

Anexo 2 : Ficheros de Configuración.

OpenStack configura las variables principales, de cada uno de sus componentes, en ficheros planos "txt". Es habitual cometer fallos en la configuración de los ficheros y encontrar problemas que requieran horas de esfuerzo depurando los logs. A continuación copio los contenidos de los ficheros de configuración que he utilizado en la solución planteada en el proyecto. Servidor, Módulo de Openstack y finalmente fichero.

Nota: He filtrado todas las líneas comentadas de los ficheros.

Ejemplo → `root@controller:/etc/keystone#grep -v "#" keystone.conf | grep -v '^$'`

Nota1: hay variables de host que apunta a los servidores en cuestión que corresponda. Estas variables las configuré en /etc/hosts. Un ejemplo es `rabbit_host=rabbithost` (`rabbithost=192.168.73.122` / para que apunte y se conecte con el servidor de colas rabbit correctamente). En algunas figura la variable y en otras directamente la ip correspondiente.

Nota2: Los ficheros y su path van en negrilla subrayados.

• **SERVIDOR CONTROLLER**

○ **Keystone**

▪ **keystone.conf**

Es primordial que la configuración inicial de las tablas de la base de datos de keystone se haga correctamente ya que todos los servicios autenticarán posteriormente de esta tabla vía keystone.

El campo `admin_token` debe borrarse una vez que el sistema se configure inicialmente. Es un token alternativo de uso para configuración inicial, pero que viene muy bien para pruebas.

```
root@controller:/etc/keystone# grep -v "#" keystone.conf | grep -v '^$'
[DEFAULT]
verbose = True
debug = True
admin_token = TOKEN_PASSWORD
log_file = keystone.log
log_dir = /var/log/keystone
[sql]
connection = mysql://keystone:keystone_password@monitor/keystone
[identity]
driver = keystone.identity.backends.sql.Identity
[credential]
driver = keystone.credential.backends.sql.Credential
[trust]
driver = keystone.trust.backends.sql.Trust
[os_inherit]
[catalog]
driver = keystone.catalog.backends.sql.Catalog
[endpoint_filter]
[token]
driver = keystone.token.backends.sql.Token
[cache]
[policy]
driver = keystone.policy.backends.sql.Policy
[ec2]
driver = keystone.contrib.ec2.backends.kvs.Ec2
[assignment]
[oauth1]
[ssl]
[signing]
[ldap]
[auth]
methods = external,password,token,oauth1
password = keystone.auth.plugins.password.Password
```

```
token = keystone.auth.plugins.token.Token
oauth1 = keystone.auth.plugins.oauth1.OAuth
[paste_deploy]
config_file = keystone-paste.ini
root@controller:/etc/keystone#
```

○ Glance

Utiliza mayormente 2 ficheros de configuración: glance-registry.conf y glance-api.conf

■ glance-api.conf

```
root@controller:/etc/glance# grep -v "#" glance-api.conf | grep -v '^$'
[DEFAULT]
verbose = True
debug = True
default_store = file
bind_host = 0.0.0.0
bind_port = 9292
log_file = /var/log/glance/api.log
backlog = 4096
sql_connection = mysql://glance:glance_password@monitor/glance
sql_idle_timeout = 3600
workers = 1
registry_host = 0.0.0.0
registry_port = 9191
registry_client_protocol = http
notifier_strategy = rabbit
rabbit_host = rabbithost
rabbit_port = 5672
rabbit_use_ssl = false
rabbit_userid = guest
rabbit_password = guest
rabbit_virtual_host = /
rabbit_notification_exchange = glance
rabbit_notification_topic = notifications
rabbit_durable_queues = False
qpid_notification_exchange = glance
qpid_notification_topic = notifications
qpid_host = localhost
qpid_port = 5672
qpid_username =
qpid_password =
qpid_sasl_mechanisms =
qpid_reconnect_timeout = 0
qpid_reconnect_limit = 0
qpid_reconnect_interval_min = 0
qpid_reconnect_interval_max = 0
qpid_reconnect_interval = 0
qpid_heartbeat = 5
qpid_protocol = tcp
qpid_tcp_nodelay = True
filesystem_store_datadir = /var/lib/glance/images/
swift_store_auth_version = 2
swift_store_auth_address = 127.0.0.1:5000/v2.0/
swift_store_user = jdoe:jdoe
swift_store_key = a86850deb2742ec3cb41518e26aa2d89
swift_store_container = glance
swift_store_create_container_on_put = False
swift_store_large_object_size = 5120
swift_store_large_object_chunk_size = 200
swift_enable_snet = False
s3_store_host = 127.0.0.1:8080/v1.0/
s3_store_access_key = <20-char AWS access key>
s3_store_secret_key = <40-char AWS secret key>
s3_store_bucket = <lowercased 20-char aws access key>glance
s3_store_create_bucket_on_put = False
rbd_store_ceph_conf = /etc/ceph/ceph.conf
rbd_store_user = glance
rbd_store_pool = images
```

```
rbd_store_chunk_size = 8
sheepdog_store_address = localhost
sheepdog_store_port = 7000
sheepdog_store_chunk_size = 64
delayed_delete = False
scrub_time = 43200
scrubber_datadir = /var/lib/glance/scrubber
image_cache_dir = /var/lib/glance/image-cache/
[keystone_authtoken]
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = service-password
[paste_deploy]
flavor = keystone
root@controller:/etc/glance#
```

■ glance-registry.conf

```
root@controller:/etc/glance# grep -v "#" glance-registry.conf | grep -v '^$'
[DEFAULT]
verbose = True
debug = True
bind_host = 0.0.0.0
bind_port = 9191
log_file = /var/log/glance/registry.log
backlog = 4096
sql_connection = mysql://glance:glance_password@monitor/glance
sql_idle_timeout = 3600
api_limit_max = 1000
limit_param_default = 25
[keystone_authtoken]
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = service-password
[paste_deploy]
flavor = keystone
root@controller:/etc/glance#
```

○ **Nova**

■ **nova.conf**

```
root@controller:/etc/nova# grep -v "#" nova.conf | grep -v '^$'
[DEFAULT]
novncproxy_base_url = http://192.168.73.128:6080/vnc_auto.html
vncserver_proxyclient_address = controller
vncserver_listen = 0.0.0.0
novncproxy_port = 6080
novnc_enabled = True
image_service = nova.image.glance.GlanceImageService
glance_api_servers = controller:9292
volume_api_class = nova.volume.cinder.API
metadata_host = 192.168.73.127
metadata_listen_port = 8775
metadata_listen = 0.0.0.0
service_neutron_metadata_proxy = True
neutron_metadata_proxy_shared_secret = service-password
security_group_api = neutron
firewall_driver = nova.virt.firewall.NoopFirewallDriver
linuxnet_interface_driver = nova.network.linux_net.LinuxOVSInterfaceDriver
libvirt_vif_driver = nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
neutron_admin_auth_url = http://controller:35357/v2.0
neutron_admin_password = service-password
neutron_admin_username = quantum
```

```

neutron_admin_tenant_name = service
neutron_auth_strategy = keystone
neutron_url = http://neutron:9696
network_api_class = nova.network.neutronv2.api.API
auth_strategy = keystone
rabbit_password = guest
rabbit_host = rabbithost
my_ip = 192.168.73.128
sql_connection = mysql://nova:nova_password@monitor/nova
dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
force_dhcp_release=True
iscsi_helper=tgtadm
iscsi_ip_address=192.168.90.128
libvirt_use_virtio_for_bridges=True
connection_type=libvirt
compute_driver=libvirt.LibvirtDriver
root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf
verbose=True
debug=True
ec2_private_dns_show_ip=True
api_paste_config=/etc/nova/api-paste.ini
volumes_path=/var/lib/nova/volumes
enabled_apis=ec2,osapi_compute,metadata
instance_usage_audit = True
root@controller:/etc/nova#

```

■ api-paste.ini

```

root@controller:/etc/nova# grep -v "#" api-paste.ini | grep -v '^$'
[composite:metadata]
/: meta
[pipeline:meta]
pipeline = ec2faultwrap logrequest metaapp
[app:metaapp]
paste.app_factory = nova.api.metadata.handler:MetadataRequestHandler.factory
[composite:ec2]
/services/Cloud: ec2cloud
[composite:ec2cloud]
use = call:nova.api.auth:pipeline_factory
noauth = ec2faultwrap logrequest ec2noauth cloudrequest validator ec2executor
keystone = ec2faultwrap logrequest ec2keystoneauth cloudrequest validator ec2executor
[filter:ec2faultwrap]
paste.filter_factory = nova.api.ec2:FaultWrapper.factory
[filter:logrequest]
paste.filter_factory = nova.api.ec2:RequestLogging.factory
[filter:ec2logout]
paste.filter_factory = nova.api.ec2:Lockout.factory
[filter:ec2keystoneauth]
paste.filter_factory = nova.api.ec2:EC2KeystoneAuth.factory
[filter:ec2noauth]
paste.filter_factory = nova.api.ec2:NoAuth.factory
[filter:cloudrequest]
controller = nova.api.ec2.cloud.CloudController
paste.filter_factory = nova.api.ec2:Requestify.factory
[filter:authorizer]
paste.filter_factory = nova.api.ec2:Authorizer.factory
[filter:validator]
paste.filter_factory = nova.api.ec2:Validator.factory
[app:ec2executor]
paste.app_factory = nova.api.ec2:Executor.factory
[composite:osapi_compute]
use = call:nova.api.openstack.urlmap:urlmap_factory
/: oscomputeversions
/v1.1: openstack_compute_api_v2
/v2: openstack_compute_api_v2
/v3: openstack_compute_api_v3
[composite:openstack_compute_api_v2]
use = call:nova.api.auth:pipeline_factory
noauth = faultwrap sizelimit noauth ratelimit osapi_compute_app_v2
keystone = faultwrap sizelimit authtoken keystonecontext ratelimit osapi_compute_app_v2

```

```

keystone_nolimit = faultwrap sizelimit authtoken keystonecontext osapi_compute_app_v2
[composite:openstack_compute_api_v3]
use = call:nova.api.auth:pipeline_factory
noauth = faultwrap sizelimit noauth_v3 ratelimit_v3 osapi_compute_app_v3
keystone = faultwrap sizelimit authtoken keystonecontext ratelimit_v3 osapi_compute_app_v3
keystone_nolimit = faultwrap sizelimit authtoken keystonecontext osapi_compute_app_v3
[filter:faultwrap]
paste.filter_factory = nova.api.openstack:FaultWrapper.factory
[filter:noauth]
paste.filter_factory = nova.api.openstack.auth:NoAuthMiddleware.factory
[filter:noauth_v3]
paste.filter_factory = nova.api.openstack.auth:NoAuthMiddlewareV3.factory
[filter:ratelimit]
paste.filter_factory = nova.api.openstack.compute.limits:RateLimitingMiddleware.factory
[filter:ratelimit_v3]
paste.filter_factory = nova.api.openstack.compute.plugins.v3.limits:RateLimitingMiddleware.factory
[filter:sizelimit]
paste.filter_factory = nova.api.sizelimit:RequestBodySizeLimiter.factory
[app:osapi_compute_app_v2]
paste.app_factory = nova.api.openstack.compute:APIRouter.factory
[app:osapi_compute_app_v3]
paste.app_factory = nova.api.openstack.compute:APIRouterV3.factory
[pipeline:oscomputeversions]
pipeline = faultwrap oscomputeversionapp
[app:oscomputeversionapp]
paste.app_factory = nova.api.openstack.compute.versions:Versions.factory
[filter:keystonecontext]
paste.filter_factory = nova.api.auth:NovaKeystoneContext.factory
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = service-password
auth_version = v2.0
root@controller:/etc/nova#

```

○ Heat

■ heat.conf

```

root@controller:/etc/heat# grep -v "#" heat.conf | grep -v '^$'
[DEFAULT]
log_dir = /var/log/heat
verbose = True
debug = True
sql_connection=mysql://heat:heat_password@monitor/heat
rabbit_host=rabbithost
[ssl]
[database]
[paste_deploy]
[rpc_notifier2]
[keystone_authtoken]
auth_host = controller
auth_port = 35357
auth_protocol = http
auth_uri = http://controller:5000/v2.0
admin_tenant_name = service
admin_user = heat
admin_password = heat_password
[ec2authtoken]
auth_uri = http://controller:5000/v2.0
keystone_ec2_uri = http://controller:5000/v2.0/ec2tokens
[heat_api_cloudwatch]
[heat_api]
[heat_api_cfn]
[auth_password]
[matchmaker_ring]
[matchmaker_redis]
root@controller:/etc/heat#

```

▪ api-paste.ini

```
root@controller:/etc/heat# grep -v "#" api-paste.ini | grep -v '^$'
[pipeline:heat-api]
pipeline = faultwrap versionnegotiation authtoken context apiv1app
[pipeline:heat-api-standalone]
pipeline = faultwrap versionnegotiation authpassword context apiv1app
[pipeline:heat-api-custombackend]
pipeline = faultwrap versionnegotiation context custombackendauth apiv1app
[pipeline:heat-api-cfn]
pipeline = cfnversionnegotiation ec2authtoken authtoken context apicfnv1app
[pipeline:heat-api-cfn-standalone]
pipeline = cfnversionnegotiation ec2authtoken context apicfnv1app
[pipeline:heat-api-cloudwatch]
pipeline = versionnegotiation ec2authtoken authtoken context apicwapp
[pipeline:heat-api-cloudwatch-standalone]
pipeline = versionnegotiation ec2authtoken context apicwapp
[app:apiv1app]
paste.app_factory = heat.common.wsgi:app_factory
heat.app_factory = heat.api.openstack.v1:API
[app:apicfnv1app]
paste.app_factory = heat.common.wsgi:app_factory
heat.app_factory = heat.api.cfn.v1:API
[app:apicwapp]
paste.app_factory = heat.common.wsgi:app_factory
heat.app_factory = heat.api.cloudwatch:API
[filter:versionnegotiation]
paste.filter_factory = heat.common.wsgi:filter_factory
heat.filter_factory = heat.api.openstack:version_negotiation_filter
[filter:faultwrap]
paste.filter_factory = heat.common.wsgi:filter_factory
heat.filter_factory = heat.api.openstack:faultwrap_filter
[filter:cfnversionnegotiation]
paste.filter_factory = heat.common.wsgi:filter_factory
heat.filter_factory = heat.api.cfn:version_negotiation_filter
[filter:cwversionnegotiation]
paste.filter_factory = heat.common.wsgi:filter_factory
heat.filter_factory = heat.api.cloudwatch:version_negotiation_filter
[filter:context]
paste.filter_factory = heat.common.context:ContextMiddleware_filter_factory
[filter:ec2authtoken]
paste.filter_factory = heat.api.aws.ec2token:EC2Token_filter_factory
[filter:authtoken]
admin_password = admin-password
admin_user = admin
admin_tenant_name = admin
auth_uri = http://controller:5000/v2.0
auth_protocol = http
auth_port = 35357
auth_host = controller
paste.filter_factory = heat.common.auth_token:filter_factory
[filter:authpassword]
paste.filter_factory = heat.common.auth_password:filter_factory
[filter:custombackendauth]
paste.filter_factory = heat.common.custom_backend_auth:filter_factory
root@controller:/etc/heat#
```

○ Ceilometer

■ ceilometer.conf

```
root@controller:/etc/ceilometer# grep -v "#" ceilometer.conf | grep -v '^$'
[DEFAULT]
http_control_exchanges=nova
http_control_exchanges=glance
http_control_exchanges=neutron
http_control_exchanges=cinder
auth_strategy=keystone
enable_v1_api=true
nova_control_exchange=nova
hypervisor_inspector=libvirt
libvirt_type=kvm
glance_control_exchange=glance
neutron_control_exchange=neutron
debug=True
verbose=True
rpc_backend=ceilometer.openstack.common.rpc.impl_kombu
rabbit_host = rabbithost
rabbit_password = guest
cinder_control_exchange=cinder
[publisher_rpc]
metering_secret = e4f83cf2ea21b4211776
[ssl]
[database]
connection = mongodb://ceilometer:ceilometer_password@monitor:27017/ceilometer
[alarm]
[rpc_notifier2]
[api]
[service_credentials]
[dispatcher_file]
[keystone_auth_token]
auth_host = controller
auth_port = 35357
auth_protocol = http
auth_uri = http://controller:35357/v2.0
admin_tenant_name = service
admin_user = ceilometer
admin_password = ceilometer_password
[collector]
dispatcher=database
dispatcher=file
[matchmaker_ring]
[matchmaker_redis]
root@controller:/etc/ceilometer#
```

• SERVIDOR CONTROLLER2 (monitorYswift)

▪ mongodb.conf

```
root@swift:/etc# grep -v "#" mongodb.conf | grep -v '^$'  
logpath=/var/log/mongodb/mongodb.log  
logappend=true  
root@swift:/etc#
```

```
dbpath=/var/lib/mongodb
```

▪ my.cnf

```
root@swift:/etc/mysql# grep -v "#" my.cnf | grep -v '^$'  
[client]  
port = 3306  
socket = /var/run/mysqld/mysqld.sock  
[mysqld_safe]  
socket = /var/run/mysqld/mysqld.sock  
nice = 0  
[mysqld]  
user = mysql  
pid-file = /var/run/mysqld/mysqld.pid  
socket = /var/run/mysqld/mysqld.sock  
port = 3306  
basedir = /usr  
datadir = /var/lib/mysql  
tmpdir = /tmp  
lc-messages-dir = /usr/share/mysql  
skip-external-locking  
bind-address = 192.168.73.122  
key_buffer = 16M  
max_allowed_packet = 16M  
thread_stack = 192K  
thread_cache_size = 8  
myisam-recover = BACKUP  
query_cache_limit = 128M  
query_cache_size = 256M  
tmp_table_size = 512M  
max_heap_table_size = 512M  
log_error = /var/log/mysql/error.log  
expire_logs_days = 10  
max_binlog_size = 100M  
[mysqldump]  
quick  
quote-names  
max_allowed_packet = 16M  
[mysql]  
[isamchk]  
key_buffer = 16M  
!includedir /etc/mysql/conf.d/  
root@swift:/etc/mysql#
```


• SERVIDOR CINDER

▪ cinder.conf

```
root@cinder: /etc/cinder# grep -v "#" cinder.conf | grep -v '^$'
[DEFAULT]
rabbit_host = rabbithost
rabbit_password = guest
sql_connection = mysql://cinder:cinder_password@monitor/cinder
rootwrap_config = /etc/cinder/rootwrap.conf
api_paste_config = /etc/cinder/api-paste.ini
iscsi_helper = tgtadm
volume_name_template = volume-%s
volume_group = cinder-volumes
verbose = True
auth_strategy = keystone
state_path = /var/lib/cinder
lock_path = /var/lock/cinder
volumes_dir = /var/lib/cinder/volumes
control_exchange = cinder
notification_driver = cinder.openstack.common.notifier.rpc_notifier
root@cinder: /etc/cinder#
```

▪ api-paste.ini

```
root@cinder: /etc/cinder# grep -v "#" api-paste.ini | grep -v '^$'
[composite:osapi_volume]
use = call:cinder.api:root_app_factory
/: apiversions
/v1: openstack_volume_api_v1
/v2: openstack_volume_api_v2
[composite:openstack_volume_api_v1]
use = call:cinder.api.middleware.auth:pipeline_factory
noauth = faultwrap sizelimit noauth apiv1
keystone = faultwrap sizelimit authtoken keystonecontext apiv1
keystone_nolimit = faultwrap sizelimit authtoken keystonecontext apiv1
[composite:openstack_volume_api_v2]
use = call:cinder.api.middleware.auth:pipeline_factory
noauth = faultwrap sizelimit noauth apiv2
keystone = faultwrap sizelimit authtoken keystonecontext apiv2
keystone_nolimit = faultwrap sizelimit authtoken keystonecontext apiv2
[filter:faultwrap]
paste.filter_factory = cinder.api.middleware.fault:FaultWrapper.factory
[filter:noauth]
paste.filter_factory = cinder.api.middleware.auth:NoAuthMiddleware.factory
[filter:sizelimit]
paste.filter_factory = cinder.api.middleware.sizelimit:RequestBodySizeLimiter.factory
[app:apiv1]
paste.app_factory = cinder.api.v1.router:APIRouter.factory
[app:apiv2]
paste.app_factory = cinder.api.v2.router:APIRouter.factory
[pipeline:apiversions]
pipeline = faultwrap osvolumeverversionapp
[app:osvolumeverversionapp]
paste.app_factory = cinder.api.versions:Versions.factory
[filter:keystonecontext]
paste.filter_factory = cinder.api.middleware.auth:CinderKeystoneContext.factory
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = cinder
admin_password = service-password
root@cinder: /etc/cinder#
```

• SERVIDOR NEUTRON

▪ neutron.conf

```
root@neutron:/etc/neutron# grep -v "#" neutron.conf | grep -v '^$'
[DEFAULT]
rabbit_password = guest
rabbit_userid = guest
rabbit_host = rabbithost
api_paste_config = api-paste.ini
bind_port = 9696
bind_host = 0.0.0.0
verbose = True
debug = True
state_path = /var/lib/neutron
lock_path = $state_path/lock
core_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2
auth_strategy = keystone
fake_rabbit = False
notification_driver = neutron.openstack.common.notifier.rpc_notifier
[quotas]
quota_driver = neutron.db.quota_db.DbQuotaDriver
[agent]
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf
[keystone_authtoken]
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = quantum
admin_password = service-password
signing_dir = $state_path/keystone-signing
[database]
connection = mysql://quantum:quantum_password@monitor/quantum
[service_providers]
service_provider=LOADBALANCER:Haproxy:neutron.services.loadbalancer.drivers.haproxy.plugin_driver.HaproxyOnHostPluginDriver:default
root@neutron:/etc/neutron#
```

▪ dnsmasq-neutron.conf

```
root@neutron:/etc/neutron# grep -v "#" dnsmasq-neutron.conf | grep -v '^$'
dhcp-option-force = 26,1400
root@neutron:/etc/neutron#
```

▪ api-paste.ini

```
root@neutron:/etc/neutron# grep -v "#" api-paste.ini | grep -v '^$'
[composite:neutron]
/: neutronversions
/v2.0: neutronapi_v2_0
[composite:neutronapi_v2_0]
use = call:neutron.auth:pipeline_factory
noauth = extensions neutronapiapp_v2_0
keystone = authtoken keystonecontext extensions neutronapiapp_v2_0
[filter:keystonecontext]
paste.filter_factory = neutron.auth:NeutronKeystoneContext.factory
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host=controller
auth_uri=http://controller:5000
admin_user=quantum
admin_tenant=service
admin_password=service-password
[filter:extensions]
paste.filter_factory = neutron.api.extensions:plugin_aware_extension_middleware_factory
[app:neutronversions]
```

```
paste.app_factory = neutron.api.versions:Versions.factory
[app:neutronapiapp_v2_0]
paste.app_factory = neutron.api.v2.router:APIRouter.factory
root@neutron:/etc/neutron#
```

▪ dhcp_agent.ini

```
root@neutron:/etc/neutron# grep -v "#" dhcp_agent.ini | grep -v '^$'
[DEFAULT]
enable_metadata_network = True
enable_isolated_metadata = True
use_namespaces = True
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
debug = True
root@neutron:/etc/neutron#
```

▪ l3_agent.ini

```
root@neutron:/etc/neutron# grep -v "#" l3_agent.ini | grep -v '^$'
[DEFAULT]
auth_url = http://controller:35357/v2.0
admin_tenant_name = service
admin_user = quantum
admin_password = service-password
metadata_ip = controller
enable_metadata_proxy = True
metadata_port = 8775
external_network_bridge = br-ex
use_namespaces = True
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
debug = True
root@neutron:/etc/neutron#
```

▪ metadata_agent.ini

```
root@neutron:/etc/neutron# grep -v "#" metadata_agent.ini | grep -v '^$'
[DEFAULT]
metadata_proxy_shared_secret = service-password
nova_metadata_port = 8775
nova_metadata_ip = controller
debug = True
auth_url = http://controller:35357/v2.0
auth_region = RegionOne
admin_tenant_name = service
admin_user = quantum
admin_password = service-password
root@neutron:/etc/neutron#
```

▪ ovs_neutron_plugin.ini

```
root@neutron:/etc/neutron/plugins/openvswitch# grep -v "#" ovs_neutron_plugin.ini | grep -v '^$'
[agent]
[securitygroup]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
[OVS]
tenant_network_type=vlan
network_vlan_ranges = physnet1:200:4094
bridge_mappings = physnet1:br-eth1,physnet2:br-ex
[database]
sql_connection = mysql://quantum:quantum_password@monitor/quantum
root@neutron:/etc/neutron/plugins/openvswitch#
```

• SERVIDOR COMPUTE1

El servidor compute2 y los sucesivos que se instalarán deberían llevar la misma configuración salvo los datos de ip,etc... de los servidores correspondientes.

▪ nova.conf

```
root@compute1:/etc/nova# grep -v "#" nova.conf | grep -v '^$'
[DEFAULT]
novncproxy_base_url = http://192.168.73.128:6080/vnc_auto.html
vncserver_proxyclient_address = compute1
vncserver_listen = 0.0.0.0
novncproxy_port = 6080
novnc_enabled = True
image_service = nova.image.glance.GlanceImageService
glance_api_servers = controller:9292
volume_api_class = nova.volume.cinder.API
metadata_host = 192.168.73.127
metadata_listen_port = 8775
metadata_listen = neutron
service_neutron_metadata_proxy = True
neutron_metadata_proxy_shared_secret = service-password
security_group_api = neutron
firewall_driver = nova.virt.firewall.NoopFirewallDriver
linuxnet_interface_driver = nova.network.linux_net.LinuxOVSIInterfaceDriver
libvirt_vif_driver = nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
neutron_admin_auth_url = http://controller:35357/v2.0
neutron_admin_password = service-password
neutron_admin_username = quantum
neutron_admin_tenant_name = service
neutron_auth_strategy = keystone
neutron_url = http://neutron:9696
network_api_class = nova.network.neutronv2.api.API
auth_strategy = keystone
rabbit_password = guest
rabbit_host = rabbithost
my_ip = compute1
sql_connection = mysql://nova:nova_password@monitor/nova
dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
force_dhcp_release=True
iscsi_helper=tgtadm
iscsi_ip_address = 192.168.90.120
libvirt_use_virtio_for_bridges=True
connection_type=libvirt
compute_driver=libvirt.LibvirtDriver
root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf
verbose=True
debug=True
ec2_private_dns_show_ip=True
api_paste_config=/etc/nova/api-paste.ini
volumes_path=/var/lib/nova/volumes
enabled_apis=ec2,osapi_compute,metadata
instance_usage_audit=True
instance_usage_audit_period=hour
notify_on_state_change=vm_and_task_state
notification_driver=nova.openstack.common.notifier.rpc_notifier
notification_driver=ceilometer.compute.nova_notifier
root@compute1:/etc/nova#
```