# 一．OpenStack的服务拓扑设计

OpenStack是一个分布式系统，由若干个节点构成，每个节点上可以部署OpenStack中的不同服务，同时每个服务的不同组件也可以分布式地部署在不同的节点上。

## （二）控制节点Controller

Controller节点上运行的服务有Keystone，Glance，Horizon以及Nova和Neutron中管理相关的组件。

还有支持OpenStack的服务，如Mysql，RabbitMQ和网络时间服务NTP。

## （二）网络节点Network

运行的服务是Neutron，为OpenStack提供L2和L3网络，包括虚拟机网络，DHCP，路由，NAT等。

## （三）存储节点Storage

提供块存储服务Cinder或对象存储服务Swift。

## （四）计算节点Compute

运行Hpervisor（默认使用KVM），该组件用于创建以及管理虚拟机，同时还运行着Neutron服务的agent代理，为虚拟机提供网络支持。

# 二．KeyStone服务

## （一）用户User

User指代任何使用OpenStack的实体，可以是真正的用户，其他系统。当User请求访问OpenStack时，Keystone会对其进行验证。除了admin和demo，OpenStack也为nova, cinder, glance ,neutron服务创建了相应的User，admin可以管理这些User。

## （二）凭证Credentials

Credentials是User用来证明自己身份的信息，可以是账户/密码，令牌Token等。

## （三）认证Authentication

Authentication是Keystone验证User身份的过程。

## （四）令牌Token

Token是由数字和字母组成的字符串，User成功Authentication之后，它由KeyStone分配给User，Token就会作为访问服务的Credential，服务会向Keystone验证Token的有效性，Token的默认有限期是24H。

## （五）项目Project

Project用于将OpenStack的资源（计算，存储和网络）进行分组和隔离。根据OpenStack服务的对象不同，Project可以是一个客户（公有云中的租户），部门。

注意，资源的所有权是属于Project的，而不是User，每个User（包括admin）都必须挂在Project里才能够访问该Project的资源，一个User可以属于多个Project。

## （六）地址Endpoint

Endpoint 是一个网络上可以访问的URL，Service通过Endpoint暴露自己的API，Keystone负责管理和维护每个Service的Endpoint。

## （七）角色Role

安全包含两部分：Authentication（认证）和Authrization（鉴权）。

Keystone通过Role来实现鉴权的，在Keystone中定义不同的Role，每个Role可能有不同的权限，可以为User分配多个Role。

每个Service决定每个Role可以做什么事情，通过各自的policy.json文件对Role进行访问控制。

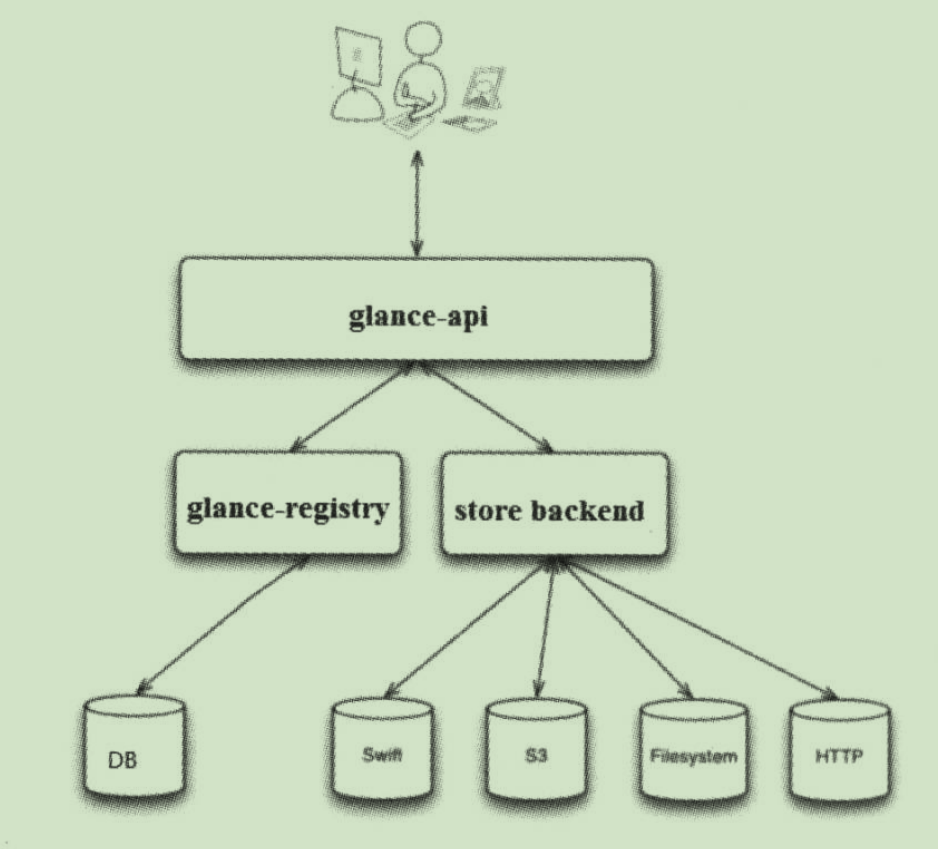
# 三．Glance服务

## （一）Glance服务的功能

1. 提供REST API，让用户能够查询和获取Image的元数据和Image本身；
2. 支持多种方式存储Image，包括普通的文件系统，Swift，Amazon S3；
3. 对实例Instance执行快照snapshot可以创建新的Image；

## （二）Glance服务组件架构

Glance的架构如下所示：



可以看到Glance由三部分组成：

### 1.glance-api

后台运行的服务进程，对外提供Rest API，响应Image查询，获取和存储的调用，它不会真真的处理请求。

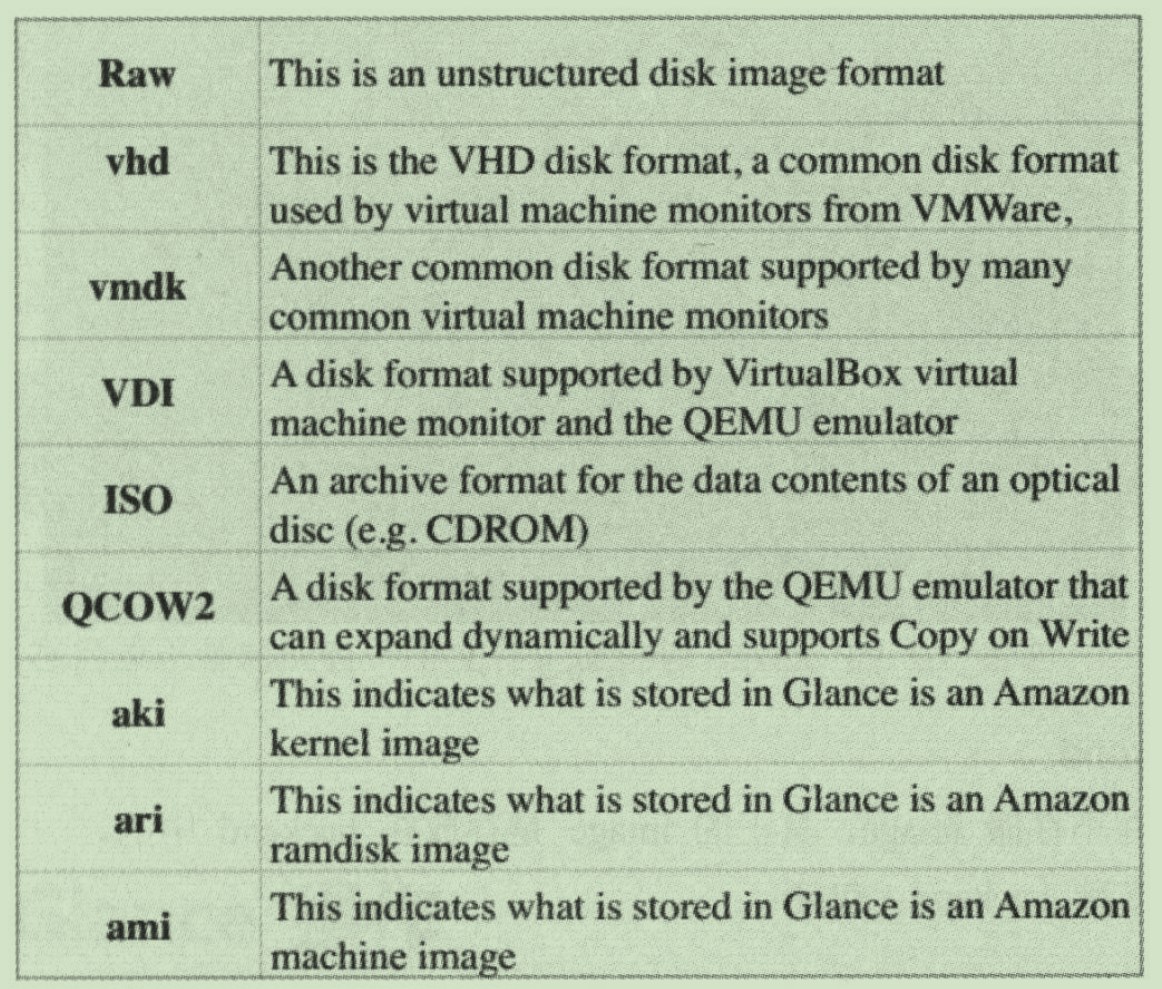
如果是和Image metadata（元数据）相关的操作，glance-api会将请求转发给glance-registry，再和数据库进行交互取出相应的数据；

如果是和Image自身存取相关的操作，glance-api会将请求转发给Image的存储系统store backend；

### 2.glance-registry

系统后台运行的服务进程，负责处理和存取Image的metadata（元数据），如Image的大小和类型，都会存储在database数据库中，默认是Mysql。

Glance支持多种类型的Image。如下所示：



### 3.store backend

Glance服务自己不会存储Image的，而是存储在backend中，Glance支持多种类型的backedn，如：默认本地文件系统，Amazon S3，Cinder，Swift，VMware ESX。

具体使用哪种backend，是在/etc/glance/glance-api.conf中配置的。

## （三）Glance的操作

OpenStack提供了Web UI（Horizon）和命令行CLI两种交互界面，命令行支持的功能更多，且参数更多，执行速度更快，且CLI可以放在脚本中批处理，同时对于耗时的操作CLI更加适合。

### 1.创建Image

如果设置Image为public，则其他Project可以使用该Image；如果设置为Protected，则该Image不允许被删除。建议使用CLI命令行操作，还可以显示创建的状态。

glance image-create –name cirrors—file /tmp/xxxx.img –disk-format qcow2 –container-format bare –progress

最后一个参数—progress可以显示Image文件上传的百分比，更加直观。

### 2.删除Image

glance image-delete xxxx

后面的是Image的ID。

## （四）Glance的日志

OpenStack排查问题的方法主要是通过日志，每个Service都有自己的日志。Glance的日志有两个，glanceapi.log和glanceregistry.log。glace-api日志记录了Rest API的调用情况，glance-registry日志记录了Glance服务处理请求的过程以及数据库的操作。如果需要得到更加详细的日志，可以在/etc/glance/\*.conf中打开debug选项。

## OpenStack的命令行操作

### 1.执行命令之前，需要设置环境变量

这些变量包含用户名，Project，密码等，如果不设置，那么每次执行命令时都必须设置相关的命令行参数。

### 2.各个服务的命令都有增删改查的操作

格式是：

CMD <obj>-create [param1][param2]…

CMD <obj>-delete [param1][param2]…

CMD <obj>-update [param1][param2]…

CMD <obj>-list [param1][param2]…

CMD <obj>-show [param1][param2]…

如glance中管理的是image，那么CMD就是glance，obj就是image；

netron中管理的是网络和子网，那么CMD就是neutron，obj就是net和subnet；

nova的obj可以省略，如针对instance的操作：

nova boot , nova delete, nova list, nova show;

3.help查看命令的用法

格式是：

CMD help [sub-CMD]

如glance help image-update可以查询image-update的用法。