

# Anexo 1 : Ficheros de instalación Scripts

A continuación se listarán los scripts utilizados para la instalación más cómoda de OpenStack. Van agrupados a efectos de que sea intuitiva y sencilla la instalación. Se ha dividido todo el proceso de la instalación en diferentes scripts para facilitar la instalación en diferentes arquitecturas, con un número diferente de servidores,etc..

En el caso que nos ocupa la arquitectura consta de 6 servidores, pero los scripts se pueden distribuir según fuera requerido en diferentes servidores:

Importante:

- Debe respetarse el orden de instalación. Si no se instala mysql antes de keystone no podrá funcionar (por poner un ejemplo).
- Es importante (no se realiza en los scripts) utilizar los nombres de los servidores con sus correspondientes ips y editar éstos en los ficheros (de todos los servidores) /etc/hosts. Es preferible usar nombres a ips.
- Como se ha indicado en la memoria, la segmentación en interfaces de red, e incluso que sean los interfaces de 10Gigaethernet es importante. La planificación del uso de los interfaces de red varía mucho, y por ello no incluye esta configuración en ningún script, aun cuando al final se copia la configuración del /etc/network/interfaces de cada servidor de este proyecto en cuestión.
- El uso de variables de entorno se usa mucho y es útil tener ficheros como "openrc" que se pueda invocar mediante "source openrc" a efectos de que estas variables estén disponibles dentro del shell para las aplicaciones.

## 1 Scripts generales

*Estos ficheros deben copiarse en todos los servidores con los mismos valores.*

### ○ Openrc (Fichero de Variables)

```
#!/bin/bash
# definicion de hosts,usuarios,passwords,etc...
HOST_IP=192.168.73.122
HOST_NOVA=192.168.73.128
HOST_CINDER=192.168.73.126
HOST_GLANCE=192.168.73.128
HOST_HEAT=192.168.73.128
HOST_KEYSTONE=192.168.73.128
HOST_QUANTUM=192.168.73.127
HOST_COMPUTE1=192.168.73.120
HOST_COMPUTE2=192.168.73.121
MYSQL_PASSWORD=mysql_password
MYSQL_NOVA_USER=nova
MYSQL_CINDER_USER=cinder
MYSQL_GLANCE_USER=glance
MYSQL_HEAT_USER=heat
MYSQL_KEYSTONE_USER=keystone
MYSQL_QUANTUM_USER=quantum
MYSQL_NOVA_PWD=nova_password
MYSQL_CINDER_PWD=cinder_password
MYSQL_GLANCE_PWD=glance_password
MYSQL_HEAT_PWD=heat_password
MYSQL_KEYSTONE_PWD=keystone_password
MYSQL_QUANTUM_PWD=quantum_password
#-----
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
```

```
export OS_PASSWORD=admin_password
export OS_AUTH_URL="http://$HOST_KEYSTONE:5000/v2.0/"
export OS_SERVICE_ENDPOINT="http://$HOST_KEYSTONE:35357/v2.0"
export OS_SERVICE_TOKEN=token_password
export SERVICE_TENANT_NAME="service"
export ADMIN_PASSWORD="admin_password"
SERVICE_PASSWORD="service_password"
DEMO_PASSWORD="demo_password"
export KEYSTONE_REGION=RegionOne
export KEYSTONE_HOST=$HOST_KEYSTONE
export NEUTRON_HOST=$HOST_NEUTRON
export CINDER_HOST=$HOST_CINDER
```

*Las variables necesarias por Openstack o algunas utilizadas por los scripts para las configuraciones de los ficheros y tablas.*

*Tipicamente se hará mediante "source openrc".*

*Nota: Tengamos presente que en muchos casos Openstack necesita variables de sesión para operar con los comandos en línea de nova,keystone,etc...*

Deben de configurarse las variables con los valores que procedan en cada caso.

## ○ **restart\_services.sh**

```
#!/bin/bash

nova_services='
nova-api
nova-cert
nova-compute
nova-conductor
nova-consoleauth
nova-novncproxy
nova-scheduler
'

neutron_services='
neutron-dhcp-agent
neutron-metadata-agent
neutron-server
neutron-l3-agent
neutron-plugin-openvswitch-agent
'

cinder_services='
cinder-api
cinder-scheduler
cinder-volume
'

heat_services='
heat-api
heat-api-cfn
heat-api-cloudwatch
heat-engine
'

glance_services='
glance-api
glance-registry
'

restart_services() {
    for service in "$@" ; do
        echo "
        echo "Restarting service $service ..."
        service $service restart
    done
    echo "
}
}
```

```

case $1 in
'all')
    echo "Not implemented yet"
;;
'nova')
    restart_services $nova_services
;;
'neutron')
    restart_services $neutron_services
;;
'cinder')
    restart_services $cinder_services
;;
'heat')
    restart_services $heat_services
;;
'glance')
    restart_services $glance_services
;;
*)
    echo "Wrong group name"
;;
esac

```

*Como son muchos procesos por cada componente de Openstack, este pequeño script automatiza el reinicio de todos los servicios. Simplemente debe pasarsele el nombre generico del Componente.*

## ○ Merge-config.sh

```

#!/bin/bash
file1=$1
file2=$2
#-----
trim() {
    echo "$1"
}
# Determinate is the given option present in the INI file
# ini_has_option config-file section option
function ini_has_option() {
    local file=$1
    local section=$2
    local option=$3
    local line
    line=$(sed -ne "/^\[$section\]/,/^\[.*\]/ { /^\$option[ \t]*= / p; }" "$file")
    [ -n "$line" ]
}
# Set an option in an INI file
# inset config-file section option value

function inset() {
    local file=$1
    local section=$2
    local option=$3
    local value=$4
    if ! grep -q "^\[$section\]" "$file"; then
        # Add section at the end
        echo -e "\n[$section]" >>"$file"
    fi
    if ! ini_has_option "$file" "$section" "$option"; then
        # Add it
        sed -i -e "/^\[$section\]/ a\\"
    fi
    $option = $value
    " "$file"
    else
        # Replace it
        sed -i -e "/^\[$section\]/,/^\[.*\]/ s|^\($option[ \t]*=[ \t]*\).*\|1$value|" "$file"
    fi
}
#-----

```

```

if [ ! -f "$file1.orig" ]; then
  cp "$file1" "$file1.orig"
fi

section=""
while IFS='=' read -r option value
do
  [ -z "$option" ] && continue

  [[ "$option" =~ ^# ]] && continue

  if [[ "$option" =~ \[.*\] ]]; then
    section=${option#\[*]}
    section=${section%]*}
    continue
  fi
  option=$(trim $option)
  value=$(trim $value)
  inset "$file1" "$section" "$option" "$value"
done < $file2

```

#####

*Este script sirve para reemplazar la información de configuración de los ficheros que se instalan por defecto con las variables predefinidas inicialmente.*

A continuación van los scripts específicos de los servidores que se han utilizado en el proyecto. Obviamente se pueden utilizar en otros escenarios de configuración simplemente configurando las variables definidas previamente y ejecutándolos en los servidores en los que procediera.

## 2 Servidor Mysql, MongoDB y Colas (MonitorYSwift). 192.168.73.122 eth0

*Servidor que contendrá las bases de Datos de Mysql, MongoDB y de Colas (RabbitMq).*

*Nota1: Inicialmente teniendo en cuenta la pequeña carga del servidor se ha instalado sobre este las herramientas de monitorización (Munin y Nagios).*

*Nota2: este servidor dispone también en pruebas de Swift.*

### ○ **install\_mysql.sh**

```
#!/bin/bash
source ./openrc
# Instalación Mysql
debconf-set-selections <<< "mysql-server mysql-server/root_password password $MYSQL_PASSWORD"
debconf-set-selections <<< "mysql-server mysql-server/root_password_again password $MYSQL_PASSWORD"

apt-get install --yes python-mysqldb mysql-server
sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf
service mysql restart
echo "configuración Base de Datos Mysql"
echo 'create database nova' | mysql -u root -p$MYSQL_PASSWORD
echo 'create database cinder' | mysql -u root -p$MYSQL_PASSWORD
echo 'create database glance' | mysql -u root -p$MYSQL_PASSWORD
echo 'create database heat' | mysql -u root -p$MYSQL_PASSWORD
echo 'create database keystone' | mysql -u root -p$MYSQL_PASSWORD
echo 'create database quantum' | mysql -u root -p$MYSQL_PASSWORD
echo 'show databases' | mysql -u root -p$MYSQL_PASSWORD
echo "grant all privileges on nova.* to '$MYSQL_NOVA_USER'@$HOST_NOVA' identified by '$MYSQL_NOVA_PWD'" | mysql -u root -p$MYSQL_PASSWORD
echo "grant all privileges on cinder.* to '$MYSQL_CINDER_USER'@$HOST_CINDER' identified by '$MYSQL_CINDER_PWD'" | mysql -u root -p$MYSQL_PASSWORD
echo "grant all privileges on glance.* to '$MYSQL_GLANCE_USER'@$HOST_GLANCE' identified by '$MYSQL_GLANCE_PWD'" | mysql -u root -p$MYSQL_PASSWORD
echo "grant all privileges on heat.* to '$MYSQL_HEAT_USER'@$HOST_HEAT' identified by '$MYSQL_HEAT_PWD'" | mysql -u root -p$MYSQL_PASSWORD
echo "grant all privileges on keystone.* to '$MYSQL_KEYSTONE_USER'@$HOST_KEYSTONE' identified by '$MYSQL_KEYSTONE_PWD'" | mysql -u root -p$MYSQL_PASSWORD
echo "grant all privileges on quantum.* to '$MYSQL_QUANTUM_USER'@$HOST_QUANTUM' identified by '$MYSQL_QUANTUM_PWD'" | mysql -u root -p$MYSQL_PASSWORD
echo "flush privileges" | mysql -u root -p$MYSQL_PASSWORD
echo "instalación phpmyadmin"
echo "seleccione opción apache/http"
echo "confirme SI cuando le pregunte ¿Desea configurar la base de datos para phpmyadmin con 'dbconfig-common'?"
echo "Deberá introducir la contraseña del MYSQL (MYSQL_PASSWORD) y la contraseña de seguridad de acceso al PhpMyAdmin"
apt-get install phpmyadmin
echo "El acceso será vía la url http://$HOST_IP/phpmyadmin"
```

### ○ **install\_mongodb.sh**

```
#!/bin/bash
source ./openrc
apt-get install mongodb
sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mongodb.cnf
echo 'db.addUser({user: "ceilometer", pwd: "ceilometer_password", roles: ["readwrite","dbAdmin"]});' > ceilometer.js
mongo ceilometer ceilometer.js
```

### ○ **install\_rabbitmq.sh**

```
#!/bin/bash
apt-get install --yes rabbitmq-server
#Caso de querer cambiar password.
# rabbitmqctl change_password guest new_password
```

```
/usr/lib/rabbitmq/bin/rabbitmq-plugins enable rabbitmq_management
service rabbitmq-server restart
netstat -ntolp | grep beam
```

```
#####
```

### 3 Servidor CINDER (Cinder) Eth0 192.168.73.126

*Este servidor instalará los procesos de gestión de almacenamiento en Bloques (Cinder)*

```
#!/bin/bash
apt-get install cinder-api cinder-scheduler cinder-volume python-cinderclient
apt-get install python-mysqldb
apt-get install mysql-client-core-5.5

source openrc_general
cat << EOF > /etc/cinder/cinder.conf.changes
[DEFAULT]
sql_connection = mysql://$MYSQL_CINDER_USER:$MYSQL_CINDER_PWD@$HOST_IP/cinder
rabbit_host = $HOST_IP
rabbit_password = guest
EOF
#-----
cat << EOF > /etc/cinder/api-paste.ini.changes
[filter:authtoken]
auth_host = $HOST_KEYSTONE
admin_tenant_name = service
admin_user = cinder
admin_password = $SERVICE_PASSWORD
EOF
#-----
./merge-config.sh /etc/cinder/cinder.conf /etc/cinder/cinder.conf.changes
./merge-config.sh /etc/cinder/api-paste.ini /etc/cinder/api-paste.ini.changes
cinder-manage db sync

./restart-os-services.sh cinder
# Asumamos que tenemos un segundo disco duro /dev/sda
sfdisk /dev/sda << EOF
,,8e,,
EOF
pvcreate /dev/sda1
vgcreate cinder-volumes /dev/sda1
```

```
#####
```

### 4 CONTROLLER (Keystone, Glance, Nova, Horizon, Heat, Ceilometer) 192.168.73.128 eth0

Servidor controller principal de OpenStack.

#### ○ **install\_keystone.sh**

```
#!/bin/bash
source openrc
# INSTALL KEYSTONE
apt-get install --yes \
    keystone \
    python-keystone \
    python-keystoneclient

apt-get install python-mysqldb
```

```

apt-get install mysql-client-core-5.5
#-----
# CONFIGURA FICHEROS CONFIGURACION KEYSTONE
cat << EOF > /etc/keystone/keystone.conf.changes
[DEFAULT]
admin_token = $OS_SERVICE_TOKEN
debug = True
verbose = True
[sql]
connection = mysql://$MYSQL_KEYSTONE_USER:$MYSQL_KEYSTONE_PWD@$HOST_IP/keystone
EOF

./merge-config.sh /etc/keystone/keystone.conf /etc/keystone/keystone.conf.changes
service keystone restart
#-----
# INICIALIZA LA BASE DE DATOS CON LA ESTRUCTURA DE KEYSTONE
keystone-manage db_sync
#-----
# CREA LAS ESTRUCTURAS DE DATOS, CUENTAS INICIALES,etc...
# Tenants
keystone tenant-create --name admin
keystone tenant-create --name demo
keystone tenant-create --name service
keystone tenant-list

ADMIN_TENANT=$(keystone tenant-list | grep admin | tr -d " " | awk -F \| ' {print $2}')
DEMO_TENANT=$(keystone tenant-list | grep demo | tr -d " " | awk -F \| ' {print $2}')
SERVICE_TENANT=$(keystone tenant-list | grep service | tr -d " " | awk -F \| ' {print $2}')
#-----
# users
keystone user-create --name admin --pass "$ADMIN_PASSWORD" --email admin@domain.com
keystone user-create --name demo --pass "$DEMO_PASSWORD" --email demo@domain.com --tenant-id $DEMO_TENANT
keystone user-create --name nova --pass "$SERVICE_PASSWORD" --tenant-id $SERVICE_TENANT --email nova@domain.com
keystone user-create --name glance --pass "$SERVICE_PASSWORD" --tenant-id $SERVICE_TENANT --email glance@domain.com
keystone user-create --name quantum --pass "$SERVICE_PASSWORD" --tenant-id $SERVICE_TENANT --email quantum@domain.com
keystone user-create --name cinder --pass "$SERVICE_PASSWORD" --tenant-id $SERVICE_TENANT --email cinder@domain.com
keystone user-create --name ceilometer --pass "$SERVICE_PASSWORD" --tenant-id $SERVICE_TENANT --email
ceilometer@domain.com

ADMIN_USER=$(keystone user-list | grep admin | tr -d " " | awk -F \| ' {print $2}')
DEMO_USER=$(keystone user-list | grep demo | tr -d " " | awk -F \| ' {print $2}')
NOVA_USER=$(keystone user-list | grep nova | tr -d " " | awk -F \| ' {print $2}')
GLANCE_USER=$(keystone user-list | grep glance | tr -d " " | awk -F \| ' {print $2}')
QUANTUM_USER=$(keystone user-list | grep quantum | tr -d " " | awk -F \| ' {print $2}')
CINDER_USER=$(keystone user-list | grep cinder | tr -d " " | awk -F \| ' {print $2}')
CEILOMETER_USER=$(keystone user-list | grep ceilometer | tr -d " " | awk -F \| ' {print $2}')
#-----
# Roles
keystone role-create --name=admin
keystone role-create --name=Member

ADMIN_ROLE=$(keystone role-list | grep admin | tr -d " " | awk -F \| ' {print $2}')
MEMBER_ROLE=$(keystone role-list | grep Member | tr -d " " | awk -F \| ' {print $2}')
#-----
# Add Roles to Users in Tenants
keystone user-role-add --user-id $ADMIN_USER --role-id $ADMIN_ROLE --tenant-id $ADMIN_TENANT
keystone user-role-add --tenant-id $SERVICE_TENANT --user-id $NOVA_USER --role-id $ADMIN_ROLE
keystone user-role-add --tenant-id $SERVICE_TENANT --user-id $GLANCE_USER --role-id $ADMIN_ROLE
keystone user-role-add --tenant-id $SERVICE_TENANT --user-id $QUANTUM_USER --role-id $ADMIN_ROLE
keystone user-role-add --tenant-id $SERVICE_TENANT --user-id $CINDER_USER --role-id $ADMIN_ROLE
keystone user-role-add --tenant-id $SERVICE_TENANT --user-id $CEILOMETER_USER --role-id $ADMIN_ROLE
keystone user-role-add --tenant-id $DEMO_TENANT --user-id $DEMO_USER --role-id $MEMBER_ROLE
#-----
# Create services
keystone service-create --name nova --type compute --description 'OpenStack Compute Service'
keystone service-create --name cinder --type volume --description 'OpenStack Volume Service'
keystone service-create --name glance --type image --description 'OpenStack Image Service'
keystone service-create --name keystone --type identity --description 'OpenStack Identity'
keystone service-create --name ec2 --type ec2 --description 'OpenStack EC2 service'
keystone service-create --name quantum --type network --description 'OpenStack Networking service'
keystone service-create --name heat --type orchestration --description 'HEAT Orchestration API'
keystone service-create --name ceilometer --type metering --description 'Ceilometer Telemetry Service'

COMPUTE_SERVICE=$(keystone service-list | grep nova | tr -d " " | awk -F \| ' {print $2}')
VOLUME_SERVICE=$(keystone service-list | grep cinder | tr -d " " | awk -F \| ' {print $2}')

```

```

IMAGE_SERVICE=$(keystone service-list | grep glance | tr -d " " | awk -F \| '{print $2}')
IDENTITY_SERVICE=$(keystone service-list | grep keystone | tr -d " " | awk -F \| '{print $2}')
EC2_SERVICE=$(keystone service-list | grep ec2 | tr -d " " | awk -F \| '{print $2}')
NETWORK_SERVICE=$(keystone service-list | grep quantum | tr -d " " | awk -F \| '{print $2}')
HEAT_SERVICE=$(keystone service-list | grep heat| tr -d " " | awk -F \| '{print $2}')
CEILOMETER_SERVICE=$(keystone service-list | grep ceilometer| tr -d " " | awk -F \| '{print $2}')
#-----
# Create endpoints
keystone endpoint-create \
  --region $KEYSTONE_REGION \
  --service-id $COMPUTE_SERVICE \
  --publicurl 'http://"$KEYSTONE_HOST":8774/v2/(tenant_id)s' \
  --adminurl 'http://"$KEYSTONE_HOST":8774/v2/(tenant_id)s' \
  --internalurl 'http://"$KEYSTONE_HOST":8774/v2/(tenant_id)s'

keystone endpoint-create \
  --region $KEYSTONE_REGION \
  --service-id $VOLUME_SERVICE \
  --publicurl 'http://"$CINDER_HOST":8776/v1/(tenant_id)s' \
  --adminurl 'http://"$CINDER_HOST":8776/v1/(tenant_id)s' \
  --internalurl 'http://"$CINDER_HOST":8776/v1/(tenant_id)s'

keystone endpoint-create \
  --region $KEYSTONE_REGION \
  --service-id $IMAGE_SERVICE \
  --publicurl 'http://"$KEYSTONE_HOST":9292' \
  --adminurl 'http://"$KEYSTONE_HOST":9292' \
  --internalurl 'http://"$KEYSTONE_HOST":9292'

keystone endpoint-create \
  --region $KEYSTONE_REGION \
  --service-id $IDENTITY_SERVICE \
  --publicurl 'http://"$KEYSTONE_HOST":5000/v2.0' \
  --adminurl 'http://"$KEYSTONE_HOST":35357/v2.0' \
  --internalurl 'http://"$KEYSTONE_HOST":5000/v2.0'

keystone endpoint-create \
  --region $KEYSTONE_REGION \
  --service-id $EC2_SERVICE \
  --publicurl 'http://"$KEYSTONE_HOST":8773/services/Cloud' \
  --adminurl 'http://"$KEYSTONE_HOST":8773/services/Admin' \
  --internalurl 'http://"$KEYSTONE_HOST":8773/services/Cloud'

keystone endpoint-create \
  --region $KEYSTONE_REGION \
  --service-id $NETWORK_SERVICE \
  --publicurl 'http://"$NEUTRON_HOST":9696' \
  --adminurl 'http://"$NEUTRON_HOST":9696' \
  --internalurl 'http://"$NEUTRON_HOST":9696'

keystone endpoint-create \
  --region $KEYSTONE_REGION \
  --service-id $HEAT_SERVICE \
  --publicurl 'http://"$KEYSTONE_HOST":8004/v1/(tenant_id)s' \
  --adminurl 'http://"$KEYSTONE_HOST":8004/v1/(tenant_id)s' \
  --internalurl 'http://"$KEYSTONE_HOST":8004/v1/(tenant_id)s'

keystone endpoint-create \
  --region $KEYSTONE_REGION \
  --service-id $CEILOMETER_SERVICE \
  --publicurl 'http://"$KEYSTONE_HOST":8777' \
  --adminurl 'http://"$KEYSTONE_HOST":8777' \
  --internalurl 'http://"$KEYSTONE_HOST":8777'

```

## ○ **install\_glance.sh**

```

#!/bin/bash
source ./openrc
apt-get install --yes sheepdog glance
#-----
cat << EOF > /etc/glance/glance-api.conf.changes
[DEFAULT]
debug = True
verbose = True

```



```

rabbit_host = $HOST_IP
sql_connection = mysql://$MYSQL_GLANCE_USER:$MYSQL_GLANCE_PWD@$HOST_IP/glance
[keystone_authtoken]
auth_host = $HOST_KEYSTONE
admin_tenant_name = service
admin_user = glance
admin_password = $SERVICE_PASSWORD
[paste_deploy]
flavor = keystone
EOF
#-----
./merge-config.sh /etc/glance/glance-api.conf /etc/glance/glance-api.conf.changes
./merge-config.sh /etc/glance/glance-registry.conf /etc/glance/glance-api.conf.changes
./restart-os-services.sh glance
glance-manage db_sync
#-----
# Inicializa glance - Carga una primera imagen inicial
glance index
wget http://download.cirros-cloud.net/0.3.1/cirros-0.3.1-x86_64-disk.img
glance image-create --name "Cirros 0.3.1" \
  --disk-format qcow2 \
  --container-format bare \
  --is-public true < cirros-0.3.1-x86_64-disk.img
glance index

```

## ○ **install\_nova.sh**

```

#!/bin/bash
apt-get install --yes \
  nova-api \
  nova-cert \
  nova-common \
  nova-conductor \
  nova-scheduler \
  python-nova \
  python-novadient \
  nova-consoleauth \
  novnc \
  nova-novncproxy
#   nova-compute-kvm
pip install --upgrade websockify
source ./openrc
cat << EOF > /etc/nova/api-paste.ini.changes
[filter:authtoken]
admin_tenant_name = service
admin_user = nova
admin_password = $SERVICE_PASSWORD
EOF
cat << EOF > /etc/nova/nova.conf.changes

[DEFAULT]
sql_connection = mysql://$MYSQL_NOVA_USER:$MYSQL_NOVA_PWD@$HOST_IP/nova
my_ip = $HOST_NOVA
rabbit_password = guest
rabbit_host=$HOST_IP
auth_strategy = keystone

# Networking
network_api_class = nova.network.neutronv2.api.API
neutron_url = http://$HOST_QUANTUM:9696
neutron_auth_strategy = keystone
neutron_admin_tenant_name = service
neutron_admin_username = quantum
neutron_admin_password = $SERVICE_PASSWORD
neutron_admin_auth_url = http://$HOST_KEYSTONE:35357/v2.0
libvirt_vif_driver = nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
linuxnet_interface_driver = nova.network.linux_net.LinuxOVSIfaceDriver

# Security Groups
firewall_driver = nova.virt.firewall.NoopFirewallDriver
security_group_api = neutron

```

```

# Metadata
neutron_metadata_proxy_shared_secret = $SERVICE_PASSWORD
service_neutron_metadata_proxy = True
metadata_listen = $HOST_NOVA
metadata_listen_port = 8775
metadata_host = $HOST_NOVA

# Cinder
volume_api_class = nova.volume.cinder.API

# Glance
glance_api_servers = $HOST_GLANCE:9292
image_service = nova.image.glance.GlanceImageService

# novnc
novnc_enabled = True
novncproxy_port = 6080
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = $HOST_NOVA
novncproxy_base_url = http://$HOST_NOVA:6080/vnc_auto.html
EOF
./merge-config.sh /etc/nova/api-paste.ini /etc/nova/api-paste.ini.changes
./merge-config.sh /etc/nova/nova.conf /etc/nova/nova.conf.changes
nova-manage db sync

```

## ○ **install\_horizon.sh**

```

#!/bin/bash
apt-get install --yes openstack-dashboard memcached python-memcache
apt-get remove --yes --purge openstack-dashboard-ubuntu-theme
service apache2 restart

```

## ○ **install\_heat.sh**

```

#!/bin/bash
apt-get install heat-api heat-api-cfn heat-api-cloudwatch heat-common heat-engine python-heat python-heatclient
source openrc
cat << EOF > /etc/heat/api-paste.ini.changes
[filter:authtoken]
auth_host = $HOST_KEYSTONE
auth_port = 35357
auth_protocol = http
auth_uri = http://$HOST_KEYSTONE:5000/v2.0
admin_tenant_name = admin
admin_user = admin
admin_password = $ADMIN_PASSWORD
EOF
#-----
cat << EOF > /etc/heat/heat.conf.changes
[DEFAULT]
debug = True
verbose = True
log_dir = /var/log/heat
sql_connection = mysql://$MYSQL_HEAT_USER:$MYSQL_HEAT_PWD@$HOST_IP/heat
rabbit_host = $HOST_IP
EOF
#-----
./merge-config.sh /etc/heat/api-paste.ini /etc/heat/api-paste.ini.changes
./merge-config.sh /etc/heat/heat.conf /etc/heat/heat.conf.changes
#-----
./restart-os-services.sh heat
heat-manage db_sync

```

## ○ **install\_ceilometer.sh**

```

#!/bin/bash
apt-get install ceilometer-api ceilometer-collector ceilometer-agent-central python-ceilometerclient
cat << EOF > /etc/heat/ceilometer.conf.changes
[database]

```

```

debug = True
verbose = True
connection = mongodb://ceilometer:ceilometer_password@$HOST_IP:27017/ceilometer
rabbit_host = $HOST_IP
[publisher_rpc]
metering_secret = metering_secret
EOF
./merge-config.sh /etc/ceilometer/ceilometer.conf /etc/ceilometer/ceilometer.conf.changes
service ceilometer-agent-central restart
service ceilometer-api restart
service ceilometer-collector restart

```

```
#####
```

## 5 NEUTRON

### ○ install\_neutron.sh

```

#!/bin/bash
apt-get install neutron-server neutron-plugin-openvswitch-agent neutron-dhcp-agent neutron-l3-agent
service openvswitch-switch start
ovs-vsctl add-br br-ex
ovs-vsctl add-br br-int
apt-get install python-mysqldb
apt-get install mysql-client-core-5.5
source ./openrc
cat << EOF >> /etc/sysctl.conf
net.ipv4.ip_forward = 1
net.ipv4.conf.all.forwarding = 1
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.default.rp_filter = 0
EOF
service networking restart
#-----
cat << EOF > /etc/neutron/neutron.conf.changes
[DEFAULT]
debug = True
verbose = True
bind_host = 0.0.0.0
bind_port = 9696
api_paste_config = api-paste.ini
rabbit_host = $HOST_IP
rabbit_userid = guest
rabbit_password = guest
[keystone_authtoken]
admin_tenant_name = service
auth_host = $HOST_KEYSTONE
admin_user = quantum
admin_password = $SERVICE_PASSWORD
[database]
sql_connection = mysql://$MYSQL_QUANTUM_USER:$MYSQL_QUANTUM_PWD@$HOST_IP/quantum

[quotas]
quota_driver = neutron.db.quota_db.DbQuotaDriver
EOF
#-----
cat << EOF > /etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini.changes
[database]
sql_connection = mysql://$MYSQL_QUANTUM_USER:$MYSQL_QUANTUM_PWD@$HOST_IP/quantum
[ovs]
network_vlan_ranges = physnet1
bridge_mappings = physnet1:br-ex
#tenant_network_type = gre
#tunnel_id_ranges = 1:1000
#enable_tunneling = True
#local_ip = 127.0.0.1
[securitygroup]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
EOF
#-----
cat << EOF > /etc/neutron/dhcp_agent.ini.changes

```

```

[DEFAULT]
debug = True
#ovs_use_veth = True
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
use_namespaces = True
enable_isolated_metadata = True
enable_metadata_network = True
#dnsmasq_config_file = /etc/neutron/dnsmasq-neutron.conf
EOF
#-----
cat << EOF > /etc/neutron/metadata_agent.ini.changes
[DEFAULT]
auth_url = http://$HOST_KEYSTONE:35357/v2.0
auth_region = RegionOne
admin_tenant_name = service
admin_user = quantum
admin_password = $SERVICE_PASSWORD
nova_metadata_ip = $HOST_NOVA
nova_metadata_port = 8775
metadata_proxy_shared_secret = $SERVICE_PASSWORD
EOF
#-----
cat << EOF > /etc/neutron/dnsmasq-neutron.conf
dhcp-option-force = 26,1400
EOF

#-----
cat << EOF > /etc/neutron/l3_agent.ini.changes
[DEFAULT]
debug = True
#ovs_use_veth = True
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
use_namespaces = True
external_network_bridge = br-ex
metadata_port = 9697
enable_metadata_proxy = True
EOF
#-----
./merge-config.sh /etc/neutron/neutron.conf /etc/neutron/neutron.conf.changes
./merge-config.sh /etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini
/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini.changes
./merge-config.sh /etc/neutron/dhcp_agent.ini /etc/neutron/dhcp_agent.ini.changes
./merge-config.sh /etc/neutron/l3_agent.ini /etc/neutron/l3_agent.ini.changes
./merge-config.sh /etc/neutron/metadata_agent.ini /etc/neutron/metadata_agent.ini.changes

ln -s /etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini /etc/neutron/plugin.ini

./restart-os-services.sh neutron

```

#####

## 6 COMPUTE

```

#!/bin/bash
source ./openrc
apt-get install nova-compute
apt-get install python-mysqldb
apt-get install mysql-client-core-5.5
apt-get install neutron-plugin-openvswitch-agent openvswitch-switch openvswitch-datapath-dkms
#-----
cat << EOF >> /etc/sysctl.conf

net.ipv4.ip_forward = 1
net.ipv4.conf.all.forwarding = 1
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.default.rp_filter = 0
EOF
service networking restart
#-----
kvm-ok

```

```

if [ $? -eq 0 ] ; then
    libvirt_type='kvm'
else
    libvirt_type='qemu'
fi

cat << EOF > /etc/nova/nova-compute.conf.changes
[DEFAULT]
libvirt_type = $libvirt_type
EOF
#-----
cat << EOF > /etc/nova/api-paste.ini.changes
[filter:authtoken]
auth_host = $HOST_KEYSTONE
admin_tenant_name = service
admin_user = nova
admin_password = $SERVICE_PASSWORD
EOF
#-----
cat << EOF > /etc/nova/nova.conf.changes
[DEFAULT]
sql_connection = mysql://$MYSQL_NOVA_USER:$MYSQL_NOVA_PWD@$HOST_IP/nova
my_ip = $HOST_NOVA
rabbit_password = guest
rabbit_host=$HOST_IP
auth_strategy = keystone
# Networking
network_api_class = nova.network.neutronv2.api.API
neutron_url = http://$HOST_QUANTUM:9696
neutron_auth_strategy = keystone
neutron_admin_tenant_name = service
neutron_admin_username = quantum
neutron_admin_password = $SERVICE_PASSWORD
neutron_admin_auth_url = http://$HOST_KEYSTONE:35357/v2.0
libvirt_vif_driver = nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
linuxnet_interface_driver = nova.network.linux_net.LinuxOVSInterfaceDriver

# Security Groups
firewall_driver = nova.virt.firewall.NoopFirewallDriver
security_group_api = neutron
# Metadata
neutron_metadata_proxy_shared_secret = $SERVICE_PASSWORD
service_neutron_metadata_proxy = True
metadata_listen = $HOST_NOVA
metadata_listen_port = 8775
metadata_host = $HOST_NOVA

# Cinder
volume_api_class = nova.volume.cinder.API

# Glance
glance_api_servers = $HOST_GLANCE:9292
image_service = nova.image.glance.GlanceImageService

# novnc
novnc_enabled = True
novncproxy_port = 6080
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = $HOST_COMPUTE1
novncproxy_base_url = http://$HOST_COMPUTE1:6080/vnc_auto.html
EOF

./merge-config.sh /etc/nova/api-paste.ini /etc/nova/api-paste.ini.changes
./merge-config.sh /etc/nova/nova.conf /etc/nova/nova.conf.changes
./merge-config.sh /etc/nova/nova-compute.conf /etc/nova/nova-compute.conf.changes

cat << EOF >
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
EOF

./merge-config.sh /etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini
/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini.changes
service nova-compute restart
service openvswitch-switch restart
ovs-vsctl add-br br-int

```

## 7 Configuración proyecto Interfaces de Red

### ○ Server Controller 1

```
auto lo
iface lo inet loopback

# Red gestión Interna
auto eth0
iface eth0 inet static
    address 192.168.73.128
    netmask 255.255.255.0
    network 192.168.73.0
    broadcast 192.168.73.255
    gateway 192.168.73.254
    dns-nameservers 8.8.8.8

# Red APIs
auto eth2
iface eth2 inet static
    address 192.168.89.128
    netmask 255.255.255.0
    network 192.168.89.0
    broadcast 192.168.89.255

# Red Almacenamiento
auto eth3
iface eth3 inet static
    address 192.168.90.128
    netmask 255.255.255.0
    network 192.168.90.0
    broadcast 192.168.90.255
```

### ○ Server Controller 2 (Monitor y Swift) – Mysql,Rabbitmq y Mongodb

```
auto lo
iface lo inet loopback

# Red Gestión
auto eth0
iface eth0 inet static
    address 192.168.73.122
    netmask 255.255.255.0
    network 192.168.73.0
    broadcast 192.168.73.255
    gateway 192.168.73.254
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 8.8.8.8
```

### ○ Server Cinder

```
auto lo
iface lo inet loopback
# Red Gestión
auto eth0
iface eth0 inet static
    address 192.168.73.126
    netmask 255.255.255.0
    network 192.168.73.0
    broadcast 192.168.73.255
    gateway 192.168.73.254
    dns-nameservers 8.8.8.8

# Red API
auto eth2
iface eth2 inet static
    address 192.168.89.126
    netmask 255.255.255.0
    network 192.168.89.0
    broadcast 192.168.89.255
```

```
# Red Storage
auto eth3
iface eth3 inet static
    address 192.168.90.126
    netmask 255.255.255.0
    network 192.168.90.0
    broadcast 192.168.90.255
```

## ○ **Server Neutron**

```
auto lo
iface lo inet loopback
# Red Gestión
auto eth0
iface eth0 inet static
    address 192.168.73.127
    netmask 255.255.255.0
    network 192.168.73.0
    broadcast 192.168.73.255
    gateway 192.168.73.254
    dns-nameservers 8.8.8.8
```

```
# Red Interna de datos con servidores computación.
auto eth1
iface eth1 inet manual
up ifconfig $IFACE 0.0.0.0 up
up ifconfig $IFACE promisc
# Red Externa
auto eth2
iface eth2 inet manual
up ifconfig $IFACE 0.0.0.0 up
up ifconfig $IFACE promisc
```

## ○ **Server Compute1**

```
auto lo
iface lo inet loopback
# Red Gestión
auto eth0
iface eth0 inet static
    address 192.168.73.120
    netmask 255.255.255.0
    network 192.168.73.0
    broadcast 192.168.73.255
    gateway 192.168.73.254
    dns-nameservers 8.8.8.8
```

```
# Red Storage
auto eth3
iface eth3 inet static
    address 192.168.90.120
    netmask 255.255.255.0
    network 192.168.90.0
    broadcast 192.168.90.255
```

```
# Red Conexión Datos Interna Computación con Neutron
auto eth1
iface eth1 inet manual
up ifconfig $IFACE 0.0.0.0 up
up ifconfig $IFACE promisc
```

## ○ Server Compute2

```
auto lo
iface lo inet loopback
# Red Gestión
auto eth0
iface eth0 inet static
    address 192.168.73.121
    netmask 255.255.255.0
    network 192.168.73.0
    broadcast 192.168.73.255
    gateway 192.168.73.254
    dns-nameservers 8.8.8.8

# Red Storage
auto eth3
iface eth3 inet static
    address 192.168.90.121
    netmask 255.255.255.0
    network 192.168.90.0
    broadcast 192.168.90.255
# Red Conexión Datos Interna Computación con Neutron
auto eth1
iface eth1 inet manual
up ifconfig $IFACE 0.0.0.0 up
up ifconfig $IFACE promisc
```