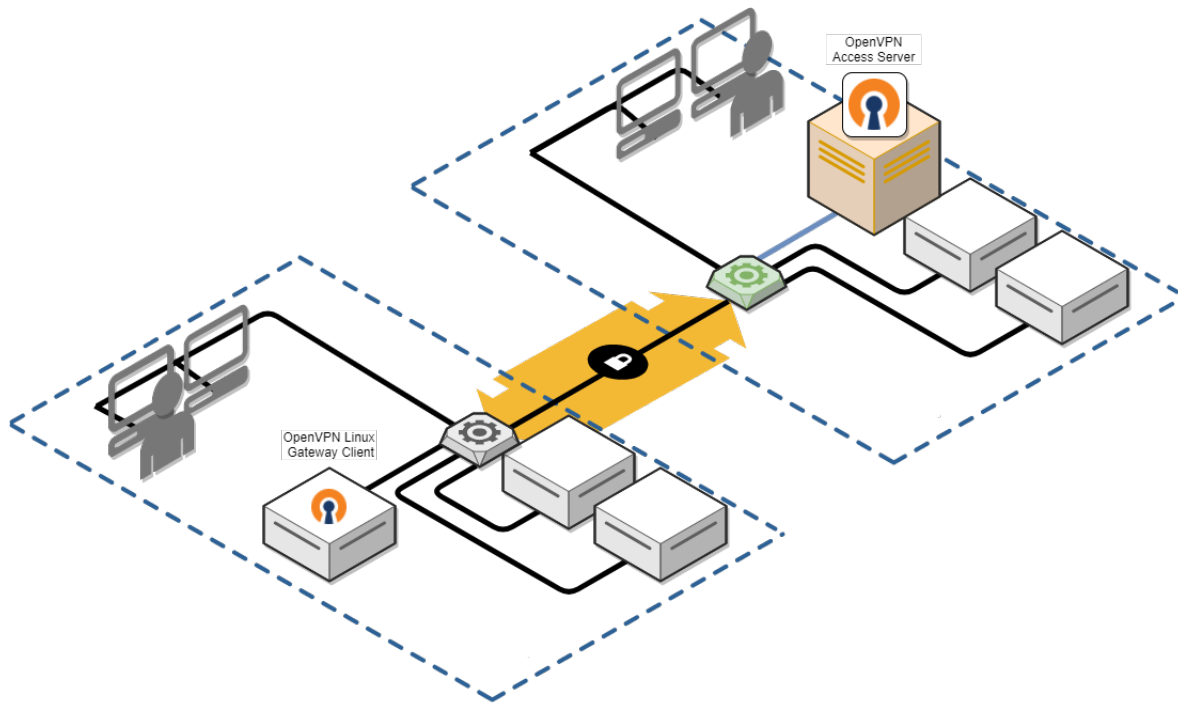


Figura 3-7: Diagrama de VPN sitio a sitio. Fuente [1]

En la figura **3-8** el servidor de acceso OpenVpn de la derecha envia la respuesta a traves de un tunel vpn encriptado al cliente OpenVpn de la izquierda.

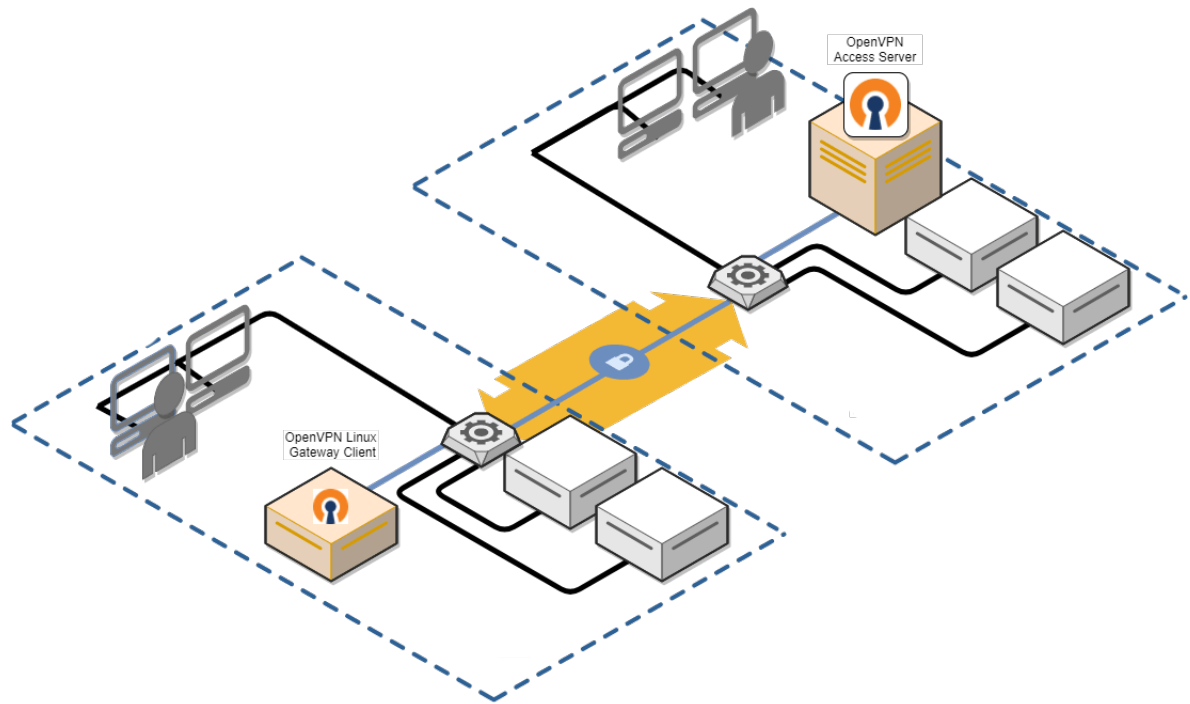


Figura 3-8: Diagrama de VPN sitio a sitio. Fuente [1]

El cliente de la puerta de enlace OpenVpn descifra el tráfico vpn y envía la respuesta a la red llegando así al objetivo a través de la red, como indica la figura 3-9.

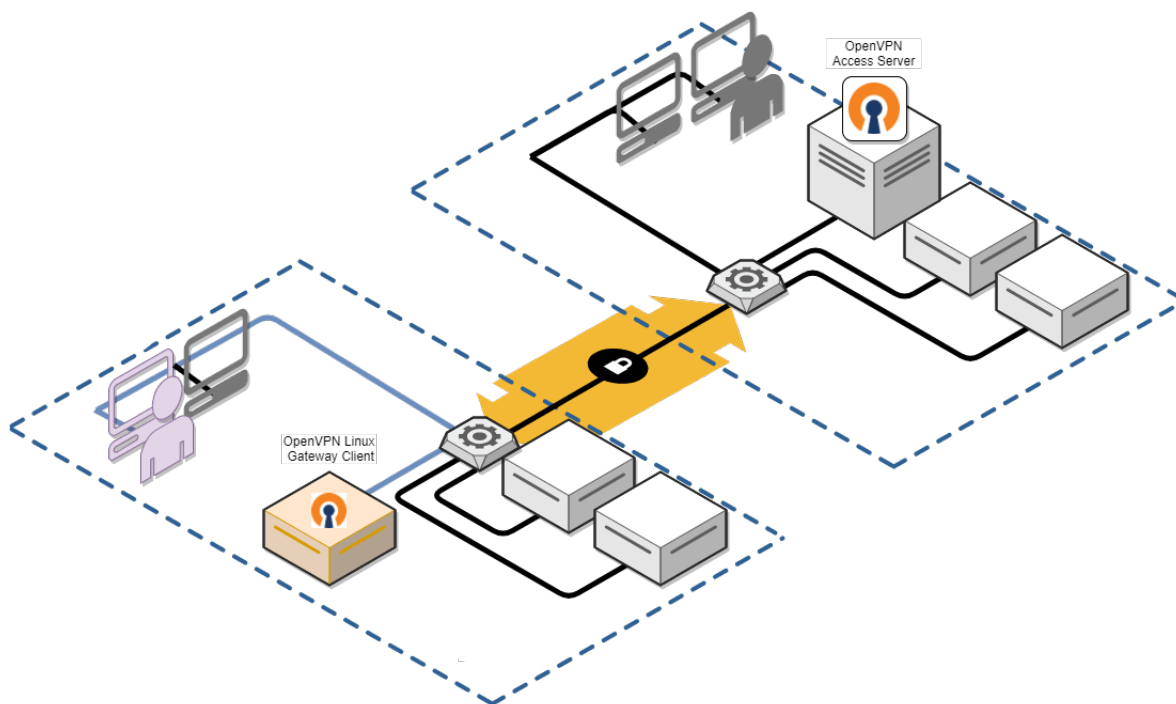


Figura 3-9: Diagrama de VPN sitio a sitio. Fuente [1]

Estas plataformas de nubes permiten administrar y otorgar más o menos recursos informáticos tal como lo solicite el cliente y de tal manera que la pérdida o fallo de cualquier sistema alojado en estas plataformas no conlleve a fallos en el sistema.

En [33] se indica que en OpenNebula se proporciona varias interfaces que se las puede utilizar para poder interactuar con su funcionalidad y de esta manera poder gestionar los recursos tanto físicos como virtuales .

Para la replicación de la Base de datos tiene que existir la conectividad entre las tres instancias y la comunicación debe ser bidireccional entre las tres instancias que tienen instalado Postgres

- Las bases de datos deben ser de la misma versión en las 3 plataformas.
- Las bases de datos AWS y Azure deben de apuntar hacia la direccion asignada por OpenVPN en la maquina de OpenNebula.

- La base de datos maestra de OpenNebula debe permitir la replicación al usuario "replicado" desde la dirección IP que le asigna OpenVPN de la instancia en AWS.
- La base de datos maestra de OpenNebula debe permitir la replicación al usuario "replicadorAzure" desde la dirección IP que le asignada OpenVPN de la instancia en Azure.
- La función del "replicador" debe colocarse solo en la base de datos que hará la vez de principal.

Igualmente para la replicación del Directorio Activo tiene que existir la conectividad entre las tres instancias y la comunicación debe ser bidireccional entre las tres instancias de Windows Server 2012 R2

- En la configuración de red en la interfaz local y la interfaz que crea OpenVPN se debe asignar la dirección IP del Servidor DNS que es la instancia de AWS, este proceso se realiza en las 3 instancias.
- Se debe promover las dos instancias de OpenNebula y Azure con el domino DMR.COM

Todas las configuraciones realizadas en la infraestructura del presente trabajo están hechas para poder cumplir con todos los requisitos mencionados anteriormente.

4 Análisis y Presentación de Resultados

En base a las pruebas realizadas y que se presentarán en este capítulo, se analizarán las pruebas que se han realizado en los servicios escogidos a replicarse para el presente trabajo, basandose en el desempeño general de la infraestructura y topología implementada, así como las condiciones que deben cumplirse para que se pueda realizar la migración y posterior replicación, no solo de la base de datos sino del Directorio Activo empleado.

4.1. Directorio Activo

- Como primer paso hay que comprobar que los 2 servidores esten activos antes de realizar cualquier proceso, tal y como se muestra en la figura 4-1.

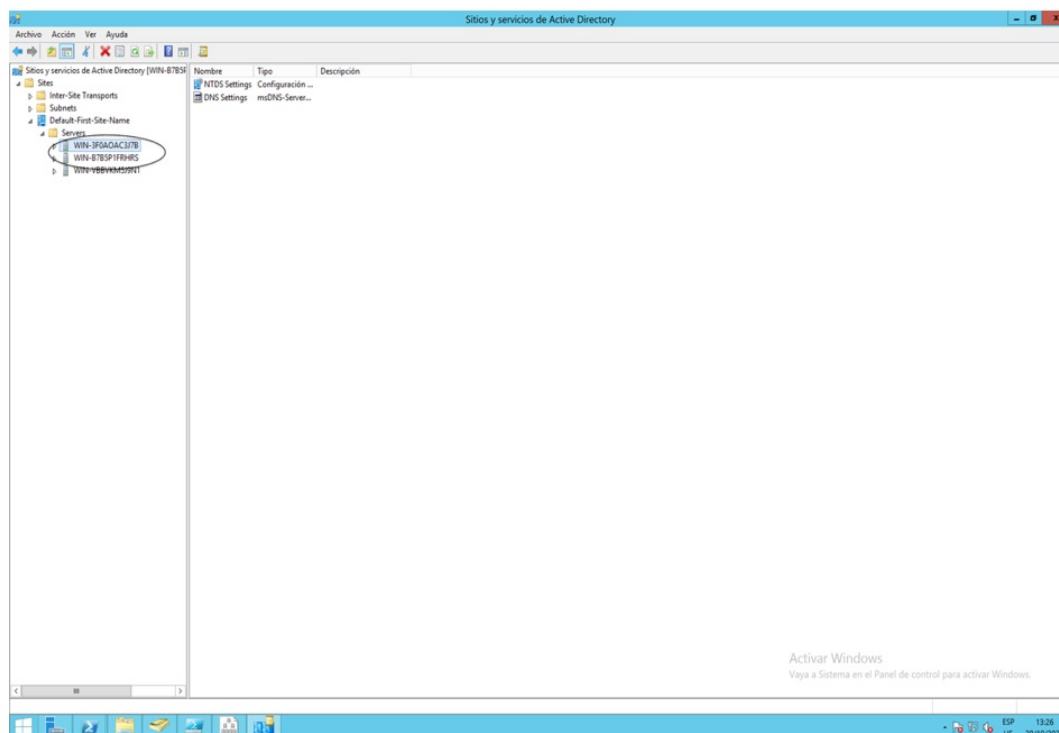


Figura 4-1: Se comprueba que el servicio este activo. Fuente Autor

Luego se crea la Unidad Organizativa UPS $\rightarrow Users \rightarrow IT$:

En la figura 4-2 se indica en donde se crearán los usuarios que se exportaran por un archivo con extensión csv con la información de los usuarios y la Unidad Organizativa perteneciente.

FirstName	Initials	Lastname	Username	Email	StreetAddress	City	ZipCode	State	Country	Department	Password	Telephone	JobTitle	Company	OU
Max	MF	Fraser	Max.Fraser	Max.Fraser@21 Baker St	London	NW1 6XE			United Kingdom	IT	Qw7_JTc	4,4123E+10	Engineer	DMR	OU=IT,OU=Users,OU=UPS,DC=DMR,DC=COM
Piers	PB	Bower	Piers.Bower	Piers.Bower@21 Baker St	Quito	NW1 6XE			Ecuador	IT	RW-cn3N	4,4123E+10	Manager	DMR	OU=IT,OU=Users,OU=UPS,DC=DMR,DC=COM
Kylie	KD	Davidson	Kylie.Davidson	Kylie.Davidson@21 Baker St	New York	NW1 6XE			EEUU	IT	3V6z2.Wm	4,4123E+10	Engineer	DMR	OU=IT,OU=Users,OU=UPS,DC=DMR,DC=COM
Richard	RG	Grant	Richard.Grant	Richard.Grant@21 Baker St	Mexico DC	NW1 6XE			Mexico	IT	JN3ZYh5	4,4123E+10	Teamleader	DMR	OU=IT,OU=Users,OU=UPS,DC=DMR,DC=COM
Boris	BC	Campbell	Boris.Campbell	Boris.Campbell@21 Baker St	Ottawa	NW1 6XE			Canada	IT	9ZesQjpq	4,4123E+10	Engineer	DMR	OU=IT,OU=Users,OU=UPS,DC=DMR,DC=COM
Nicholas	NM	Murray	Nicholas.Murray	Nicholas.Murray@21 Baker St	Roma	NW1 6XE			Italia	IT	KK*rB72p	4,4123E+10	Manager	DMR	OU=IT,OU=Users,OU=UPS,DC=DMR,DC=COM
Leonard	LC	Clark	Leonard.Clark	Leonard.Clark@21 Baker St	Medellin	NW1 6XE			Colombia	IT	Aj+Jc35	4,4123E+10	Engineer	DMR	OU=IT,OU=Users,OU=UPS,DC=DMR,DC=COM
Ruth	RD	Dickens	Ruth.Dickens	Ruth.Dickens@21 Baker St	Lima	NW1 6XE			Peru	IT	Jgv4lB65	4,4123E+10	Engineer	DMR	OU=IT,OU=Users,OU=UPS,DC=DMR,DC=COM
Jonathan	JF	Fisher	Jonathan.Fisher	Jonathan.Fisher@21 Baker St	San Salvador	NW1 6XE			Salvador	IT	u*PQIAx5	4,4123E+10	Engineer	DMR	OU=IT,OU=Users,OU=UPS,DC=DMR,DC=COM
Grace	GR	Rees	Grace.Rees	Grace.Rees@21 Baker St	Moscu	NW1 6XE			Rusia	IT	wl[6p&Rt	4,4123E+10	Engineer	DMR	OU=IT,OU=Users,OU=UPS,DC=DMR,DC=COM

Figura 4-2: Lista de usuarios en archivo csv. Fuente Autor

- Se verifica que el archivo csv este correcto, ejecutandolo en la PowerShell con el siguiente comando:

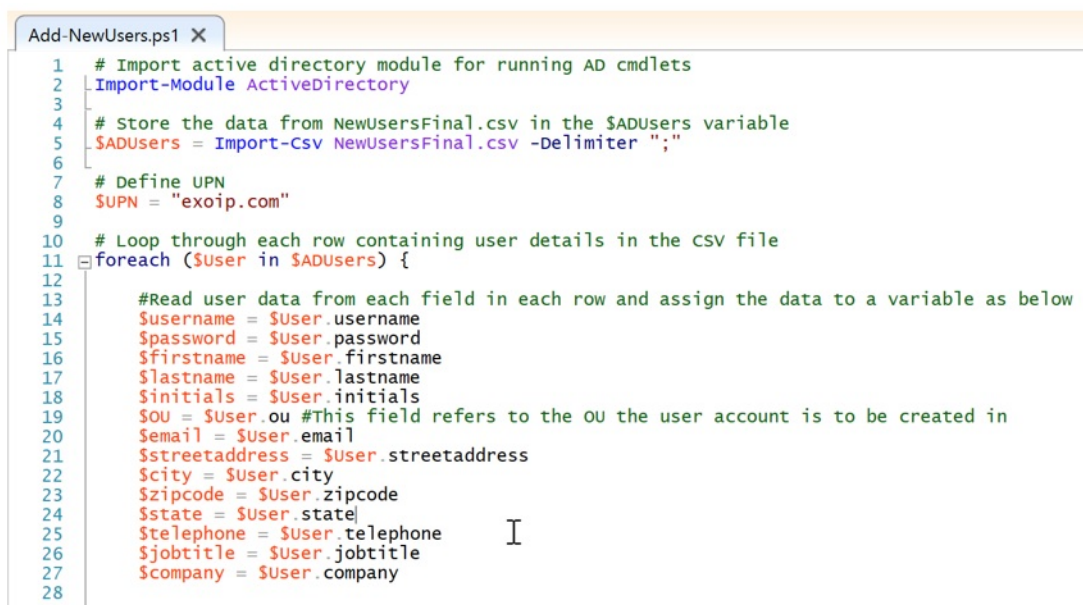
Import-Csv NewUsersFinal.csv -Delimiter ";" — Format-Table

La salida del commando debe ser muy parecida ha esta y en forma de table, como lo indica la figura 4-3.

FirstName	Initials	Lastname	Username	Email	StreetAddress	City	ZipCode	State	Country
Max	MF	Fraser	Max.Fraser	Max.Fraser...	21 Baker St	London	Nw1 6XE		United Kin...
Piers	PB	Bower	Piers.Bower	Piers.Bowe...	21 Baker St	Quito	Nw1 6XE		Ecuador
Kylie	KD	Davidson	Kylie.Davi...	Kylie.Davi...	21 Baker St	New York	Nw1 6XE		EEUU
Richard	RG	Grant	Richard.Gr...	Richard.Gr...	21 Baker St	Mexico DC	Nw1 6XE		Mexico
Boris	BC	Campbell	Boris.Camp...	Boris.Camp...	21 Baker St	Ottawa	Nw1 6XE		Canada
Nicholas	NM	Murray	Nicholas.M...	Nicholas.M...	21 Baker St	Roma	Nw1 6XE		Italia
Leonard	LC	Clark	Leonard.Cl...	Leonard.Cl...	21 Baker St	Medellin	Nw1 6XE		Colombia
Ruth	RD	Dickens	Ruth.Dicke...	Ruth.Dicke...	21 Baker St	Lima	Nw1 6XE		Peru
Jonathan	JF	Fisher	Jonathan.F...	Jonathan.F...	21 Baker St	San Salvador	Nw1 6XE		Salvador
Grace	GR	Rees	Grace.Rees...	Grace.Rees...	21 Baker St	Moscu	Nw1 6XE		Rusia

Figura 4-3: Salida que se muestra al ejecutar el comando. Fuente Autor

En la figura 4-4 se indica que para importar el archivo csv con los datos de usuarios se usará el siguiente script:



```
1 # Import active directory module for running AD cmdlets
2 Import-Module ActiveDirectory
3
4 # Store the data from NewUsersFinal.csv in the $ADUsers variable
5 $ADUsers = Import-Csv NewUsersFinal.csv -Delimiter ";"
6
7 # Define UPN
8 $UPN = "exoip.com"
9
10 # Loop through each row containing user details in the csv file
11 foreach ($User in $ADUsers) {
12     #Read user data from each field in each row and assign the data to a variable as below
13     $username = $User.username
14     $password = $User.password
15     $firstname = $User.firstname
16     $lastname = $User.lastname
17     $initials = $User.initials
18     $OU = $User.ou #This field refers to the OU the user account is to be created in
19     $email = $User.email
20     $streetaddress = $User.streetaddress
21     $city = $User.city
22     $zipcode = $User.zipcode
23     $state = $User.state
24     $telephone = $User.telephone
25     $jobtitle = $User.jobtitle
26     $company = $User.company
27 }
28
```

Figura 4-4: Script usado para la importación de archivos. Fuente Autor

Al ejecutar el script deberá aparecer algo similar como indica la figura 4-5:



```
The user account Max.Fraser is created.
The user account Piers.Bower is created.
The user account Kylie.Davidson is created.
The user account Richard.Grant is created.
The user account Boris.Campbell is created.
The user account Nicholas.Murray is created.
The user account Leonard.Clark is created.
The user account Ruth.Dickens is created.
The user account Jonathan.Fisher is created.
```

Figura 4-5: Salida al ejecutar el script. Fuente Autor

En la figura 4-6 y 4-7 se verifica en Usuarios y Equipos del Active Directory que se hayan creado correctamente los usuarios tanto en la nube privada como en la pública.

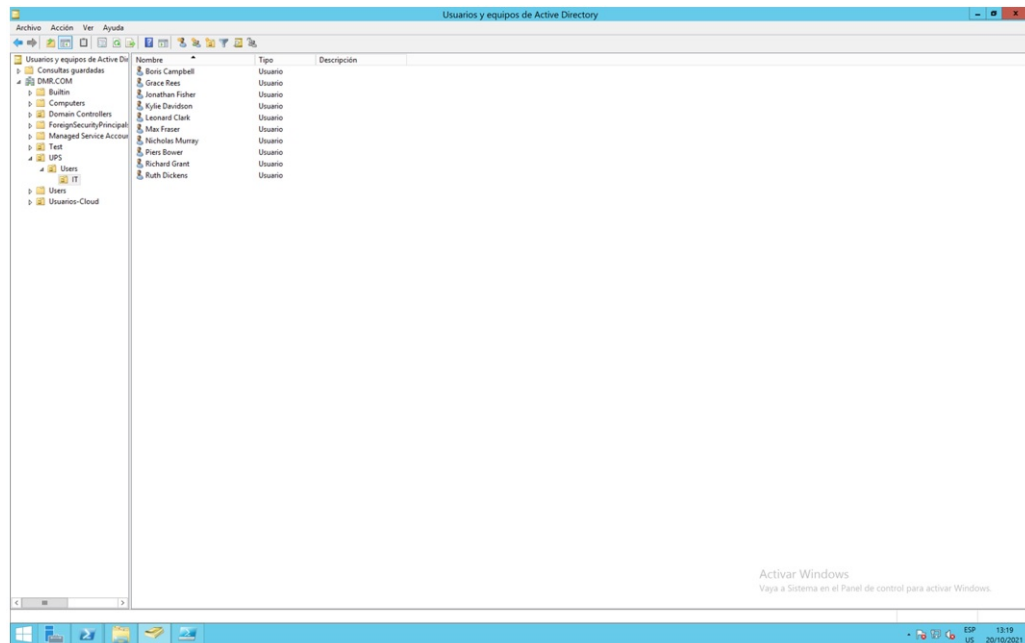


Figura 4-6: Usuarios en la nube privada. Fuente Autor

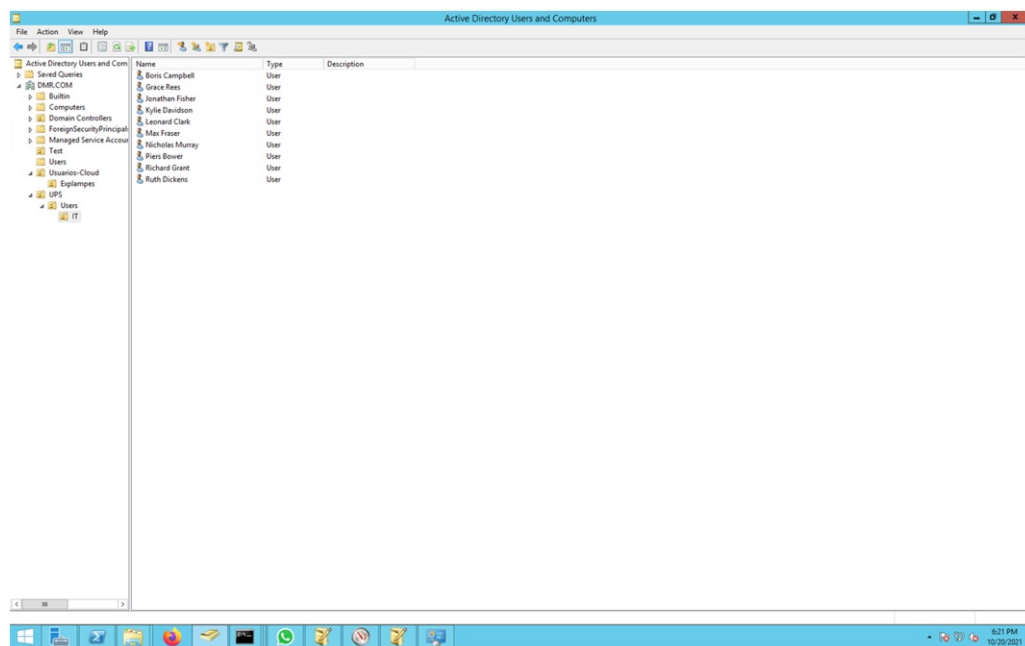


Figura 4-7: Usuarios en la nube pública. Fuente Autor

Para comprobar que no hay problemas con la replicación del Active Directory se instalará una herramienta llamada ADREPLSTATUS, que analizará el estado de replicación de los controladores de dominio de Active Directory. Se lo puede descargar desde aquí: <https://www.microsoft.com/en-us/download/details.aspx?id=30005>

Se procede a ejecutar la herramienta que se indica en la figura 4-8 y se realizará un análisis en el cual mostrará si existe algún error, para este caso se puede ver que las operaciones se completaron de manera correcta desde el Origen(Servidor en la Nube Privada) hasta el Destino(Servidor en la Nube Pública) y no existe ningún error al replicar.

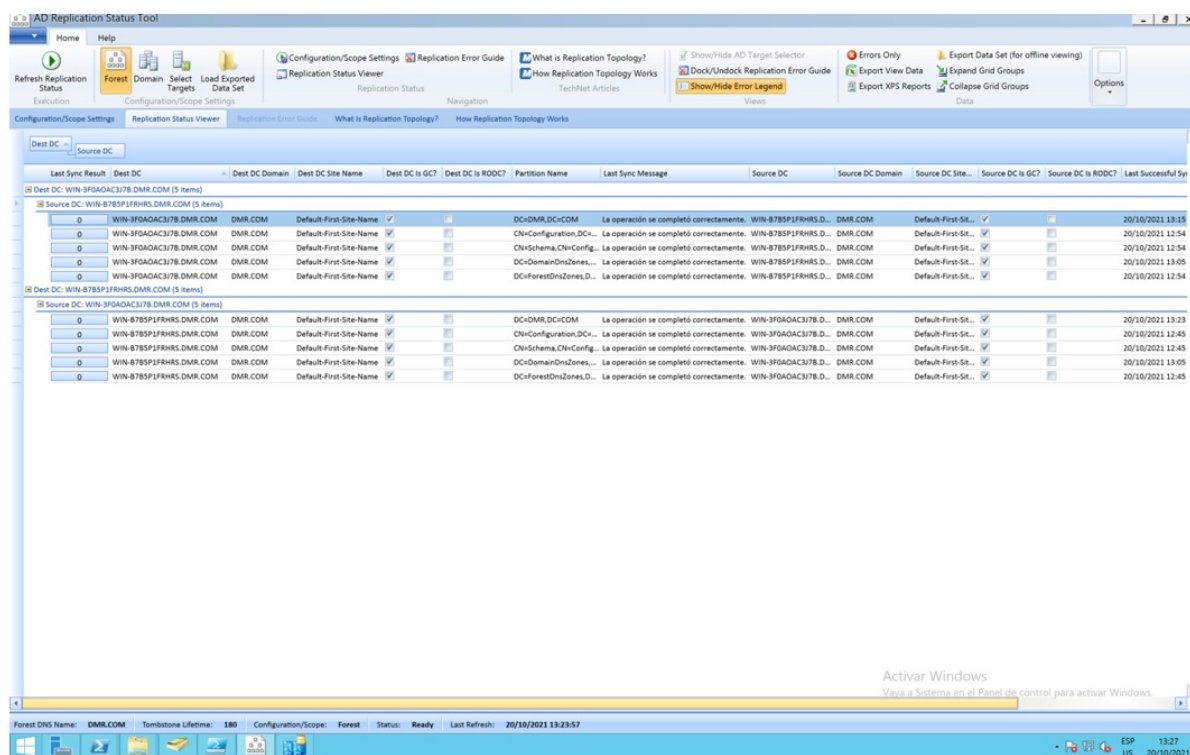


Figura 4-8: Analizando el estado de la replicación. Fuente Autor

4.2. Base de datos - Postgres

Se comienza generando los hilos donde se van a crear los procesos y las posteriores pruebas, en OpenNebula, AWS como se muestra en la figura 4-9 y 4-10.

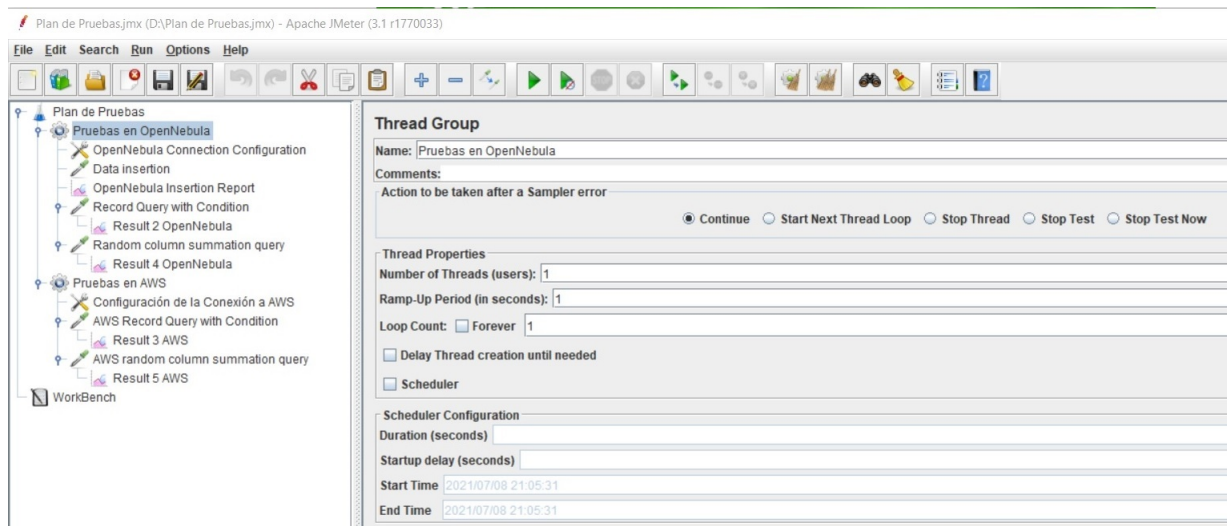


Figura 4-9: Inicio del proceso en OpenNebula. Fuente Autor

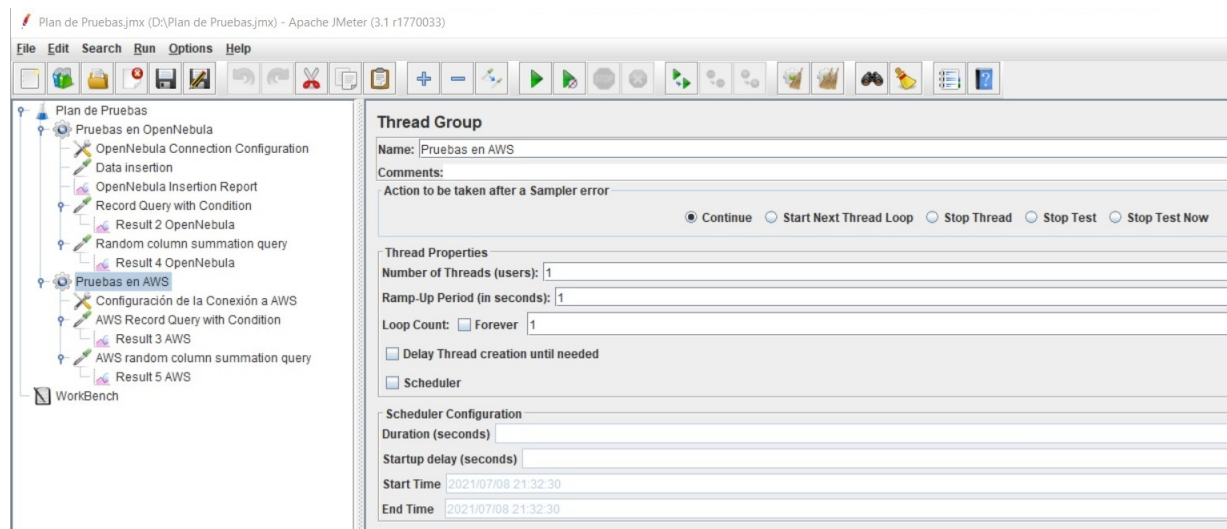


Figura 4-10: Inicio del proceso en AWS. Fuente Autor

En la figura 4-11 y 4-12 se procede a configurar las respectivas conexiones a las bases alojadas en OpenNebula y en AWS con la ip pública

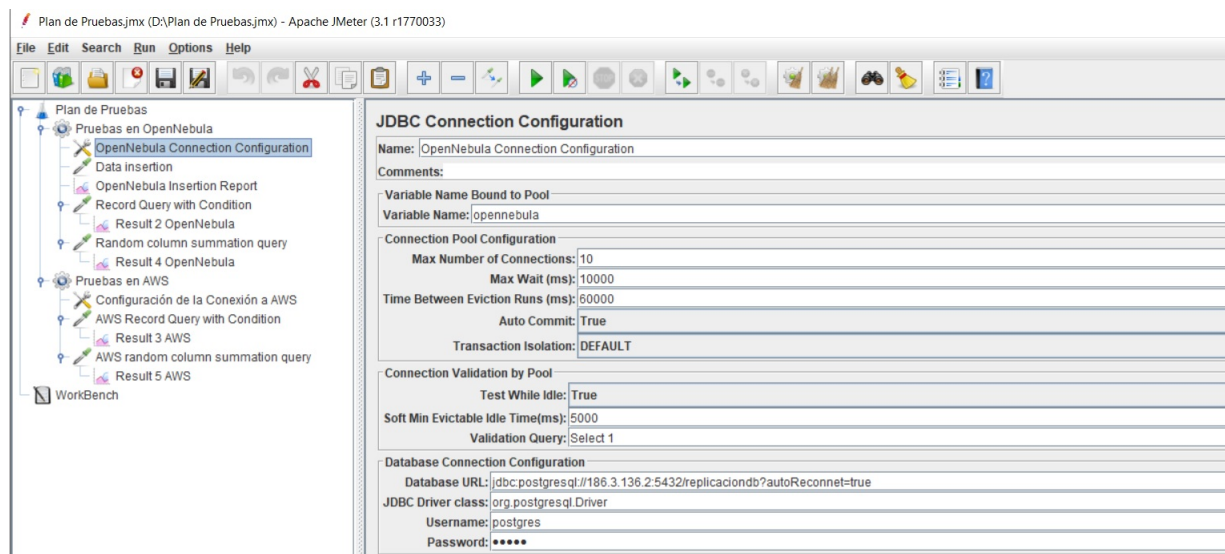


Figura 4-11: Conexión a la base de OpenNebula. Fuente Autor

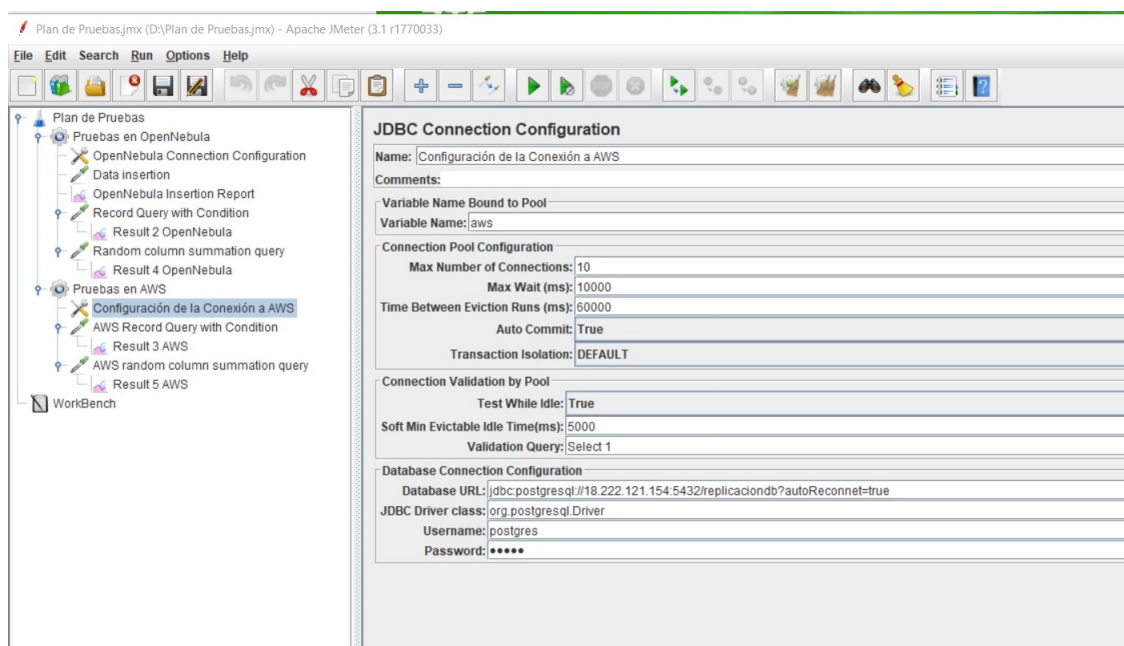


Figura 4-12: Conexión a la base de AWS. Fuente Autor

Después se procede con la inserción de los datos en la base de OpenNebula, figura 4-13

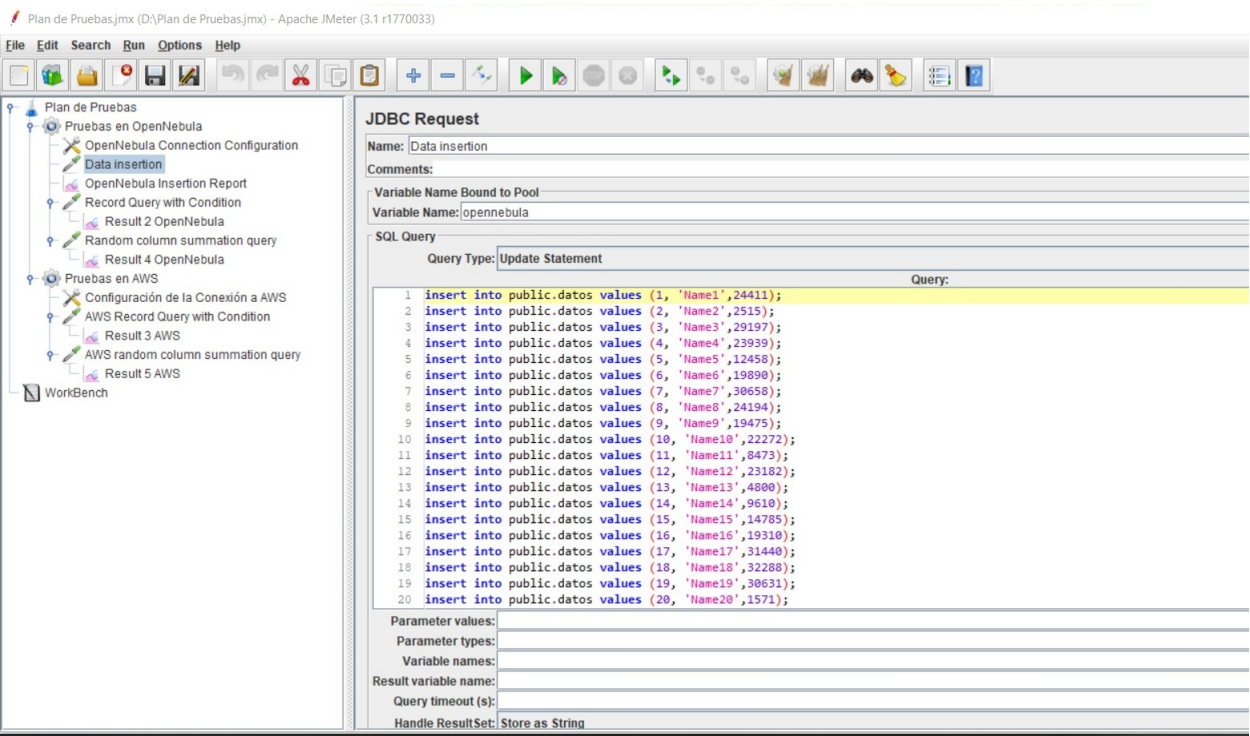


Figura 4-13: Inserción de datos en la base de OpenNebula. Fuente Autor

En la figura 4-14 se procede a verificar que los datos se hayan replicado de manera correcta

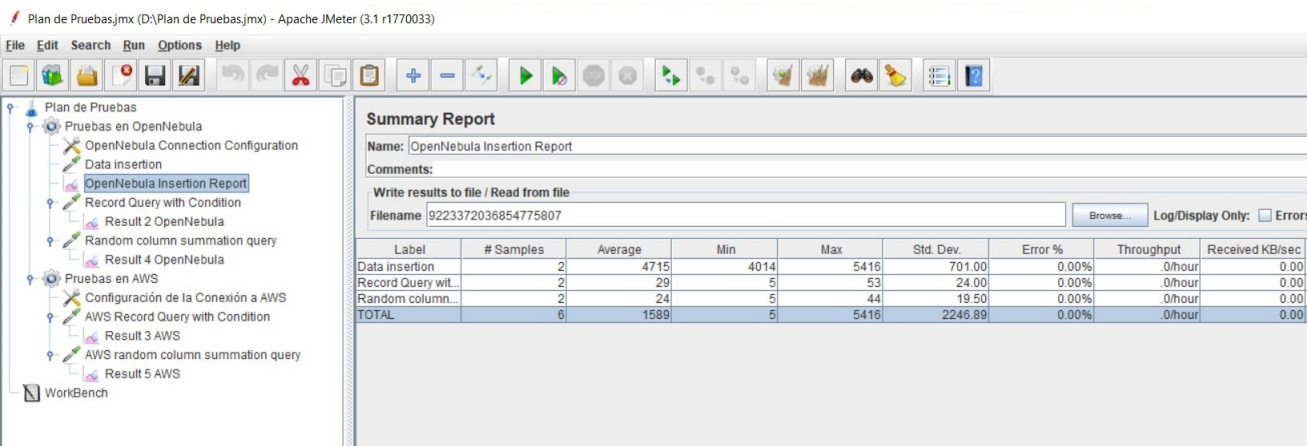


Figura 4-14: Datos replicados de manera correcta. Fuente Autor

Luego en las figuras 4-15 y 4-16 se ejecuta la siguiente consulta para verificar los valores correctos

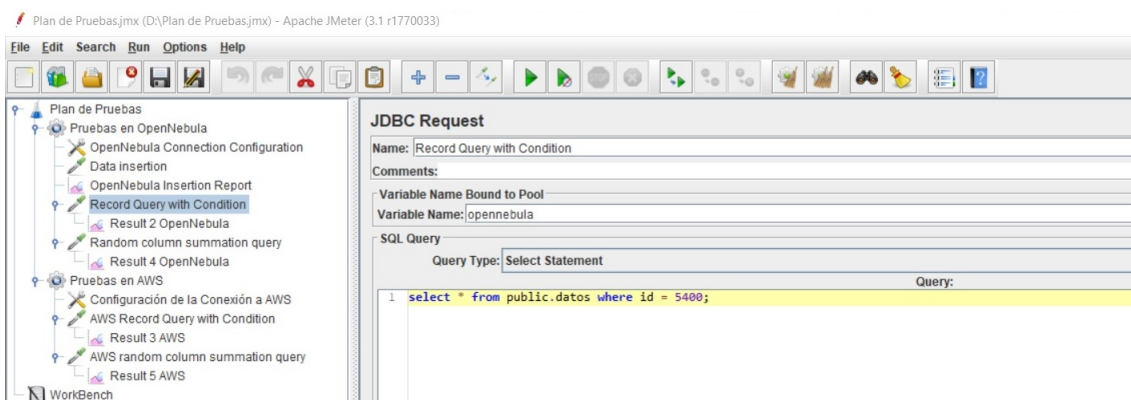


Figura 4-15: Consuta SQL en OpenNebula. Fuente Autor

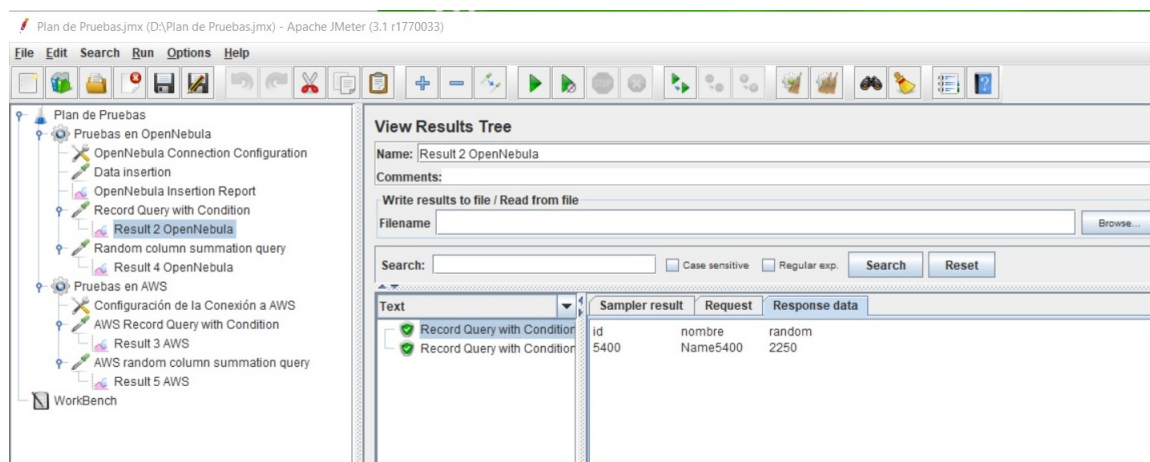


Figura 4-16: Resultado de la consulta en OpenNebula. Fuente Autor

Después en la figura 4-17 y 4-18 se verifica la consulta ejecutada en OpenNebula devuelve el mismo valor en AWS

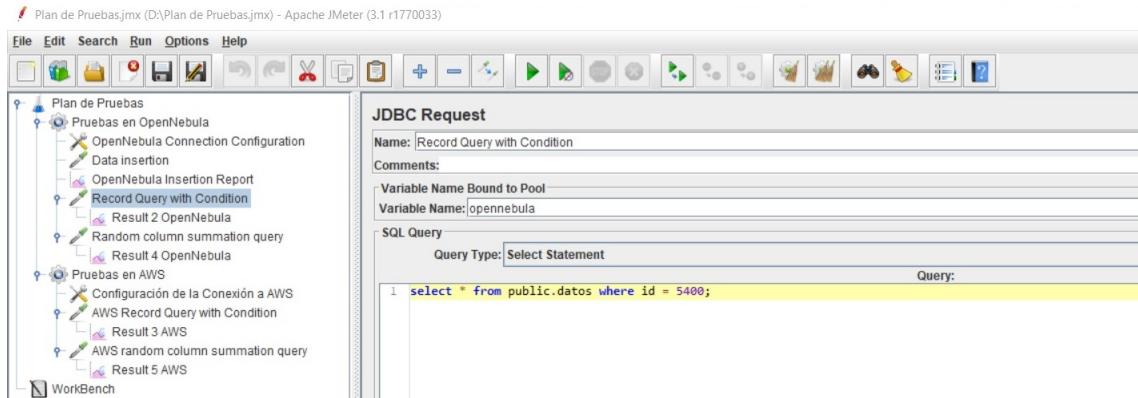


Figura 4-17: Consuta SQL en AWS. Fuente Autor

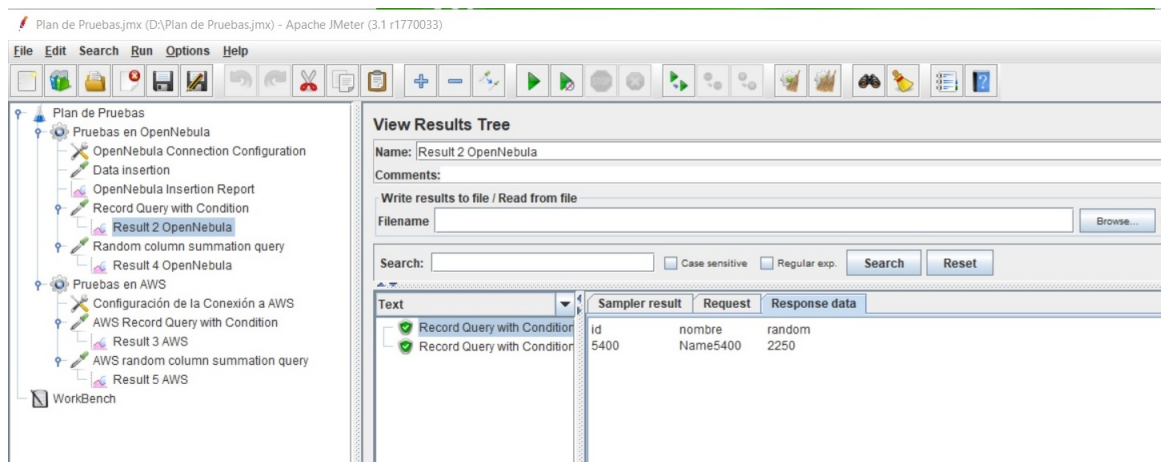


Figura 4-18: Resultado de la consulta en AWS. Fuente Autor

La segunda prueba en una consulta random y se visualizara el mismo resultado en ambas bases de datos como se muestra en las figuras 4-19, 4-20, 4-21 y 4-22.

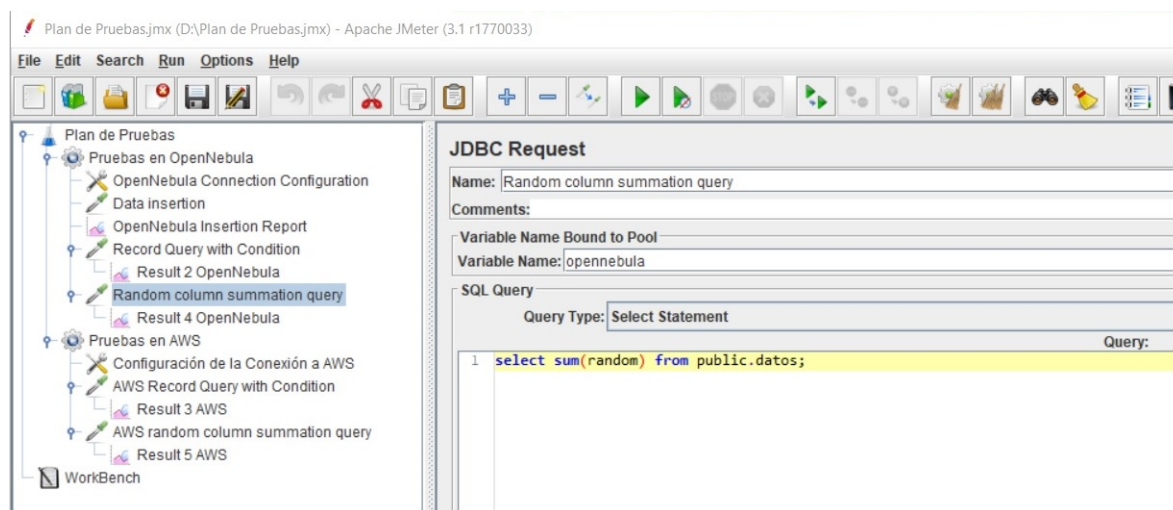


Figura 4-19: Consulta en OpenNebula. Fuente Autor

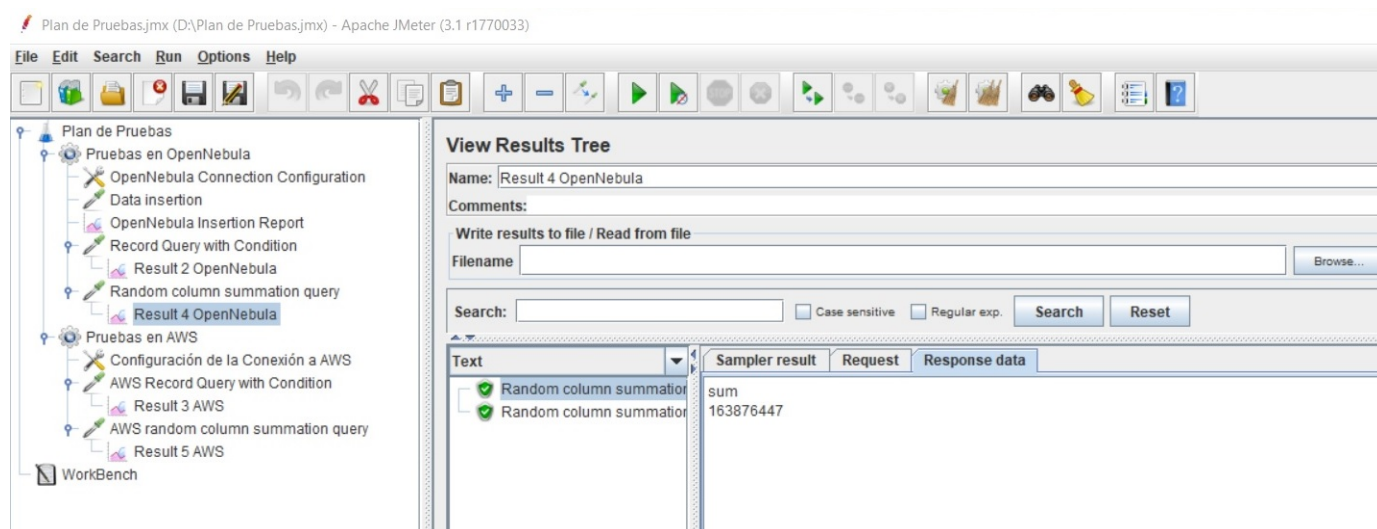


Figura 4-20: Resultado de la consulta en OpenNebula. Fuente Autor

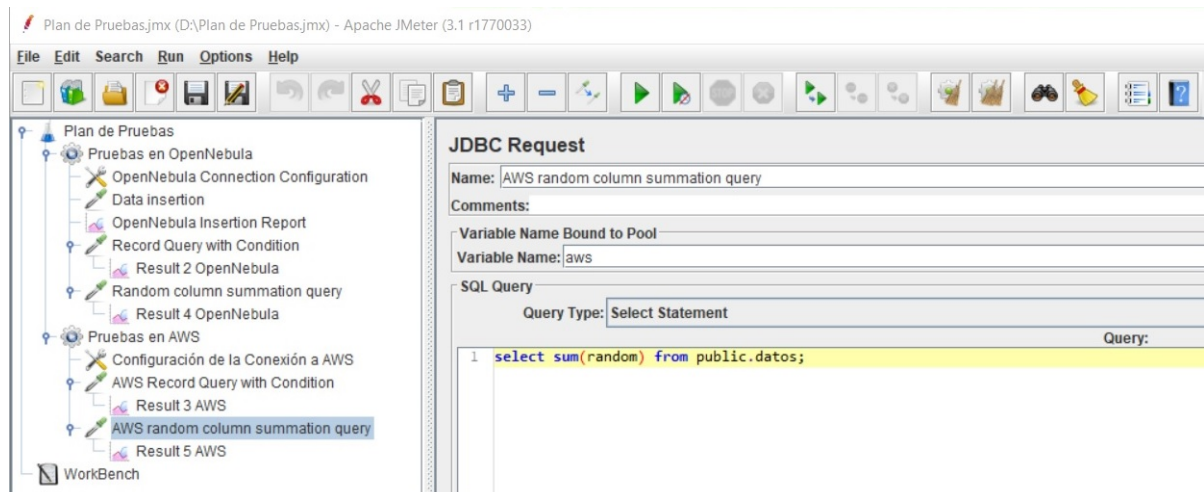


Figura 4-21: Consulta en AWS. Fuente Autor

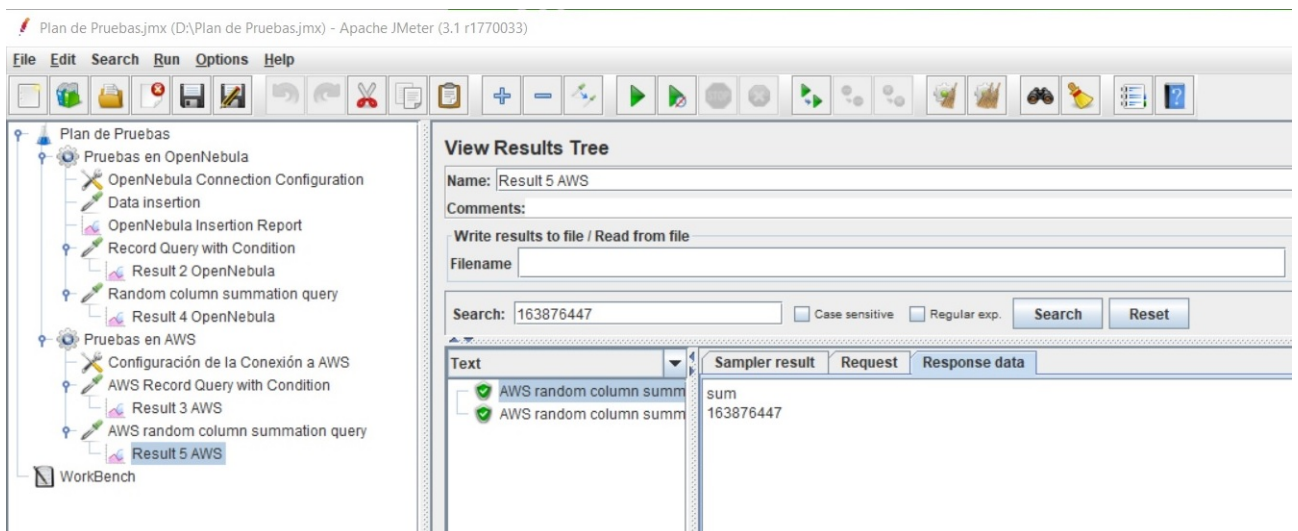


Figura 4-22: Resultado de la consulta en AWS. Fuente Autor

Como prueba final en la figura 4-23 también usaremos DBeaver para crear una base de datos en AWS y se replicará en OpenNebula

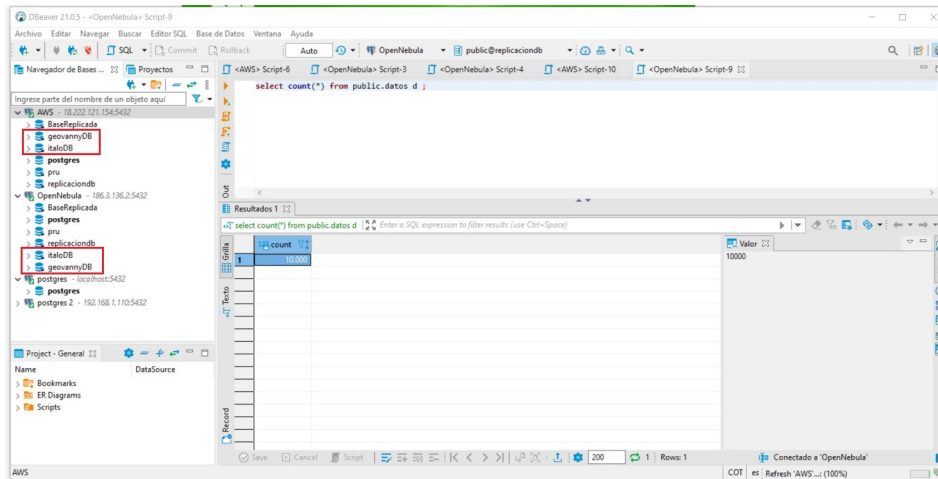


Figura 4-23: Replicación usando DBeaver. Fuente Autor

5 Conclusiones y Recomendaciones

En este último capítulo se brindará las conclusiones y recomendaciones que fueron tomadas directamente de las pruebas de migración y replicación entre los entornos de nube privada y pública realizadas en los capítulos anteriores.

5.1. Conclusiones

- Se ha conseguido determinar y deducir que la integración de una cloud pública en esta caso ubicada en AWS y una cloud privada ubicada en OpenNebula es posible, ya que al usar las herramientas de una manera correcta y al basarse en los servicios disponibles tanto de AWS como de OpenNebula, permite al usuario una gestión y administración completa de ambos entornos, por lo que se deduce que en base a esta integración de ambas plataformas es posible la migración y replicación de datos de una cloud a otra.
- Se logra implementar una arquitectura para la administración de máquinas virtuales, el cual ofrece servicios de replicación de datos para los usuarios de la nube. Esta arquitectura permite gestionar los recursos de los 2 entornos, y así implementar máquinas virtuales como servidores, equipos de red y equipos de almacenamiento y lo más importante, lograr que la comunicación entre ambas plataformas sea eficiente.
- El realizar un estudio y posterior análisis de OpenNebula y AWS, resultó de una gran importancia, puesto que con sus principios de compatibilidad e integración para ambientes híbridos se pudo definir a OpenNebula en conjunto con Postgres, como una metodología para un proceso de migración y replicación de datos de forma ordenada, correcta y eficiente, y el cual brinda la accesibilidad a otros proyectos o infraestructuras independientemente de la infraestructura de OpenNebula.
- Utilizar OpenVPN nos ayudó a crear una infraestructura la cual nos permitió conectar una nube privada (OpenNebula) y dos nubes públicas (AWS y Azure), esto puede ayudar a proyectos o empresas que estén comenzando y no tengan presupuesto para adquirir un IP pública. Ya que utilizando OpenVPN no tenemos restricción de puertos, como es el caso de un plan básico de un proveedor de servicios de internet (ISP) que bloquean los puertos.
- Con la infraestructura propuesta logramos crear un cloud híbrido replicando así un AD (Active Directory), teniendo así tres centros de datos dando ventaja que todos

ellos contendran los mismos datos para que clientes puedan acceder a sistemas o otras aplicaciones.

- En base al análisis de los resultados en cuanto a la replicación de datos, se determina que existe una replicación correcta de todos los datos, esto es posible solo si se configura de manera correcta Postgres y ambas plataformas.
- Como conclusión final, el poder acceder a los servicios de una infraestructura de cloud como lo es AWS para entornos de desarrollo, es factible ya que permite al usuario o empresas que lo adquieren, el poder contar con muchos servicios y simplificar mucho mas sus tiempos de proceso, por lo que se puede determinar que una infraestructura de cloud híbrida es una excelente opción de implementación para las empresas ya sean estas medianas o grandes y de esta manera el poder tener sus datos y servicios seguros protegidos y permitirse crear ambientes en un entorno privado y con ello tambien en entornos de producción en un cloud público como AWS.

5.2. Recomendaciones

- Para crear, gestionar y trabajar con virtuales en la plataforma de OpenNebula de manera más sencilla es recomendable utilizar el formato qcow2 al momento de instanciar las Isos a usar, ya que este formato es compatibles con las aplicaciones de software disponibles las cuales ejecutan Linux.
- Al momento que se desee promover un nuevo servidor en Windows Server para que replique ,tener en cuenta muy bien el dominio intentar agregar el dominio completo por ejemplo: drm.com o dmr.
- Al crear Instancias en Aws o maquinas virtuales en Azure no crear muchas maquinas ya que podriamos exceder los creditos gratuitos que nos proporcionan cada plataformas llevando asi a pagar un valor que no teniamos pensado al principio

6 Anexos

6.1. Instalación y configuración de OpenNebula

En este apartado se va a detallar paso a paso la correcta instalación de Opennebula:

- Características
 - 2 CentOS 7 x86.
 - 5 Gb de memoria RAM.
 - 50 Gb de disco (Centos Maestro).
 - 90 Gb de disco (Centos Nodo).
- Como primer paso al momento de la instalación de Centos, se debe aclarar que el nodo que hará de esclavo para OpenNebula debera llevar una tarjeta de red adicional, pero en modo bridge.
- Como segundo paso y antes de empezar con la instalación de la plataforma OpenNebula, debemos tener los repositorios y dependencias actualizadas en ambos Centos, para esto se debe ejecutar el siguiente comando.

```
yum -y update
```

- Luego se procede a asignar una IP fija a ambas máquinas, para evitar que el DHCP otorgue una direccion distinta en cada sesion, y esto se consigue con el comando nmtui desde la consola del sistema operativo.

Se procede con los pasos a seguir para la correcta instalación de OpenNebula:

- Instalación en el nodo Master, figura 6-1

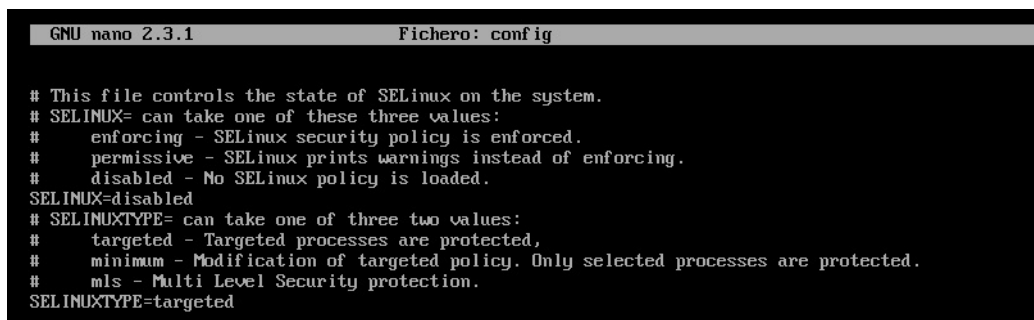
```
yum install vim net-tools bridge-utils nfs-utils
```

```
yum install epel-release
```

Se procede a deshabilitar SELINUX

```
/etc/selinux/config
```

```
SELINUX=disabled
```



```

GNU nano 2.3.1          Fichero: config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three two values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

```

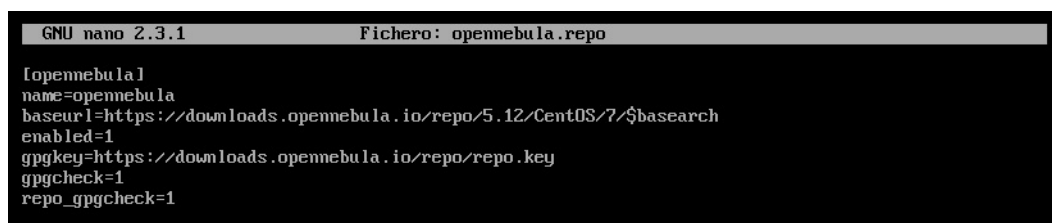
Figura 6-1: Se deshabilita SELINUX. Fuente Autor

Luego a agregar los repositorios, figura 6-2.

```

cat « " EOT " » /etc/yum.repos.d/opennebula.repo
[opennebula]
name=opennebula
baseurl=https://downloads.opennebula.io/repo/5.12/CentOS/7/$basearch
enabled=1
gpgkey=https://downloads.opennebula.io/repo/repo.key
gpgcheck=1
repo_gpgcheck=1
EOT

```



```

GNU nano 2.3.1          Fichero: opennebula.repo

[opennebula]
name=opennebula
baseurl=https://downloads.opennebula.io/repo/5.12/CentOS/7/$basearch
enabled=1
gpgkey=https://downloads.opennebula.io/repo/repo.key
gpgcheck=1
repo_gpgcheck=1

```

Figura 6-2: Se agrega los repositorios. Fuente Autor

```

yum install opennebula-server opennebula-sunstone opennebula-ruby opennebula-gate opennebula-
flow

```

- Instalación en el Nodo Esclavo figura 6-3

```

yum install vim net-tools bridge-utils nfs-utils

```

```

yum install epel-release

```

Se procede a deshabilitar SELINUX

```

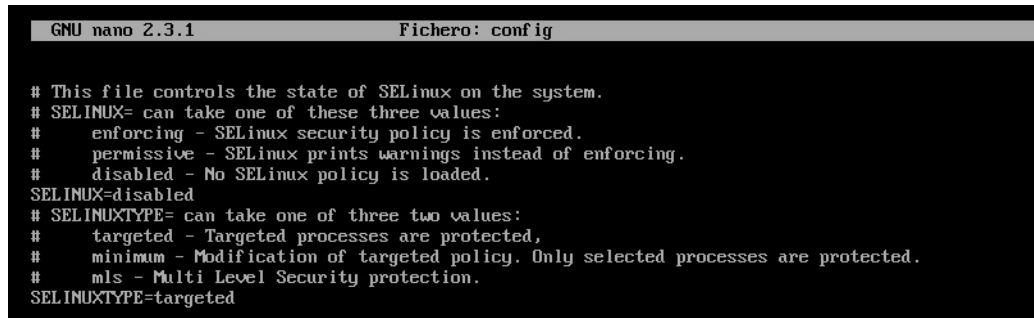
/etc/selinux/config

```

```

SELINUX=disabled

```

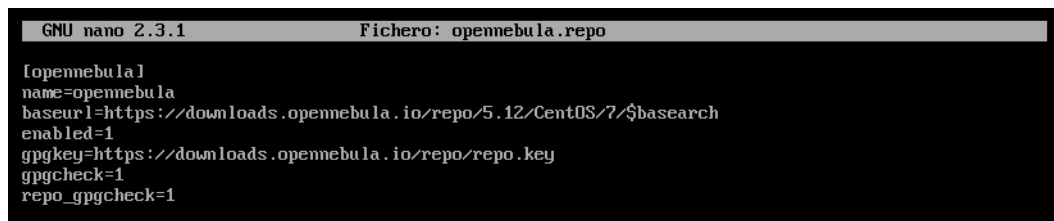
```
GNU nano 2.3.1          Fichero: config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three two values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Figura 6-3: Se deshabilita SELINUX. Fuente Autor

Luego a agregar los repositorios, figura 6-4

```
cat « " EOT " » /etc/yum.repos.d/opennebula.repo
[opennebula]
name=opennebula
baseurl=https://downloads.opennebula.io/repo/5.12/CentOS/7/$basearch
enabled= 1
gpgkey=https://downloads.opennebula.io/repo/repo.key
gpgcheck=1
repo_gpgcheck=1
EOT
```



```
GNU nano 2.3.1          Fichero: opennebula.repo

[opennebula]
name=opennebula
baseurl=https://downloads.opennebula.io/repo/5.12/CentOS/7/$basearch
enabled=1
gpgkey=https://downloads.opennebula.io/repo/repo.key
gpgcheck=1
repo_gpgcheck=1
```

Figura 6-4: Se agrega los repositorios. Fuente Autor

- En este paso se procede a instalar el kvm de opennebula.

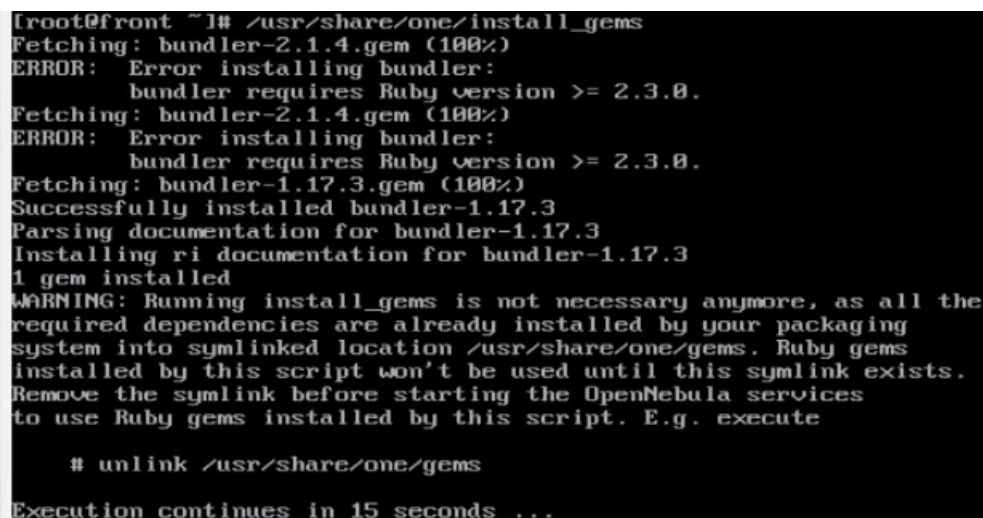
```
sudo yum install opennebula-node-kvm
```

```
sudo systemctl restart libvirtds
```

- Nodo Maestro

- En este paso se instalará las gemas como se indica en la figura 6-5.

/usr/share/one/install_gems



```
[root@front ~]# /usr/share/one/install_gems
Fetching: bundler-2.1.4.gem (100%)
ERROR: Error installing bundler:
       bundler requires Ruby version >= 2.3.0.
Fetching: bundler-2.1.4.gem (100%)
ERROR: Error installing bundler:
       bundler requires Ruby version >= 2.3.0.
Fetching: bundler-1.17.3.gem (100%)
Successfully installed bundler-1.17.3
Parsing documentation for bundler-1.17.3
Installing ri documentation for bundler-1.17.3
1 gem installed
WARNING: Running install_gems is not necessary anymore, as all the
required dependencies are already installed by your packaging
system into symlinked location /usr/share/one/gems. Ruby gems
installed by this script won't be used until this symlink exists.
Remove the symlink before starting the OpenNebula services
to use Ruby gems installed by this script. E.g. execute

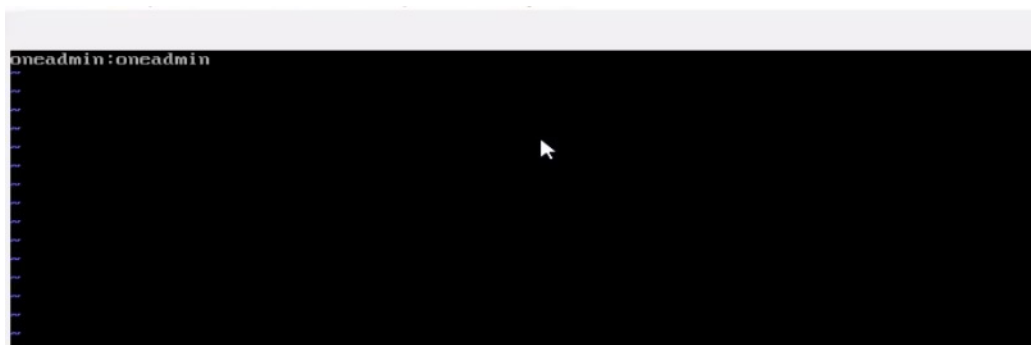
    # unlink /usr/share/one/gems

Execution continues in 15 seconds ...
```

Figura 6-5: Instalación de las gemas. Fuente Autor

- En este paso se procede a cambiar a contraseña por defecto, a la contraseña oneadmin, tal y como muestra la figura 6-6.

/var/lib/one/.one/one_auth



```
oneadmin:oneadmin
```

Figura 6-6: Se cambia la contraseña por defecto. Fuente Autor

- Ahora se procede a iniciar el servicio de opennebula, donde si los pasos anteriores están bien, el servicio se iniciará sin problemas.

systemctl start opennebula

systemctl start opennebula-sunstonen

- Luego se procede a conectarnos como Oneadmin con el siguiente comando:

su oneadmin

- Nodo Esclavo

- Desde el nodo esclavo se procede a reiniciar libvirtd que permitirá la virtualización.

systemctl restart libvirtd

- Se procede a instalar los paquetes de virtualización, se detalla en la figura 6-7

yum install centos-release-gemu-ev

yum install qemu-kvm-ev

```
usbredir.x86_64 0:0.7.1-3.el7

¡Listo!
[root@nodo network-scripts]# systemctl restart libvirtd
[root@nodo network-scripts]# yum install centos-release-gemu-ev
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.uta.edu.ec
 * epel: mirror.uta.edu.ec
 * extras: mirror.uta.edu.ec
 * updates: mirror.uta.edu.ec
Resolviendo dependencias
--> Ejecutando prueba de transacción
--> Paquete centos-release-gemu-ev.noarch 0:1.0-4.el7.centos debe ser instalado
--> Procesando dependencias: centos-release-virt-common para el paquete: centos-release-gemu-ev-1.0-4.el7.centos.noarch
--> Ejecutando prueba de transacción
--> Paquete centos-release-virt-common.noarch 0:1-1.el7.centos debe ser instalado
--> Resolución de dependencias finalizada

Dependencias resueltas

=====
Package                                Arquitectura  Versión                Repositorio  Tamaño
=====
Instalando:
centos-release-gemu-ev                  noarch       1.0-4.el7.centos       extras       11 k
Instalando para las dependencias:
centos-release-virt-common              noarch       1-1.el7.centos         extras       4.5 k

Resumen de la transacción
=====
Instalar 1 Paquete (+1 Paquete dependiente)

Tamaño total de la descarga: 16 k
Tamaño instalado: 19 k
Is this ok [y/d/N]:
```

Figura 6-7: Instalación de paquetes de virtualización. Fuente Autor

- Nodo Maestro

- Como primer paso, ay que conectarse como oneadmin

su oneadmin

- Como siguiente paso ay que dirigirse a esta ruta y borrar todos los archivos que están alojados en esta ruta, figura 6-8:

```
cd /var/lib/one/.ssh
```

```
[root@front ~]# su oneadmin
[oneadmin@front root]$ cd /var/lib/one/.ssh/
[oneadmin@front .ssh]$ ls
authorized_keys  config  id_rsa  id_rsa.pub
[oneadmin@front .ssh]$ rm id_rsa
[oneadmin@front .ssh]$ rm id_rsa.pub
[oneadmin@front .ssh]$ rm authorized_keys
[oneadmin@front .ssh]$ rm config
[oneadmin@front .ssh]$ ls
[oneadmin@front .ssh]$ _
```

Figura 6-8: Eliminando de todos los archivos dentro de esta ruta. Fuente Autor

- Ahora se instalará las llaves públicas y privadas como se indica en la figura 6-9.

```
ssh-keygen
```

```
[oneadmin@front .ssh]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/one/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/one/.ssh/id_rsa.
Your public key has been saved in /var/lib/one/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:h1y9BwNeUwIKCuaUlzRSb+c2bWw/RaCcEHNR0An1gUQ oneadmin@front.local
The key's randomart image is:
+----[RSA 2048]-----+
|      =+++ .XXBE=|
|      +.oo=B.=o+.|
|      ...o+o.o .|
|      . o.oo=.. |
|      S . =+ =. |
|      . . =.. |
|      o |
|      o |
|      . |
+----[SHA256]-----+
[oneadmin@front .ssh]$
```

Figura 6-9: Instalación de las llaves. Fuente Autor

- Nodo Esclavo

Ahora se instalará las llaves públicas y privadas, se indica en la figura 6-10.

ssh-keygen

```
[oneadmin@front .ssh]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/one/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/one/.ssh/id_rsa.
Your public key has been saved in /var/lib/one/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:h1y9BWNeUwIKCuaUlzRSb+c2bWw/RaCcEHNR0An1gUQ oneadmin@front.local
The key's randomart image is:
+---[RSA 2048]-----+
|      =+++..XXBE=|
|      +.oo=B.=o+.|
|      ...o+o.o .|
|      .o.oo=..|
|      S . =+ =.|
|      . . =..|
|      .o|
|      .o|
|      .|
+---[SHA256]-----+
[oneadmin@front .ssh]$
```

Figura 6-10: Instalación de las llaves. Fuente Autor

- Nodo Maestro

En la figura 6-11 como primer paso se cierra la sesión con el usuario oneadmin para poder cambiar la contraseña.

passwd oneadmin

```
[oneadmin@front .ssh]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/one/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/one/.ssh/id_rsa.
Your public key has been saved in /var/lib/one/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:h1y9BWNeUwIKCuaUlzRSb+c2bWw/RaCcEHNR0An1gUQ oneadmin@front.local
The key's randomart image is:
+---[RSA 2048]-----+
|      =+++..XXBE=|
|      +.oo=B.=o+.|
|      ...o+o.o .|
|      .o.oo=..|
|      S . =+ =.|
|      . . =..|
|      .o|
|      .o|
|      .|
+---[SHA256]-----+
[oneadmin@front .ssh]$
```

Figura 6-11: Cambio de clave en el nodo Maestro. Fuente Autor

- Nodo Esclavo

En este nodo también se cierra la sesión con el usuario oneadmin para poder cambiar la contraseña, figura 6-12.

passwd oneadmin

```
[root@nodo ~]# passwd oneadmin
Cambiando la contraseña del usuario oneadmin.
Nueva contraseña:
CONTRASEÑA INCORRECTA: De alguna manera, en la contraseña se lee el nombre del usuario
Vuelva a escribir la nueva contraseña:
passwd: todos los símbolos de autenticación se actualizaron con éxito.
[root@nodo ~]#
```

Figura 6-12: Cambio de clave en el nodo Esclavo. Fuente Autor

- Nodo Maestro

En la figura 6-13 se procede a copiar la llave pública al nodo esclavo.

scp /var/lib/one/.ssh/id_rsa.pub oneadmin@nodo.local:/var/lib/one/.ssh/llavefront

```
[oneadmin@front root]$ scp /var/lib/one/.ssh/id_rsa.pub oneadmin@nodo.local:/var/lib/one/.ssh/llavefront
The authenticity of host 'nodo.local (192.168.1.41)' can't be established.
ECDSA key fingerprint is SHA256:x8j4DTy+R5CgyGMIf+sr2+VdH0jyKbHeo3J28IU10lo.
ECDSA key fingerprint is MD5:5e:17:c5:fc:3a:ca:d1:e4:8a:1e:4a:85:1c:8b:f7:e7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'nodo.local,192.168.1.41' (ECDSA) to the list of known hosts.
oneadmin@nodo.local's password:
id_rsa.pub 100% 402 452.5KB/s 00:00
[oneadmin@front root]$ _
```

Figura 6-13: Se copia la llave pública al nodo esclavo. Fuente Autor

- Nodo Esclavo

De igual manera en la figura 6-14 se procede a copiar la llave pública pero en este caso al nodo maestro.

scp /var/lib/one/.ssh/id_rsa.pub oneadmin@front.local:/var/lib/one/.ssh/llavenodo

```
[oneadmin@nodo ~]$ scp /var/lib/one/.ssh/id_rsa.pub oneadmin@front.local:/var/lib/one/.ssh/llavenodo
Warning: Permanently added 'front.local,192.168.1.40' (ECDSA) to the list of known hosts.
oneadmin@front.local's password:
id_rsa.pub 100% 401 504.7KB/s 00:00
[oneadmin@nodo ~]$ _
```

Figura 6-14: Se copia la llave pública al nodo maestro. Fuente Autor

- Nodo Maestro

Se cierra la sesión con el usuario `oneadmin`, para poder eliminar la clave con el comando:

```
passwd -d oneadmin
```

- Nodo Esclavo

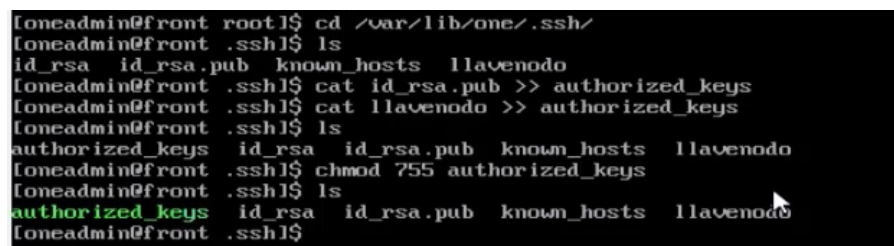
De la misma manera se cierra la sesión con el usuario `oneadmin`, para poder eliminar la clave en el nodo esclavo con el comando:

```
passwd -d oneadmin
```

- Nodo Maestro

Ahora en la figura **6-15** se va a crear una carpeta que contendrá los archivos de las llaves autorizadas en el nodo maestro, para posterior dar permisos.

```
cd /var/lib/one/.ssh  
cat id_rsa.pub authorized_keys  
cat llavenodo authorized_keys  
chmod 755 authorized_keys
```



```
loneadmin@front root1$ cd /var/lib/one/.ssh/  
loneadmin@front .ssh1$ ls  
id_rsa id_rsa.pub known_hosts llavenodo  
loneadmin@front .ssh1$ cat id_rsa.pub >> authorized_keys  
loneadmin@front .ssh1$ cat llavenodo >> authorized_keys  
loneadmin@front .ssh1$ ls  
authorized_keys id_rsa id_rsa.pub known_hosts llavenodo  
loneadmin@front .ssh1$ chmod 755 authorized_keys  
loneadmin@front .ssh1$ ls  
authorized_keys id_rsa id_rsa.pub known_hosts llavenodo  
loneadmin@front .ssh1$
```

Figura 6-15: Creación de la carpeta que contendrá las llaves autorizadas. Fuente Autor

- Nodo Esclavo

Se repite el paso anterior en la figura **6-16** pero ahora desde el nodo esclavo y se va a crear una carpeta que contendrá los archivos de las llaves autorizadas en el nodo esclavo, para posterior dar permisos.

```
cd /var/lib/one/.ssh  
cat id_rsa.pub authorized_keys  
cat llavemaster authorized_keys  
chmod 755 authorized_keys
```

```

loneadmin@nodo /1$ cd /var/lib/one/.ssh/
loneadmin@nodo .ssh1$ ls
config id_rsa id_rsa.pub known_hosts llavefront
loneadmin@nodo .ssh1$ cat id_rsa.pub >> authorized_keys
loneadmin@nodo .ssh1$ cat llavefront >> authorized_keys
loneadmin@nodo .ssh1$ chmod 755 authorized_keys
loneadmin@nodo .ssh1$ ls
authorized_keys config id_rsa id_rsa.pub known_hosts llavefront
loneadmin@nodo .ssh1$ _

```

Figura 6-16: Creación de la carpeta que contendrá las llaves autorizadas. Fuente Autor

- Nodo Maestro y Esclavo

Se edita el archivo de configuración de ssh para permitir la conexión entre ambos nodos, tal y como se muestra en la figura 6-17.

```
nano /etc/ssh/sshd_config
```

```
systemctl restart sshd
```

```

#      $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $
#
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.
#
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin
#
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.
#
# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
#
HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#
# Ciphers and keying
#RekeyLimit default none
#
# Logging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO
#
# Authentication:
"/etc/ssh/sshd_config" 139L, 3907C
1,1 Comienzo

```

Figura 6-17: Se edita el archivo de configuración de SSH. Fuente Autor

- Nodo Maestro

Ahora en la figura 6-18 se creará un nodo interno desde el nodo maestro para poder desplegar las maquinas virtuales:

```
su oneadmin
```

```
onehost create nodo.local -i kvm -v kvm
```

```
onehost list
```

```
[oneadmin@front root]$ onehost create nodo.local -i kvm -v kvm
ID: 0
[oneadmin@front root]$ onehost list
```

ID	NAME	CLUSTER	TUN	ALLOCATED_CPU	ALLOCATED_MEM	STAT
0	nodo.local	default	0	0 / 100 (0%)	0K / 2.8G (0%)	on

```
[oneadmin@front root]$ _
```

Figura 6-18: Creación del nodo para despliegue de las maquinas virtuales. Fuente Autor

Con todos estos pasos correctamente ejecutados, y como se observa en la figura 6-19 ya se puede abrir el gestor de OpenNebula desde el navegador, con la IP del nodo maestro, el puerto 9869 y credenciales de usuario y contraseña con oneadmin.

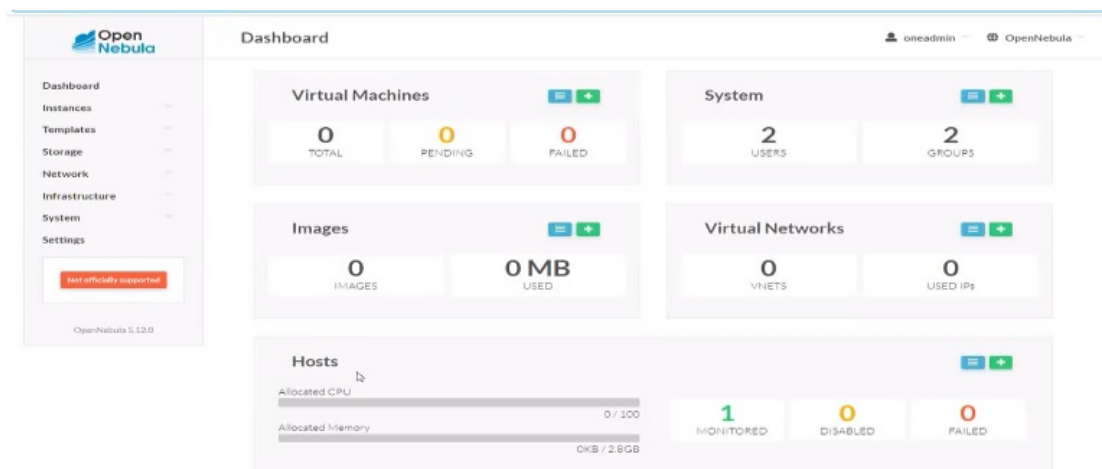


Figura 6-19: Despliegue de OpenNebula. Fuente Autor

6.2. Configuración para la replicación

Para poder implementar la metodología mencionada anteriormente se procede a seguir los siguientes pasos:

- Se configura el reenvío de puertos y los grupos de seguridad en AWS y Azure.
- Se configura la nube privada.
- Se configura las nubes públicas.
- Se configuramos la base de datos en la nube privada.

6.2.1. Configuración para permitir la salida a internet

- Para permitir la conexión a Internet de la instancia de OpenNebula, AWS y Azure y así poder replicar las bases de datos se debe configurar el reenvío de puertos en el dispositivo de tercera capa donde está conectada la interfaz de red OpenNebula, no se debe olvidar que hay que definir los puertos usados en el conexión, como se indica en la figura 6-20.

Figura 6-20: Configuración de Port Forwarding para la salida a internet. Fuente Autor

- Configuración de la nube privada (OpenNebula): En la plataforma de OpenNebula, se hizo uso de dos máquinas virtuales, la primera máquina virtual que se utilizará como maestra, es decir para el Front, donde incluye la interfaz Opennebula-sunstone la cual sirve para la gestión web de la nube, y también para el control de las interfaces de red instaladas. Y la segunda máquina virtual utilizada denominada esclavo la cual se usará para "Node", donde se almacenarán los archivos de las instancias y la cual se encargará de asignar memoria RAM a las instancias que se ha creado.

- Configuración de las nubes públicas: En AWS permite crear instancias de varios sistemas operativos, la creación de la instancia empleada fue orientada por la consola EC2 para elegir el sistema operativo y el tipo de instancia que se desea usar, AWS posee 55 tipos de instancias las cuales constan de varios recursos, para el caso se empleo el tipo "t2 .micro con el sistema operativo ubuntu 16.04.

La figura 6-21 muestra la instancia en AWS, posterior a crear la instancia, se puede observar que a la instancia se le ha asignado una IPv4 pública, esta IP es la que se utilizó para realizar las conexiones entre las instancias y poder conectar las bases de datos para realizar la posterior replicación.

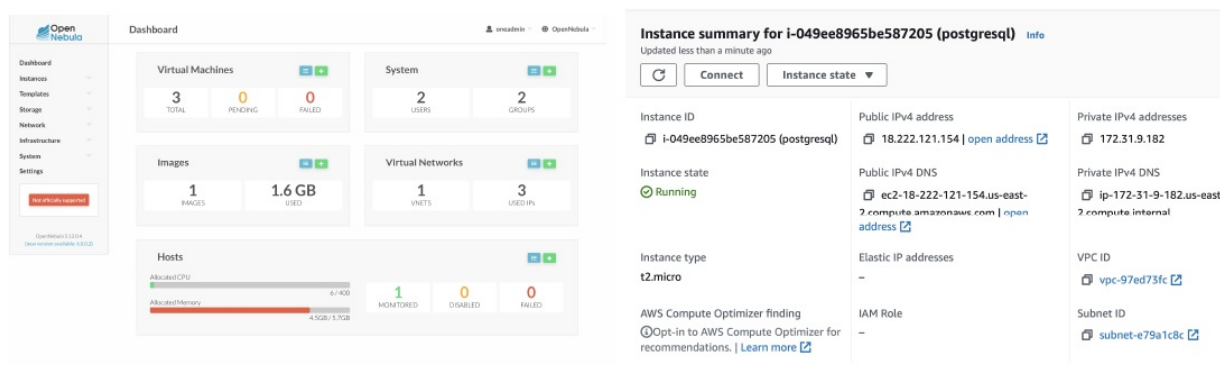


Figura 6-21: Dashboard de Opennebula y EC2. Fuente Autor

6.3. Configuración de instancias AWS y Azure

Para poder configurar ambas plataformas para la posterior replicación de los datos alojados en la base de datos es necesario ejecutar los siguientes pasos:

- Una vez que se haya desplegado OpenNebula, verificamos que en el Nodo Front este creado el kvm, esto permitirá el almacenamiento de los recursos para las instancias creadas en OpenNebula, después de verificar esto, se procederá a abrir un navegador con la ip del front de OpenNebula, para poder ingresar el usuario y la contraseña para acceder a OpenNebula mediante Dashboard, el mismo que permitirá la administración y manejo de todos los elementos de OpenNebula, ya que resultará más sencillo debido a que se tiene una interfaz amigable para poder visualizar.

Los recursos de las instancias creadas en OpenNebula para su posterior replicación de servicios, el cual está explicado en el capítulo 2 del presente documento.

Configuración de Amazon Web Services

- En AWS se desplegará un t2.micro que es equivalente a una instancia en OpenNebula, para ello es necesario realizar algunas configuraciones.
- Una vez creada la instancia, se podrá abrir la misma y verificar que se tiene una dirección IPV4 pública y un DNS de IPV4 pública, las mismas que serán usadas para poder realizar las conexiones necesarias a la instancia que se creó anteriormente, y a

la cual se puede conectar via SSH, usando la propia consola de AWS como se puede observar en la figura **6-22** o una conexión SSH a través de una terminal linux en una computadora o un cliente SSH con GUI.

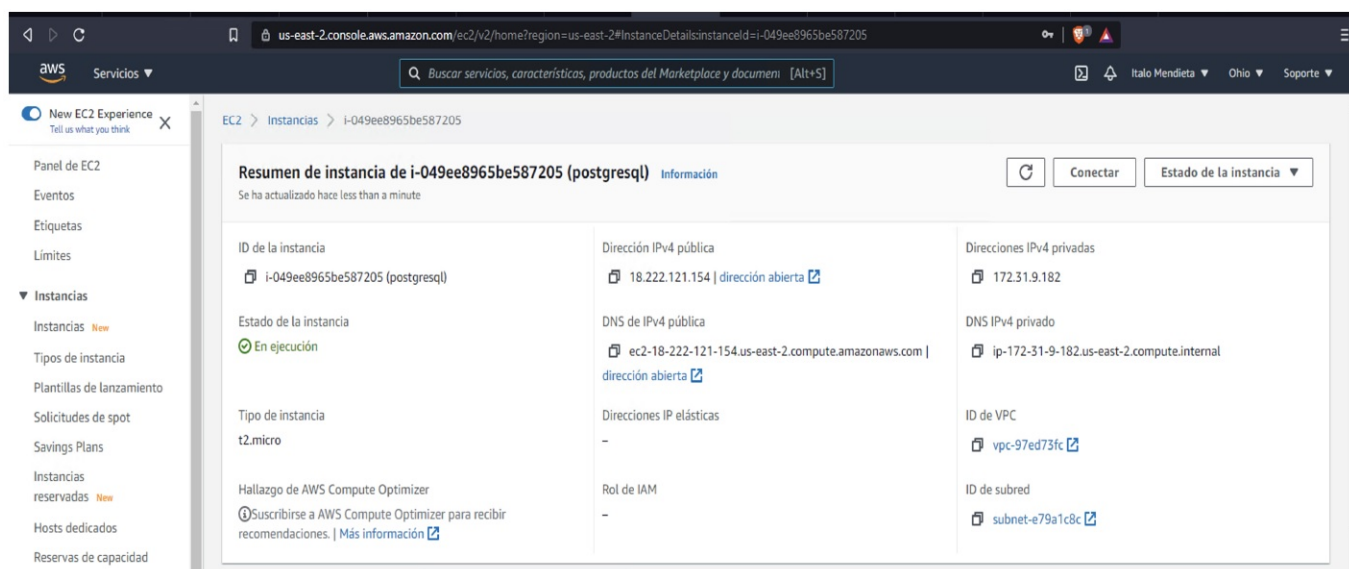


Figura 6-22: Instancia AWS. Fuente Autor

Configuración de Azure

- En Azure se desplegará un DS1 que es equivalente a una instancia en OpenNebula, para ello es necesario realizar algunas configuraciones.
- Una vez creada la instancia, figura **6-23**, se podrá abrir la misma y verificar que se tiene una dirección IPV4 pública, la mismas que sera usada para poder realizar las conexiones necesarias a la instancias que se creo anteriormente, y a la cual se puede conectar via SSH.

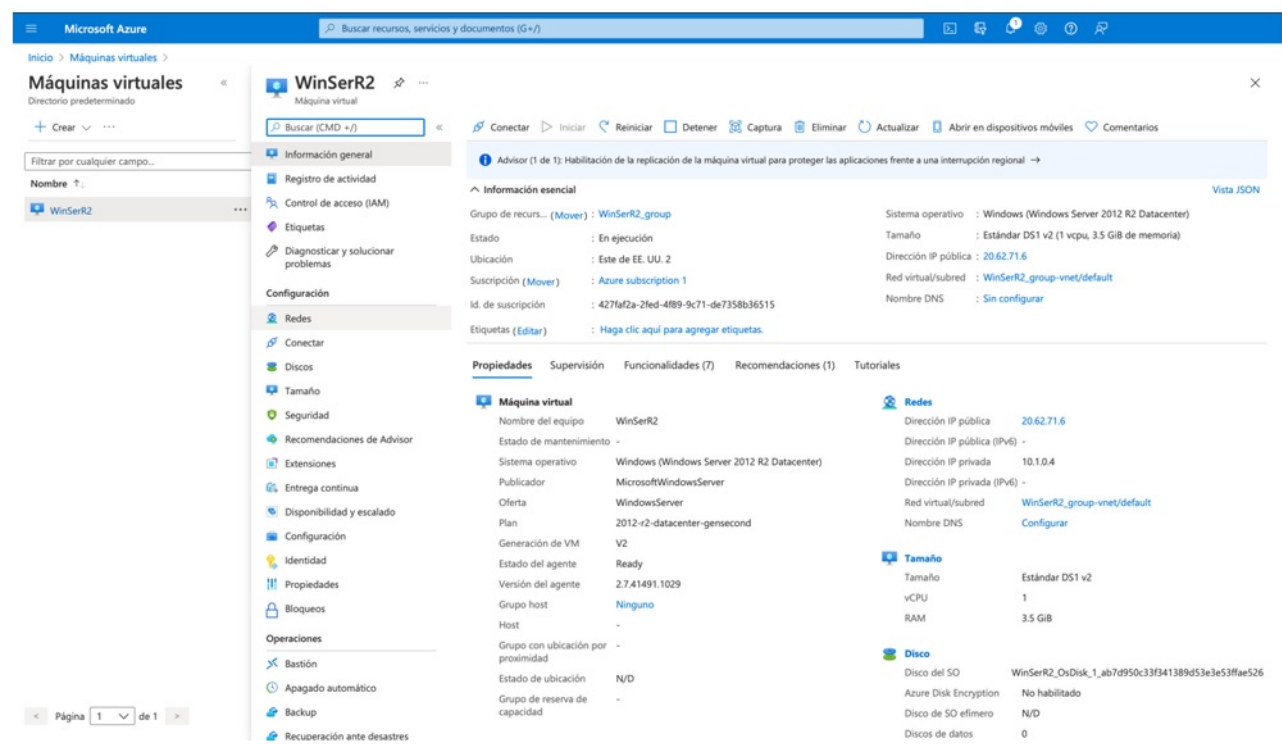


Figura 6-23: Instancia Azure. Fuente Autor

- Instalación y configuración de Postgres en la instancia de Ubuntu en OpenNebula
 1. Como primer paso se procede a instalar postgres desde la terminal de Ubuntu con los siguientes comandos:
 - yum update
 - yum upgrade -y
 - reboot
- Se procede a instalar Postgres en ambos nodos y posterior hay que asegurarse de que PostgreSQL esté habilitado y ejecutándose, como se indica en la figura 6-24:

```

yum install -y postgresql
systemctl enable --now postgresql@11-main.service

```

```

Resumen de la transacción
=====
Actualizar 1 Paquete (+2 Paquetes dependientes)

Tamaño total de la descarga: 4.2 M
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
(1/3): postgresql-libs-9.2.24-7.el7_9.x86_64.rpm                | 235 kB  00:00:00
(2/3): postgresql-devel-9.2.24-7.el7_9.x86_64.rpm              | 952 kB  00:00:00
(3/3): postgresql-9.2.24-7.el7_9.x86_64.rpm                   | 3.0 MB  00:00:00
-----
Total                                                                6.0 MB/s | 4.2 MB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Actualizando : postgresql-libs-9.2.24-7.el7_9.x86_64          1/6
  Actualizando : postgresql-9.2.24-7.el7_9.x86_64              2/6
  Actualizando : postgresql-devel-9.2.24-7.el7_9.x86_64        3/6
  Limpieza     : postgresql-devel-9.2.24-4.el7_8.x86_64         4/6
  Limpieza     : postgresql-9.2.24-4.el7_8.x86_64              5/6
  Limpieza     : postgresql-libs-9.2.24-4.el7_8.x86_64         6/6
  Comprobando  : postgresql-devel-9.2.24-7.el7_9.x86_64       1/6
  Comprobando  : postgresql-9.2.24-7.el7_9.x86_64             2/6
  Comprobando  : postgresql-libs-9.2.24-7.el7_9.x86_64        3/6
  Comprobando  : postgresql-9.2.24-4.el7_8.x86_64             4/6
  Comprobando  : postgresql-devel-9.2.24-4.el7_8.x86_64       5/6
  Comprobando  : postgresql-libs-9.2.24-4.el7_8.x86_64        6/6

Actualizado:
  postgresql.x86_64 0:9.2.24-7.el7_9

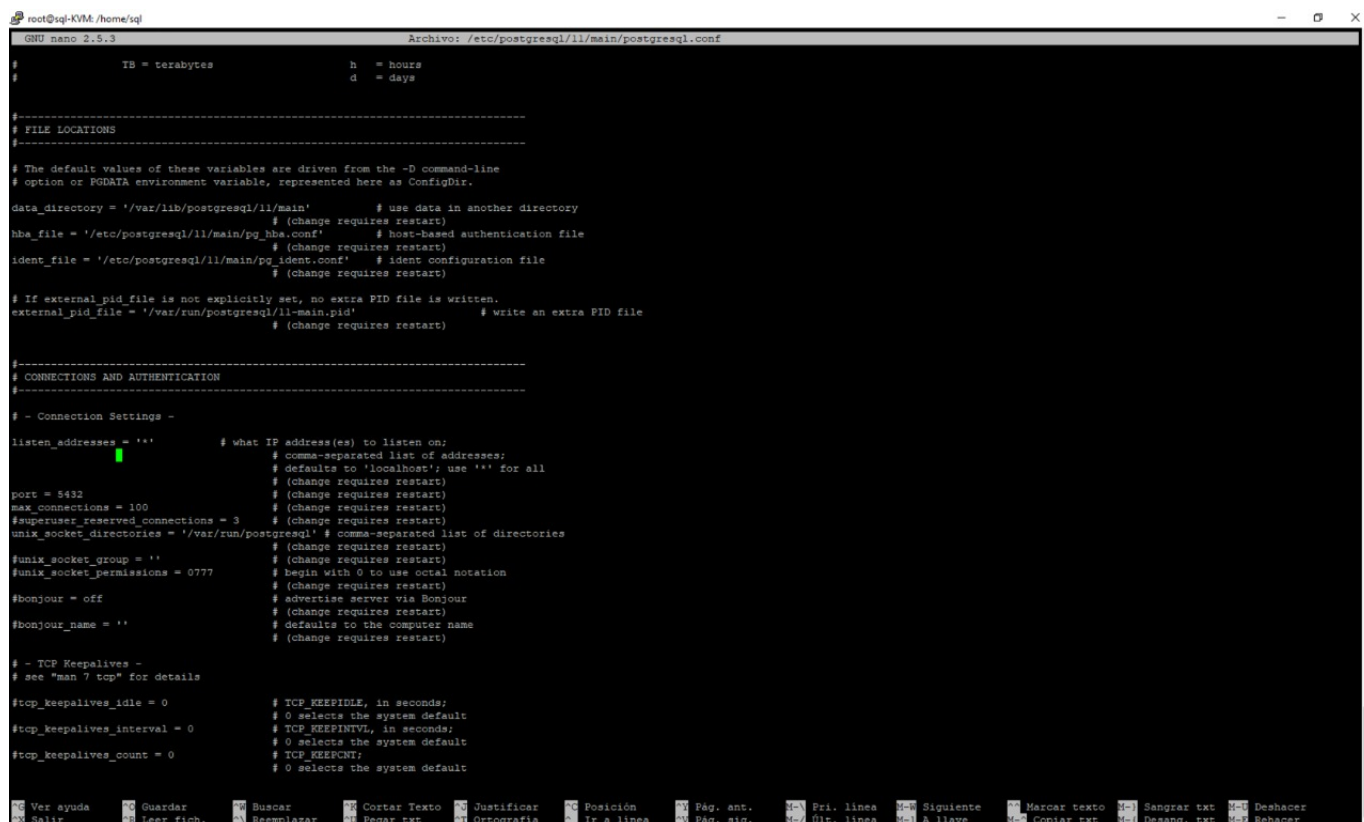
Dependencia(s) actualizada(s):
  postgresql-devel.x86_64 0:9.2.24-7.el7_9          postgresql-libs.x86_64 0:9.2.24-7.el7_9

¡Listo!
[root@master ~]#

```

Figura 6-24: Instalación Postgres. Fuente Autor

- Por defecto, PostgreSQL solo escucha en la interfaz loopback y no es accesible externamente, para esto se cambia la dirección de escucha en los nodos editando el archivo postgresql.conf, en donde se busca la línea "listen addresses = 'localhost'", se modifica y se coloca "listen addresses = 'node ip address,127.0.0.1'", este proceso se indica en la figura 6-25 y 6-26.



```
root@sql-KVM: /home/sql
GNU nano 2.5.3 Archivo: /etc/postgresql/11/main/postgresql.conf
#
#         TB = terabytes          h = hours
#         d = days
#
#-----
# FILE LOCATIONS
#-----
# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.
data_directory = '/var/lib/postgresql/11/main' # use data in another directory
# (change requires restart)
hba_file = '/etc/postgresql/11/main/pg_hba.conf' # host-based authentication file
# (change requires restart)
ident_file = '/etc/postgresql/11/main/pg_ident.conf' # ident configuration file
# (change requires restart)
# If external_pid_file is not explicitly set, no extra PID file is written.
external_pid_file = '/var/run/postgresql/11-main.pid' # write an extra PID file
# (change requires restart)
#-----
# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -
listen_addresses = 'localhost' # what IP address(es) to listen on;
# comma-separated list of addresses;
# defaults to 'localhost'; use '*' for all
# (change requires restart)
port = 5432 # (change requires restart)
max_connections = 100 # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
# (change requires restart)
#unix_socket_group = '' # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
# (change requires restart)
#bonjour = off # advertise server via Bonjour
# (change requires restart)
#bonjour_name = '' # defaults to the computer name
# (change requires restart)
# - TCP Keepalives -
# see "man 7 tcp" for details
#tcp_keepalives_idle = 0 # TCP_KEEPIDL, in seconds;
# 0 selects the system default
#tcp_keepalives_interval = 0 # TCP_KEEPINVL, in seconds;
# 0 selects the system default
#tcp_keepalives_count = 0 # TCP_KEEPCNT;
# 0 selects the system default
```

Figura 6-25: Cambio en la dirección de escucha, nodo maestro. Fuente Autor

```

ubuntu@ip-172-31-9-182: ~
GNU nano 2.5.3 File: /etc/postgresql/11/main/postgresql.conf

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost'; use '*' for all
                                # (change requires restart)
port = 5432                     # (change requires restart)
max_connections = 100           # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
                                # (change requires restart)
#unix_socket_group = ''         # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
                                # (change requires restart)
#bonjour = off                 # advertise server via Bonjour
                                # (change requires restart)
#bonjour_name = ''             # defaults to the computer name
                                # (change requires restart)

# - TCP Keepalives -
# see "man 7 tcp" for details

#tcp_keepalives_idle = 0        # TCP_KEEPIDL, in seconds;
                                # 0 selects the system default
#tcp_keepalives_interval = 0    # TCP_KEEPI, in seconds;

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text   ^J Justify    ^C Cur Pos    ^Y Prev Page  M-^ First Line
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell    ^_ Go To Line   ^V Next Page  M-/_ Last Line

```

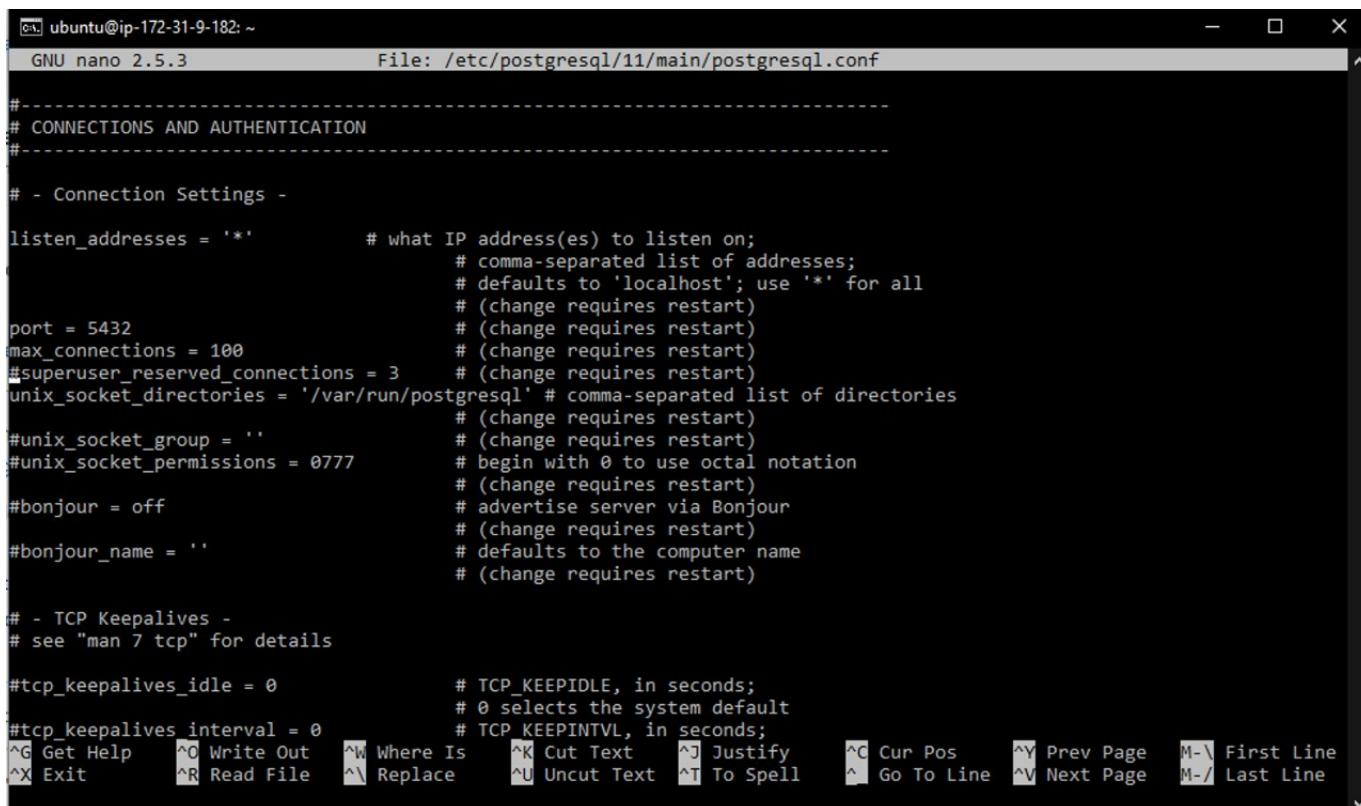
Figura 6-26: Cambio en la dirección de escucha, nodo secundario. Fuente Autor

Se procede a guardar los cambios y luego reiniciar en ambas instancias con el siguiente comando:

```
systemctl restart postgresql@11-main.service
```

- Se procede con la configuración del nodo maestro

Este paso solo se lo hace en el nodo servidor primario (maestro), se abre la terminal de Postgres con "sudo -u postgres psql, luego se ingresa el siguiente comando `CREATE ROLE replicator LOGIN REPLICATION ENCRYPTED PASSWORD 'replicator_sword'`, posterior se creara una ranura de replicación, y luego salir del archivo, se indica en la figura 6-27:



```
ubuntu@ip-172-31-9-182: ~
GNU nano 2.5.3 File: /etc/postgresql/11/main/postgresql.conf

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost'; use '*' for all
                                # (change requires restart)
port = 5432                     # (change requires restart)
max_connections = 100           # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
                                # (change requires restart)
#unix_socket_group = ''         # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
                                # (change requires restart)
#bonjour = off                 # advertise server via Bonjour
                                # (change requires restart)
#bonjour_name = ''             # defaults to the computer name
                                # (change requires restart)

# - TCP Keepalives -
# see "man 7 tcp" for details

#tcp_keepalives_idle = 0        # TCP_KEEPIDLE, in seconds;
                                # 0 selects the system default
#tcp_keepalives_interval = 0    # TCP_KEEPINTVL, in seconds;

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify    ^C Cur Pos    ^Y Prev Page  M-^ First Line
^X Exit      ^R Read File  ^_ Replace   ^U Uncut Text ^T To Spell    ^_ Go To Line  ^V Next Page  M-/ Last Line
```

Figura 6-27: Configuración nodo maestro. Fuente Autor

- Luego en la figura **6-28**, se creará una entrada en el archivo `pg_hba.conf`, esto es para permitir que el usuario del nodo replicador se conecte desde el nodo esclavo al maestro. Se procede a agregar la siguiente línea `host replication replicator standby_ip_address/32 md5` para que los cambios tengan efecto, se reinicia la instancia maestra con el comando `systemctl restart postgresql@11-main.service`.

```
#
# Database administrative login by Unix domain socket
local    all             postgres                                peer

# TYPE      DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local    all             all              peer
# IPv4 local connections:
host     all             all              127.0.0.1/32            md5
# IPv6 local connections:
host     all             all              ::1/128                  md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local    replication     all              peer
host     replication     all              127.0.0.1/32            md5
host     replication     all              ::1/128                  md5
host     replication     replicator       18.222.121.154/32       md5
```

Figura 6-28: Configuración del archivo de postgres. Fuente Autor

- Configuración del nodo esclavo

Este paso solo se lo hace en el servidor secundario (esclavo), donde se habilita el modo de espera activa en el archivo `postgresql.conf`, para ello se digita el comando `/etc/postgresql/11/main/postgresql.conf`, se busca la siguiente línea y se la descomenta `hot_standby = on`, como se muestra en la figura **6-29**

```

ubuntu@ip-172-31-9-182: ~
GNU nano 2.5.3 File: /etc/postgresql/11/main/postgresql.conf

#vacuum_defer_cleanup_age = 0 # from standby(s); '*' = all
# number of xacts by which cleanup is delayed

# - Standby Servers -

# These settings are ignored on a master server.

hot_standby = on # "off" disallows queries during recovery
# (change requires restart)
#max_standby_archive_delay = 30s # max delay before canceling queries
# when reading WAL from archive;
# -1 allows indefinite delay

```

Figura 6-29: Configuración del archivo de postgres. Fuente Autor

- Luego en la figura 6-30 y 6-31 se crea el archivo recovery.conf en el directorio de datos de Postgres del esclavo con el comando `.EDITOR /var/lib/postgresql/11/main/recovery.conf`, y se edita la línea "standby", esto para habilitar el modo standby, y con esto ya se puede iniciar nuevamente el servicio de Postgres `systemctl start postgresql@11-main.service`

```

ubuntu@ip-172-31-9-182: ~
GNU nano 2.5.3 File: /var/lib/postgresql/11/main/recovery.conf

standby_mode = 'on'
primary_conninfo = 'host=181.198.241.207 port=5432 user=replicator password=replicator_password'
primary_slot_name = 'replicator'
trigger_file = '/var/lib/postgresql/11/main/failover.trigger'

```

Figura 6-30: Creación del archivo recovery. Fuente Autor

```

ubuntu@ip-172-31-9-182:~$
ubuntu@ip-172-31-9-182:~$
ubuntu@ip-172-31-9-182:~$ sudo chown postgres:postgres /var/lib/postgresql/11/main/recovery.conf
ubuntu@ip-172-31-9-182:~$ systemctl start postgresql@11-main.service_

```

Figura 6-31: Iniciamos Postgres. Fuente Autor

6.4. Instalación y configuración de OpenVPN

Para poder realizar la instalación se debe utilizar el instalador:

openvpn-install-2.3.2-I006-x86_64.exe

Importante: No se debe instalar una versión superior ya que la instalación y configuración es distinta.

Luego se procede a ejecutar el instalador como: Administrador

6.4.1. Instalación de OpenVPN Server

- El proceso de instalación es de forma típica, es decir siguiente, siguiente, excepto en los pasos donde solicita seleccionar los componentes que se instalarán que deben ser seleccionados todos, como se indica en la figura 6-32.



Figura 6-32: Instalación OpenVpn. Fuente Autor

- En el servidor se debe marcar las dos opciones, las cuales son: OpenSSL Utilities y OpenVPN RSA Certificate Management Scripts, luego se instalará "Tap" que es la interfaz virtual que utiliza OpenVPN, figura 6-33.



Figura 6-33: Tap de OpenVpn. Fuente Autor

- Desmarcar el check de show Readme y finalizar, figura 6-34.

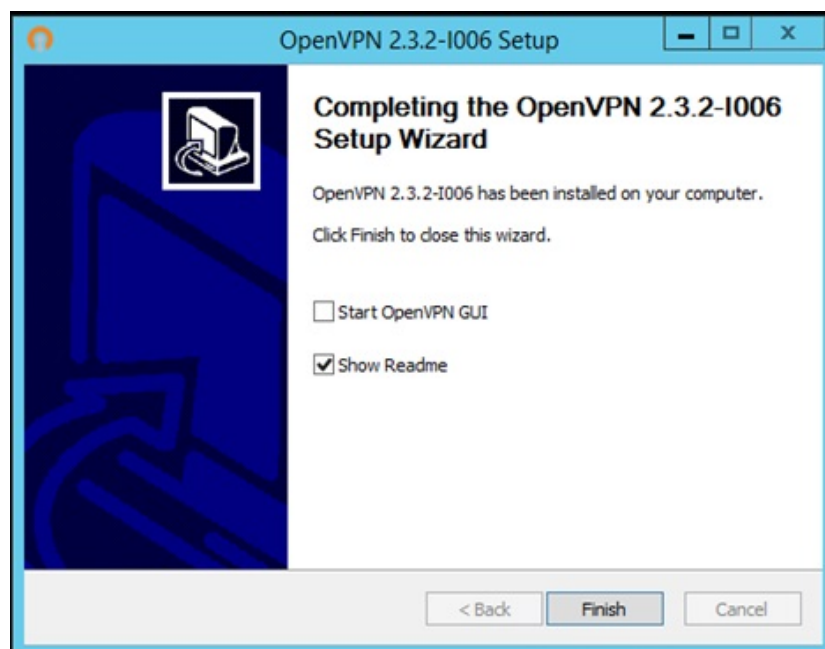


Figura 6-34: Finalización de instalación OpenVpn. Fuente Autor

6.4.2. Instalación de OpenVPN Client

- Para el proceso de instalación es de forma típica siguiente ¿siguiente, en donde solicita seleccionar los componentes que se instalarán se deben dejar los seleccionados por defecto, tal y como se indica en la figura 6-35.



Figura 6-35: Instalación OpenVpn Client. Fuente Autor

- En el servidor se debe dejar sin marcar el check en las dos opciones: OpenSSL Utilities y OpenVPN RSA Certificate Management Scripts, de igual manera se pedirá instalar Tap que es la interfaz virtual que utiliza OpenVPN, esto se indica en las figuras 6-36 y 6-37.

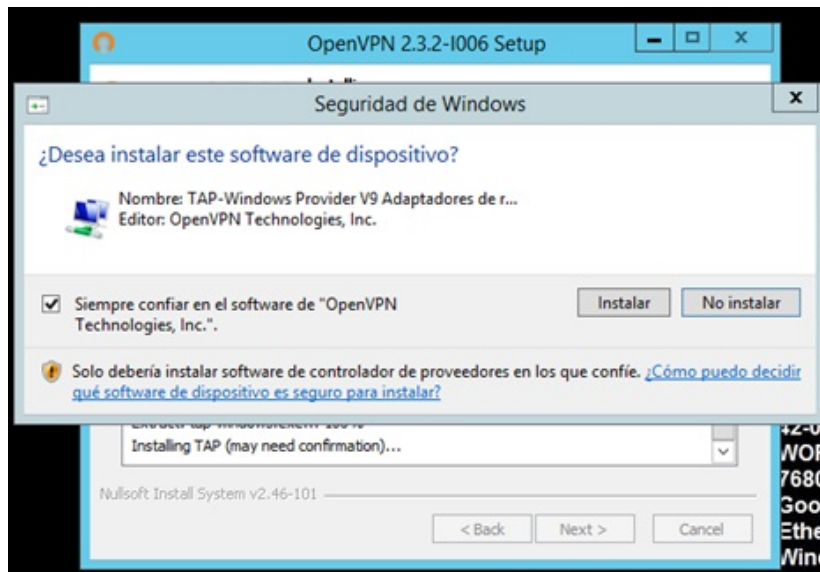


Figura 6-36: Tap de OpenVpn Client. Fuente Autor

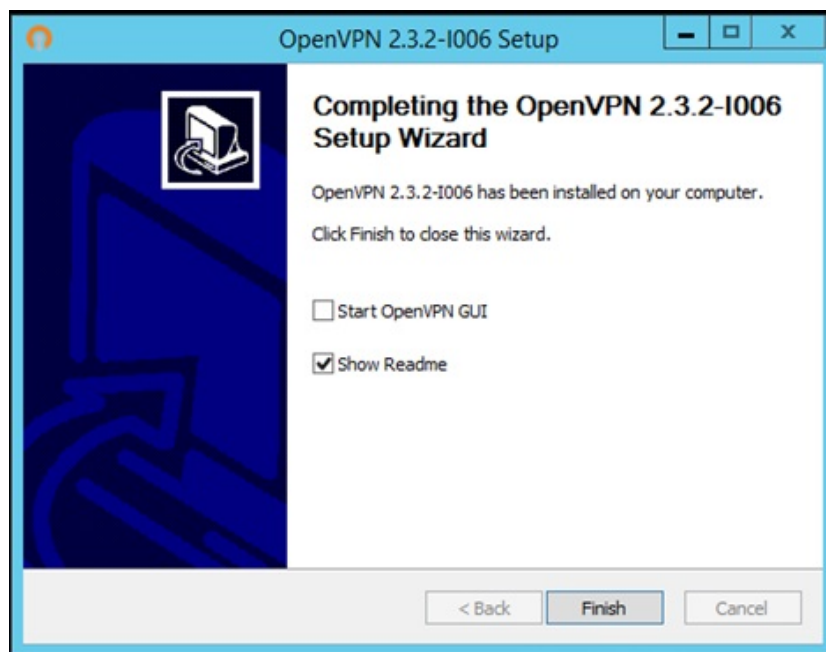


Figura 6-37: Finalización de instalacion OpenVpn Client. Fuente Autor

6.4.3. Configuración de OpenVPN

- Configuración de OpenVPN Server: Para esto se procede con un click derecho sobre el icono de OpenVPN y luego en ubicación del archivo; dar clic derecho al archivo ir a la pestaña compatibilidad y marcar el check de Ejecutar como administrador.

Ejecutar cmd como Administrador

ir a la ruta: *C:\ProgramFiles\OpenVPN\easy - rsa*

En la ruta indicada se debe buscar el archivo .cnf, una vez localizado crear una sentencia debe ser similar a la siguiente y procedemos a ingresar en el cmd con la ruta establecida:

set OPENSSL_CONF=C:\ProgramFiles\OpenVPN\easy - rsa\openssl - 1,0,0.cnf

No devuelve nada como resultado

init-config

Devuelve: 1 archivo(s) copiado(s).

notepad vars.bat

Cambiamos las siguientes líneas al final del archivo que se abre

```
set KEY_COUNTRY=EC
set KEY_PROVINCE=AZUAY
set KEY_CITY=CUENCA
set KEY_ORG=UPS
set KEY_EMAIL=tesis96@gmail.com
set KEY_CN=changeme
set KEY_NAME=changeme
set KEY_OU=changeme
set PKCS11_MODULE_PATH=changeme
set PKCS11_PIN=1234
```

Grabamos y salimos, **pero no se debe cerrar la terminal, ya que si se la cierra se debe volver a ejecutar el set con el archivo .cnf**

- Crear la llave Maestra: Para esto se procede a generar la llave maestra o también denominada ca, en la misma terminal y ruta ejecutamos:

```
- vars
- clean-all
- build-ca
```


- Al ejecutar se procede a configurar como se indica a continuación:

Loading 'screen' into random state - done

Generating a 1024 bit RSA private key

.....++++++

.....++++++

writing new private key to 'keys\ca.key'

*You are about to be asked to enter information that will be incorporated
into your certificate request.*

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [EC] :

State or Province Name (full name) [AZUAY] :

Locality Name (eg, city) [CUENCA] :

Organization Name (eg, company) [UPS] :

Organizational Unit Name (eg, section) [changeme] : UPS

Common Name (eg, your name or your server's hostname) [changeme] : llave – servidor

Name [changeme] : llave – servidor

Email Address [tesis96@gmail.com] :

- De igual manera, tampoco se cierra la terminal, y cuando se haya terminado el proceso en la carpeta keys, se dispondrá los siguientes archivos, detallados en la figura 6-38:

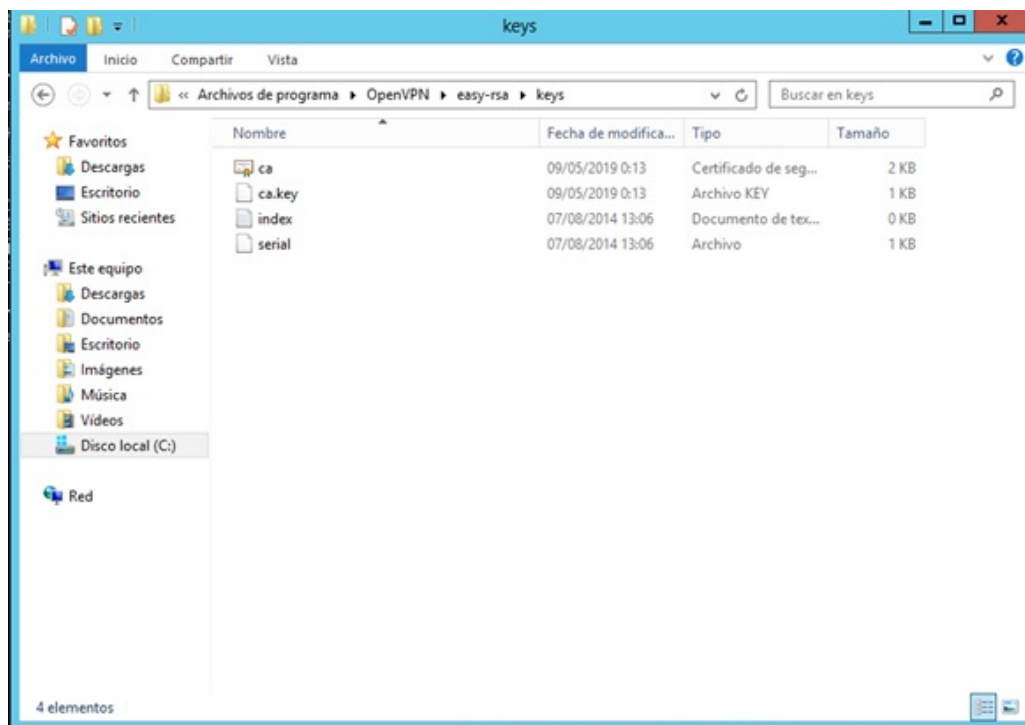


Figura 6-38: Creación de llave maestra. Fuente Autor

- Creación de la llave en el Servidor

En la terminal con la misma ruta se debe ejecutar:

build-key-server servidor-vpn

Al ingresar en la terminal se debería visualizar lo siguiente:

Loading 'screen' into random state - done

Generating a 1024 bit RSA private key

.....++++++

.++++++

writing new private key to 'keys\servidor - vpn.key'

— — — —

*You are about to be asked to enter information that will be incorporated
into your certificate request.*

*What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank*

*For some fields there will be a default value,
If you enter '.', the field will be left blank.*

CountryName(2 letter code)[EC] :
State or ProvinceName(full name)[AZUAY] :
LocalityName(eg, city)[CUENCA] :
OrganizationName(eg, company)[UPS] :
OrganizationalUnitName(eg, section)[changeme] : UPS
CommonName(eg, your name or your server's hostname)[changeme] : SERVIDOR - VPN
Name[changeme] :
EmailAddress[tesis96@gmail.com] :

Please enter the following 'extra' attributes
to be sent with your certificate request

A challenge password []:Hpurl

An optional company name []:

Using configuration from openssl-1.0.0.cnf

Loading 'screen' into random state - done

Check that the request matches the signature

Signature ok

The Subject's Distinguished Name is as follows

countryName :PRINTABLE:'EC'

stateOrProvinceName :PRINTABLE:'AZUAY'

localityName :PRINTABLE:'CUENCA'

organizationName :PRINTABLE:'UPS'

organizationalUnitName:PRINTABLE:'UPS'

commonName :PRINTABLE:'SERVIDOR-VPN'

name :PRINTABLE:'changeme'

emailAddress :IA5STRING:'tesis96@gmail.com'

Certificate is to be certified until May 6 00:32:21 2029 GMT (3650 days)

Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y

Write out database with 1 new entries

Data Base Updated

- Como en los pasos anteriores, en la figura **6-39** se indica que la terminal no se la cierra y cuando haya finalizado este proceso en la carpeta keys se tendrá los siguientes archivos:



- build-dh

El cual devuelve un resultado similar al siguiente:

```

Loading 'screen' into random state - done
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time

```

.....+.....+.....+.....
.....+.....
.....+.....+.....+.....
+.....
.....+.....
.....+.....
.....+.....
.....+.....
.....+.....+.....+.....
.....+.....

```

.....+.....+.....
.....+.....
.....+.....+.....+*+*+*

```

- Y en la carpeta keys se tendrá ya los siguientes archivos, figura 6-40:

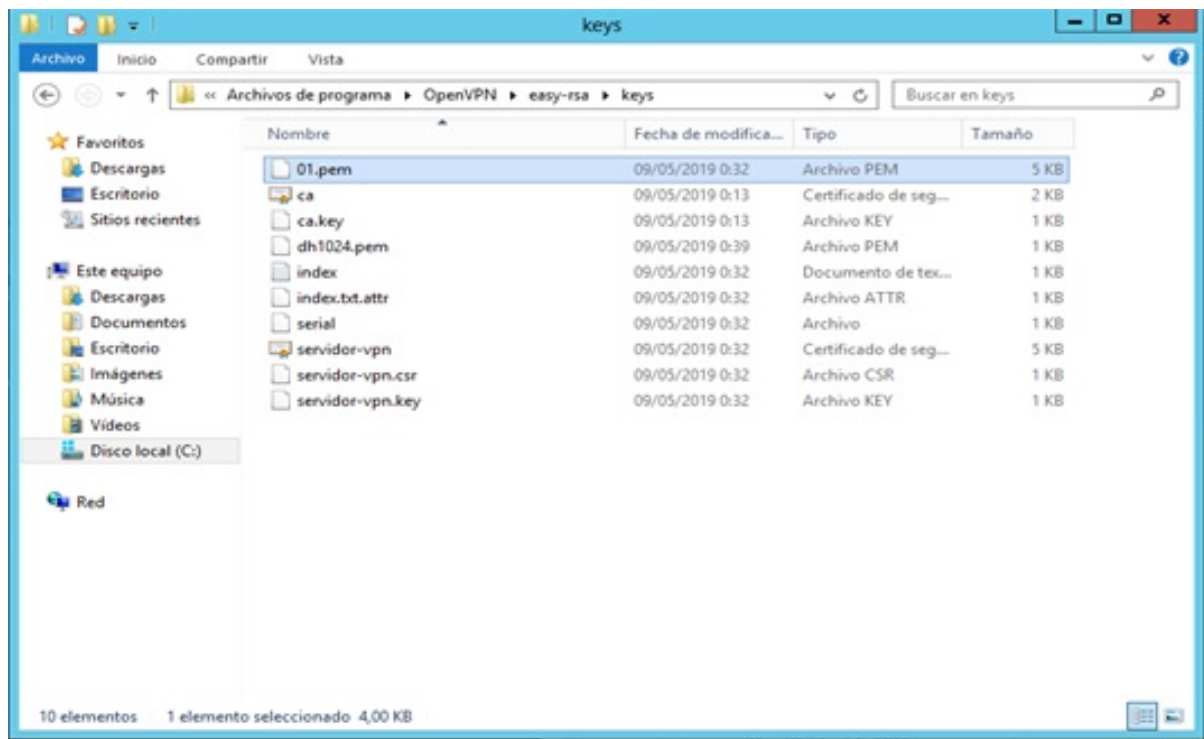


Figura 6-40: Generación de la encriptación. Fuente Autor

- Ahora en la figura 6-41 se debe copiar desde la carpeta keys a la carpeta config los siguientes archivos:

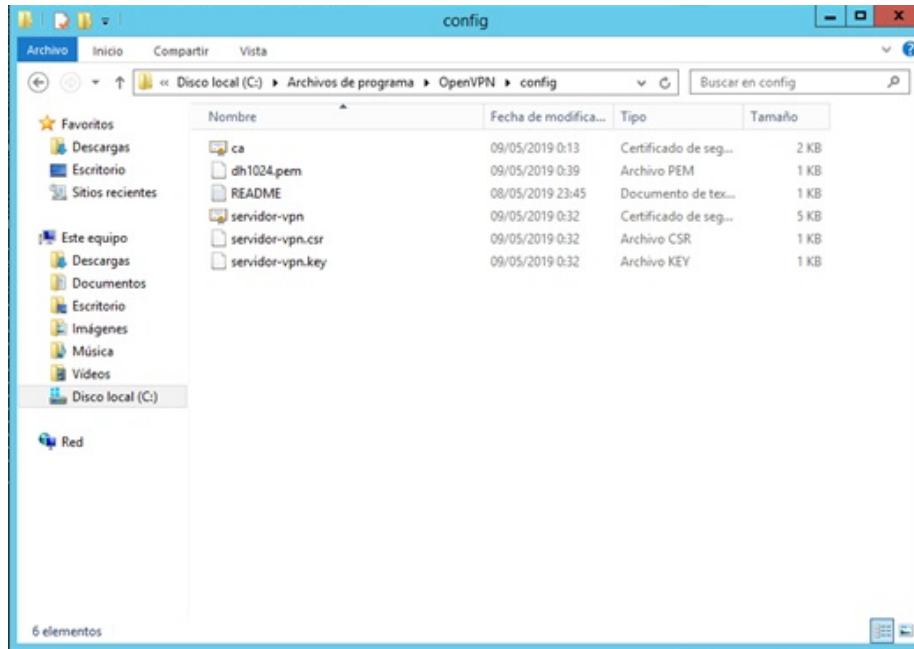


Figura 6-41: Copia de archivos. Fuente Autor

- En la figura 6-42, ir a la carpeta, con la ruta: *C:\Program Files\OpenVPN\sample-config*

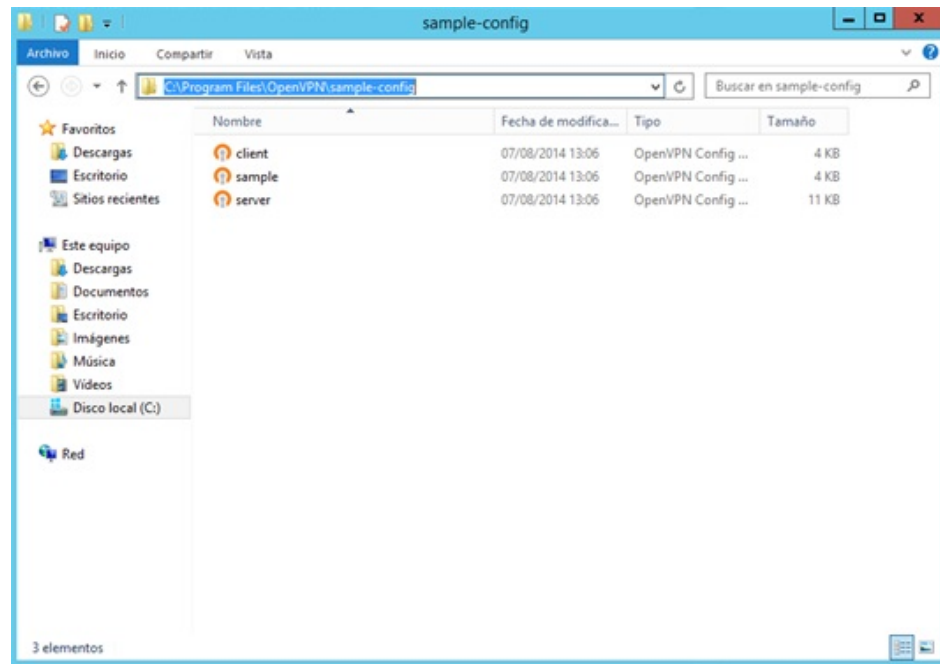


Figura 6-42: Copia de archivos. Fuente Autor

- Aquí se copia el archivo server a la carpeta config el archivo server, se indica en la figura 6-43:

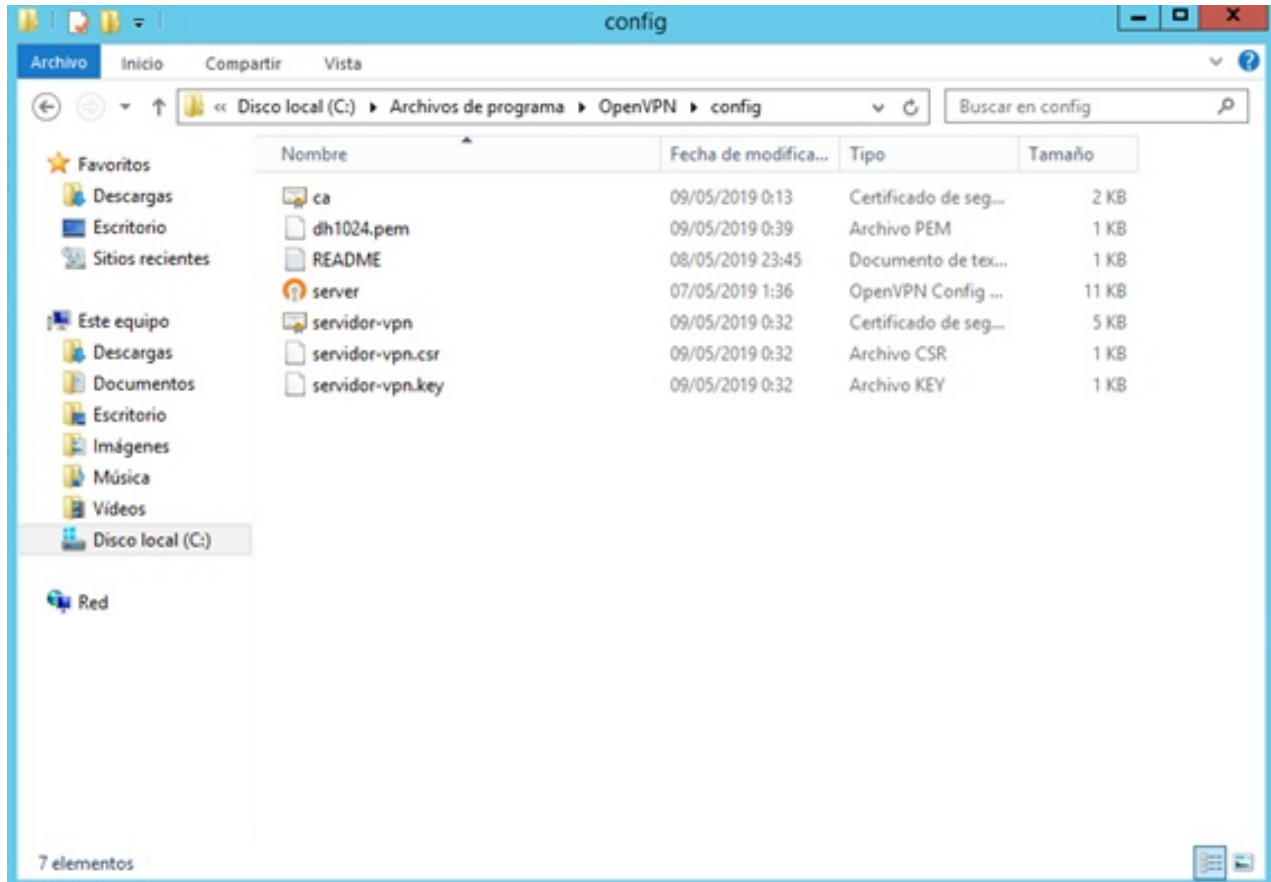


Figura 6-43: Copia de archivos. Fuente Autor

- En la figura 6-44 se procede a abrir las conexiones de red y a editar el nombre de la que dice TAP-Windows Adapter, se lo cambia por VPN o un nombre cualquiera, pero se debe tener presente el nombre que se le asigne.

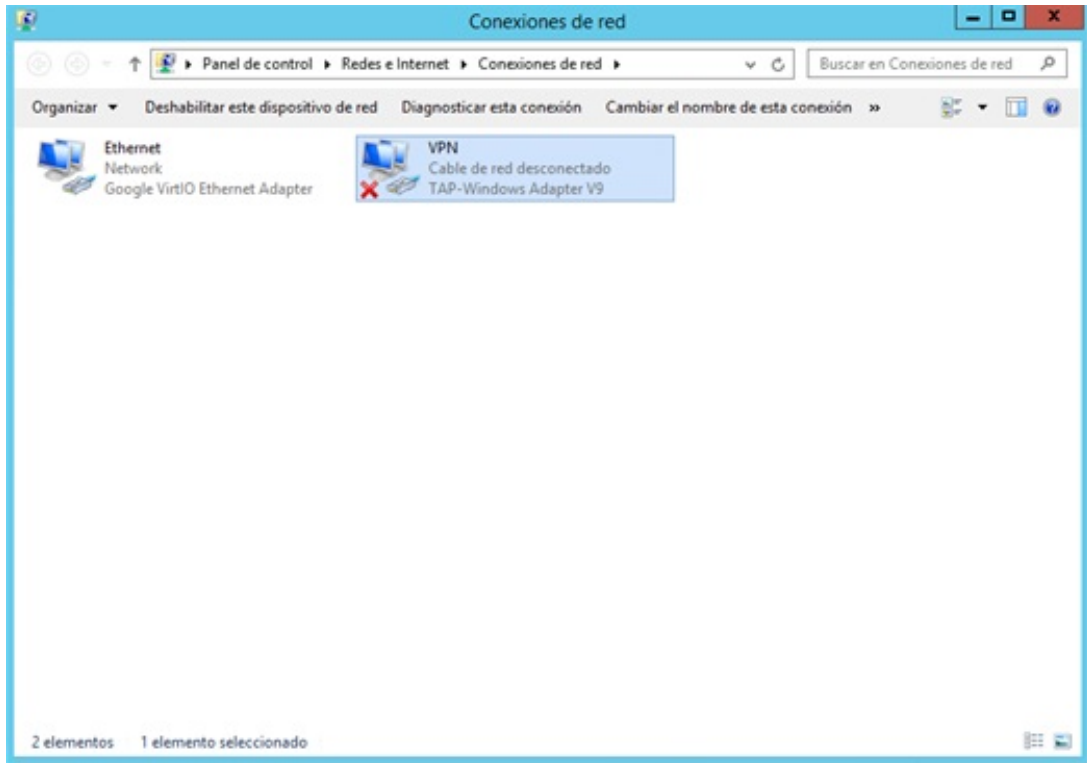


Figura 6-44: Conexiones de red. Fuente Autor

- Ahora se procede a editar el archivo server, para ello se ejecuta un editor como administrador y abrimos el archivo server en la ruta *C:\ProgramFiles\OpenVPN\config*, y se modificamos lo siguiente:

```
# Which local IP address should OpenVPN
# listen on? (optional)
;local a.b.c.d
```

```
# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one. You will need to
# open up this port on your firewall.
port 1194
```

```
# TCP or UDP server?
proto tcp
;proto udp

# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one. On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
dev-node VPN

# Any X509 key management system can be used.
# OpenVPN can also use a PKCS 12 formatted key file
# (see "pkcs12" directive in man page).
ca ca.crt
cert servidor-vpn.crt
key servidor-vpn.key This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
# openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh dh1024.pem

# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
```

```
# ethernet bridging. See the man page for more info.
server 10.10.10.0 255.255.255.0

# NIC interface. Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0. Finally we
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients. Leave this line commented
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
push route 10.158.0.0 255.255.240.0"
;push route 192.168.20.0 255.255.255.0"

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
client-config-dir ccd
route 192.168.1.0 255.255.255.0

# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.
# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# Then add this line to ccd/Thelonious:
# ifconfig-push 10.9.0.1 10.9.0.2

# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
```

client-to-client

```
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120
```

```
# Generate with:
# openvpn --genkey --secret ta.key
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 This file is secret
```

```
# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nobody
```

```
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun
```

- Configuración de reenvío en Windows (Bit de forwarding): Para esto se hace lo siguiente:
Ejecutar *Regedit*

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
\services\Tcpip\Parameters\IPEnableRouter
```

Doble clic en él, y cambiamos el valor de 0 a 1, para posterior darle en Aceptar.

- Configuración Regla de entrada en el Firewall: En la figura **6-45** se detalla que se ubica en:
Panel de control\Sistemapseguridad\FirewalldeWindows
ConfiguraciónAvanzada – ReglasdeEntrada – NuevaRegla

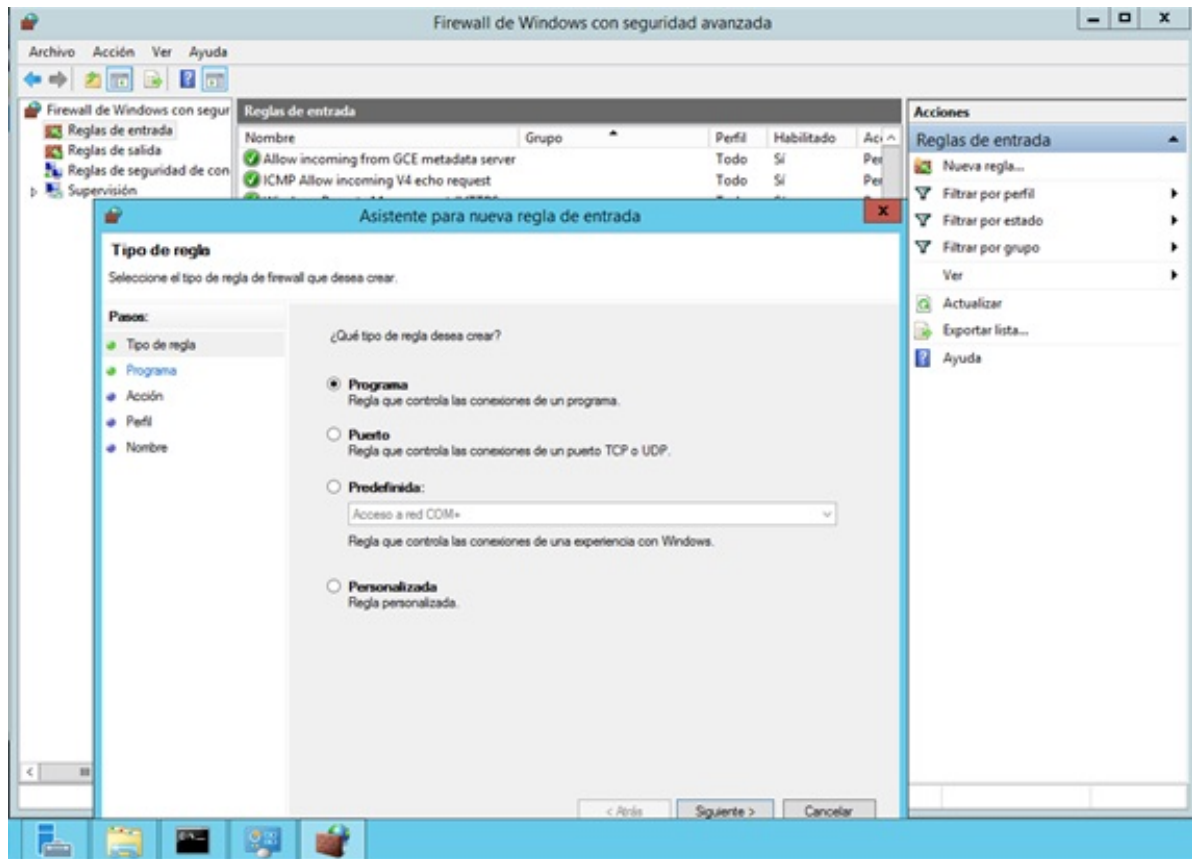


Figura 6-45: Configuración del Firewall. Fuente Autor

- Se selecciona el puerto, puertos locales específicos **6-46**.

Asistente para nueva regla de entrada

Protocolo y puertos

Especifique los puertos y protocolos a los que se aplica esta regla.

Pasos:

- Tipo de regla
- Protocolo y puertos**
- Acción
- Perfil
- Nombre

¿Se aplica esta regla a TCP o UDP?

☒ TCP

☐ UDP

¿Se aplica esta regla a todos los puertos locales o a unos puertos locales específicos?

☐ Todos los puertos locales

☒ Puertos locales específicos:

1194

Ejemplo: 80, 443, 5000-5010

< Atrás Siguiente > Cancelar

Figura 6-46: Selección de puerto. Fuente Autor

- En la figura 6-47 y 6-48 se escoge TCP y se escribe el puerto por defecto de OpenVPN que es el 1194, se da en Permitir Conexión - Siguiente y además se establece los perfiles.

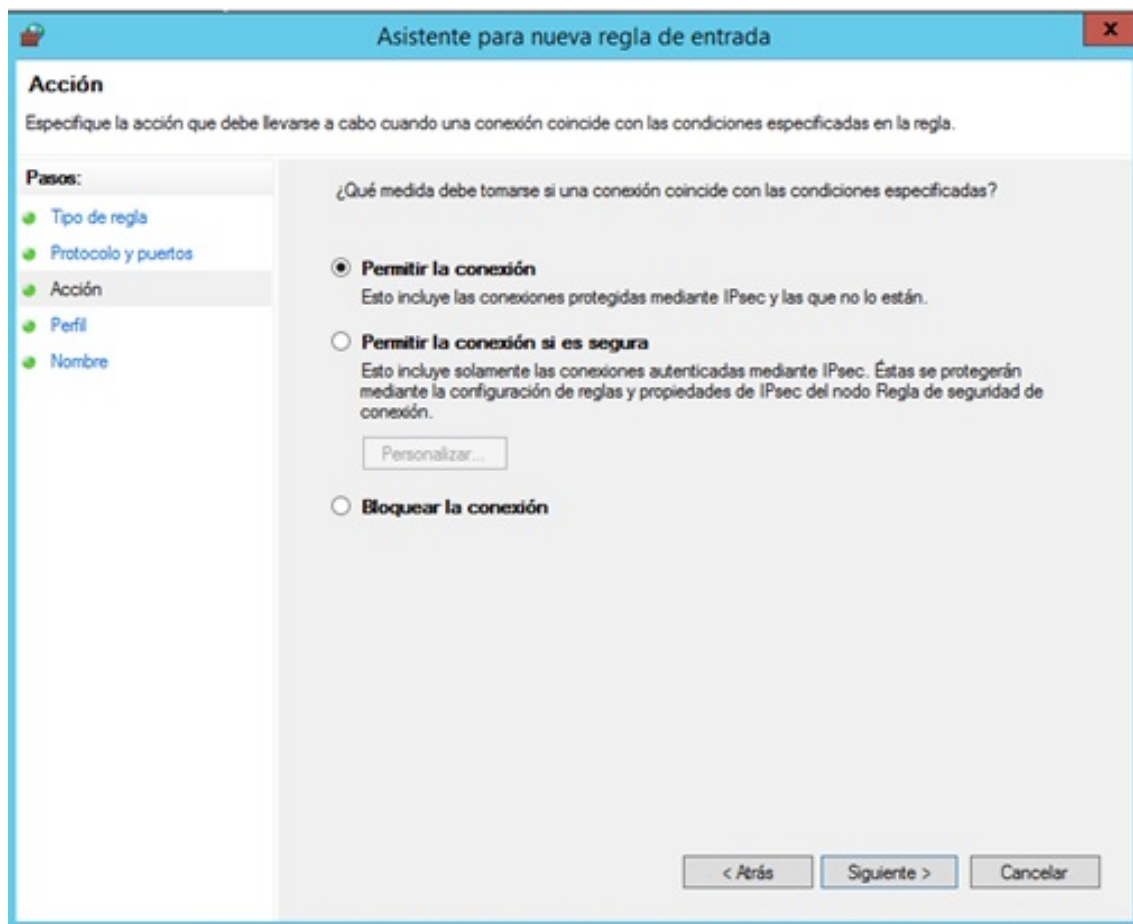


Figura 6-47: Se permite la conexión. Fuente Autor

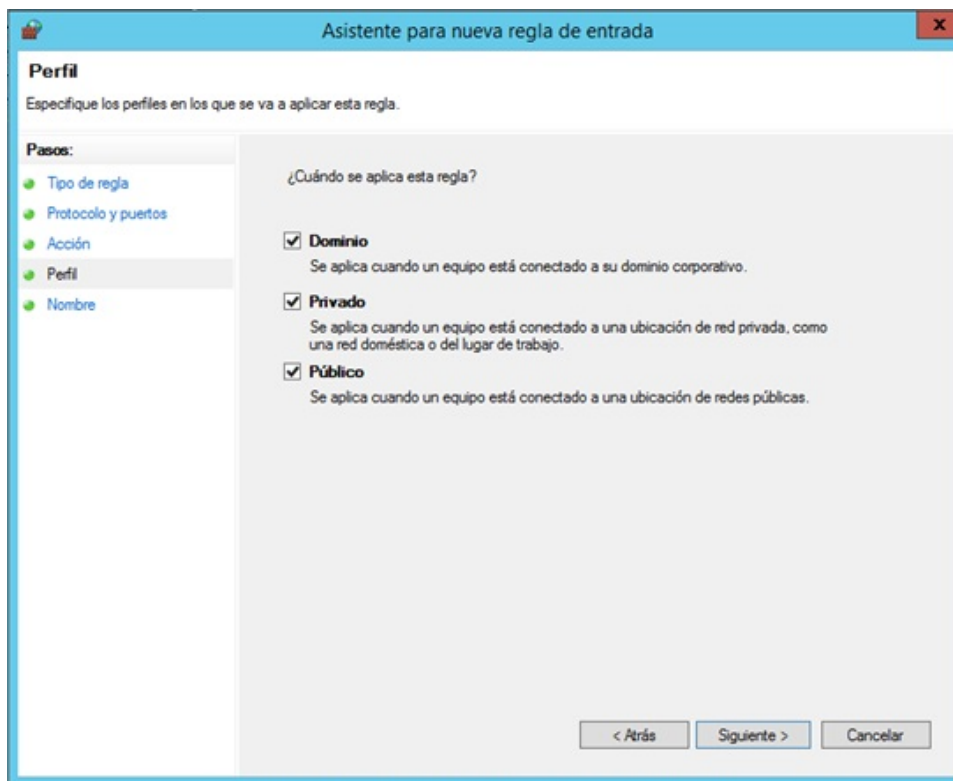


Figura 6-48: Se escoge los perfiles. Fuente Autor

- Se coloca un nombre y luego en Finalizar, como se observa en la figura 6-49.

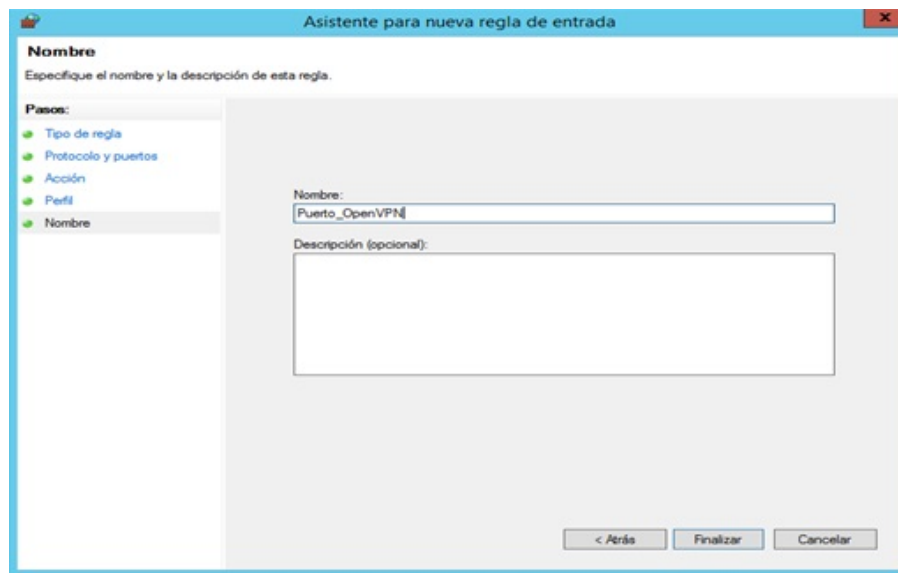


Figura 6-49: Se escoge un nombre. Fuente Autor

- Creación de certificados para los usuarios: En la misma terminal y en la misma ruta ejecutar:

vars

build-key usuario01

Luego se ingresan los valores, a continuación se presenta como se deben ingresar los valores:

```

Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
....++++++
.....++++++
writing new private key to 'keys\usuario01.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
CountryName(2 letter code)[EC] :
State or Province Name(full name)[AZUAY] :
Locality Name(eg, city)[CUENCA] :
Organization Name(eg, company)[UPS] :
Organizational Unit Name(eg, section)[changeme] : UPS
Common Name(eg, your name or your server's hostname)[changeme] : usuario01
Name[changeme] : usuario01
Email Address[tesis96@gmail.com] :

```

Please enter the following 'extra' attributes
to be sent with your certificate request

A challenge password []:Hpurl

An optional company name []:

Using configuration from openssl-1.0.0.cnf

Loading 'screen' into random state - done

Check that the request matches the signature

Signature ok

The Subject's Distinguished Name is as follows

countryName :PRINTABLE:'EC'

stateOrProvinceName :PRINTABLE:'AZUAY'

```

localityName :PRINTABLE:'CUENCA'
organizationName :PRINTABLE:'UPS'
organizationalUnitName:PRINTABLE:'UPS'
commonName :PRINTABLE:'usuario01'
name :PRINTABLE:'usuario01'
emailAddress :IA5STRING:'tesis96@gmail.com'
' Certificate is to be certified until May 6 01:26:28 2029 GMT (3650 days)
Sign the certificate? [y/n]:y

```

```

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries

```

- Cuando finalice el proceso se tendrá los siguientes archivos como se observa en la figura 6-50:

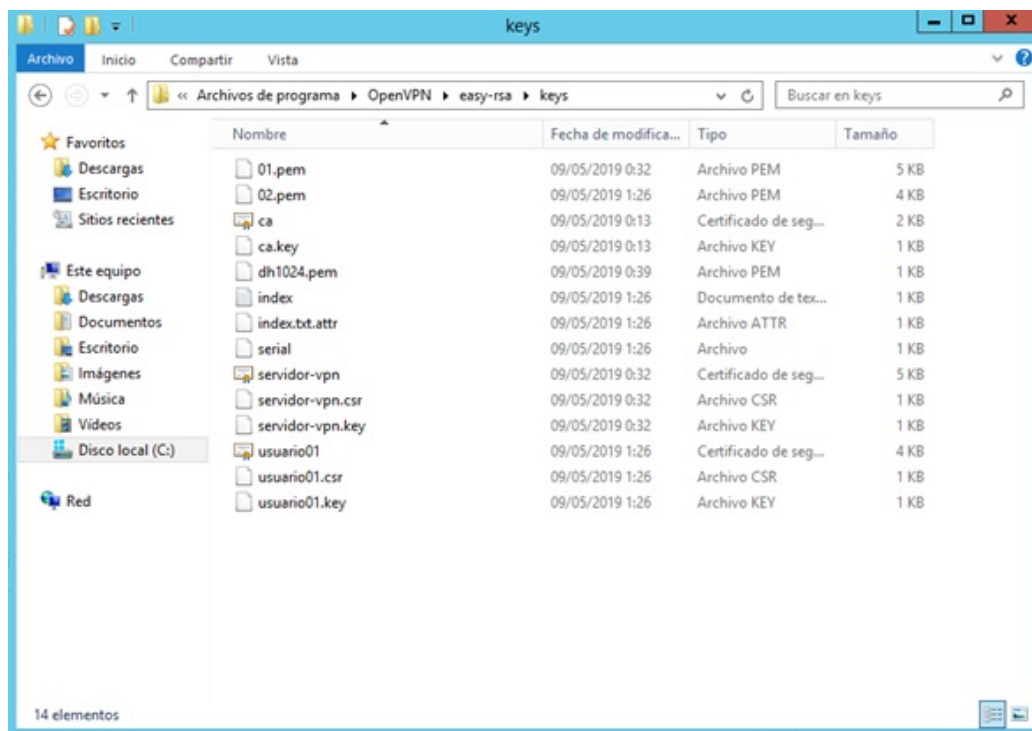


Figura 6-50: Certificados. Fuente Autor

- En la figura 6-51 se procede a crear una carpeta y desde la carpeta keys se copia a una nueva carpeta los siguientes archivos, estos archivos serán utilizados para configurar el usuario.

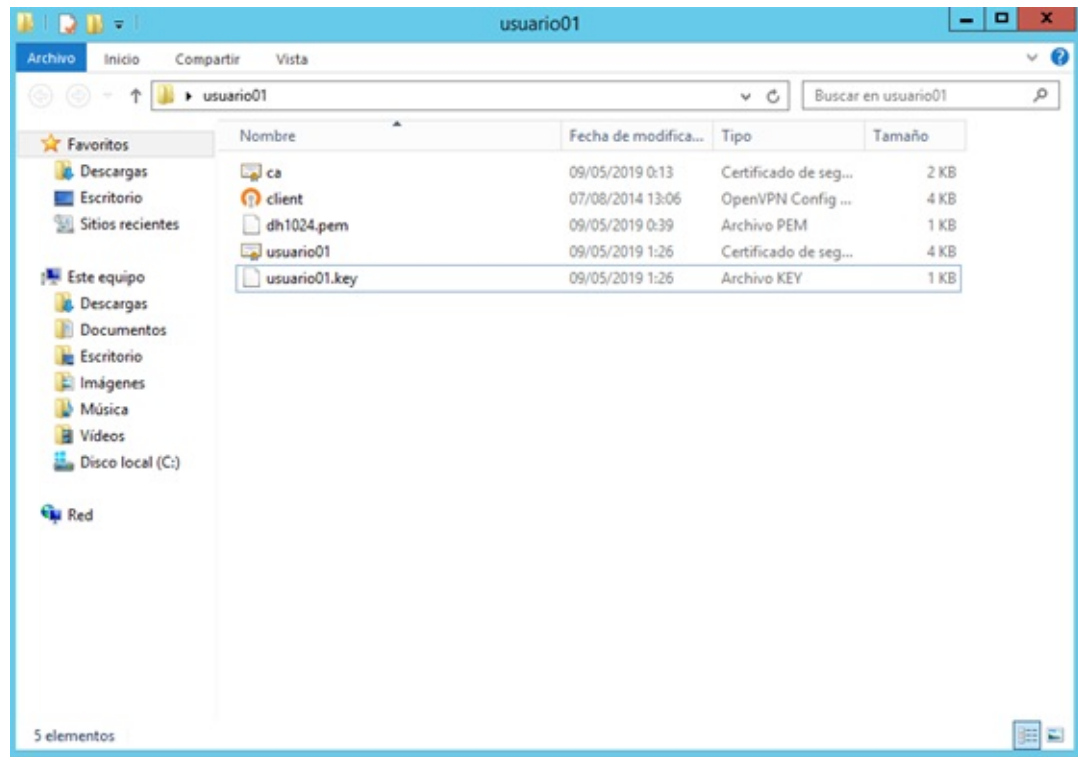


Figura 6-51: Configuración de usuarios. Fuente Autor

- Configuración de OpenVPN Client: Para esto se procede a abrir la carpeta de instalación de OpenVPN e ingresar a la ruta, como se mira en la figura 6-52:

C:\ProgramFiles\OpenVPN\config

Pegar los archivos generados en el servidor y destinados para este cliente.

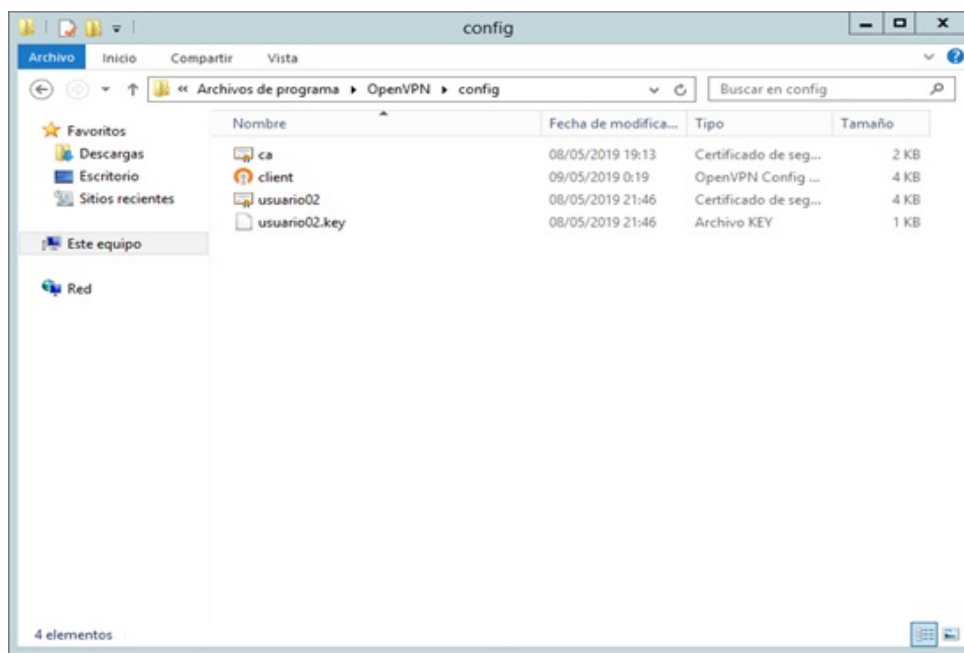


Figura 6-52: Configuración de OpenVPN Client. Fuente Autor

- Una vez pegados los archivos con el notepad abierto como administrador, se procede a modificar los siguientes parámetros:

```
# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client
```

```
# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one. On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node mytap
```

```
# Are we connecting to a TCP or
# UDP server? Use the same setting as
# on the server.
proto tcp
;proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
# Aquí remplazamos la ip 35.199.124.67 por nuestra dirección IP
remote 35.199.124.67 1194
;remote my-server-2 1194

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server. Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nobody

# Try to preserve some state across restarts.
persist-key
persist-tun

# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here. See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry  retry on connection failures
```

```
;http-proxy [proxy server] [proxy port ]
```

```
# file can be used for all clients.
```

```
ca ca.crt
```

```
cert usuario02.crt
```

```
key usuario02.key
```

```
# If a tls-auth key is used on the server
```

```
# then every client must also have the key.
```

```
;tls-auth ta.key 1
```

■ OpenVPN site to site

Para esto se procede a editar el archivo como administrador la ruta *C:\ProgramFiles\OpenVPN\config, donde*

client-config-dir ccd route 192.168.1.0 255.255.255.0

Aquí se tiene que crear la carpeta llamada ccd en : *C:\ProgramFiles\OpenVPN\config*

Aquí, como se observa en la figura **6-53**, adentro se crea un archivo sin extensión con el nombre del usuario, necesariamente debe ser el mismo nombre.

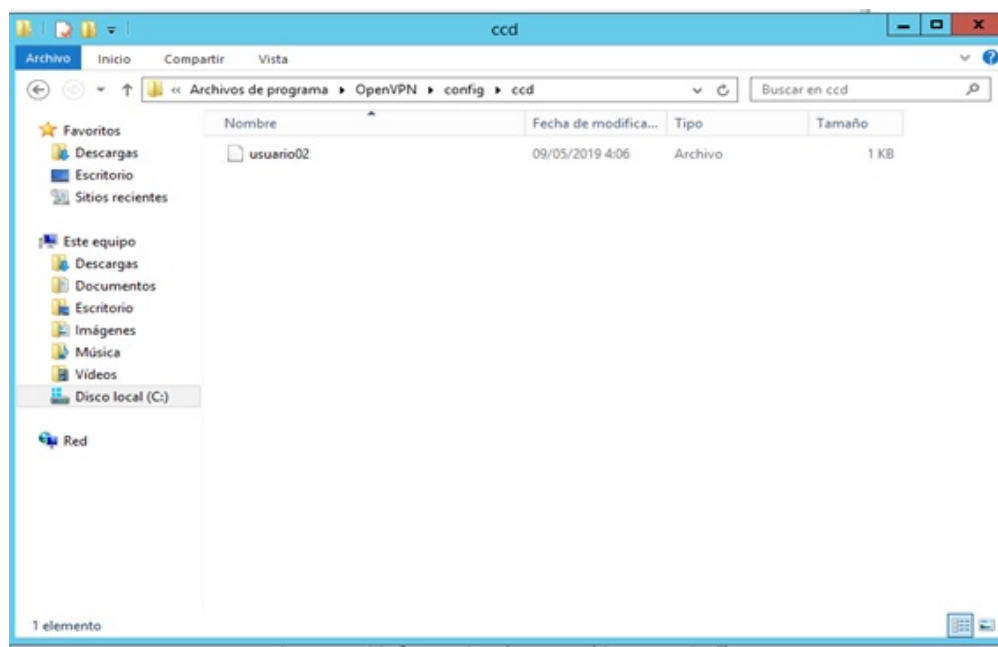


Figura 6-53: OpenVpn site to site. Fuente Autor

- Dentro del archivo se debe tener el siguiente contenido en base a la red:

iroute 192.168.1.0 255.255.255.0

- Se reinicia la máquina y se vuelve a probar y ya se dispondrá de conexión a las LAN de los extremos de la VPN.
- Configuración de Replicación en Windows Server 2012

Para la configuración en Windows server 2012 r2 se debe considerar lo siguiente:

Una vez que ya se ha configurado correctamente el directorio Activo y se ha configurado el principal como un nuevo bosque, entonces en el segundo server se instalara los servicios de active directory, pero cuando se promueva el servidor a controlador de dominio, se lo agregará a uno ya existente, en este caso dmr.

6.5. Azure

- Configuración de regla de entrada para la conexión de OpenVPN que utiliza el puerto 1194 y posterior se procede a ingresar a Maquinas virtuales, como se indica en la figura 6-54.

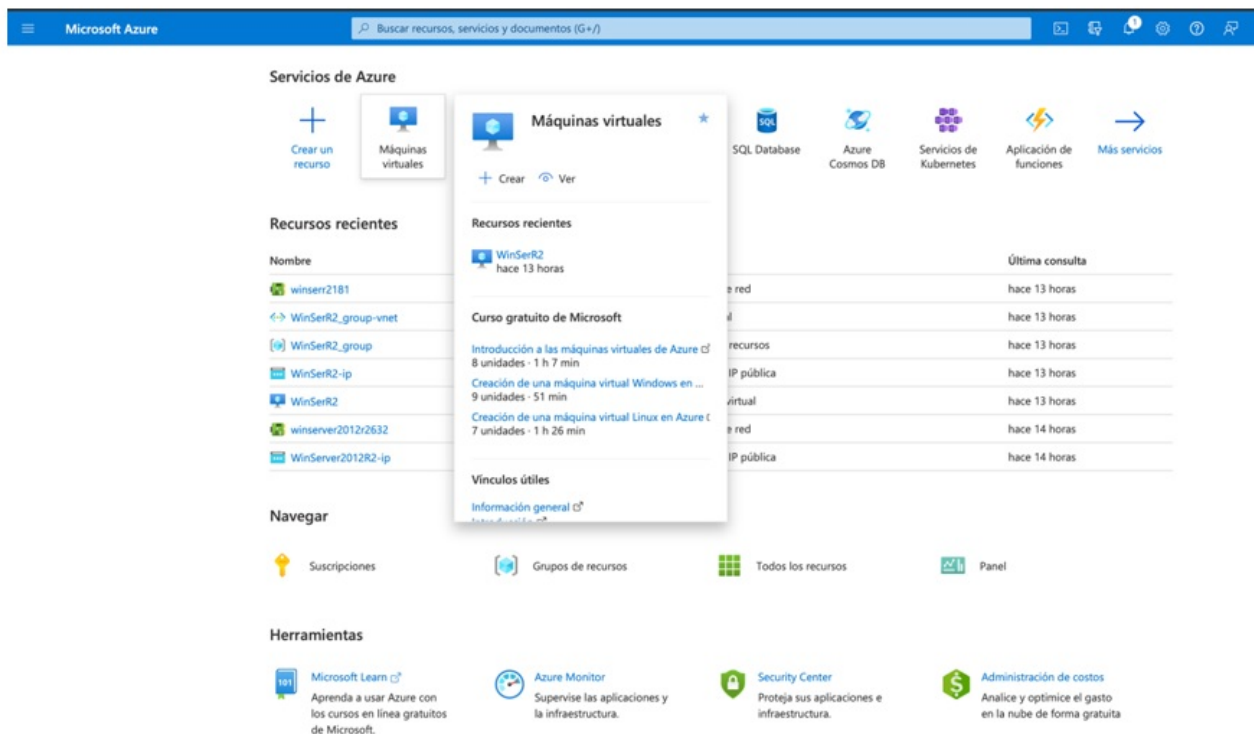


Figura 6-54: Máquinas virtuales de Azure. Fuente Autor

- Se selecciona la máquina virtual, figura 6-55

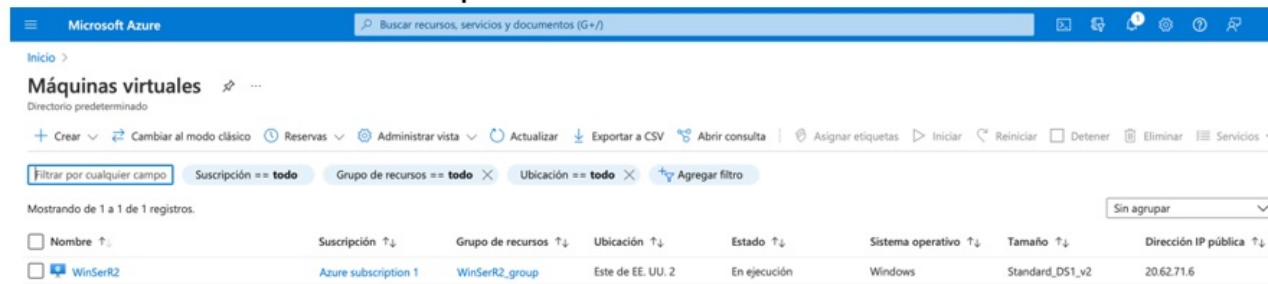


Figura 6-55: Listado de máquinas virtuales. Fuente Autor

- Se escoge el apartado de redes, figura 6-56

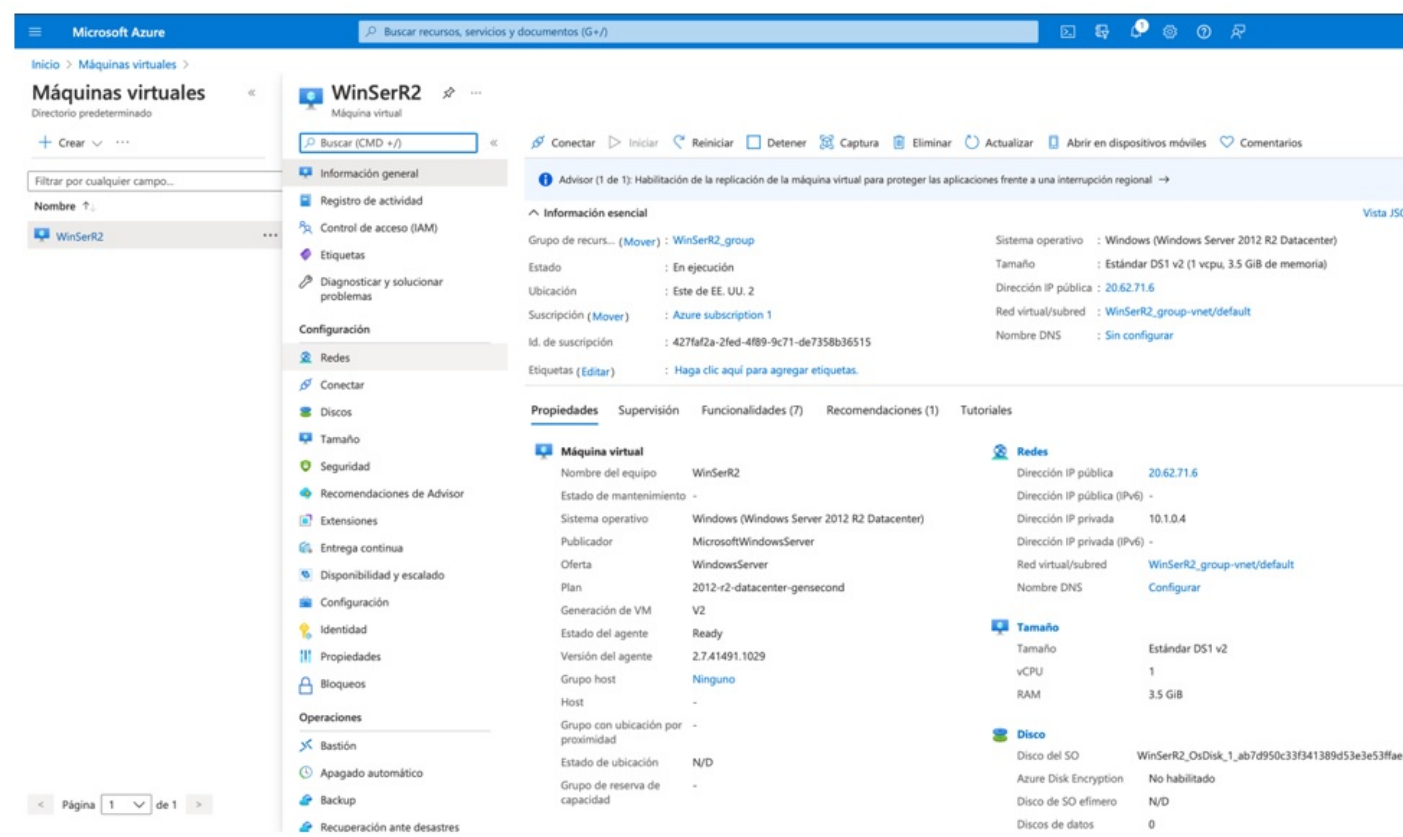
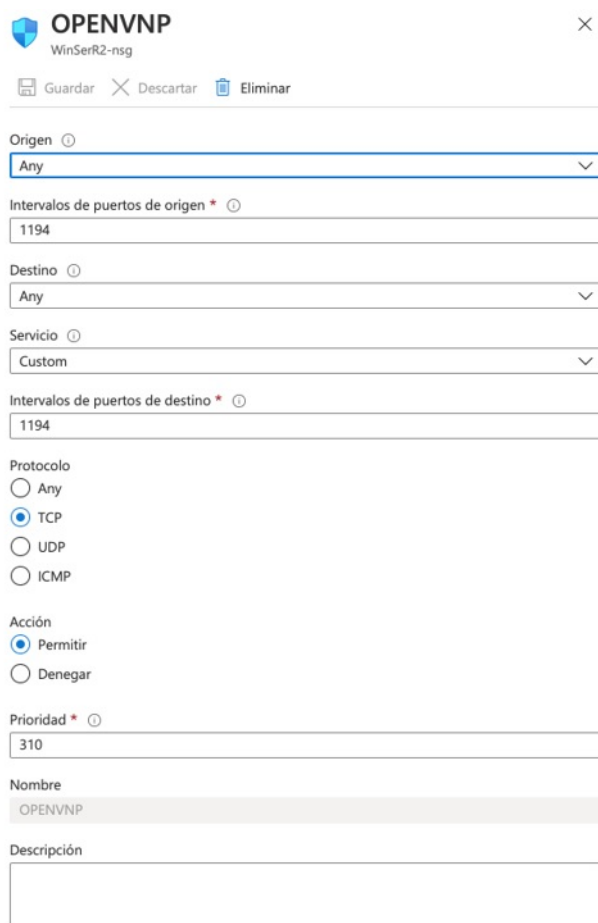


Figura 6-56: Apartado de redes. Fuente Autor

- Se agrega una nueva regla de entrada al puerto y se guarda, se observa en la figura 6-57.



OPENVNP WinSerR2-nsg

Guardar Descartar Eliminar

Origen ⓘ
Any

Intervalos de puertos de origen * ⓘ
1194

Destino ⓘ
Any

Servicio ⓘ
Custom

Intervalos de puertos de destino * ⓘ
1194

Protocolo
☐ Any
☒ TCP
☐ UDP
☐ ICMP

Acción
☒ Permitir
☐ Denegar

Prioridad * ⓘ
310

Nombre
OPENVNP

Descripción

Figura 6-57: Regla de entrada. Fuente Autor

Bibliografía

- [1] Enrutamiento vpn de sitio a sitio explicado en detalle, 2020.
- [2] Y. Sung D. Maltz S. Rao K. Sripanidkulchai M. Hajjat, X. Sun. Cloudward bound. 2010.
- [3] P. Ray. A survey of iot cloud platforms. 2016.
- [4] M. Vukoli. The byzantine empire in the intercloud. 2010.
- [5] S. R. Zeebaree Z. N. Rashid and A. Shengul. Design and analysis of proposed remote controlling distributed parallel computing system over the cloud. 2019.
- [6] Thomas Schenke Hong, Jiangshui Dreibholz. An overview of multicloud computing. 2019.
- [7] Marko. Vukolic. The byzantine empire in the intercloud. 2010.
- [8] Rean G. Anthony J. Randy K. Andrew K. Gunho L. David P. Ariel R. Ion S Armando, F. Above the clouds: A berkeley view of cloud computing. tech. rep. 2009.
- [9] Schahram Dustdar. Cloud computing. 2016.
- [10] Odina Orozco, Igor; Jacobs. La nueva era de los negocios: Computación en la nube. 2016.
- [11] Tim G. Peter, M. The nist definition of cloud computing. 2011.
- [12] Sarwar Goswami Parijat. Islam, Mohammad Manzurul Morshed. Survey on its limitations and potential solutions. 2013.
- [13] Centro Telecom. ¿qué es multicloud y por qué sí usarlo?, 2019.
- [14] RedHat. ¿qué es una nube pública?
- [15] CloudFlare. ¿qué es una nube pública?
- [16] Einatec. Nube pública: qué es, ventajas y desventajas.
- [17] hostingred. Los tipos de nube- servidores en la nube.

- [18] Luis Joyanes Aguilar. Computación en la nube e innovaciones tecnológicas. 2018.
- [19] Wiley Brand. Cloud management for dummies. 2018.
- [20] Tomás Cabacas. ¿qué es la nube híbrida y por qué no paras de oír hablar de ella?, 2020.
- [21] Ramón Miralles. Revista de internet, derecho y política. 2011.
- [22] Azahara Benito Carrillo. Los 3 tipos de servicios que existen dentro del cloud computing en las empresas, 2019.
- [23] Juan Aranda, Erwin J Sacoto-Cabrera, Daniel Haro-Mendoza, and Fabián Astudillo-Salinas. Redes 5g: una revisión desde las perspectivas de arquitectura, modelos de negocio, ciberseguridad y desarrollos de investigación. *Revista Digital Novasinergia*, 4(1):6–41, 2021.
- [24] Erwin Sacoto Cabrera. *Análisis basado en teoría de juegos de modelos de negocio de operadores móviles virtuales en redes 4G y 5G*. PhD thesis, Universitat Politècnica de València, 2021.
- [25] Erwin J Sacoto-Cabrera, Luis Guijarro, Jose R Vidal, and Vicent Pla. Economic feasibility of virtual operators in 5g via network slicing. *Future Generation Computer Systems*, 109:172–187, 2020.
- [26] Erwin J Sacoto-Cabrera, Angel Sanchis-Cano, Luis Guijarro, José Ramón Vidal, and Vicent Pla. Strategic interaction between operators in the context of spectrum sharing for 5g networks. *Wireless Communications and Mobile Computing*, 2018.
- [27] Erwin J Sacoto-Cabrera, Gabriel León-Paredes, and Walter Verdugo-Romero. Lorawan: Application of nonlinear optimization to base stations location. In *Communication, Smart Technologies and Innovation for Society*, pages 515–524. Springer, 2022.
- [28] Erwin J Sacoto Cabrera, Sonia Palaguachi, Gabriel A León-Paredes, Pablo L Gallegos-Segovia, and Omar G Bravo-Quezada. Industrial communication based on mqtt and modbus communication applied in a meteorological network. In *The International Conference on Advances in Emerging Trends and Technologies*, pages 29–41. Springer, 2020.
- [29] Victor Vimos and Erwin J Sacoto Cabrera. Results of the implementation of a sensor network based on arduino devices and multiplatform applications using the standard opc ua. *IEEE Latin America Transactions*, 16(9):2496–2502, 2018.
- [30] Erwin Sacoto-Cabrera, Jorge Rodriguez-Bustamante, Pablo Gallegos-Segovia, Gabriela Arevalo-Quishpi, and Gabriel León-Paredes. Internet of things: Informatic system for metering with communications mqtt over gprs for smart meters. In *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, pages 1–6. IEEE, 2017.

- [31] V Vimos, E Sacoto, and DX Morales. Conceptual architecture definition: Implementation of a network sensor using arduino devices and multiplatform applications through opc ua. In *2016 IEEE International Conference on Automatica (ICA-ACCA)*, pages 1–5. IEEE, 2016.
- [32] OpenNebula. Opennebula overview. urlhttps://docs.opennebula.io/6.0/overview/opennebula_conceptual_architecture.html, 2020.
- [33] Sourabh Kumawat³ Sunil Kumar Jangir² Rakesh Kumar¹, Laveena Adwani². Opennebula: Open source iaas cloud computing software platforms. 2014.
- [34] Eucalyptus. Eucalyptus overview. urlhttps://docs.eucalyptus.cloud/eucalyptus/5/user_guide/overview.html, 2020.
- [35] OpenStack. What is openstack? url<https://www.openstack.org/software/>, 2020.
- [36] OpenStack. Vision for openstack clouds. urlhttps://governance.openstack.org/tc/reference/technical/vision.html?_ga=2,245419460,1341316551,1623293841-2007073677,1623293841, 2020.
- [37] Ignacio Ordorica. Qué es google cloud y para qué sirve. url<https://www.incentro.com/es-es/blog/stories/que-es-google-cloud-platform/>, 2020.
- [38] Mandeep Kumar. Google cloud platform: A powerful big data. 2016.
- [39] Javier Arbiol. ¿qué es microsoft azure y para qué sirve? por qué elegir la plataforma cloud de microsoft. url <https://www.hiberus.com/crecemos-contigo/azure-la-plataforma-cloud-de-microsoft/>, 2019.
- [40] Oracle. Documentación de oracle cloud infrastructure. url <https://docs.oracle.com/es-ww/iaas/Content/GSG/Concepts/baremetalintro.htm>, 2019.
- [41] David Núñez Escobedo Jesús María Rodríguez. ciberseguridad, seguridad de la información y privacidad. 2020.
- [42] José Miguel Linares Sillero. Introducción a amazon web services. url <https://www2.deloitte.com/es/es/pages/technology/articles/introduccion-a-amazon-web-services.html>, 2017.
- [43] Mónica Cecilia Gallegos Varela. *Introducción a los Servicios Web*. 2020.
- [44] W3C Consortium. Web services architecture. 2008.
- [45] Angel Robledano. Qué es mysql: Características y ventajas. url<https://openwebinars.net/blog/que-es-mysql/>, 2019.
- [46] RUIZ BERMEO LENIN ALEXANDER. Aplicación web para la gestión de inventarios, pedidos y reservas de productos para la empresa ecgrub. url<https://dspace.uniandes.edu.ec/bitstream/123456789/8522/1/TUAEXCOMSIS020-2018.pdf>, 2018.

- [47] Steve Suehring. *MySQL Bible*. 2002.
- [48] Jonathan Morin. *Network Virtualization for Dummies*. 2019.