

OPENSTACK

GERARDO CORREA VARGAS



@gcorreav



openstack®



Red Hat

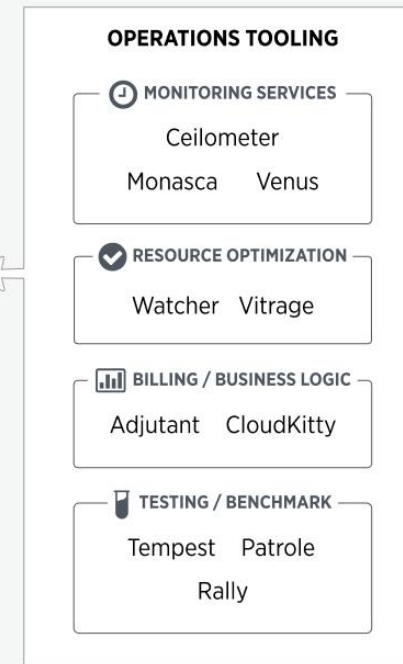
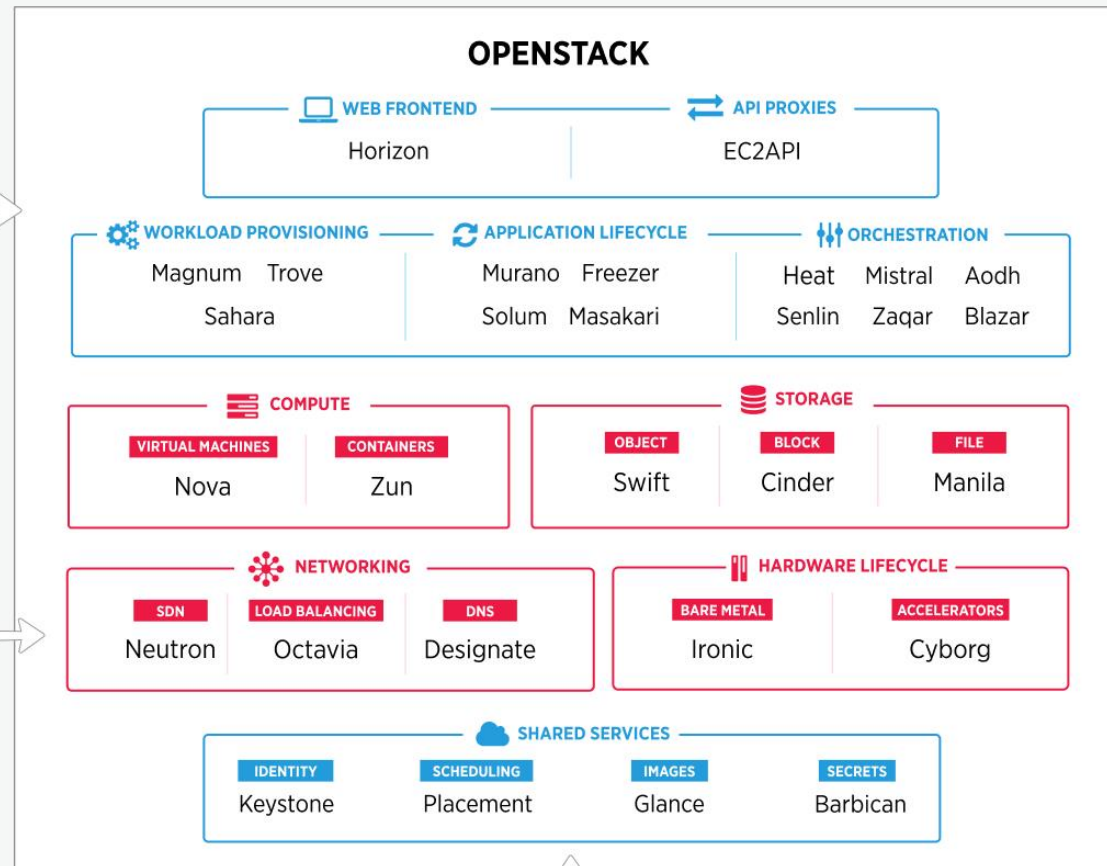
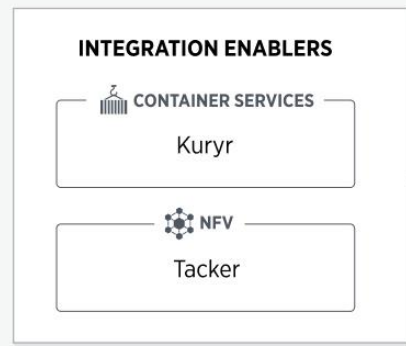
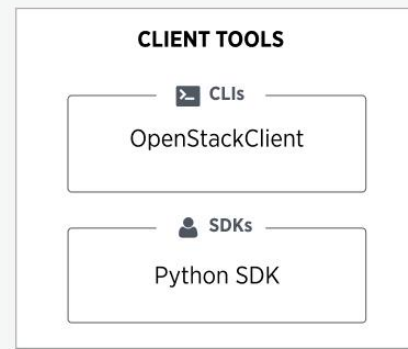
OpenStack Platform

REQUISITOS

- Laptop con sistema operativo Linux (de preferencia) con las siguientes herramientas instaladas:
 - Cliente SSH
 - Firefox
 - Comando git
- Conocimientos básicos de:
 - Configuración de red en RHEL
 - Instalación de paquetes vía dnf/yum en RHEL
 - Particionado de discos y asignación de sistemas de archivo xfs y ext4

¿QUÉ ES OPENSTACK?

- Proyecto de software open source para crear nubes privadas o públicas. Controla pools de recursos de procesamiento (compute), almacenamiento (storage) y redes (network), todos ellos administrados y aprovisionados a través de APIs con mecanismos en común de autenticación.
- Soportado a través de la “OpenInfra Foundation”:
- <https://openinfra.dev/about/>
- Conjunto de proyectos:
- <https://www.openstack.org/software/project-navigator/openstack-components#openstack-services>
- Existen diversos “sabores” en la industria:
- <https://www.openstack.org/marketplace/distros/>



Version 2022.10.01

 openstack.

MODELOS DE CONSUMO

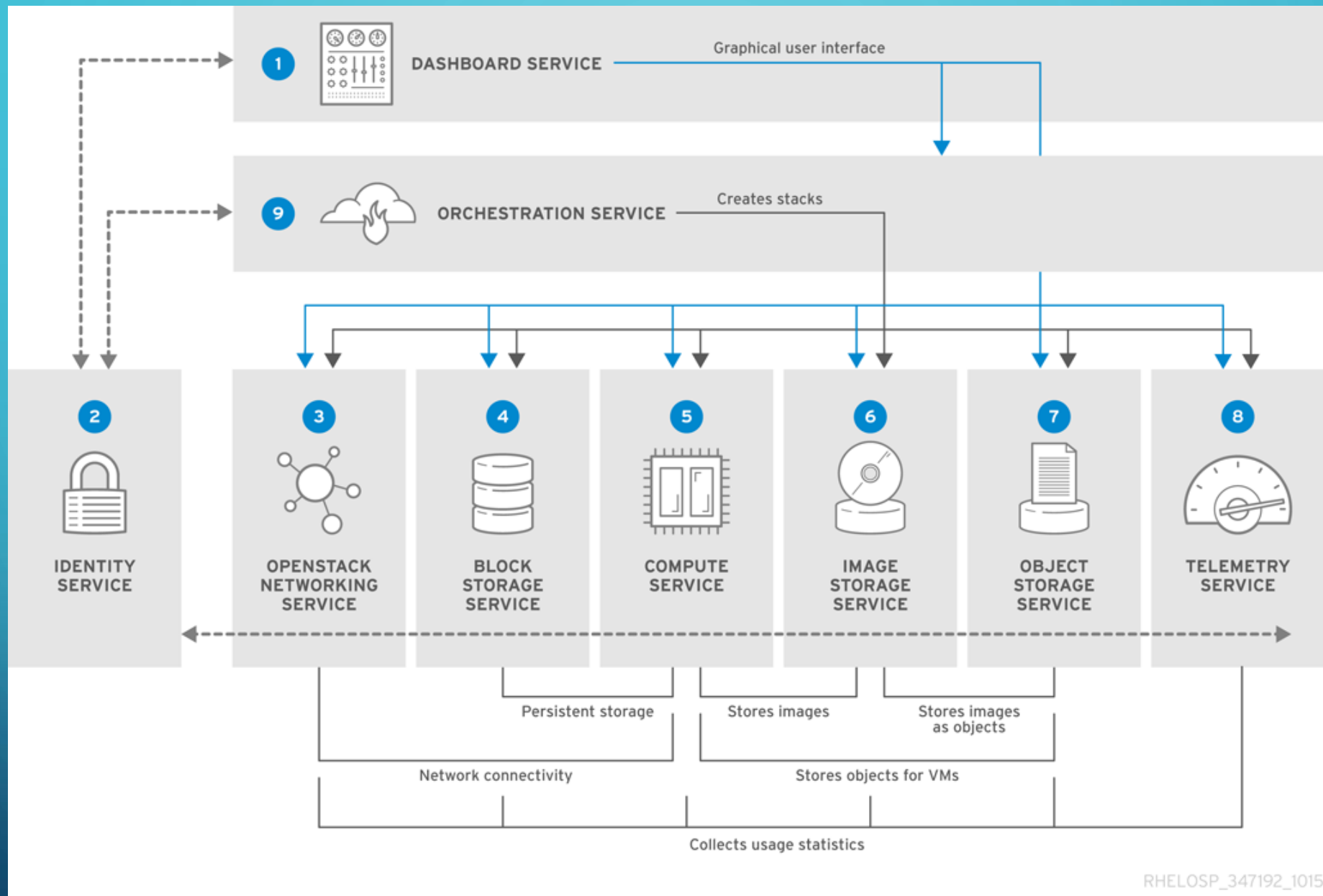
- Manual (on-demand):
 - Horizon
 - CLI
- Automatizado (Infrastructure-as-Code):
 - Heat Orchestration Templates (HOT)
 - Ansible
 - Puppet / Chef
 - Shell Scripts



@gcorreav

ARQUITECTURA SIMPLIFICADA

- **Nodos Control:** Mínimo 1 o más de 3. Componentes propios de la infraestructura (mensajería, API endpoints, autenticación).
- **Nodos Compute:** Sistema operativo + hypervisor + agentes (red, telemetría, entre otros).
- **Nodos Storage:** Agentes de almacenamiento de bloques, de objetos
- **Nodos Network:** Agentes L3, DHCP, LBaaS, entre otros.



RHELOSP_347192_1015

SDN

- Software-defined networking (SDN) es una tecnología que permite configurar redes de manera dinámica, programática y por lo tanto eficiente.
- Lo anterior permite un mejor rendimiento y monitoreo de la red, apegándose más al modelo de cómputo en la nube que la administración tradicional de redes.

SDN EN OPENSTACK

- OpenStack toma estas tecnologías para:
 - Mapear de manera directa VLANs subyacentes en la infraestructura.
 - Túneles y redes virtuales por VXLAN y GRE.
- Ejemplos de estas tecnologías son:
 - Linuxbridge
 - Ejemplos de controladores SDN Open source o comerciales:
 - Open vSwitch (OVS), OpenContrail, OpenDaylight, VMWare NSX, Cisco

RED HAT OPENSTACK PLATFORM

- La versión 16.2 está basada en la versión “Train” del proyecto libre.
- Esta versión sólo soporta hypervisores KVM en los nodos compute a través del driver **libvirt**.
- https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/16.2/html/release_notes/assembly_relnotes-intro

OpenStack Releases

Release Series

OpenStack is developed and released around 6-month cycles. After the initial release, additional stable point releases will be released in each release series. You can find the detail of the various release series here on their series page. Subscribe to the [combined release calendar](#) for continual updates.

Series	Status	Initial Release Date	Next Phase	EOL Date
2023.1 Antelope (SLURP)	Development	2023-03-22 <i>estimated (schedule)</i>	Maintained <i>estimated 2023-03-22</i>	
Zed	Maintained	2022-10-05	Extended Maintenance <i>estimated 2024-04-05</i>	
Yoga	Maintained	2022-03-30	Extended Maintenance <i>estimated 2023-09-30</i>	
Xena	Maintained	2021-10-06	Extended Maintenance <i>estimated 2023-04-06</i>	
Wallaby	Extended Maintenance (see note below)	2021-04-14	Unmaintained <i>TBD</i>	
Victoria	Extended Maintenance (see note below)	2020-10-14	Unmaintained <i>TBD</i>	
Ussuri	Extended Maintenance (see note below)	2020-05-13	Unmaintained <i>TBD</i>	
Train	Extended Maintenance (see note below)	2019-10-16	Unmaintained <i>TBD</i>	
Stein	Extended Maintenance (see note below)	2019-04-10	Unmaintained <i>TBD</i>	
Rocky	Extended Maintenance (see note below)	2018-08-30	Unmaintained <i>TBD</i>	
Queens	Extended Maintenance (see note below)	2018-02-28	Unmaintained <i>TBD</i>	
Pike	End Of Life	2017-08-30		2022-10-13

- <https://releases.openstack.org/>

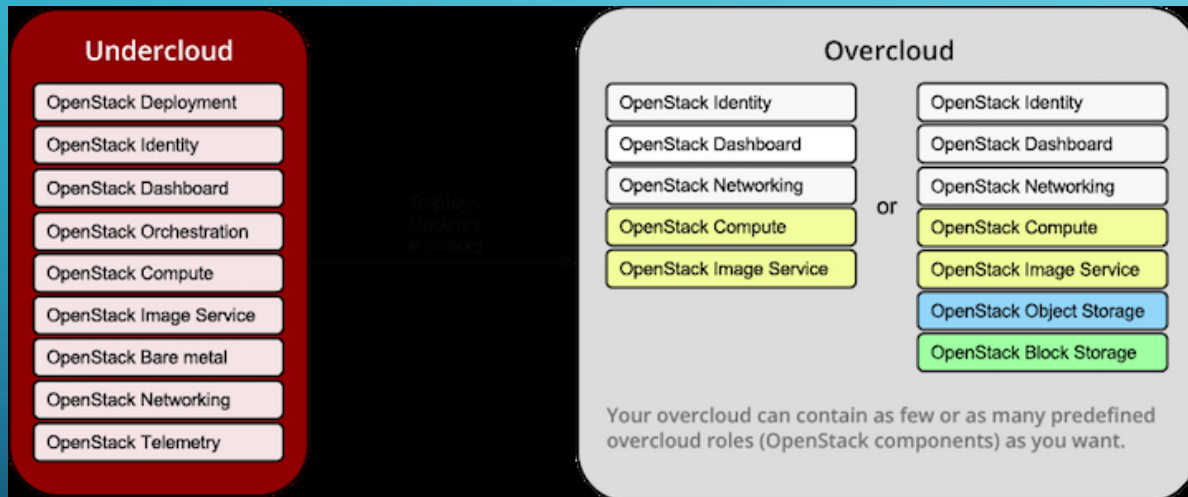


@gcorreav

FORMAS DE DESPLIEGUE

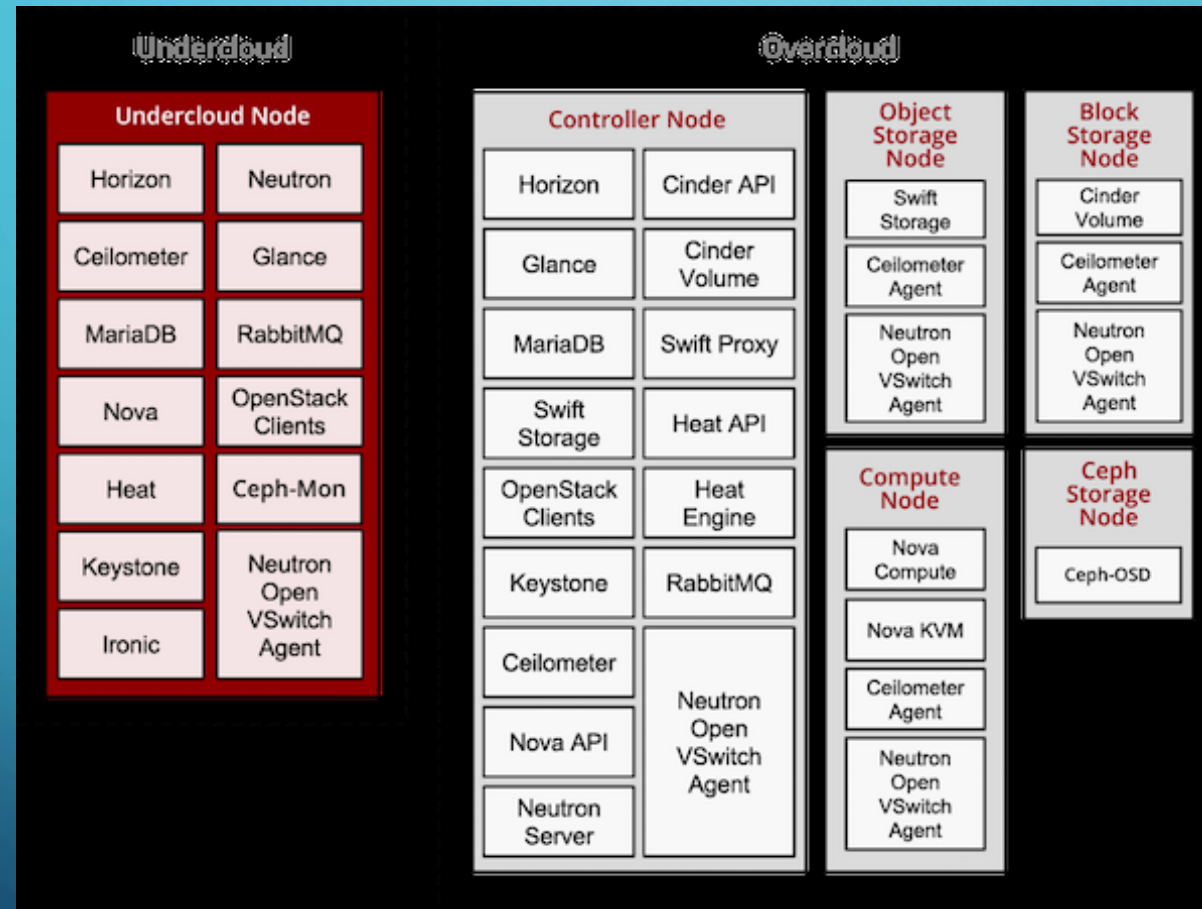
- All-In-One: Más orientado a ser una prueba de concepto de OpenStack.
Ejemplo: <https://www.rdoproject.org/install/packstack/>
- TripleO (OpenStack On OpenStack) : Es una manera de instalar, actualizar y operar entornos de OpenStack a través de los mismos proyectos que componen a OpenStack, por ejemplo nova, neutron, y heat.

ARQUITECTURA TRIPLE-O



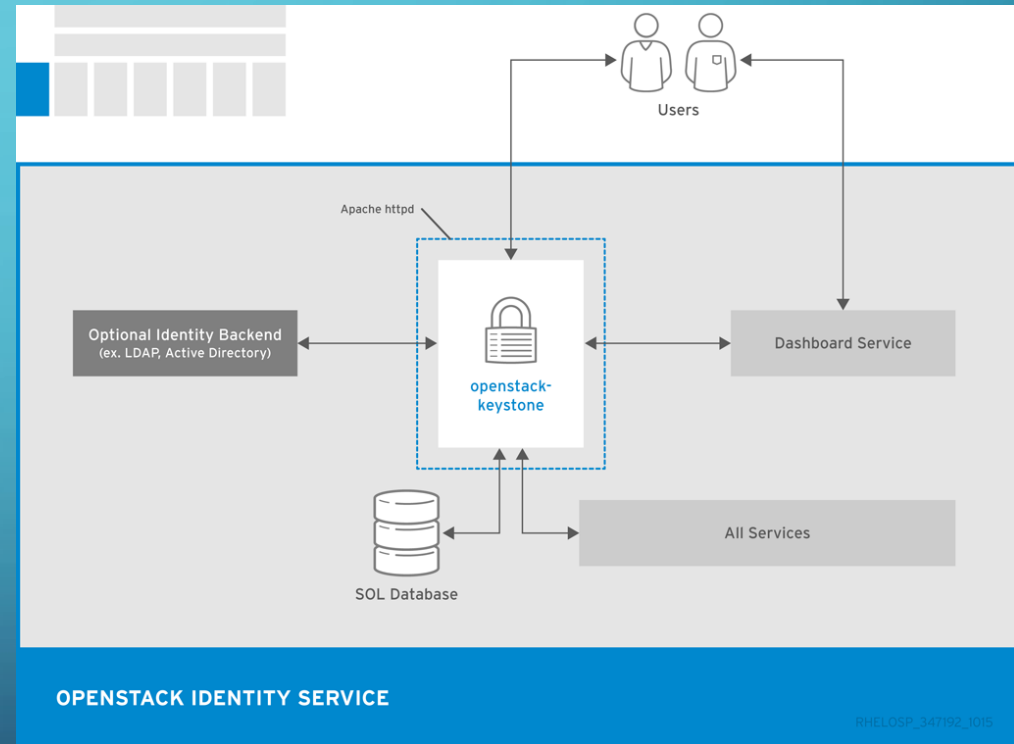
- **undercloud:** Es una nube de despliegue (OpenStack) con los componentes necesarios para desplegar y administrar un Overcloud
- **overcloud:** Es una nube para cargas de trabajo (OpenStack), es decir, para producción.

RELACIÓN DE ROLES DE NODOS CON TRIPLE-O



KEYSTONE

- Provee de autenticación y autorización a los componentes de OpenStack



KEYSTONE

- Accesible vía WebGUI (Horizon) y por CLI (con un archivo RC).
- OpenStack usa UUIDs (Universal Unique Identifiers) para cada recurso que administra.
- Administra 5 tipos de recursos:
 - Identidad: Usuarios y grupos
 - Dominios y Proyectos
 - Roles
 - Tokens
 - Catálogo y Endpoints
 - Políticas

KEYSTONE: CONCEPTOS

- Domain: Conjunto de usuarios, grupos y proyectos.
- Proyecto: Conjunto de recursos (compute, almacenamiento, redes).
- Política: Regla que declara qué acciones es posible ejecutar sobre qué objetos. Se definen en los archivos policy.json de cada servicio.
- Rol: Conjunto de políticas bajo un nombre específico: admin, member, reader.
- Endpoint: URL para realizar peticiones REST.
- Catálogo: Lista de servicios (componentes de OpenStack) y sus respectivos endpoints.

KEYSTONE: ARCHIVO RC

- Contiene variables de entorno que permiten conectarse vía CLI a una nube OpenStack.
- El cliente puede instalarse de la siguiente manera en distribuciones RHEL:
 - `sudo dnf install -y python3-openstackclient`

```
export OS_USERNAME=gerardo.correa  
export OS_PASSWORD='myPassW0rd'  
export OS_PROJECT_NAME=my-project  
export OS_USER_DOMAIN_NAME=IDM  
export OS_PROJECT_DOMAIN_NAME=Default  
export OS_AUTH_URL=http://horizon.my.org:13000
```



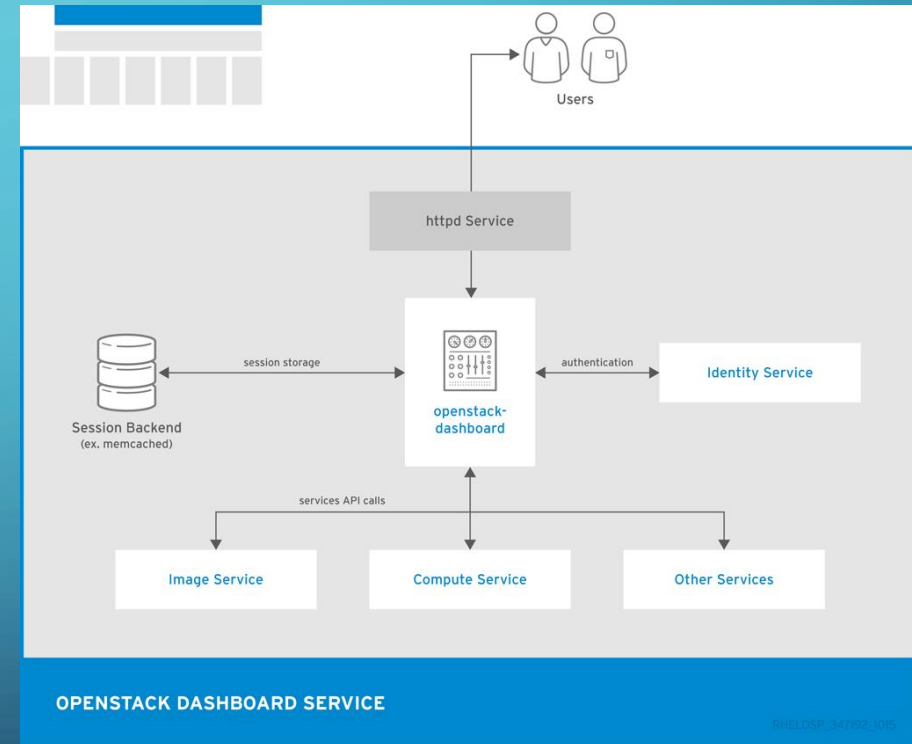
openstack®

ACTIVIDAD 1

- Instala el cliente CLI de OpenStack en tu laptop.
- Crea el archivo RC para el usuario que se te proporcionó.

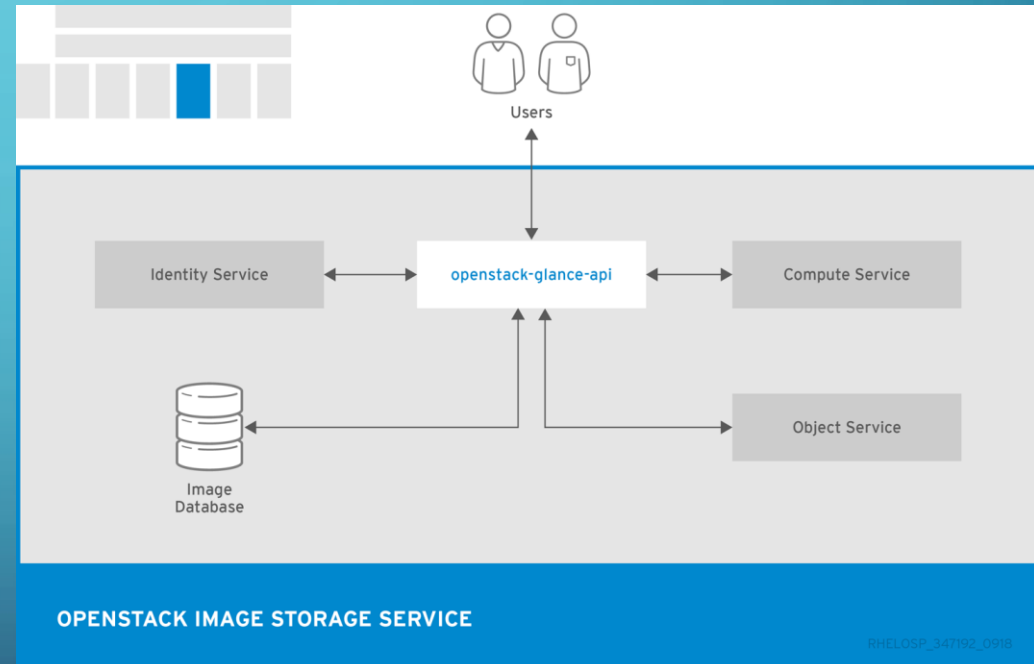
HORIZON

- Interfaz web para administración de OpenStack



GLANCE

- Repositorio de imágenes de discos virtuales.
- Soporta los siguientes formatos:
 - aki/ami/ari (Amazon kernel, ramdisk, or machine image)
 - iso (archive format for optical discs, such as CDs)
 - qcow2 (Qemu/KVM, supports Copy on Write)
 - raw (unstructured format)
 - vhd (Hyper-V, common for virtual machine monitors from vendors such as VMware, Xen, Microsoft, and VirtualBox)
 - vdi (Qemu/VirtualBox)
 - vmdk (VMware)



GLANCE

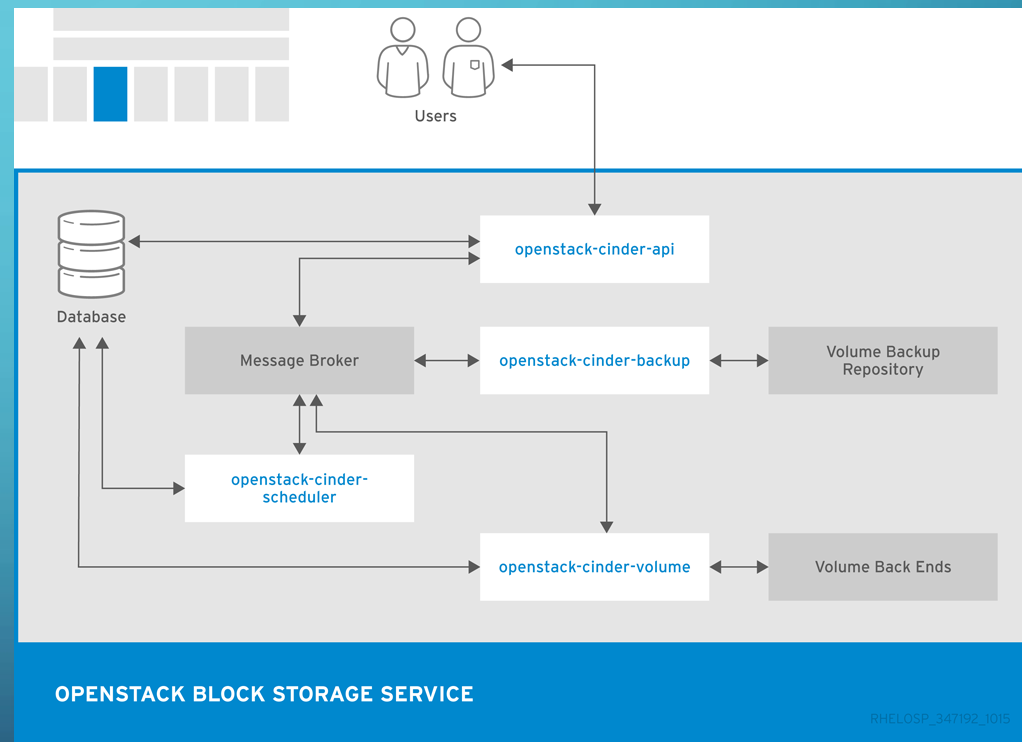
- Imagen: Archivo que contiene un disco virtual booteable, listo para ejecutar una instancia.
- Instancia: Servidor virtual resultado de la “ejecución” de una imagen.
- Por lo general contienen software adicional como cloud-init.
- cloud-init: Método estandarizado multiplataforma para inicialización de instancias:
- <https://cloudinit.readthedocs.io/en/latest/>
- Lista de imágenes listas para usarse en OpenStack:
<https://docs.openstack.org/image-guide/obtain-images.html>



@gcorreav

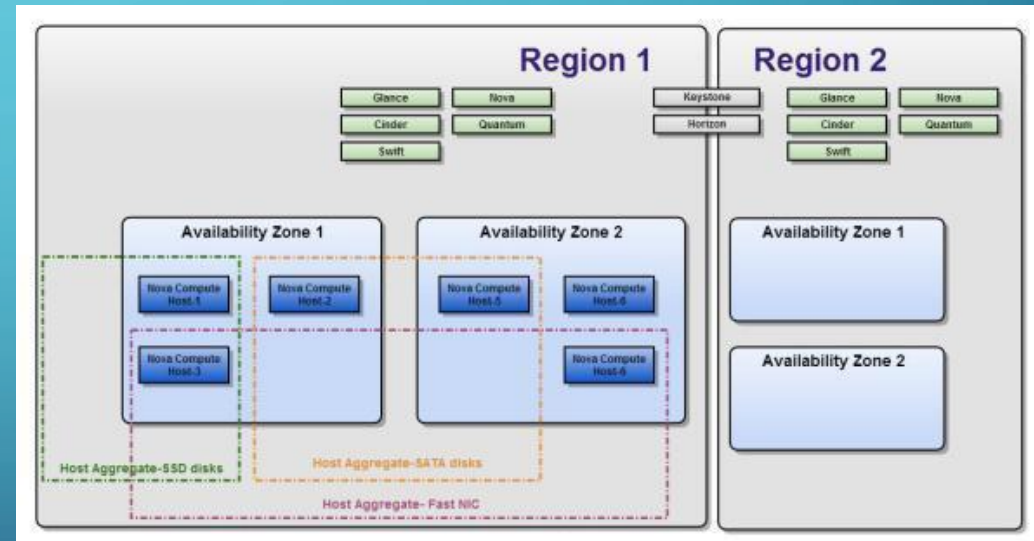
CINDER

- Provee dispositivos de bloque para discos virtuales.
- Los snapshots dependen del driver del almacenamiento subyacente.
- Además de crear y eliminar dispositivos de bloque, este servicio permite conectar y desconectarlos de instancias.



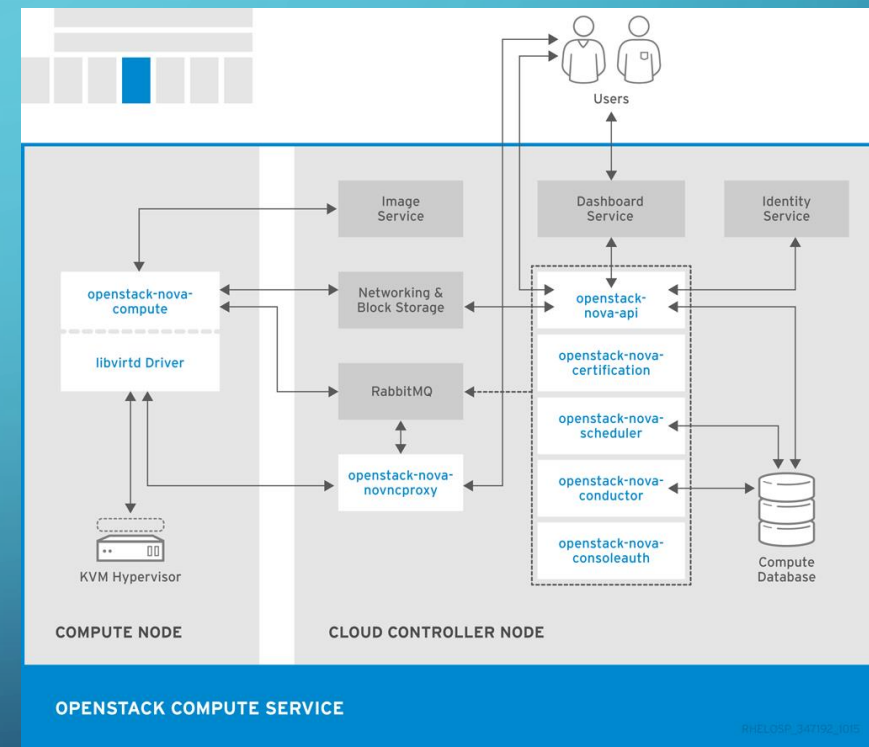
NOVA

- Region: Despliegue completo de OpenStack (API endpoints, redes, nodos compute).
- Availability Zone: Grupo lógico de nodos compute dentro de una región.
- Host Aggregate: Agrupación de nodos compute de acuerdo con características específicas (declaradas como metadatos). Sólo son visibles a los administradores de OpenStack



NOVA

- Red Hat soporta el driver libvirt que usa como hypervisor KVM.
- Key Pair: Llave pública SSH para inyectar a las instancias.





openstack®

ACTIVIDAD 2 – PARTE I

- En Horizon, crea un volumen de 2 GB
- Crea una instancia usando como imagen cirros (5 GB de disco booteable).
- Conecta el volumen creado a la instancia de Cirros, asígnale un sistema de archivos, móntalo en el sistema, escribe un archivo de texto plano.
- Desmonta el disco del sistema operativo y crea un snapshot del volumen.



@gcorreav



openstack®

ACTIVIDAD 2 – PARTE II

- En Horizon, monta el volumen.
- Borra el archivo del volumen y desmonta el volumen del sistema operativo.
- Crea un volumen a partir del snapshot. Desconecta de la instancia el volumen que tiene y conecta el nuevo volumen. Comprueba que el archivo original aún existe.
- Agrega tu key pair al proyecto.



@gcorreav

NEUTRON

- La red en OpenStack está definida por software SDN: (Software-Defined Networking), debido a esto puede reaccionar en tiempo real a las necesidades.
- Algunas de las ventajas de las redes en de OpenStack son:
 - Los usuarios pueden crear redes, conectar servidores y otros dispositivos a una o más redes.
 - Las direcciones IP pueden ser flotantes.

NEUTRON

- Dentro de OpenStack, neutron puede implementarse con varios drivers y plugins. Ejemplos de ellos son:
 - Open vSwitch
 - OpenDaylight
 - Juniper Opencontrail
 - VMware NSX Plugin
 - Cisco Nexus 1000V
- Una lista de drivers y plugins disponibles puede encontrarse en:
- [https://www.openstack.org/marketplace/drivers/#project=neutron%20\(networking\)](https://www.openstack.org/marketplace/drivers/#project=neutron%20(networking))



@gcorreav

NEUTRON: COMPONENTES

- Network agent
 - Servicio que se ejecuta en cada nodo OpenStack para realizar localmente configuración de red.
- neutron-dhcp-agent
 - Agente que provee servicios de DHCP a las redes de proyecto (tenant networks).
- neutron-ml2
 - Plug-in que administra drivers de red y provee servicios de routing y switching.
- neutron-server
 - Demonio de Python que administra las peticiones de usuarios y expone las APIs de redes. Algunos plug-ins como openvswitch y linuxbridge usan mecanismos de red propios de Linux, mientras que otros interactúan con dispositivos externos o controladores SDN.
- neutron
 - Línea de comando para acceder a la API

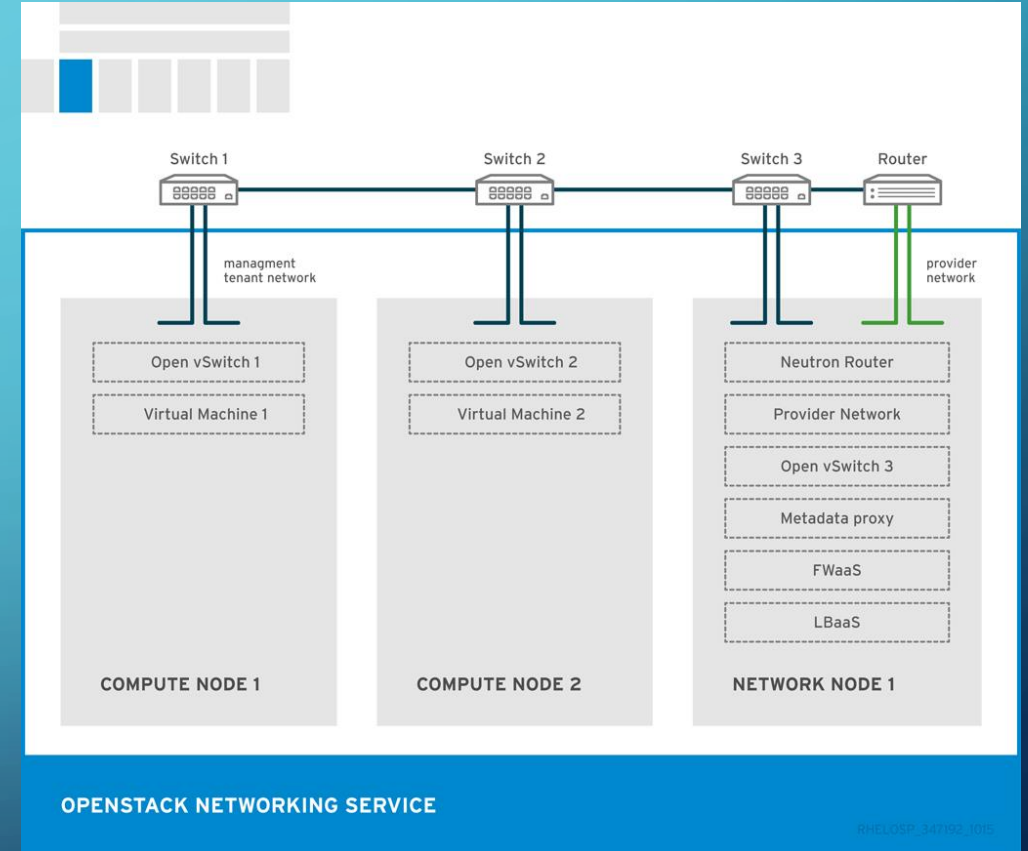
NEUTRON (EJEMPLO: OVS)

- 2 nodos compute y 1 nodo de red.
- Los nodos compute ejecutan el agente ovs.
- Mientras que el nodo de red realiza las siguientes funciones:

L3 routing

DHCP

NAT y servicios como FWaaS y LBaaS



NEUTRON: AGENTES

- Agente L2
- Un agente L2 provee conectividad de red a nivel de capa 2 a los recursos de OpenStack, típicamente se ejecuta en nodos de red (network nodes) y en cada nodo compute. Ejemplos:
 - Open vSwitch agent
 - Linux bridge agent
 - SRIOV Nic Switch agent
 - MacVTap agent
- Agente L3
- Un agente L3 ofrece servicios avanzados de capa 3, por ejemplo routers virtuales y direcciones IP flotantes. Requiere de un agente L2 ejecutándose en paralelo.
- Agente DHCP
- El agente DHCP se encarga de los servicios DHCP (Dynamic Host Configuration Protocol) y RADVD (Router Advertisement Daemon). Requiere de un agente L2 en el mismo nodo.



@gcorreav

NEUTRON: AGENTES

- Agente Metadata
 - El agente Metadata permite a las instancias acceder a metadatos de cloud-init y datos de usuarios a través de la red. Requiere de un agente L2 ejecutándose en el mismo nodo.
- Agente L3 metering
 - El agente L3 metering permite la medición de tráfico de capa 3. Requiere de un agente L3 ejecutándose en el mismo nodo.
- Más información: <https://docs.openstack.org/neutron/zed/admin/config-m12.html#l2-agent>



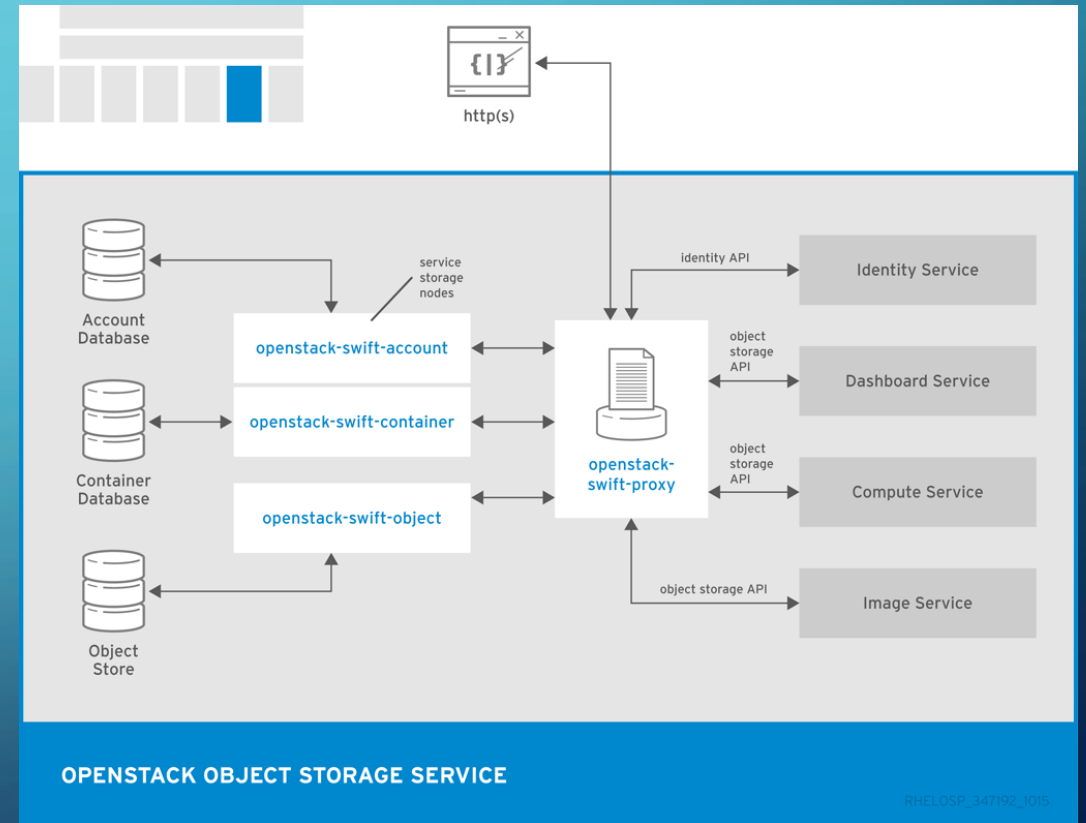
@gcorreav

NEUTRON: REDES

- Redes de proyecto (tenant/Project networks): Creadas por usuarios dentro del contexto del proyecto.
- Redes externas (external networks): Creadas por el administrador de la nube, se utilizan para la comunicación de las instancias con elementos fuera del contexto del proyecto.
- Router este-oeste: Router que interconecta redes de proyecto.
- Router norte-sur: Router que interconecta una red de proyecto con una red externa.

SWIFT

- Sistema de almacenamiento de objetos, accesible vía HTTP. Usado comúnmente para almacenar datos como videos, imágenes o imágenes de máquinas virtuales.
- Soporta escalamiento horizontal así como réplica de datos basada en software.

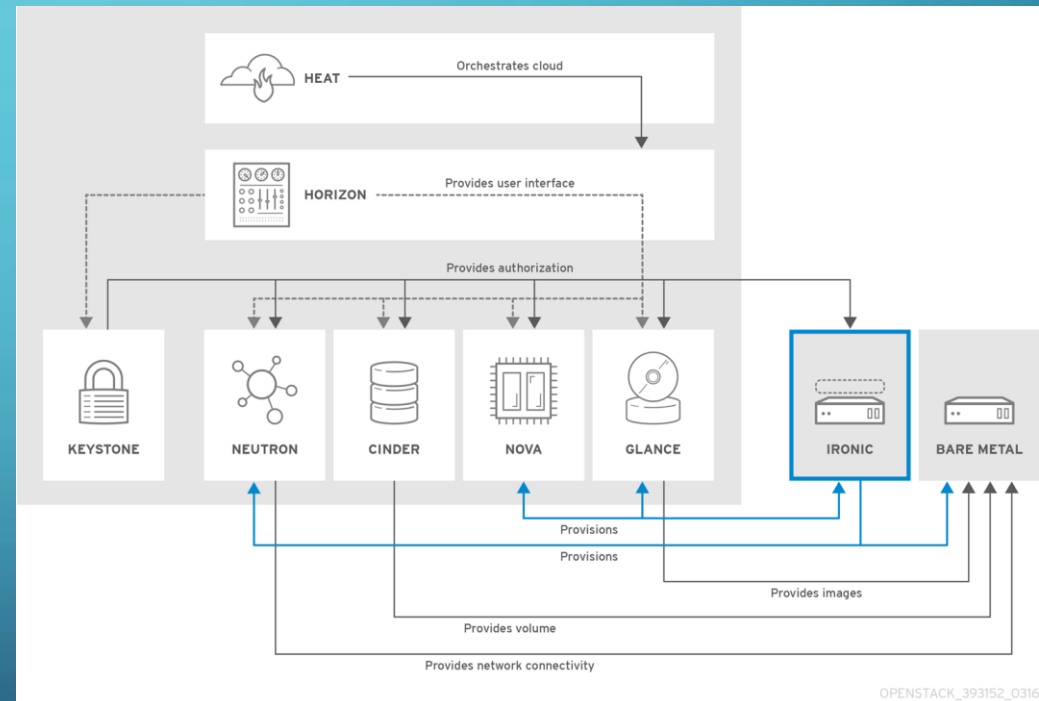


SWIFT

- Se recomienda un mínimo de 3 réplicas.
- Zonas: Se aseguran de que cada réplica de un objeto se almacene de manera separada. Puede ser desde un disco, un arreglo, un servidor, un rack o un datacenter inclusive.
- Regiones: Grupos de zonas por ubicación geográfica. Cada región tiene un endpoint independiente.
- Account: Proyecto
- Container: “Espacio” que almacena objetos.

IRONIC

- Permite el aprovisionamiento bare metal a través de drivers de diversos fabricantes.
- Este servicio se integra con Nova para aprovisionar máquinas bare metal de la misma manera que aprovisiona máquinas virtuales.





openstack®

ACTIVIDAD 3

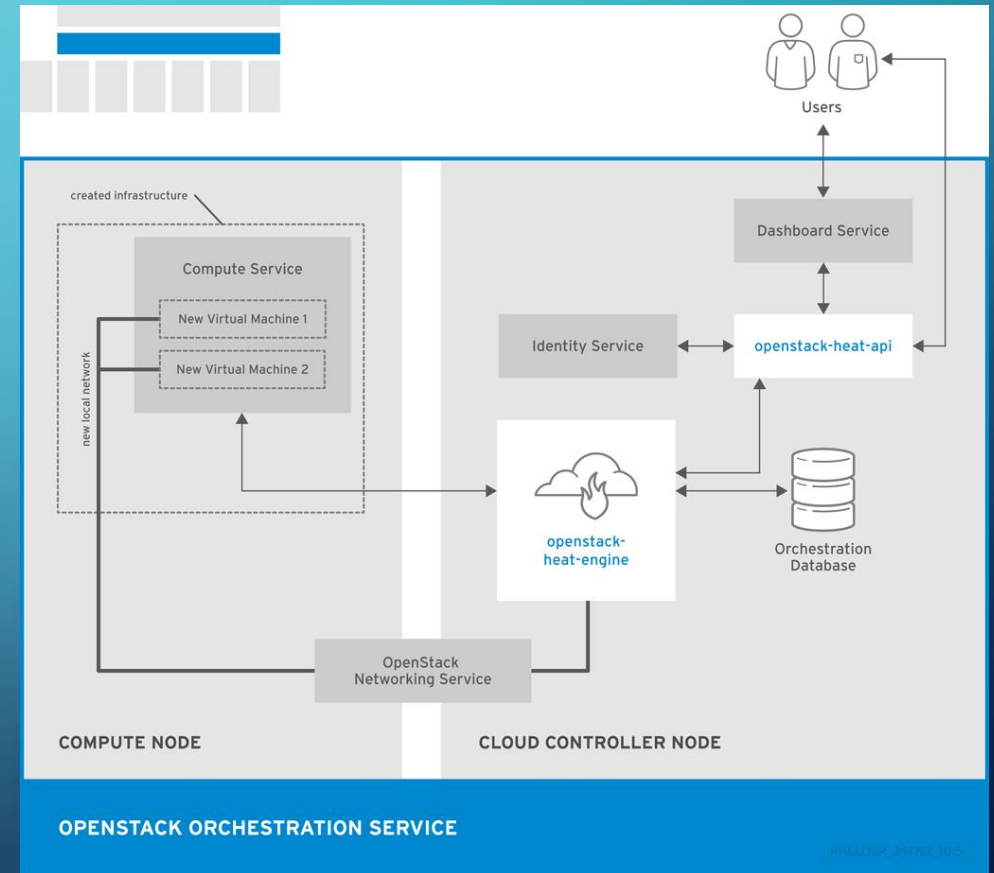
- En Horizon, crea un security group que contemple los servicios HTTP y HTTPS.
- Aloja una dirección IP flotante en el proyecto.
- Crea 2 redes y conéctalas mediante un router E-W.



@gcorreav

HEAT: ORQUESTACIÓN

- Provee plantillas para crear y administrar recursos como almacenamiento, redes, instancias o aplicaciones. Las plantillas se usan para crear stacks.
- Stack: Conjunto de recursos.
- HOT: Heat Orchestration Template.



HEAT: VENTAJAS

- Una plantilla provee acceso a todas las APIs subyacentes.
- Las plantillas son modulares y orientado a recursos.
- Las plantillas se pueden definir de manera recursiva, y usarse como stacks anidados.

HEAT: HOT

- Las partes que componen a una plantilla HOT son:
 - `heat_template_versión`: obligatorio - Versión de HOT (estructura) que se usará.
 - `description`: opcional – Descripción de la plantilla.
 - `parameters`: opcional – Usado para proporcionar valores cuando se utiliza la plantilla.
 - `resources`: obligatorio – Debe definirse al menos un recurso.
 - `outputs`: opcional – Usado para proporcionar salidas (datos) al usuario cuando se utiliza la plantilla.
 - `conditions`: opcional – Usado para restringir un recurso o una propiedad. Pueden asociarse con recursos o sus propiedades.
- Más información en: https://docs.openstack.org/heat/latest/template_guide/hot_guide.html



openstack®

ACTIVIDAD 4

- En Horizon, crea una instancia RHEL en la red DMZ, con flavor small, security group que creaste y tu key pair; usa cloud-init para asignar una contraseña al usuario cloud-user y ponerle una suscripción RHEL.
- Asígnale a tu instancia la dirección IP flotante que alojaste anteriormente. Accede por SSH usando la dirección IP flotante.
- Instala en tu instancia Apache y presenta el “it works” por default.



@gcorreav



openstack®

ACTIVIDAD 5

- En Horizon, crea un stack que cree una instancia de cirros, con 5GB de disco booteable.



@gcorreav



openstack®

ACTIVIDAD 6

- Crea un stack que cree una instancia RHEL, con tu key pair, m1.medium, interfaces en DMZ y una de las redes a la que creaste. Usa cloud-init para definir la contraseña del usuario cloud-user, cambia el teclado a español latinoamericano y asígnale una suscripción RHEL.
- Haz que esta instancia pueda hacer ping a la instancia cirros creada anteriormente. Consejo: Analiza tu topología de red.



@gcorreav



openstack®

ACTIVIDAD 7

- Crea un stack con parámetros, con valores por default de imagen RHEL 8.5, tu key pair, flavor m1.medium, red en DMZ. Usa cloud-init para definir la contraseña del usuario cloud-user, cambiar el teclado a español latinoamericano y asígnale una suscripción RHEL.



openstack®

ACTIVIDAD 8

- Crea un stack con parámetros, con valores por default de imagen RHEL 8.5, tu key pair, flavor m1.medium, red en DMZ. Usa cloud-init para definir la contraseña del usuario cloud-user, cambia el teclado a español latinoamericano, y asígnale una suscripción RHEL, además, aloja una dirección IP flotante al proyecto y asígnala a la instancia creada.



@gcorreav