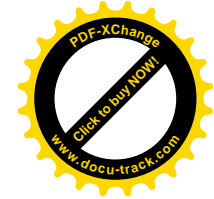


openstack™
CLOUD SOFTWARE

OpenStack从入门到精通实践 第五周

Openstack从入门到精通实践 讲师 chong-de.fang

DATAGURU专业数据分析社区



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



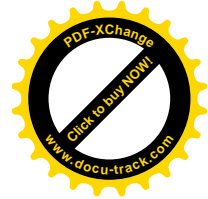
- Dataguru (炼数成金) 是专业数据分析网站 , 提供教育 , 媒体 , 内容 , 社区 , 出版 , 数据分析业务等服务。我们的课程采用新兴的互联网教育形式 , 独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围 , 重竞争压力的特点 , 同时又发挥互联网的威力打破时空限制 , 把天南地北志同道合的朋友组织在一起交流学习 , 使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本 , 直线下降至百元范围 , 造福大众。我们的目标是 : 低成本传播高价值知识 , 构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情 , 请看我们的培训网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



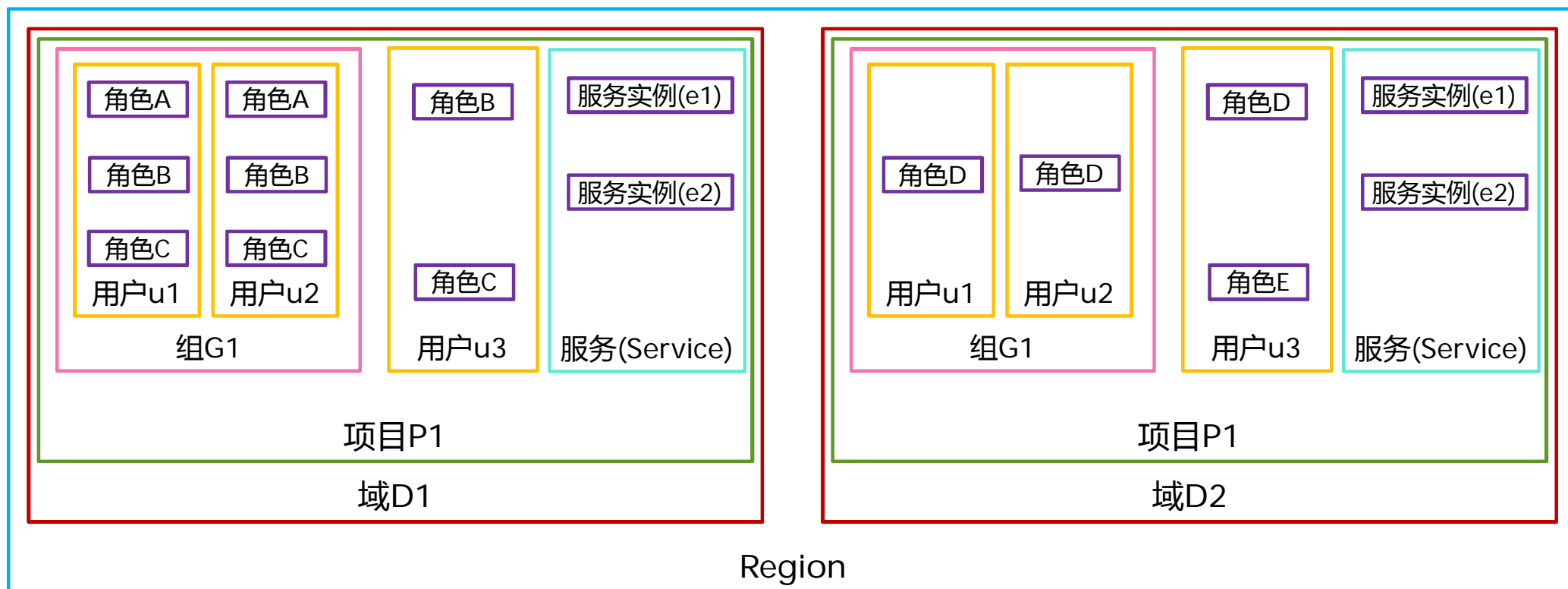
OpenStack 核心组件详解



- Keystone模块详解
- Cinder模块详解
- Swift模块详解



Keystone核心对象之间的关系

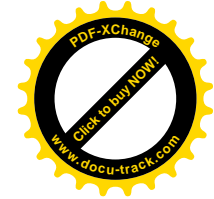




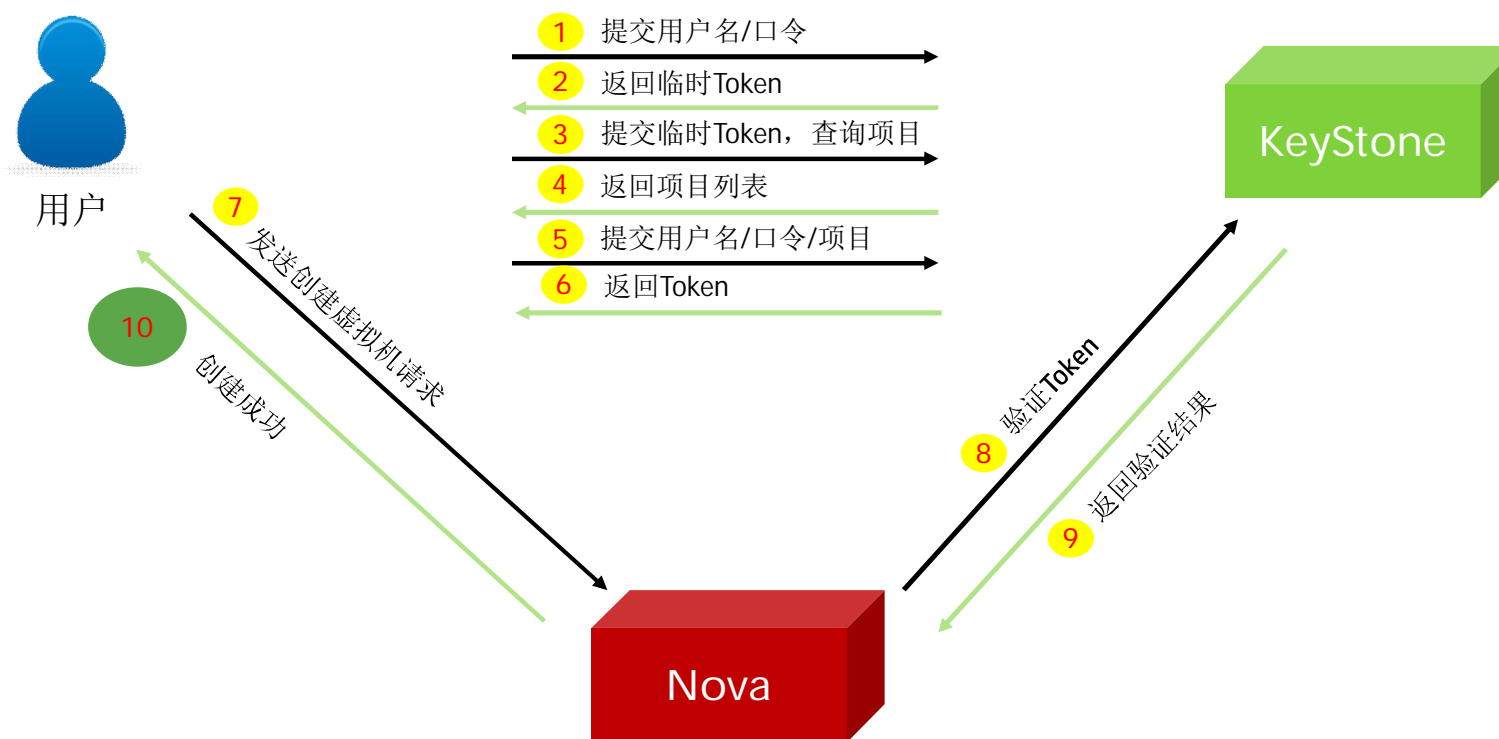
Keystone配置文件



- keystone-dist-paste.ini
 - 应用配置文件。定义keystone启动和运行过程中哪些模块需要加载，请求要经过哪些过滤器进行过滤等。
 - 配置文件路径：/usr/share/keystone
- logging.conf
 - 日志组件配置文件。定义了日志级别、日志格式等参数。
 - 配置文件路径：/etc/keystone
- policy.json
 - 定义Keystone内部模块的访问鉴权配置
 - 修改policy.json的配置参数，不需要重启keystone相关服务
 - 配置文件路径：/etc/keystone



Keystone认证流程



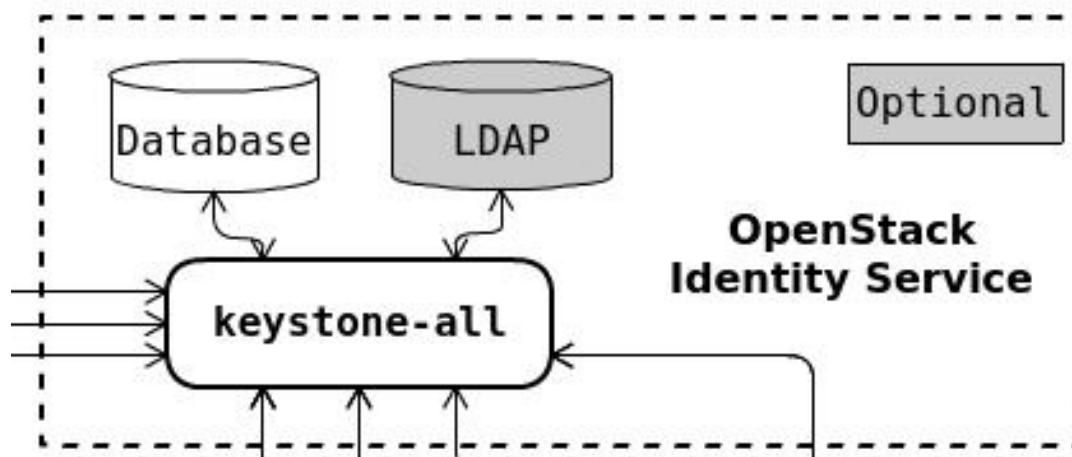


Keystone-Idap认证



■ LDAP

- 轻量目录访问协议 (Lightweight Directory Access Protocol)。按照树状存储信息的模式，适合于进行大量数据的检索。





OpenStack 核心组件详解

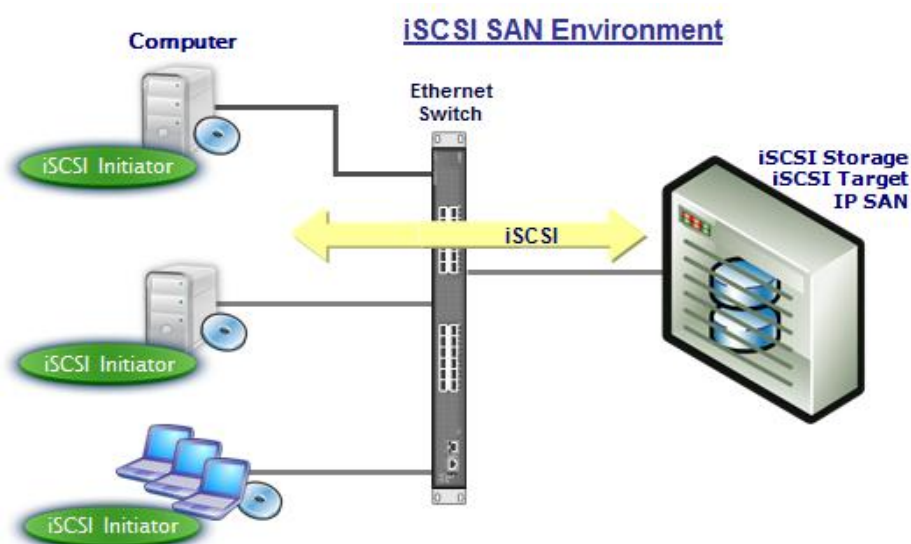


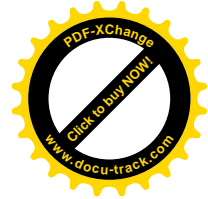
- Keystone模块详解
- Cinder模块详解
- Swift模块详解

Internet 小型计算机系统接口 (iSCSI)

■ iSCSI定义

- Internet 小型计算机系统接口 (iSCSI)。主要是透过TCP/IP的技术，将储存设备端透过iSCSI target (iSCSI目标端)功能，做成可以提供磁盘的服务器端，再透过iSCSI initiator (iSCSI初始化用户)功能，做成能够挂载使用 iSCSI target的用户端。

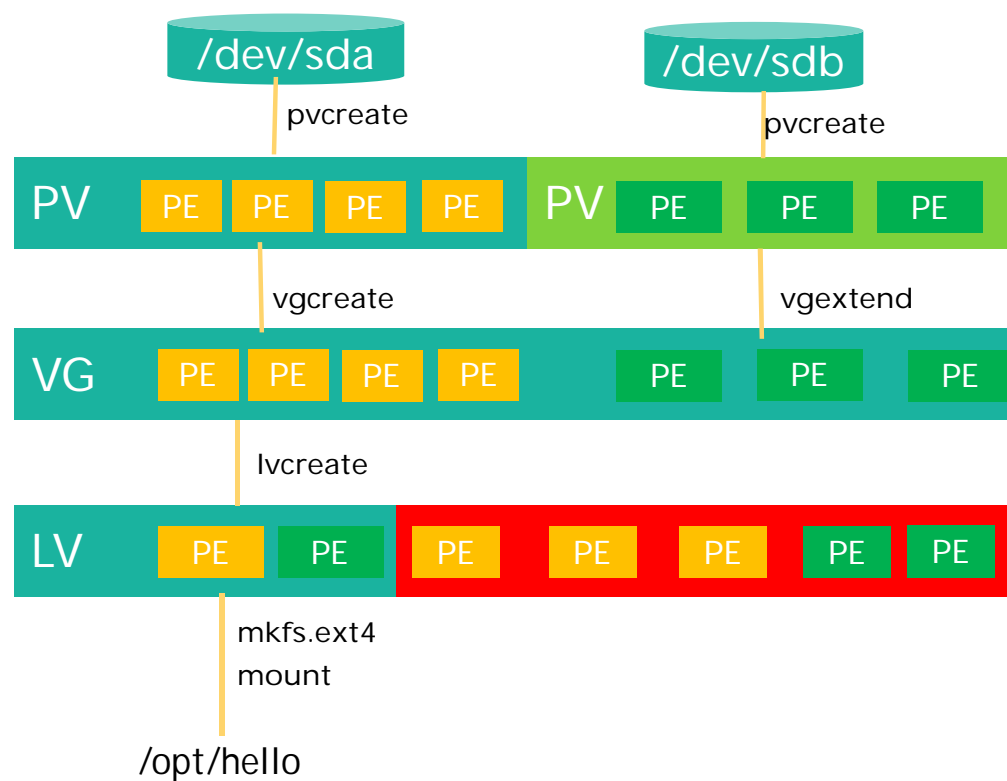




认识LVM



- PE(Physical Extend):物理拓展
- PV(Physical Volume):物理卷
- VG(Volume Group):卷组
- LV(Logical Volume):逻辑卷

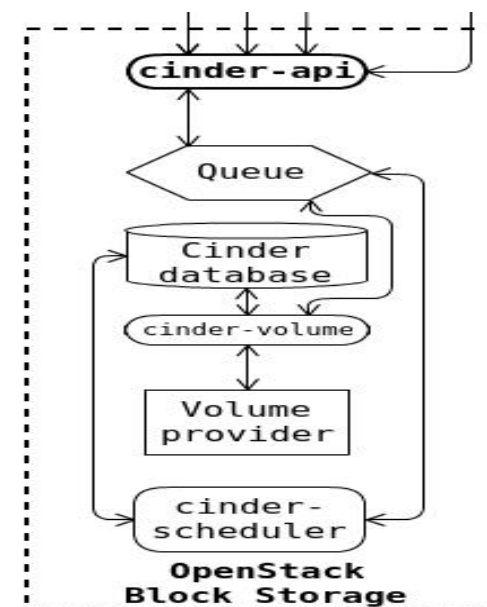


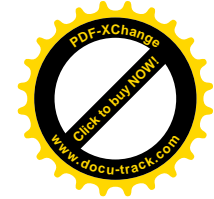


Cinder架构



- cinder-api
 - 负责接受和处理 Rest 请求，并将请求放入 RabbitMQ队列。
- cinder-scheduler
 - 处理任务队列的任务，并根据预定策略选择合适的 Volume Service 节点来执行任务。
- cinder-volume
 - 该服务运行在存储节点上，管理存储空间。每个存储节点都有一个 Volume Service，若干个这样的存储节点联合起来可以构成一个存储资源池。
- volume provider
 - 用来连接不同类型和型号的存储 <https://wiki.openstack.org/wiki/CinderSupportMatrix>

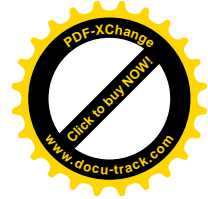




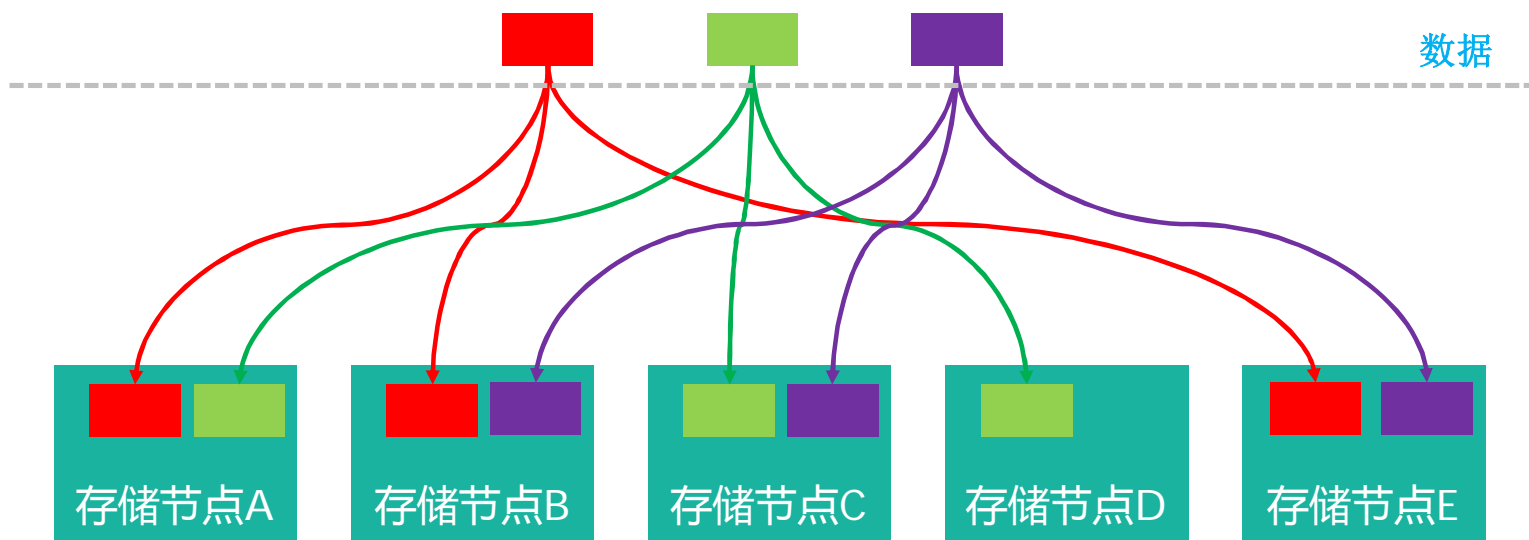
OpenStack 核心组件详解



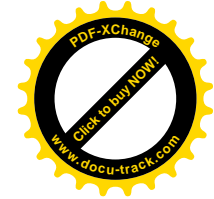
- Keystone模块详解
- Cinder模块详解
- Swift模块详解



分布式存储



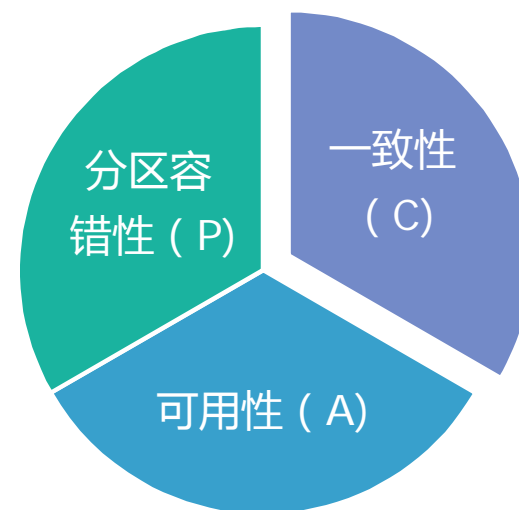
分布式存储特点： 数据分散存储在多台独立的设备上,每份数据存在多个备份



CAP



- 一致性(Consistency):所有的节点上的数据时刻保持同步
- 可用性(Availability):每个请求都能接受到一个响应, 无论响应成功或失败
- 分区容错性(Partition tolerance):系统应该能持续提供服务, 即使系统内部有消息丢失



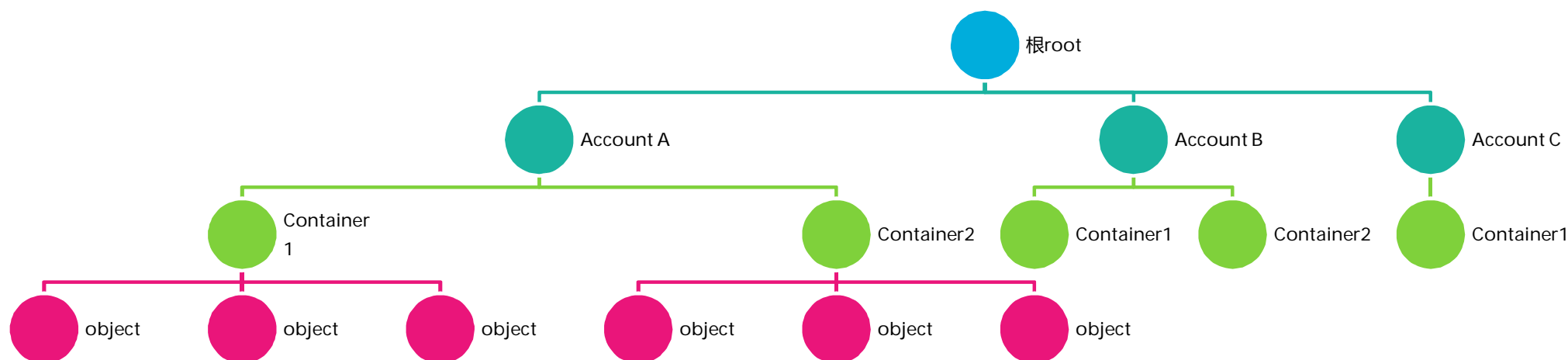
- ✓ 任何分布式系统只可同时满足二点, 没法三者兼顾
- ✓ swift放弃了一致性, 采用了**最终一致性**模型



Swift数据模型



- Account:本身只是一个存储区域，并不代表认证系统里的“账号”，但是通常会让每个Account对应一个租户
- Container:容器代表封装一组对象，类似文件夹或目录
- Object：对象，由元数据和内容组成，可以简单理解为一个对象就是一个文件

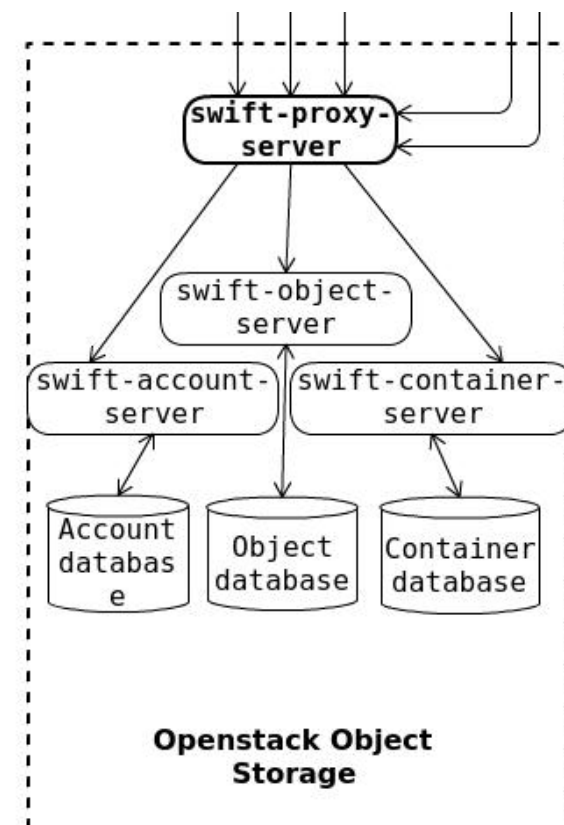




Swift核心组件

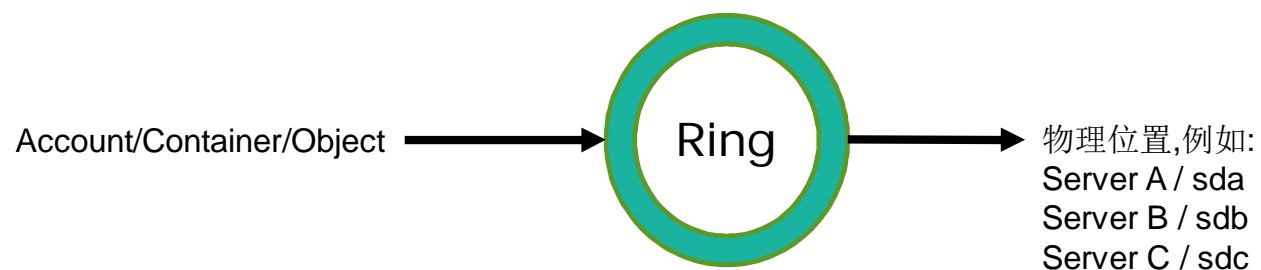


- 代理服务 (Proxy Server)
 - 代理服务器,负责Swift架构的其余组件间的相互通信。
- 账户服务 (Account Server)
 - 提供账户元数据和统计信息,并维护所含容器列表的服务,每个账户的信息被存储在一个 SQLite 数据库中。
- 容器服务 (Container Server)
 - 提供容器元数据和统计信息,并维护所含对象列表的服务,每个容器的信息也存储在一个 SQLite 数据库中。
- 对象服务 (Object Server)
 - 提供对象元数据和内容服务,每个对象的内容会以文件的形式存储在文件系统中,元数据会作为文件属性来存储,建议采用支持扩展属性的 XFS 文件系统。





Swift-环(Ring)-1



- 记录存储对象与物理位置间的映射关系
- Account/Container/Object分别定义了一个Ring
- 采用一致性哈希算法



Swift-hash算法



对象



■ 算法过程

- 1. 计算object的hash值Key

$\text{hash}(\text{object1})=1, \text{hash}(\text{object2})=2, \text{hash}(\text{object3})=3, \text{hash}(\text{object4})=4$

- 2. 计算Key mod N值,N等于节点数量

$1 \bmod 2=1$

$2 \bmod 2=0$

$3 \bmod 2=1$

$4 \bmod 2=0$

- 3. object2、object4存放在节点0, object1、object3存放在节点1

存储节点0

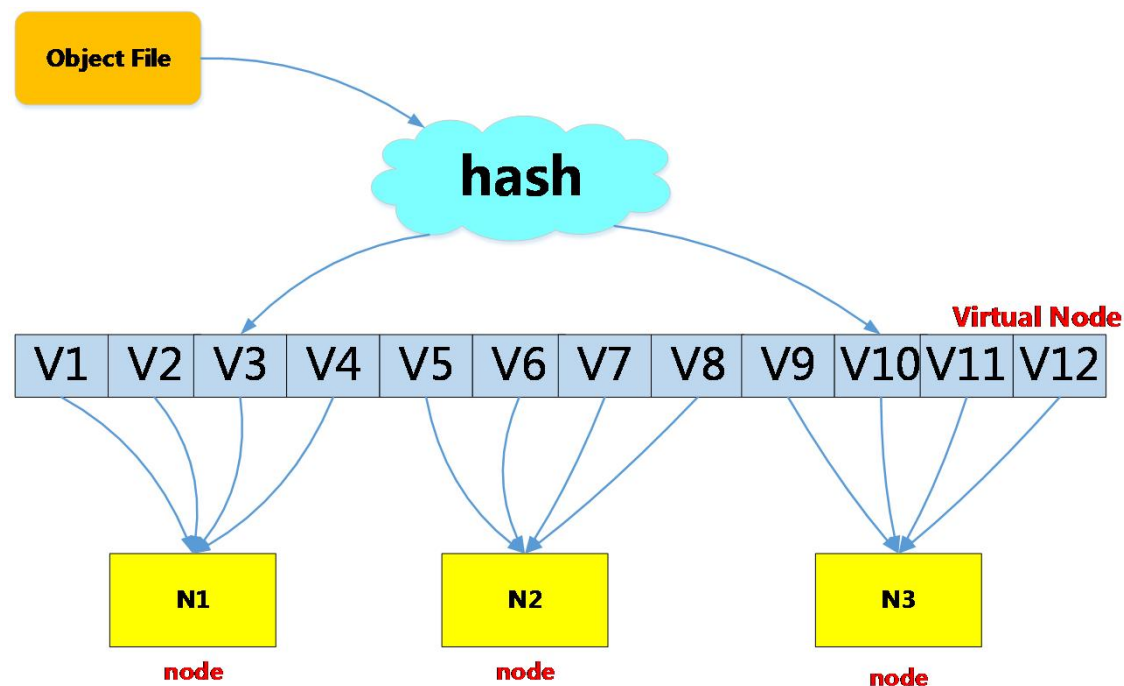
存储节点1

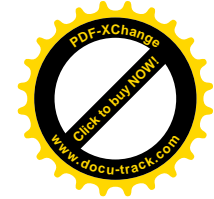


Swift环映射



- object--->node □ object--->virtual node---> node
- Partition :维护virtual node与node之间的映射关系
- Partition = 100*实际的磁盘数





Swift环创建命令解释



```
swift-ring-builder account.builder create 10 3 1
```

环类型

10 : 2^{10} , 虚拟节点个数

3 : 副本个数

1 : 在规定的时间内, 同一个虚拟节点不能被移动2次

```
swift-ring-builder account.builder add r1z1-192.168.1.34:6002/sdb1 100
```

r1:region, 表示地域

z1:zone, 表示区域

192.168.1.34:6002/sdb1 : IP地址:端口/设备名称

100:权重

```
swift-ring-builder account.builder rebalance
```

建立虚拟节点与物理设备的映射关系



Thanks

FAQ时间