Criando Nuvens Privadas usando OpenStack com suporte ao IPv6

Alex Sandro de Lacerda Vitola
alex.vitola@gmail.com
http://www.vitola.net.br

Algumas observações sobre o ambiente criado

Este projeto foi apresentado originalmente num trabalho interno de conhecimento do OpenStack e uma futura utilização do mesmo para orquestração de nuvens privadas da empresa onde eu Trabalhava.
Fiz algumas adaptações para torná-lo compatível com o IPv6. Já que no ambiente original ele era somente IPv4.
Para este projeto foi utilizado um tunel da sixxs com um prefixo /64, posteriormente foi liberada uma subrede /48
Tentei criar um ambiente simples nas configurações de Rede, por isso optei por utilizar o nova-network ao invés do Quantum.
Isso trouxe algumas limitações já que o nova-network está praticamente sem atualizações.
O floating ip's não é 100% suportado, até por não fazer muito sentido, já que ná pratica ele é quase um NAT.
Apenas de usar um prefixo /48 na minha subrede a nova-network sempre especifica um prefixo /64.
Por padrão o nova-network usa o radvd, como já tenha o radvd instalado no meu gateway, a mesma máquina que faz o tunel com a sixxs isso causou alguns conflitos também.

Neste projeto descrevo passo a passo da instalação e configuração de todos os serviços utilizados.

Requisitos do Ambiente

Máquina virtual criada usando o VMWare Workstation com o Ubuntu 12.04.2

Duas placas de rede
eth0 192.168.0.51 (configuração em modo bridge)
eth1 172.16.100.51 (configuração em modo nat com dhcp desativado)

Discos
8G para a instalação do Sistema Operacional
20G para o /var/lib/nova/instances, diretório onde são armazenadas as instâncias
20G para o LVM cinder-volumes, LVM onde são criados os discos adicionais

Preparando o Servidor

Adicionando o repositório do Grizzly

```
~# apt-get install ubuntu-cloud-keyring
```

~# echo deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-updates/grizzly main >> /etc/apt/sources.list.d/grizzly.list

Efetuando o update do Sistema:

```
~# apt-get update ; apt-get upgrade ; apt-get dist-upgrade
~# reboot
```

Instalar outros serviços:

```
~# apt-get install -y vlan bridge-utils
```

Configuração de Rede

Configurar o arquivo /etc/hosts ou adicionar no DNS caso use um dominio válido

```
127.0.0.1       grizzly.vitola.net.br           grizzly
172.16.100.51   eth1.grizzly.vitola.net.br      eth1.grizzly
192.168.0.51    eth0.grizzly.vitola.net.br      eth0.grizzly
```

Configurar o arquivo /etc/network/interfaces

```
auto lo
iface lo inet loopback
auto br100
iface br100 inet static
      address 192.168.0.51
      netmask 255.255.255.0
      gateway 192.168.0.1
      dns-nameservers 8.8.8.8
      bridge_ports eth0
      bridge_stp off
      bridge_maxwait 0
      bridge_fd 0
iface br100 inet6 static
   address 2001:1291:02d2:0000:0000:0000:0000:0051
   netmask 48
   gateway fe80::20c:29ff:fee2:ccdb
auto eth1
iface eth1 inet static
    address 172.16.100.51
    netmask 255.255.255.0
```

Criar a bridge

```
~# brctl addbr br100; service networking restart
```

Reiniciar a rede:

```
~# service networking restart
```

MySQL, RabbitMQ & NTP

Instalando o Mysql e definindo a senha de root:

```
~# apt-get install mysql-server python-mysqldb
```

Configurar o MySQL para aceitar conexões de todos os hosts:

```
~# sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf
~# service mysql restart
```

Configurar o acesso remoto do root no MySQL

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'SENHADOROOT' WITH GRANT OPTION
```

Instalando RabbitMQ:

```
~# apt-get install rabbitmq-server
```

Instalando o serviço de NTP:

```
~# apt-get install ntp
```

Habilitando o IP_Forwarding:

```
~# sed -i 's/#net.ipv4.ip_forward=1/net.ipv4.ip_forward=1/' /etc/sysctl.conf
```

Keystone

Instalando o keystone:

```
~# apt-get install keystone
```

Verificar se o serviço esta rodando:

```
~# service keystone status
```

Criando a base de dados:

```
~# mysql -u root -p
mysql> CREATE DATABASE keystone;
mysql> GRANT ALL ON keystone.* TO 'keystoneUser'@'%' IDENTIFIED BY 'keystonePass';
mysql> quit;
```

Ajustando o arquivo /etc/keystone/keystone.conf:

```
conf> connection = mysql://keystoneUser:keystonePass@eth1.grizzly.vitola.net.br/keystone
```

Reiniciar o serviço e sincronizar a base de dados:

```
~# service keystone restart
~# keystone-manage db_sync
```

Baixar os scripts de criação dos usuários e Tenants:

```
~# cd /usr/local/src
~# wget
https://raw.github.com/mseknibilel/OpenStack-Grizzly-Install-Guide/master/KeystoneScripts/keystone_basic.
sh
~# wget
```

```
https://raw.github.com/mseknibilel/OpenStack-Grizzly-Install-Guide/master/KeystoneScripts/keystone_endpoi
nts_basic.sh
```

```
~# chmod +x keystone_basic.sh
~# chmod +x keystone_endpoints_basic.sh
```

Fazer os ajustes necessários (IP's, nomes e email de contato) e executá-los

```
~# ./keystone_basic.sh
~# ./keystone_endpoints_basic.sh
```

Criar um arquivo de credencial local para gerenciar o servidor usando a linha de comando

```
~# vi ~/.creds
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=admin_pass
export OS_AUTH_URL="http://eth0.grizzly.vitola.net.br:5000/v2.0/"
```

Carregar as informações no ambiente do usuário:

```
~# source ~/.creds
~# echo " source ~/.creds" >> ~/.bash_profile
```

Testar na linha de comando se as informações foram criadas corretamente:

```
~# keystone user-list
```

Glance

Instalar os pacotes do Glance:

```
~# apt-get install glance
```

Verificar se os serviços estão rodando:

```
~# service glance-api status
~# service glance-registry status
```

Criar a base de dados:

```
~# mysql -u root -p
mysql> CREATE DATABASE glance;
mysql> GRANT ALL ON glance.* TO 'glanceUser'@'%' IDENTIFIED BY 'glancePass';
mysql> quit;
```

Atualizar o arquivo /etc/glance/glance-api-paste.ini:

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = eth1.grizzly.vitola.net.br
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = service_pass
```

Atualizar o arquivo /etc/glance/glance-api.conf:

```
sql_connection = mysql://glanceUser:glancePass@eth1.grizzly.vitola.net.br/glance

[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = eth1.grizzly.vitola.net.br
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = service_pass

[paste_deploy]
flavor = keystone
```

Atualizar o arquivo /etc/glance/glance-registry-paste.ini:

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = eth1.grizzly.vitola.net.br
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = service_pass
```

Atualizar o arquivo /etc/glance/glance-registry.conf:

```
sql_connection = mysql://glanceUser:glancePass@private.grizzly-proj02.vitola.net.br/glance

[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = eth1.grizzly.vitola.net.br
auth_port = 35357
```

```
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = service_pass

[paste_deploy]
flavor = keystone
```

Reiniciar os serviços:

```
~# service glance-api restart; service glance-registry restart
```

Sincronizar a base de dados:

```
~# glance-manage db_sync
```

Reiniciar novamente os serviços:

```
~# service glance-registry restart; service glance-api restart
```

Testar o Glance usando uma imagem do Cirros (*):
(*) Uma instalação mínima de ums sistema para realizar testes básicos de funcionamento do ambiente

```
~# glance image-create --name Cirros \
                       --is-public true \
                       --container-format bare \
                       --disk-format qcow2 \
                                                                                                  --location
https://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86_64-disk.img
```

Verificar se o upload foi feito corretamente:

```
~# glance image-list
```

KVM

Ter certeza que oprocessador usado tenha suporte à Virtualização

```
~# apt-get install cpu-checker
~# kvm-ok
   INFO: /dev/kvm exists
   KVM acceleration can be used
```

Instalar os pacotes do kvm e libvirt:

```
~# apt-get install kvm libvirt-bin pm-utils
```

Editar o arquivo /etc/libvirt/qemu.conf com as infomações abaixo:

```
cgroup_device_acl = [
"/dev/null", "/dev/full", "/dev/zero",
"/dev/random", "/dev/urandom",
"/dev/ptmx", "/dev/kvm", "/dev/kqemu",
"/dev/rtc", "/dev/hpet","/dev/net/tun"
]
```

Para evitar conflitos delete as bridge criadas pelo Libvirt:

```
~# virsh net-destroy default
~# virsh net-undefine default
```

Obs.: Como este ambiente tem apenas um node não há necessidade de habilitar o live migration

Reiniciar o Libvirt:

```
~# service libvirt-bin restart
```

Nova-*

Instalar os componentes do Nova:

```
~# apt-get install nova-api \
                   nova-cert \
                   novnc \
                   nova-consoleauth \
                   nova-scheduler \
                   nova-novncproxy \
                   nova-doc \
                   nova-conductor \
                   nova-compute-kvm \
                   nova-network
```

E checar se todos os serviços estão rodando:

```
~# cd /etc/init.d/; for i in $( ls nova-* ); do service $i status; cd; done
```

Obs.: Para evitar conflitos remova as configuração no Quantum, caso contrario o nova irá utiliza-lo por padrão

Localize o ID de endpoint

```
~# keystone endpoint-list | grep 9696
| 37ca5ffb7de4452189b77a9657c168cf | RegionOne |         http://eth0.grizzly.vitola.net.br:9696
```

E em seguida faça a remoção

```
~# keystone endpoint-delete 37ca5ffb7de4452189b77a9657c168cf
```

Localize o ID do serviço do Quantum:

```
~# keystone service-list | grep quantum
| 6519e132ed2d4c5d9867474c4dd320b6 | quantum  | network  | OpenStack Networking service |
```

Remova o serviço

```
~# keystone service-delete 6519e132ed2d4c5d9867474c4dd320b6
```

Remover o usuário quantum

```
~# keystone user-list|grep quantum
| a0827d8d647b4a01bd579dbfc26bdd07 | quantum |   True  | quantum@vitola.net.br |
~# keystone user-delete a0827d8d647b4a01bd579dbfc26bdd07
```

Prepare a Mysql database for Nova:

```
~# mysql -u root -p
mysql> CREATE DATABASE nova;
mysql> GRANT ALL ON nova.* TO 'novaUser'@'%' IDENTIFIED BY 'novaPass';
mysql> quit;
```

Alterar o arquivo /etc/nova/api-paste.ini:

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = eth1.grizzly.vitola.net.br
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = service_pass
signing_dirname = /tmp/keystone-signing-nova
# Workaround for https://bugs.launchpad.net/nova/+bug/1154809
auth_version = v2.0
```

Modifique o arquivo /etc/nova/nova.conf:

```
[DEFAULT]
dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
dhcp_domain=vitola.net.br
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
force_dhcp_release=True
iscsi_helper=ietadm
libvirt_use_virtio_for_bridges=True
connection_type=libvirt
#root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf
verbose=False
ec2_private_dns_show_ip=True
api_paste_config=/etc/nova/api-paste.ini
volumes_path=/var/lib/nova/volumes

#scheduler_driver=nova.scheduler.simple.SimpleScheduler
s3_host=eth1.grizzly.vitola.net.br
ec2_host=eth1.grizzly.vitola.net.br
ec2_dmz_host=eth1.grizzly.vitola.net.br
rabbit_host=eth1.grizzly.vitola.net.br
cc_host=eth1.grizzly.vitola.net.br
metadata_host=192.168.0.51
metadata_listen=0.0.0.0
nova_url=http://eth1.grizzly.vitola.net.br:8774/v1.1/
sql_connection=mysql://novaUser:novaPass@eth1.grizzly.vitola.net.br/nova
ec2_url=http://eth1.grizzly.vitola.net.br:8773/services/Cloud
rootwrap_config=/etc/nova/rootwrap.conf

log-config=/etc/nova/logging.conf
debug = False
use_syslog = False
syslog_log_facility = LOG_LOCAL0

# Auth
use_deprecated_auth=false
auth_strategy=keystone
keystone_ec2_url=http://eth1.grizzly.vitola.net.br:5000/v2.0/ec2tokens

# Imaging service
glance_api_servers=eth1.grizzly.vitola.net.br:9292
image_service=nova.image.glance.GlanceImageService

# Vnc configuration
novnc_enabled=true
```

```
novncproxy_base_url=http://eth0.grizzly.vitola.net.br:6080/vnc_auto.html
novncproxy_port=6080
vncserver_proxyclient_address=eth1.grizzly.vitola.net.br
vncserver_listen=0.0.0.0

# NETWORK
network_manager=nova.network.manager.FlatDHCPManager
force_dhcp_release=True
dhcpbridge_flagfile=/etc/nova/nova.conf
#firewall_driver=nova.virt.firewall.IptablesFirewallDriver
my_ip=192.168.0.51
public_interface=br100
vlan_interface=eth0
flat_network_bridge=br100
flat_interface=eth0
flat_injected=True
compute_driver=libvirt.LibvirtDriver
use_ipv6=True
#gateway_v6=None
#ipv6_backend=rfc2462

# Cinder #
volume_api_class=nova.volume.cinder.API
osapi_volume_listen_port=5900
```

Editar o arquivo /etc/nova/nova-compute.conf:

```
[DEFAULT]
libvirt_type=kvm
compute_driver=libvirt.LibvirtDriver
```

Sincronize a base de dados:

```
~# nova-manage db sync
```

Reinicie todos os serviços nova-* :

```
~# cd /etc/init.d/; for i in $( ls nova-* ); do sudo service $i restart; done
```

Um sorriso indica que esta tudo Ok:

```
~# nova-manage service list
Binary           Host                                 Zone             Status     State Updated_At
nova-consoleauth grizzly.vitola.net.br                internal         enabled    :-)   None
nova-cert        grizzly.vitola.net.br                internal         enabled    :-)   None
nova-conductor   grizzly.vitola.net.br                internal         enabled    :-)   None
nova-scheduler   grizzly.vitola.net.br                internal         enabled    :-)   None
```

```
nova-compute     grizzly.vitola.net.br                nova             enabled    :-)   None
nova-network     grizzly.vitola.net.br                internal         enabled    :-)   None
```

Cinder

Instalar os pacotes do Cinder:

```
~# apt-get install cinder-api cinder-scheduler cinder-volume iscsitarget open-iscsi iscsitarget-dkms
```

Configurar o ISCSI:

```
~# sed -i 's/false/true/g' /etc/default/iscsitarget
```

Reinciar os serviços:

```
~# service iscsitarget restart
~# service open-iscsi restart
```

Criar a base de dados:

```
~# mysql -u root -p
mysql> CREATE DATABASE cinder;
mysql> GRANT ALL ON cinder.* TO 'cinderUser'@'%' IDENTIFIED BY 'cinderPass';
mysql> quit;
```

Configurar o arquivo /etc/cinder/api-paste.ini:

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
service_protocol = http
service_host = eth0.grizzly.vitola.net.br
service_port = 5000
auth_host = eth1.grizzly.vitola.net.br
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = cinder
admin_password = service_pass
```

Configurar o arquivo /etc/cinder/cinder.conf:

```
[DEFAULT]
rootwrap_config=/etc/cinder/rootwrap.conf
sql_connection = mysql://cinderUser:cinderPass@eth1.grizzly.vitola.net.br/cinder
api_paste_config = /etc/cinder/api-paste.ini
```

```
iscsi_helper=ietadm
volume_name_template = volume-%s
volume_group = cinder-volumes
verbose = True
auth_strategy = keystone
#osapi_volume_listen_port=5900
lock_path = /var/lock/cinder
volumes_dir = /var/lib/cinder/volumes
```

Sincronizar a base de dados:

```
~# cinder-manage db sync
```

Criar o volume LVM que será usado para discos adicionais:

```
~# fdisk /dev/sdb

fdisk> n
fdisk> p
fdisk> 1
fdisk> ENTER
fdisk> ENTER
fdisk> t
fdisk> 8e
fdisk> w
```

Criar o volume físico e o volume grupo:

```
~# pvcreate /dev/sdb1
~# vgcreate cinder-volumes /dev/sdb1
```

Reiniciar os serviços do Cinder:

```
~# cd /etc/init.d/; for i in $( ls cinder-* ); do sudo service $i restart; done
```

Horizon

Instalar os pacotes do Horizon

```
~# apt-get install openstack-dashboard memcached
```

Reiniciar o Apache e o memcached:

```
~# service apache2 restart; service memcached restart
```

Criando um Projeto e as VM's

Acessar a URL https://grizzly.vitola.net.br/horizon com os dados abaixo:

username: admin
password: admin_pass

Admin > Projects > Create Project

Project Info
* Name: IPv6-EAD1-Turma2
* Description:
* Enable: Yes
* Project Members: Incluir o usuário admin com a Role admin
* Quota: Não Alterar
* Finish

Anotar o Project ID: d7f616918ee14682b6c735518896540f

Instalar o radv e configurar o sysctl

```
~# apt-get install -y radvd
~# bash -c "echo 1 > /proc/sys/net/ipv6/conf/all/forwarding"
~# bash -c "echo 0 > /proc/sys/net/ipv6/conf/all/accept_ra"
```

Criar uma rede e associar ao projeto IPv6-EAD1-Turma2

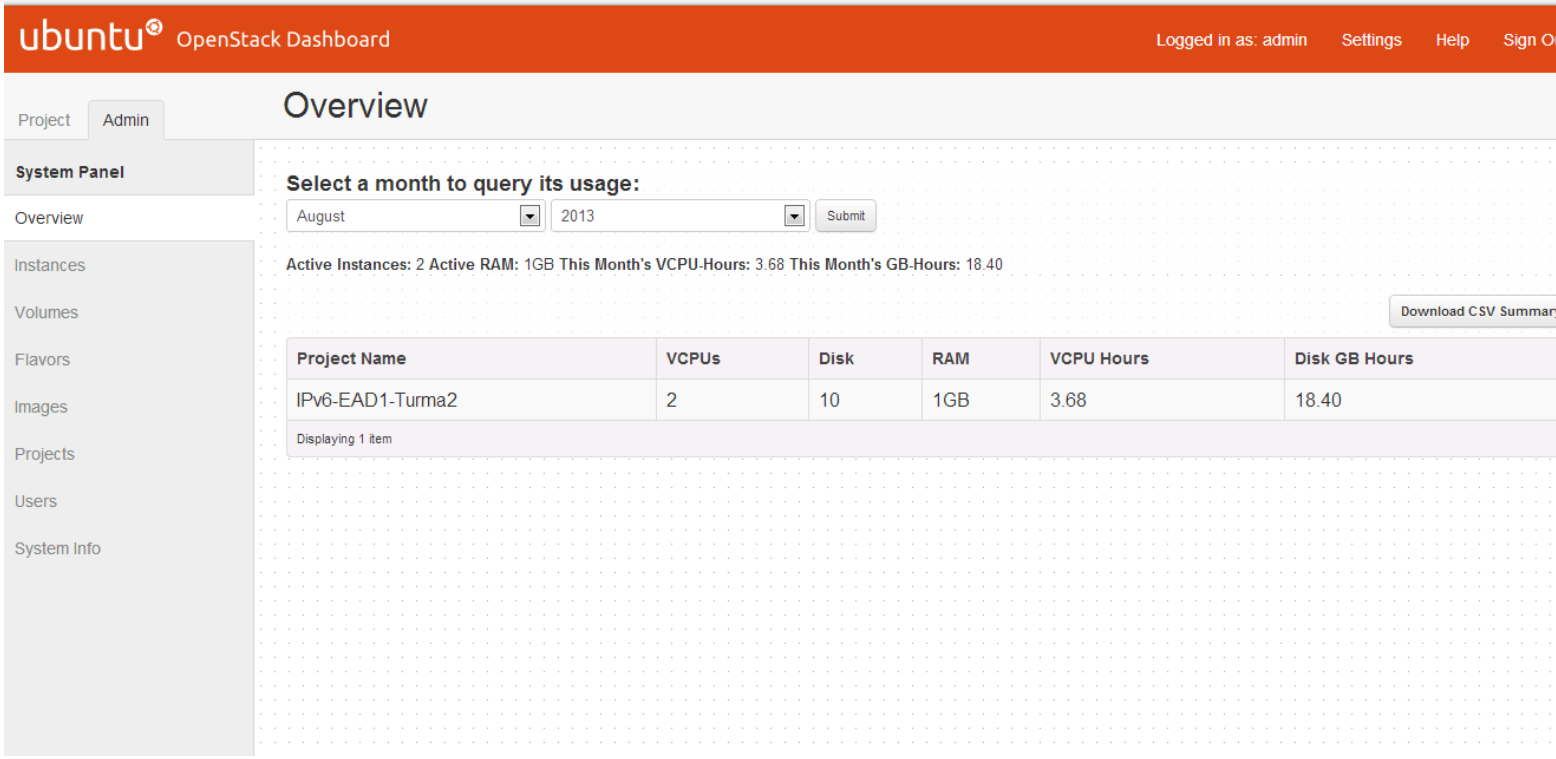
```
~# nova-manage network create \
  --label=rede1 \
  --fixed_range_v4=172.16.50.0/28  \
  --fixed_range_v6=2001:1291:2d2::/48 \
  --bridge=br100 \
  --project_id=d7f616918ee14682b6c735518896540f
```

Verificar se a rede foi criada corretamente

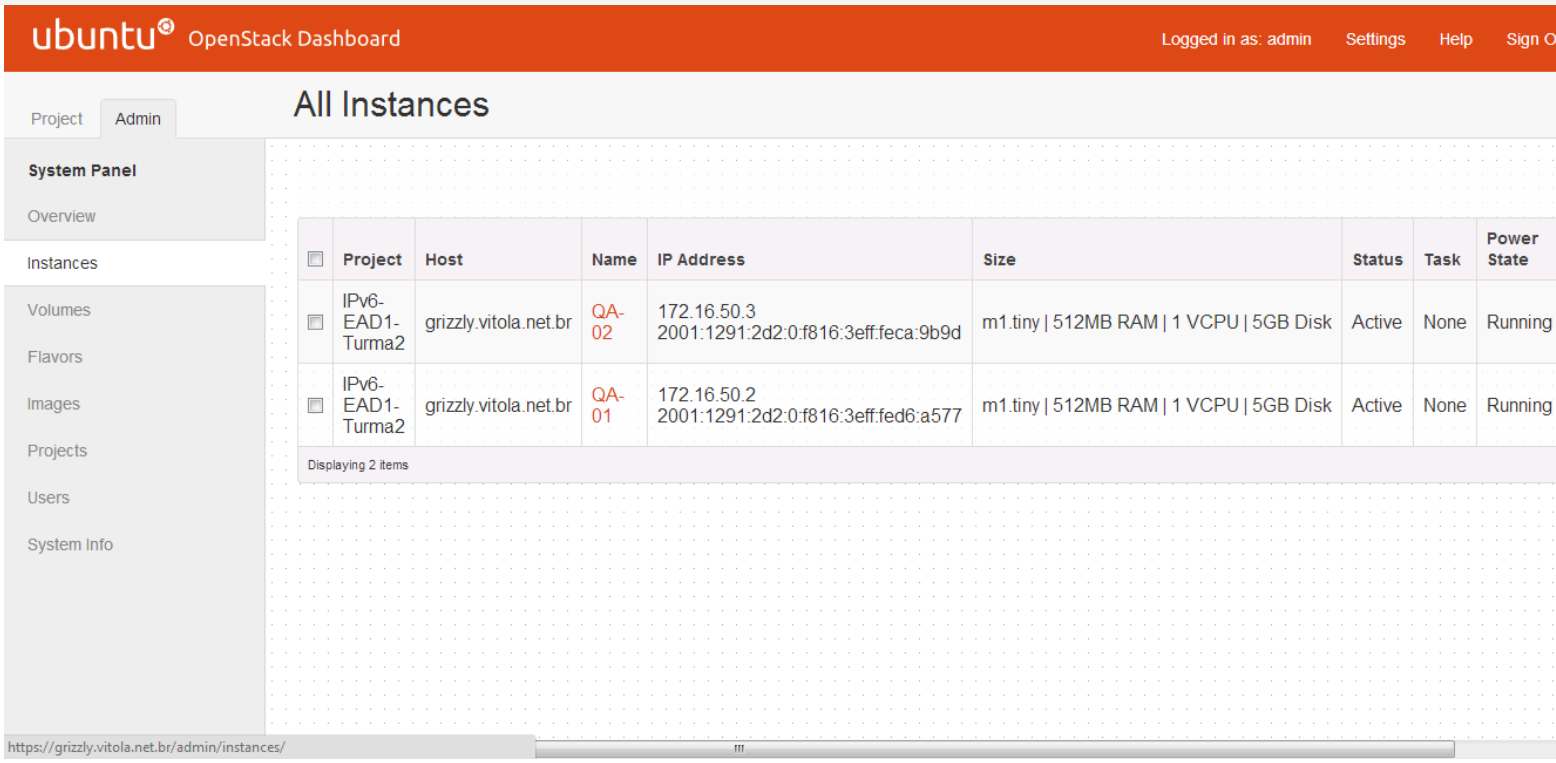
```
~# nova-manage network list
id    IPv4                  IPv6                          start  address     DNS1         DNS2    VlanID  project
uuid
2           172.16.50.0/28    2001:1291:2d2::/64    172.16.50.2                  8.8.4.4       None       None
d7f616918ee14682b6c735518896540f    95fbde3a-8831-431f-918d
```

Quando criadas o nova-network iniciar uma instancia do dnsmasq com as configurações que cada máquina deve utilizar.

Abaixo uma imagem do Painel onde mostra algumas informações básicas do ambiente



Na imagem abaixo as intâncias criadas com seus respectivos ipv4 e ipv6



Abaixo as duas maquinas criadas respondendo pelo IPv6

Selecionar Administrador: C:\Windows\system32\cmd.exe

```
C:\Users\vitola>ping 2001:1291:2d2:0:f816:3eff:feca:9b9d

Disparando 2001:1291:2d2:0:f816:3eff:feca:9b9d com 32 bytes de dados:
Resposta de 2001:1291:2d2:0:f816:3eff:feca:9b9d: tempo=1ms
Resposta de 2001:1291:2d2:0:f816:3eff:feca:9b9d: tempo<1ms
Resposta de 2001:1291:2d2:0:f816:3eff:feca:9b9d: tempo<1ms
Resposta de 2001:1291:2d2:0:f816:3eff:feca:9b9d: tempo<1ms

Estatísticas do Ping para 2001:1291:2d2:0:f816:3eff:feca:9b9d:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de
             perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 0ms, Máximo = 1ms, Média = 0ms

C:\Users\vitola>ping 2001:1291:2d2:0:f816:3eff:fed6:a577

Disparando 2001:1291:2d2:0:f816:3eff:fed6:a577 com 32 bytes de dados:
Resposta de 2001:1291:2d2:0:f816:3eff:fed6:a577: tempo=1ms
Resposta de 2001:1291:2d2:0:f816:3eff:fed6:a577: tempo<1ms
Resposta de 2001:1291:2d2:0:f816:3eff:fed6:a577: tempo<1ms
Resposta de 2001:1291:2d2:0:f816:3eff:fed6:a577: tempo<1ms

Estatísticas do Ping para 2001:1291:2d2:0:f816:3eff:fed6:a577:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de
             perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 0ms, Máximo = 1ms, Média = 0ms

C:\Users\vitola>_
```