

# OPENSTACK Y PROXMOX

JOSÉ DOMINGO MUÑOZ RODRÍGUEZ

IES GONZALO NAZARENO

ABRIL 2022



- **José Domingo Muñoz Rodríguez**
- Doy clases en el IES Gonzalo Nazareno (Dos Hermanas)
- Correo: [josedom24@josedomingo.org](mailto:josedom24@josedomingo.org)
- Web: [www.josedomingo.org](http://www.josedomingo.org)
- Twitter: [pledin\\_jd](#)

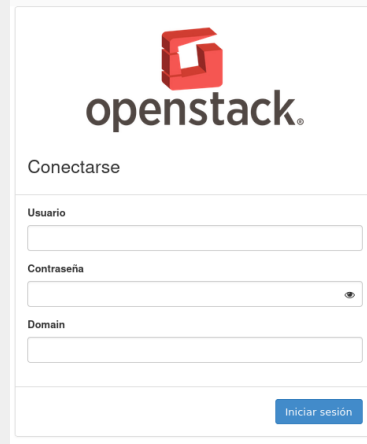


# INTRODUCCIÓN A OPENSTACK



# CONCEPTOS PREVIOS. HORIZON.

- **Horizon:** Aplicación web escrita en python/django para la gestión de OpenStack.
- Acceso mediante usuario/contraseña.
- El **Dominio** nos permite escoger los usuarios del LDAP.
- Podemos realizar muchas operaciones:
  - ▶ Gestionar instancias
  - ▶ Gestionar claves ssh
  - ▶ Gestionar volúmenes
  - ▶ Gestionar redes
  - ▶ Gestionar el cortafuego de la instancia
  - ▶ Gestionar la IP flotantes para acceder a las instancias
  - ▶ ...
- Otras operaciones es necesario el uso del CLI (cliente de terminal):openstack-client.



The screenshot shows the OpenStack Horizon login interface. At the top is the OpenStack logo, which consists of a red 3D cube icon followed by the word "openstack" in a lowercase, sans-serif font. Below the logo is the heading "Conectarse". There are three input fields: "Usuario" (User), "Contraseña" (Password), and "Domain". The "Contraseña" field has a small eye icon on the right side to toggle password visibility. At the bottom right of the form is a blue button with the text "Iniciar sesión" (Log in).

## CONCEPTOS PREVIOS. INSTANCIAS

- **Imagen:** Imagen de sistema preconfigurado que se utiliza como base para crear instancias. [Get images](#). Normalmente ficheros qcow2 o img. El componente que gestiona las imágenes es **Glance**.
- **Instancia:** Clon de una imagen que se crea a demanda del usuario en uno de los nodos de computación del cloud.
- Las instancias son gestionadas por el componente **Nova** de Openstack.

### ¿Cómo accedemos a las instancias?

- Se puede usar distintos protocolos (ssh,RDP,...).
- **Par de claves ssh:** Utilizadas para acceder por ssh a las instancias desde fuera del cloud.



## CONCEPTOS PREVIOS. CONFIGURACIÓN DE LA INSTANCIA

- Al crear una instancia se elige el **Sabor**: Que nos determina el nº de vCPU, RAM y HD de la máquina.
- Al crear una instancia desde una imagen se ejecuta **Cloud-init** que es el programa encargado de configurar la máquina.

### ¿Qué se configura?

- El hostname de la máquina.
- Se inyecta la clave pública para que podamos acceder por ssh.
- Se configura la red.
- Se puede configurar las contraseñas de los usuarios.
- ...



## CONCEPTOS PREVIOS. REDES.

Las instancias se crean conectas a redes privada definidas por el usuario.

- La instancia toma por DHCP una **IP fija** en el direccionamiento de la red interna.
- Para acceder desde el exterior asignamos a la instancia una **IP flotante** en el direccionamiento de la red externa.
- La **IP flotante** se configura como una regla DNAT en el router que une la red interna con la externa.
- Cada interface de red tiene un cortafuego que controlamos con el **Grupo de Seguridad** donde podemos abrir distintos puertos y protocolos.
- Las redes son gestionadas por el componente **Neutron** de Openstack.



# DEMO 1: CREACIÓN Y GESTIÓN DE UNA INSTANCIA





- La instancia que hemos creado en la DEMO 1 pierde toda la información cuando la eliminamos.
- Si queremos trabajar con información persistente necesitamos usar los **Volúmenes**.
- Los volúmenes son gestionados por el componente **Cinder** de Openstack.
- Varias funcionalidades:
  - ▶ Creación de volúmenes que podemos conectar a una instancia para guardar información.
  - ▶ Creación de instancias sobre volúmenes. Si eliminamos la instancia la información no se pierde.
  - ▶ ...



# **DEMO 2: CREACIÓN DE UNA INSTANCIA SOBRE UN VOLUMEN. TRABAJO CON VOLÚMENES.**



# INTRODUCCIÓN A PROXMOX



# CONCEPTOS PREVIOS

- Proxmox nos permite gestionar de forma sencilla virtualizar máquinas virtuales y LXC.
- Podemos gestionar el almacenamiento (volúmenes) con los que trabaja cada MV.
- Podemos crear distintos linux bridge para conectar las MV en redes internas.

## Definiciones

- **Plantillas:** Podemos convertir una MV en una plantilla desde la que crearemos nuevas MV mediante un proceso de clonación.
- **Clonación:** Tenemos dos tipos:
  - ▶ **Completa:** Se crea una nuevo medio de almacenamiento para la nueva MV copiando el original.
  - ▶ **Ligera:** Se crea el almacenamiento de la nueva máquina con aprovisionamiento ligero.
- **Pool de recursos:** Nos permite agrupar MV, LXC y almacenamiento. No se puede agrupar las redes.

# ¿QUÉ QUERÍAMOS CONSEGUIR?

- Queremos crear usuario en proxmox que controlen sus recursos (MV, LXC, almacenamiento).
- Una limitación es que las redes no la pueden controlar los usuarios.
- Aunque siempre un usuario puede crear una MV desde una ISO, queremos que los usuarios creen de forma rápida nuevas MV. Para ello podrán clonar plantillas que ya tenemos predefinidas.

## ¿Cómo podemos conseguir qué cada usuario controle sus recursos?

- **Permisos:** Nos permite asignar a un recurso y a un usuario un conjunto de permisos (rol). De esta manera podemos controlar las operaciones que puede hacer un usuario sobre un recurso.



# ¿CÓMO PODEMOS CONSEGUIRLO?

- Los usuarios se pueden agrupar en **Grupos**. Por ejemplo, podemos crear grupos por cursos (asir1) o simplemente creamos un grupo alumnos.
- Vamos a crear distintos **Pools de recursos**:
  - ▶ Un pool que se asigna a cada usuario. Nosotros lo hemos llamado Proyecto de usuario. Cada usuario creara sus recursos en su pool.
  - ▶ Un pool que hemos llamado Imágenes, donde vamos a guardar las plantillas que los usuarios pueden clonar.
- Vamos a crear dos nuevos roles (conjunto de permisos): **rol\_proyecto**: rol para asignar a los pools de los usuarios, y **rol\_imágenes**: rol para asignar al pool Imágenes.



## ■ Rol **rol\_proyecto**:

Datastore.AllocateSpace Datastore.Audit Permissions.Modify Pool.Audit Sys.Audit Sys.Console Sys.Modify Sys.Syslog VM.Allocate VM.Audit VM.Backup VM.Clone VM.Config.CDROM  
VM.Config.CPU VM.Config.Cloudinit VM.Config.Disk VM.Config.HWType VM.Config.Memory VM.Config.Network VM.Config.Options VM.Console VM.Migrate VM.Monitor  
VM.PowerMgmt VM.Snapshot VM.Snapshot.Rollback

## ■ Rol **rol\_imágenes**:

Pool.Audit VM.Audit VM.Clone



# ASIGNACIÓN DE PERMISOS

- A cada Pool de cada usuario le asignamos el rol **rol\_proyecto**.
- Al grupo Imágenes le asignamos el rol **rol\_imágenes**.
- en nuestro caso los roles se llaman iesgn y iesgn-template-clone.

Path ↑	User/Group/API Token	Role
/	admin@pve	Administrator
/pool/Imagenes	@profesores-ldap	iesgn-template-clone
/pool/Imagenes	@asir1-ldap	iesgn-template-clone
/pool/Imagenes	@asir2-ldap	iesgn-template-clone
/pool/Proyecto_...@ldap	...@ldap	iesgn

