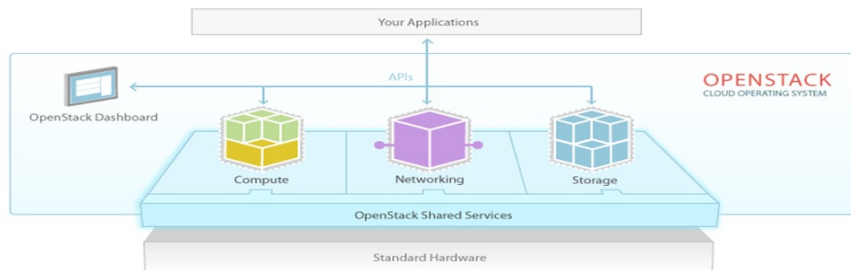


组网及说明

CloudOS正常组网

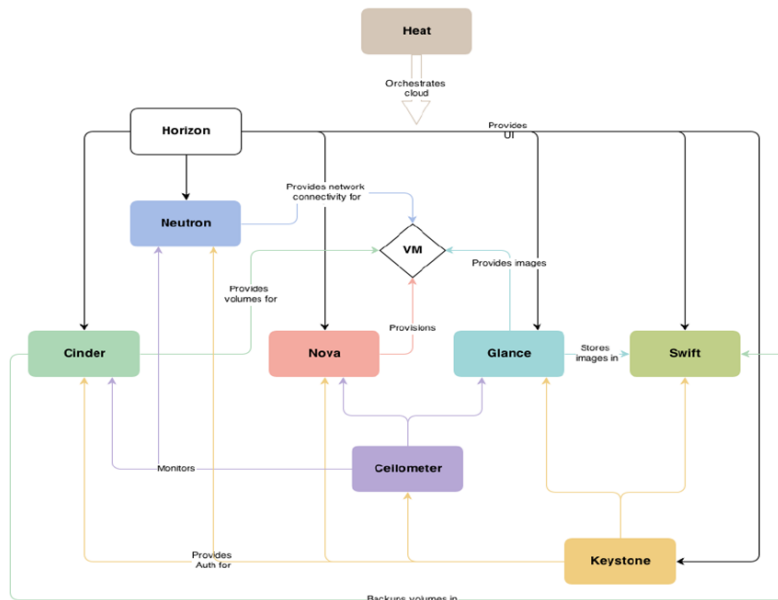
配置步骤

一、OpenStack的基本架构：



OpenStack拥有众多的组件，通过组件之间协同进行工作，管理计算、存储、网络等资源，便于用户调用和使用。

OpenStack的每个组件都有自己的功能定位，都可以算作独立的一个程序，Open即为开放之意，Stack则是堆砌，也就是许多开放的程序的集合。



OpenStack的核心部件即包括Nova（用于计算）、Keystone（用于身份服务）、Neutron（用于网络和地址管理）、Cinder（块存储）、Swift（对象存储）、Glance（镜像服务）、Horizon（UI界面）、Heat（用于部署编排）和Ceilometer（监控测量）。其中最核心的就是计算基础架构Nova、存储基础架构Swift和镜像服务Glance。

Nova是OpenStack云计算架构控制器，管理OpenStack云里的计算资源、网络、授权、和扩展，负责虚拟机创建、开机、关机、挂起、暂停、调整、迁移、重启、销毁等操作，配置CPU、内存等信息规格。

Swift为OpenStack提供分布式的虚拟对象存储，通过分布式的穿过节点，Swift有能力存储数十亿计的对象，Swift具有内置冗余、容错管理、存档、流媒体的功能，并且高度扩展，不论大小（多个PB级别）和能力（对象的数量）。

Glance镜像服务用于查找和检索虚拟机的镜像系统。

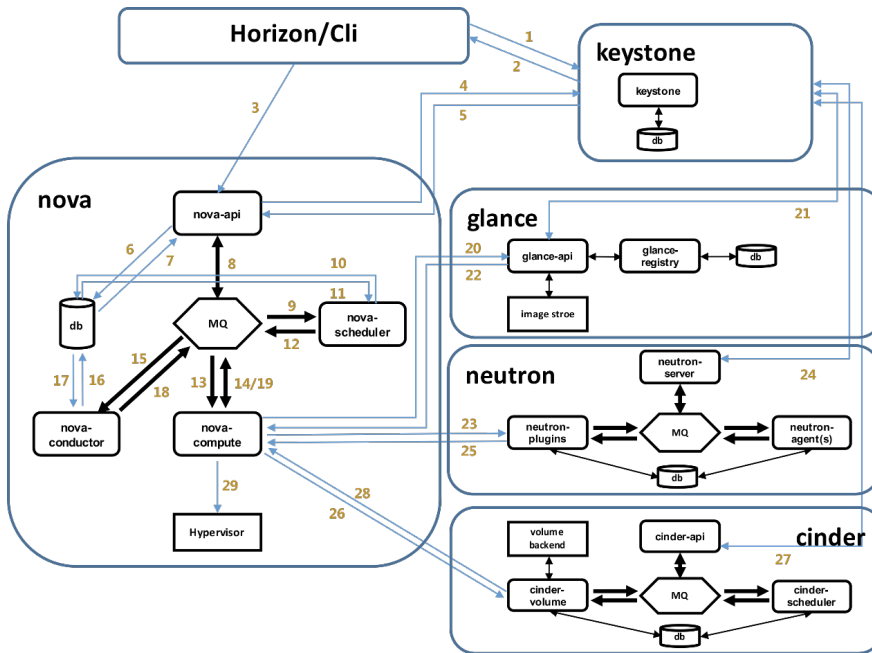
Horizon(Dashboard)是OpenStack中各种服务的Web管理门户，与其他主要模块相连接，用于简化操作，便于用户管理与控制各个模块。通过Ceilometer可以监控的模块包括Nova、Glance、Cinder和Neutron，可以收集记录各种信息，为其他服务提供数据支撑。Keystone模块可以对其他模块进行身份及权限验证，包括Nova、Glance、Cinder、Swift、Neutron和Ceilometer。

对于虚拟机(Virtual Machine, VM)，Nova可以为其提供计算资源，Glance为其提供镜像，Cinder为其提供块存储资源，Neutron为其提供网络资源及网络连接。Cinder连接VM后所产生的数据可以备份到Swift对象存储中，Glance提供镜像可以保存在Swift对象存储中。

Open Stack的各个服务之间通过统一的REST风格的API调用，实现系统的松耦合。它内部组件的工作过程是一个有序的整体。诸如计算资源分配、控制调度、网络通信等都通过AMQP实现。Open Stack的上层用户是程序员、一般用户和Horizon界面等模块。这三者都是采用OpenStack各个组件提供的AP

接口进行交互，而它们之间则是通过AMQP进行互相调用，它们共同利用底层的虚拟资源为上层用户和程序提供云计算服务。

二、OpenStack创建虚拟机流程：



1. 创建流程

创建虚拟机是OpenStack最基本的且较为核心的功能，整个流程以Nova组件为核心，并与其他组件如Keystone、Glance、Neutron、Cinder等进行交互。下面为创建虚拟机过程的具体流程：

- 1、登录界面或命令行通过RESTful API向Keystone获取认证信息。
- 2、Keystone通过用户请求认证信息，并生成Auth-Token返回给对应的认证请求。
- 3、界面或命令行通过RESTful API向Nova-api发送一个boot instance的请求（携带Auth-Token）。
- 4、Nova-api接受请求后向Keystone发送认证请求，验证Token的有效性。
- 5、Keystone验证Token是否有效，如有效则返回有效的认证和对应的角色。
- 6、通过认证后，Nova-api和数据库（DataBase, db）进行通讯。
- 7、初始化新建虚拟机的数据库记录。
- 8、Nova-api通过rpc.call向Nova-scheduler请求是否有创建虚拟机的资源(Host ID)。通过消息队列（MQ）。
- 9、Nova-scheduler进程侦听消息队列，获取Nova-api的请求。
- 10、Nova-scheduler通过查询Nova数据库中计算资源的情况，并通过调度算法计算符合虚拟机创建需要的主机。
- 11、对于有符合虚拟机创建的主机，Nova-scheduler更新数据库中虚拟机对应的物理主机信息。
- 12、Nova-scheduler通过rpc.cast向Nova-compute发送对应的创建虚拟机请求的消息。
- 13、Nova-compute会从对应的消息队列中获取创建虚拟机请求的消息。
- 14、Nova-compute通过rpc.call向Nova-conductor请求获取虚拟机消息。
- 15、Nova-conductor从消息队列中拿到Nova-compute请求消息。
- 16、Nova-conductor根据消息查询虚拟机对应的信息。
- 17、Nova-conductor从数据库中获得虚拟机对应信息。
- 18、Nova-conductor把虚拟机信息通过消息的方式发送到消息队列中。
- 19、Nova-compute从对应的消息队列中获取虚拟机信息消息。
- 20、Nova-compute通过Keystone的RESTful API拿到认证的Token，并通过HTTP请求Glance-api获取创建虚拟机所需要镜像。
- 21、Glance-api向Keystone认证Token是否有效，并返回验证结果。
- 22、Token验证通过，Nova-compute获得虚拟机镜像信息(URL)。
- 23、Nova-compute通过Keystone的RESTful API拿到认证的Token，并通过HTTP请求Neutron-server获取创建虚拟机所需要的网络信息。
- 24、Neutron-server向Keystone认证Token是否有效，并返回验证结果。
- 25、Token验证通过，Nova-compute获得虚拟机网络信息。
- 26、Nova-compute通过Keystone的RESTful API拿到认证的Token，并通过HTTP请求Cinder-api获取创建虚拟机所需要的持久化存储信息。
- 27、Cinder-api向Keystone认证Token是否有效，并返回验证结果。
- 28、Token验证通过，Nova-compute获得虚拟机持久化存储信息。
- 29、Nova-compute根据instance的信息调用配置的虚拟化驱动来创建虚拟机。

2. OpenStack组件之间的通信

OpenStack各个组件之间大部分是通过调用其他组件的REST api方式进行通信。OpenStack的各个服务组件都有提供相应的API接口，如Nova-api、Glance-api等。使用这些API和扩展可以让用户在Open

Stack中实现通讯并进行一系列的操作，如启动服务主机、创建镜像、给主机和镜像设置元数据、创建容器和对象等。

Restful API从字面就可以知道，他是REST式的接口，REST指的是一组架构约束条件和原则，基于HTTP，提供了一个新的架构设计思路，满足这些约束条件和原则的应用程序或设计就是 RESTful。REST定义中把所有事物都定义为资源，每一个资源对应一个独有的ID，在OpenStack中每一个资源也对应一个UUID。RESTful API 就可以通过一套统一的接口为所有客户端提供web服务，实现前后端分离。Keystone中Token是一个字符串表示，作为访问资源的令牌，Token包含了在指定范围和有效期内可以被访问的资源。

Keystone中，Endpoint是一个可以通过网络来访问和定位某个OpenStack service的地址，通常是一个URL。比如，当Nova需要访问Glance服务去获取image 时，Nova通过访问Keystone拿到Glance的endpoint，然后通过访问该endpoint去获取Glance服务。我们可以通过Endpoint的region属性去定义多个region。Endpoint 该使用对象分为三类：

admin URL --给admin用户使用，Post: 35357

internal URL--OpenStack内部服务用来跟别的服务通信，Port: 5000

public URL --其它用户可以访问的地址，Post: 5000

创建完service后创建API EndPoint。在OpenStack中，每一个service都有三种end points: Admin、public、internal。Admin是用作管理用途的，如它能够修改user/tenant(project)；public是让客户调用的，比如可以部署在外网上让客户可以管理自己的云；internal是OpenStack内部调用的。三种endpoints 在网络上开放的权限一般也不同，Admin通常只能对内网开放，public通常可以对外网开放，internal通常只能对安装有OpenStack对服务的机器开放。

3.Nova、Cinder和Neutron

3.1 Nova主要组成

Nova-api、Nova-scheduler、Nova-compute、Nova-conductor。

Nova各组件功能：

Nova-api是一个交互接口，管理者可以通过这个接口来管理内部基础设施，也可以通过这个接口向用户提供服务。基于Web的管理也是通过这个接口，然后向消息队列发送消息，达到资源调度的功能。

Nova-compute是最核心的组件，实现虚拟机管理功能。通过消息队列接收请求，并承担操作工作。

Nova-scheduler是一个拥有把Nova-api调用映射为OpenStack功能的组件，会根据诸如CPU构架、可用域的物理距离、内存、负载等作出调度决定。

Nova-conductor负责数据库的访问权限控制，避免Nova-compute直接访问数据库。

3.2 Cinder主要组成

Cinder-api、Cinder-scheduler、Cinder-volume。

Cinder各组件功能：

Cinder-api是Cinder服务的endpoint，提供REST接口，负责处理client请求，并将RPC请求发送至Cinder-scheduler组件。

Cinder-scheduler负责Cinder请求调度，其核心部分就是scheduler_driver，作为scheduler manager的driver，负责Cinder-volume具体的调度处理，发送Cinder RPC请求到选择的Cinder-volume。

Cinder-volume负责具体的volume请求处理，由不同后端存储提供volume存储空间。

3.3 Neutron包含组件

Neutron-server、Neutron-plugin、Neutron-agent。

Neutron各组件功能：

Neutron-server可以理解为一个专门用来接收Neutron REST API调用的服务器，然后负责将不同的REST API分发到不同的Neutron-plugin上。

Neutron-plugin可以理解成不同网络功能实现的入口，Neutron-plugin接收Neutron-server分发过来的REST API，向Neutron database完成一些信息的注册，然后将具体要执行的业务操作和参数通知给自身对应的Neutron agent。

Neutron-agent可以理解成Neutron-plugin在设备上的代理，接收相应的Neutron-plugin通知的业务操作和参数，并转换为具体的设备级操作，以指导设备的动作。当设备本地发生问题时，Neutron-agent会将情况通知给Neutron-plugin。

配置关键点

Openstack创建虚拟机流程以Nova组件为核心，并与其他组件

如Keystone、Glance、Neutron、Cinder等进行交互，共同完成这一任务。流程中OpenStack各个组件之间大部分是通过调用其他组件的REST api方式进行通信。OpenStack的各个服务组件都有提供相应的API接口，如Nova-api、Glance-api等。从Horizon开始到keystone获取token，再到nova中进行虚拟机的创建流程，借用nova-compute来调用cinder、neutron、glance中的存储资源、网络资源和镜像。流程完成之后再由nova将请求传递到hypervisor，进行下一步的创建工作。

附件下载：OpenStack创建虚拟机流程.rar