



沖縄に集まろう。

SDN、クラウドのブルーオーシャンへ  
世界から沖縄へー沖縄から世界へ

# 2014年度 クラウドネイティブPJ 成果報告

一般社団法人 沖縄オープンラボラトリ  
林 秀保

1. クラウドネイティブPJのご紹介
2. **RACK**について
3. **OpenStack Summit Paris 2014**デモのご紹介
4. **OpenStack Days Tokyo 2015**展示内容のご紹介
5. 今後の取り組み

# 1.クラウドネイティブPJのご紹介

# クラウドネイティブPJについて

## ➤ Overview

- OpenStackの仮想マシンを自律的に**スケールUP & Down**可能なアプリケーションを実装できる**RACK**（Real Application Centric Kernel）というソフトウェア。CTCがOSS化。
  - ・RACKはクラウドネイティブなアプリケーションを簡単に作成できる。
  - ・ユースケース：処理量に応じた動的なWebサーバのスケールアウト等
- RACKの動作検証及び、簡単にRACKを開発できる環境を構築する。

## ➤ Motivation

- OpenStack上でアプリケーションがリソースを自律的にスケールさせることは、これまでのアプリ開発手法では開発敷居が高い。
- RACKによりクラウドネイティブなアプリケーションの開発敷居が低くなり、アプリケーション開発が活発になることに期待が見込まれる  
⇒ラボとしても注目。

## ➤ Goal

- RACKを動かす環境を構築 & 検証を実施。  
⇒まずは、RACKをうごかしてみる。
- RACKをWEB UIから動かせる開発ツールを構築する。  
⇒構築とアプリケーションの実行の方法がCLIのみ。  
複数のアプリを動かす場合等、コンソールを複数立ち上げたり  
アプリケーションを作成する手順が煩雑。

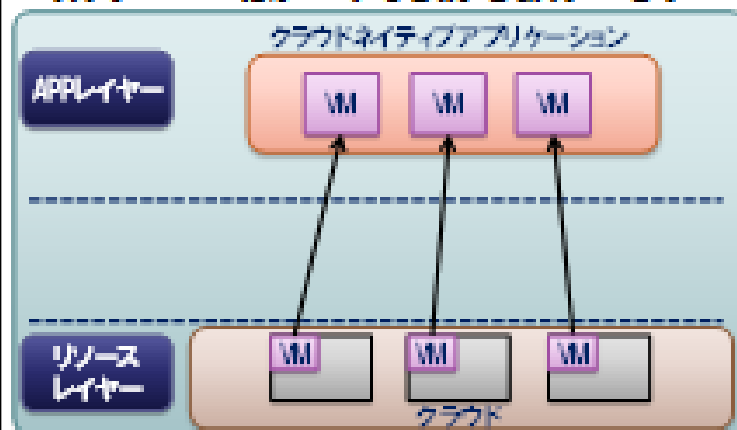
## 2.RACKについて

# RACKとは（概要）

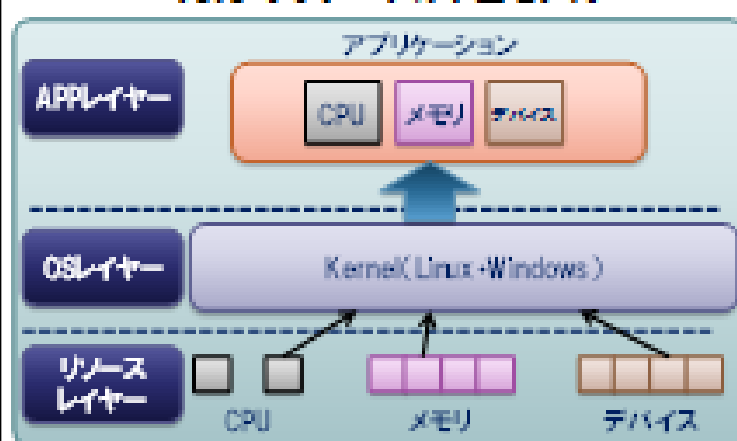
## 【参考】RACKとは



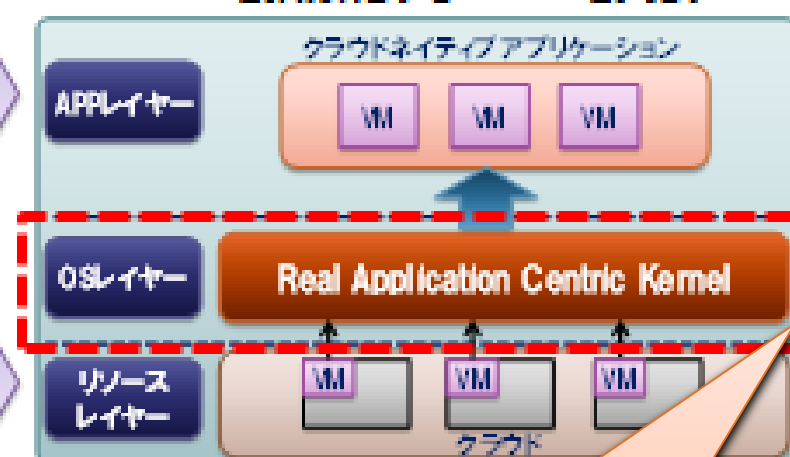
素のIaaSはプログラムから扱いづらい



現行のリソース管理モデル



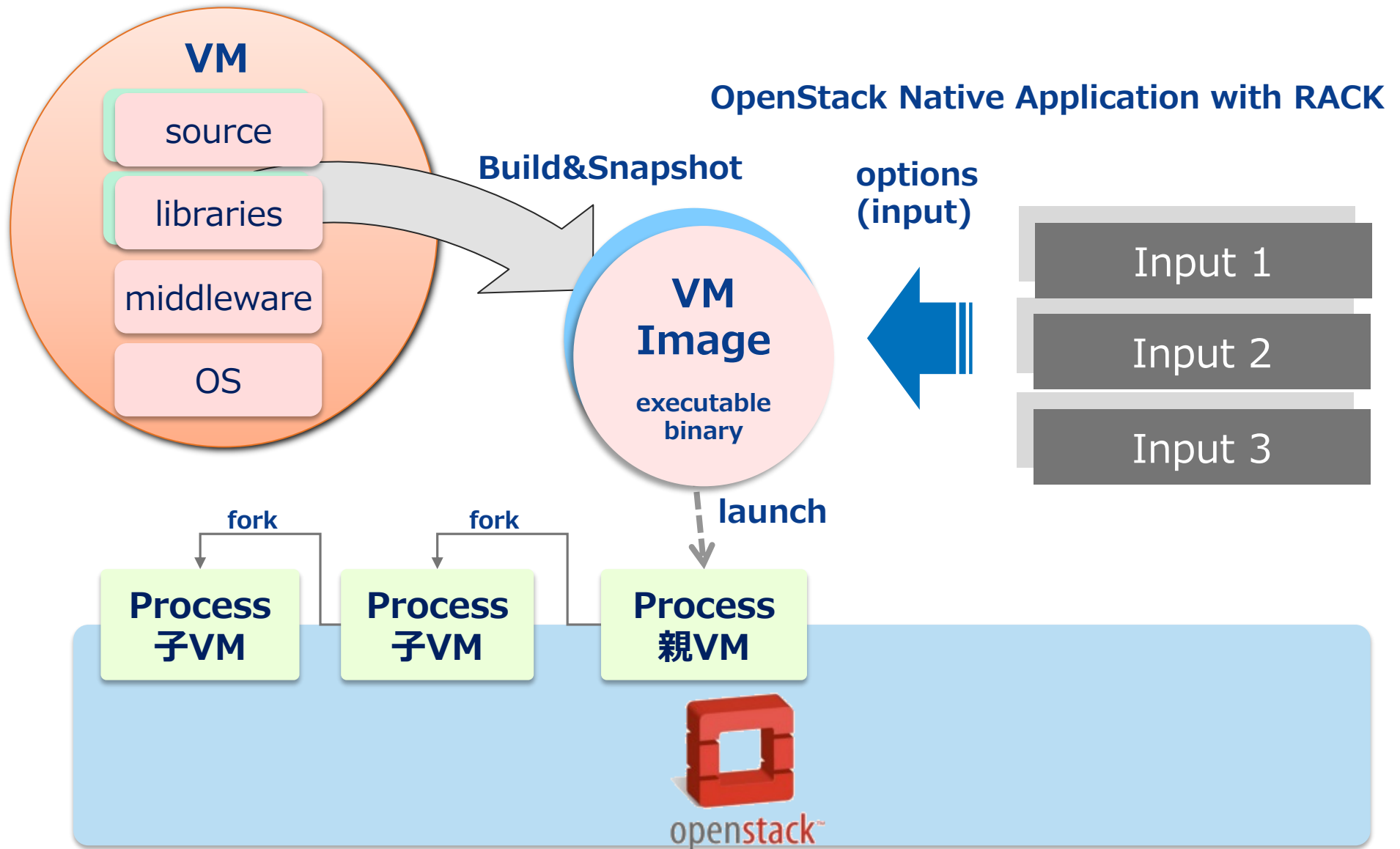
IaaSを抽象化するRACKを実装



RACKは以下のAPIを提供

- 1) VMの起動、終了
- 2) VM間のメッセージ通信
- 3) VM間のファイル共有

# RACKとは (OpenStackでの仕組み)





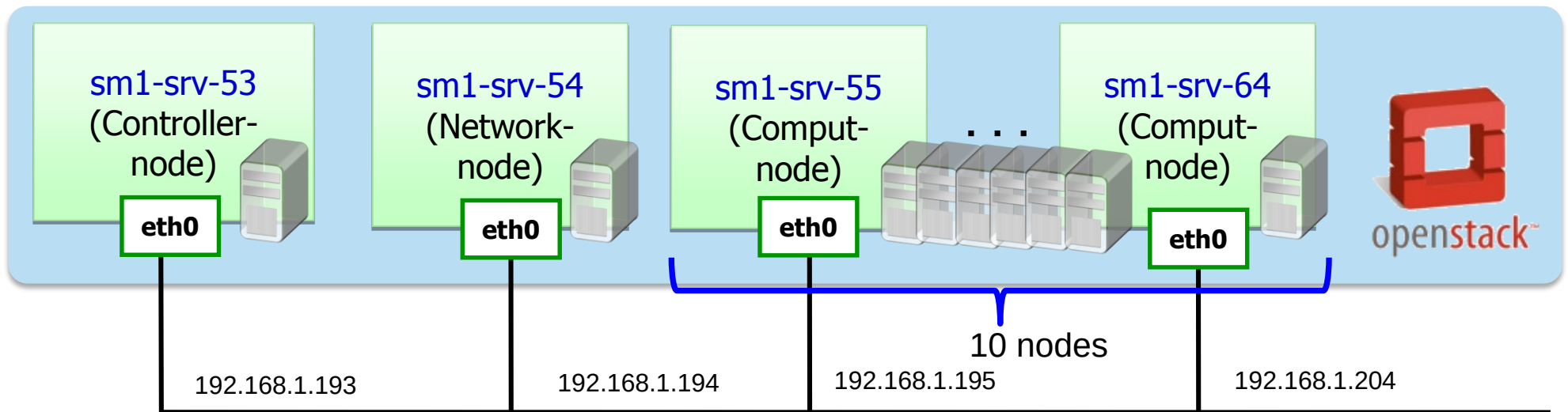
# **OpenStack Summit Paris 2014の展示**

## **(RACK動作環境の構築 & 検証)**

# RACK構築環境



## AMD SeaMicro15000



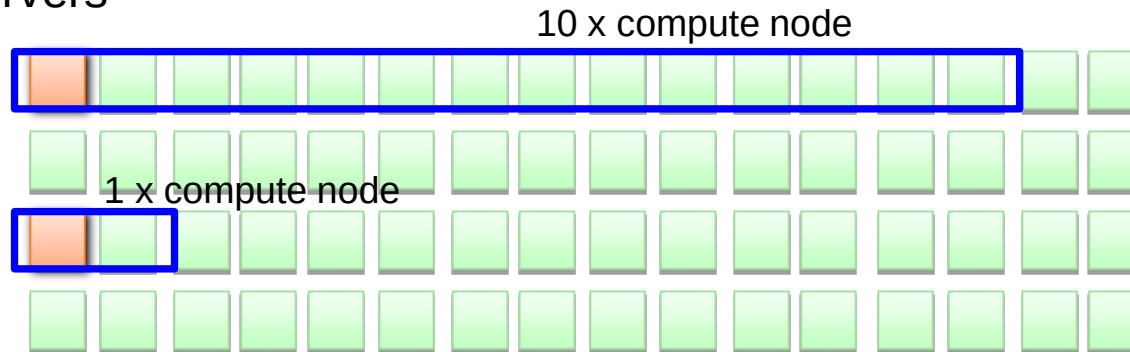
C-Plane

# RACK Test Suite Environment

AMD/Seamicro SM15K



64 servers



## 条件

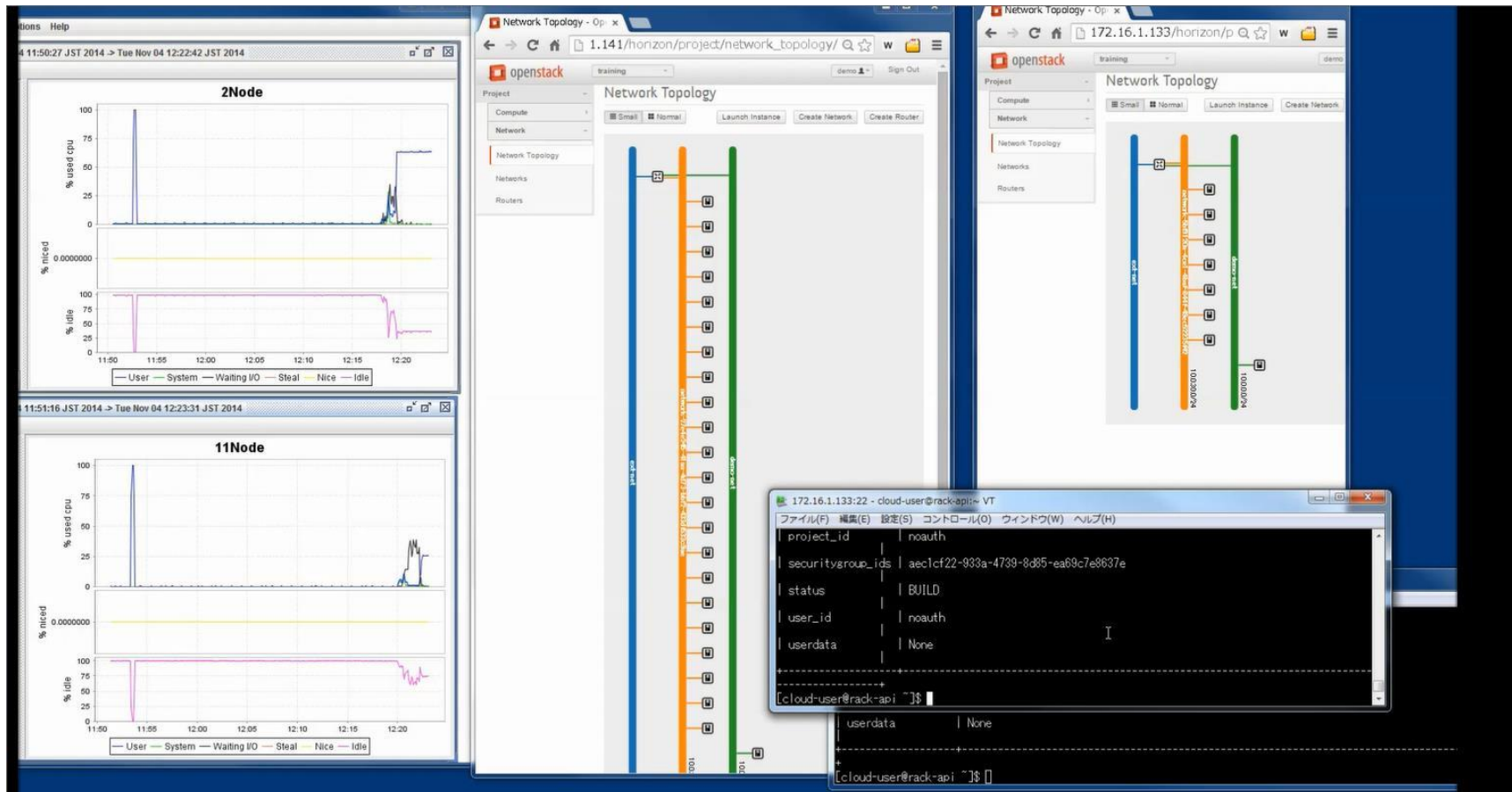
- 1) OpenStack環境を 2 つ作成⇒ 2 台構成と 1 1 台構成の系
- 2) 高負荷な演算処理を実施。

それぞれの環境でサーバのCPU使用率、処理時間を観測

| 構成     | 構成 1 | 構成 2 |
|--------|------|------|
| サーバー台数 | 2    | 11   |
| 仮想マシン数 | 5    | 20   |
| 試行回数   | 30億回 | 30億回 |

# RACK 検証

結果：2 台構成の系では、CPU 資料率はピークに張り付いたまま。  
1 1 Node 構成の系では CPU 使用率はすぐに収束 ⇒ 処理が分散



## 結果 (実行時間、MAX-CPU使用率)

| 構成              | 構成 1 | 構成 2 |
|-----------------|------|------|
| サーバー台数          | 2    | 11   |
| 仮想マシン数          | 5    | 20   |
| 試行回数            | 30億回 | 30億回 |
| 実行時間 (秒)        | 423  | 135  |
| CPU使用率<br>(MAX) | 約63% | 約37% |

実行時間が短縮され、CPU使用率のピークが低減  
RACKを用いて処理が分散

# **OpenStack Summit Tokyo2015の展示 (*RACK* 開発ツールの作成)**

# RACK開発Toolのコンセプト

Developer



1.push!

2.build!

3.Load&Run!

push



Source

Git Hub

pull



Your APP

Application  
Builder

Load



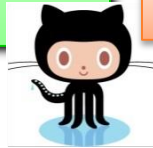
Your Image

Application  
Runner

Run



Result



## RACK Applications List

+Create Application

| Application Name | Image Id       | Built Date          | Status                                                                                      | Operation  |
|------------------|----------------|---------------------|---------------------------------------------------------------------------------------------|------------|
| Monte carlo      | F13878498-3132 | 2014/11/28:13:14:51 |  Building | Edit More▼ |
| Twitter          | 7eqw08eq-12344 | 2014/11/23:11:14:51 | Built                                                                                       | Edit More▼ |
| Pipe test        | 789w08eq-1234  | 2014/11/23:11:14:51 | ready                                                                                       | Edit More▼ |

Build  
Build Log  
Delete

## Run Enviroment List

+Create Application

| Gid   | Application Name | Flavor                            | Running VM Count | Run Date            | Status                                                                                  | Operation  |
|-------|------------------|-----------------------------------|------------------|---------------------|-----------------------------------------------------------------------------------------|------------|
| 12345 | Monte carlo      | m1.tiny VCPU 1 RAM 512G HDD 1G    | 4                | 2014/11/28:13:14:51 |  run | Edit More▼ |
| 23569 | Twitter          | m1.small VCPU 1 RAM 2048G HDD 20G | 0                | 2014/11/23:11:14:51 | stop                                                                                    | Edit More▼ |
| 77369 | Pipe test        | m1.small VCPU 1 RAM 2048G HDD 20G | 3                | 2014/11/23:11:14:51 | run                                                                                     | Edit More▼ |

## RACK Applications

### RACK Applications List

| Gid   | Application Name | Flavor                            | Running VM Count | Run Date            | Status | Operation  |
|-------|------------------|-----------------------------------|------------------|---------------------|--------|------------|
| 40534 | monte-test       | m1.tiny VCPU 1 RAM 512G HDD 1G    | 4                | 2014/11/28:13:14:51 | run    | Edit More▼ |
| 78403 | twitter-test     | m1.small VCPU 1 RAM 2048G HDD 20G | 0                | 2014/11/23:11:14:51 | stop   | Edit More▼ |
| 90365 | pipe-test        | m1.small VCPU 1 RAM 2048G HDD 20G | 3                | 2014/11/23:11:14:51 | run    | Edit More▼ |
| 8769  | Monte-test       | m1.tiny VCPU 1 RAM 512G HDD 1G    | 4                | 2014/11/28:13:14:51 | run    | Edit More▼ |
| 11638 |                  | 00ab0729-4183                     |                  | 2014/12/05 10:36:00 |        | Edit More▼ |

Result of monte-test

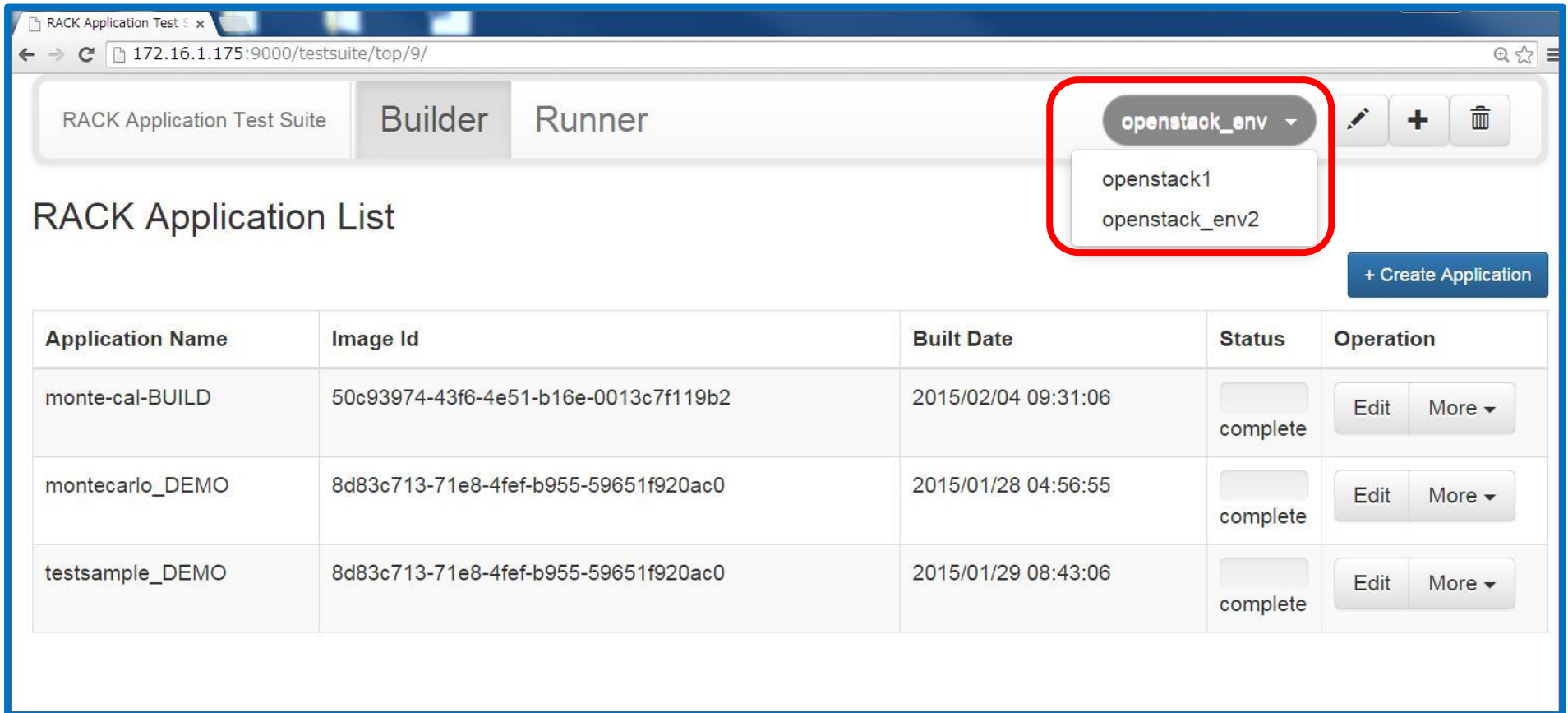
```
$ rack file-get --proxy_ip 172.10.3.51 /output/result.txt
$ cat result.txt
{
  "Property": {
    "Value": {
      " trials": 2000000
    }
  }
}
```

Run  
Log  
Result  
Delete



# RACK DeployTool (特徴 その1)

OpenStack環境を複数登録 & 選択可能  
⇒プライベートクラウド環境や、パブリッククラウド環境で開発可能



The screenshot shows the RACK DeployTool interface. At the top, there are tabs for "RACK Application Test Suite", "Builder", and "Runner". A dropdown menu is open, showing "openstack\_env" selected, with options "openstack1" and "openstack\_env2" listed below it. Below the tabs, the title "RACK Application List" is displayed. A table lists the applications, and a "+ Create Application" button is visible on the right.

| Application Name | Image Id                             | Built Date          | Status   | Operation   |
|------------------|--------------------------------------|---------------------|----------|-------------|
| monte-cal-BUILD  | 50c93974-43f6-4e51-b16e-0013c7f119b2 | 2015/02/04 09:31:06 | complete | Edit More ▾ |
| montecarlo_DEMO  | 8d83c713-71e8-4fef-b955-59651f920ac0 | 2015/01/28 04:56:55 | complete | Edit More ▾ |
| testsample_DEMO  | 8d83c713-71e8-4fef-b955-59651f920ac0 | 2015/01/29 08:43:06 | complete | Edit More ▾ |

# RACK DeployTool (特徴 その2)

Imageの作成は手順が多く、Imageを作成するたびに  
1)～6)を実施⇒わずらわしい。

1)VMを作成

```
nova boot --flavor 2 --image centos build-vm
```

2)VMにログインしGitからソースとビルドシェルを取得

```
ssh cloud-user@191.168.1.244 -i mykey.pem  
git clone https://github.com/stackforge/rack
```

3)Build

```
rack/tools/setup/imagebuild.sh
```

4)VMをShutDown

```
nova stop build-vm
```

5)VMのSnapShotを作成

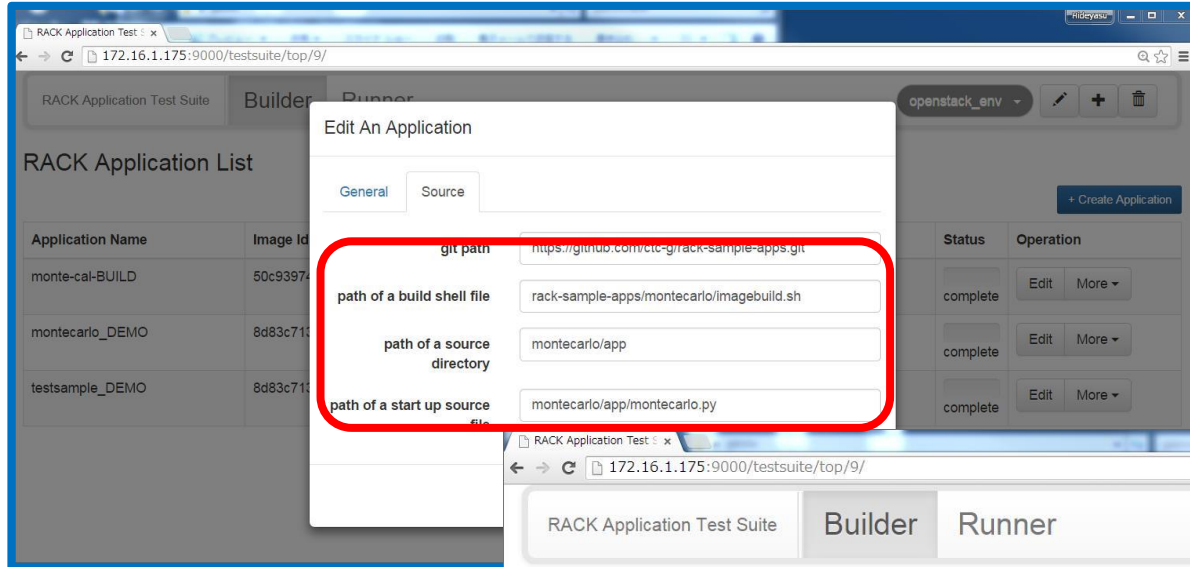
```
nova create-image build-vm build-img
```

6)VMの削除

```
nova delete build-vm
```

# RACK DeployTool (特徴 その2)

RACKアプリケーションを登録したGitHubのパスを指定しBuildを実行  
⇒メニュー選択のみで簡単にイメージ作成が可能。



Edit An Application

General Source

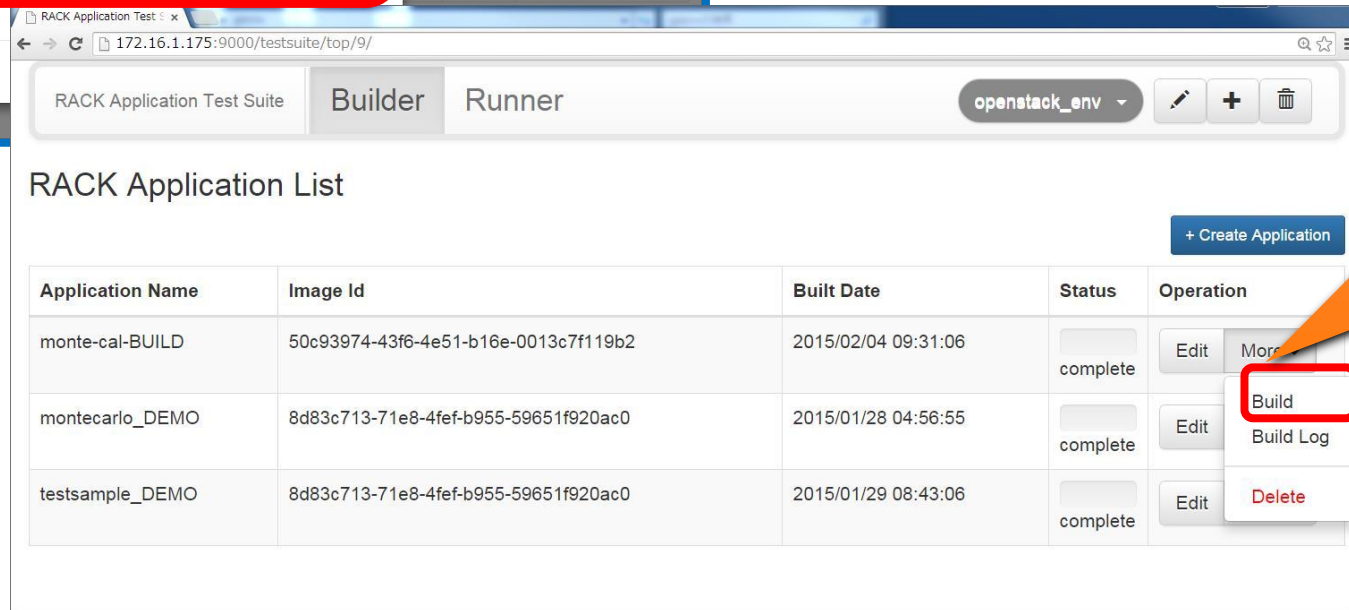
git path:

path of a build shell file:

path of a source directory:

path of a start up source file:

+ Create Application

RACK Application Test Suite Builder Runner openstack\_env

+ Create Application

| Application Name | Image Id                             | Built Date          | Status   | Operation            |
|------------------|--------------------------------------|---------------------|----------|----------------------|
| monte-cal-BUILD  | 50c93974-43f6-4e51-b16e-0013c7f119b2 | 2015/02/04 09:31:06 | complete | Edit More            |
| montecarlo_DEMO  | 8d83c713-71e8-4fef-b955-59651f920ac0 | 2015/01/28 04:56:55 | complete | Edit Build Build Log |
| testsample_DEMO  | 8d83c713-71e8-4fef-b955-59651f920ac0 | 2015/01/29 08:43:06 | complete | Edit Delete          |

## 5. 今後の取り組み

- ◆ RACKを用いたアプリケーションのユースケースの提案、アプリケーションの開発。
- ◆ RACKのAPIの提案（開発を通して感じた不足の機能など）

***Thank You!***