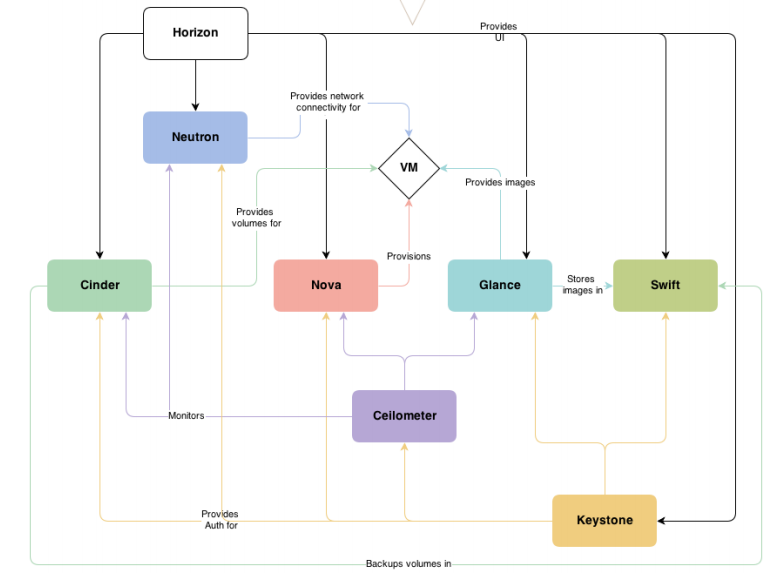
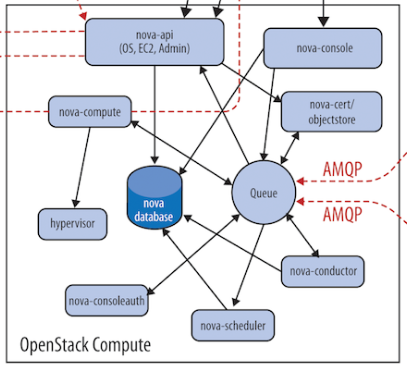


Compute Service Nova 是 OpenStack 最核心的服务，负责维护和管理云环境的计算资源。   
OpenStack 作为 IaaS 的云操作系统，虚拟机生命周期管理也就是通过 Nova 来实现的。



在上图中可以看到，Nova 处于 Openstak 架构的中心，其他组件都为 Nova 提供支持：  
Glance 为 VM 提供 image   
Cinder 和 Swift 分别为 VM 提供块存储和对象存储   
Neutron 为 VM 提供网络连接

Nova 架构如下



Nova 的架构比较复杂，包含很多组件。   
这些组件以子服务（后台 deamon 进程）的形式运行，可以分为以下几类：

**API**

**nova-api**  
接收和响应客户的 API 调用。

除了提供 OpenStack 自己的API，nova-api 还支持 Amazon EC2 API。

也就是说，如果客户以前使用 Amazon EC2，并且用 EC2 的 API 开发了些工具来管理虚机，那么如果现在要换成 OpenStack，这些工具可以无缝迁移到 OpenStack，因为 nova-api 兼容 EC2 API，无需做任何修改。

**Compute Core**

**nova-scheduler**虚机调度服务，负责决定在哪个计算节点上运行虚机

**nova-compute**管理虚机的核心服务，通过调用 Hypervisor API 实现虚机生命周期管理

**Hypervisor**计算节点上跑的虚拟化管理程序，虚机管理最底层的程序。

不同虚拟化技术提供自己的 Hypervisor。

常用的 Hypervisor 有 KVM，Xen， VMWare 等

**nova-conductor**nova-compute 经常需要更新数据库，比如更新虚机的状态。

出于安全性和伸缩性的考虑，nova-compute 并不会直接访问数据库，而是将这个任务委托给 nova-conductor，这个我们在后面会详细讨论。

**Console Interface**

**nova-console**用户可以通过多种方式访问虚机的控制台：

nova-novncproxy，基于 Web 浏览器的 VNC 访问

nova-spicehtml5proxy，基于 HTML5 浏览器的 SPICE 访问

nova-xvpnvncproxy，基于 Java 客户端的 VNC 访问

**nova-consoleauth**负责对访问虚机控制台请求提供 Token 认证

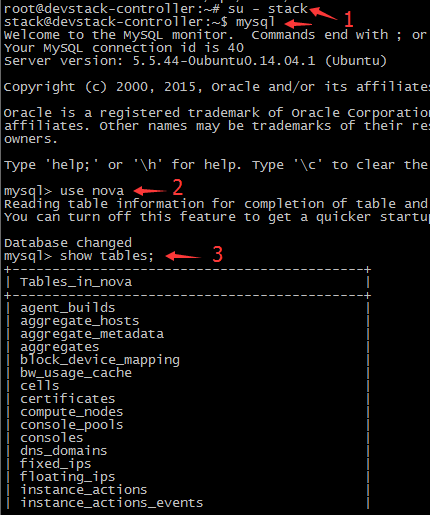
**nova-cert**提供 x509 证书支持

**Database**

Nova 会有一些数据需要存放到数据库中，一般使用 MySQL。

数据库安装在控制节点上。

Nova 使用命名为 “nova” 的数据库。

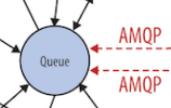


**Message Queue**

在前面我们了解到 Nova 包含众多的子服务，这些子服务之间需要相互协调和通信。

为解耦各个子服务，Nova 通过 Message Queue 作为子服务的信息中转站。

所以在架构图上我们看到了子服务之间没有直接的连线，它们都通过 Message Queue 联系。



OpenStack 默认是用 RabbitMQ 作为 Message Queue。

MQ 是 OpenStack 的核心基础组件，我们后面也会详细介绍。  
  
下一节我们讨论这些 Nova 组件如何协调工作。