**OpenStack**

## 一、基础知识

**1 Openstack简介**

• OpenStack是一个由NASA(美国国家航空航天局)和Rackspace合作研发并发起的项目

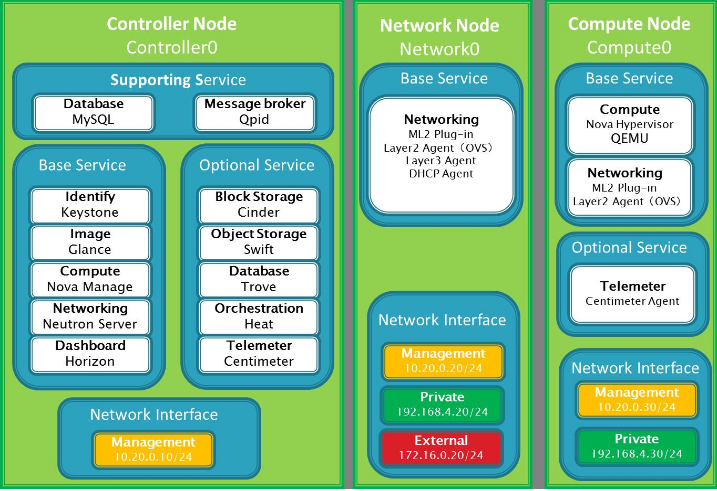
• OpenStack是一套IaaS解决方案

• OpenStack是一个开源的云计算管理平台

• 以Apache许可证为授权

OpenStack是一个美国国家航空航天局和Rackspace合作研发的云计算软件，以Apache授权条款授权，并且是一个自由软件和开放源代码项目。OpenStack是一个云操作系统，通过数据中心可控制大型的计算、存储、网络等资源池。所有的管理通过前端界面管理员就可以完成，同样也可以通过web接口让最终用户部署资源。OpenStack包含了一组由社区维护的开源项目，主要项目有Compute（Nova）， Object Storage（Swift），Image Service（Glance）。Nova提供虚拟计算服务，Swift提供存储服务，Glance提供虚拟机镜像的注册、分发服务。

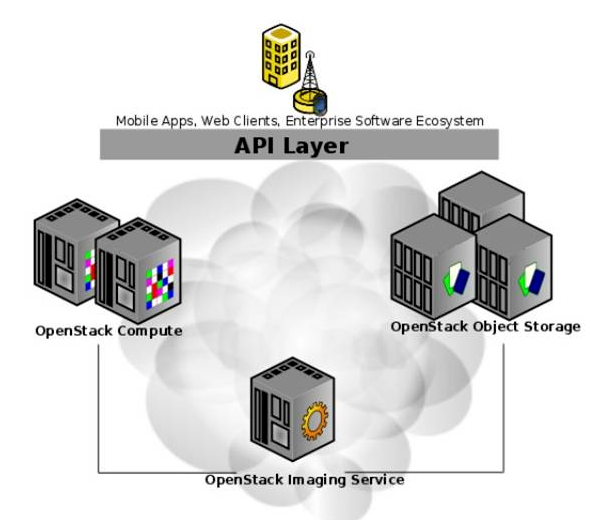
OpenStack是一整套开源软件项目的综合，它允许企业或服务提供者建立、运行自己的云计算和存储设施。Rackspace与NASA是最初重要的两个贡献者，前者提供了“云文件”平台代码，该平台增强了OpenStack对象存储部分的功能，而后者带来了“Nebula”平台形成了OpenStack其余的部分。而今，OpenStack基金会已经有150多个会员，包括很多知名公司如“Canonical、DELL、Citrix”等。



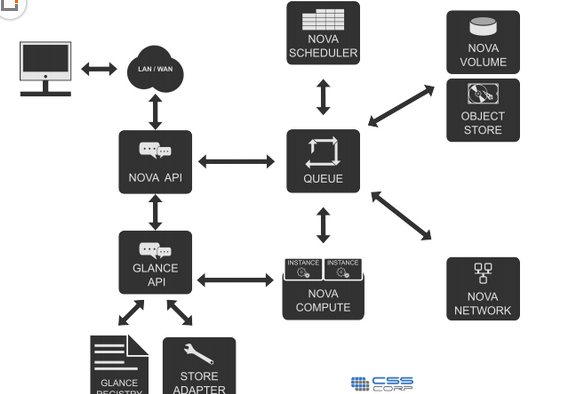
也是一个项目和一个开源软件，提供开放源码软件，建立公共和私有云，它提供了一个部署云的操作平台或工具集，其宗旨在于：帮助组织运行为虚拟计算或存储服务的云，为公有云、私有云，也为大云、小云提供可扩展的、灵活的云计算。

OpenStackd开源项目由社区维护，包括OpenStack计算（代号为Nova），OpenStack对象存储（代号为Swift），并OpenStack镜像服务（代号Glance）的集合。 OpenStack提供了一个操作平台，或工具包，用于编排云。

OpenStack提供开放源码软件，建立公共和私有云。 OpenStack是一个社区和一个项目，以及开放源码软件，以帮助企业运行的虚拟计算或者存储云。 OpenStackd开源项目由社区维护，包括OpenStack计算（代号为Nova），OpenStack对象存储（代号为SWIFT），并OpenStack镜像服务（代号Glance）的集合。 OpenStack提供了一个操作平台，或工具包，用于编排云。



**2 Openstack主要组件**



**• Horizon**

– 用于管理Openstack各种服务的、基于web的管理接口

– 通过图形界面实现创建用户、管理网络、启动实例等操作

**• Keystone**

– 为其他服务提供认证和授权的集中身份管理服务

– 也提供了集中的目录服务

– 支持多种身份认证模式,如果密码认证、令牌认证、以及AWS(亚马逊Web服务)登陆

– 为用户和其他服务提供了SSO认证服务

**• Neutron**

– 一种软件定义网络服务

– 用于创建网络、子网、路由器、管理浮动IP地址

– 可以实现虚拟交换机、虚拟路由器

– 可用于在项目中创建VPN

**• Cinder**

– 为虚拟机管理存储卷的服务

– 为运行在Nova中的实例提供永久的块存储

– 可以通过快照进行数据备份

– 经常应用在实例存储环境中,如果数据库文件

**• Nova**

– 在节点上用于管理虚拟机的服务

– Nova是一个分布式的服务,能够与Keystone交互实现认证,与Glance交互实现镜像管理

– Nova被设计成在标准硬件上能够进行水平扩展

– 启动实例时,如果有需要则下载镜像

**• Glance**

– 扮演虚拟机镜像注册的角色

– 允许用户为直接存储拷贝服务器镜像

– 这些镜像可以用于新建虚拟机的模板

**3 Openstack的网络拓扑结构图**

整个OpenStack是由控制节点，计算节点，网络节点，存储节点四大部分组成。（这四个节点也可以安装在一台机器上，单机部署）

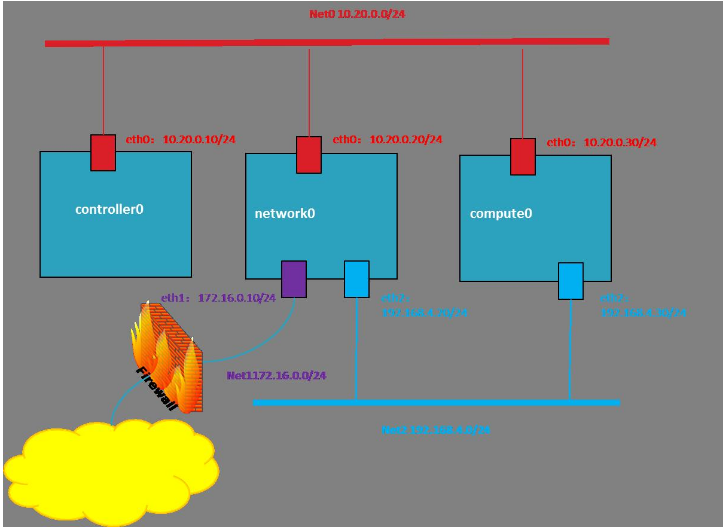
其中：

控制节点负责对其余节点的控制，包含虚拟机建立，迁移，网络分配，存储分配等等

计算节点负责虚拟机运行

网络节点负责对外网络与内网络之间的通信

存储节点负责对虚拟机的额外存储管理等等



OpenStack当前主要有三个组件：计算，存储，镜像。

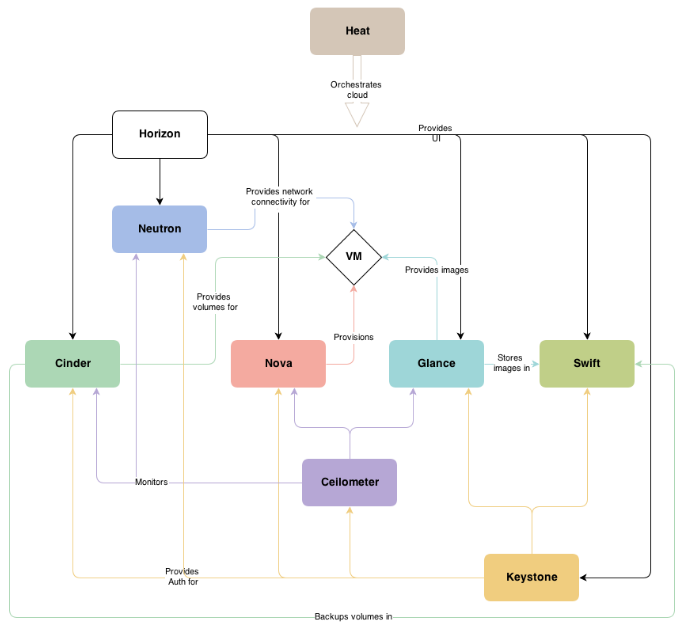
OpenStack计算是一个云控制器，用来启动一个用户或一个组的虚拟实例，它也用于配置每个实例或项目中包含多个实例为某个特定项目的联网。

OpenStack对象存储是一个在具有内置冗余和容错的大容量系统中存储对象的系统。对象存储有各种应用，如备份或存档数据，存储图形或视频（流媒体数据传输到用户的浏览器），储存二级或三级静态数据，发展与数据存储集成新的应用程序，当预测存储容量困难时存储数据，创造弹性和灵活的云存储Web应用程序。

OpenStack镜像服务是一个查找和虚拟机图像检索系统。它可以配置三种方式：使用OpenStack对象存储来存储图像;使用亚马逊S3直接存储，或使用S3对象存储作为S3访问中间存储。

* **终端用户通过nova-api 接口与Openstack 计算交互。**
* **OpenStack计算守护进程通过队列的交换信息（行动）和数据库（信息）进行API请求。**
* **OpenStack Glance是一个完全独立的基础上设施。**

## 核心技术



**1 OpenStack计算设施----Nova**

Nova是OpenStack计算的弹性控制器。OpenStack云实例生命期所需的各种动作都将由Nova进行处理和支撑，这就意味着Nova以管理平台的身份登场，负责管理整个云的计算资源、网络、授权及测度。虽然Nova本身并不提供任何虚拟能力，但是它将使用libvirt API与虚拟机的宿主机进行交互。Nova通过Web服务API来对外提供处理接口，而且这些接口与Amazon的Web服务接口是兼容的。

**（1）功能及特点**

实例生命周期管理

计算资源管理

网络与授权管理

基于REST的API

异步连续通信

支持各种宿主：Xen、XenServer/XCP、KVM、UML、VMware vSphere及Hyper-V

**（2）OpenStack计算部件**

Nova弹性云包含以下主要部分：

API Server（nova-api）

消息队列（rabbit-mq server）

运算工作站（nova-compute）

网络控制器（nova-network）

卷管理（nova-volume）

调度器（nova-scheduler）

**（3）API服务器（nova-api）**

API服务器提供了云设施与外界交互的接口，它是外界用户对云实施管理的唯一通道。通过使用web服务来调用各种EC2的API，接着API服务器便通过消息队列把请求送达至云内目标设施进行处理。作为对EC2-api的替代，用户也可以使用OpenStack的原生API，我们把它叫做“OpenStack API”。

**（4）消息队列（Rabbit MQ Server）**

OpenStack内部在遵循AMQP（高级消息队列协议）的基础上采用消息队列进行通信。Nova对请求应答进行异步调用，当请求接收后便则立即触发一个回调。由于使用了异步通信，不会有用户的动作被长置于等待状态。例如，启动一个实例或上传一份镜像的过程较为耗时，API调用就将等待返回结果而不影响其它操作，在此异步通信起到了很大作用，使整个系统变得更加高效。

**（5）运算工作站（nova-compute）**

运算工作站的主要任务是管理实例的整个生命周期。他们通过消息队列接收请求并执行，从而对实例进行各种操作。在典型实际生产环境下，会架设许多运算工作站，根据调度算法，一个实例可以在可用的任意一台运算工作站上部署。

**（6）网络控制器（nova-network）**

网络控制器处理主机的网络配置，例如IP地址分配，配置项目VLAN，设定安全群组以及为计算节点配置网络。

**（7）卷工作站（nova-volume）**

卷工作站管理基于LVM的实例卷，它能够为一个实例创建、删除、附加卷，也可以从一个实例中分离卷。卷管理为何如此重要？因为它提供了一种保持实例持续存储的手段，比如当结束一个实例后，根分区如果是非持续化的，那么对其的任何改变都将丢失。可是，如果从一个实例中将卷分离出来，或者为这个实例附加上卷的话，即使实例被关闭，数据仍然保存其中。这些数据可以通过将卷附加到原实例或其他实例的方式而重新访问。因此，为了日后访问，重要数据务必要写入卷中。这种应用对于数据服务器实例的存储而言，尤为重要。

**（8）调度器（nova-scheduler）**

调度器负责把nova-API调用送达给目标。调度器以名为“nova-schedule”的守护进程方式运行，并根据调度算法从可用资源池中恰当地选择运算服务器。有很多因素都可以影响调度结果，比如负载、内存、子节点的远近、CPU架构等等。强大的是nova调度器采用的是可插入式架构。目前nova调度器使用了几种基本的调度算法：

随机化：主机随机选择可用节点；

可用化：与随机相似，只是随机选择的范围被指定；

简单化：应用这种方式，主机选择负载最小者来运行实例。负载数据可以从别处获得，如负载均衡服务器。

**2 OpenStack镜像服务器----Glance**

OpenStack镜像服务器是一套虚拟机镜像发现、注册、检索系统，我们可以将镜像存储到以下任意一种存储中：

本地文件系统（默认）

OpenStack对象存储

S3直接存储

S3对象存储（作为S3访问的中间渠道）

HTTP（只读）

**（1）功能及特点**

提供镜像相关服务

**（2）Glance构件**

Glance控制器

Glance注册器

**3 OpenStack存储设施----Swift**

Swift为OpenStack提供一种分布式、持续虚拟对象存储，它类似于Amazon Web Service的S3简单存储服务。Swift具有跨节点百级对象的存储能力。Swift内建冗余和失效备援管理，也能够处理归档和媒体流，特别是对大数据（千兆字节）和大容量（多对象数量）的测度非常高效。

**（1）功能及特点**

海量对象存储

大文件（对象）存储

数据冗余管理

归档能力-----处理大数据集

为虚拟机和云应用提供数据容器

处理流媒体

对象安全存储

备份与归档

良好的可伸缩性

**（2）Swift组件**

Swift账户

Swift容器

Swift对象

Swift代理

Swift RING

**（3）Swift代理服务器**

用户都是通过Swift-API与代理服务器进行交互，代理服务器正是接收外界请求的门卫，它检测合法的实体位置并路由它们的请求。

此外，代理服务器也同时处理实体失效而转移时，故障切换的实体重复路由请求。

**（4）Swift对象服务器**

对象服务器是一种二进

制存储，它负责处理本地存储中的对象数据的存储、检索和删除。对象都是文件系统中存放的典型的二进制文件，具有扩展文件属性的元数据（xattr）。

注意：xattr格式被Linux中的ext3/4，XFS，Btrfs，JFS和ReiserFS所支持，但是并没有有效测试证明在XFS，JFS，ReiserFS，Reiser4和ZFS下也同样能运行良好。不过，XFS被认为是当前最好的选择。

**（5）Swift容器服务器**

容器服务器将列出一个容器中的所有对象，默认对象列表将存储为SQLite文件（译者注：也可以修改为MySQL，安装中就是以MySQL为例）。容器服务器也会统计容器中包含的对象数量及容器的存储空间耗费。

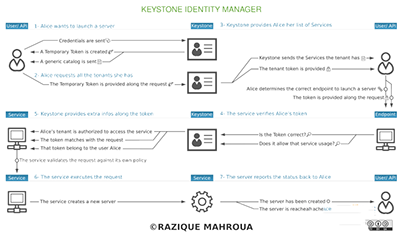
**（6）Swift账户服务器**

账户服务器与容器服务器类似，将列出容器中的对象。

**（7）Ring（索引环）**

Ring容器记录着Swift中物理存储对象的位置信息，它是真实物理存储位置的实体名的虚拟映射，类似于查找及定位不同集群的实体真实物理位置的索引服务。这里所谓的实体指账户、容器、对象，它们都拥有属于自己的不同的Rings。

**4 OpenStack认证服务（Keystone）**



**（1）Keystone为所有的OpenStack组件提供认证和访问策略服务**，它依赖自身REST（基于Identity API）系统进行工作，主要对（但不限于）Swift、Glance、Nova等进行认证与授权。事实上，授权通过对动作消息来源者请求的合法性进行鉴定。如下图所示：

Keystone采用两种授权方式，一种基于用户名/密码，另一种基于令牌（Token）。除此之外，Keystone提供以下三种服务：

令牌服务：含有授权用户的授权信息

目录服务：含有用户合法操作的可用服务列表

策略服务：利用Keystone具体指定用户或群组某些访问权限

**（2）认证服务组件**

服务入口：如Nova、Swift和Glance一样每个OpenStack服务都拥有一个指定的端口和专属的URL，我们称其为入口（endpoints）。

区位：在某个数据中心，一个区位具体指定了一处物理位置。在典型的云架构中，如果不是所有的服务都访问分布式数据中心或服务器的话，则也称其为区位。

用户：Keystone授权使用者

译者注：代表一个个体，OpenStack以用户的形式来授权服务给它们。用户拥有证书（credentials），且可能分配给一个或多个租户。经过验证后，会为每个单独的租户提供一个特定的令牌。[来源：//www.jb51.net/article/96989.htm]

服务：总体而言，任何通过Keystone进行连接或管理的组件都被称为服务。举个例子，我们可以称Glance为Keystone的服务。

角色：为了维护安全限定，就云内特定用户可执行的操作而言，该用户关联的角色是非常重要的。

译者注：一个角色是应用于某个租户的使用权限集合，以允许某个指定用户访问或使用特定操作。角色是使用权限的逻辑分组，它使得通用的权限可以简单地分组并绑定到与某个指定租户相关的用户。

租间：租间指的是具有全部服务入口并配有特定成员角色的一个项目。

译者注：一个租间映射到一个Nova的“project-id”，在对象存储中，一个租间可以有多个容器。根据不同的安装方式，一个租间可以代表一个客户、帐号、组织或项目。

**5 OpenStack管理的Web接口----Horizon**

Horizon是一个用以管理、控制OpenStack服务的Web控制面板，它可以管理实例、镜像、创建密匙对，对实例添加卷、操作Swift容器等。除此之外，用户还可以在控制面板中使用终端（console）或VNC直接访问实例。总之，Horizon具有如下一些特点：

实例管理：创建、终止实例，查看终端日志，VNC连接，添加卷等

访问与安全管理：创建安全群组，管理密匙对，设置浮动IP等

偏好设定：对虚拟硬件模板可以进行不同偏好设定

镜像管理：编辑或删除镜像

查看服务目录

管理用户、配额及项目用途

用户管理：创建用户等

卷管理：创建卷和快照

对象存储处理：创建、删除容器和对象

为项目下载环境变量

**6 Openstack案例**

对于OpenStack的具体安装，可以查看“五岳之巅”的博客 http://blog.chinaunix.net/uid/22414998/frmd/154248.html 讲解较为详细。

以下内容是如何在linux命令行下进行nova-compute实例的相关操作，这些操作均可以通过OpenStack的Horizon服务在web界面上完成，本文只是作者笔记，供参考用。

客户端系统环境：

ubuntu-desktop-12.04

python-novaclient glance-client swift kvm

export SERVICE\_TOKEN=admin

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=admin

export OS\_AUTH\_URL="http://192.168.111.128:5000/v2.0/"

export SERVICE\_ENDPOINT=http://192.168.111.128:35357/v2.0

以上是客户端主机系统的基本要求，本文中由于Nova是安装在192.168.111.128上面的，所以读者应根据实际情况设置。

在拥有以上的环境后，可以通过命令操作来查看、创建、管理实例。

查看Image的信息：

root@client:~# nova image-list

+--------------------------------------+--------------+--------+--------+

| ID | Name | Status | Server |

+--------------------------------------+--------------+--------+--------+

| 50a278f9-54e2-485c-9592-f0e485689df0 | ubuntu-12.04 | ACTIVE | |

+--------------------------------------+--------------+--------+--------+

root@client:~# nova image-list

+--------------------------------------+--------------+--------+--------+

| ID | Name | Status | Server |

+--------------------------------------+--------------+--------+--------+

| 50a278f9-54e2-485c-9592-f0e485689df0 | ubuntu-12.04 | ACTIVE | |

+--------------------------------------+--------------+--------+--------+

查看可用的kvm实例化配置信息：

root@client:~# nova flavor-list

+----+--------------+-----------+------+-----------+------+-------+-------------+

| ID | Name | Memory\_MB | Disk | Ephemeral | Swap | VCPUs | RXTX\_Factor |

+----+--------------+-----------+------+-----------+------+-------+-------------+

| 1 | m1.tiny | 512 | 0 | 0 | | 1 | 1.0 |

| 2 | m1.small | 2048 | 10 | 20 | | 1 | 1.0 |

| 3 | m1.medium | 4096 | 10 | 40 | | 2 | 1.0 |

| 4 | m1.large | 8192 | 10 | 80 | | 4 | 1.0 |

| 5 | m1.xlarge | 16384 | 10 | 160 | | 8 | 1.0 |

| 6 | svr\_2-512-10 | 512 | 10 | 20 | | 2 | 1.0 |

+----+--------------+-----------+------+-----------+------+-------+-------------+

启动一个实例：

root@client:~# nova boot --flavor=1 --image=50a278f9-54e2-485c-9592-f0e485689df0 testserver

+-------------------------------------+--------------------------------------+

| Property | Value |

+-------------------------------------+--------------------------------------+

| OS-DCF:diskConfig | MANUAL |

| OS-EXT-SRV-ATTR:host | None |

| OS-EXT-SRV-ATTR:hypervisor\_hostname | None |

| OS-EXT-SRV-ATTR:instance\_name | instance-00000004 |

| OS-EXT-STS:power\_state | 0 |

| OS-EXT-STS:task\_state | scheduling |

| OS-EXT-STS:vm\_state | building |

| accessIPv4 | |

| accessIPv6 | |

| adminPass | sJcaPJYp22Fc |

| config\_drive | |

| created | 2012-12-22T06:36:02Z |

| flavor | m1.tiny |

| hostId | |

| id | 6b92ce5a-5499-4b42-a346-13b78e28d3e8 |

| image | ubuntu-12.04 |

| key\_name | |

| metadata | {} |

| name | testserver |

| progress | 0 |

| status | BUILD |

| tenant\_id | dac9bcbdf94d4764b1fc919b126727ca |

| updated | 2012-12-22T06:36:02Z |

| user\_id | bb1468f01a8548cf8160836ca6a82679 |

+-------------------------------------+--------------------------------------+

该实例使用flavor中ID为1的资源配置，镜像使用ID为50a278f9-54e2-485c-9592-f0e485689df0 的镜像，实例的显示名称为testserver

启动完该实例后，直观的从web上看：

命令行查看实例情况：

root@client:~# nova list

+--------------------------------------+------------+--------+----------+

| ID | Name | Status | Networks |

+--------------------------------------+------------+--------+----------+

| 6b92ce5a-5499-4b42-a346-13b78e28d3e8 | testserver | ACTIVE | |

+--------------------------------------+------------+--------+----------+

查看实例的详细信息：

root@client:~# nova show 6b92ce5a-5499-4b42-a346-13b78e28d3e8

+-------------------------------------+----------------------------------------------------------+

| Property | Value |

+-------------------------------------+----------------------------------------------------------+

| OS-DCF:diskConfig | MANUAL |

| OS-EXT-SRV-ATTR:host | svr1 |

| OS-EXT-SRV-ATTR:hypervisor\_hostname | None |

| OS-EXT-SRV-ATTR:instance\_name | instance-00000004 |

| OS-EXT-STS:power\_state | 1 |

| OS-EXT-STS:task\_state | None |

| OS-EXT-STS:vm\_state | active |

| accessIPv4 | |

| accessIPv6 | |

| config\_drive | |

| created | 2012-12-22T06:36:02Z |

| flavor | m1.tiny |

| hostId | 0cb64f5bbe94b0a71f29d17025afe844d0fd047f8dae71d0142ebae1 |

| id | 6b92ce5a-5499-4b42-a346-13b78e28d3e8 |

| image | ubuntu-12.04 |

| key\_name | |

| metadata | {} |

| name | testserver |

| progress | 0 |

| status | ACTIVE |

| tenant\_id | dac9bcbdf94d4764b1fc919b126727ca |

| updated | 2012-12-22T06:36:05Z |

| user\_id | bb1468f01a8548cf8160836ca6a82679 |

+-------------------------------------+----------------------------------------------------------+