

OpenStack 構築手順書

Havana 版

Ver2.2

2014 年 3 月 20 日作成



VirtualTech Japan

日本仮想化技術株式会社

目次

1. 構築する環境について	5
1-1 作成するノード	5
1-2 仮想マシンの設定	5
1-3 ネットワーク全体の設定	5
1-4 各ノードのネットワーク設定	6
1-5 Ubuntu のインストール	7
1-6 Ubuntu へのログインと root 権限.....	8
1-7 設定ファイル等の記述について	8
1-8 作業上の注意点.....	9
2. controller ノードの設定.....	10
2-1 ネットワークの設定.....	10
2-1-1 IP アドレスの設定	10
2-1-2 静的名前解決の設定	11
2-2 systemctl によるシステムの設定.....	11
2-3 apt の設定	11
2-4 NTP のインストール	12
2-5 Python 用 MySQL クライアントのインストール	12
2-6 MySQL サーバーのインストール	12
2-6-1 パッケージのインストール	12
2-6-2 バインドアドレスの変更	12
2-6-3 サービスの再起動	12
2-6-4 データベースとユーザーの作成	12
2-7 RabbitMQ のインストール.....	13
2-7-1 パッケージのインストール	13
2-7-2 guest ユーザーのパスワード変更	13
2-8 Keystone のインストール.....	14
2-8-1 パッケージのインストール	14
2-8-2 設定の変更.....	14
2-8-3 使用しないデータベースファイルの削除.....	14
2-8-4 データベースに表を作成.....	14
2-8-5 サービスの再起動	14
2-8-6 ログの確認.....	14
2-8-7 各種認証情報設定用スクリプトの作成と実行	15
2-8-8 環境変数設定ファイルの作成と読み込み	18
2-8-9 Keystone の動作確認	18
2-9 Glance のインストール	19
2-9-1 パッケージのインストール	19
2-9-2 JSON ファイルの取得.....	19
2-9-3 使用しないデータベースファイルの削除.....	19

2-9-4 設定の変更.....	19
2-9-5 データベースに表を作成.....	20
2-9-6 サービスの再起動.....	20
2-9-7 ログの確認.....	20
2-9-8 CirrOS イメージの取得と登録.....	20
2-9-9 Ubuntu Server イメージの取得と登録(任意).....	21
2-10 Nova のインストール.....	22
2-10-1 パッケージのインストール.....	22
2-10-2 設定の変更.....	22
2-10-3 使用しないデータベースファイルの削除.....	23
2-10-4 データベースに表を作成.....	23
2-10-5 サービスの再起動.....	24
2-10-6 ログの確認.....	24
2-11 Cinder のインストール.....	24
2-11-1 パッケージのインストール.....	24
2-11-2 設定の変更.....	24
2-11-3 使用しないデータベースファイルの削除.....	25
2-11-4 データベースに表を作成.....	25
2-11-5 サービスの再起動.....	25
2-11-6 ログの確認.....	25
2-11-7 イメージ格納用ボリュームの作成.....	25
2-11-8 サービスの再起動.....	25
2-11-9 ログの確認.....	26
2-12 Neutron のインストール.....	26
2-12-1 パッケージのインストール.....	26
2-12-2 設定の変更.....	26
2-12-3 使用しないデータベースファイルの削除.....	27
2-12-4 サービスの再起動.....	27
2-12-5 ログの確認.....	27
2-13 Horizon のインストール.....	27
2-13-1 パッケージのインストール.....	27
2-13-2 設定の変更.....	27
2-13-3 サービスの再起動.....	28
2-13-4 Web 管理画面へのアクセス.....	28
3. network ノードの設定.....	29
3-1 ネットワークの設定.....	29
3-1-1 IP アドレスの設定.....	29
3-1-2 静的名前解決の設定.....	29
3-2 systemctl によるシステムの設定.....	30
3-3 apt の設定.....	30

3-4 NTP のインストール	30
3-5 Python 用 MySQL クライアントのインストール	31
3-6 Neutron のインストール	31
3-6-1 パッケージのインストール	31
3-6-2 設定の変更	31
3-6-3 使用しないデータベースファイルの削除	33
3-6-4 サービスの再起動	33
3-6-5 ログの確認	33
3-7 仮想ネットワークの設定	34
3-7-1 ブリッジの作成	34
3-7-2 ネットワークインターフェースの設定変更	34
3-7-3 システム再起動	34
3-7-4 ネットワークインターフェースの確認	35
3-7-5 環境変数設定ファイルの作成と読み込み	35
3-7-6 インスタンス用ネットワーク作成スクリプトの作成	35
3-7-7 外部接続用ネットワークの作成	36
3-8 ネットワークの確認	37
3-8-1 管理者ユーザーとしての確認	37
3-8-2 一般ユーザーとしての確認	37
4. compute1 ノードの設定	39
4-1 ネットワークの設定	39
4-1-1 IP アドレスの設定	39
4-1-2 静的名前解決の設定	40
4-2 sysctl によるシステムの設定	40
4-3 apt の設定	40
4-4 NTP のインストール	41
4-5 Python 用 MySQL クライアントのインストール	41
4-6 Nova (nova-compute) のインストール	41
4-6-1 パッケージのインストール	41
4-6-2 Linux カーネルのパーミッションの修正	41
4-6-3 設定の変更	42
4-6-4 使用しないデータベースファイルの削除	43
4-6-5 サービスの再起動	43
4-6-6 ログの確認	43
4-7 Neutron のインストール	43
4-7-1 パッケージのインストール	43
4-7-2 ブリッジの作成	43
4-7-3 設定の変更	43
4-7-4 サービスの再起動	44
4-7-5 ログの確認	44

1. 構築する環境について

1-1 作成するノード

今回構築するOpenStack環境は、以下の3ノードで構成します。

- **controller**ノード
OpenStack環境全体を管理するコントローラーとして機能します。
- **network**ノード
外部ネットワークとインスタンスの間のネットワークを制御します。
- **compute1**ノード
仮想マシンインスタンスを実行します。

1-2 仮想マシンの設定

本手順書はMac OS X上で動作するParallels Desktop 9を使用した環境での構築手順を解説しています。その他の仮想マシンソフトウェアを使用する場合は、適宜用語、設定などを読み替えて下さい。

- **controller**ノードにゲストOSのイメージを配置するので、仮想ハードディスクを別途追加しておきます。Linuxからは/dev/sdbとして認識されます。
- **compute1**ノードはCPU仮想化機能を有効にしておきます。
- ネットワークインターフェースを2つ作成します。1つ目をホストオンリーネットワーク、2つ目を共有ネットワークに接続します。
※Parallels Desktopの仕様上、外部に接続できる共有ネットワーク(NAT)が1つしか作成できないため、このような設定となっています。
外部へNATできる仮想ネットワークを複数作成できる仮想マシンソフトウェアの場合には、両方のネットワークを共有ネットワークに接続しても構いません。

1-3 ネットワーク全体の設定

- 内部ネットワーク
ノード間の通信のために使用するネットワーク。インターネットへの接続は行えなくても構いません。

- 外部ネットワーク

外部との接続に使用するネットワーク。OpenStackのFloating IPを利用して、クライアントからcompute1ノード上のインスタンスへと接続できます。

aptコマンドを使って外部リポジトリからパッケージなどをダウンロードするため、外部接続が必要となります。今回はeth1から外部インターネット接続を行っています。各種APIを外部公開する際にも使用できますが、今回は公開は行っていません。

	内部ネットワーク	外部ネットワーク
インターフェース	eth0	eth1
仮想化ソフト	ホストオンリーネットワーク	共有ネットワーク
ネットワーク	192.168.0.0/24	10.0.0.0/24
ゲートウェイ	なし	10.0.0.1
ネームサーバー	なし	10.0.0.1

1-4 各ノードのネットワーク設定

- controllerノード

インターフェース	eth0	eth1
IPアドレス	192.168.0.10	10.0.0.10
ネットマスク	255.255.255.0	255.255.255.0
ゲートウェイ	なし	10.0.0.1
ネームサーバー	なし	10.0.0.1

- networkノード

インターフェース	eth0	eth1
IPアドレス	192.168.0.9	10.0.0.9
ネットマスク	255.255.255.0	255.255.255.0
ゲートウェイ	なし	10.0.0.1
ネームサーバー	なし	10.0.0.1

- compute1ノード

インターフェース	eth0	eth1
IPアドレス	192.168.0.11	10.0.0.11
ネットマスク	255.255.255.0	255.255.255.0
ゲートウェイ	なし	10.0.0.1
ネームサーバー	なし	10.0.0.1

1-5 Ubuntu のインストール

- 優先ネットワークインターフェースの指定
インストール時は優先ネットワークインターフェース(Primary network interface)に選択したインターフェースのみDHCPで設定されるので、外部インターネットに接続できる側のネットワークインターフェースを選択します。
今回の環境ではeth1を選択します。
- OpenSSH serverをインストールする
compute1ノードでKVMを実行するが、Virtual machine hostはインストールしないこと。
- インストール時の設定パラメータ例

設定項目	設定例
初期起動時のLanguage	English
起動	Install Ubuntu Server
Language	English - English
Country location	other→Asia→Japan
Country locale	United States - en_US.UTF-8
Detect keyboard layout	No
Keyboard	Japanese→Japanese
Primary network interface	eth1: Ethernet
Hostname	それぞれのノード名
ユーザ名	フルネームで入力
アカウント名	ユーザ名のファーストネームで設定される
パスワード	任意のパスワード
Weak password(出ない場合も)	<Yes>を選択
ホームの暗号化	任意
タイムゾーン	Asia/Tokyoであることを確認
パーティション設定	Guided - use entire disk and set up LVM
パーティション選択	sdaを選択
パーティション書き込み	<Yes>を選択
パーティションサイズ	デフォルトのまま
変更の書き込み	<Yes>を選択
HTTP proxy	環境に合わせて任意
アップグレード	任意
ソフトウェア	OpenSSH serverのみ選択
GRUB	<Yes>を選択
インストール完了	<Continue>を選択

1-6 Ubuntu へのログインと root 権限

インストールしたUbuntuはセキュリティの設定が行われているため、root権限が必要となる作業は以下のように行ってください。

- rootで直接ログインできないので、インストール時に作成したアカウントでログインする。
- root権限が必要な場合には、sudoコマンドを使用する。
- rootで連続して作業したい場合には、sudo -sコマンドでシェルを起動する。

1-7 設定ファイル等の記述について

- 設定ファイルは特別な記述が無い限り、必要な設定を抜粋したものです。
- 特に変更が必要ない設定項目は省略されています。
- [見出し]が付いている場合、その見出しから次の見出しまでの間に設定を記述します。
- コメントアウトされていない設定項目が存在する場合には、値を変更してください。多くの設定項目は記述が存在しているため、エディタの検索機能で検索することをお勧めします。
- 既存の設定を記述に含む場合には、値の変更が不要であることを示すために斜体に行っている場合があります。
- 一部改行してしまっているものがありますが、PDFからのコピー&ペースト時に改行が入ってしまう場合があります。そのような設定がある場合には、注記があります。
- **赤字**は既存の設定を削除、またはコメントアウトなどで無効にしたい設定です。
- **青字**は新たに追加する設定、または変更、注目して欲しい点などを指定しています。

設定ファイルの記述例

```
[DEFAULT] ←この見出しから次の見出しまでの間に以下を記述
#sql_connection = sqlite:///var/lib/glance/glance.sqlite ←既存設定
sql_connection = mysql://glance:password@controller/glance
↑ 既存設定を削除、またはコメントアウトして記述

[keystone_authtoken] ←見出し
#auth_host = 127.0.0.1 ←既存設定
auth_host = controller ←変更
auth_port = 35357 ←既存のまま
auth_protocol = http ←既存のまま
auth_uri = http://controller:5000/v2.0 ←追加
admin_tenant_name = service ←変更
admin_user = glance ←変更
admin_password = password ←変更
```

1-8 作業上の注意点

手順の中では解説していない注意点を挙げておきます。

・エラーログに偶発的に記録されるAMQP serverへの接続エラー
手順では、設定などにエラーが無いか確認するために、設定後にログファイルを削除し、サービスを再起動、そしてログファイルを確認するという手順を踏んでいます。ログには通常WARNINGやERRORのログが記録されませんが、設定や仕様上やむを得ないWARNINGが記録される場合には、無視しても構わない旨記述しています。ただし、偶発的にAMQP server (RabbitMQ) への接続エラーがログに記録される場合があります。

エラーの例

```
2014-02-18 15:36:44.667 4099 ERROR cinder.openstack.common.rpc.com
mon [req-9b14549c-b17e-48d2-ba05-851055628167 None None] AMQP serv
er on controller:5672 is unreachable: Socket closed. Trying again
in 30 seconds.
```

このエラーは再現性が低いため、手順内では記述していません。このエラーが記録されていた場合でも、自動的にAMQP serverへの再接続を行うので、無視して構いません。もし気になる場合には、再度ログの削除、サービスの再起動を行い、エラーが発生しないことを確認してください。

2. controller ノードの設定

controllerノードに、以下の設定を行います。

- Ubuntu Server 12.04LTS(12.04.4)のインストール
- ネットワークの設定
- systemctlによるシステムの設定
- aptの設定
- NTPのインストール
- Python用MySQLクライアントのインストール
- MySQLサーバーのインストール
- RabbitMQのインストール
- Keystoneのインストール
- Glanceのインストール
- Novaのインストール
- Cinderのインストール
- Neutronのインストール
- Horizonのインストール

2-1 ネットワークの設定

IPアドレスと静的名前解決の設定を行います。

2-1-1 IP アドレスの設定

/etc/network/interfacesを編集し、固定のIPアドレスを設定します。

/etc/network/interfaces

```
auto eth0
iface eth0 inet static
    address 192.168.0.10
    netmask 255.255.255.0

auto eth1
iface eth1 inet static
    address 10.0.0.10
    netmask 255.255.255.0
    gateway 10.0.0.1
    dns-nameservers 10.0.0.1
```

networkingサービスを再起動します。serviceコマンドでは正しく設定が反映されない
ので、直接/etc/init.d/networkingスクリプトを実行します。

```
# /etc/init.d/networking restart
```

2-1-2 静的名前解決の設定

/etc/hostsに各ノードのIPアドレスとホスト名を記述します。127.0.1.1の行はコメントアウトします。

/etc/hosts

```
127.0.0.1    localhost
#127.0.1.1  controller
192.168.0.10 controller
192.168.0.9  network
192.168.0.11 compute1
```

2-2 sysctl によるシステムの設定

Linuxのネットワークパケット処理について設定を行います。

/etc/sysctl.confを編集します。

/etc/sysctl.conf

```
net.ipv4.conf.default.rp_filter=0
net.ipv4.conf.all.rp_filter=0
```

sysctlコマンドで設定を適用します。

```
# sysctl -p
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
```

2-3 apt の設定

aptの設定を行い、OpenStack関係のパッケージをThe Ubuntu Cloud Archiveから取得するようにします。

```
# apt-get install -y python-software-properties
# add-apt-repository cloud-archive:havana
Press [ENTER] to continue or ctrl-c to cancel adding it
※Enter キーを押す
```

パッケージの更新を行い、再起動を行います。

```
# apt-get update && apt-get dist-upgrade
# reboot
```

2-4 NTP のインストール

時刻合わせを正確にするためにNTPをインストールします。

```
# apt-get install -y ntp
```

2-5 Python 用 MySQL クライアントのインストール

Python用のMySQLクライアントをインストールします。

```
# apt-get install -y python-mysqldb
```

2-6 MySQL サーバーのインストール

MySQLサーバーをインストールします。

2-6-1 パッケージのインストール

インストール中にMySQLのrootパスワードを設定する必要があります。パスワードはpasswordに設定します。

```
# apt-get install -y mysql-server
```

2-6-2 バインドアドレスの変更

別のノードからMySQLへアクセスできるよう、バインドアドレスを変更します。

/etc/mysql/my.cnf

```
[mysqld]
#bind-address = 127.0.0.1
bind-address = 192.168.0.10
```

2-6-3 サービスの再起動

変更後、サービスを再起動します。

```
# service mysql restart
```

2-6-4 データベースとユーザーの作成

OpenStackの各コンポーネントが使用するデータベースとユーザーを作成します。別ノードから接続されるデータベースには、そのノード用のユーザーを作成する必要があります。

```
# mysql -u root -p <<EOF
CREATE DATABASE nova;
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'controller' \
IDENTIFIED BY 'password';

CREATE DATABASE cinder;
GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'controller' \
IDENTIFIED BY 'password';

CREATE DATABASE glance;
GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'controller' \
IDENTIFIED BY 'password';

CREATE DATABASE keystone;
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'controller' \
IDENTIFIED BY 'password';

CREATE DATABASE neutron;
GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'controller' \
IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'network' \
IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'compute1' \
IDENTIFIED BY 'password';

FLUSH PRIVILEGES;
EOF
Enter password: ※MySQL の root パスワード password を入力
```

2-7 RabbitMQ のインストール

RabbitMQをインストールします。

2-7-1 パッケージのインストール

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

2-7-2 guest ユーザーのパスワード変更

```
# rabbitmqctl change_password guest password
Changing password for user "guest" ...
...done.
```

2-8 Keystone のインストール

Keystoneをインストールします。

2-8-1 パッケージのインストール

```
# apt-get install -y keystone
```

2-8-2 設定の変更

/etc/keystone/keystone.conf

```
[DEFAULT]
# admin_token = ADMIN
admin_token = password

# The base endpoint URLs for keystone that are advertised to clients
# (NOTE: this does NOT affect how keystone listens for connections)
# public_endpoint = http://localhost:%(public_port)s/
# admin_endpoint = http://localhost:%(admin_port)s/
public_endpoint = http://controller:%(public_port)s/
admin_endpoint = http://controller:%(admin_port)s/

[sql]
#connection = sqlite:///var/lib/keystone/keystone.db
connection = mysql://keystone:password@controller/keystone
```

2-8-3 使用しないデータベースファイルの削除

```
# rm /var/lib/keystone/keystone.db
```

2-8-4 データベースに表を作成

```
# keystone-manage db_sync
```

2-8-5 サービスの再起動

```
# rm /var/log/keystone/*
# service keystone restart
```

2-8-6 ログの確認

```
# cat /var/log/keystone/keystone.log | grep "ERROR\|WARNING"
※特に何も表示されなければ OK
```

2-8-7 各種認証情報設定用スクリプトの作成と実行

keystone_init.sh

```
#!/bin/bash

# Modify these variables as needed
export OS_SERVICE_TOKEN="password"
export OS_SERVICE_ENDPOINT="http://controller:35357/v2.0"

ADMIN_USER=${ADMIN_USER:-admin}
NOVA_USER=${NOVA_USER:-nova}
GLANCE_USER=${GLANCE_USER:-glance}
NEUTRON_USER=${NEUTRON_USER:-neutron}
CINDER_USER=${CINDER_USER:-cinder}
DEMO_USER=${DEMO_USER:-demo}

ADMIN_PASSWORD=${ADMIN_PASSWORD:-password}
SERVICE_PASSWORD=${SERVICE_PASSWORD:-$ADMIN_PASSWORD}
DEMO_PASSWORD=${DEMO_PASSWORD:-$ADMIN_PASSWORD}

ADMIN_EMAIL=${ADMIN_EMAIL:-admin@example.com}
NOVA_EMAIL=${NOVA_EMAIL:-nova@example.com}
GLANCE_EMAIL=${GLANCE_EMAIL:-glance@example.com}
NEUTRON_EMAIL=${NEUTRON_EMAIL:-neutron@example.com}
CINDER_EMAIL=${CINDER_EMAIL:-cinder@example.com}
DEMO_EMAIL=${DEMO_EMAIL:-demo@example.com}

ADMIN_TENANT=${ADMIN_TENANT:-admin}
SERVICE_TENANT=${SERVICE_TENANT:-service}
DEMO_TENANT=${DEMO_TENANT:-demo}

ADMIN_ROLE=${ADMIN_ROLE:-admin}
MEMBER_ROLE=${MEMBER_ROLE:-Member}

#Keystone
KEYSTONE_REGION=RegionOne
KEYSTONE_HOST=controller

# Create Tenants
keystone tenant-create --name $ADMIN_TENANT
keystone tenant-create --name $SERVICE_TENANT
keystone tenant-create --name $DEMO_TENANT

# Create Admin User
keystone user-create --name $ADMIN_USER --pass "$ADMIN_PASSWORD" \
--email $ADMIN_EMAIL
keystone role-create --name $ADMIN_ROLE
keystone user-role-add --user $ADMIN_USER --role $ADMIN_ROLE \
--tenant $ADMIN_TENANT

#Create Service Users
keystone user-create --name $NOVA_USER \
```

```

--pass "$SERVICE_PASSWORD" --tenant $SERVICE_TENANT \
--email $NOVA_EMAIL
keystone user-create --name $GLANCE_USER \
--pass "$SERVICE_PASSWORD" --tenant $SERVICE_TENANT \
--email $CINDER_EMAIL
keystone user-create --name $NEUTRON_USER \
--pass "$SERVICE_PASSWORD" --tenant $SERVICE_TENANT \
--email $NEUTRON_EMAIL
keystone user-create --name $CINDER_USER \
--pass "$SERVICE_PASSWORD" --tenant $SERVICE_TENANT \
--email $CINDER_EMAIL

# Add Admin Roles to Service Users in Service Tenants
keystone user-role-add --tenant $SERVICE_TENANT \
--user $NOVA_USER --role $ADMIN_ROLE
keystone user-role-add --tenant $SERVICE_TENANT \
--user $GLANCE_USER --role $ADMIN_ROLE
keystone user-role-add --tenant $SERVICE_TENANT \
--user $NEUTRON_USER --role $ADMIN_ROLE
keystone user-role-add --tenant $SERVICE_TENANT \
--user $CINDER_USER --role $ADMIN_ROLE

#Create Demo User
keystone user-create --name $DEMO_USER \
--pass "$DEMO_PASSWORD" --email $DEMO_EMAIL --tenant $DEMO_TENANT

# Create Member Role for Demo
keystone role-create --name $MEMBER_ROLE

# Add Member Role to Demo User in Demo Tenant
keystone user-role-add --user $DEMO_USER --role $MEMBER_ROLE \
--tenant $DEMO_TENANT

# Create Services
keystone service-create --name keystone --type identity \
--description 'OpenStack Identity'
keystone service-create --name glance --type image \
--description 'OpenStack Image Service'
keystone service-create --name nova --type compute \
--description 'OpenStack Compute Service'
keystone service-create --name ec2 --type ec2 \
--description 'OpenStack EC2 service'
keystone service-create --name cinder --type volume \
--description 'OpenStack Volume Service'
keystone service-create --name neutron --type network \
--description 'OpenStack Networking service'

# Create Endpoints
keystone endpoint-create --region $KEYSTONE_REGION \
--service keystone \
--publicurl 'http://'"$KEYSTONE_HOST"'':5000/v2.0' \
--adminurl 'http://'"$KEYSTONE_HOST"'':35357/v2.0' \
--internalurl 'http://'"$KEYSTONE_HOST"'':5000/v2.0'
keystone endpoint-create --region $KEYSTONE_REGION \
--service glance --publicurl 'http://'"$KEYSTONE_HOST"'':9292' \
--adminurl 'http://'"$KEYSTONE_HOST"'':9292' \
--internalurl 'http://'"$KEYSTONE_HOST"'':9292'
keystone endpoint-create --region $KEYSTONE_REGION \
--service nova \

```

```

--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 $KEYSTONE_REGION --service ec2 \
--publicurl 'http://"$KEYSTONE_HOST":8773/services/Cloud' \
--adminurl 'http://"$KEYSTONE_HOST":8773/services/Admin' \
--internalurl 'http://"$KEYSTONE_HOST":8773/services/Cloud'
keystone endpoint-create --region $KEYSTONE_REGION \
--service cinder \
--publicurl 'http://"$KEYSTONE_HOST":8776/v1/$(tenant_id)s' \
--adminurl 'http://"$KEYSTONE_HOST":8776/v1/$(tenant_id)s' \
--internalurl 'http://"$KEYSTONE_HOST":8776/v1/$(tenant_id)s'
keystone endpoint-create --region $KEYSTONE_REGION \
--service neutron --publicurl 'http://"$KEYSTONE_HOST":9696/' \
--adminurl 'http://"$KEYSTONE_HOST":9696/' \
--internalurl 'http://"$KEYSTONE_HOST":9696/'

```

実行権限を付与し、実行します。

```

# chmod +x keystone_init.sh
# ./keystone_init.sh

```

Property	Value
adminurl	http://controller:35357/v2.0
id	55a96ec8819947308e5b573efb209c53
internalurl	http://controller:5000/v2.0
publicurl	http://controller:5000/v2.0
region	RegionOne
service_id	7ade46b5a72945f7a36369387d310544

Property	Value
adminurl	http://controller:9292
id	a9f306e8e5434286844b02f9b651d4cf
internalurl	http://controller:9292
publicurl	http://controller:9292
region	RegionOne
service_id	8efbae2a457e411eb49231f2cda4c4bb

Property	Value
adminurl	http://controller:8774/v2/\$(tenant_id)s
id	1f831a97970e4681a06b89b774db13c0
internalurl	http://controller:8774/v2/\$(tenant_id)s
publicurl	http://controller:8774/v2/\$(tenant_id)s
region	RegionOne
service_id	15970c0b312a4eca84970854b7d7e90a

Property	Value
adminurl	http://controller:8773/services/Admin
id	270399fe253648e98d0c32f6c8ff8bbb
internalurl	http://controller:8773/services/Cloud
publicurl	http://controller:8773/services/Cloud
region	RegionOne
service_id	aca4333f1f694954b7b4c6d3d2ff82f2

Property	Value
adminurl	http://controller:8776/v1/\$(tenant_id)s
id	d5076e6f8e5e4e23b45d280f4e5e9a83

internalurl	http://controller:8776/v1/\$(tenant_id)s
publicurl	http://controller:8776/v1/\$(tenant_id)s
region	RegionOne
service_id	19837a93ba8d4ba59ac37fb9368c19f6

Property	Value
adminurl	http://controller:9696/
id	5a31276e303f46c5807e4be28b372d3e
internalurl	http://controller:9696/
publicurl	http://controller:9696/
region	RegionOne
service_id	8a19a6bc1808487393ff11d3c1b0c7f9

2-8-8 環境変数設定ファイルの作成と読み込み

環境変数を設定するファイルopenrcを作成します。

~/openrc

```
export OS_USERNAME=admin
export OS_PASSWORD=password
export OS_TENANT_NAME=admin
export OS_AUTH_URL="http://controller:5000/v2.0/"
```

sourceコマンドで読み込み、環境変数を設定します。

```
# source openrc
```

2-8-9 Keystone の動作確認

認証情報を表示します。

```
# keystone token-get
```

ユーザーリストを表示します。

```
# keystone user-list
```

id	name	enabled	email
aa3a80543a8641d39b22d9eb07181f92	admin	True	admin@example.com
896b4606c00c4ee4958e5e1a7ceae7a	cinder	True	cinder@example.com
bc41130da8894b469839edf654ba32ec	demo	True	demo@example.com
316ef9ee7ddf4c9d98419917f93960ac	glance	True	glance@example.com
98ab978a43e84161b7d5159ec564ed33	neutron	True	neutron@example.com
3eff607998ce43db8ab924eb074fe35d	nova	True	nova@example.com

2-9 Glance のインストール

Glanceをインストールします。

2-9-1 パッケージのインストール

Glanceが内部でsheepdogを要求しているので同時にインストールします。

```
# apt-get install -y glance python-glanceclient sheepdog
```

2-9-2 JSON ファイルの取得

パッケージミスで含まれていないスキーマ定義のためのJSONファイルを取得します。

```
# cd /etc/glance
# wget https://raw.githubusercontent.com/openstack/glance/master/etc/schema-image.json
# chown glance:glance schema-image.json
# cd
```

※wgetコマンドは1行です。

2-9-3 使用しないデータベースファイルの削除

```
# rm /var/lib/glance/glance.sqlite
```

2-9-4 設定の変更

/etc/glance/glance-api.conf

```
[DEFAULT]
#sql_connection = sqlite:///var/lib/glance/glance.sqlite
sql_connection = mysql://glance:password@controller/glance

[keystone_authtoken]
#auth_host = 127.0.0.1
auth_host = controller
auth_port = 35357
auth_protocol = http
#admin_tenant_name = %SERVICE_TENANT_NAME%
#admin_user = %SERVICE_USER%
#admin_password = %SERVICE_PASSWORD%
auth_uri = http://controller:5000/v2.0
admin_tenant_name = service
admin_user = glance
admin_password = password

#flavor=
flavor=keystone
```

/etc/glance/glance-registry.conf

```
[DEFAULT]
```

```
#sql_connection = sqlite:///var/lib/glance/glance.sqlite
sql_connection = mysql://glance:password@controller/glance

[keystone_authtoken]
#auth_host = 127.0.0.1
auth_host = controller
auth_port = 35357
auth_protocol = http
#admin_tenant_name = %SERVICE_TENANT_NAME%
#admin_user = %SERVICE_USER%
#admin_password = %SERVICE_PASSWORD%
auth_uri = http://controller:5000/v2.0
admin_tenant_name = service
admin_user = glance
admin_password = password

#flavor=
flavor=keystone
```

2-9-5 データベースに表を作成

```
# glance-manage db_sync
```

2-9-6 サービスの再起動

```
# rm /var/log/glance/*
# service glance-registry restart && service glance-api restart
```

2-9-7 ログの確認

以下のWARNINGは無視しても構いません。

```
# cat /var/log/glance/api.log | grep "ERROR\|WARNING"
2014-01-29 13:30:02.252 6268 WARNING glance.store.base [-] Failed
to configure store correctly: Store cinder could not be configured
correctly. Reason: Cinder storage requires a context. Disabling a
dd method.
```

2-9-8 CirrOS イメージの取得と登録

CirrOSがインストールされたイメージをダウンロード取得します。

```
# wget http://download.cirros-cloud.net/0.3.1/cirros-0.3.1-x86_64-
disk.img
```

※wgetコマンドは1行です。

Glanceに登録します。

```
# 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
```

Property	Value
checksum	d972013792949d0d3ba628fbe8685bce
container_format	bare
created_at	2014-02-17T12:32:47
deleted	False
deleted_at	None
disk_format	qcow2
id	17bc08b9-fafd-402d-bf31-776af864db10
is_public	True
min_disk	0
min_ram	0
name	Cirros 0.3.1
owner	c15b53c23c694575b31610a90812b7a6
protected	False
size	13147648
status	active
updated_at	2014-02-17T12:32:47

登録されたことを確認します。

```
# glance image-list
```

ID	Name	Disk Format	Container Format	Size	Status
17bc08b9-fafd-402d-bf31-776af864db10	Cirros 0.3.1	qcow2	bare	13147648	active

2-9-9 Ubuntu Server イメージの取得と登録（任意）

Ubuntu Serverをインストールする場合には、以下の通りダウンロードします。Ubuntu Serverの実行には、コンピュータノードにメモリが多く割り当てられている必要があります。また、起動時のフレーバーの指定で「m1.small」以上を指定する必要があります。

```
# wget http://uec-images.ubuntu.com/precise/current/precise-server-cloudimg-amd64-disk1.img
```

※wgetコマンドは1行です。

Glanceに登録します。

```
glance image-create --name="Ubuntu Server 12.04" \
--disk-format=qcow2 --container-format=bare --is-public=true \
< precise-server-cloudimg-amd64-disk1.img
```

Property	Value
checksum	e13d526cceddf80f41dfd23e7e6cf2e0

container_format	bare
created_at	2014-02-26T11:45:58
deleted	False
deleted_at	None
disk_format	qcow2
id	5e10d9a8-c737-43a2-a24d-61d770c0ec3f
is_public	True
min_disk	0
min_ram	0
name	Ubuntu Server 12.04
owner	c15b53c23c694575b31610a90812b7a6
protected	False
size	255393792
status	active
updated_at	2014-02-26T11:46:05

2-10 Nova のインストール

Novaをインストールします。

2-10-1 パッケージのインストール

```
# apt-get install -y nova-novncproxy novnc nova-api \
nova-ajax-console-proxy nova-cert nova-conductor \
nova-consoleauth nova-doc nova-scheduler python-novaclient
```

2-10-2 設定の変更

/etc/nova/nova.conf

```
[DEFAULT]
※既にある設定はそのままにして、以下を追加
rpc_backend = nova.rpc.impl_kombu
rabbit_host = controller
rabbit_password = password

auth_strategy=keystone

api_paste_config=/etc/nova/api-paste.ini

my_ip=controller
vncserver_listen=controller
vncserver_proxyclient_address=controller

# Networking
network_api_class=nova.network.neutronv2.api.API
neutron_url=http://controller:9696
neutron_auth_strategy=keystone
neutron_admin_tenant_name=service
neutron_admin_username=neutron
neutron_admin_password=password
neutron_admin_auth_url=http://controller:35357/v2.0
linuxnet_interface_driver=nova.network.linux_net.LinuxOVSInterface
```

Driver

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

# Metadata
neutron_metadata_proxy_shared_secret=password
service_neutron_metadata_proxy=true
metadata_listen = controller
metadata_listen_port = 8775

[database]
sql_connection=mysql://nova:password@controller/nova

[keystone_authtoken]
auth_host = controller
auth_port = 35357
auth_protocol = http
auth_uri = http://controller:5000/v2.0
admin_tenant_name = service
admin_user = nova
admin_password = password
```

※linuxnet_interface_driver の設定は1行です。

/etc/nova/api-paste.ini

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter
_factory
#auth_host = 127.0.0.1
auth_host = controller
auth_port = 35357
auth_protocol = http
#admin_tenant_name = %SERVICE_TENANT_NAME%
#admin_user = %SERVICE_USER%
#admin_password = %SERVICE_PASSWORD%
auth_uri = http://controller:5000/v2.0
admin_tenant_name = service
admin_user = nova
admin_password = password
```

2-10-3 使用しないデータベースファイルの削除

```
# rm /var/lib/nova/nova.sqlite
```

2-10-4 データベースに表を作成

```
# nova-manage db sync
2014-02-17 21:49:48.069 11523 INFO migrate.versioning.api [-] 132
-> 133...
2014-02-17 21:49:48.609 11523 INFO migrate.versioning.api [-] done
(略)
2014-02-17 21:49:51.480 11523 INFO migrate.versioning.api [-] 215
-> 216...
```

```
2014-02-17 21:49:51.494 11523 INFO migrate.versioning.api [-] done
```

2-10-5 サービスの再起動

```
# rm /var/log/nova/*  
# service nova-api restart && service nova-cert restart && \  
service nova-conductor restart && \  
service nova-consoleauth restart && \  
service nova-scheduler restart && service nova-novncproxy restart
```

2-10-6 ログの確認

```
# cat /var/log/nova/nova-api.log | grep "ERROR\|WARNING"  
※特に何も表示されなければ OK
```

2-11 Cinder のインストール

Cinderをインストールします。

2-11-1 パッケージのインストール

```
# apt-get install -y cinder-api cinder-scheduler cinder-volume
```

2-11-2 設定の変更

/etc/cinder/cinder.conf

```
[DEFAULT]  
※既にある設定はそのままにして、以下を追加  
rpc_backend = cinder.openstack.common.rpc.impl_kombu  
rabbit_host = controller  
rabbit_port = 5672  
rabbit_userid = guest  
rabbit_password = password  
  
[database]  
connection = mysql://cinder:password@controller/cinder
```

/etc/cinder/api-paste.ini

```
[filter:authtoken]  
paste.filter_factory = keystoneclient.middleware.auth_token:filter  
_factory  
#auth_host = 127.0.0.1  
auth_host = controller  
auth_port = 35357  
auth_protocol = http  
#admin_tenant_name = %SERVICE_TENANT_NAME%  
#admin_user = %SERVICE_USER%  
#admin_password = %SERVICE_PASSWORD%
```

```
auth_uri = http://controller:5000/v2.0
admin_tenant_name = service
admin_user = cinder
admin_password = password
```

2-1 1-3 使用しないデータベースファイルの削除

```
# rm /var/lib/cinder/cinder.sqlite
```

2-1 1-4 データベースに表を作成

```
# cinder-manage db sync
2014-02-17 21:55:58.039 13721 INFO migrate.versioning.api [-] 0 -> 1...
2014-02-17 21:55:58.119 13721 INFO migrate.versioning.api [-] done
(略)
2014-02-17 21:55:58.354 13721 INFO migrate.versioning.api [-] 19 -> 20...
2014-02-17 21:55:58.364 13721 INFO migrate.versioning.api [-] done
```

2-1 1-5 サービスの再起動

```
# rm /var/log/cinder/*
# service cinder-scheduler restart && service cinder-api restart
```

2-1 1-6 ログの確認

cinder-api.logは以下のWARNINGが大量に出ますが、無視して構いません。

```
# cat /var/log/cinder/cinder-scheduler.log | grep "ERROR\|WARNING"
※特に何も表示されなければOK

# cat /var/log/cinder/cinder-api.log | grep "ERROR\|WARNING"
2014-01-29 13:44:04.563 13212 WARNING cinder.keymgr.conf_key_mgr
[-] This key manager is insecure and is not recommended for produc
tion deployments
2014-01-29 13:44:04.563 13212 WARNING cinder.keymgr.conf_key_mgr
[-] config option keymgr.fixed_key has not been defined: some oper
ations may fail unexpectedly
```

2-1 1-7 イメージ格納用ボリュームの作成

```
# pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created
# vgcreate cinder-volumes /dev/sdb
Volume group "cinder-volumes" successfully created
```

2-1 1-8 サービスの再起動

```
# service cinder-volume restart && service tgt restart
```

2-1 1-9 ログの確認

```
# cat /var/log/cinder/cinder-volume.log | grep "ERROR\|WARNING"
※特に何も表示されなければ OK

# cat /var/log/syslog | grep "ERROR\|WARNING"
※特に何も表示されなければ OK
```

2-1 2 Neutron のインストール

Neutronをインストールします。

2-1 2-1 パッケージのインストール

```
# apt-get install -y neutron-server
```

2-1 2-2 設定の変更

/etc/neutron/neutron.conf

```
[DEFAULT]
# IP address of the RabbitMQ installation
# rabbit_host = localhost
rabbit_host = controller
# Password of the RabbitMQ server
# rabbit_password = guest
rabbit_password = password

[keystone_authtoken]
#auth_host = 127.0.0.1
auth_host = controller
auth_port = 35357
auth_protocol = http
#admin_tenant_name = %SERVICE_TENANT_NAME%
#admin_user = %SERVICE_USER%
#admin_password = %SERVICE_PASSWORD%
auth_uri = http://controller:5000/v2.0
admin_tenant_name = service
admin_user = neutron
admin_password = password
signing_dir = $state_path/keystone-signing

[database]
#connection = sqlite:///var/lib/neutron/neutron.sqlite
connection = mysql://neutron:password@controller/neutron
```

/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini

```
※既にある設定はそのままにして、以下を追加

[ovs]
tenant_network_type = gre
tunnel_id_ranges = 1:1000
```

```
enable_tunneling = True

[securitygroup]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridI
ptablesFirewallDriver
```

※firewall_driverの設定は1行です。

2-1 2-3 使用しないデータベースファイルの削除

```
# rm /var/lib/neutron/neutron.sqlite
```

2-1 2-4 サービスの再起動

```
# rm /var/log/neutron/*
# service neutron-server restart
```

2-1 2-5 ログの確認

以下のエラーが出ますが、新規にNeutronを構築した場合には無視して構いません。また、いくつかのWARNINGが出ますが、無視して構いません。

```
# cat /var/log/neutron/server.log | grep "ERROR\|WARNING"
2014-02-17 22:07:37.465 14384 ERROR neutron.common.legacy [-] Skip
ping unknown group key: firewall_driver
2014-02-17 22:07:37.802 14384 WARNING neutron.api.extensions [-] E
xtension vpnaas not supported by any of loaded plugins
(略)
```

2-1 3 Horizon のインストール

Horizonをインストールします。

2-1 3-1 パッケージのインストール

```
# apt-get install -y openstack-dashboard memcached python-memcache
```

2-1 3-2 設定の変更

エラーが発生するとInternal errorが表示されるだけで分かりにくいので、エラー画面が表示されるように設定します。また、ログアウト時のURLを追加で設定します。

/etc/openstack-dashboard/local_settings.py

```
#DEBUG = False
DEBUG = True

# Default Ubuntu apache configuration uses /horizon as the applica
tion root.
```

```
# Configure auth redirects here accordingly.
LOGIN_URL='/horizon/auth/login/'
LOGIN_REDIRECT_URL='/horizon'
LOGOUT_URL='/horizon/auth/logout/'
```

2-1 3-3 サービスの再起動

```
# service apache2 restart
```

2-1 3-4 Web 管理画面へのアクセス

Webブラウザから、以下のアドレスにアクセスします。

```
http://192.168.0.10/horizon
```

以下のユーザー／パスワードが登録されているので、それぞれログインできるか確認します。

	管理者	一般ユーザー
ユーザー名	admin	demo
パスワード	password	password

3. network ノードの設定

networkノードに、以下の設定を行います。

- Ubuntu Server 12.04LTS(12.04.4)のインストール
- ネットワークの設定
- systemctlによるシステムの設定
- aptの設定
- NTPのインストール
- Python用MySQLクライアントのインストール
- Neutronのインストール
- 仮想ネットワークの設定
- ネットワークの確認

3-1 ネットワークの設定

IPアドレスと静的名前解決の設定を行います。

3-1-1 IP アドレスの設定

/etc/network/interfacesを編集し、固定のIPアドレスを設定します。

/etc/network/interfaces

```
auto eth0
iface eth0 inet static
    address 192.168.0.9
    netmask 255.255.255.0

auto eth1
iface eth1 inet static
    address 10.0.0.9
    netmask 255.255.255.0
    gateway 10.0.0.1
    dns-nameservers 10.0.0.1
```

networkingサービスを再起動します。serviceコマンドでは正しく設定が反映されない
ので、直接/etc/init.d/networkingスクリプトを実行します。

```
# /etc/init.d/networking restart
```

3-1-2 静的名前解決の設定

/etc/hostsに各ノードのIPアドレスとホスト名を記述します。127.0.1.1の行はコメントア
ウトします。

/etc/hosts

```
127.0.0.1    localhost
#127.0.1.1  network
192.168.0.10 controller
192.168.0.9  network
192.168.0.11 compute1
```

3-2 sysctl によるシステムの設定

Linuxのネットワークパケット処理について設定を行います。

/etc/sysctl.confを編集します。

/etc/sysctl.conf

```
net.ipv4.conf.default.rp_filter=0
net.ipv4.conf.all.rp_filter=0
net.ipv4.ip_forward=1
```

sysctlコマンドで設定を適用します。

```
# sysctl -p
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
net.ipv4.ip_forward = 1
```

3-3 apt の設定

aptの設定を行い、OpenStack関係のパッケージをThe Ubuntu Cloud Archiveから取得するようにします。

```
# apt-get install -y python-software-properties
# add-apt-repository cloud-archive:havana
Press [ENTER] to continue or ctrl-c to cancel adding it
※Enter キーを押す
```

パッケージの更新を行い、再起動を行います。

```
# apt-get update && apt-get dist-upgrade
# reboot
```

3-4 NTP のインストール

時刻合わせを正確にするためにNTPをインストールします。

```
# apt-get install -y ntp
```

3-5 Python 用 MySQL クライアントのインストール

Python用のMySQLクライアントをインストールします。

```
# apt-get install -y python-mysqldb
```

3-6 Neutron のインストール

Neutronをインストールします。

3-6-1 パッケージのインストール

openvswitch-datapath-dkmsを必ずインストールします。

```
# apt-get install -y neutron-plugin-openvswitch-agent \  
neutron-dhcp-agent neutron-l3-agent openvswitch-datapath-dkms
```

3-6-2 設定の変更

/etc/neutron/neutron.conf

```
[DEFAULT]  
# IP address of the RabbitMQ installation  
# rabbit_host = localhost  
rabbit_host = controller  
# Password of the RabbitMQ server  
# rabbit_password = guest  
rabbit_password = password  
  
[keystone_authtoken]  
#auth_host = 127.0.0.1  
auth_host = controller  
auth_port = 35357  
auth_protocol = http  
#admin_tenant_name = %SERVICE_TENANT_NAME%  
#admin_user = %SERVICE_USER%  
#admin_password = %SERVICE_PASSWORD%  
auth_uri = http://controller:5000/v2.0  
admin_tenant_name = service  
admin_user = neutron  
admin_password = password  
signing_dir = $state_path/keystone-signing  
  
[database]  
#connection = sqlite:///var/lib/neutron/neutron.sqlite  
connection = mysql://neutron:password@controller/neutron
```

/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini

```
※既にある設定はそのままにして、以下を追加  
[ovs]
```

```

tenant_network_type = gre
tunnel_id_ranges = 1:1000
enable_tunneling = True
integration_bridge = br-int
tunnel_bridge = br-tun
local_ip = 192.168.0.9

[securitygroup]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridI
ptablesFirewallDriver

```

※firewall_driverの設定は1行です。

/etc/neutron/dhcp_agent.ini

```

# Example of interface_driver option for OVS based plugins(OVS, Ryu, NEC,
NVP,
# BigSwitch/Floodlight)
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver

# The agent can use other DHCP drivers. Dnsmasq is the simplest and requi
res
# no additional setup of the DHCP server.
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq

# Allow overlapping IP (Must have kernel build with CONFIG_NET_NS=y and
# iproute2 package that supports namespaces).
use_namespaces = True

# The DHCP server can assist with providing metadata support on isolated
# networks. Setting this value to True will cause the DHCP server to appen
d
# specific host routes to the DHCP request. The metadata service will onl
y
# be activated when the subnet gateway_ip is None. The guest instance mus
t
# be configured to request host routes via DHCP (Option 121).
enable_isolated_metadata = True

# Allows for serving metadata requests coming from a dedicated metadata
# access network whose cidr is 169.254.169.254/16 (or larger prefix), and
# is connected to a Neutron router from which the VMs send metadata
# request. In this case DHCP Option 121 will not be injected in VMs, as
# they will be able to reach 169.254.169.254 through a router.
# This option requires enable_isolated_metadata = True
enable_metadata_network = True

```

※enable_isolated_metadataとenable_metadata_networkのデフォルト値はFalseになっているので、Trueに変更するのを忘れないこと。

/etc/neutron/l3_agent.ini

```

# Example of interface_driver option for OVS based plugins (OVS, Ryu, NEC)
# that supports L3 agent
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver

# Allow overlapping IP (Must have kernel build with CONFIG_NET_NS=y and
# iproute2 package that supports namespaces).
use_namespaces = True

```

/etc/neutron/metadata_agent.ini

```
[DEFAULT]
#auth_url = http://localhost:5000/v2.0
auth_url = http://controller:5000/v2.0
auth_region = RegionOne
#admin_tenant_name = %SERVICE_TENANT_NAME%
#admin_user = %SERVICE_USER%
#admin_password = %SERVICE_PASSWORD%
admin_tenant_name = service
admin_user = neutron
admin_password = password

# IP address used by Nova metadata server
# nova_metadata_ip = 127.0.0.1
nova_metadata_ip = controller

# When proxying metadata requests, Neutron signs the Instance-ID header with a
# shared secret to prevent spoofing. You may select any string for a secret,
# but it must match here and in the configuration used by the Nova Metadata
# Server. NOTE: Nova uses a different key: neutron_metadata_proxy_shared_secret
# metadata_proxy_shared_secret =
metadata_proxy_shared_secret = password
```

3-6-3 使用しないデータベースファイルの削除

```
# rm /var/lib/neutron/neutron.sqlite
```

3-6-4 サービスの再起動

```
# rm /var/log/neutron/*
# service neutron-dhcp-agent restart && \
service neutron-l3-agent restart && \
service neutron-metadata-agent restart && \
service openvswitch-switch restart
```

3-6-5 ログの確認

以下のエラーが出ますが、無視して構いません。

```
# grep ERROR /var/log/neutron/*
/var/log/neutron/dhcp-agent.log:2014-02-02 12:45:20.885 1211 ERROR
neutron.common.legacy [-] Skipping unknown group key: firewall_dr
iver
/var/log/neutron/l3-agent.log:2014-02-02 12:45:21.140 1216 ERROR n
eutron.common.legacy [-] Skipping unknown group key: firewall_driv
er
/var/log/neutron/l3-agent.log: 2014-02-02 12:45:24.130 1216 ERROR
neutron.agent.l3_agent [-] The external network bridge 'br-ex' doe
s not exist
/var/log/neutron/server.log:2014-02-02 12:45:20.871 1223 ERROR neu
tron.common.legacy [-] Skipping unknown group key: firewall_driver
```

3-7 仮想ネットワークの設定

Open vSwitchを利用した仮想ネットワークの設定を行います。

3-7-1 ブリッジの作成

内部通信用と外部通信用の2つのブリッジを作成します。

内部通信用のブリッジの作成

```
# ovs-vsctl add-br br-int
```

外部通信用のブリッジの作成

```
# ovs-vsctl add-br br-ex
```

eth1を外部通信用ブリッジに接続

```
# ovs-vsctl add-port br-ex eth1
```

3-7-2 ネットワークインターフェースの設定変更

/etc/network/interfaces

```
auto eth0
iface eth0 inet static
    pre-up service openvswitch-switch start || true
    address 192.168.0.9
    netmask 255.255.255.0

auto br-ex
iface br-ex inet static
    pre-up service openvswitch-switch start || true
    address 10.0.0.9
    netmask 255.255.255.0
    gateway 10.0.0.1
    dns-nameservers 10.0.0.1

auto eth1
iface eth1 inet manual
    pre-up service openvswitch-switch start || true
    up ip address add 0/0 dev $IFACE
    up ip link set $IFACE up
    down ip link set $IFACE down
```

3-7-3 システム再起動

インターフェースの設定を適用するために、システムを再起動します。

```
# reboot
```

3-7-4 ネットワークインターフェースの確認

各種ブリッジが正常に設定されていることを確認します。

```
# ifconfig
br-ex      Link encap:Ethernet  HWaddr 00:1c:42:fc:0f:a8
           inet addr:10.0.0.9  Bcast:10.0.0.255  Mask:255.255.255.0
           inet6 addr: fe80::4ccf:a3ff:fe95:f85c/64 Scope:Link

br-int     Link encap:Ethernet  HWaddr aa:b7:7f:87:a3:41
           inet6 addr: fe80::5476:7cff:febc:2e46/64 Scope:Link

br-tun     Link encap:Ethernet  HWaddr a2:c9:1e:14:f4:40
           inet6 addr: fe80::455:7fff:feda:98e7/64 Scope:Link

eth0       Link encap:Ethernet  HWaddr 00:1c:42:05:75:a4
           inet addr:192.168.0.9  Bcast:192.168.0.255  Mask:255.255.255.0
           inet6 addr: fe80::21c:42ff:fe05:75a4/64 Scope:Link

eth1       Link encap:Ethernet  HWaddr 00:1c:42:fc:0f:a8
           inet6 addr: fe80::21c:42ff:fefc:fa8/64 Scope:Link
```

3-7-5 環境変数設定ファイルの作成と読み込み

環境変数を設定するファイルopenrcを作成します。

~/openrc

```
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=password
export OS_AUTH_URL="http://controller:5000/v2.0/"
export SERVICE_ENDPOINT="http://controller:35357/v2.0"
export SERVICE_TOKEN=password
```

sourceコマンドで読み込み、環境変数を設定します。

```
# source openrc
```

3-7-6 インスタンス用ネットワーク作成スクリプトの作成

インスタンスが起動した際に接続されるネットワークを作成するスクリプトを作成します。

~/create-virt-net.sh

```
#!/bin/bash
TENANT_NAME="demo"
TENANT_NETWORK_NAME="demo-net"
TENANT_SUBNET_NAME="${TENANT_NETWORK_NAME}-subnet"
TENANT_ROUTER_NAME="demo-router"
FIXED_RANGE="10.5.5.0/24"
```

```

NETWORK_GATEWAY="10.5.5.1"
TENANT_ID=$(keystone tenant-list | grep " $TENANT_NAME " \
| awk '{print $2}')

# Create Router
ROUTER_ID=$(neutron router-create --tenant_id $TENANT_ID \
$TENANT_ROUTER_NAME | grep " id " | awk '{print $4}')

# Create Network
TENANT_NET_ID=$(neutron net-create --tenant_id $TENANT_ID \
$TENANT_NETWORK_NAME --provider:network_type gre \
--provider:segmentation_id 1 | grep " id " | awk '{print $4}')

# Create Subnet
TENANT_SUBNET_ID=$(neutron subnet-create --tenant_id $TENANT_ID \
--ip_version 4 --name $TENANT_SUBNET_NAME $TENANT_NET_ID \
$FIXED_RANGE --gateway $NETWORK_GATEWAY \
--dns_nameservers list=true 8.8.8.8 | grep " id " \
| awk '{print $4}')

# Add Interface to Router
neutron router-interface-add $ROUTER_ID $TENANT_SUBNET_ID

```

実行権限を付与します。

```
# chmod +x create-virt-net.sh
```

スクリプトを実行します。あらかじめ~/openrcを読み込み、各種環境変数が設定されている必要があります。

```
# ./create-virt-net.sh
Added interface 465843eb-4b71-45f0-aa62-86db0d6b3f77 to router 209
47fdf-65ed-4171-aa99-c65b336a181a.
```

3-7-7 外部接続用ネットワークの作成

外部接続用のネットワークを作成します。

外部ネットワークext-netを作成します。

```
# neutron net-create ext-net --router:external=True
Created a new network:
```

Field	Value
admin_state_up	True
id	14fd4653-19ce-46ce-8ece-db450c0b3189
name	ext-net
provider:network_type	gre
provider:physical_network	
provider:segmentation_id	2
router:external	True
shared	False
status	ACTIVE

subnets	
tenant_id	c15b53c23c694575b31610a90812b7a6

外部ネットワーク用サブネットext-net-subnetを作成します。--allocation-poolで指定したIPアドレスの範囲がFloating IPとして利用されます。

```
# neutron subnet-create ext-net 10.0.0.0/24 --name ext-net-subnet \
--allocation-pool start=10.0.0.200,end=10.0.0.250 \
--gateway 10.0.0.1 --disable-dhcp
```

Field	Value
allocation_pools	{"start": "10.0.0.200", "end": "10.0.0.250"}
cidr	10.0.0.0/24
dns_nameservers	
enable_dhcp	False
gateway_ip	10.0.0.1
host_routes	
id	ebaec0bb-79f6-4082-acc6-2e3ce54e244c
ip_version	4
name	ext-net-subnet
network_id	14fd4653-19ce-46ce-8ece-db450c0b3189
tenant_id	c15b53c23c694575b31610a90812b7a6

インスタンス用ネットワークのルーターを外部接続用のネットワークに接続します。

```
# neutron router-gateway-set demo-router ext-net
Set gateway for router demo-router
```

3-8 ネットワークの確認

設定したネットワークをWeb管理画面で確認します。

3-8-1 管理者ユーザーとしての確認

Web管理画面に管理者ユーザーとして接続し、各種ネットワーク設定を確認します。

- ・ ネットワーク
システムパネル→ネットワークで、作成されているネットワークが確認できます。
- ・ ルーター
システムパネル→ルーターで、作成されているルーターが確認できます。

3-8-2 一般ユーザーとしての確認

Web管理画面に一般ユーザーとして接続し、各種ネットワーク設定を確認します。

- ネットワークトポロジ
ネットワークの管理→ネットワークトポロジで、グラフィカルにネットワーク接続の
状態が確認できます。
- ネットワーク
システムパネル→ネットワークで、作成されているネットワークが確認できます。
- ルーター
システムパネル→ルーターで、作成されているルーターが確認できます。

4. compute1 ノードの設定

compute1ノードに、以下の設定を行います。

- Ubuntu Server 12.04LTS(12.04.4)のインストール
- ネットワークの設定
- systemctlによるシステムの設定
- aptの設定
- NTPのインストール
- Python用MySQLクライアントのインストール
- Nova (nova-compute) のインストール
- Neutronのインストール
- 仮想ネットワークの設定
- ネットワークの確認

4-1 ネットワークの設定

IPアドレスと静的名前解決の設定を行います。

4-1-1 IP アドレスの設定

/etc/network/interfacesを編集し、固定のIPアドレスを設定します。

/etc/network/interfaces

```
auto eth0
iface eth0 inet static
    pre-up service openvswitch-switch start || true
    address 192.168.0.11
    netmask 255.255.255.0

auto eth1
iface eth1 inet static
    pre-up service openvswitch-switch start || true
    address 10.0.0.11
    netmask 255.255.255.0
    gateway 10.0.0.1
    dns-nameservers 10.0.0.1
```

networkingサービスを再起動します。serviceコマンドでは正しく設定が反映されない
ので、直接/etc/init.d/networkingスクリプトを実行します。

```
# /etc/init.d/networking restart
```

4-1-2 静的名前解決の設定

/etc/hostsに各ノードのIPアドレスとホスト名を記述します。127.0.1.1の行はコメントアウトします。

/etc/hosts

```
127.0.0.1    localhost
#127.0.1.1  computel
192.168.0.10 controller
192.168.0.9  network
192.168.0.11 computel
```

4-2 sysctl によるシステムの設定

Linuxのネットワークパケット処理について設定を行います。

/etc/sysctl.confを編集します。

/etc/sysctl.conf

```
net.ipv4.conf.default.rp_filter=0
net.ipv4.conf.all.rp_filter=0
```

sysctlコマンドで設定を適用します。

```
# sysctl -p
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
```

4-3 apt の設定

aptの設定を行い、OpenStack関係のパッケージをThe Ubuntu Cloud Archiveから取得するようにします。

```
# apt-get install -y python-software-properties
# add-apt-repository cloud-archive:havana
Press [ENTER] to continue or ctrl-c to cancel adding it
※Enter キーを押す
```

パッケージの更新を行い、再起動を行います。

```
# apt-get update && apt-get dist-upgrade
# reboot
```

4-4 NTP のインストール

時刻合わせを正確にするためにNTPをインストールします。

```
# apt-get install -y ntp
```

4-5 Python 用 MySQL クライアントのインストール

Python用のMySQLクライアントをインストールします。

```
# apt-get install -y python-mysqldb
```

4-6 Nova (nova-compute) のインストール

Nova(nova-compute)をインストールします。

4-6-1 パッケージのインストール

nova-compute-kvmをインストールすると、KVM環境も同時に構築されます。

```
# apt-get install -y nova-compute-kvm python-guestfs
```

"Create or update supermin appliance now?"に対して、Yesを選択します。

4-6-2 Linux カーネルのパーミッションの修正

Linuxカーネルのパーミッションが正しくないので、修正します。

```
# dpkg-statoverride --update \  
--add root root 0644 /boot/vmlinuz-$(uname -r)
```

今後、Linuxカーネルを更新した際に、自動的にパーミッションが修正されるようにスクリプトを作成しておきます。

/etc/kernel/postinst.d/statoverride

```
#!/bin/sh  
version="$1"  
# passing the kernel version is required  
[ -z "${version}" ] && exit 0  
dpkg-statoverride --update \  
--add root root 0644 /boot/vmlinuz-${version}
```

実行権限を付与します。

```
# chmod +x /etc/kernel/postinst.d/statoverride
```

4-6-3 設定の変更

/etc/nova/nova.conf

```
※既にある設定はそのままにして、以下を追加
rpc_backend = nova.rpc.impl_kombu
rabbit_host = controller
rabbit_password = password

auth_strategy=keystone

api_paste_config=/etc/nova/api-paste.ini

my_ip=compute1
vnc_enabled=True
vncserver_listen=0.0.0.0
vncserver_proxyclient_address=compute1
novncproxy_base_url=http://controller:6080/vnc_auto.html

glance_host=controller

# Networking
network_api_class=nova.network.neutronv2.api.API
neutron_url=http://controller:9696
neutron_auth_strategy=keystone
neutron_admin_tenant_name=service
neutron_admin_username=neutron
neutron_admin_password=password
neutron_admin_auth_url=http://controller:35357/v2.0
linuxnet_interface_driver=nova.network.linux_net.LinuxOVSIInterface
Driver

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

[database]
connection = mysql://nova:password@controller/nova
```

※linuxnet_interface_driverの設定は1行です。

/etc/nova/api-paste.ini

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter
_factory
#auth_host = 127.0.0.1
auth_host = controller
auth_port = 35357
auth_protocol = http
#admin_tenant_name = %SERVICE_TENANT_NAME%
#admin_user = %SERVICE_USER%
#admin_password = %SERVICE_PASSWORD%
auth_uri = http://controller:5000/v2.0
admin_tenant_name = service
admin_user = nova
```

```
admin_password = password
```

4-6-4 使用しないデータベースファイルの削除

```
# rm /var/lib/nova/nova.sqlite
```

4-6-5 サービスの再起動

```
# rm /var/log/nova/*  
# service nova-compute restart
```

4-6-6 ログの確認

```
# grep "ERROR\|WARNING" /var/log/nova/*  
※特に何も表示されなければ OK
```

4-7 Neutron のインストール

Neutronをインストールします。

4-7-1 パッケージのインストール

openvswitch-datapath-dkmsを必ずインストールします。

```
# apt-get install -y neutron-plugin-openvswitch-agent \  
openvswitch-datapath-dkms
```

4-7-2 ブリッジの作成

内部通信用のブリッジを作成します。

```
# ovs-vsctl add-br br-int
```

4-7-3 設定の変更

/etc/neutron/neutron.conf

```
[DEFAULT]  
# IP address of the RabbitMQ installation  
# rabbit_host = localhost  
rabbit_host = controller  
# Password of the RabbitMQ server  
# rabbit_password = guest  
rabbit_password = password  
  
[keystone_authtoken]  
#auth_host = 127.0.0.1  
auth_host = controller
```

```

auth_port = 35357
auth_protocol = http
#admin_tenant_name = %SERVICE_TENANT_NAME%
#admin_user = %SERVICE_USER%
#admin_password = %SERVICE_PASSWORD%
auth_uri = http://controller:5000/v2.0
admin_tenant_name = service
admin_user = neutron
admin_password = password
signing_dir = $state_path/keystone-signing

[database]
#connection = sqlite:///var/lib/neutron/neutron.sqlite
connection = mysql://neutron:password@controller/neutron

```

/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini

```

※既にある設定はそのままにして、以下を追加
[ovs]
tenant_network_type = gre
tunnel_id_ranges = 1:1000
enable_tunneling = True
integration_bridge = br-int
tunnel_bridge = br-tun
local_ip = 192.168.0.11

[securitygroup]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridI
ptablesFirewallDriver

```

※firewall_driverの設定は1行です。

4-7-4 サービスの再起動

```

# rm /var/log/neutron/*
# service neutron-plugin-openvswitch-agent restart

```

4-7-5 ログの確認

```

# grep "ERROR\|WARNING" /var/log/neutron/*
※特に何も表示されなければ OK

```

これで OpenStack 環境の構築作業は終了です。Web 管理画面にユーザーdemoで接続し、インスタンスの起動や Floating IP の割り当てなどを行い、外部クライアント (10.0.0.0/24 に属する) からインスタンスにネットワーク接続できることを確認してください。

たとえば、テスト用にインストールした CirrOS のイメージからインスタンスを起動し、Floating IP 10.0.0.201 を割り当てた場合、以下のコマンドで SSH 接続が行えます。

```

$ ssh cirros@10.0.0.201

```

```
cirros@10.0.0.201's password: cubswin:) ※実際には表示されない
```

Ubuntu Server のイメージをインストールしている場合には、コンピュータノードにメモリが多く割り当てられていることを確認してください。また、起動時のフレーバーの指定で「m1.small」以上を指定する必要があります。SSH 接続時のユーザー名は `ubuntu` を使用してください。パスワードは設定されていないので、キーペアを設定し、公開鍵認証で接続する必要があります。