

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS



**Prototipo centrado en una Infraestructura de**

**Nube Comunitaria Geo-distribuida**

PRESENTADO POR:

**Angulo García, Roxana Mercedes**

**Hernández Cortez, Jairo Antonio**

**Novoa Bernabé, Valeria Michelle**

PARA OPTAR AL TÍTULO DE:

**INGENIERO DE SISTEMAS INFORMÁTICOS**

CIUDAD UNIVERSITARIA, NOVIEMBRE DE 2023

**UNIVERSIDAD DE EL SALVADOR**

**RECTOR:**

**MSC. JUAN ROSA QUINTANILLA**

**SECRETARIO GENERAL:**

**LIC. PEDRO ROSALÍO ESCOBAR CASTANEDA**

**FACULTAD DE INGENIERÍA Y ARQUITECTURA**

**DECANO:**

**ING. LUIS SALVADOR BARRERA MANCÍA**

**SECRETARIO:**

**ARQ. RAUL ALEXANDER FABIAN ORELLANA**

**ESCUELA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS**

**DIRECTOR:**

**ING. CÉSAR AUGUSTO GONZÁLEZ**

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

Trabajo de Graduación previo a la opción al Grado de:

**INGENIERO DE SISTEMAS INFORMÁTICOS**

Título:

**Prototipo centrado en infraestructura de nube comunitaria geo-distribuida**

Presentado por:

**Angulo García, Roxana Mercedes**

**Hernández Cortez, Jairo Antonio**

**Novoa Bernabé, Valeria Michelle**

Trabajo de Graduación Aprobado por:

Docente Asesor:

**MSc. DAMIÁN MORALES**

SAN SALVADOR, NOVIEMBRE DE 2023

Trabajo de Graduación Aprobado por:

Docente Asesor:

**MSc. DAMIÁN MORALES**

## **Agradecimientos**

A Dios por caminar de la mano conmigo y ayudarme a superar obstáculos y llegar a culminar esta etapa de mi vida.

A mis padres por su apoyo constante en esta aventura y a lo largo de mi vida, por siempre estar presentes, tenerme paciencia y brindarme su cariño incondicional y valiosos consejos.

A mis mejores amigas por darme una mano siempre que lo he necesitado, sus palabras de aliento, tiempo y cariño a pesar de la vida tan apresurada y ocupada que todos vivimos hoy en día.

A los padres de mis mejores amigas por recibirme con brazos abiertos cada vez que teníamos trabajos en grupo y debíamos reunirnos.

A los docentes que me han animado a seguir adelante y que han sabido transmitir sus conocimientos a lo largo de la carrera.

### **Roxana Angulo**

Quiero expresar mi profundo agradecimiento a Dios, quien ha sido mi guía y fortaleza a lo largo de este arduo pero gratificante viaje académico. Su infinita sabiduría y amor han sido la luz que ha iluminado mi camino, dándome la fuerza necesaria para superar los desafíos y alcanzar este logro.

A mis padres, Israel Hernández y Rosa Del Carmen Cortez, les debo mi más sincero agradecimiento. Su apoyo inquebrantable, sacrificios y amor incondicional han sido la base sobre la cual he construido mi educación y mi vida. Cada éxito que alcanzo es también suyo, y estoy eternamente agradecido por su dedicación y confianza en mí.

A mis queridos hermanos, Josemaría y Selvin, agradezco su constante aliento y comprensión. Su presencia ha sido un faro de apoyo emocional, haciéndome sentir respaldado en cada paso de

esta carrera. Su influencia positiva ha sido un recordatorio constante de la importancia de la unidad familiar, y estoy agradecido por tenerlos como cómplices en este viaje hacia el logro de mis metas.

A mi amada, Flor, quiero expresar mi más profundo agradecimiento. Su amor incondicional, paciencia y apoyo constante han sido pilares fundamentales en mi vida. Su presencia ha sido mi fuente de inspiración y motivación. Gracias por comprender mis compromisos, por celebrar mis logros y por ser mi compañera inquebrantable en este emocionante viaje. Tu presencia ha convertido cada desafío en una oportunidad para crecer juntos, y estoy agradecido por tenerte a mi lado.

Este logro no hubiera sido posible sin el apoyo de cada persona que ha cruzado mi camino. A todos ustedes, les dedico mi más sincero agradecimiento.

Con gratitud,

**Jairo Hernández**

Quiero aprovechar esta oportunidad para expresar mi más profundo agradecimiento a Dios, mi familia y amigos, cuyo apoyo ha sido un pilar fundamental en la realización de esta tesis.

A mis padres por su amor incondicional, su paciencia y su fe inquebrantable en mí. Su apoyo emocional y aliento constante han sido esenciales en cada paso de este camino. Su presencia y comprensión en los momentos más desafiantes de este proceso han sido una fuente de motivación y fortaleza.

También deseo agradecer sinceramente a mis amigos Gabriela, Roxana, Jaime, Daniela, Martha, Nicole, Milady y Lucia, por estar siempre allí para mí. Gracias por todas las palabras de aliento, los momentos de relajación y diversión que me ayudaron a mantener el equilibrio, y por ser una

fuente constante de alegría y apoyo. Vuestra amistad ha sido una luz brillante durante los tiempos de estudio y esfuerzo.

Este logro no sólo es un reflejo de mi esfuerzo, sino también de la amorosa presencia y el apoyo incondicional que he recibido de cada uno de ustedes. Gracias por ser parte de este viaje.

**Valeria Novoa**

## Tabla de contenido

1.	Introducción .....	1
2.	Antecedentes .....	3
3.	Justificación .....	5
4.	Pregunta de investigación .....	7
5.	Objetivos .....	8
5.1.	Objetivo General.....	8
5.2.	Objetivos Específicos .....	8
6.	Alcances .....	10
7.	Limitaciones.....	12
8.	Marco teórico .....	13
8.1.	Computación en la Nube .....	13
8.1.1.	Modelos de Servicio en la Nube .....	15
8.1.2.	Modelo de despliegue en la nube .....	16
8.1.3.	Infraestructura de la Nube .....	20
8.1.4.	Arquitectura de la Nube .....	21
8.1.5.	Regiones en la nube.....	21
8.1.6.	Cluster .....	24
8.2.	Virtualización .....	25
8.2.1.	Hipervisor.....	26

8.3.	OpenStack como Plataforma de Nube.....	29
8.3.1.	Componentes Principales de OpenStack.....	30
8.3.2.	Mensajería y colas.....	36
8.4.	Orquestación.....	38
8.4.1.	Orquestación del almacenamiento de datos en la nube.....	39
8.4.2.	Almacenamiento en Caché.....	40
8.5.	Control y accesos en la nube.....	42
8.5.1.	Usuario.....	42
8.5.2.	Rol.....	42
8.5.3.	Usuarios Federados.....	43
8.5.4.	Lightweight Directory Access Protocol (LDAP).....	43
8.5.5.	OpenLDAP.....	44
9.	Metodología de la investigación.....	46
9.1.	Población y muestreo.....	46
9.1.1.	Instrumentos y Técnicas de Recopilación de Datos:.....	47
9.2.	Resultados.....	48
9.2.1.	Análisis de datos cuantitativos.....	48
10.	Desarrollo de la solución.....	54

10.1.	Requerimientos Funcionales: .....	54
10.2.	Arquitectura .....	56
10.3.	Configuración .....	59
10.4.	Topología de red .....	61
11.	Caso de estudio.....	62
12.	Factibilidad.....	64
12.1.	Factibilidad Técnica: .....	64
12.2.	Factibilidad Económica: .....	65
12.3.	Comparativa de Costos .....	67
12.4.	Factibilidad humana .....	68
13.	Conclusiones .....	70
14.	Recomendaciones.....	72
15.	Referencias .....	74
16.	Anexo .....	78
16.1.	Encuesta.....	78

## TABLA DE ILUSTRACIONES

ILUSTRACIÓN 1: CLASIFICACIÓN DE LA COMPUTACIÓN EN NUBE EN DOS CONJUNTOS DISTINTOS DE MODELOS ...	14
ILUSTRACIÓN 2: MODELO DE SERVICIO EN LA NUBE .....	15
ILUSTRACIÓN 3: MODELO DE DESPLIEGUE EN LA NUBE .....	17
ILUSTRACIÓN 4: COMO SE COMUNICAN LOS SERVICIOS DE OPENSTACK.....	35
ILUSTRACIÓN 5: DIAGRAMA DE LA ARQUITECTURA DEL PROTOTIPO EN LA NUBE COMUNITARIA ENFOCADO EN LA INFRAESTRUCTURA .....	56
ILUSTRACIÓN 6: TOPOLOGÍA DE RED DE LA INFRAESTRUCTURA .....	61
ILUSTRACIÓN 7: DIAGRAMA DE NUBE COMUNITARIA.....	63

## 1. Introducción

En la era digital actual, la computación en la nube se ha establecido como un pilar esencial para el almacenamiento, procesamiento y acceso a datos, siendo una herramienta clave tanto para usuarios individuales como para organizaciones en todo el mundo. A pesar de su amplia adopción, el acceso a estos servicios de nube a menudo está restringido a entidades o regiones individuales, lo que plantea desafíos significativos en términos de colaboración y compartición de recursos a nivel global. Abordando esta problemática, nuestra investigación, realizada por un equipo multidisciplinario de expertos en tecnología de la información y desarrollo de software, se centra en el desarrollo de un prototipo de nube comunitaria. Esta nube, basada en la robusta infraestructura de OpenStack, busca superar las limitaciones de las nubes tradicionales, promoviendo una cooperación efectiva y un intercambio fluido de recursos entre diversas regiones geográficas.

El proyecto se concentra en diseñar la infraestructura necesaria para implementar una nube comunitaria, caracterizada por su capacidad de integrar un número ilimitado de regiones. Se presta especial atención a garantizar un funcionamiento eficiente y seguro, contemplando la federación de usuarios y la persistencia de sesión para proporcionar una experiencia de usuario coherente y continua. Una parte crucial del proyecto es la creación de una interfaz web intuitiva, diseñada para simplificar la gestión y el monitoreo de las distintas regiones de la nube comunitaria, asegurando así una administración eficaz de los servicios disponibles.

Además, el estudio se contextualiza en un caso práctico específico: una empresa de logística de transporte de vehículos operando entre Estados Unidos y El Salvador. La aplicación de la nube comunitaria en este entorno busca optimizar los procesos logísticos, con una infraestructura central en El Salvador y extensiones en varias regiones estadounidenses. Este caso de estudio provee un marco tangible para evaluar la implementación del prototipo de nube comunitaria, ilustrando su efectividad y los desafíos inherentes en un contexto empresarial real.

En resumen, esta investigación se enfoca en el desarrollo de un prototipo de nube comunitaria diseñado para impulsar la cooperación y el intercambio de recursos a través de fronteras regionales, utilizando OpenStack como su piedra angular. El objetivo es crear un modelo de nube que no solo atienda las necesidades actuales de los usuarios, sino que también establezca un marco para el futuro de la gestión eficiente de recursos en la nube comunitaria.

## 2. Antecedentes

Las infraestructuras de nube han revolucionado la forma en que las organizaciones despliegan y gestionan sus recursos digitales. En el sector de la logística y el transporte, la capacidad de escalar recursos rápidamente y gestionar grandes volúmenes de datos ha sido fundamental. La nube ha permitido a empresas como las de logística de vehículos mejorar su eficiencia operativa y capacidad de respuesta.

Las infraestructuras de nube han revolucionado la forma en que las organizaciones despliegan y gestionan sus recursos digitales. En el sector de la logística y el transporte, la capacidad de escalar recursos rápidamente y gestionar grandes volúmenes de datos ha sido fundamental. La nube ha permitido a empresas como las de logística de vehículos mejorar su eficiencia operativa y capacidad de respuesta.

Con el desarrollo de Microsoft Azure, Google Cloud Platform y otros servicios, la nube se ha convertido en una solución omnipresente para empresas de todos los tamaños, incluyendo aquellas en el sector logístico.

Desde su lanzamiento en el 2010, OpenStack se ha destacado por su adaptabilidad a diferentes entornos, convirtiéndose en una opción preferida para organizaciones que buscan soluciones de nube personalizadas. Su desarrollo continuo ha permitido adaptaciones específicas para industrias como la logística de vehículos.

Las nubes comunitarias ganaron tracción como una solución colaborativa para compartir recursos en sectores como la educación y la investigación. Programas como EGI

Federated Cloud han demostrado cómo diferentes organizaciones pueden beneficiarse de la colaboración en la nube.

La autenticación federada y el SSO, con tecnologías como OAuth y SAML, se han convertido en estándares de facto para la gestión de identidades en la última década. LDAP, implementado inicialmente en los 90, sigue siendo un componente crucial en muchos sistemas.

Las interfaces web para la gestión de sistemas en la nube han evolucionado continuamente, con ejemplos notables como el panel de control de AWS lanzado en los 2000 y las iteraciones sucesivas en plataformas como OpenStack Horizon.

Este panorama histórico y técnico ilustra cómo las tecnologías relacionadas con la computación en la nube han evolucionado para satisfacer las necesidades cambiantes de las organizaciones y cómo han influido en sectores críticos como la logística y el transporte de vehículos. La adaptabilidad y la innovación continua en estas áreas no solo han permitido una gestión más eficiente y segura de los recursos digitales, sino que también han abierto camino para nuevas formas de colaboración y operación en la era digital. Este contexto enriquece la base sobre la cual se desarrolla el prototipo de Nube Comunitaria para la empresa de logística de vehículos, resaltando la relevancia y la oportunidad que representa este proyecto en el marco actual de las tecnologías de nube.

### 3. Justificación

La implementación de una nube comunitaria basada en OpenStack para una empresa de logística de vehículos que opera entre Estados Unidos y El Salvador representa una estrategia crucial para enfrentar los desafíos contemporáneos y aprovechar las oportunidades en el ámbito de la tecnología de la información. Esta iniciativa se justifica por la necesidad de mejorar la eficiencia operativa y la gestión de datos en un entorno cada vez más digitalizado y competitivo.

La infraestructura de TI actual, limitada en términos de escalabilidad y flexibilidad, se verá mejorada con la adopción de una nube comunitaria. Esta nueva infraestructura promete no solo una gestión más eficiente de los recursos, sino también una reducción en los tiempos de procesamiento y una mejor coordinación entre las diferentes regiones operativas. La expectativa es que tales mejoras se reflejen en una reducción notable de los tiempos de procesamiento, aumentando la capacidad de respuesta de la empresa ante las demandas del mercado.

Además, la centralización de los datos en una nube comunitaria facilitará un acceso más rápido y seguro a la información crítica, algo esencial para la toma de decisiones estratégicas y para mantener la integridad de los datos a lo largo de toda la cadena logística. La implementación de una base de datos centralizada y un sistema de gestión de identidades robusto promete mejorar la coherencia y seguridad de los datos, aspectos fundamentales en el sector logístico.

Por otro lado, se espera que la implementación de una interfaz web intuitiva y una autenticación federada simplifiquen el acceso y la gestión de los sistemas para los usuarios finales, mejorando su experiencia y reduciendo la curva de aprendizaje. La adopción de estas tecnologías avanzadas en la nube no solo mejorará las operaciones actuales, sino que también fomentará la innovación y mantendrá a la empresa competitiva en un mercado en constante cambio.

Desde el punto de vista de la factibilidad, OpenStack, como una plataforma de nube de código abierto probada y confiable, garantiza una base sólida para el desarrollo del proyecto. La infraestructura propuesta es viable, con numerosos casos de éxito en industrias similares que respaldan su aplicación y beneficios. Aunque la implementación inicial implica una inversión grande, el retorno a largo plazo en términos de eficiencia operativa y reducción de costos justifica este gasto. Además, la escalabilidad de la nube comunitaria permitirá un crecimiento sostenible con una inversión incremental, y la existencia de una amplia comunidad y soporte para OpenStack asegura asistencia continua y actualizaciones para mantener el sistema operativo y seguro.

En resumen, la nube comunitaria basada en OpenStack se presenta como una oportunidad estratégica no solo para responder a las necesidades actuales de la empresa, sino también para posicionarla de manera óptima para futuras innovaciones y mejoras en un sector altamente competitivo. La implementación de este proyecto promete transformar la manera en que la empresa gestiona sus operaciones logísticas, marcando un hito significativo en su evolución tecnológica y operativa.

## 4. Pregunta de investigación

¿Cómo puede implementarse de manera eficiente un prototipo de infraestructura de Nube Comunitaria basada en OpenStack que brinde a las entidades miembros una plataforma centralizada para compartir recursos tecnológicos, administrar múltiples regiones y garantizar la autenticación segura de usuarios y el acceso a servicios a través de un API?

## 5. Objetivos

### 5.1. Objetivo General

Desarrollar un prototipo de Infraestructura de nube comunitaria basada en OpenStack compuesta por múltiples regiones, ofreciendo una interfaz Web para la gestión de las regiones, servicio de autenticación, basado en usuarios servicios federados y persistente a servicios en la nube, garantizando la disponibilidad y seguridad de sus datos.

### 5.2. Objetivos Específicos

1. Diseñar y desarrollar un prototipo de infraestructura de nube comunitaria basada en OpenStack que permita el despliegue de múltiples regiones.
2. Validar la funcionalidad de usuarios federados y persistencia de sesión en la nube comunitaria.
3. Configurar la interfaz web para la gestión y supervisión de las diferentes regiones dentro de la nube.
4. Configurar una base de datos centralizada que respalde aspectos clave de la nube comunitaria y la operación logística.
5. Crear un servicio de identificación que proporcione autenticación de usuarios y descubrimiento de servicios para toda la nube comunitaria.
6. Establecer una API que permita a las diferentes regiones consumir los servicios proporcionados por la nube comunitaria.

7. Realizar pruebas exhaustivas del prototipo para validar su funcionalidad, escalabilidad y seguridad.
8. Crear documentación detallada sobre la infraestructura y configuración de la nube comunitaria.

## 6. Alcances

1. Diseño y desarrollo de la infraestructura basada en OpenStack para el despliegue de una Nube Comunitaria compuesta por un número ilimitado de regiones.
2. Desarrollo de un sistema de usuarios federados que permita a los usuarios acceder a diferentes regiones de la nube con una única identidad, facilitando la integración y simplificación de la autenticación.
3. Desarrollo de mecanismos de persistencia de sesión para garantizar que los usuarios mantengan su sesión activa al moverse entre regiones, evitando interrupciones y pérdida de datos.
4. Creación de una interfaz web intuitiva y fácil de usar que permita a los usuarios gestionar y administrar las diferentes regiones de la nube comunitaria, brindando control y visibilidad sobre los recursos y servicios disponibles.
5. Desarrollo de una base de datos centralizada que almacene y gestione los datos de los usuarios en todas las regiones, asegurando la consistencia y disponibilidad de la información.
6. Desarrollo de un servicio de identificación seguro y confiable que permita la autenticación de los usuarios en todas las regiones de la nube comunitaria, así como el descubrimiento de servicios disponibles, facilitando la integración de aplicaciones y servicios.

7. Creación de un API que permita la comunicación entre las diferentes regiones de la nube comunitaria, facilitando el consumo de los servicios proporcionados y asegurando la interoperabilidad, excluyendo la integración de un sistema de almacenamiento ya que esto va en un desarrollo más profundo de regiones.

## 7. Limitaciones

**Coordinación entre Regiones:** La gestión de múltiples regiones, cada una con sus propias características logísticas y requisitos específicos, puede requerir una coordinación efectiva. La necesidad de mantener una sincronización y coherencia en la operación entre las diferentes yardas y regiones se convierte en un desafío.

**Dependencia de Proveedores Externos:** Para ciertos componentes y servicios, puede haber una dependencia de proveedores externos, lo que podría generar riesgos en términos de costos, disponibilidad y control.

**Dependencia de la Conectividad a Internet:** Al estar basada en la nube, la infraestructura depende en gran medida de una conexión a internet estable y rápida, lo que podría ser un problema en regiones con infraestructura de internet limitada.



















































































## 10. Desarrollo de la solución

Se ha desarrollado un prototipo de infraestructura de nube comunitaria geo-distribuida basada en OpenStack. Este prototipo se destaca por su capacidad para admitir múltiples regiones, y actúa como el núcleo central de alojamiento para estas regiones, que utilizan los servicios ofrecidos. Es importante mencionar que la construcción del prototipo de infraestructura para nube comunitaria se basó en documentos de Alto Nivel (HLD) y Bajo Nivel (LLD), que se pueden encontrar en los Anexos.

A continuación, se detallan los distintos requisitos que se precisaron para el desarrollo de este proyecto.

### 10.1. Requerimientos Funcionales:

1. Creación de Regiones: El sistema permite la creación de un número ilimitado de regiones dentro de la Nube Comunitaria. Cada Región puede configurarse de manera independiente.
2. Usuarios Federados: El sistema admite la autenticación de usuarios de manera federada, lo que significa que los usuarios pueden acceder a las diferentes regiones de la Nube Comunitaria con un solo conjunto de credenciales.
3. Persistencia de Sesión: Existe la funcionalidad de persistencia de sesión para garantizar que los usuarios puedan mantener su sesión activa al cambiar de una región a otra sin necesidad de volver a autenticarse.













































Red Hat. (s.f.). *What is lightweight directory access protocol (LDAP) authentication?*

Obtenido de Red Hat: <https://www.redhat.com/en/topics/security/what-is-ldap-authentication>

Stephen J. Bigelow. (10 de 05 de 2019). *Improve cloud performance with these 6 techniques.*

Obtenido de TechTarget:

<https://www.techtarget.com/searchcloudcomputing/tip/Improve-cloud-performance-with-these-6-techniques>

SUSE . (s.f.). *Computer Cluster*. Obtenido de Technology Definitions | SUSE Defines:

<https://www.suse.com/suse-defines/definition/computer-cluster/>

































































































































## *Listado de Nodos a Configurar – Regiones*

Para esta instalación nos basaremos primero a instalar Openstack en una sola region, y ocuparemos primero la guía oficial de Openstack, luego de ello se hacen las configuraciones correspondientes para conectarnos y/o consumir los servicios de nuestra infraestructura.

<https://docs.openstack.org/install-guide/>

Así como la guía tendremos 2 nodos un Controller y un Compute.

### Creación de cola en RabbitMQ

Sera necesario crear una cola para cada una de las regiones, se tomará el siguiente ejemplo como base para una region:

```
$ sudo rabbitmqctl add_user NAME_REGION PASS
$ sudo rabbitmqctl add_vhost /VHOST_REGION
$ sudo rabbitmqctl set_permissions -p /VHOST_REGION
NAME_REGION ".*" ".*" ".*"
```

En nuestro caso NAME\_REGION = regionone y VHOST\_REGION = regionone.

NOTA: El password que normalmente ocupamos es icc115.

### Integración Servicios OpenStack RegionOne

#### *Glance*

Ingresamos a mysql y agregamos una base de datos para Glance de RegionOne llamado glance\_one.

```
# mysql
MariaDB [(none)]> CREATE DATABASE glance_one;
MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance_one.* TO
'glance_one'@'localhost' IDENTIFIED BY 'GLANCE_DBPASS';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance_one.* TO
'glance_one'@'%' IDENTIFIED BY 'GLANCE_DBPASS';
```

Cambiar GLANCE\_DBPASS por icc115.

En Keystone agregamos los servicios para Glance RegionOne.

```
$ . admin-openrc
$ openstack user create --domain default --password-prompt glance
$ openstack role add --project service --user glance admin
$ openstack service create --name glance --description "OpenStack Image" image
$ openstack endpoint create --region RegionOne image public http://172.16.0.101:9292
$ openstack endpoint create --region RegionOne image internal http://172.16.0.101:9292
$ openstack endpoint create --region RegionOne image admin http://172.16.0.101:9292
```

Abrimos el archivo de configuración de Glance `/etc/glance/glance-api.conf` en la

**RegionOne** y reemplazamos con lo siguiente:

```
[database]
# ...
connection =
mysql+pymysql://glance_one:iccl15@172.16.0.31:4008/glance_one

[keystone_authtoken]
# ...
www_authenticate_uri = http://172.16.0.11:5000
auth_url = http://172.16.0.11:5000
memcached_servers = 172.16.0.21:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = glance
password = iccl15

[oslo_limit]
auth_url = http://172.16.0.11:5000
auth_type = password
user_domain_id = default
username = glance
system_scope = all
password = iccl15
endpoint_id = ENDPOINT_PUBLIC
region_name = RegionOne
```

El `ENDPOINT_PUBLIC` es el que se registró en el nodo de Keystone. Si un dado caso

perdemos ese dato, podemos ejecutar el siguiente comando para recuperarlo:

```
$ openstack endpoint list --service image
```

Nos mostrara una lista de endpoint para el servicio de Glance, luego solo lo copiamos, en este caso nuestro id publico es: cb267e7fc5274596b8d71cb8ace40bf4

Luego ejecutamos los siguientes comandos en el nodo de Keystone:

```
$ openstack role add --user glance --user-domain Default --system all reader
# su -s /bin/sh -c "glance-manage db_sync" glance
```

### *Placement*

Agregamos lo siguiente a la base de datos.

```
MariaDB [(none)]> CREATE DATABASE placement_one;
MariaDB [(none)]> GRANT ALL PRIVILEGES ON placement_one.* TO 'placement_one'@'localhost' IDENTIFIED BY 'iccl15';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON placement_one.* TO 'placement_one'@'%' IDENTIFIED BY 'iccl15';
```

Agregamos los endpoint al nodo Keystone

```
$ . admin-openrc
$ openstack user create --domain default --password-prompt placement
$ openstack role add --project service --user placement admin
$ openstack service create --name placement --description "Placement API" placement
$ openstack endpoint create --region RegionOne placement public http://172.16.0.101:8778
$ openstack endpoint create --region RegionOne placement internal http://172.16.0.101:8778
$ openstack endpoint create --region RegionOne placement admin http://172.16.0.101:8778
```

Configuramos el archivo en la RegionOne /etc/placement/placement.conf

```
[placement_database]
connection =
mysql+pymysql://placement_one:iccl15@172.16.0.31:4008/placement_one

[keystone_authtoken]
# ...
auth_url = http://172.16.0.11:5000/v3
memcached_servers = 172.16.0.21:11211
auth_type = password
project_domain name = Default
```

```
user_domain_name = Default
project_name = service
username = placement
password = icc115
```

Ejecutamos lo siguiente para poblar la base de datos de placement:

```
# su -s /bin/sh -c "placement-manage db sync" placement
```

Reiniciamos el servicio de placement

```
# service apache2 restart
```

*Nova*

Agregamos lo siguiente en la base de datos.

```
MariaDB [(none)]> CREATE DATABASE novaone_api;
MariaDB [(none)]> CREATE DATABASE novaone;
MariaDB [(none)]> CREATE DATABASE novaone_cell0;

MariaDB [(none)]> GRANT ALL PRIVILEGES ON novaone_api.* TO
'novaone'@'localhost' \
  IDENTIFIED BY 'NOVA_DBPASS';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON novaone_api.* TO
'novaone'@'%' \
  IDENTIFIED BY 'NOVA_DBPASS';

MariaDB [(none)]> GRANT ALL PRIVILEGES ON novaone.* TO
'novaone'@'localhost' \
  IDENTIFIED BY 'NOVA_DBPASS';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON novaone.* TO
'novaone'@'%' \
  IDENTIFIED BY 'NOVA_DBPASS';

MariaDB [(none)]> GRANT ALL PRIVILEGES ON novaone_cell0.* TO
'novaone'@'localhost' \
  IDENTIFIED BY 'NOVA_DBPASS';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON novaone_cell0.* TO
'novaone'@'%' \
  IDENTIFIED BY 'NOVA_DBPASS';
```

Sustituimos NOVA\_DBPASS por icc115.

Agregamos el usuario, el servicio y los endpoints de nova en el nodo de Keystone.

```
$ . admin-openrc
```

```

$ openstack user create --domain default --password-prompt
nova
$ openstack role add --project service --user nova admin
$ openstack service create --name nova --description
"OpenStack Compute" compute
$ openstack endpoint create --region RegionOne compute public
http://172.16.0.101:8774/v2.1
$ openstack endpoint create --region RegionOne compute
internal http://172.16.0.101:8774/v2.1
$ openstack endpoint create --region RegionOne compute admin
http://172.16.0.101:8774/v2.1

```

Modificamos el siguiente archivo del Controller de RegionOne /etc/nova/nova.conf

```

[DEFAULT]
# ...
transport_url =
rabbit://regionone:iccl15@172.16.0.22:5672//regionone
my_ip = 172.16.0.101

[api_database]
# ...
connection =
mysql+pymysql://novaone:iccl15@172.16.0.31:4008/novaone_api

[database]
# ...
connection =
mysql+pymysql://novaone:iccl15@172.16.0.31:4008/novaone

[keystone_authtoken]
# ...
www_authenticate_uri = http://172.16.0.11:5000/
auth_url = http://172.16.0.11:5000/
memcached_servers = 172.16.0.21:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = nova
password = iccl15

[service_user]
send_service_user_token = true
auth_url = https://172.16.0.11:5000/v3
auth_strategy = keystone
auth_type = password
project_domain_name = Default

```

```
project_name = service
user_domain_name = Default
username = nova
password = icc115

[glance]
# ...
api_servers = http://172.16.0.101:9292

[placement]
# ...
region_name = RegionOne
project_domain_name = Default
project_name = service
auth_type = password
user_domain_name = Default
auth_url = http://172.16:5000/v3
username = placement
password = icc115
```

**Poblamos las bases de datos:**

```
# su -s /bin/sh -c "nova-manage api_db sync" nova
# su -s /bin/sh -c "nova-manage cell_v2 map_cell0" nova
# su -s /bin/sh -c "nova-manage cell_v2 create_cell --
name=cell1 --verbose" nova
# su -s /bin/sh -c "nova-manage db sync" nova
# su -s /bin/sh -c "nova-manage cell_v2 list_cells" nova
```

**Reiniciamos los servicios de Nova**

```
# service nova-api restart
# service nova-scheduler restart
# service nova-conductor restart
# service nova-novncproxy restart
```

**Modificamos el siguiente archivo en el nodo Compute de RegionOne /etc/nova/nova.conf**

```
[DEFAULT]
# ...
transport_url =
rabbit://regionone:icc115@172.16.0.22:5672//regionone
my_ip = 172.16.0.102

[keystone_authtoken]
# ...
www_authenticate_uri = http://172.16.0.11:5000/
auth_url = http://172.16.0.11:5000/
memcached_servers = 172.16.0.21:11211
```

```

auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = nova
password = iccl15

[service_user]
send_service_user_token = true
auth_url = https://172.16.0.11:5000/v3
auth_strategy = keystone
auth_type = password
project_domain_name = Default
project_name = service
user_domain_name = Default
username = nova
password = iccl15

[vnc]
novncproxy_base_url = http://172.16.0.101:6080/vnc_auto.html

[glance]
# ...
api_servers = http://172.16.0.101:9292

[placement]
# ...
region_name = RegionOne
project_domain_name = Default
project_name = service
auth_type = password
user_domain_name = Default
auth_url = http://172.16.0.11:5000/v3
username = placement
password = iccl15

```

**Reiniciamos el servicio de Nova:**

```
# service nova-compute restart
```

En el nodo Controller de RegionOne confirmamos que se encuentren los hosts compute:

```
$ . admin-openrc
$ openstack compute service list --service nova-compute
```

**Podamos la base de datos:**

```
# su -s /bin/sh -c "nova-manage cell_v2 discover_hosts --
verbose" nova
```

### Neutron

Agregamos la siguiente base de datos y permisos:

```
MariaDB [(none)]> CREATE DATABASE neutron_one;
MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron_one.* TO
'neutron_one'@'localhost' IDENTIFIED BY 'iccl15';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron_one.* TO
'neutron_one'@'%' IDENTIFIED BY 'iccl15';
```

En el nodo Keystone de nuestra infraestructura agregamos el usuario, el servicio y los endpoint para neutron.

```
$ . admin-openrc
$ openstack user create --domain default --password-prompt
neutron
$ openstack role add --project service --user neutron admin
$ openstack service create --name neutron --description
"OpenStack Networking" network
$ openstack endpoint create --region RegionOne network public
http://172.16.0.101:9696
$ openstack endpoint create --region RegionOne network
internal http://172.16.0.101:9696
$ openstack endpoint create --region RegionOne network admin
http://172.16.0.101:9696
```

Editamos el archivo de configuración de neutron en el nodo Controller de nuestra

RegionOne /etc/neutron/neutron.conf:

```
[DEFAULT]
# ...
transport_url =
rabbit://regionone:iccl15@172.16.0.22:5672//regionone

[database]
# ...
connection =
mysql+pymysql://neutron_one:iccl15@172.16.0.31:4008/neutron_o
ne

[keystone_auth token]
www_authenticate_uri = http://172.16.0.11:5000
auth_url = http://172.16.0.11:5000
```

```

memcached_servers = 172.16.0.21:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = neutron
password = icc115

[nova]
# ...
auth_url = http://172.16.0.11:5000
auth_type = password
project_domain_name = Default
user_domain_name = Default
region_name = RegionOne
project_name = service
username = nova
password = icc115

```

**Modificamos el archivo openvswitch del Controller**

`/etc/neutron/plugins/ml2/openvswitch_agent.ini`

```

[vxlan]
local_ip = 172.16.0.101
l2_population = true

```

**Modificamos el archivo de configuración de neutron `/etc/nova/nova.conf`:**

```

[neutron]
# ...
auth_url = http://172.16.0.11:5000
auth_type = password
project_domain_name = Default
user_domain_name = Default
region_name = RegionOne
project_name = service
username = neutron
password = icc115
service_metadata_proxy = true

```

**Poblamos la base de datos de neutron**

```

su -s /bin/sh -c "neutron-db-manage --config-file
/etc/neutron/neutron.conf --config-file
/etc/neutron/plugins/ml2/ml2_conf.ini upgrade head" neutron

```

**Reiniciamos servicios:**

```

# service nova-api restart

```

```
# service neutron-server restart
# service neutron-openvswitch-agent restart
# service neutron-dhcp-agent restart
# service neutron-metadata-agent restart
# service neutron-l3-agent restart
```

En el nodo Compute de RegionOne editamos el archivo `/etc/neutron/neutron.conf`:

```
[DEFAULT]
# ...
transport_url = rabbit://regionone:iccl15@172.16.0.22:5672//regionone

[keystone_authtoken]
# ...
www_authenticate_uri = http://172.16.0.11:5000
auth_url = http://172.16.0.11:5000
memcached_servers = 172.16.0.21:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = neutron
password = iccl15
```

Editamos el archivo de configuración de Nova `/etc/nova/nova.conf`:

```
[neutron]
# ...
auth_url = http://172.16.0.11:5000
auth_type = password
project_domain_name = Default
user_domain_name = Default
region_name = RegionOne
project_name = service
username = neutron
password = iccl15
```

Reiniciamos los servicios de nodo Compute:

```
# service nova-compute restart
# service neutron-openvswitch-agent restart
```

## Instalación Horizon

Horizon se puede instalar en cualquier nodo de nuestra infraestructura y además también en los nodos de nuestras Regiones, por convicción lo instalaremos en el nodo Keystone de nuestra infraestructura a modo de ejem

Para instalarlo ejecutamos:

```
# apt install openstack-dashboard
```

Editamos el siguiente archiv `tc/openstack-dashboard/local_settings.py`, contiene la configuracion de Horizon para conectarse a los servicios que hemos creado:

```
OPENSTACK_HOST = "172.16.0.11"
...
SESSION_ENGINE = 'django.contrib.sessions.backends.cache'
...
CACHES = {
    'default': {
        'BACKEND':
'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '10.0.0.21:11211',
    }
}
...
OPENSTACK_KEYSTONE_URL = "http://%s:5000/identity/v3" %
OPENSTACK_HOST

OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True

OPENSTACK_API_VERSIONS = {
    "identity": 3,
    "image": 2,
    "volume": 3,
}

OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"

OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"
```

Las credenciales en nuestro caso son usuario: admin, password: iccl15.

