Cloud技術動向

2013年1月

OSS基盤技術センター
OSS技術第二課
岩本 俊弘

目次

- CloudStack内部構造
- OpenStack新プロジェクト
  - Ceilometer
  - Heat

CloudStack

- git-wip-us.apache.org から checkout
- わりと最近(4.1.0)からmavenというビルドシステムを使うようになった
  - 古い手順書にまどわされないように注意

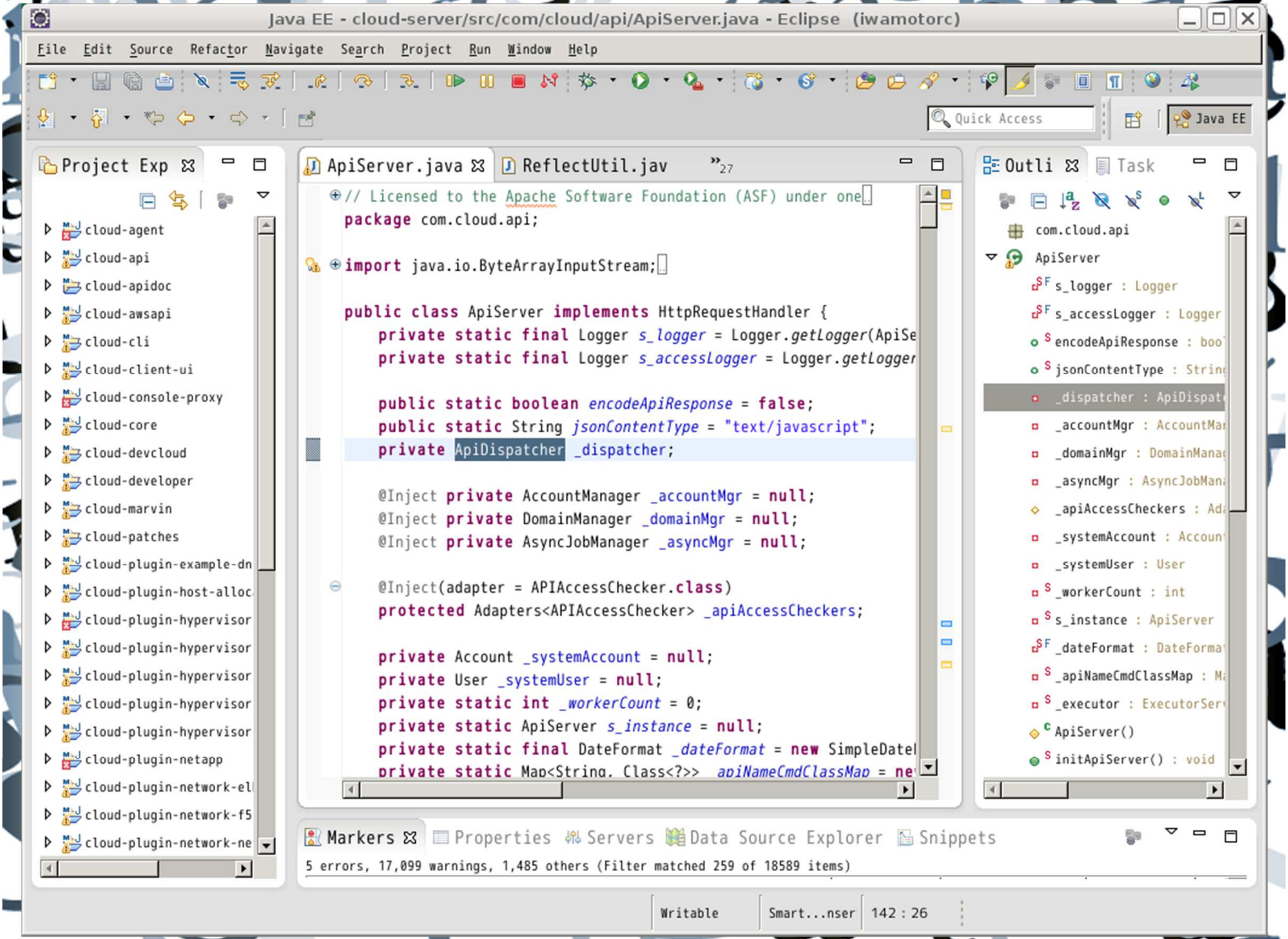
CloudStack – eclipseの設定

- 最近のEclipse（J か K ）
- M2eclipse もinstallする
- Maven (mvnコマンド)も必要
  - Pom.xmlでビルドを制御するもの
- git checkoutしたものを File->Import
- 手順

https://cwiki.apache.org/CLOUDSTACK/using-eclipse-with-cloudstack.html

これに従って追加のjarをinstallする(ライセンス問題で別配布)

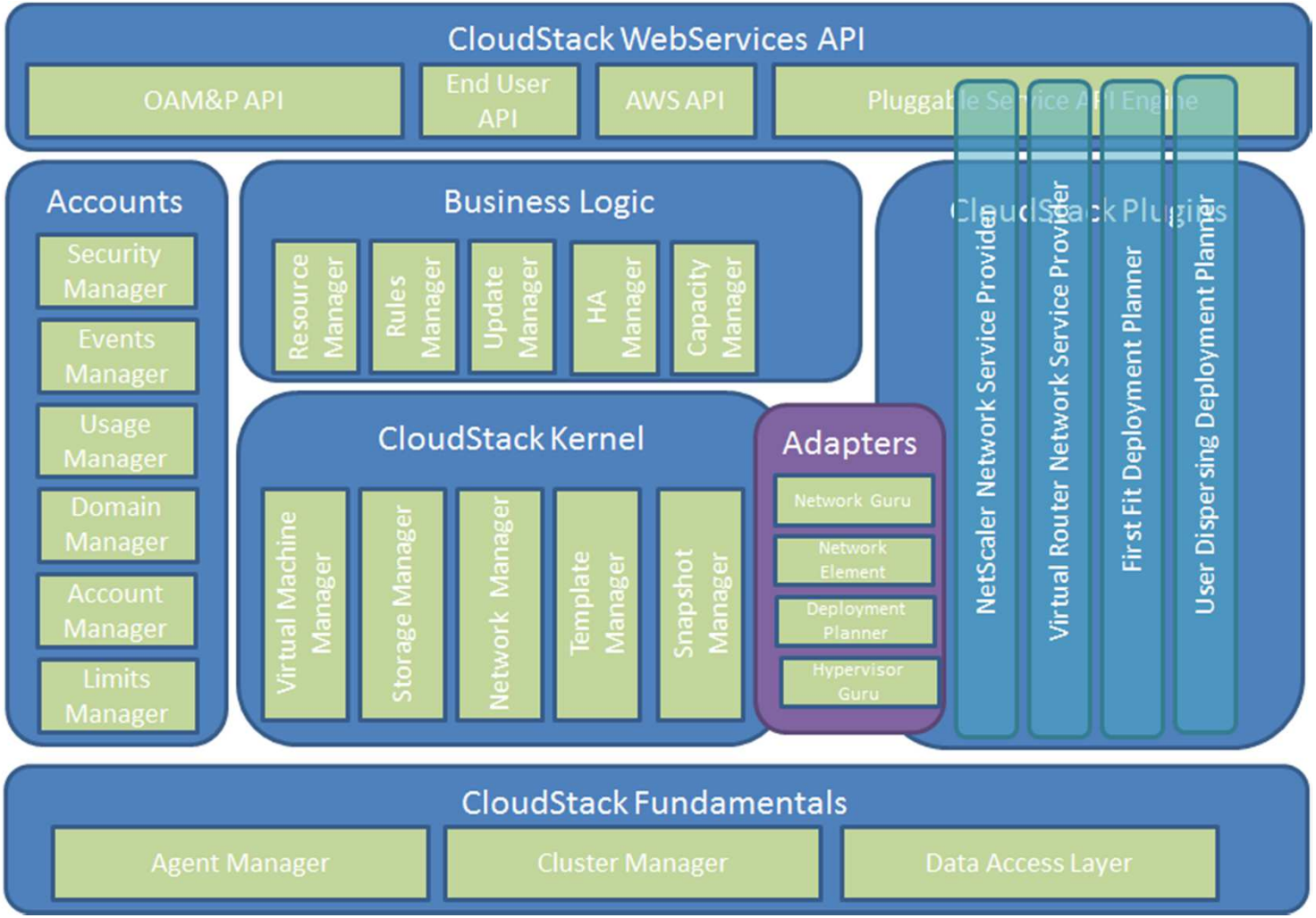
https://cwiki.apache.org/CLOUDSTACK/building-with-maven.html#BuildingwithMaven-Dependencies


Java EE - cloud-server/src/com/cloud/api/ApiServer.java - Eclipse (iwamotorc)
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java EE
Project Exp
cloud-agent
cloud-api
cloud-apidoc
cloud-awsapi
cloud-cli
cloud-client-ui
cloud-console-proxy
cloud-core
cloud-devcloud
cloud-developer
cloud-marvin
cloud-patches
cloud-plugin-example-dn
cloud-plugin-host-alloc
cloud-plugin-hypervisor
cloud-plugin-hypervisor
cloud-plugin-hypervisor
cloud-plugin-hypervisor
cloud-plugin-hypervisor
cloud-plugin-netapp
cloud-plugin-network-el
cloud-plugin-network-f5
cloud-plugin-network-ne
ApiServer.java
ReflectUtil.jav
»27
// Licensed to the Apache Software Foundation (ASF) under one
package com.cloud.api;
import java.io.ByteArrayInputStream;
public class ApiServer implements HttpRequestHandler {
private static final Logger s_logger = Logger.getLogger(ApiSe
private static final Logger s_accessLogger = Logger.getLogger
public static boolean encodeApiResponse = false;
public static String jsonContentType = "text/javascript";
private ApiDispatcher _dispatcher;
@Inject private AccountManager _accountMgr = null;
@Inject private DomainManager _domainMgr = null;
@Inject private AsyncJobManager _asyncMgr = null;
@Inject(adapter = APIAccessChecker.class)
protected Adapters<APIAccessChecker> _apiAccessCheckers;
private Account _systemAccount = null;
private User _systemUser = null;
private static int _workerCount = 0;
private static ApiServer s_instance = null;
private static final DateFormat _dateFormat = new SimpleDatel
private static Map<String, Class<?>> _apiNameCmdClassMap = ne
Outli
Task
com.cloud.api
ApiServer
s_logger : Logger
s_accessLogger : Logger
encodeApiResponse : boo
jsonContentType : Strin
_dispatcher : ApiDispat
_accountMgr : AccountMa
_domainMgr : DomainMana
_asyncMgr : AsyncJobMan
_apiAccessCheckers : Ad
_systemAccount : Accoun
_systemUser : User
_workerCount : int
s_instance : ApiServer
_dateFormat : DateForma
_apiNameCmdClassMap : M
_executor : ExecutorSer
ApiServer()
initApiServer() : void
Markers
Properties
Servers
Data Source Explorer
Snippets
5 errors, 17,099 warnings, 1,485 others (Filter matched 259 of 18589 items)
Writable
Smart...nser
142 : 26

CloudStackの構成

- 予備知識
  - Platform API
    - PluggableService
  - Agent API (JSON)
    - CloudStack componentが使う
    - com.cloud.agent.transport.Request など
  - Plugin API (Java)

CloudStackの構成



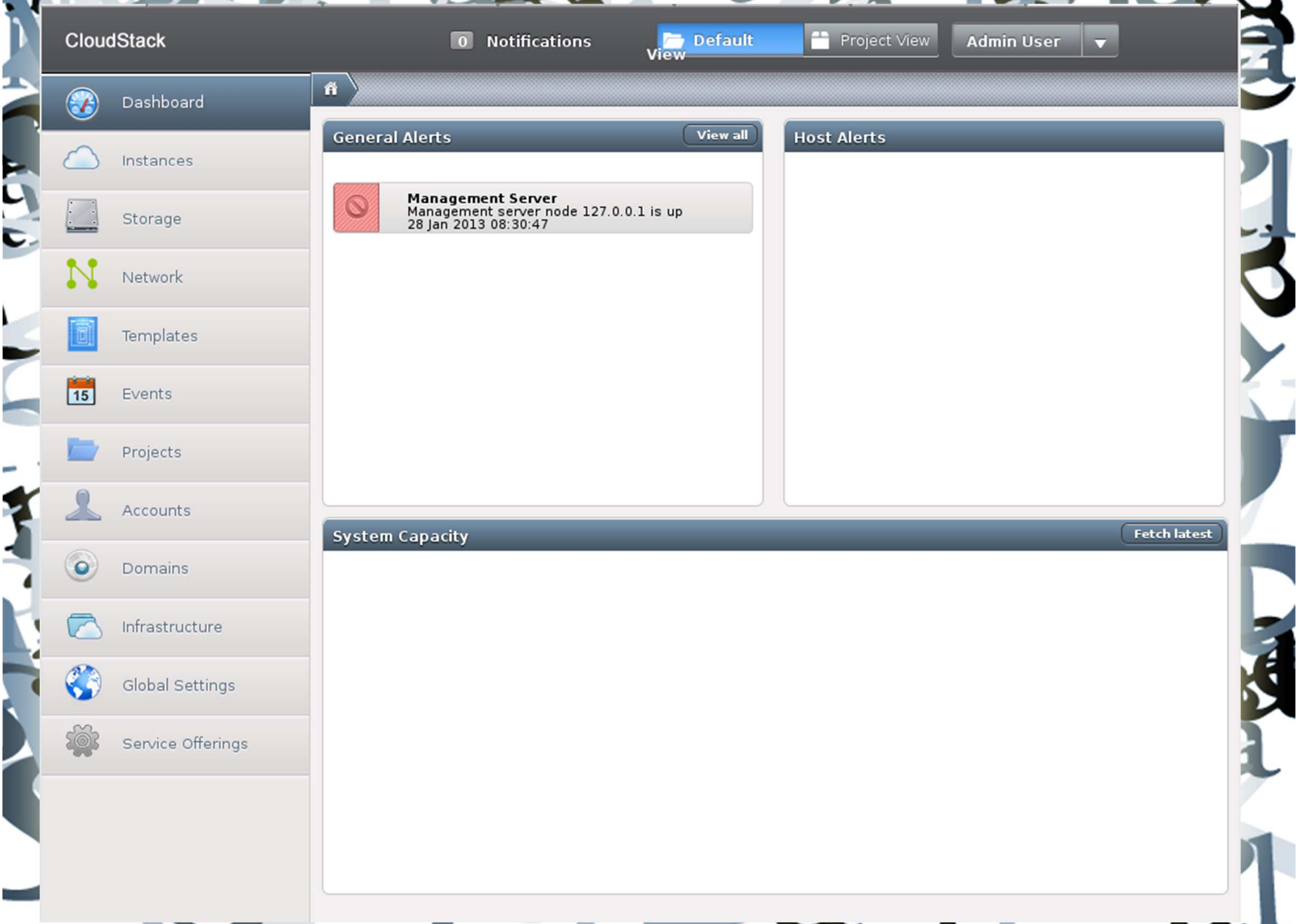
出典 ： https://cwiki.apache.org/CLOUDSTACK/development-101.html

Components.xml

- 以下のcomponentを記述
- ComponentLocatorがパース
  - Manager
  - Adapter
  - DAO
  - Pluggableservice
  - Checker
  - Interceptor

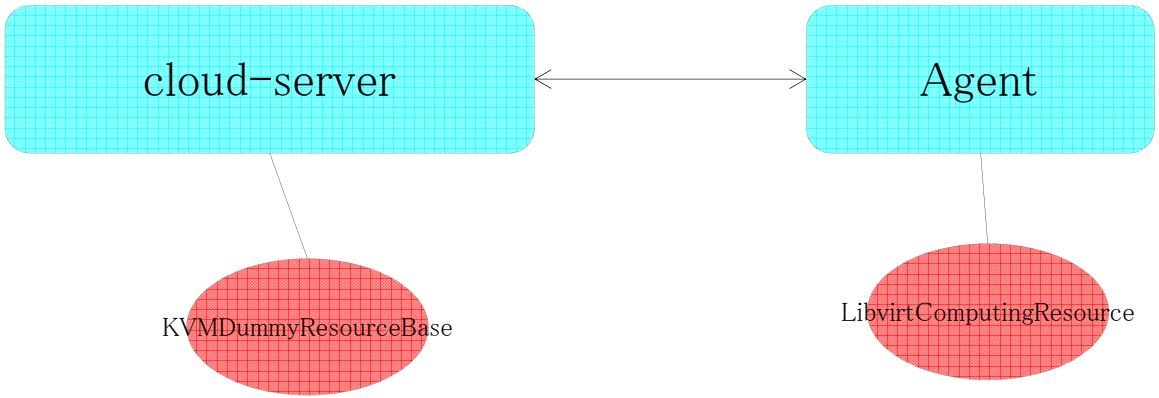
DevCloud

- 開発用のCloudStack環境
- apache.orgにやり方が一応書いてある
  - mysqldの起動、python-setuptools, mysql-connector-python のインストールが必要
  - 先に deploydbしてからinstall するのがよいようである


CloudStack
0 Notifications
Default View
Project View
Admin User
Dashboard
Instances
Storage
Network
Templates
Events
Projects
Accounts
Domains
Infrastructure
Global Settings
Service Offerings
General Alerts
View all
Management Server
Management server node 127.0.0.1 is up
28 Jan 2013 08:30:47
Host Alerts
System Capacity
Fetch latest

Agent, ServerResource

- AgentはJsonでmanagement serverと通信
- AgentはDBを直接は操作しない
- 各AgentにはServerResourceが対応づけられている



Agent側

- こういうことになっている

```
public class Agent implements HandlerFactory, IAgentControl {
(略)
    ServerResource              _resource;
(略)
    protected void processRequest(final Request request, final Link link) {
(略)
                    } else {
                        if (cmd instanceof ReadyCommand) {
                            processReadyCommand((ReadyCommand)cmd);
                        }
                        _inProgress.incrementAndGet();
                        try {
                            answer = _resource.executeRequest(cmd);
```

```
public class LibvirtComputingResource extends ServerResourceBase implements
        ServerResource {
(略)
    @Override
    public Answer executeRequest(Command cmd) {

        try {
            if (cmd instanceof StopCommand) {
                return execute((StopCommand) cmd);
            } else if (cmd instanceof GetVmStatsCommand) {
                return execute((GetVmStatsCommand) cmd);
            } else if (cmd instanceof RebootRouterCommand) {
                return execute((RebootRouterCommand) cmd);
            } else if (cmd instanceof RebootCommand) {
                return execute((RebootCommand) cmd);
            } else if (cmd instanceof GetHostStatsCommand) {
                return execute((GetHostStatsCommand) cmd);
            } else if (cmd instanceof CheckStateCommand) {
                return executeRequest(cmd);
            } else if (cmd instanceof CheckHealthCommand) {
(略)
```

Server側

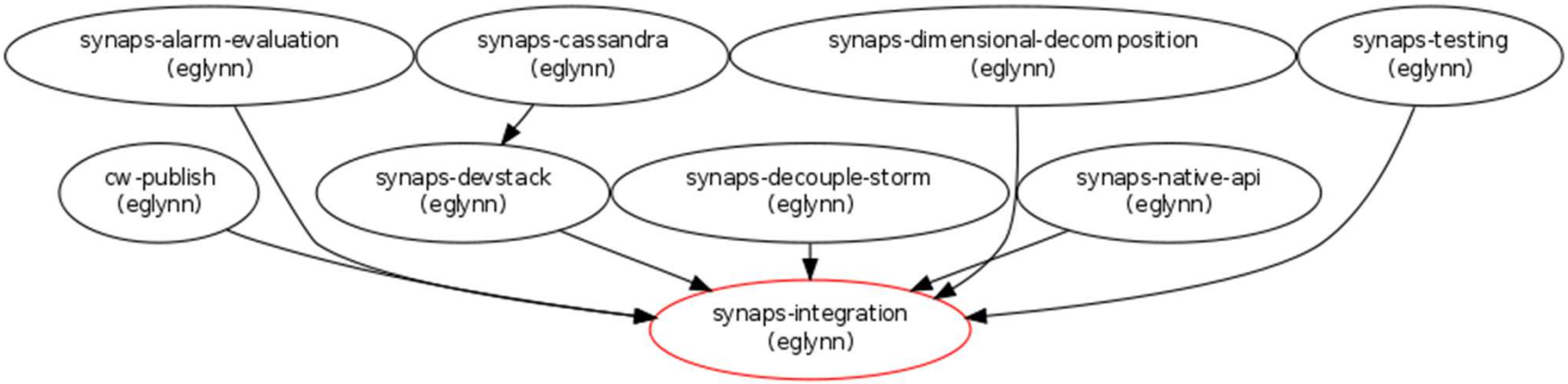
- こういうものが動く

```
public class KvmServerDiscoverer extends DiscovererBase implements Discoverer,
                                Listener, ResourceStateAdapter {
(略)
                @Override
                public Map<? extends ServerResource, Map<String, String>> find(long dcId,
                                                Long podId, Long clusterId, URI uri, String username,
                                                String password, List<String> hostTags) throws DiscoveryException {
```

  - KvmServerDiscoverer は components.xml に書いてある

OpenStack Ceilometer

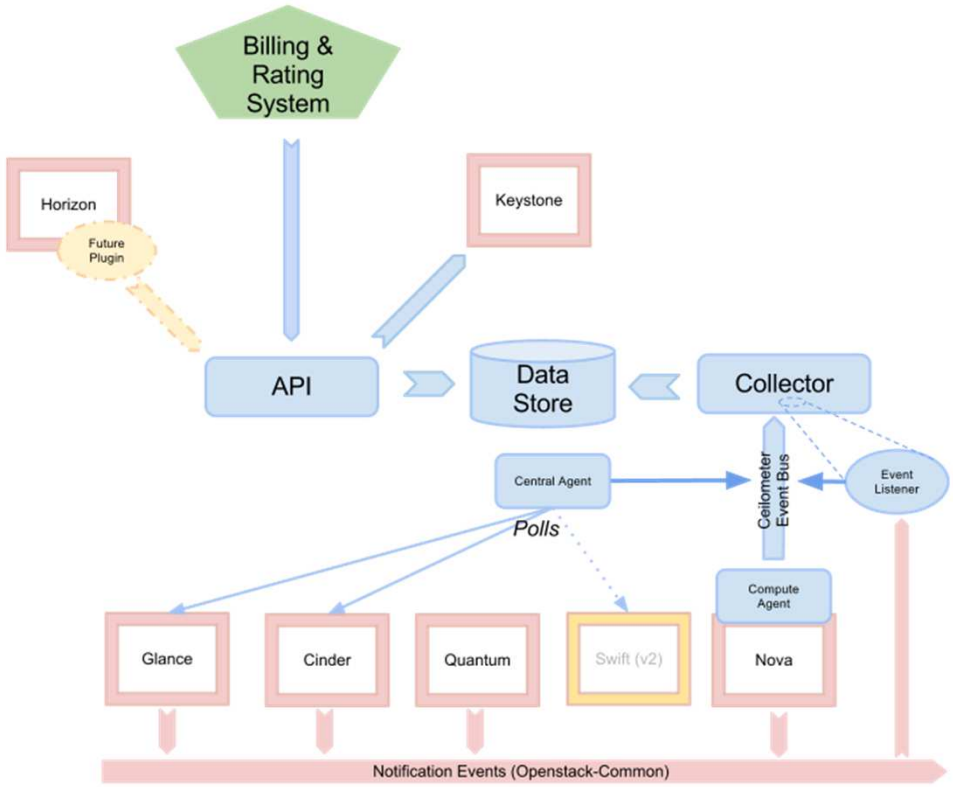
- Incubated Project (Coreの前段階)
- OpenStackでの測定全てを担当するのが目標
  - 現状はそうでもない
- 関連プロジェクト
  - Synaps
  - CloudWatch互換機能をCeilometerに統合予定
  - Samsung製
  - Ganglia (OpenStackとは直接関係ない)



出典：https://blueprints.launchpad.net/ceilometer/+spec/synaps-integration

Ceilometer構成

- Compute-agentがpollingしてデータをとる
- Collectorがメッセージを監視してDBに入力
- 利用者はAPI経由でDBにアクセス
- DBデフォルトはmongodb



測れるもの

```
$ ceilometer meter-list
INFO Starting new HTTP connection (1): 172.17.88.20
```

Name	Type	Resource ID	User ID	Project ID
cpu	cumulative	0d4d7d34-e206-418a-b76d-412c6e2b0aa8	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
cpu_util	gauge	0d4d7d34-e206-418a-b76d-412c6e2b0aa8	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
disk.read.bytes	cumulative	0d4d7d34-e206-418a-b76d-412c6e2b0aa8	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
disk.read.requests	cumulative	0d4d7d34-e206-418a-b76d-412c6e2b0aa8	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
disk.write.bytes	cumulative	0d4d7d34-e206-418a-b76d-412c6e2b0aa8	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
disk.write.requests	cumulative	0d4d7d34-e206-418a-b76d-412c6e2b0aa8	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
instance	gauge	0d4d7d34-e206-418a-b76d-412c6e2b0aa8	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
instance:t1.micro	gauge	0d4d7d34-e206-418a-b76d-412c6e2b0aa8	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
network	gauge	f684acad-0746-499b-9c0f-843ce3a3f035	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
network.create	delta	f684acad-0746-499b-9c0f-843ce3a3f035	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
port	gauge	574314bd-7d3a-4c05-b5e1-3a71b6290810	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
port.create	delta	574314bd-7d3a-4c05-b5e1-3a71b6290810	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
subnet	gauge	1743d07f-ddff-4e74-a7f3-641d11e957ae	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc
subnet.create	delta	1743d07f-ddff-4e74-a7f3-641d11e957ae	6d13ed3e567b440f9a14b68daba4a7b5	ce8c7370af824696b2c61a4bc89c5acc

- タイプは積算、差分、ゲージの3種類
- 測れるものは設定によってかわる
  http://docs.openstack.org/developer/ceilometer/measurements.html

測定データ

```
$ ceilometer sample-list -r  0d4d7d34-e206-418a-b76d-412c6e2b0aa8 -c cpu
INFO Starting new HTTP connection (1): 172.17.88.20
```

Resource ID	Name	Type	Volume	Timestamp
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	21660000000	2013-01-28T02:23:23.000000
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	48820000000	2013-01-28T02:33:25.000000
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	73670000000	2013-01-28T02:43:28.000000
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	98620000000	2013-01-28T02:53:30.000000
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	123590000000	2013-01-28T03:03:33.000000
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	148770000000	2013-01-28T03:13:35.000000
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	173740000000	2013-01-28T03:23:38.000000
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	198810000000	2013-01-28T03:33:40.000000
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	223830000000	2013-01-28T03:43:43.000000
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	248930000000	2013-01-28T03:53:45.000000
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	273960000000	2013-01-28T04:03:48.000000
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	298910000000	2013-01-28T04:13:50.000000
0d4d7d34-e206-418a-b76d-412c6e2b0aa8	cpu	cumulative	323860000000	2013-01-28T04:23:53.000000

Ceilometer on devstack

- 手順
  - ENABLED_SERVICESにceilometer-acompute,ceilometer-acentral,ceilometer-collector,ceilometer-api を足す
  - EXTRA_OPTS の設定はいらないはず
  - (http://ceilometer.readthedocs.org/en/latest/install.html)

ceilometer-agent-compute

- novaの動くホスト上で起動されて、LibvirtでCPU時間などを収集する
- デフォルトのpoll間隔は10分
- ドメイン毎の測定値をAMQPに流す

AWS CloudWatch

- Metrics, Dimensions
- Alarmが設定できる(PutMetricAlarm)
  - 指定した値を越えるか下回ると発動する
  - AutoScalingはこれを使っている

OpenStack Heat

- Incubated Project (Coreの前段階)
- AWS CloudFormationと同じテンプレート(Json形式)のサポートが目標
- devstackで試せる(はず)
- 関連プロジェクト
  - Synaps
  - Ganglia

AWS CloudFormation

- Template (Json形式)にAWS資源を記述
- Templateを指定してCreateStackという操作を行うとAWS資源がつくられる
- Template例 (S3だけ)

```
{
    "Resources" : {
        "HelloBucket" : {
            "Type" : "AWS::S3::Bucket"
        }
    }
}
```

Template例(wordpress)

- https://s3.amazonaws.com/cloudformation-templates-us-east-1/WordPress_Single_Instance_With_RDS.template

```
{
  "Parameters" : {
(略)
  "Mappings" : {
(略)
  "Resources" : {
(略)
    "WebServer": {
     "Type": "AWS::EC2::Instance",
     "Metadata" : {
      "AWS::CloudFormation::Init" : {
       "config" : {
        "packages" : {
 (略)
        "sources" : {
         "/var/www/html" : "http://wordpress.org/latest.tar.gz"
        },
        "files" : {
         "/var/www/html/wordpress/wp-config.php" : {
          "content" : { "Fn::Join" : ["", [
           "<?php¥n",
           "define('DB_NAME',          '", {"Ref" : "DBName"}, "');¥n",
           "define('DB_USER',          '", {"Ref" : "DBUsername"}, "');¥n",
(略)
    "DBInstance" : {
     "Type": "AWS::RDS::DBInstance",
     "Properties": {
      "DBName"            : { "Ref" : "DBName" },
(略)
    "WebServerSecurityGroup" : {
     "Type" : "AWS::EC2::SecurityGroup",
     "Properties" : {
      "GroupDescription" : "Enable HTTP access via port 80 and SSH access",
(略)
}
```

Templateについて

- Json形式
- トップレベルに書けるのは以下の6つ
  - AWSTemplateFormatVersion,Description, Parameters, Mappings, Resources, Outputs
- AWSのResourceにはEC2 Instance, データベース, AutoScalingなどがある
- Resourceに対応するPropertyを記述
- HeatのRoadmapでは、AWS VPC (Virtual Private Cloud) の優先度が高

HeatでのTemplate対応状況

- まだまだこれから(AutoScalingも)
- http://wiki.openstack.org/Heat/AWS-template-support-parity

(Parameters)

Property	Value	Status
Type	String	Implemented
	Number	Implemented
	CommaDelimitedList	Implemented
Default		Implemented
NoEcho		**Required**
AllowedValues		Implemented
AllowedPattern		Implemented
MaxLength		Implemented
MinLength		Implemented
MaxValue		Implemented
MinValue		Implemented
Description		Implemented
ConstraintDescription		Implemented

Resource	Property	Status
AWS::AutoScaling::AutoScalingGroup		*In Progess*
AWS::AutoScaling::LaunchConfiguration		*In Progress*
AWS::AutoScaling::ScalingPolicy		*In Progress*
AWS::AutoScaling::Trigger		**Required**
AWS::CloudFormation::Authentication		"Partial"
AWS::CloudFormation::Init		Implemented
AWS::CloudFormation::Stack		Implemented
AWS::CloudFormation::WaitCondition		Implemented
AWS::CloudFormation::WaitConditionHandle		Implemented
AWS::CloudFront::Distribution		Out of Scope
AWS::CloudWatch::Alarm		Implemented
AWS::EC2::Volume		Implemented
	AvailabilityZone	Implemented
	Size	Implemented
	SnapshotId	**Required**

HeatのCloudWatchとCeilometer

- Alarm機能にceilometerを使うようにしたいという話はある
  https://blueprints.launchpad.net/heat/+spec/watch-ceilometer
- まだApproveされてない