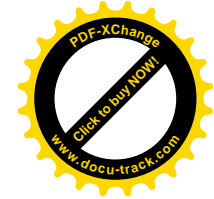


openstack™
CLOUD SOFTWARE

OpenStack从入门到精通实践 第2周

DATAGURU专业数据分析社区

Openstack从入门到精通实践 讲师 chong-de.fang



法律声明



【声明】 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散布，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>



炼数成金逆向收费式网络课程



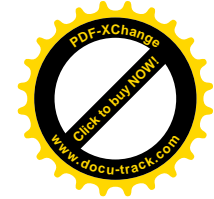
- Dataguru (炼数成金) 是专业数据分析网站 , 提供教育 , 媒体 , 内容 , 社区 , 出版 , 数据分析业务等服务。我们的课程采用新兴的互联网教育形式 , 独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围 , 重竞争压力的特点 , 同时又发挥互联网的威力打破时空限制 , 把天南地北志同道合的朋友组织在一起交流学习 , 使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本 , 直线下降至百元范围 , 造福大众。我们的目标是 : 低成本传播高价值知识 , 构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情 , 请看我们的培训网站 <http://edu.dataguru.cn>



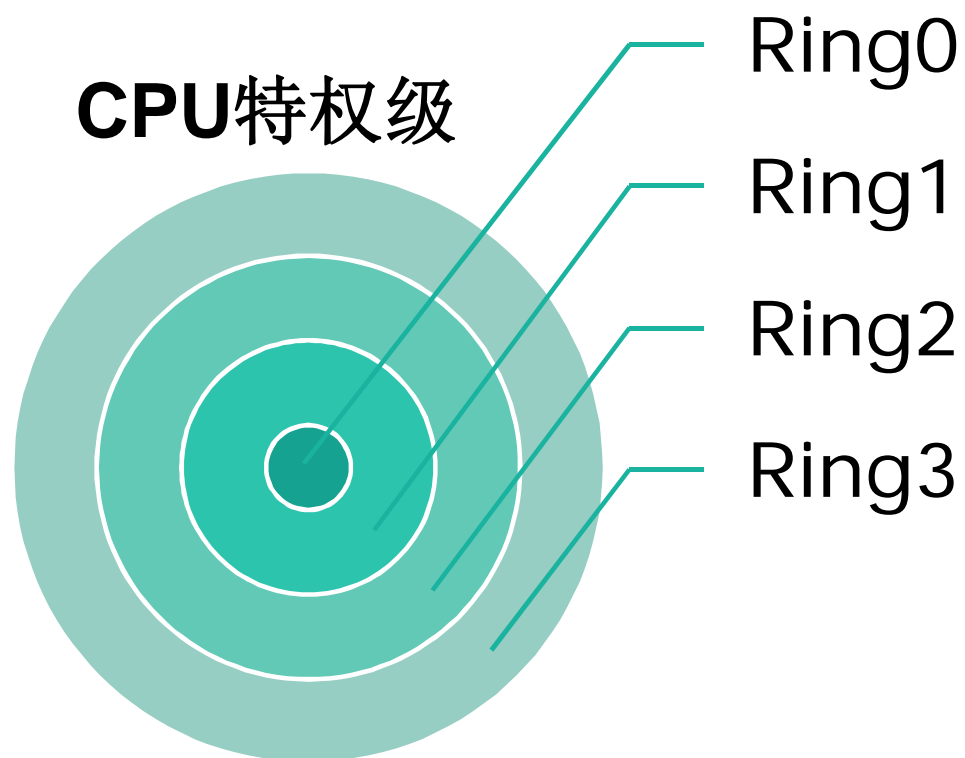
Openstack底层技术及通用组件讲解

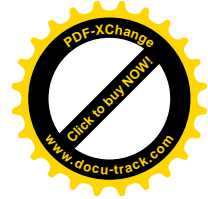


- 计算虚拟化相关技术介绍
- 网络虚拟化相关技术介绍
- OpenStack通用组件介绍

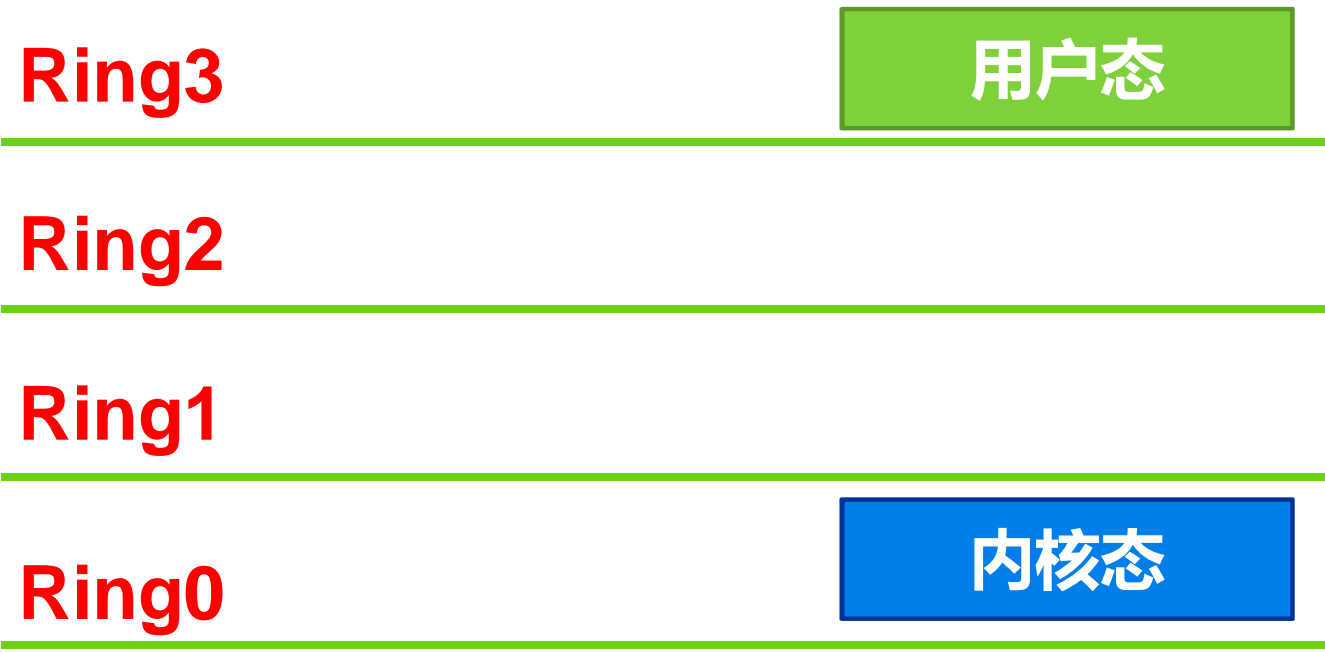


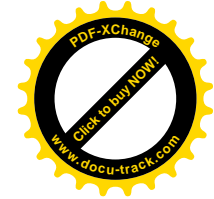
CPU特权级





内核态与用户态



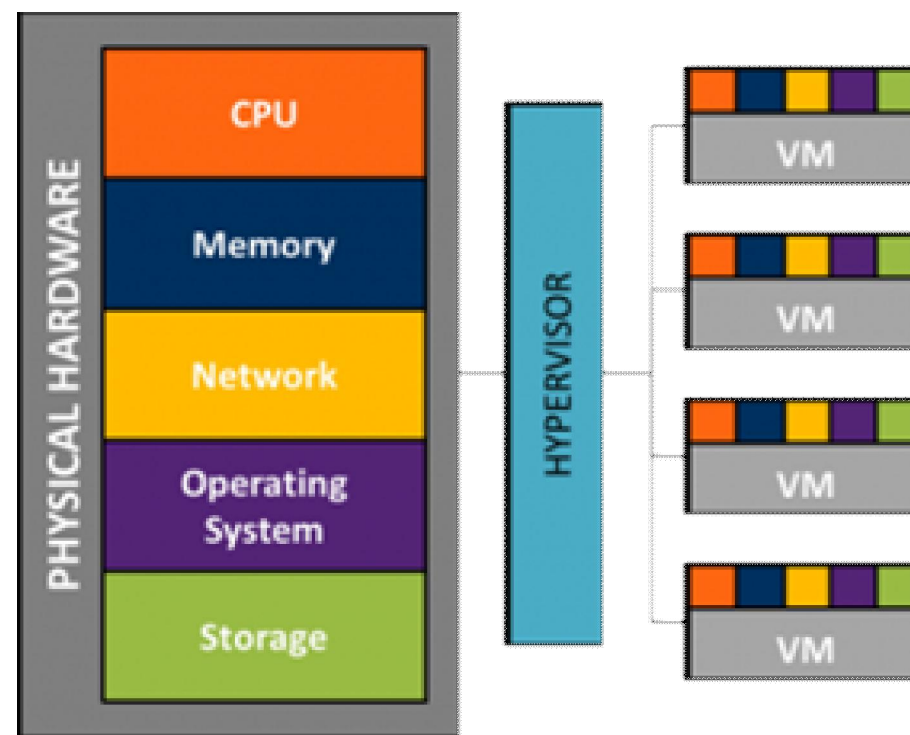


hypervisor(VMM)



■ 虚拟化管理程序

一种运行在基础物理服务器和操作系统之间的中间软件层,可允许多个操作系统和应用共享硬件。也可叫做VMM (virtual machine monitor) ,即虚拟机监视器。



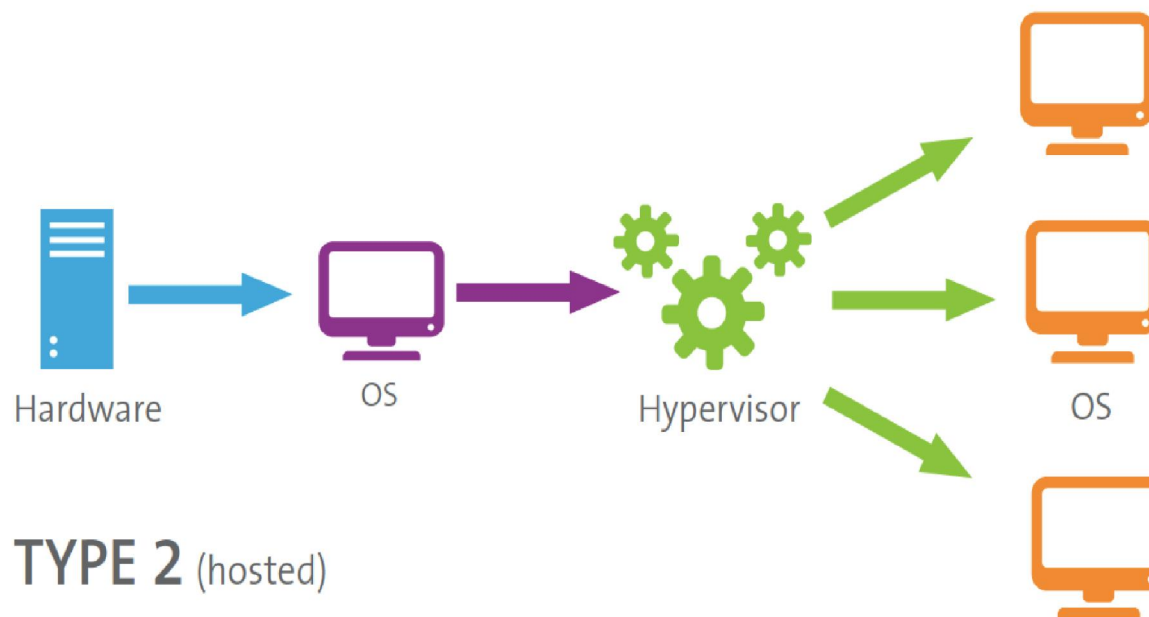


hypervisor类型：半虚拟化 (TYPE 2)



■ 半虚拟化

- 对客户操作系统 (VM)的内核进行修改
，将运行在Ring 0上的指令转为调用hypervisor



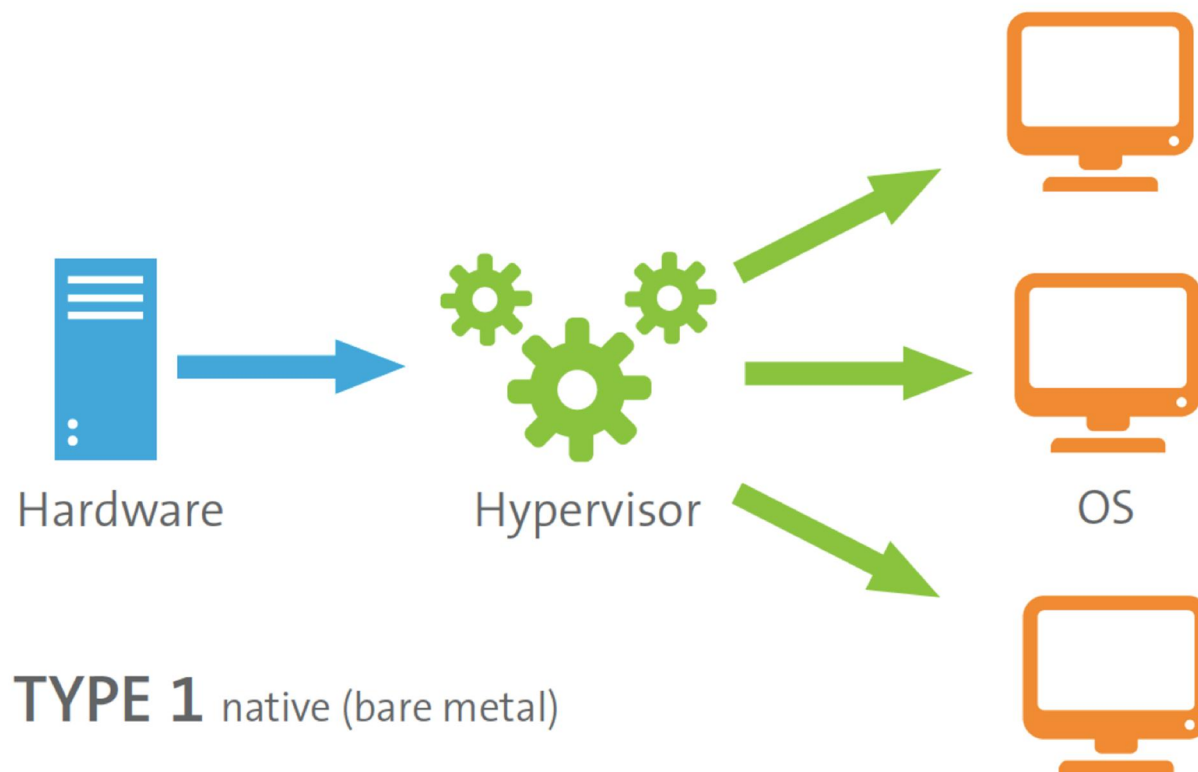


hypervisor类型：全虚拟化（TYPE 1）

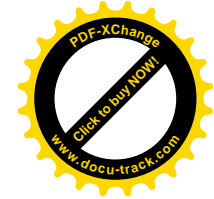


■ 硬件辅助全虚拟化

- Intel VT 和AMD-V技术
- 客户操作系统可以直接使用Ring 0而无需修改
- 查看CPU是否支持：
 - `grep "vmx" /proc/cpuinfo`
 - `grep "svm" /proc/cpuinfo`



附：支持Intel VT技术的CPU列表：<http://ark.intel.com/Products/VirtualizationTechnology>

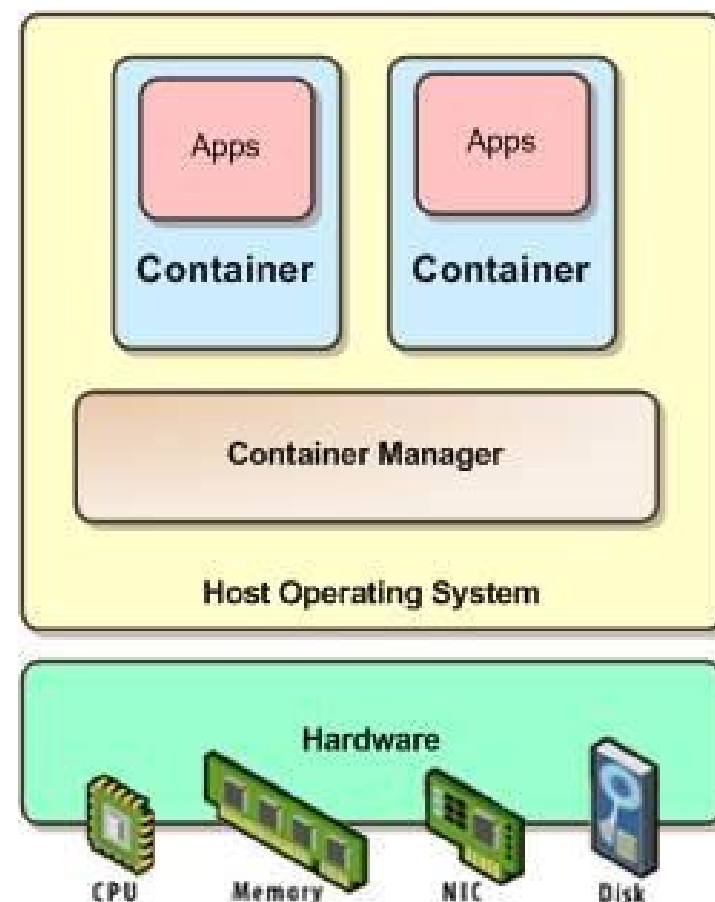


计算虚拟化的其他方式



■ 操作系统虚拟化

- 允许操作系统内核拥有彼此隔离和分割的多用户空间实例。这些用户空间实例，也称之为容器
- 基于linux内核中的namespace、chroot、cgroup实现

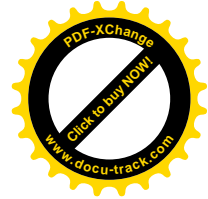




qemu



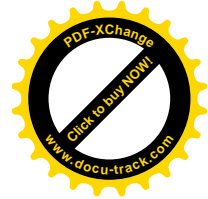
- 可以在一种架构(如PC机)下运行另一种架构(如ARM)下的操作系统和程序。
- x86架构，支持半虚拟化技术。
- 能让多个虚拟机使用同一镜像，并为每一个虚拟机配置个性化硬件环境（网卡、磁盘、图形适配器.....）。
- qemu官方网站（<http://www.qemu.org>）。



kvm



- KVM是开源软件，全称是kernel-based virtual machine（基于内核的虚拟机）。
- 是x86架构且硬件支持虚拟化技术（如 intel VT 或 AMD-V）的Linux全虚拟化解决方案。
- KVM还需要一个经过修改的QEMU软件（qemu-kvm），作为虚拟机上层控制和界面。
- KVM能让多个虚拟机使用同一镜像，并为每一个虚拟机配置个性化硬件环境（网卡、磁盘、图形适配器.....）。
- 在主流的Linux内核，如2.6.20以上的内核均已包含了KVM。



hypervisor软件比较



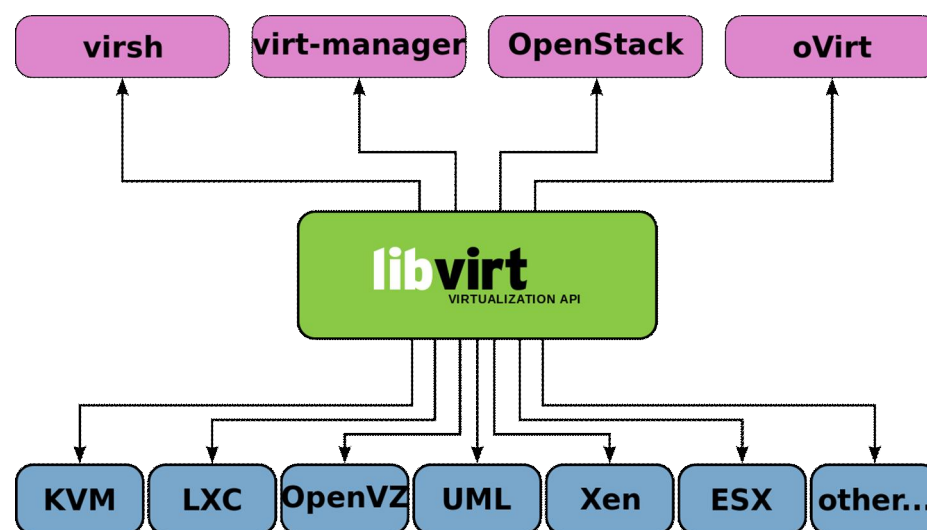
名称	厂商	主CPU	目标CPU	主系统	目标系统
qemu	Fabrice Bellard (其他开发者帮助)	Intel , AMD , ARM	x86 , x86-64 , ARM	Windows , Linux , Mac OS X	Linux
kvm	Red Hat	Intel , AMD (硬件辅助全虚拟化)	x86 , x86-64	Linux	Linux , Windows
Xen	英国剑桥大学 , Intel , AMD	Intel , AMD (半虚拟化、硬件辅助全虚拟化)	x86/x86-64	Linux , Solaris , Windows	Linux , Windows
Hyper-V	微软	Intel , AMD (硬件辅助全虚拟化)	x86/x86-64	Windows	Linux , Windows
VMware ESXi Server	VMware	Intel , AMD (硬件辅助全虚拟化)	x86/x86-64	裸机安装	Linux , Windows
Oracle VM	Oracle	Intel , AMD (硬件辅助全虚拟化)	x86/x86-64	裸机安装	Linux , Windows

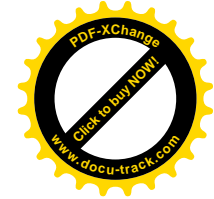


libvirt



- libvirt是一套免费、开源的支持Linux下主流虚拟化管理程序的C函数库，其旨在为包括KVM在内的各种虚拟化管理程序提供一套方便、可靠的编程接口。
- 当前主流Linux平台上默认的虚拟化管理工具virt-manager(图形化),virtsh（命令行模式）等均基于libvirt开发而成。





libvirt



■ Libvirt关键词解释

- 节点 (Node) : 一个物理机器 , 上面可能运行着多个虚拟客户机。Hypervisor和Domain都运行在Node之上。
- 域 (Domain) : 是在Hypervisor上运行的一个客户机操作系统实例。域也被称为实例 (instance , 如亚马逊的AWS云计算服务中客户机就被称为实例) 、 客户机操作系统 (guest OS) 、 虚拟机 (virtual machine) , 它们都是指同一个概念。



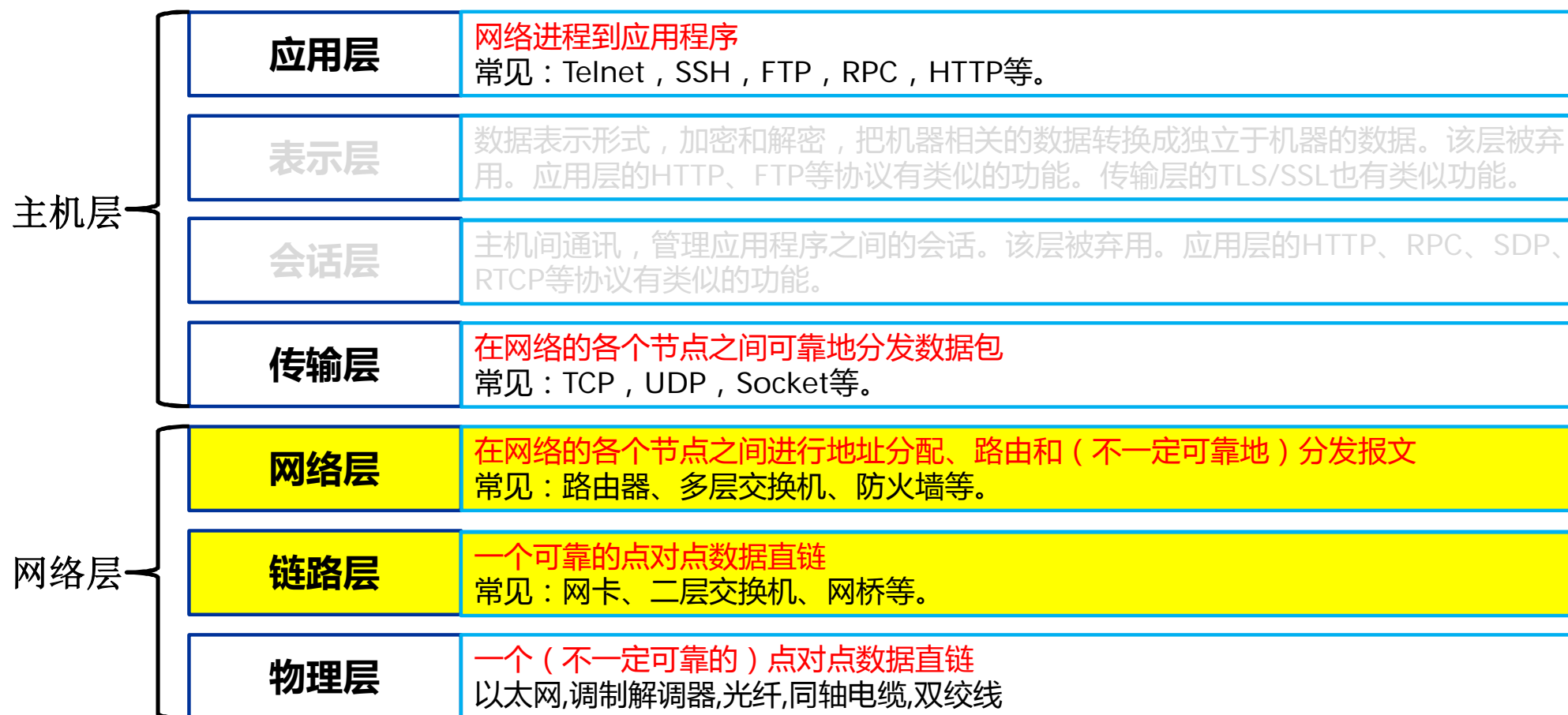
Openstack底层技术及通用组件讲解

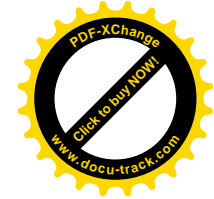


- 计算虚拟化相关技术介绍
- 网络虚拟化相关技术介绍
- OpenStack通用组件介绍



OSI七层模型





软件定义网络(SDN)

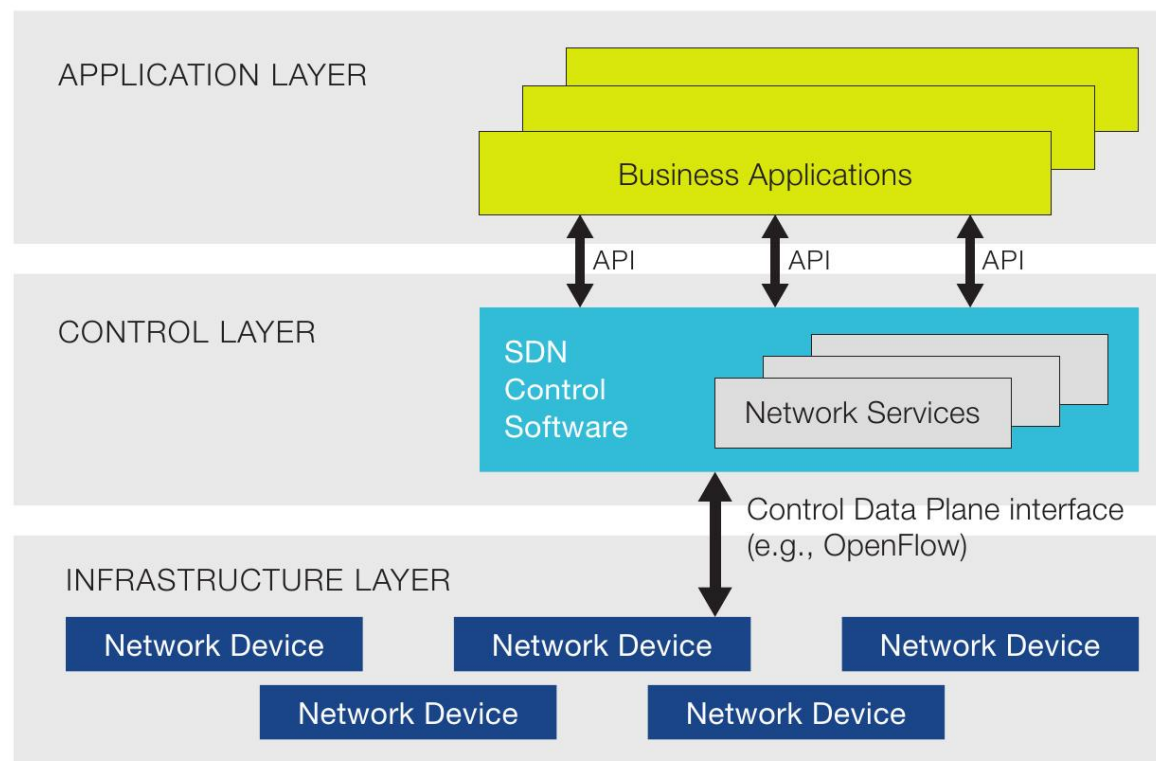


■ SDN定义

- 将网络的控制平面与数据转发平面进行分离，从而通过集中的控制器中的软件平台去实现可编程化控制底层硬件，实现对网络资源灵活的按需调配。

■ SDN架构

- 应用层：包括各种不同的业务和应用；
- 控制层：主要负责处理数据平面资源的编排，维护网络拓扑、状态信息等；
- 基础设施层：负责基于流表的数据处理、转发和状态收集。



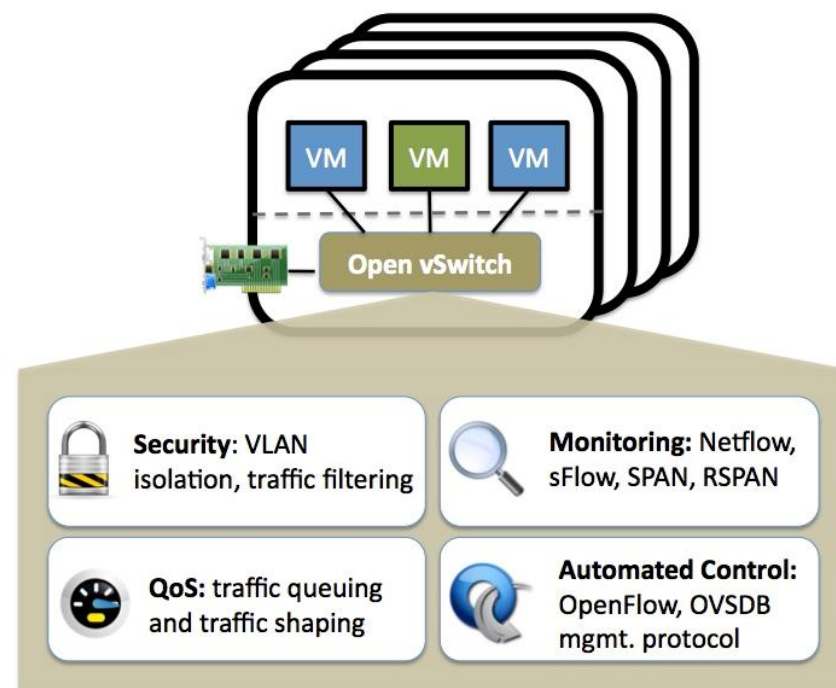


Open vSwitch



■ Open vSwitch

- 简称OVS。常用在虚拟化平台，为虚拟机提供二层交换功能。支持Xen/XenServer, KVM, and VirtualBox多种虚拟化技术。
- 支持openflow协议。可以使用任何支持 OpenFlow 协议的控制器对 OVS 进行远程管理控制

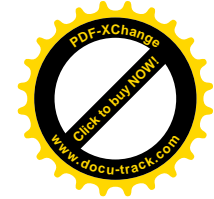




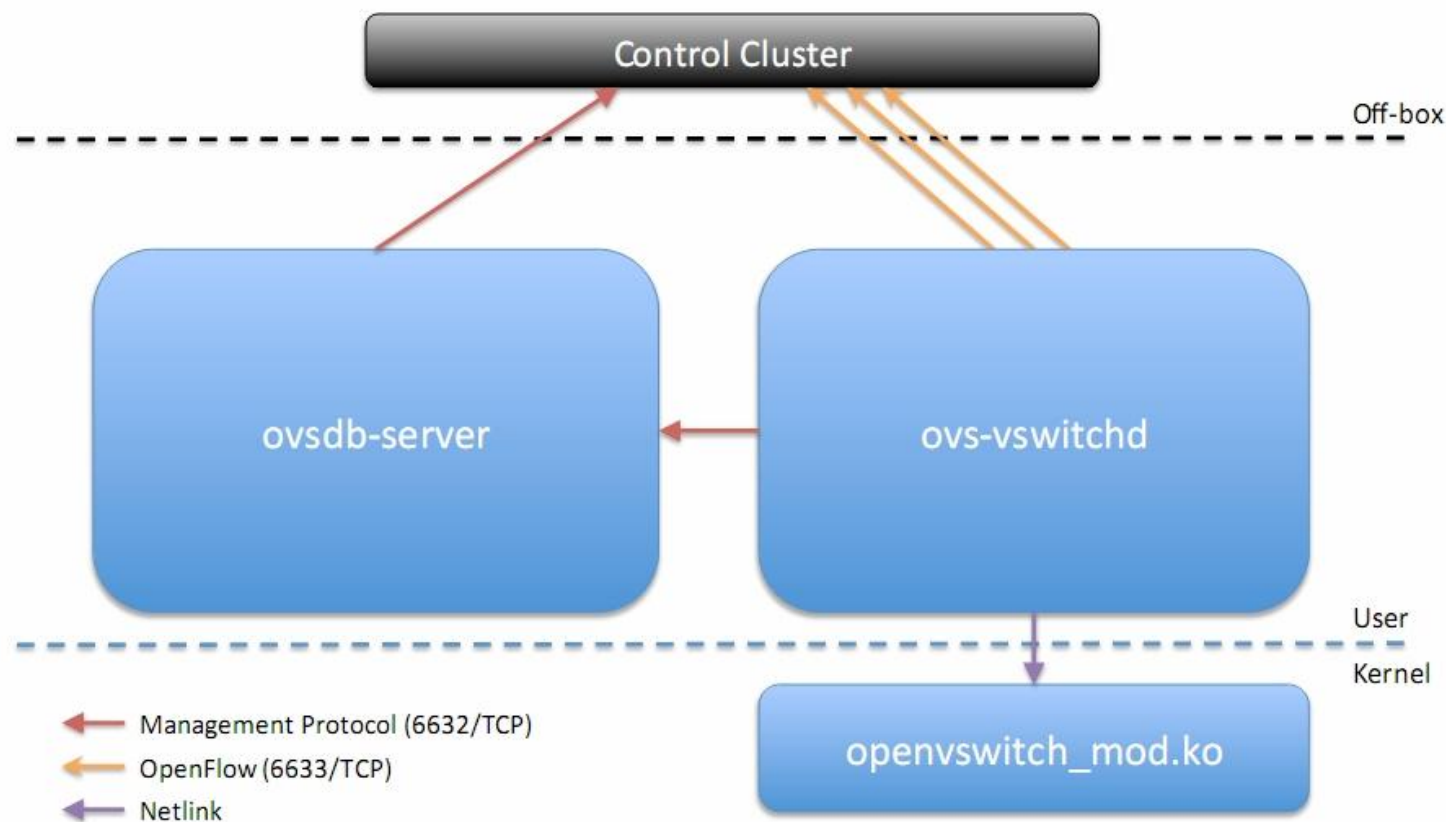
Open vSwitch**相关概念**



- Bridge: Bridge 代表一个以太网交换机（Switch），一个主机中可以创建一个或者多个 Bridge 设备。
- Port: 端口与物理交换机的端口概念类似，每个 Port 都隶属于一个 Bridge。
- Interface: 连接到 Port 的网络接口设备。
- Controller: OpenFlow 控制器。
- datapath: 负责执行数据交换，也就是把从接收端口收到的数据包在流表中进行匹配，并执行匹配到的动作。
- Flow table: 每个 datapath 都和一个“flow table”关联，当 datapath 接收到数据之后，OVS 会在 flow table 中查找可以匹配的 flow，执行对应的操作，例如转发数据到另外的端口。



Open vSwitch架构

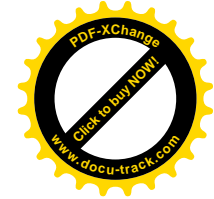




Open vSwitch常用组件及操作

- ovs-dpctl : 命令行工具, 用来配置交换机内核模块, 可以控制转发规则。
 - ovs-dpctl dump-flows br0 #查看指定bridge上的datapath信息
- ovs-vsctl : 主要是获取或者更改ovs-vswitchd的配置信息, 此工具操作的时候会更新ovsdb-server中的数据库。
 - ovs-vsctl add-br br0 #添加网桥
 - ovs-vsctl list-br #列出所有网桥
 - ovs-vsctl set-controller ovs-switch tcp:9.181.137.182:6633 #指定controller控制器
- ovs-ofctl : 用来控制OVS作为OpenFlow交换机工作时候的流表内容
 - ovs-ofctl add-flow ovs-switch "table=0, dl_src=01:00:00:00:00:00/01:00:00:00:00:00, actions=drop" #屏蔽所有进入 OVS 的以太网广播数据包

openflow控制器(floodlight): <http://www.projectfloodlight.org/floodlight/>



Linux Bridge

■ 桥接

- 定义1：是指依据OSI网络模型的链路层的地址，对网络数据包进行转发的过程，工作在OSI的第二层。
- 定义2：把一台机器上的若干个网络接口“连接”起来。

■ Linux Bridge

- Linux上用来做二层协议交换的虚拟设备，与现实世界中的交换机功能相似。这个虚拟设备可以绑定若干个以太网接口设备，从而将它们桥接起来。
- 安装： `yum install bridge-utils -y`
- Linux Bridge配置命令： `brctl` (可以通过执行 `brctl help` 查看命令帮助)



Openstack底层技术及通用组件讲解



- 计算虚拟化相关技术介绍
- 网络虚拟化相关技术介绍
- OpenStack通用组件介绍



python相关说明



- openstack基于python2.7版本开发。
- openstack Liberty版本才支持Python3。
- 检查操作系统默认python版本命令：

```
python -V
```



- pip 是一个安装和管理 Python 包的工具

- 需要配置国内镜像源加速安装

```
vi ~/.pip/pip.conf
```

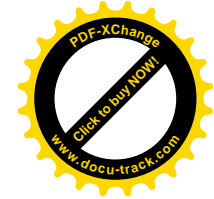
```
[global]
```

```
index-url = http://pypi.douban.com/simple
```

- 常用pip命令: pip install Package , pip show --files Package , pip uninstall Package
- 安装特定版本的package , 通过使用==, >=, <=, >, <来指定一个版本号。例如：

```
pip install 'Markdown<2.0'
```

DATAGURU专业数据分析社区



REST



- REST是一种架构风格，其核心是面向资源
- 基于Http协议
- HTTP协议里面，四个表示操作方式的动词：
GET、POST、PUT、DELETE
- GET用来获取资源，POST用来新建资源，PUT
用来更新或新建资源，DELETE用来删除资源

internalurl

```
http://controller:8774/v2/%(tenant_id)s
http://controller:5000/v2.0
http://controller:8000/v1
http://controller:8080/v1/AUTH_%(tenant_id)s
http://controller:8776/v1/%(tenant_id)s
http://controller:9696
http://controller:9292
http://controller:8004/v1/%(tenant_id)s
http://controller:8776/v2/%(tenant_id)s
```

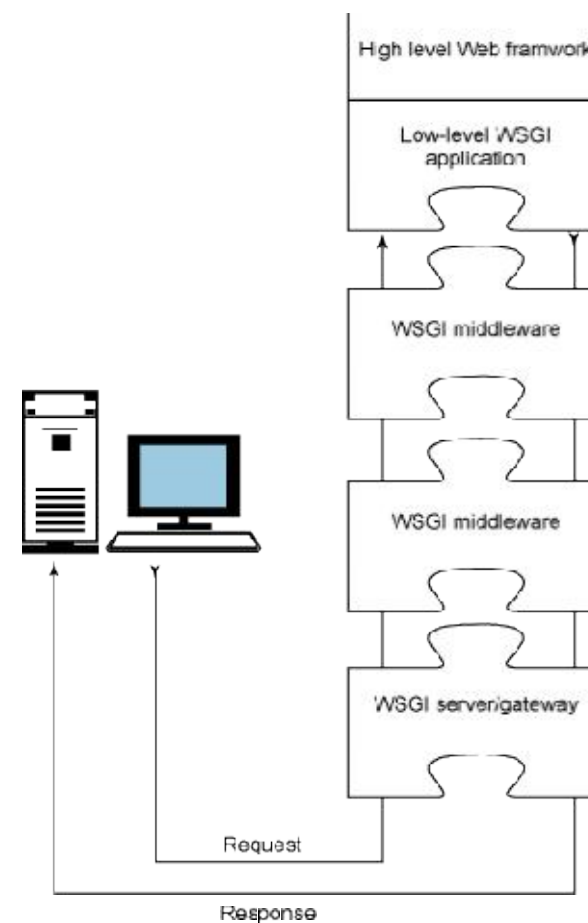


WSGI



■ WSGI定义

- WSGI的全称是Web Server Gateway Interface , 这是一个规范 , 描述了web server如何与web application交互、web application如何处理请求
- WSGI包含Server , Middleware , Application。WSGI server接收客户请求 , 传递给Middleware , Middleware根据相关配置路由给WSGI application 处理

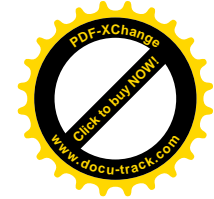




Paste Deployment



- Paste Deployment(简称PD)是一个WSGI工具包
- 基于PD的应用配置文件，内容被分为很多段（section），PD只关心带有前缀的段，比如 [app:main]或者[filter:errors]
- 一个section的内容是以键=值来标示的。#是一个注释。在段的定义中，有以下几类：
 - [app:main]:定义WSGI应用，main表示只有一个应用，有多个应用的话main改为应用名字
 - [server:main]:定义WSGI的一个server
 - [composite:xxx]：表示需要将一个请求调度定向（dispatched）到多个,或者多个应用上
 - [fliter:]：定义“过滤器”，将应用进行进一步的封装。
 - [DEFAULT]：定义一些默认变量的值



MariaDB



- MariaDB是一个采用Maria存储引擎的MySQL分支版本
- MariaDB完全兼容 MySQL , 包括 API 和客户端协议
- Openstack的核心项目Keystone,Cinder,Glance,Neutron,Nova等均使用到它来存放相关持久数据。
- 常见操作：
 - create database keystone;
 - GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY 'keystone';
 - GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY 'keystone';

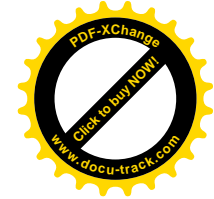




RabbitMQ



- Message Queue(MQ)定义
 - MQ是消费-生产者模型的一个典型的代表，一端往消息队列中不断写入消息，而另一端则可以读取或者订阅队列中的消息
- RabbitMQ(官网地址：<http://www.rabbitmq.com>)是一个由erlang开发的基于AMQP协议 (Advanced Message Queue Protocol) 的开源实现。通常用于应用程序之间或者程序的不同组件之间通过消息来进行集成。主要名词解释：
 - 交换器 (Exchange) ，它是发送消息的实体。
 - 队列 (Queue) ，这是接收消息的实体。
 - 绑定器 (Bind) ，将交换器和队列连接起来，并且封装消息的路由信息。
- OpenStack中模块Cinder、Neutron、Nova等项目的内部组件之间的通信是通过AMQP协议实现，消息由RabbitMQ作为中间件转发



Thanks

FAQ时间