OpenStack的核心结构如下所示：

架构

认证KeyStone

镜像Glance

计算Nova

OpenStack核心

存储Clinder

网络Neutron

介绍：

1. OpenStack是一个平台，不是一个具体的实施方案。它的各个组件都采用Driver的架构，支持各种具体的实现技术。比如OpenStack的存储服务Cinder只定义了上层抽象API，具体的实现交给下面的各种Driver，比如基于LVM的iSCSI Driver, EMC, IBM等商业存储产品的Driver，或者是开源的分布式存储软件，比如Ceph、GlusterFS的Driver。
2. OpenStack本身是一个分布式系统。
3. 虚拟化

虚拟化是云计算的基础。简单的说，虚拟化使得在一台物理设备上可以跑多台虚拟机，虚拟机共享物理机的CPU、内存、IO硬件资源，但逻辑上虚拟机之间时相互隔离的。物理机一般称之为宿主机（Host），宿主机上面的虚拟机成为客户机（Guest）。

那么Host是如何将自己的硬件资源虚拟化，并提供给Guest使用的呢？这主要是通过一个叫做Hypervisor的程序实现的。根据Hypervisor的实现方式和所处的位置，虚拟化又分为两种：1型虚拟化和2型虚拟化。

**（一）1型虚拟化**

Hypervisor直接安装在物理机上，多个虚拟机在Hypervisor上运行。Hypervisor实现方式一般是一个特俗定制的Linux系统。Xen和VMWare的ESXi都属于这个类型。

Windows OS

VM2

Linux OS

VM3

Linux OS

VM1

Hypervisor（ESXi，Xen）

Server Hardware – DELL, HP etc

**（二）2型虚拟化**

物理机上首先安装常规的操作系统，比如Redhat、Ubuntu和Windows。Hypervisor作为OS上的一个程序模块运行，并对管理虚拟机进行管理。KVM、VirtualBox和VMWare Workstation都属于这个类型。

Linux OS

VM1

Windows OS

VM2

Linux OS

VM3

Hypervisor（KVM）

Linux OS

Server Hardware – DELL, HP etc

理论上讲：1型虚拟化对硬件虚拟化功能进行了特别优化，性能上比2型要高；2型虚拟化因为基于普通的操作系统，会比较灵活，比如支持虚拟嵌套。嵌套意味着可以在KVM虚拟机中再运行KVM。

**（三）KVM**

基本概念：在x86平台上最热门运用最广泛的虚拟化方案莫过于KVM了。OpenStack对KVM支持得也最好。KVM的全称是：Kernel-Based Virtual Machine。也就是说KVM是基于Linux内核来实现的。KVM有一个内核模块叫做kvm.ko，只用于管理虚拟CPU和内存。那IO的虚拟化，比如存储和网络设备由谁实现呢？这个就交给Linux内核和Qemu来实现。说白了，作为一个Hypervisor，KVM本身只关注虚拟机调度和内存管理这两个方面。IO外设的任务交给Linux内核和Qemu。

**Libvirt**

Libvirt就是KVM的管理工具。其实，Libvirt除了能管理KVM这种Hypervisor，还能管理Xen，VirtualBox等等。OpenStack底层也使用Libvirt。它包含三个东西：后台daemon程序libvirt、API库和命令行工具virsh。

1. libvirtd是服务程序，接收和处理API请求；
2. API库使得其他人可以开发基于Libvirt的高级工具，比如virt-manager，这是个图形化的KVM管理工具。
3. virsh是我们经常要使用的KVM命令行工具。

作为KVM和OpenStack的实施人员，virsh和virt-manager是一定要会使用的。