Руководство по установке и настройке OpenStack

Выпуск 1.0

Леонид Лабошин

07 March 2013

Оглавление

OpenStack — это комплекс проектов свободного (Apache License, Version 2.0) программного обеспечения, предназначенного для создания вычислительных облаков и облачных хранилищ.

На данный момент OpenStack содержит семь ключевых компонентов:

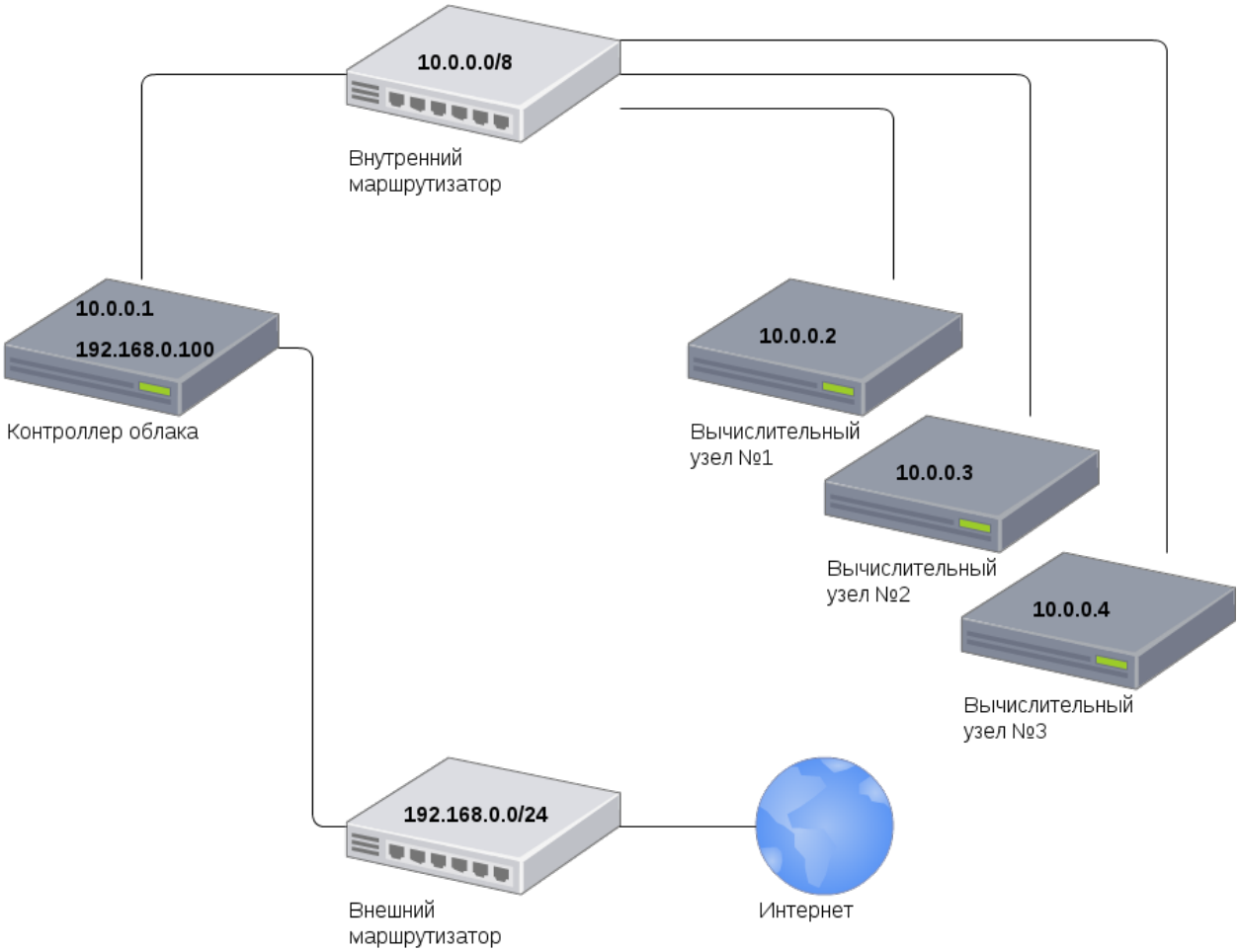
- Сервис авторизации и аутентификации «Keystone»
- Хранилище образов «Glance»
- Вычислительный сервис «Nova»
- Сетевой сервис Quantum
- Сервис хранения блоков данных «Cinder»
- Хранилище объектов «Swift»
- Веб-интерфейс «Horizon»

В данное руководство включена так же установка

- Распределенной файловой системы CEPH
- Системы мониторинга Munin
- Сервера доменных имен MyDNS

Подготовка к установке

1.1 Структура облака



1.2 Подготовка узлов

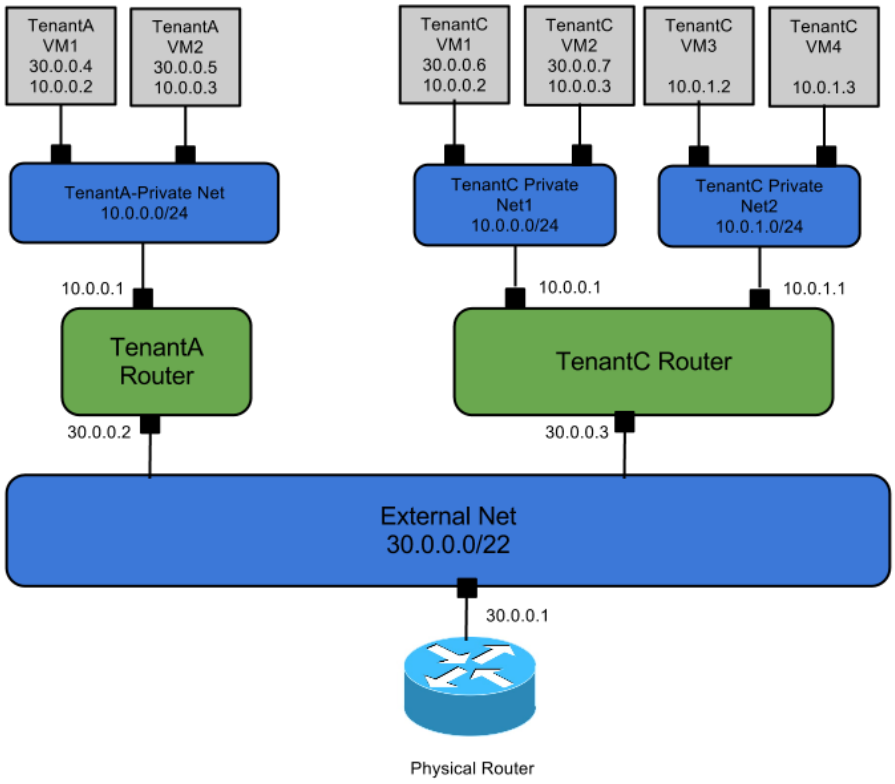
Установка сервисов OpenStack будет производиться из пакетов, на основе операционной системы Ubuntu 12.04 «Precise», с использованием системы автоматизации процесса установки и настройки программного обеспечения Chef.

В данном руководстве рассмаривается структура облака, когда все облачные сервисы, кроме вычислительного, расположены на одном узле, называемом “контроллером облака”, к которому по локальной сети подключаются вычислительные узлы.

Вычислительный сервис nova-compute использует гипервизор KVM

Публичным и приватным адресам вируальных машин с использованием MyDns автоматически выдаются доменные имена.

Хранилища образов, дисков виртуальных машин и объектов располагаются на LVM томах.



За работу с сетевой частью OpenStack отвечает библиотека Quantum, которая обеспечивает функцию «сеть как сервис» между сетевыми интерфейсами ВМ (vNIC) под управлением других сервисов OpenStack, фактически предоставляя API, позволяющее управлять всей сетевой частью облака. В зависимости от поставленных задач и спроектированной целевой конфигурации облака, к Quantum

можно подключать плагины, такие как Open vSwith, Cisco UCS/Nexus, Linux Brige, NEC OpenFlow, Nicira Network Virtualization Platform (NVP) и некоторые другие.

Наиболее продвинутый вариант реализации сетевой инфраструктуры, в котором каждый(!) тенант получает приватный роутер, с возможностью создания дополнительных роутеров для каждого тенанта через Quantum API. Тенант может создавать свои сети, с возможностью подключения к роутеру. Теперь самое главное: данная схема позволяет каждому тенанту использовать любые сети, т.к. доступ вовне обеспечивается или через SNAT или Floating IPs. Иными словами, в облаке может быть несколько ВМ с одинаковыми(!) внутренними IP-адресами. Это может пригодиться, например, при переходе с одного облака на другое – запаковал машины, слил образ, настроил требуемую инфраструктуру на другом облаке, назначил IP-адреса, которые у тебя были ранее, развернул образы и все полетело без дополнительных изменений. Тот, кто часто вынужден был переносить сервера из одной подсети в другую, наверняка оценят эту возможность. С другой стороны, как часто вам может потребоваться таскать свою инфраструктуру между разными облаками?

Подготовка сетевых интерфейсов:

```
# Public interface
auto eth0
iface eth0 inet static
address 195.208.117.140
netmask 255.255.255.224

# Private interface
auto br-int
iface br-int inet static
        bridge_ports eth1
        address 10.10.10.0
        netmask 255.0.0.0
```

В настоящем руководстве для установки компонентов облачной инфраструктуры OpenStack, мы будем использовать продукт OpstChef. Это Open-Source инструмент управления инфраструктурой серверов. Мы будем рассматривать упрощенный вариант использования этого инструмента, без настройки Chef-сервера, а ограничимся использованием knife-solo.

Первым делом на управляющей машине необходимо установить менеджер пакетов для языка программирования Руби - RubyGems. В операционной системе Ubuntu сделать это можно с помощью стандартного пакетного менеджера командой:

```
sudo apt-get install -y rubygems
```

```
sudo gem-install knife-solo
```

```
knife configure -r .defaults
```

Используемые скрипты для chef расчитаны на использование на будующих узлах облачной инфраструктуры опреационной системы Ubuntu версии 12.04, и налагает следующие требования:

```
set[:mysql][:password]="mySuperSecret"

set[:controller][:private_ip]="10.10.10.1"
set[:controller][:public_ip]="192.168.0.101"

set[:keystone][:token]="mySuperSecret"
set[:keystone][:password]="mySuperSecret"
set[:keystone][:email]="admin@post.domain.ru"

set[:dns][:zone]="cloud.domain.ru"
```

Работа с облачной системой OpenStack, помимо Web-интерфейса, может осуществляться с помощью

nova-client или инструмента euca2ools. В данном руководстве мы постараемся приводить команды для обоих этих инструментов.

Для использования команд nova, на клиентской машине необходимо установить пакет python-novaclient:

sudo apt-get install python-novaclient

и аутентифицироваться с помощью openrc файла для нужного проекта, который можно загрузить через Web-интерфейс, на вкладке settings

---

Примечание: Добавьте в файл openrc строчку export OS_NO_CACHE=1 это избавит Вас от необходимости каждый раз вводить пароль связки ключей, при использовании последней версии клиента

После загрузки файла выполните команду:

source openrc.sh

Для использования команд euca, на клиентской машине необходимо установить пакет euca2ools:

sudo apt-get install euca2ools

и аутентифицироваться с помощью сертификатов EC2. Скачайте zip-архив на странице настроек проекта, распакуйте файлы и выполните

source ec2rc.sh

Для запуска виртуальных в облачной инфраструктуре OpenStack используются образы дисков операционных систем. В последней версии Glance добавлена также возможность создания виртуальной машины и установки операционной системы из iso-образа. Мы рассмотрим два основных формата образов дисков:

- AMI (англ. Amazon Machine Image). Образ операционной системы в этом формате состоит из трех частей: AKI (англ. Amazon Kernel Image) , ARI (англ. Amazon Ramdisk Image) AMI
- QCOW2 - это формат дискового образа программы QEMU. Название является аббревиатурой названия формата Copy-On-Write (копирование при записи).

Первым шагом будет создание пустого файла образа диска. Современная операционная система семейства Windows требует для работы большое количество дискового пространства, не менее чем 20 GB.

kvm-img create -f raw windowsserver.img 20G

Openstack использует интерфейс Virtio для дисков и сетевых адаптеров при запуске виртуальных машин. Это означает, что операционная система виртуальной машины должна иметь драйверы для Virtio. По умолчанию операционные машины семейства Windows не содержат таких драйверов, их необходимо предоставить операционной системе в процессе установки. Образ дискеты с последней версией необходимых драйверов доступен для скачивания с сайта проекта fedora Процесс установки Windows можно запустить с помощью команды.

kvm -m 1024 -cdrom windows.iso -drive file=windowsserver.img, if=virtio,boot=on -fda virtio-win-1.1.16.vfd -boot d -nographic -vnc :0

Здесь параметром cdrom указывается путь к образу установочного диска операционной системы, например windows.iso. Параметр drive - предварительно созданный образ жесткого диска. Параметром fda задается путь к загруженному образу дискеты с драйверами windows. После запуска команды начнется стандартный процесс установки Windows. К Консоли управления можно подключиться с помощью любого обозревателя VNC, например vncviewer:

sudo apt-get install vncviewer -y

на порт 5900.

vncviewer localhost:5900

Если создание образа производится на удаленной машине, подключиться к VNC-консоли можно с использованием ssh-туннеля, например так:

vncviewer -via "laboshinl@192.168.0.100 -p 22"localhost:0

Необходимо следовать инструкциям, появляющимся на экране. В процессе установки в окне выбора жесткий диск, не будет выведено каких-либо устройств. Необходимо нажать на кнопку “Загрузить драйверы”, в левом нижнем углу и указать путь к подключенным драйверам после чего продолжить установку

Примечание: Для удобства дальнейшего использования образа рекомендуется после завершения установки разрешить удаленное администрирование системы через RDP(RemoteDesktopProtocol)

Тестовые образы от Cirros

Официальные образы релизов Ubuntu

Готовые образы некоторых операционных систем доступны так же на сайте нашего проекта

nova cloudpipe-create $project_ID

Шаблон конфирурационного файла для openvpn

```
# Edit the following lines to point to your cert files:
cert cert.pem
key pk.pem

ca cacert.pem

client
dev tap
proto udp

remote $controller_public_ip $port
resolv-retry infinite
nobind

# Downgrade privileges after initialization (non-Windows only)
user nobody
group nogroup
comp-lzo

# Set log file verbosity.
verb 2

keepalive 10 120
ping-timer-rem
persist-tun
persist-key
```

CEPH

Ceph — свободная распределённая файловая система. Ceph может использоваться на системах, состоящих как из нескольких машин, так и из тысяч узлов. Общий объем хранилища данных может измеряться петабайтами, встроенные механизмы продублированной репликации данных (не зависит от отказа отдельных узлов) обеспечивают чрезвычайно высокую живучесть системы, при добавлении или удалении новых узлов, массив данных автоматически перебалансируется с учетом новшеств.

Установка

gpg --keyserver keyserver.ubuntu.com --recv 17ED316D

gpg --export --armor 17ED316D | apt-key add -

Необходимо добавить репозиторий “Grizzly”

echo "deb http://ppa.launchpad.net/openstack-ubuntu-testing/grizzly-trunk-testing/ubuntu/ precise main">> /etc/apt/sources.list

Получение ключа

gpg --keyserver keyserver.ubuntu.com --recv 3B6F61A6 && gpg --export --armor 3B6F61A6 | apt-key add -

Обновление списка пакетов

apt-get update

apt-get install mysql-server python-mysqldb -y

sed -i 's/127.0.0.1/10.10.10.0/g' /etc/mysql/my.cnf

Подсказка: Здесь 10.10.10.0 ip-адрес сетевого интерфейса во внутренней сети

service mysql restart

apt-get install rabbitmq-server

Keystone

3.1 Установка

apt-get install keystone

mysql -uroot -pMysqlPass -e "CREATE DATABASE keystone;"

mysql -uroot -pMysqlPass -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'MysqlPass';"

Подсказка: Здесь и далее MysqlPass - пароль, введенный при установке пакета mysql-server

---

В конфигурационном файле /etc/keystone/keystone.conf необходимо:

- раскоментировать строчку и изменить токен в секции [DEFAULT]:

  admin_token = AdminToken

- в секции [sql] указать путь к созданной базе данных:

  connection = mysql://root:MysqlPass@10.10.10.0/keystone

- секцию [catalog] привести к следующему виду:

```
# dynamic, sql-based backend (supports API/CLI-based management commands)
# driver = keystone.catalog.backends.sql.Catalog

# static, file-based backend (does *NOT* support any management commands)
driver = keystone.catalog.backends.templated.TemplatedCatalog
template_file = default_catalog.templates
```

- в секции [signing]:

  token_format = UUID

Перезапуск сервиса

service keystone restart

Синхронизация с базой данных

keystone-manage db_sync

Аутентификация

export SERVICE_TOKEN=AdminToken

export SERVICE_ENDPOINT="http://10.10.10.0:35357/v2.0"

Для дальнейшей работы необходимо создать два проекта. Проект, роль, и пользователь admin, необходим для функционирования сервисов и администрирования облака.

keystone tenant-create --name=admin

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |                                  |
|   enabled   |               True               |
|      id     | 1f155208db0a4c959365a0002b8b507e |
|     name    |              admin               |
+-------------+----------------------------------+
```

keystone user-create --name=admin --pass=cl0udAdmin --email=cloud@admin.com

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|  email   |         cloud@admin.com          |
| enabled  |               True               |
|    id    | 1d2a73ea87f249769f6669ee2f812932 |
|   name   |              admin               |
| tenantId |                                  |
+----------+----------------------------------+
```

keystone role-create --name=admin

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|    id    | 424f7b79893c4266bf5753894a4668d2 |
|   name   |              admin               |
+----------+----------------------------------+
```

keystone user-role-add --user-id 1d2a73ea87f249769f6669ee2f812932 --role-id 424f7b79893c4266bf5753894a4668d2 --tenant-id 1f155208db0a4c959365a0002b8b507e

Роль Member - роль по умолчанию для добавления пользователей облака. Пользователь tester и проект test необходимы для проверки работы сервисов облачной инфраструктуры после установки.

keystone tenant-create --name=test

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |                                  |
|   enabled   |               True               |
|      id     | 37cfbd624d0242b995fa695d8b134bb6 |
|     name    |               test               |
+-------------+----------------------------------+
```

keystone user-create --name=tester --pass=cl0udAdmin --email=cloud@admin.com

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|  email   |         cloud@admin.com          |
| enabled  |               True               |
|    id    | cf0828666bfd4a24b12dcd83848ef360 |
|   name   |              tester              |
| tenantId |                                  |
+----------+----------------------------------+
```

keystone role-create --name=Member

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|    id    | 01242eec84c14106a10759e210c98dee |
|   name   |              Member              |
+----------+----------------------------------+
```

keystone user-role-add --user-id cf0828666bfd4a24b12dcd83848ef360 --role-id 01242eec84c14106a10759e210c98dee --tenant-id 37cfbd624d0242b995fa695d8b134bb6

Файл /etc/keystone/default_catalog.templates необходимо привести к следующему виду

```
# config for TemplatedCatalog, using camelCase because I don't want to do
# translations for keystone compat
catalog.RegionOne.identity.publicURL = http://195.208.117.140:$(public_port)s/v2.0
catalog.RegionOne.identity.adminURL = http://195.208.117.140:$(admin_port)s/v2.0
catalog.RegionOne.identity.internalURL = http://195.208.117.140:$(public_port)s/v2.0
catalog.RegionOne.identity.name = Identity Service

# fake compute service for now to help novaclient tests work
catalog.RegionOne.compute.publicURL = http://195.208.117.140:$(compute_port)s/v1.1/$(tenant_id)s
catalog.RegionOne.compute.adminURL = http://195.208.117.140:$(compute_port)s/v1.1/$(tenant_id)s
catalog.RegionOne.compute.internalURL = http://195.208.117.140:$(compute_port)s/v1.1/$(tenant_id)s
catalog.RegionOne.compute.name = Compute Service

catalog.RegionOne.volume.publicURL = http://195.208.117.140:8776/v1/$(tenant_id)s
catalog.RegionOne.volume.adminURL = http://195.208.117.140:8776/v1/$(tenant_id)s
catalog.RegionOne.volume.internalURL = http://195.208.117.140:8776/v1/$(tenant_id)s
catalog.RegionOne.volume.name = Volume Service

catalog.RegionOne.ec2.publicURL = http://195.208.117.140:8773/services/Cloud
catalog.RegionOne.ec2.adminURL = http://195.208.117.140:8773/services/Admin
catalog.RegionOne.ec2.internalURL = http://195.208.117.140:8773/services/Cloud
catalog.RegionOne.ec2.name = EC2 Service

catalog.RegionOne.image.publicURL = http://195.208.117.140:9292/v1
catalog.RegionOne.image.adminURL = http://195.208.117.140:9292/v1
catalog.RegionOne.image.internalURL = http://195.208.117.140:9292/v1
catalog.RegionOne.image.name = Image Service

catalog.RegionOne.network.publicURL = http://195.208.117.140:9696
catalog.RegionOne.network.adminURL = http://195.208.117.140:9696
catalog.RegionOne.network.internalURL = http://195.208.117.140:9696
catalog.RegionOne.network.name = Network Service

catalog.RegionOne.object_store.publicURL = http://195.208.117.140:8080/v1/AUTH_$(tenant_id)s
catalog.RegionOne.object_store.adminURL = http://195.208.117.140:8080/
```

catalog.RegionOne.object_store.internalURL = http://195.208.117.140:8080/v1/AUTH_$(tenant_id)s
catalog.RegionOne.object_store.name = S3 Service

Примечание: Здесь и далее 195.208.117.140 ip-адрес сетевого интерфейса контроллера облака во внешней сети

3.2 Использование

keystone user-list

```
+----------------------------------+--------+---------+----------------+
|                id                |  name  | enabled |      email     |
+----------------------------------+--------+---------+----------------+
| 1d2a73ea87f249769f6669ee2f812932 | admin  |   True  | cloud@admin.com |
| cf0828666bfd4a24b12dcd83848ef360 | tester |   True  | cloud@admin.com |
+----------------------------------+--------+---------+----------------+
```

Glance

4.1 Установка

apt-get install glance

mysql -uroot -pMysqlPass -e "CREATE DATABASE glance;"

В конфигурационных файлах /etc/glance glance-api.conf и /etc/glance/glance-registry.conf необходимо изменить:

```
[DEFAULT]
sql_connection = mysql://root@MysqlPass@10.10.10.0/glance

[keystone_authtoken]
auth_host = 127.0.0.1
auth_port = 35357
auth_protocol = http
admin_tenant_name = admin
admin_user = admin
admin_password = cl0udAdmin

[paste_deploy]
flavor = keystone
```

service glance-api restart

service glance-registry restart

glance-manage db_sync

> Предупреждение: Glance требует версию пакета warlock>=0.7.0,<2 а в репозитории Ubuntu 'Precise' версия 0.1.0, необходимо установить свежую версию с помощью pip install

apt-get install python-pip

pip install warlock --upgrade

4.2 Команды Glance

Загрузка тестового образа

glance image-create --name cirros-0.3.0 --is-public true --container-format bare --disk-format qcow2 --copy-from https://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86_64-disk.img

---

Примечание: Для тестового образа cirros-0.3.0 помимо ssh-ключа для авторизации можно использовать логин cirros и пароль cubswin:)

Nova

5.1 Установка

apt-get install -y nova-api nova-cert novnc nova-consoleauth nova-scheduler

mysql -uroot -pMysqlPass -e "CREATE DATABASE nova;"

В файле /etc/nova/api-paste.ini:

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = 127.0.0.1
auth_port = 35357
auth_protocol = http
admin_tenant_name = admin
admin_user = admin
admin_password = cl0udAdmin
signing_dir = /tmp/keystone-signing-nova
```

В файле /etc/nova/nova.conf:

```
[DEFAULT]
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/run/lock/nova
verbose=True
api_paste_config=/etc/nova/api-paste.ini
compute_scheduler_driver = nova.scheduler.filter_scheduler.FilterScheduler
s3_host=10.10.10.0
ec2_host=10.10.10.0
ec2_dmz_host=10.10.10.0
rabbit_host=10.10.10.0
cc_host=10.10.10.0
dmz_cidr=169.254.169.254/32
metadata_host=10.10.10.0
metadata_listen=0.0.0.0
nova_url=http://10.10.10.0:8774/v1.1/
sql_connection=mysql://root:MysqlPass@10.10.10.0/nova
ec2_url=http://10.10.10.0:8773/services/Cloud
root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf
```

```
# Auth
use_deprecated_auth=false
auth_strategy=keystone
keystone_ec2_url=http://10.10.10.0:5000/v2.0/ec2tokens
# Imaging service
glance_api_servers=10.10.10.0:9292
image_service=nova.image.glance.GlanceImageService

# Vnc configuration
novnc_enabled=true
novncproxy_base_url=http://195.208.117.140:6080/vnc_auto.html
novncproxy_port=6080
vncserver_proxyclient_address=195.208.117.140
vncserver_listen=0.0.0.0

# Network settings
network_api_class=nova.network.quantumv2.api.API
quantum_url=http://10.10.10.0:9696
quantum_auth_strategy=keystone
quantum_admin_tenant_name=service
quantum_admin_username=quantum
quantum_admin_password=service_pass
quantum_admin_auth_url=http://10.10.10.0:35357/v2.0
libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
linuxnet_interface_driver=nova.network.linux_net.LinuxOVSInterfaceDriver
firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver

# Compute #
compute_driver=libvirt.LibvirtDriver

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

Синхронизация с базой

```
nova-manage db_sync
```

```
apt-get install -y kvm libvirt-bin pm-utils nova-conductor
```

Перезапуск сервисов

```
find /etc/init.d -name nova* -exec {} restart \;
```

Подсказка: Посмотреть список работающих сервисов Nova можно командой nova-manage service list

---

Cinder

6.1 Установка

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

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

service iscsitarget start

service open-iscsi start

mysql -uroot -pMysqlPass -e "CREATE DATABASE cinder;"

В файле /etc/cinder/api-pate.ini:

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
service_protocol = http
service_host = 127.0.0.1
service_port = 5000
auth_host = 127.0.0.1
auth_port = 35357
auth_protocol = http
admin_tenant_name = admin
admin_user = admin
admin_password = cl0udAdmin
signing_dir = /var/lib/cinder
```

В файле /etc/cinder/cinder.conf:

```
[DEFAULT]
rootwrap_config = /etc/cinder/rootwrap.conf
sql_connection = mysql://root:MysqlPass@10.10.10.0/cinder
api_paste_confg = /etc/cinder/api-paste.ini
iscsi_helper = tgtadm
volume_name_template = volume-%s
volume_group = tn0
verbose = True
auth_strategy = keystone
state_path = /var/lib/cinder
volumes_dir = /var/lib/cinder/volumes
```

Подсказка: Здесь tn0 - название группы логических томов lvm2

cinder-manage db sync

service cinder-volume restart

service cinder-api restart

Dashboard

7.1 Установка

apt-get install openstack-dashboard memcached node-less

Quantum

8.1 Установка

pip install cliff –upgrade

Глава 9

Swift

9.1 Установка