

日本仮想化技術株式会社 概要

- 社名: 日本仮想化技術株式会社
 - 英語名: VirtualTech Japan Inc.
 - 略称: 日本仮想化技術/V TJ
- 設立: 2006年12月
- 資本金: 20,000,000円
- 本社: 東京都渋谷区渋谷1-1-10
- 取締役: 宮原 徹 (代表取締役社長兼CEO)
- 伊藤 宏通 (取締役CTO)
- スタッフ: 7名 (うち、6名が仮想化技術専門エンジニアです)
- URL: <http://VirtualTech.jp/>
- 仮想化技術に関する研究および開発
 - 仮想化技術に関する各種調査
 - 仮想化技術に関連したソフトウェアの開発
 - 仮想化技術を導入したシステムの構築

ベンダーニュートラルな
独立系仮想化技術
専業会社



仮想化環境構築をトータルサポート

戦略立案



設計



導入・移行



運用保守

- 戦略立案
 - コスト削減、社内標準化、将来プランのコンサルティング
- 設計
 - 要求仕様の策定
 - サーバ、ストレージからネットワークまでアプリケーションまで考慮した設計最適化
 - キャパシティプランニング(ベンチマーク)
- 導入
 - 仮想化ソリューションパッケージの提供
 - 仮想化統合(P2V既存環境移行)
- 運用保守
 - エンジニア教育
 - 技術サポートの提供
 - OSSソースコードレベルサポート

ベンダーニュートラルなワンストップ・サポートをご提供

3



スタッフ大募集中

- 仮想化技術関連業務 拡大中
 - ハイパーバイザー関係未経験もOK
 - ストレージ、ネットワーク系を強化中
- OpenStack案件プロジェクトメンバー急募
 - インフラ構築・運用経験ある方歓迎
 - オブジェクト指向プログラミングができる方
 - Pythonなどで開発します

4



VTJで働くと、いいことあるよ

渋谷駅徒歩5分の新オフィス



幅140cmのゆったりデスクとアロンチェア



充実の検証環境

5



仮想化技術にどっぷり浸りたい人

- 展示ブースに来てね
- 後日メールもOK

recruit@VirtualTech.jp

6

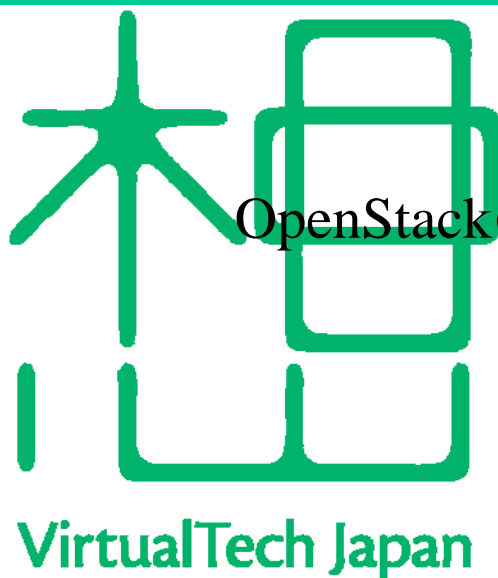


本日のアジェンダ

- OpenStackの概要
- OpenStack導入手順
 - Ubuntu Server 12.04のインストールと設定
 - DevStackによるOpenStackのインストール
 - イメージの作成
 - インスタンスの起動
 - Floating IPによる外部接続設定

まずはOpenStackを体験してみたい人向け

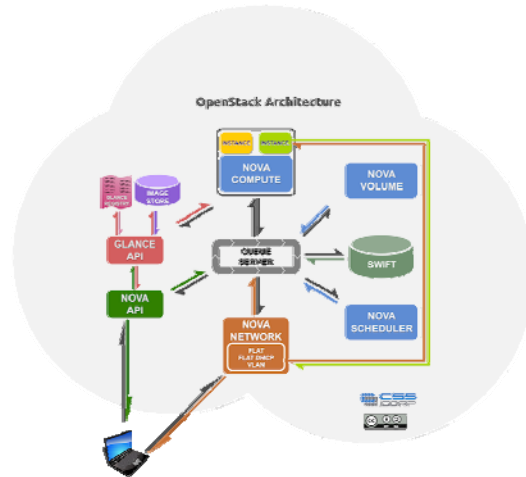
7



OpenStackの概要



OpenStack構成図



9



OpenStackの構成要素(1)

- Nova
 - コンピュータ(計算機)インフラ
 - 仮想マシンやネットワークを管理
- Swift
 - ストレージインフラ
 - 今回は少しだけ使ってる
- Glance
 - イメージ管理
 - 仮想マシンのイメージを登録

10



OpenStackの構成要素(2)

- Horizon
 - Dashboardとも呼ばれる
 - Web管理インターフェース
- Keystone
 - 認証系サービス
 - Identity/Token/Catalog/Policy

11



Novaの主なコンポーネント

- API Server (nova-api)
 - 外部にAPIを提供
- Message Queue (rabbit-mq server)
 - 各コンポーネントへの通信をキューイング
- Compute Workers (nova-compute)
 - インスタンスの管理
- Network Controller (nova-network)
 - ネットワークの管理
- Volume Worker (nova-volume)
 - ボリュームの管理
- Scheduler (nova-scheduler)
 - API呼び出しのコンポーネントへの割り当て

12



今回の方針

- Ubuntu Server 12.04をベースに構築
- DevStackを使ってとりあえずOpenStackを動かす
- すべてを1台でまかなう
 - 必要に応じて複数台での構成も可能
- 参考)ゲストOSをUbuntu Server 12.04.1で構築
- Floatin IPで外部からインスタンスに接続できるようにする

13

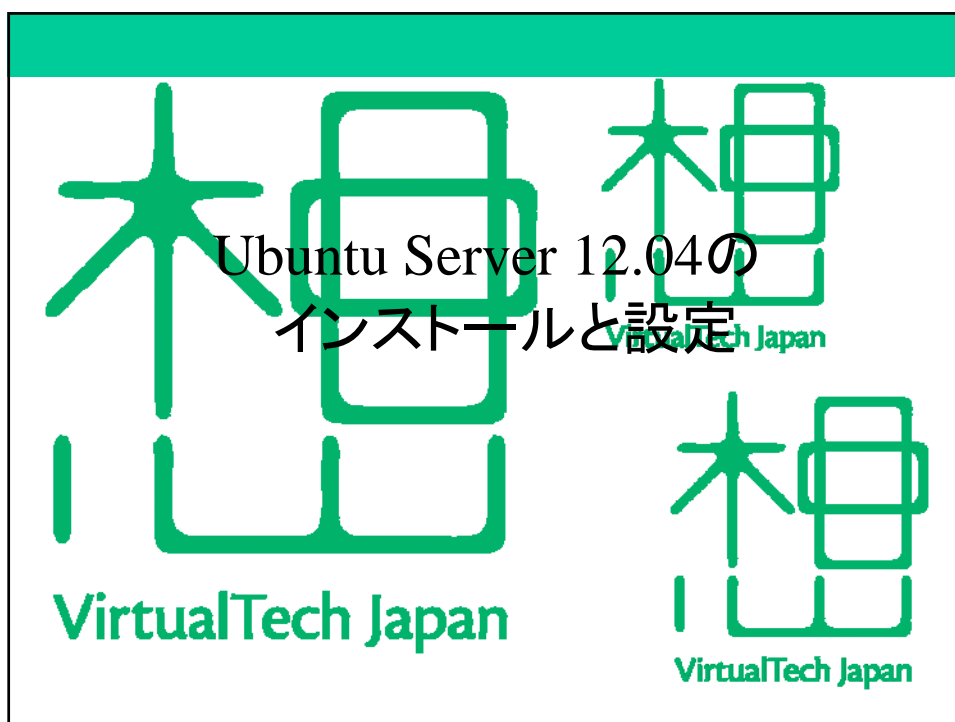


DevStackとは

- OpenStackの開発環境を容易に構築できるシェルスクリプト群
 - 動作検証目的でも十分利用可能
 - All in Oneや複数Novaノードなど各種形態
- 必要となるパッケージをaptでインストール
- gitのリポジトリからOpenStackの各種ファイルを取得
- 一般ユーザー権限で実行

14





Ubuntu Server 12.04のインストールと設定

1. ベースOSとしてのインストール
 - Englishでインストールする
 - ロケールをUnited Statesに設定する
 - KVMを有効にしておく
2. デスクトップ環境のインストール
3. 仮想マシンマネージャーのインストール
4. IPアドレスの設定
 - 固定IPアドレスに設定

ベースOSとしてのインストール

1. インストール時に以下のコンポーネントを追加でインストールしておく
 - OpenSSH server
 - Virtual Machine host
2. デスクトップ環境のインストール

```
# apt-get update
# aptitude -y install
  ubuntu-desktop
```
3. システムの再起動
4. 仮想マシンマネージャーのインストール

```
# apt-get -y install virt-manager
```

17



固定IPアドレスの設定

1. /etc/network/interfacesの編集

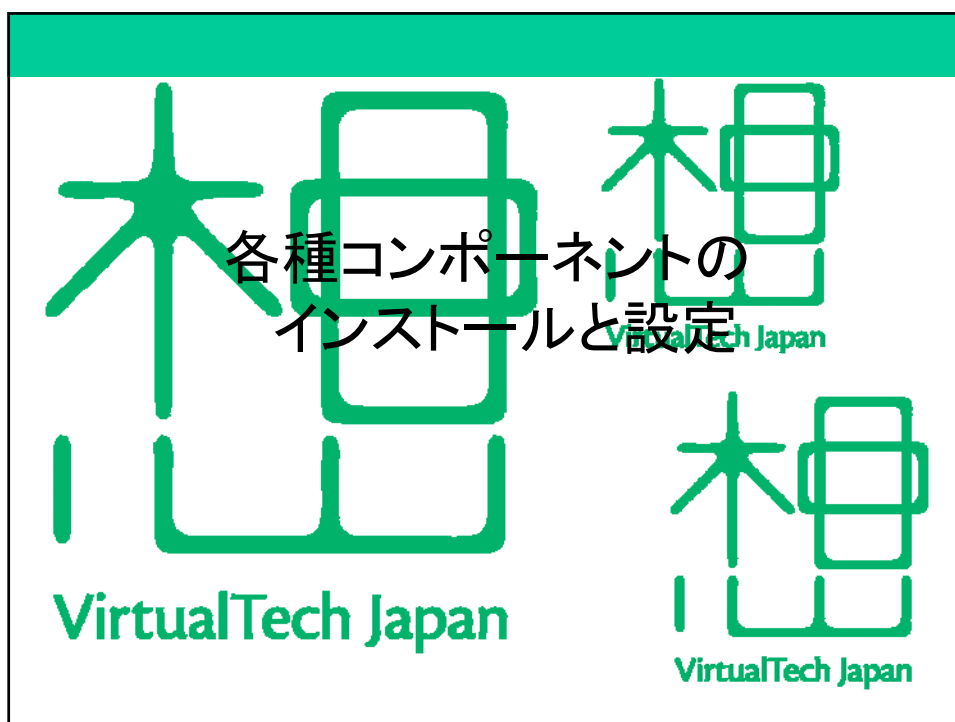
```
auto eth0
iface eth0 inet static
    address 192.168.1.100
    network 192.168.1.0
    netmask 255.255.255.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
    dns-nameservers 192.168.1.1
```

1. networkingサービスの再起動

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

18





各種コンポーネントのインストールと設定

1. gitのインストール
2. DevStackのソースを取得
3. DevStackの設定
 - localrcの作成
4. stack.shの実行

gitのインストール

- DevStackのソースを取得するためにgitをインストールする
\$ sudo apt-get update
\$ sudo apt-get -y install git

21



DevStackのソースを取得

1. githubよりDevStackを取得
\$ cd
\$ git clone https://github.com/openstack-dev/devstack.git
\$ cd devstack/
2. 安定版のEssexを使うようにブランチを設定
\$ git checkout -b essex remotes/origin/stable/essex
3. ブランチの設定が完了すると以下のメッセージを表示
Branch essex set up to track remote branch stable/essex from origin.
Switched to a new branch 'essex'

22



~/devstack/localrcの設定(1)

~/devstack/localrcに様々な設定を記述

1. adminのパスワード
2. MySQLのパスワード
3. ホストのIP
4. Floating RangeのIP範囲の設定
5. Fixed RangeのIP範囲の設定

23



~/devstack/localrcの設定(2)

6. 導入するOpenStackのバージョンを指定
 - DevStackは標準で開発版(最新版)のOpenStackをインストールするため
 - バージョン固定して安定版をインストール

24



localrcの編集(1)

```
ADMIN_PASSWORD=openstack
MYSQL_PASSWORD=stackdb
RABBIT_PASSWORD=stackrabbit
SERVICE_PASSWORD=$ADMIN_PASSWORD
HOST_IP=192.168.1.100
FLOATING_RANGE=192.168.1.192/26
FIXED_RANGE=10.0.0.0/24
FIXED_NETWORK_SIZE=256
SERVICE_TOKEN=token
FLAT_INTERFACE=br100
```

25

※次のページに続く



localrcの編集(2)

```
GLANCE_BRANCH=stable/essex
HORIZON_BRANCH=stable/essex
KEYSTONE_BRANCH=stable/essex
NOVA_BRANCH=stable/essex
QUANTUM_BRANCH=stable/essex
SWIFT_BRANCH=1.4.8
ENABLE_SERVICE=swift
EXTRA_OPTS=(
vnc_keymap=ja
)
```

26



stack.shの実行

- stack.shを実行
 - \$ cd ~/devstack
 - \$./stack.sh
- 環境の構成が行なわれる
 - localrcに指定した構成でインストール開始
 - sudoコマンドを呼び出す(要パスワード)
 - 必要なパッケージを自動でインストール
 - 構成に時間がかかる(デモ環境で15分程度)
 - 2回目で3分ぐらい

27



インストールが成功すると

```
Horizon is now available at
http://192.168.1.100/
Keystone is serving at
http://192.168.1.100:5000/v2.0/
Examples on using novaclient command line is in
exercise.sh
The default users are: admin and demo
The password: openstack
This is your host ip: 192.168.1.100
stack.sh completed in 926 seconds.
```

28



インストールが成功しない？

- ベースOSとしてUbuntu Server 12.04をインストールする時に日本語を選択すると、stack.sh実行時にKeystoneの初期化に失敗します
 - <https://review.openstack.org/#/c/9217/>
 - Allow wget to handle Non-English output
The stack.sh has a loop to wait for keystone to start. But the output of wget tool has been globalized which means it won't return the English word 'refused' on Non-English environment even the keystone is not up. So the script will assume the keystone has been started and continue. The command of keystone tenant-create always failed immediately after skipping this loop since it require keystone to be started to authenticate. That's why you can see authentication error in the log because the tenant information was not correctly set up.
- 解決するには環境設定でEnglishを優先言語に変更し、ログインし直してください

29



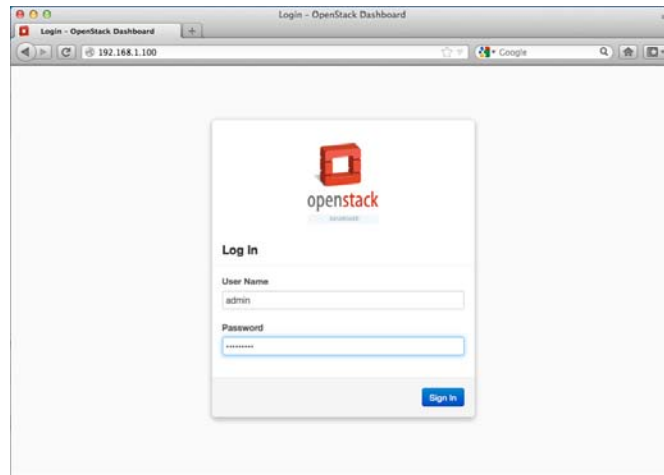
Web管理インターフェースへの接続

- 以上の設定でOpenStackの導入は完了
- テスト用にイメージが登録されている
 - CirrOSが動作する
- Web管理インターフェイス(Horizon)にアクセスするとインスタンスの作成などが行なえる
 - <http://192.168.1.100/>
 - ユーザー名 : admin
 - パスワード : openstack

30



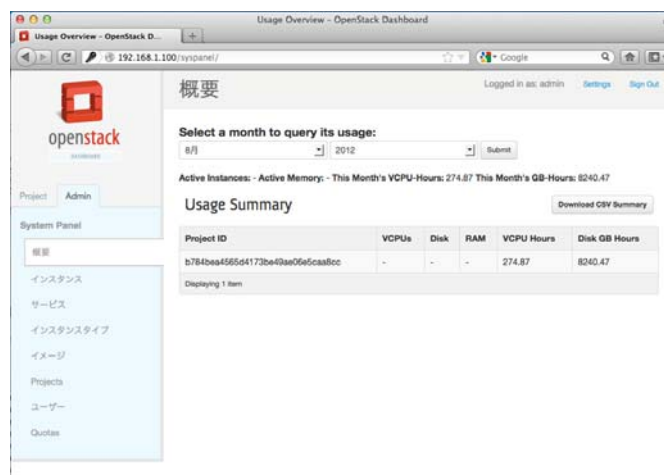
Horizon 認証画面



31



Horizon 概要画面

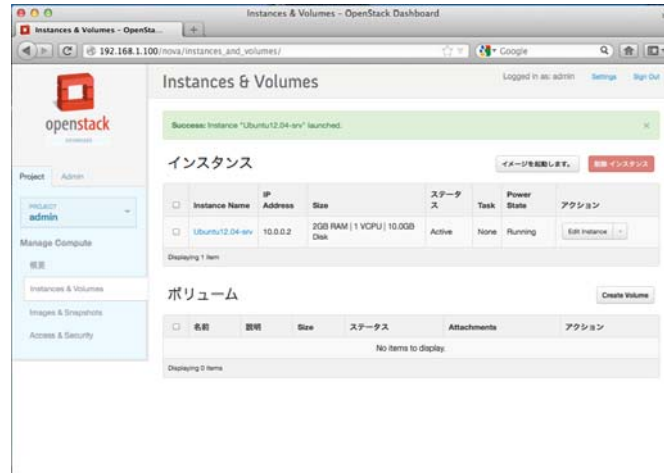


Project ID	VCPU's	Disk	RAM	VCPU Hours	Disk GB Hours
b784bea4565d4173be49ae0fa5ca8fcc	-	-	-	274.87	8240.47

32



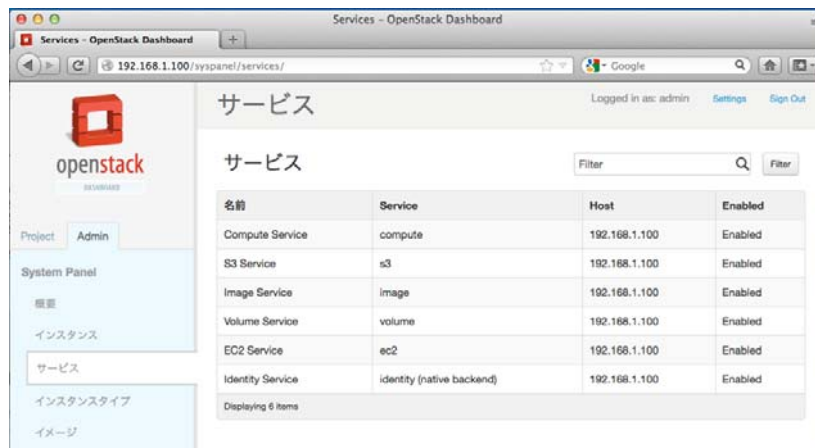
Horizon インスタンス画面



33



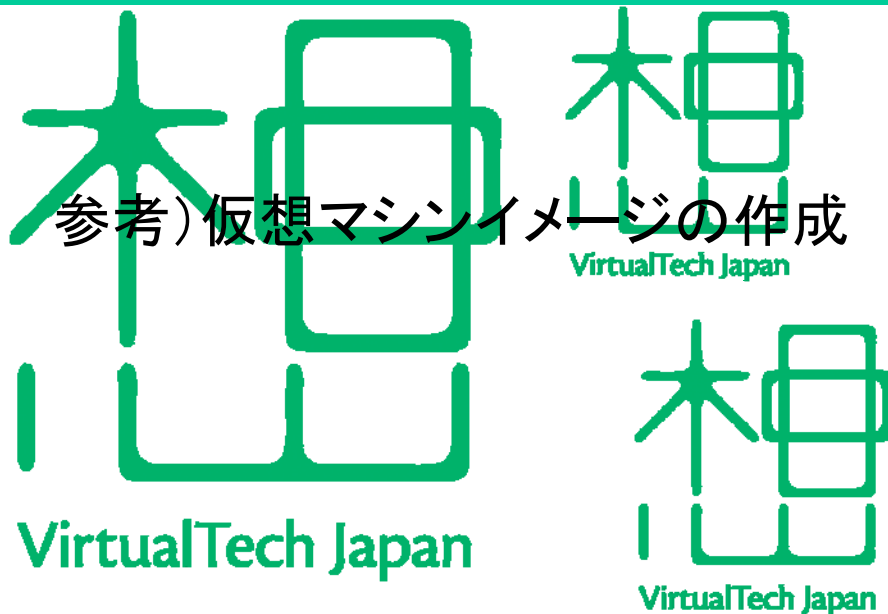
サービス状態を確認



いくつかTips

- DevStackを停止する方法
 - \$./unstack.shを実行
 - 仮想マシンのインスタンスは実行されたまま
- インターネット接続無しで起動する
 - 最低でも初回はインターネット接続が必要
 - \$ OFFLINE=True ./stack.shとして実行
- stack.shの中身を読むと色々書いてある

35



手順概要

- 以下の手順でOpenStackでお好みのOSインスタンスを利用できるようになります。
 1. 仮想マシンマネージャーでVMを作成
 2. VMにOSをインストール
 3. glance addコマンドで登録



ゲストOSのインストール

- 仮想マシンマネージャーを使って仮想マシンを作成
- ゲストOSとしてUbuntu 12.04.1 Serverをインストール
 - OpenSSH Serverをインストールすること
- 仮想ディスク作成の注意点
 - ストレージを手動で定義
 - server.imgとして作成
 - ボリューム形式はqcow2



仮想ディスク作成例

ストレージボリュームを追加

新規ストレージボリューム
仮想マシンから直接使用できるストレージユニットを作成します。

名前(N): server .img
形式(F): qcow2

ストレージボリュームのクォータ
default's 使用可能な領域: 126.54 GB

最大容量(P): 8000 MB
割り当て(A): 8000 MB

Name: 作成する
ボリュームの名前。ファイルの
拡張子を追加
可能

Format: ボリュームのファイル/パーティション
形式

Capacity: ボリュームの最大
サイズ。

Allocation: ボリュームに割り当てる
今回の
実際のサイズ。

Cancel 完了(F)



ゲストOSの設定

1. ゲストOSにログイン
2. アップデートの適用
\$ sudo apt-get update
\$ sudo apt-get upgrade
\$ sudo apt-get -y install cloud-init
※cloud-initはメタパッケージ
3. udevルールの削除
\$ sudo rm -rf /etc/udev/rules.d/70-persistent-net.rules
4. ゲストOSをシャットダウン
※OSにより作業する項目が異なるので注意！



OpenStackにVMイメージを登録

1. openrcファイルを読み込む
\$ source ~/devstack/openrc
2. OpenStackにglance addで登録
\$ **glance add** name="ubuntusrv" is_public=true
container_format=**bare** disk_format=**qcow2** <
server.img

Added new image with ID: 5ae3e229-46cc-4e11-8b3d-5d8b75a20723

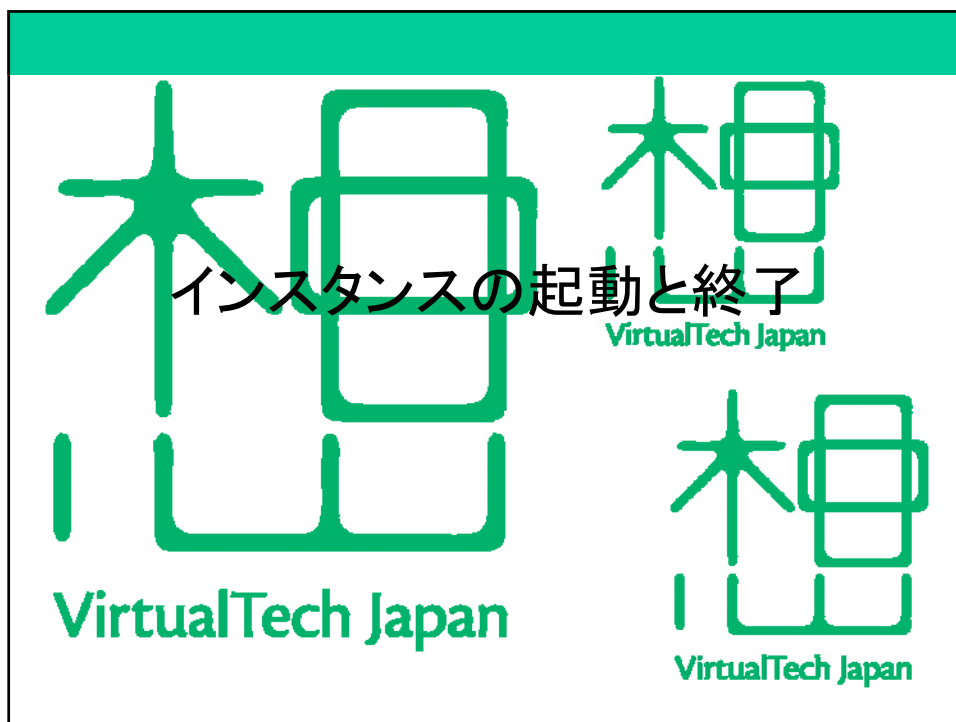


登録を確認

- glance indexコマンドでGlanceに仮想マシン登録されたことを確認

ID	Name	Disk Format	Container Format
5ae3e229-46cc-4e11-8b3d-5d8b75a20723	ubuntusrv	qcow2	bare
81280170-544f-4727-9a9c-d7d4c11b1ae7	cirros-0.3.0-x86_64-uec	ami	ami
624ba501-7462-4834-a2f9-4396b9897e96	cirros-0.3.0-x86_64-uec-ramdis	ari	ari
0d8866f0-5cb4-4249-9332-28bbdba128ed	cirros-0.3.0-x86_64-uec-kernel	aki	aki





インスタンスとは

- OpenStack上でテンプレートイメージから作成した仮想マシンのこと
- インスタンスの管理はダッシュボード上やコマンドラインで操作可能



インスタンスの作成

1. Web管理インターフェースにログイン
2. 「Project」タブをクリック
3. 「Project→Instance&Volumes」を選択
4. 「イメージを起動します。」ボタンをクリック
5. 画面が「Images & Snapshots」に切替わる
– 最初からこっちを選んで良い
6. 「イメージ」から使用したいイメージを選択
7. 「Launch」ボタンをクリック
8. インスタンスの定義を入力



Instance&Volumes 画面

Instances & Volumes - OpenStack Dashboard

Instances & Volumes

Logged in as: admin

プロジェクト admin

Manage Compute

インスタンス

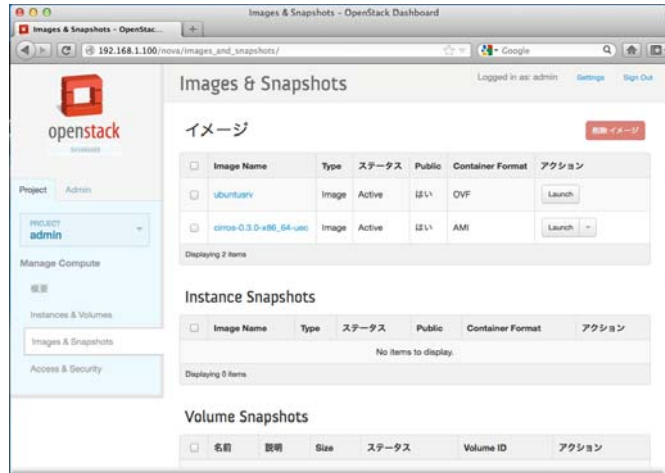
Instance Name	IP Address	Size	ステータス	Task	Power State	アクション
No items to display.						

ボリューム

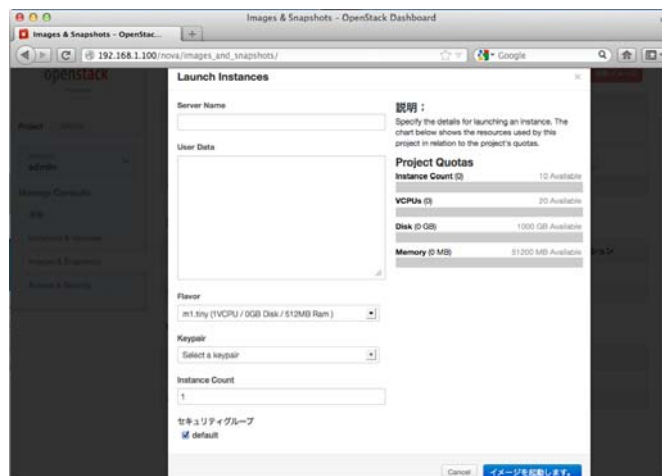
名前	説明	Size	ステータス	Attachments	アクション
No items to display.					



Images & Snapshots 画面



Launch Instances 画面

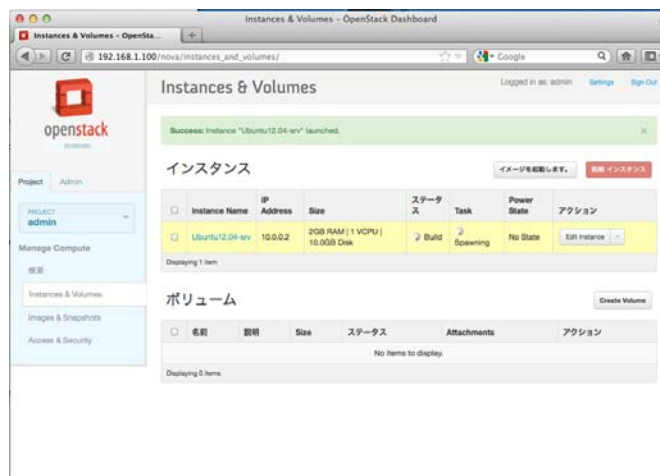


インスタンスの定義

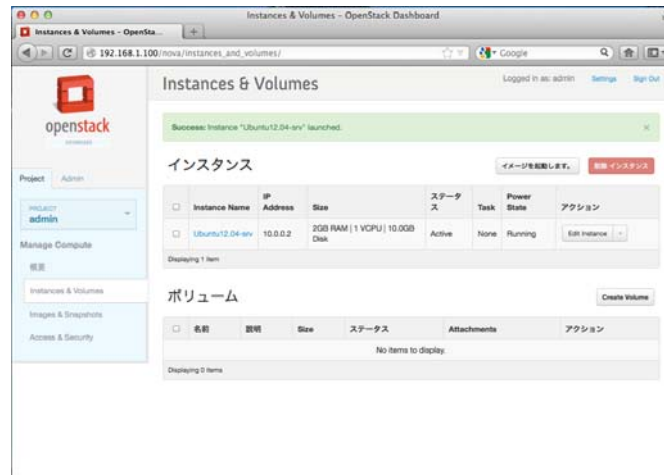
項目	内容
Server Name	インスタンスに付ける名前を入力
User Data	Amazon EC2のUserDataのようにインスタンスに対してスクリプトや設定を渡すことが可能。またメモ書き用としても使われる
Flavor	インスタンスに割り当てる性能を選ぶ
Keypair	鍵認証によってログインする場合に利用。あらかじめキーを発行しておく
Instance Count	一度に作成するインスタンス数を設定
セキュリティグループ	セキュリティ関連の設定を定義したグループの中から任意のものを選択



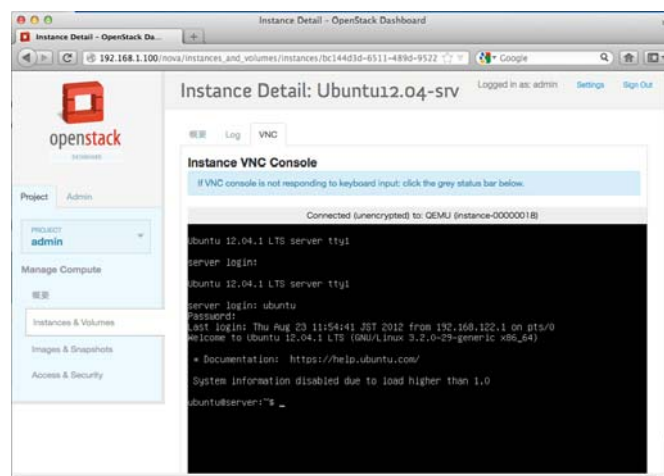
インスタンス作成中



インスタンス作成完了



VNC Consoleで接続



参考)コマンドラインからの操作

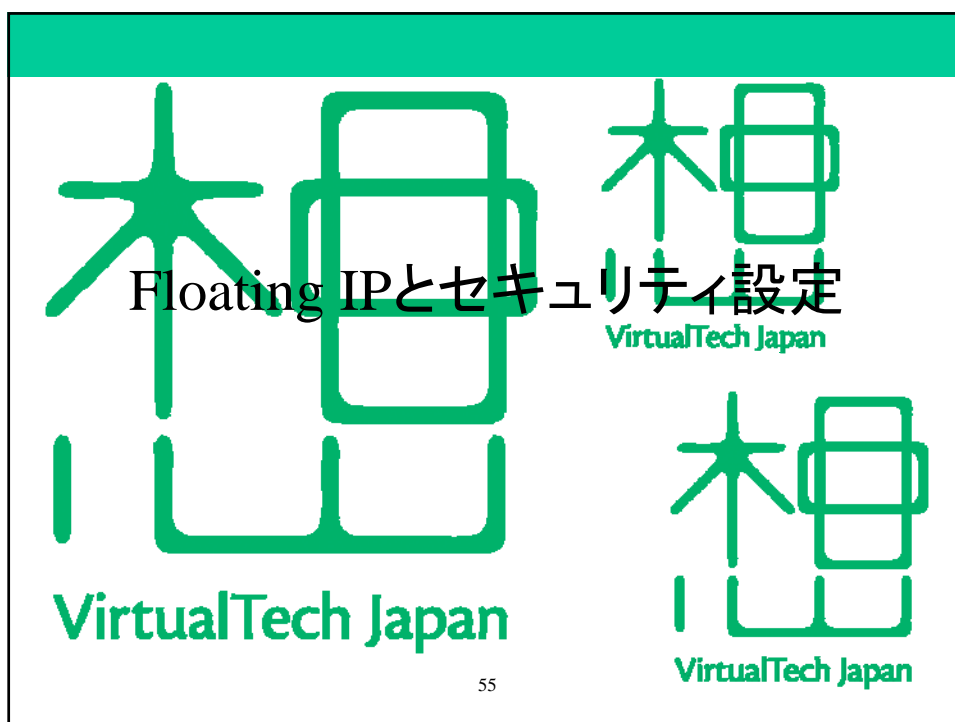
- コマンドを利用したインスタンスの起動や終了も可能
- OpenStackは多くのコマンドとサブコマンドから構成されている
 - OpenStack標準コマンド
 - Euca2oolsコマンド



参考)インスタンスの起動と終了

- 登録イメージの一覧
 - # euca-describe-images
- インスタンスの起動
 - # euca-run-instance [AMIのID](#)
- インスタンスの一覧
 - # euca-describe-instances
- インスタンスへ接続
 - # ssh user@IPアドレス
- インスタンスの終了
 - # euca-terminate-instances [インスタンスID](#)



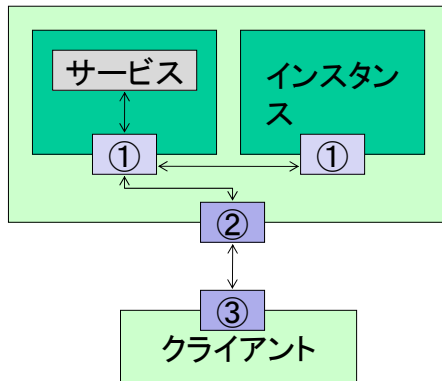


Floating IP 概要

- Floating IPを利用することで外部からインスタンスへアクセス可能
 1. Floating IPを発行
 2. インスタンスにFloating IPを割当
 3. アクセスに必要なポートを開放
 - SSH接続なら22/TCP など



Floating IPの仕組み



- ① FIXED_RANGEで割り当てられる。
①同士は通信できるが、③とは通信できない
- ② FLOATING_RANGEで割り当てられる。実際には②と①の間で静的NATを行っている。
③→②→①と繋がる

57



Floating IPを発行

1. 「Project→Access & Security」を開く
2. Floating IPsの「Allocate IP to Project」をクリック
3. Poolを選択(ここではnova)
4. 「Allocate IP」をクリック

The screenshot shows the 'Allocate Floating IP' dialog box. The 'Pool' dropdown is set to 'nova'. The 'Project Quotas' section shows 'Floating IP (0)' and '10 Available'. The 'Allocate IP' button is highlighted.



Floating IPを発行

4. Floating IPが発行される

- このIPをインスタンスに割当ることで外部からのアクセスが可能になる

Floating IPs				Allocate IP To Project	リリース Floating IPs
<input type="checkbox"/>	IP Address	Instance	Floating IP Pool	アクション	
<input type="checkbox"/>	192.168.1.193	-	nova	Associate IP	▼

Displaying 1 item

59



Floating IPを割当

1. 「Project→Access&Security」を開く
2. 「Floating IPs」の「Allocate IP To Project」ボタンを押す
3. IPを選択して「Associate IP」ボタンを押す

Floating IPs				Allocate IP To Project	リリース Floating IPs
<input type="checkbox"/>	IP Address	Instance	Floating IP Pool	アクション	
<input type="checkbox"/>	192.168.1.193	-	nova	Associate IP	▼

Displaying 1 item

60



Floating IPを割当

5. 割り当てるInstanceを選択して「Associate IP」ボタンを押す

Associate Floating IP ×

Floating IP: 説明: Associate a floating ip with an instance.

Instance:

61



Floating IPを割当

6. インスタンスにFloating IPが設定される

Floating IPs				<input type="button" value="Allocate IP To Project"/>	<input type="button" value="リリース Floating IPs"/>
<input type="checkbox"/>	IP Address	Instance	Floating IP Pool	アクション	
<input type="checkbox"/>	192.168.1.193	6b450791-d42f-4ca0-aa8a-7bda0a79403a	nova	<input type="button" value="Disassociate IP"/>	

Displaying 1 item

62



ポートの開放

- デフォルトでは全てのポートが閉じた状態
 - defaultセキュリティグループ
- セキュリティグループを作成し、必要なポートを開ける
- 「Access & Security→セキュリティグループ」から設定
- Pingも許可する場合はICMPの設定を追加
- アクセス許可するクライアントの対象範囲はSourceで指定

63



ポート開放例

Edit Security Group Rules ×

Security Group Rules 削除 Rules

	IP Protocol	From Port	To Port	Source	アクション
<input type="checkbox"/>	ICMP	-1	-1	0.0.0.0/0 (CIDR)	削除 Rule
<input type="checkbox"/>	TCP	80	80	0.0.0.0/0 (CIDR)	削除 Rule
<input type="checkbox"/>	TCP	22	22	0.0.0.0/0 (CIDR)	削除 Rule

Displaying 3 items

64



Pingを発行

```
$ ping 192.168.1.193
PING 192.168.1.193 (192.168.1.193): 56 data bytes
64 bytes from 192.168.1.193: icmp_seq=0 ttl=63 time=3.023 ms
64 bytes from 192.168.1.193: icmp_seq=1 ttl=63 time=1.241 ms

--- 192.168.1.193 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.241/1.952/3.023/0.771 ms
```

65



インスタンスにSSH接続

```
$ ssh ubuntu@192.168.1.193
ubuntu@192.168.1.193's password:
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux
 3.2.0-29-generic x86_64)

ubuntu@ubuntu-srv:~$
```

66



Webサーバーにアクセス

1. インスタンス上でApacheを実行
\$ sudo apt-get install apache2
2. クライアントからサーバーにアクセス



トラブルシューティング

- 何かうまく動作しない
 - イメージサービスは処理完了に時間がかかる
 - Web管理インターフェースでサービス状態を確認
 - topコマンドでプロセス状態を確認
- ログはどれを見ればいい？
 - /var/log/nova/ディレクトリのnova-compute.log、nova-api.logあたりを見る
 - DevStackの場合、ログは別途設定が必要
- ドキュメントは？
 - <http://docs.openstack.org/> を参照



いくつかの宿題

- 複数台によるOpenStack環境の構成
 - nova-computeを複数台とか
- 外部からのネットワーク接続
 - Floating IPを利用してグローバル接続
 - VPNを利用してプライベート接続
- Swiftの利用
 - 今回は使ってるような、使っていないような
- DevStackを使わないインストール

69



スタッフ大募集中！

- 日本仮想化技術株式会社は常時スタッフを募集中
- 『クラウド方面で何かチャレンジしたい！』という人
- これまでのスキルを活かしたキャリアを探している人
- 将来起業してみたいという人

自由闊達な社風の我が社と一緒に働きませんか？

まずは気軽に recruit@VirtualTech.jp まで

70



仮想化について相談したい

無料コンサルティング実施中

- 既存環境を仮想化移行する概要設計
- 他社提案に対する「セカンドオピニオン」
- もちろんOpenStack構築・カスタマイズも

お気軽にお問い合わせください

71



お問い合わせ先

「仮想化環境を構築したいが、どこに相談すればいいの？」

まずは我々にご相談ください



日本仮想化技術株式会社

<http://VirtualTech.jp/>
sales@VirtualTech.jp
03-6419-7841

72

