

# OpenStack Client

Gestión de OpenStack desde la línea de comandos

 José Domingo Muñoz

 IES Gonzalo Nazareno · Dos Hermanas

 IV · Infraestructura Virtual

IV · OPENSTACK CLIENT (OSC)

01

# Instalación y configuración

---

OpenStack Client (OSC)

## ¿Qué es OpenStack Client?

" Además de usar *Horizon* (interfaz web), podemos gestionar OpenStack con un **cliente de línea de comandos**: *OpenStack Client (OSC)*.

### INSTALACIÓN CON ENTORNO VIRTUAL PYTHON

```
python3 -m venv os
source os/bin/activate
pip install python-openstackclient
```

### INSTALACIÓN CON APT

```
apt install python3-openstackclient
```

## Fichero de credenciales (RC file)

Para autenticarnos en nuestro proyecto OpenStack necesitamos el **fichero de credenciales**.

### DESCARGAR DESDE HORIZON

**Tu usuario → Fichero OpenStack RC**

Guarda las credenciales de tu proyecto en variables de entorno.

### CARGAR LAS CREDENCIALES

```
source "Proyecto de josedom-openrc.sh"  
# Introduce tu contraseña cuando se solicite
```

Una vez cargado, ya puedes usar el comando

```
openstack .
```

## Comandos básicos de consulta

### LISTAR RECURSOS DEL PROYECTO

```
openstack server list      # Máquinas virtuales
openstack keypair list    # Claves SSH
openstack image list      # Imágenes disponibles
openstack network list    # Redes
openstack flavor list     # Sabores (tipos)
```

### GRUPOS DE SEGURIDAD

```
# Ver reglas del grupo default
openstack security group rule list default

# Abrir puerto 22 (SSH)
openstack security group rule create \
  --protocol tcp \
  --remote-ip 0.0.0.0/0 \
  --dst-port 22 default
```

IV · OPENSTACK CLIENT (OSC)

# 02

## Gestión de instancias

---

Crear, operar y configurar máquinas virtuales

## Conceptos previos

### RECURSOS DE CÓMPUTO

- **Imagen:** Plantilla con SO base para clonar instancias
- **Instancia:** Servidor virtual creado a partir de una imagen
- **Sabor** (*flavor*): Define vCPU, RAM y disco de la instancia

### CONECTIVIDAD Y SEGURIDAD

- **IP fija:** Dirección interna, no cambia durante la vida de la instancia
- **IP flotante:** Dirección externa asignable/designable
- **Grupo de Seguridad:** Reglas de cortafuegos a nivel de instancia

## Crear una instancia

```
openstack server create \  
  --flavor m1.normal \  
  --image "Debian 13 Trixie" \  
  --security-group default \  
  --key-name clave_jdmr \  
  --network "Red de josedom" \  
  instancia_prueba
```

 Todos los parámetros son obligatorios: sabor, imagen, grupo de seguridad, clave SSH y red.

# Gestión del ciclo de vida

## OPERACIONES BÁSICAS

```
openstack server list
openstack server show instancia_prueba
openstack server start instancia_prueba
openstack server stop instancia_prueba
openstack server delete instancia_prueba
```

## REINICIO Y AHORRO DE RECURSOS

```
# Soft reboot
openstack server reboot instancia_prueba

# Hard reboot
openstack server reboot --hard instancia_prueba

# Suspender / Reanudar
openstack server suspend instancia_prueba
openstack server resume instancia_prueba

# Hibernar / Reactivar
openstack server shelve instancia_prueba
openstack server unshelve instancia_prueba
```

## IP flotantes

Las **IP flotantes** permiten acceder a las instancias desde el exterior mediante **DNAT** en el router.

```
# Ver IPs flotantes reservadas
openstack floating ip list

# Solicitar una nueva IP flotante
openstack floating ip create ext-net

# Asignar a una instancia
openstack server add floating ip instancia_prueba 172.22.201.111

# Quitar IP flotante
openstack server remove floating ip instancia_prueba 172.22.201.111
```

## cloud-init: configuración automática

" *cloud-init* es el estándar para configurar instancias en el arranque. Está instalado en las imágenes para IaaS.

### TIPOS DE DATOS RECIBIDOS

- **Metadatos:** Características de la instancia — obtenidos del servidor de metadatos ( 169.254.169.254 )
- **User-data:** Configuración adicional proporcionada por el usuario
- **Vendor-data:** Datos adicionales del proveedor

### PASAR USER-DATA AL CREAR

```
openstack server create \  
  --flavor m1.normal \  
  --image "Debian 13 Trixie" \  
  --security-group default \  
  --key-name clave_jdmr \  
  --network "Red de josedom" \  
  --user-data cloud-config.yaml \  
  instancia_prueba
```

## Ejemplo de cloud-config

```
#cloud-config
package_update: true
package_upgrade: true
packages:
  - apache2
fqdn: maquina.example.org
hostname: maquina
manage_etc_hosts: true
users:
  - name: debian
    sudo: ALL=(ALL) NOPASSWD:ALL
    groups: users, admin
    shell: /bin/bash
    ssh-authorized-keys:
      - ssh-rsa AAAA...
chpasswd:
  expire: False
  users:
    - {name: debian, password: asdasd, type: text}
```

## cloud-init y redes sin DHCP

⚠ Si la instancia se conecta a una red **sin DHCP**, no podrá acceder al servidor de metadatos y cloud-init fallará.

**Solución:** usar `--config-drive True` para que cloud-init lea la configuración desde un **CDROM virtual**.

```
openstack server create \  
  --flavor m1.mini \  
  --image "Debian 13 Trixie" \  
  --security-group default \  
  --key-name clave_jdmr \  
  --network red_sin_dhcp \  
  --config-drive True \  
  instancia_prueba
```

# Snapshots de instancias

" *Un snapshot de instancia es una imagen en Glance creada a partir del disco raíz de la instancia.*

## CREAR Y CONSULTAR

```
# Crear snapshot
openstack server image create \
  --name snapshot-instancia \
  instancia_prueba

# Listar imágenes (ver estado)
openstack image list

# Ver detalles
openstack image show snapshot-instancia

# Eliminar
openstack image delete snapshot-instancia
```

## CREAR INSTANCIA DESDE SNAPSHOT

```
openstack server create \
  --flavor m1.normal \
  --image "snapshot-instancia" \
  --security-group default \
  --network "Red de josedom" \
  instancia_prueba2
```

IV · OPENSTACK CLIENT (OSC)

# 03

## Almacenamiento en bloque

---

Gestión de volúmenes con Cinder

## Conceptos de almacenamiento en bloque

### ¿QUÉ ES CINDER?

- Componente de OpenStack para gestión de **almacenamiento en bloque** (SAN virtual)
- Equivalente a **Amazon EBS**

### ¿QUÉ ES UN VOLUMEN?

- **Dispositivo de bloques** asociable/desasociable a instancias
- Proporciona **almacenamiento permanente**, independiente de la vida de la instancia
- Equivalente a una *unidad lógica en SAN*

## Operaciones básicas con volúmenes


```
# Crear un volumen de 1 GiB
openstack volume create --size 1 mi_disco1

# Listar volúmenes
openstack volume list

# Asociar a una instancia
openstack server add volume \
  --device /dev/sdb instancia_prueba mi_disco1

# Desasociar de la instancia
openstack server remove volume instancia_prueba mi_disco1

# Eliminar el volumen
openstack volume delete mi_disco1
```

 No se puede eliminar un volumen mientras está asociado a una instancia.

## Instancia con disco raíz en volumen

*" Al usar un volumen como disco raíz, la instancia persiste aunque la eliminemos — el sistema está en el volumen.*

```
# 1. Crear volumen arrancable desde una imagen
openstack volume create \
  --bootable --size 8 \
  --image "Debian 13.0 - Trixie" \
  disco_debian

# 2. Crear instancia usando el volumen (sabor con 0 disco)
openstack server create \
  --flavor vol.mini \
  --volume disco_debian \
  --security-group default \
  --key-name clave_jdmr \
  --network "red de josedom" \
  instancia_prueba2
```

## Snapshots de volúmenes

 Desasocia el volumen antes de hacer el snapshot para evitar corrupción.

```
# Desasociar y crear snapshot
openstack server remove volume instancia_prueba mi_disco1
openstack volume snapshot create --volume mi_disco1 copia_mi_disco1

# Listar snapshots
openstack volume snapshot list

# Crear nuevo volumen desde el snapshot
openstack volume create --snapshot copia_mi_disco1 disco2

# Borrar: primero el snapshot, luego el volumen
openstack volume snapshot delete copia_mi_disco1
openstack volume delete mi_disco1
```

## Extender el tamaño de un volumen

```
openstack volume set --size 2 disco2
```

 Tras ampliar el volumen, también hay que redimensionar el sistema de archivos desde dentro de la instancia.

IV · OPENSTACK CLIENT (OSC)

# 04

## Redes en OpenStack

---

Gestión de redes NAT con Neutron

## Conceptos de red en OpenStack (Neutron)

### ELEMENTOS DE RED

- **Red:** Dominio aislado de capa 2 (equivale a una VLAN)
- **Subred:** Bloque de direcciones IPv4/IPv6
- **Router:** Dispositivo de capa 3 para interconectar redes
- **Puerto:** Puerto virtual de un router o instancia

### TIPOS DE IP

- **IP Fija:** Dirección interna, no cambia durante la vida de la instancia
- **IP Flotante:** Dirección externa asignable/desasignable para acceso desde el exterior

## Gestión de routers

Los routers interconectan redes. Conectados a la red externa permiten **SNAT** (acceso a internet) y **DNAT** (IP flotantes).

```
# Listar routers
openstack router list

# Crear router y conectarlo a la red externa
openstack router create mi_router
openstack router set mi_router --external-gateway ext-net

# Ver información y puertos
openstack router show mi_router
openstack port list --router mi_router

# Quitar puerta de enlace y eliminar
openstack router unset mi_router --external-gateway
openstack router delete mi_router
```

## Crear una red NAT

Una **red NAT** es una red interna conectada a un router que tiene salida al exterior.

```
# Crear la red
openstack network create red1

# Crear subred con DHCP (parámetros habituales)
openstack subnet create \
  --network red1 \
  --subnet-range 192.168.0.0/24 \
  --dns-nameserver 172.22.0.1 \
  subred1

# Conectar la subred al router
openstack router add subnet mi_router subred1

# Listar redes y subredes
openstack network list
openstack subnet list
```

## Parámetros para crear subredes

PARÁMETRO	OBLIGATORIO	DESCRIPCIÓN
<code>--network</code>	Sí	Red a la que se asocia la subred
<code>--subnet-range</code>	Sí	Direccionamiento CIDR
<code>--gateway</code>	No	Puerta de enlace (por defecto: primera IP del rango)
<code>--no-dhcp</code>	No	Desactivar servidor DHCP
<code>--allocation-pool</code>	No	Rango de IPs repartidas por DHCP
<code>--dns-nameserver</code>	Sí*	Servidor DNS enviado a las instancias

\* El DNS es necesario aunque no haya DHCP, para que cloud-init configure la red estáticamente.

## Instancia en red NAT con DHCP

```
openstack server create \  
  --flavor m1.mini \  
  --image "Debian 13 Trixie" \  
  --security-group default \  
  --key-name jdmr \  
  --network red1 \  
  maquina1  
  
# Asignar IP flotante  
openstack floating ip create ext-net  
openstack server add floating ip maquina1 <IP_FLOTANTE>  
  
# Ver puertos de la instancia  
openstack port list --server maquina1
```

cloud-init configura la interfaz **por DHCP** → se puede verificar en la configuración de **netplan**.

## Reserva de IP con puertos (red con DHCP)

Si necesitamos que una instancia tenga una **IP fija determinada**, creamos un puerto con esa dirección.

```
# Crear puerto con IP fija (reserva)
openstack port create \
  --network red1 \
  --fixed-ip ip-address=192.168.0.100 \
  mi_port

# Crear instancia usando el puerto
openstack server create \
  --flavor m1.mini \
  --image "Debian 13 Trixie" \
  --security-group default \
  --key-name jdmr \
  --port mi_port \
  maquina2
```

## Red NAT sin DHCP

```
# Crear red sin DHCP
openstack network create red2
openstack subnet create \
  --network red2 \
  --no-dhcp \
  --subnet-range 192.168.10.0/24 \
  --dns-nameserver 172.22.0.1 \
  subred2

# Conectar al router
openstack router add subnet mi_router subred2
```

## Red NAT sin DHCP

### SIN DHCP → CLOUD-INIT CONFIGURA ESTÁTICAMENTE

cloud-init usa el **DNS indicado en la subred** para configurar la red de forma estática en netplan.

### CONFIG DRIVE OBLIGATORIO

Sin DHCP no hay acceso al servidor de metadatos → hay que usar `--config-drive True`.

## Instancia en red NAT sin DHCP

```
# Con config-drive porque no hay servidor de metadatos
openstack server create \
  --flavor m1.mini \
  --image "Debian 13 Trixie" \
  --security-group default \
  --key-name jdmr \
  --network red2 \
  --config-drive True \
  maquina3

# IP fija estática con puerto
openstack port create \
  --network red2 \
  --fixed-ip ip-address=192.168.10.100 \
  mi_port_sin_dhcp

openstack server create ... --port mi_port_sin_dhcp --config-drive True maquina4
```

## Eliminación de infraestructura de red

El orden de eliminación es importante para evitar errores de dependencias.

### ORDEN CORRECTO

1. Eliminar todas las **instancias** conectadas a la red
2. Eliminar **puertos desasociados** de la red
3. Desconectar la red del router

```
openstack router remove subnet \  
mi_router subred1
```

4. Eliminar la **red** y el **router**

### ⚠ SI NO SIGUES EL ORDEN...

OpenStack devolverá errores porque no puede eliminar recursos con dependencias activas.

Por ejemplo: no puedes eliminar una red si tiene puertos activos o está conectada a un router.

IV · OPENSTACK CLIENT (OSC)

# ¡Gracias!

OpenStack Client (OSC)

 José Domingo Muñoz

 IES Gonzalo Nazareno · Dos Hermanas

 IV