云计算基础架构与案例实操

江苏省信息技术应用创新培训中心

张 伟

目录

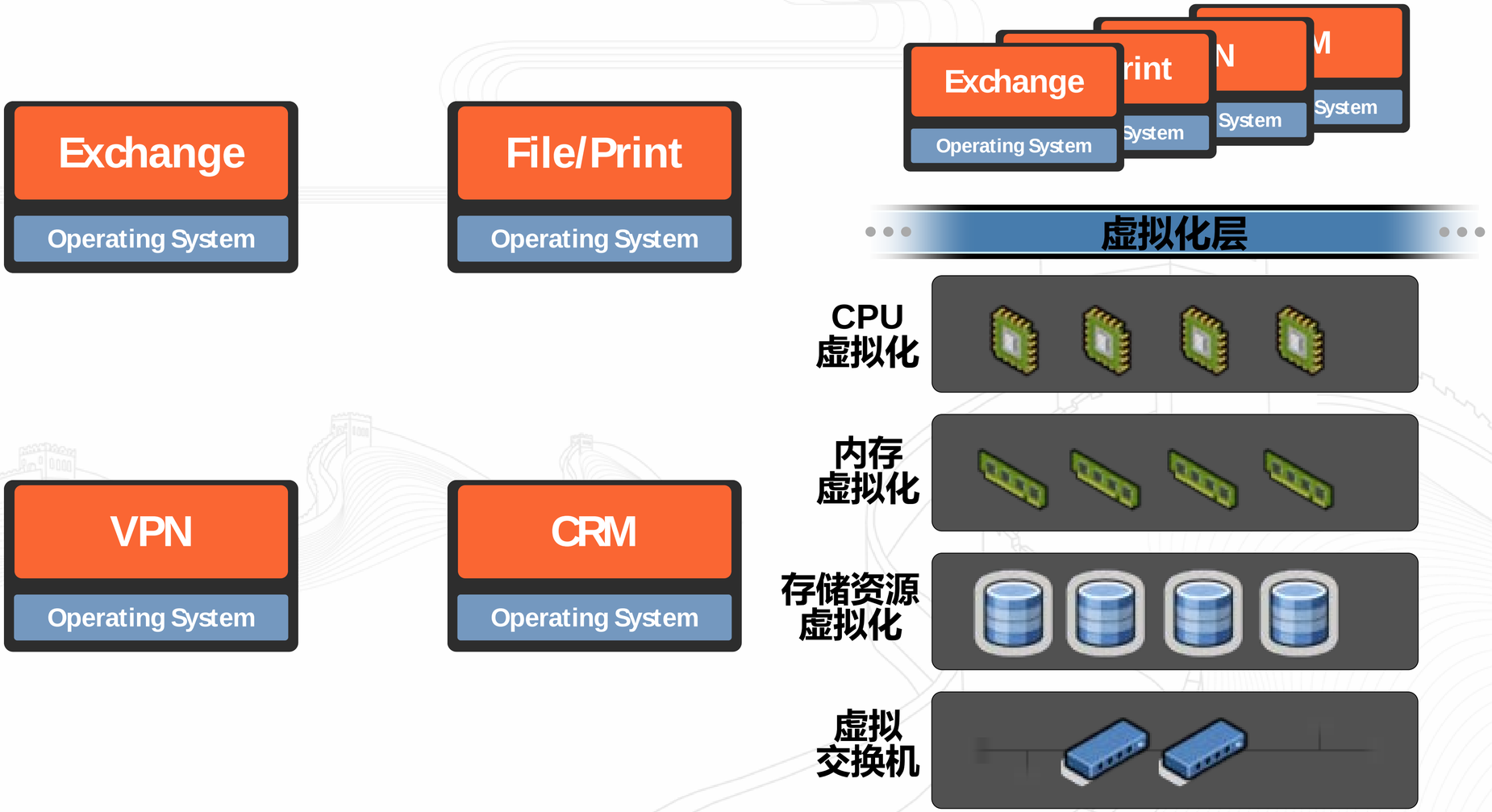
一、云计算的发展与基础架构

个人介绍

2001年开始网站的后台开发，2004 年从事ARM 嵌入式系统开发，之后一直致力于Linux 及开源软件相关技术的研究开发工作，主编《网站开发实训教程》第一版、第二版；2013 年进入的云计算领域，17年主编了《深度实践OpenStack》教材（在18年出版），近几年参与过无锡市大数据局的平台建设项目、机要局终端建设项目等，同时还参与工委会信创人才图谱等的编写关工作及江苏省应用信创培训中心相关工作。

研究领域包括：Linux系统开发，云计算的架构、开发、测试和维护，应用系统开发

服务器虚拟化对传统IT架构的改变



- 提高硬件资源的利用率
- 节省数据中心的能耗
- 提高业务的连续性
- 灵活的资源调度

软件定义网络SDN

- SDN是一种新兴的控制与转发分离并可编程的网络架构；
- 传统网络设备紧耦合的网络架构被拆分成应用、 控制、转发三层分离的架构；

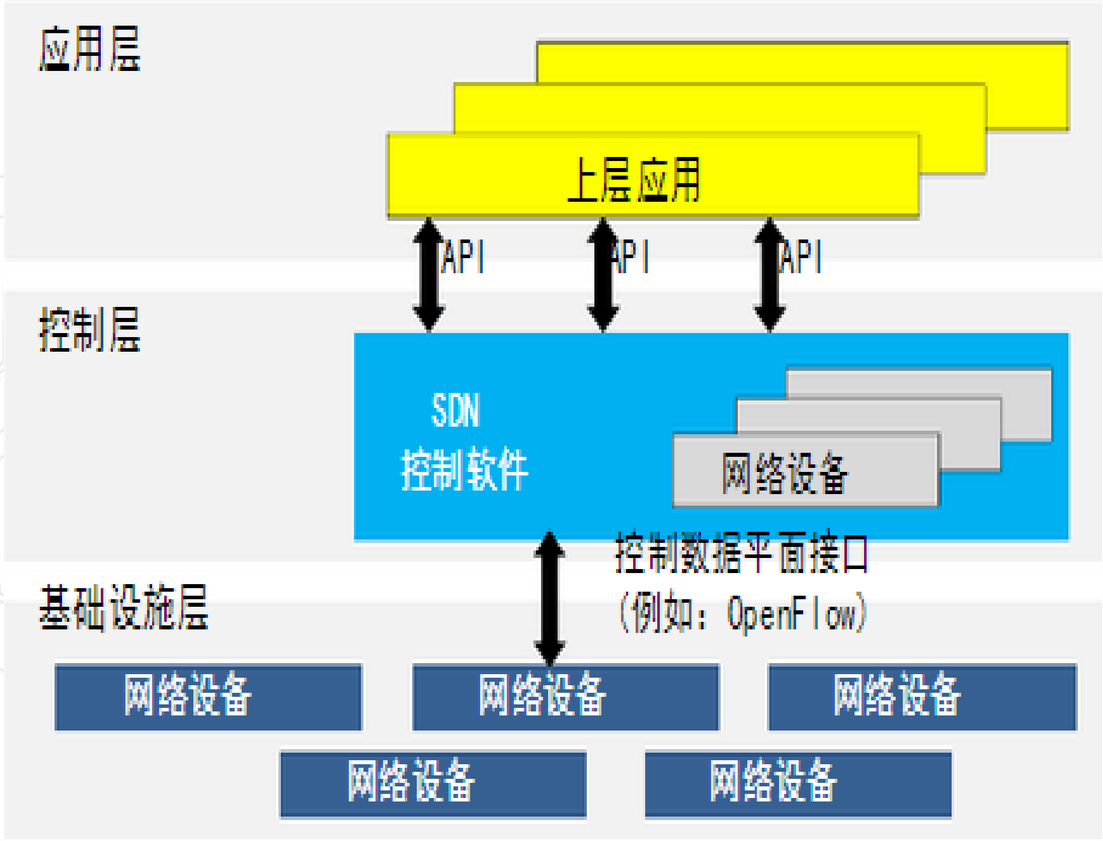
- SDN特征

1) 控制转发分离： 支持第三方控制面设备通过OpenFlow等开放协议远程控制通用硬件的交换/路由功能；

2) 控制平台集中化：提高路由管理灵活性，加快业务开通速度，简化运维；

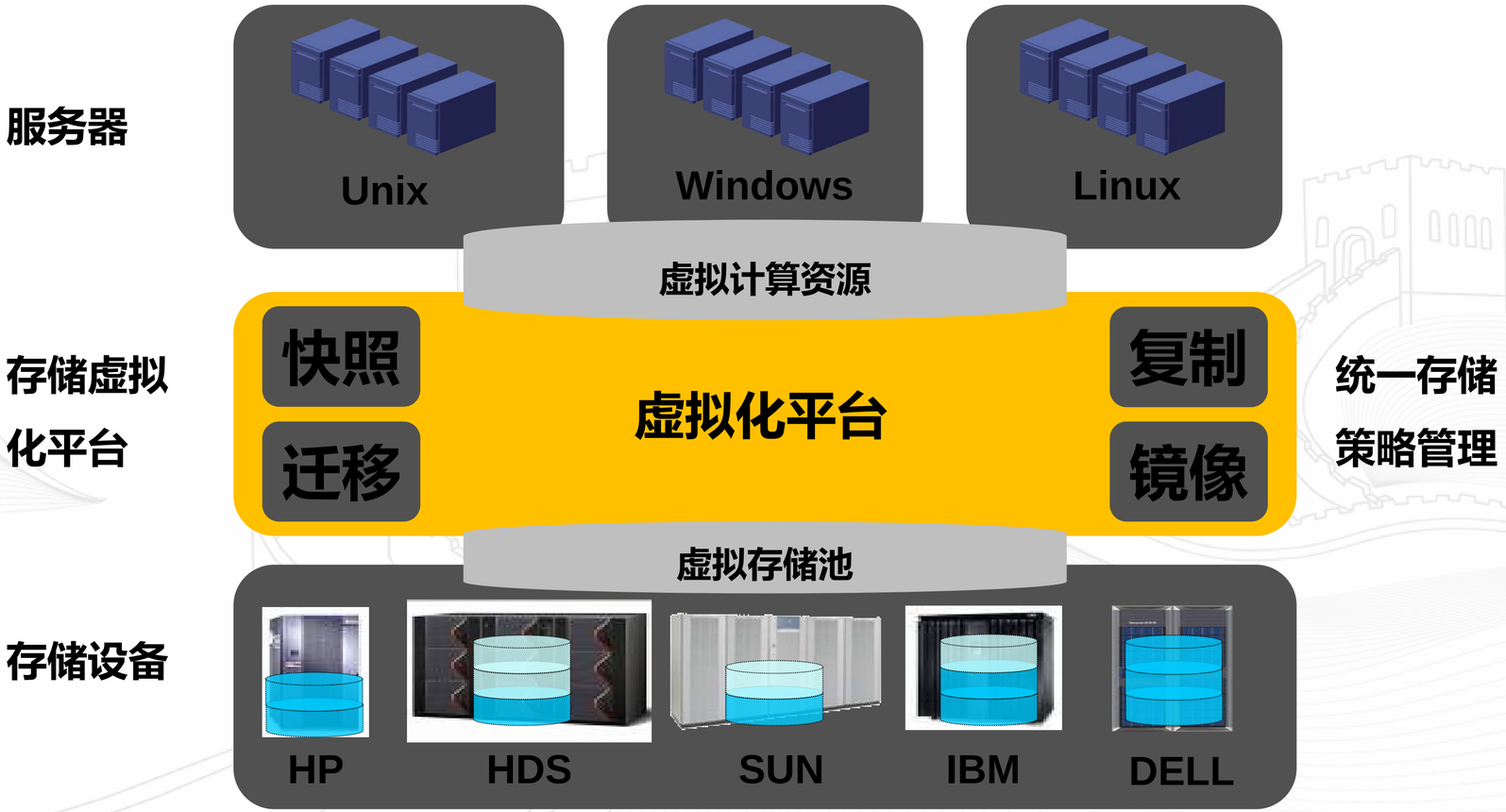
3) 控制平台通用化：多种交换、路由功能共享通用硬件设备；

4) 控制器软件可编程：可通过软件实时调整配置；



存储虚拟化

- 统一不同操作系统和异构存储资源；
- 大容量数据扩容和分区管理，实现数据统一管理；
- 数据自动备份和恢复。



OpenStack优势

OpenStack是除Linux之外的第一大社区，本节将以OpenStack为例讲解组件化及微服务化，再加上案例。

Lines of code by Module

Show 10 entries　　Search

#	Module	Lines of code
1	openstack-manuals	964153
2	edge-computing	384000
3	airshipui	247290
4	congress	146942
5	nova	89616
6	airshipctl	87769
7	dragonflow	65856
8	tacker	62450
9	openstack-ux	53265
10	tricircle	51538

Showing 1 to 10 of 762 entries　　Previous　Next

代码行数

Emails by Module

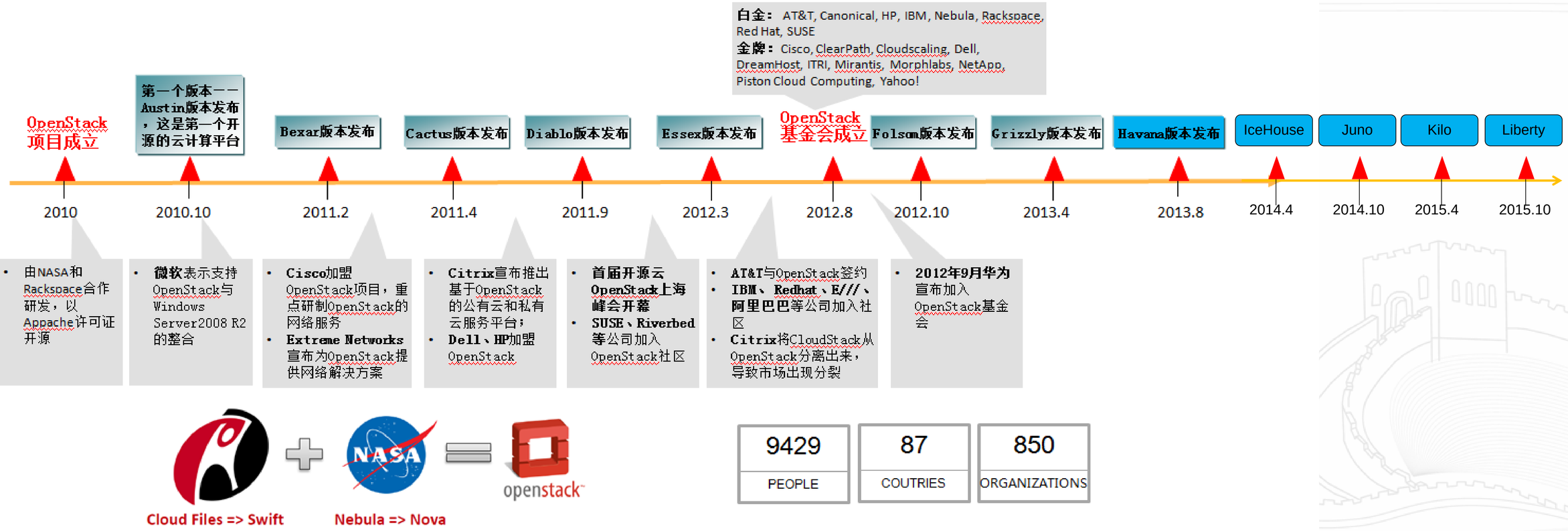
Show 10 entries　　Search

#	Module	Emails
1	neutron	308
2	nova	160
3	airship-treasuremap	130
4	unknown	125
5	ironic	119
6	octavia	89
7	election	62
8	kolla	60
9	horizon	52
10	magnum	49

Showing 1 to 10 of 55 entries　　Previous　Next

开发者数量

OpenStack发展简史



OpenStack既是一个社区，也是一个项目和一个开源软件，它提供了一个部署云的操作平台或工具集。其宗旨在于，帮助组织运行为虚拟计算或存储服务的云，为公有云、私有云，也为大云、小云提供可扩展的、灵活的云计算。

- OpenStack项目由NASA（美国国家航空航天局）和Rackspace合作研发并发起的，以Apache许可证授权的自由软件和开放源代码项目。
- 2012年OpenStack基金会成立，成为第2大开源基金会至今（仅次于Linux基金会）
- 版本周期：每年发布2个主版本（4月和10月各发布一个），主版本发布后会进行多次小版本更新，小版本更新以修正BUG为主。
- 版本命名规则：每个主版本系列以字母表顺序（A~Z）命名，以年份及当年内的排序做版本号，如 Kilo 2015.1.0

国内的发展情况-2010-12年

- 2010年张征宇、李华创立海于捷迅AWCloud
- 三个开源于计算的领军人物：程辉、杜玉杰、陈沙克
- 2012年5月，张淳创立九州于99Cloud
- 2012年9月，OpenStack基金会正式宣告成立，程辉、杜玉杰当选个人董事。

国内的发展情况-2013-14年

- 2013年1月，OpenStack基金会董事会改选，程辉连任个人董事。
- 2013年2月，程辉离开新浪，创立UnitedStack有于-紫光。
- 2014年1月，OpenStack基金会董事会改选，杜玉杰当选个人董事。
- 2014年2月，陈喜伦、刘国辉、王瑞琳创立易捷思达EasyStack-CEC-PK。
- 2014年3月，姜林创立刻通于KeyTone Cloud-。
- 2014年7月，刘江涛创立于途腾T2Cloud-百度。
- 2014年7月，查乾创立领航盛辉（ZETTAKIT）

国内的发展情况-2015-16年

- 2015年1月，OpenStack基金会董事会改选，没有大陆华人当选个人董事。年中，由于印度个董退出，王庆接任。
- 2015年5月，胥昕创立星辰天合XSKY。
- 2016年1月，OpenStack基金会董事会改选，王庆连任个人董事；同月，UCloud和Mirantis的合资公司上海优铭UCloud宣布成立。
- 2016年4月，李明宇创立奥思数据OStorage，基于OpenStack Swift研发对象存储产品开提供技术服务，这是国内第一个以单个OpenStack项目垂直提供产品不服务的公司。
- 2016年7月，经OpenStack基金会授权，第一届OpenStack中国日在北京隆重召开，盛况空前。

国内的发展情况-2016-18年

- 2017年7月，经OpenStack基金会授权，第二届OpenStack中国日在北京隆重召开，继续吸睛。陈绪、梁冰分获2017 OpenStack中国超级先生、超级女士殊荣。
- 2017年11月，腾讯TStack勇夺OpenStack峰会超级用户桂冠！
- 2017年11 ，清华同方与UnitedStack有云达成控股收购
- 2018年1月，OpenStack基金会董事会改选，王庆、郭长波连任个人董事。
- 2018年6月，改名OpenInfra中国日的前OpenStack中国日，将在北京召开。

私有云的事实标准

OpenStack China day

http://openstackdaychina.org/

OpenStack阵营

- 第一个层面是OpenStack厂商，一般称为一大八小：一大是华为，八小是 AWCloud、99Cloud、EasyStack、UnitedStack、T2Cloud、UMCloud、KeyTone Cloud、Kylin Cloud。
- 第二个层面是硬件厂商，华为，联想，浪潮，中兴，宝德，华三等，均下注OpenStack。
- 第三个层面是大规模部署的亏联网公司，百度，金山于，乐视于，京东于， 爱奇艺等。
- 第四个层面是企业用户，移劢，联通，电信，银联，人民日报等。
- 来源：陈绪 《生死时速：中国私有于格局大裂》）
- 第五个层面广大的中小企业及普通用户。

Openstack介绍

OpenStack是一个开源的云计算管理平台，其核心的服务组件如下：

优势：

开源项目

兼容各种云平台

标准统一规范

劣势：

部署运维升级复杂

性能与扩展性较差

容灾能力不足

服务名称	功能描述
Nova	计算服务
Neutron	网络服务
Keystone	认证与授权服务
Glance	镜像服务
Swift	对象存储服务
Cinder	块存储服务
Horizon	图形化管理界面
Ceilometer	监控计量服务
Heat	编排服务调度

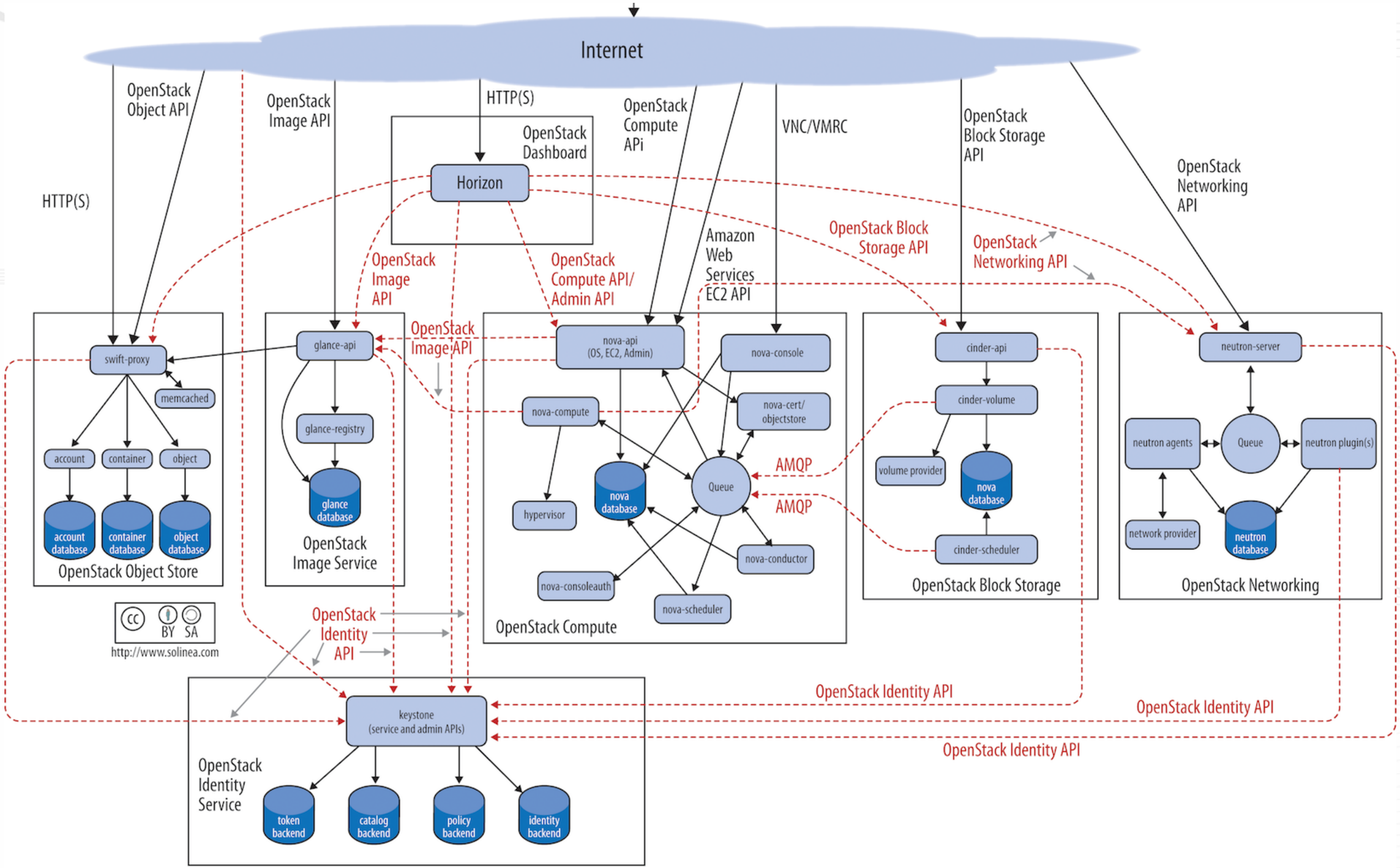
云计算的组件化

OpenStack组件化

组件内做拆分

Nova:

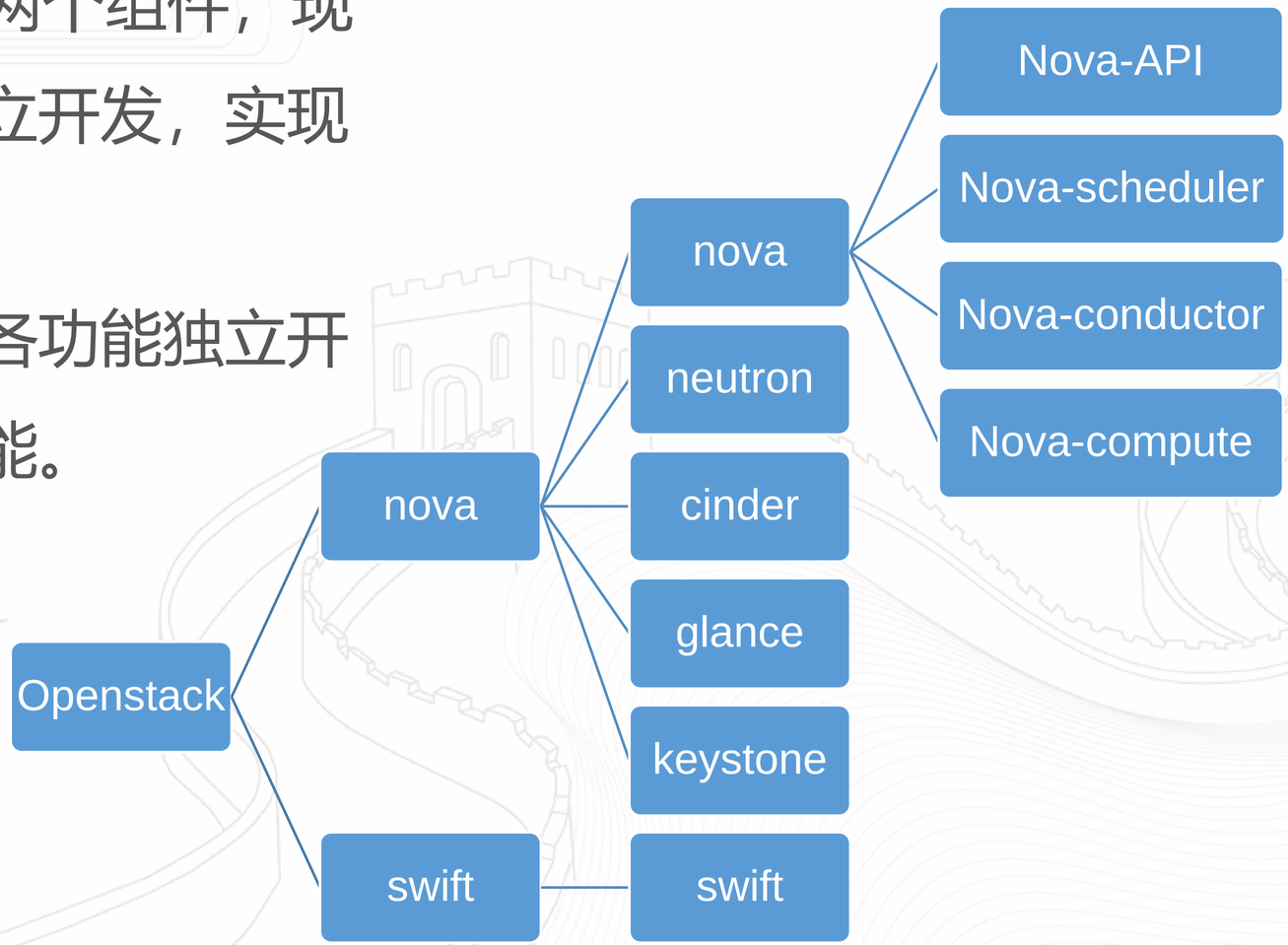
Nuetron:



OpenStack组件的发展

Openstack发布的第一版只有两个组件，现在有几十个组件。每个组件独立开发，实现一部分独立的功能。

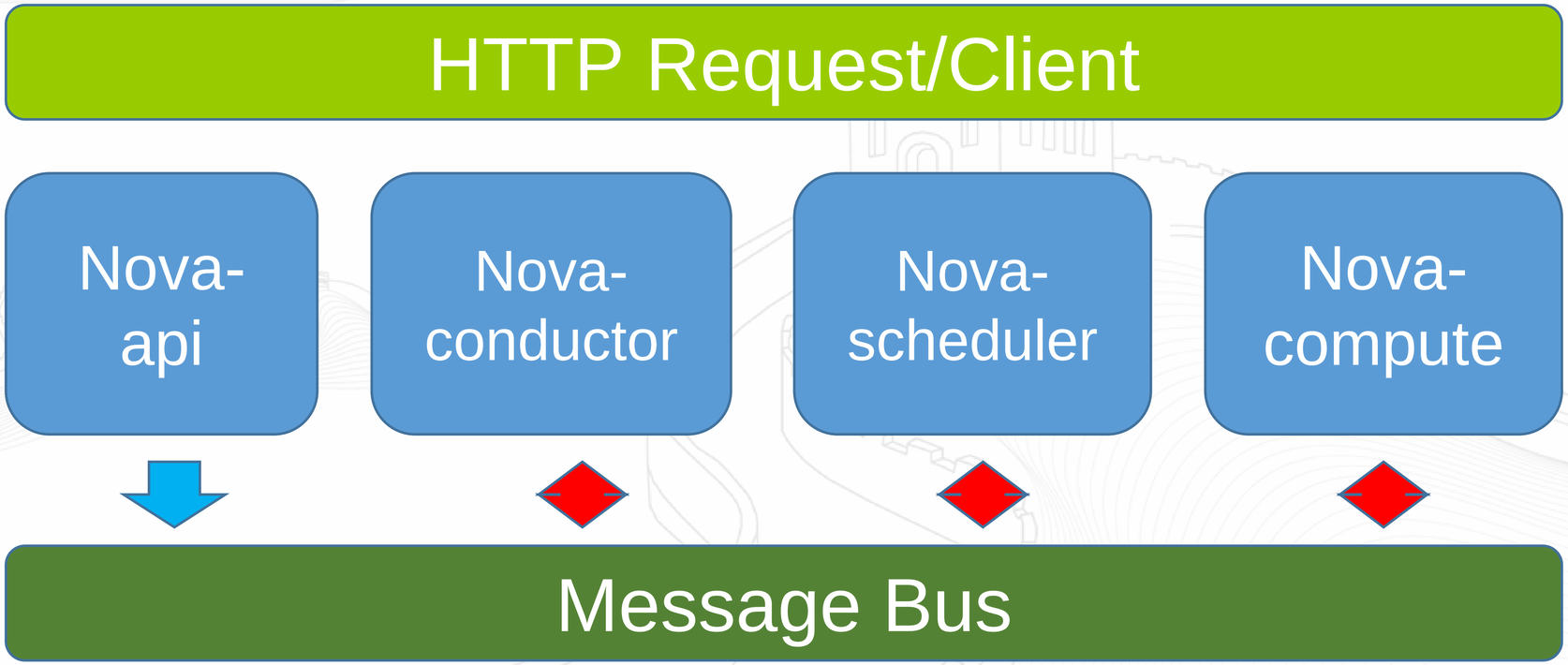
每个组件内部再做功能区分，各功能独立开发，分工协作实现一个完整功能。



Nova组件内部架构图

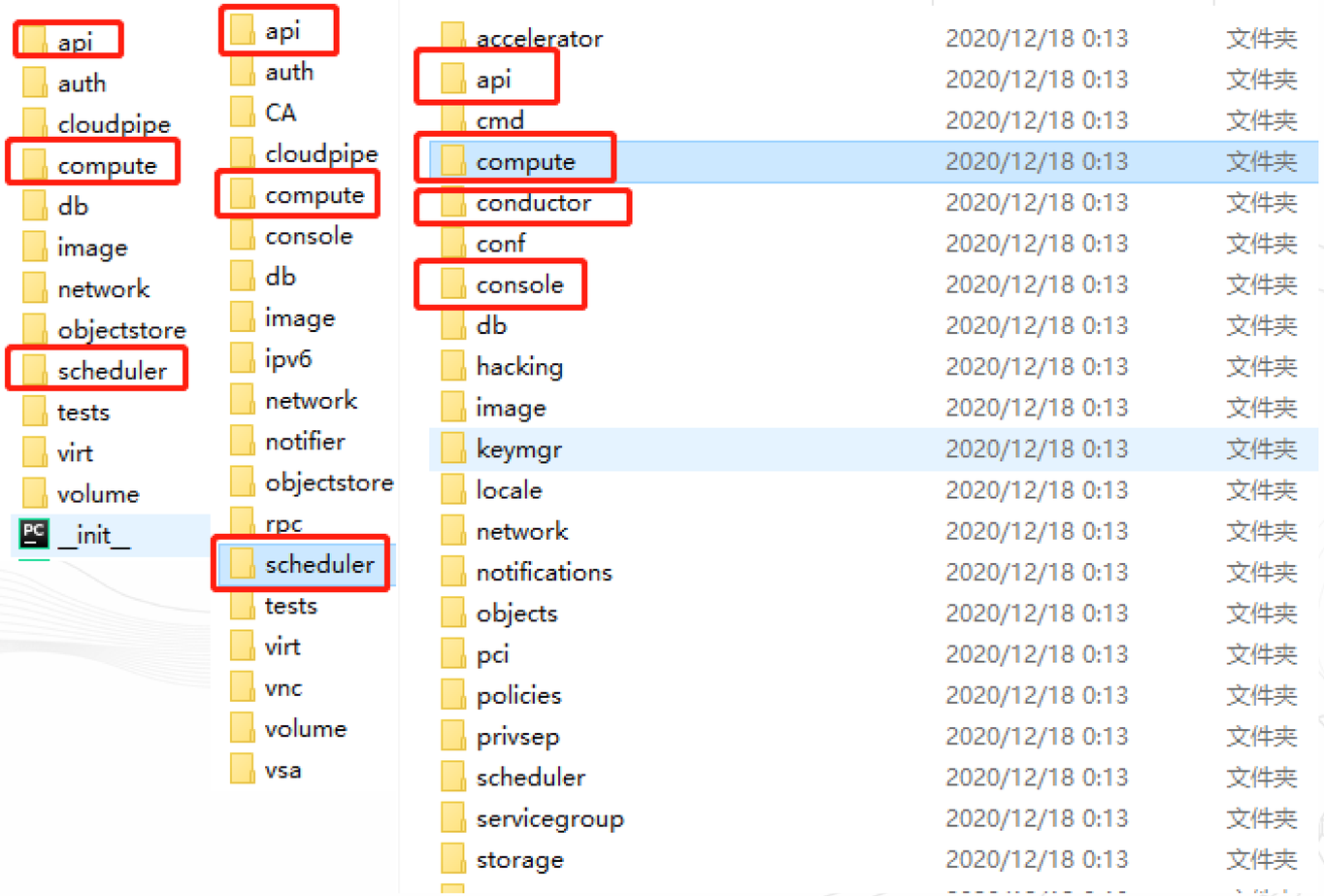
●Nova各个组件功能

*Ap*负责进行前端请求的相应；Conductor负责数据库的写入和查询；Scheduler负责资源的调度和分配；Compute负责vm的管理。

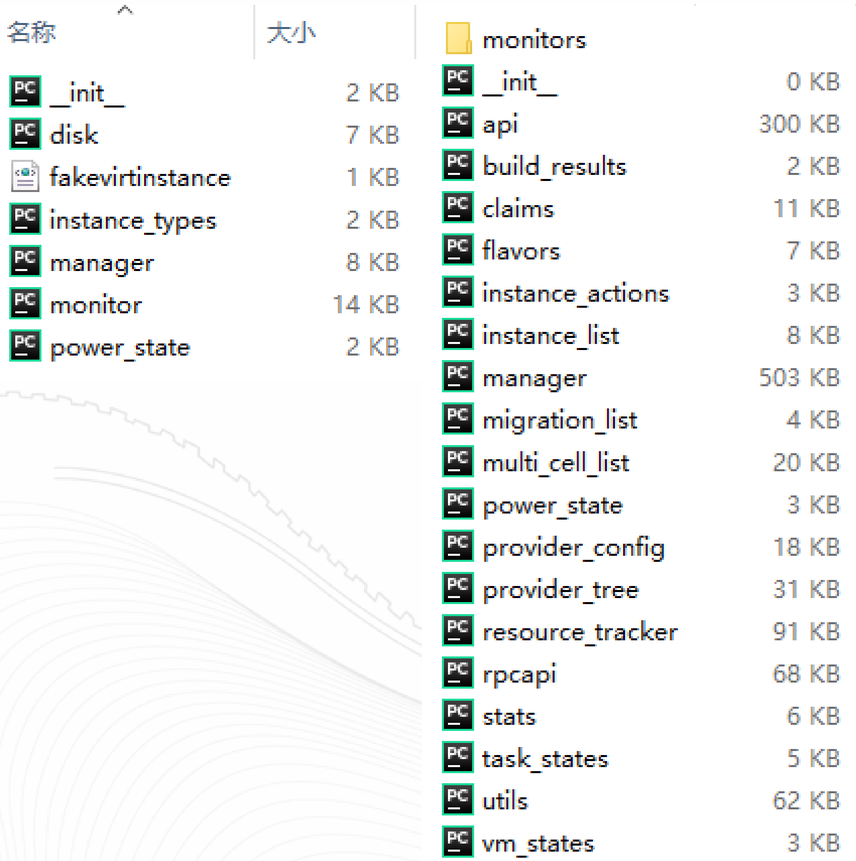


Nova组件化详解

Nova不同版本源文件目录结构对比



Nova代码中不同版本compute目录对比



https://launchpad.net/nova/aus

Nova组件之间的调用

Nova API采用的是Rest api，前后端分离的一套规范。

Rest api好处：

1、轻量直接通过https/https，不需要额外的协议，通常有post/get/put/deletec操作。

2、面向资源，一目了然，具有自解释性.

3、数据描述简单，一般通过json或者xml做数据通讯。

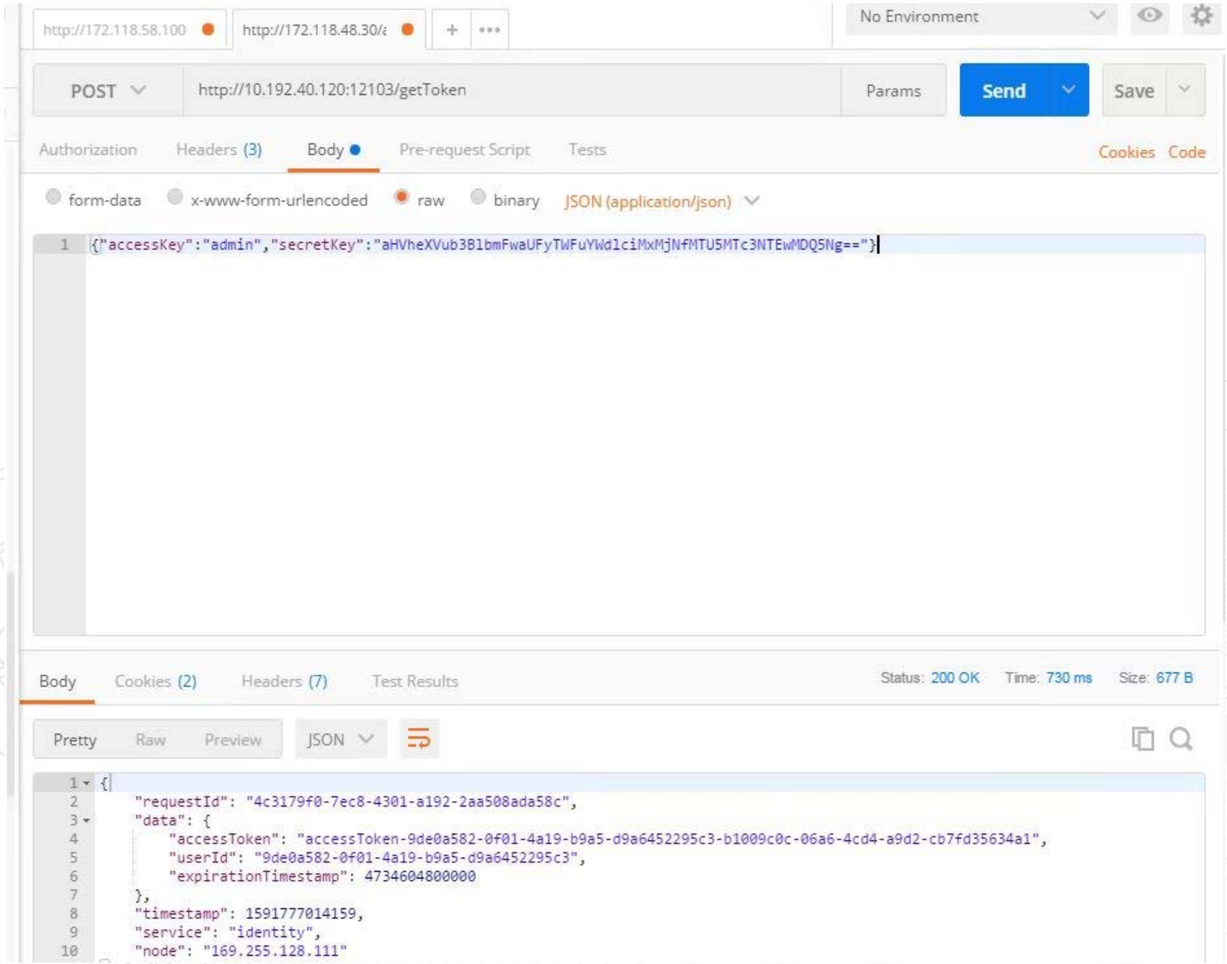
请求信息：

```
POST http://[Endpoint]/getToken
Content-Type: application/json;charset=UTF-8
{
    "accessKey":"huayun",
    "secretKey":"ZG9ub3R0ZWxseW91SHVheXVuIzEyM18xNTYyMDU5ODkxOTMw"
}
```

响应信息：

```
{
    "requestId": "e5686ad6-b674-4a23-a9f1-17ff559fed22",
    "data":{
      "accessToken":"accessToken-b9c84da0-ac21-460f-8692-142284adb70f-a0b55f1a-3a7d-4b22-9bcb-96bada99b488",
      "userId":"03ba7428-b53f-47d4-860c-082f661ef055"
    },
    "timestamp": 1515489166281
}
```

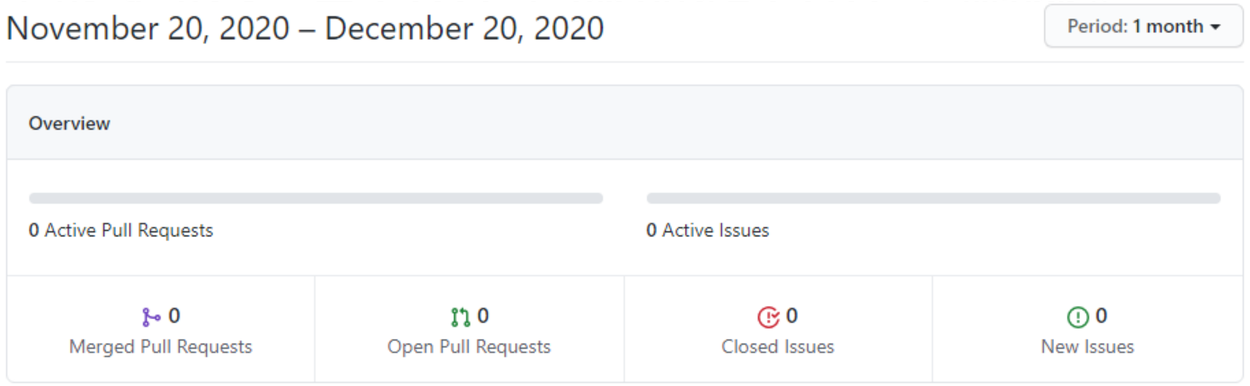
Nova API的示例



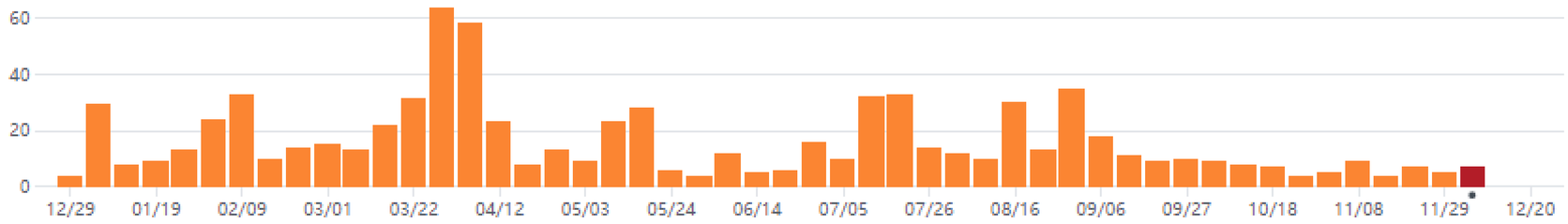
Nova API测试的示例

Nova组件开发统计

11个开发将28个提交推送到主节点，39个提交推送到所有分支。在主机上，有92个文件被更改，有1151个添加和1197个删除。



https://github.com/openstac



网络组件内部通信

网络组件内部通信是通过rabbitmq，rabbitmq也进行了容器化。

1、查询rabbitmq的容器名称

2、登陆容器

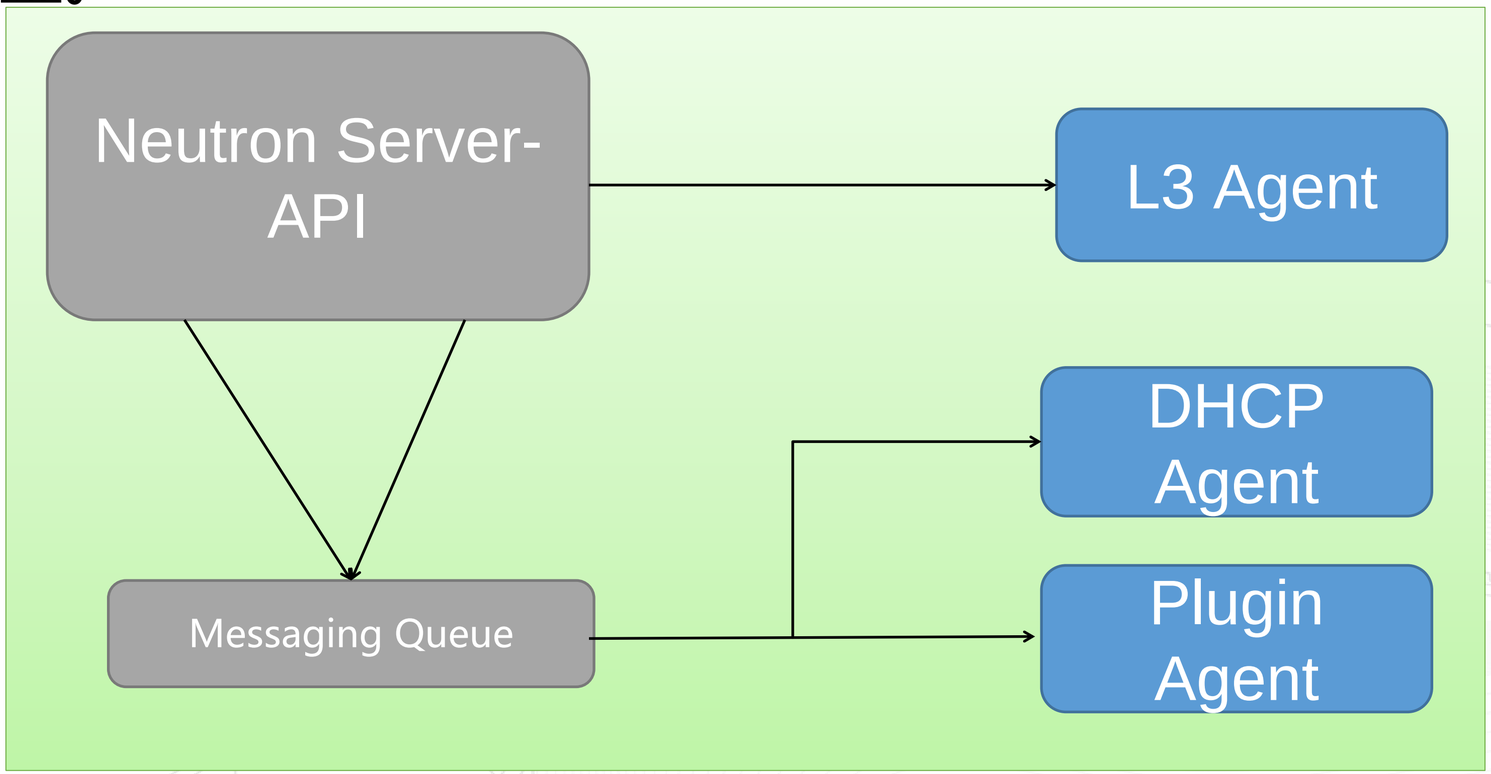
3、查询消息队列

```
[root@kolla kolla]# docker ps |grep rabb
be46fdf26b5c        kolla/centos-binary-rabbitmq:master
                    rabbitmq
[root@kolla kolla]# docker exec -i -t be46fdf26b5c  /bin/bash
(rabbitmq)[rabbitmq@kolla /]$ rabbitmqctl list_queues |grep neutron
neutron-vo-Port-1.6     0
neutron-vo-Network-1.1.kolla    0
neutron-vo-AddressGroup-1.0     0
neutron-vo-AddressGroup-1.0_fanout_f39347ec7c684d3db4b27d04e85f4f6f      0
neutron-vo-SubPort-1.0_fanout_3c6bce06b4974d2e8d05ebaec6d1f05a  0
neutron-vo-SecurityGroupRule-1.0_fanout_dc7e36abbece43a1be6d033df816d67c
neutron-vo-AddressGroup-1.0.kolla       0
neutron-vo-SubPort-1.0  0
neutron-vo-Subnet-1.1_fanout_aa021889582243759bb90ffb0fdae4a4    0
neutron-vo-SecurityGroupRule-1.0.kolla  0
neutron-vo-Port-1.6.kolla       0
neutron-vo-SecurityGroupRule-1.0        0
neutron-vo-Trunk-1.1_fanout_9804cd802c314aaa9766fb971dbe2349     0
neutron-vo-SecurityGroup-1.2_fanout_133e942b79674a9b843d8dd97d3becd4     0
neutron-vo-Trunk-1.1    0
neutron-vo-Network-1.1_fanout_c2f21ea99e4b4925b3a7bf3bda28bb1a  0
neutron-vo-Subnet-1.1.kolla     0
neutron-vo-SecurityGroup-1.2    0
neutron-vo-SecurityGroup-1.2.kolla      0
neutron-vo-SubPort-1.0.kolla    0
neutron-vo-Subnet-1.1   0
neutron-vo-Network-1.1  0
neutron-vo-Port-1.6_fanout_e3b28f4848fe4735a8948942a0ffdfc0      0
neutron-vo-Trunk-1.1.kolla      0
```

网络组件详解

- 网络比较特殊，不是以组件的形式，而是以驱动插件的形式开展的；根据不同的场景选择不同的驱动插件，可以借鉴。

模块	功能
neutron-server	API服务
neutron-*(l2)-agent	网桥、安全组等
neutron-dhcp-agent	Dhcp服务
neutron-l3-agent	Router/防火墙
neutron-vpn-agent	Router/防火墙/vpn
neutron-lbaas-agent	负载均衡
neutron-metadata-agent	Metadata代理服务



网络的组件化

网络组件化特点

复杂，灵活，可以依据不同的环境进行配置，主要以插件agent的形式来添加：

L2(vlan/

```
[root@kolla kolla]# vi neutron-l3-agent/l3_agent.ini
[DEFAULT]
agent_mode = legacy

[agent]

[ovs]
ovsdb_connection = tcp:127.0.0.1:6640
```

```
[root@kolla kolla]# vi neutron-openvswitch-agent/openvswitch_agent.ini
[agent]
tunnel_types = vxlan
l2_population = true
arp_responder = true

[securitygroup]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver

[ovs]
bridge_mappings = physnet1:br-ex
datapath_type = system
ovsdb_connection = tcp:127.0.0.1:6640
local_ip = 192.168.226.128
```

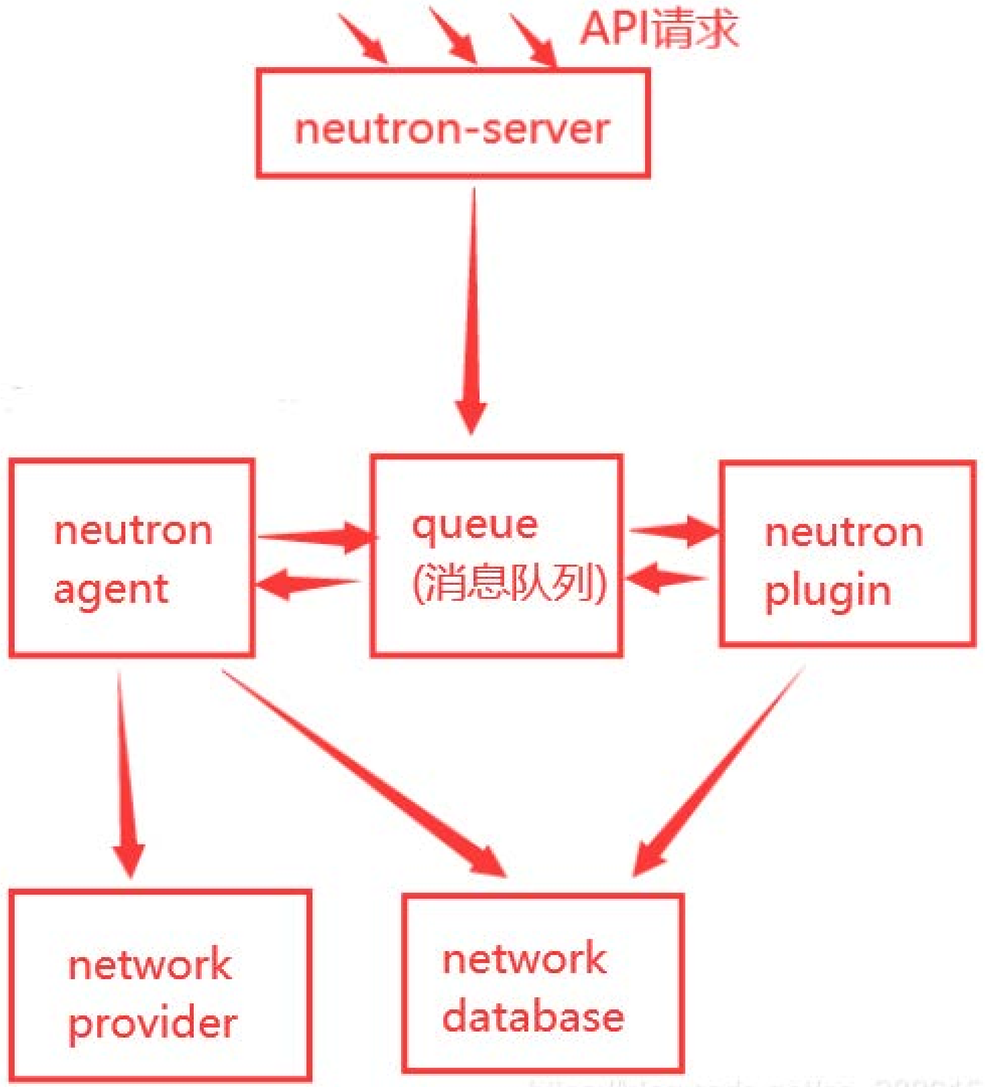
网络组件详解

OpenStack组件化

组件内做拆分

Nova:

Nuetron:



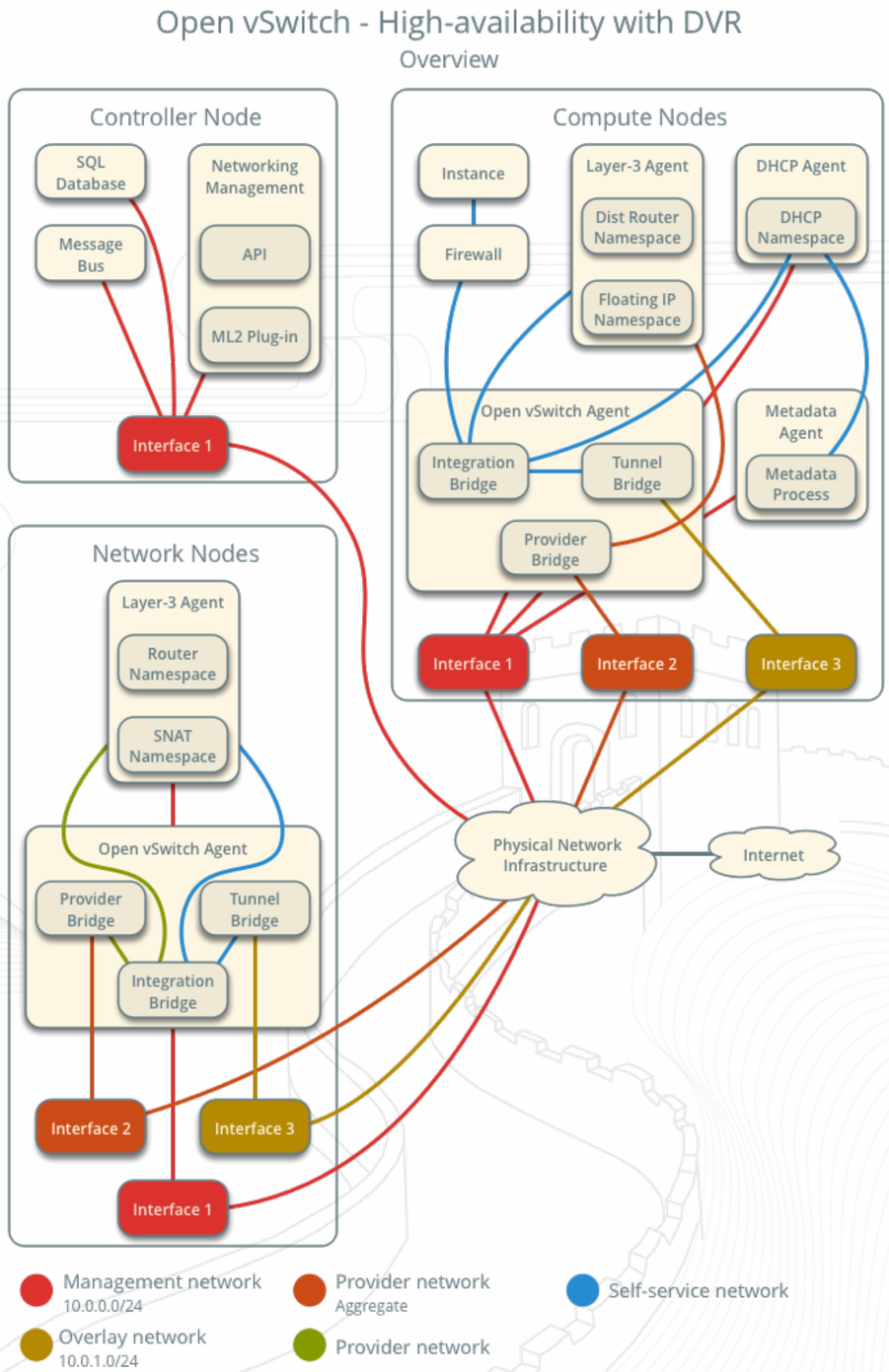
网络的组件化

OpenStack组件化

组件内做拆分

Nova:

Nuetron:



其它模块的组件化及小结

- **Cinder组件**

API service：负责接受和处理Rest请求，本质上就是一个http servier。

Scheduler service: 创建volume时，按照预定策略选择合适的Volume Service节点来执行任务，本质上就是一个rpc server。

Volume service: 该服务连接到后端存储，管理volumes ，本质上就是一个rpc server 。

Backup service: 该服务用来对volume进行备份，本质上就是一个rpc server 。

- **Glance组件**

Glance API：处理API请求

Glance Registry： 处理镜像的metadata存储

Store Adapter： 镜像本身的存储

小结：通过OpenStack的代码讲解了体现组件化的好处，组件之间通信通过Rest API，组件内部通过消息队列，对经常变化的模块可以采用驱动插件的形式。结论：在软件开发中一定采用组件化的开发方法。

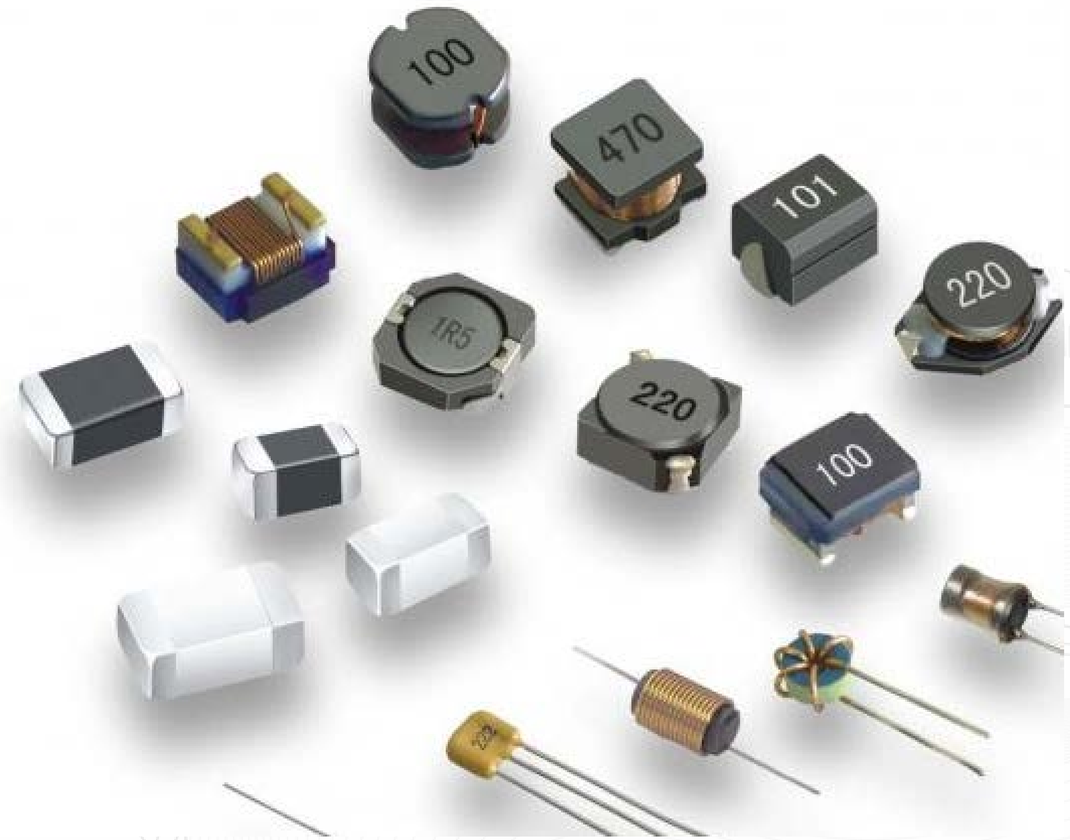
组件化的好处

为什么要组件化、模块化，项目存在问题：

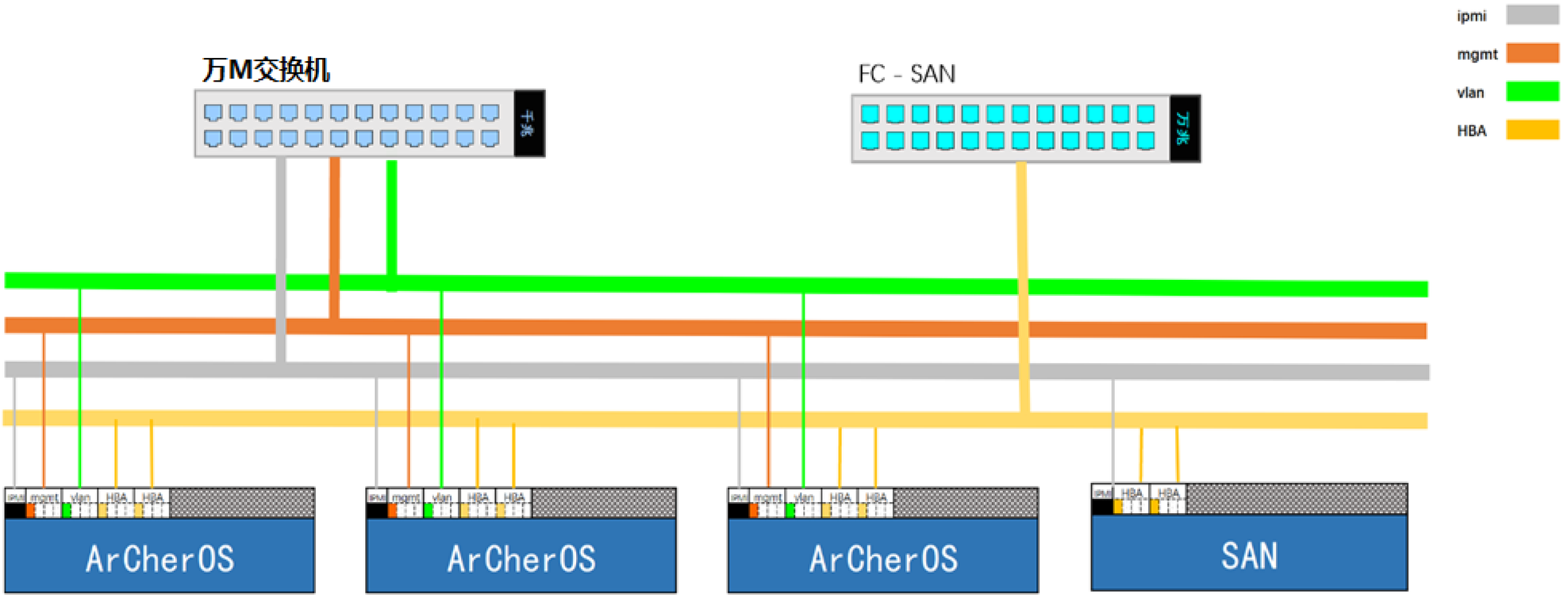
- 代码量大，耦合严重
- 编译慢，效率低
- 业务开发分工不明确，开发人员要关心非业务的代码
- 改代码时，可能会影响其他业务，牵一发动全身

组件化优点：

- 架构更清晰，解耦
- 加快编译速度
- 业务分工明确，开发人员仅专注与自己的业务
- 提高开发效率
- 组件、业务独立更新版本，可回滚，持续集成



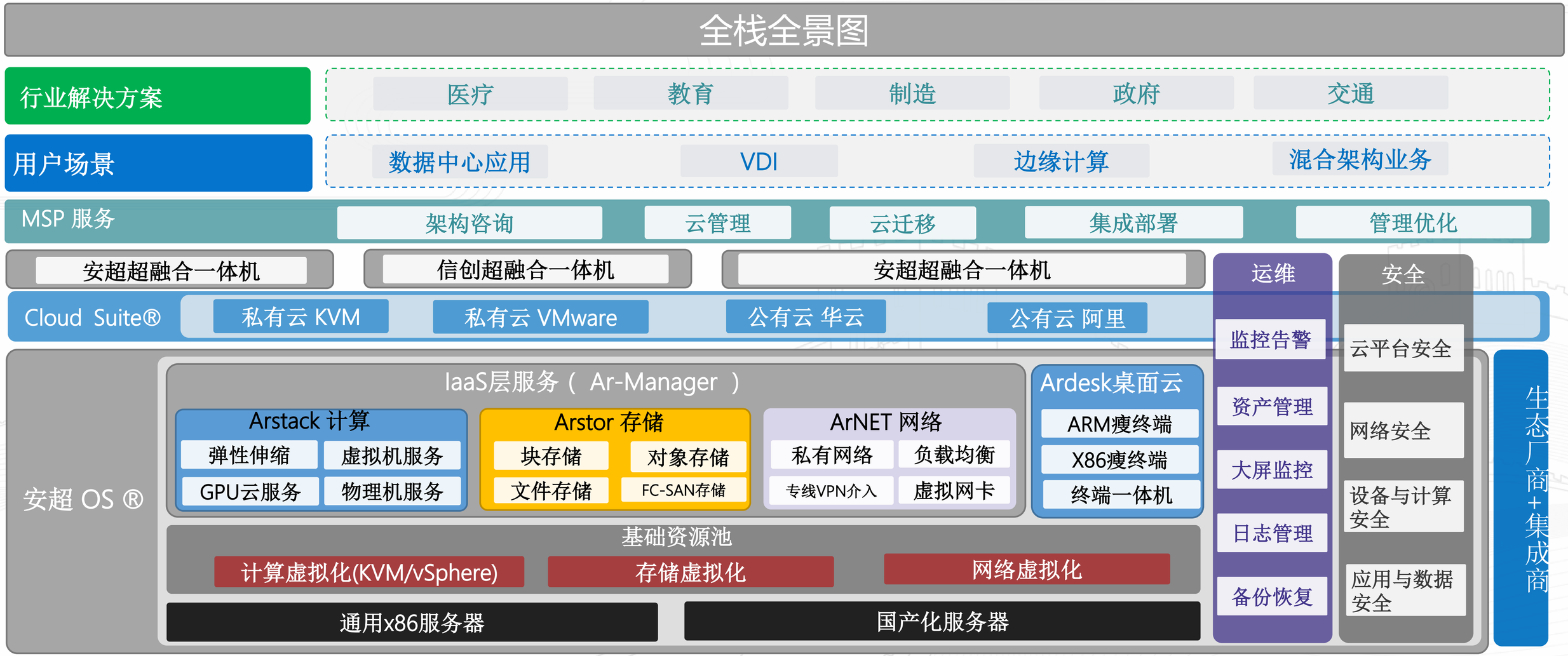
01信创云部署典型拓扑



本方案启用业务融合方式承载业务:
【管理】业务<橙色>标注,通过eth1千兆端口承载
【虚拟网络】业务<绿色>标注,通过eth2千兆端口承载
【存储】业务<黄色>标注,通过HBA,光纤端口

节点、设置*角色*，至少3个控制节点，3个存储和计算融合节点

01 信创云全栈全景图


全栈全景图
行业解决方案
医疗
教育
制造
政府
交通
用户场景
数据中心应用
VDI
边缘计算
混合架构业务
MSP 服务
架构咨询
云管理
云迁移
集成部署
管理优化
安超超融合一体机
信创超融合一体机
安超超融合一体机
Cloud Suite®
私有云 KVM
私有云 VMware
公有云 华云
公有云 阿里
安超 OS ®
IaaS层服务（ Ar-Manager ）
Arstack 计算
弹性伸缩
虚拟机服务
GPU云服务
物理机服务
Arstor 存储
块存储
对象存储
文件存储
FC-SAN存储
ArNET 网络
私有网络
负载均衡
专线VPN介入
虚拟网卡
Ardesk桌面云
ARM瘦终端
X86瘦终端
终端一体机
基础资源池
计算虚拟化(KVM/vSphere)
存储虚拟化
网络虚拟化
通用x86服务器
国产化服务器
运维
监控告警
资产管理
大屏监控
日志管理
备份恢复
安全
云平台安全
网络安全
设备与计算安全
应用与数据安全
生态厂商+集成商

二、容器与OpenStack的Kolla项目

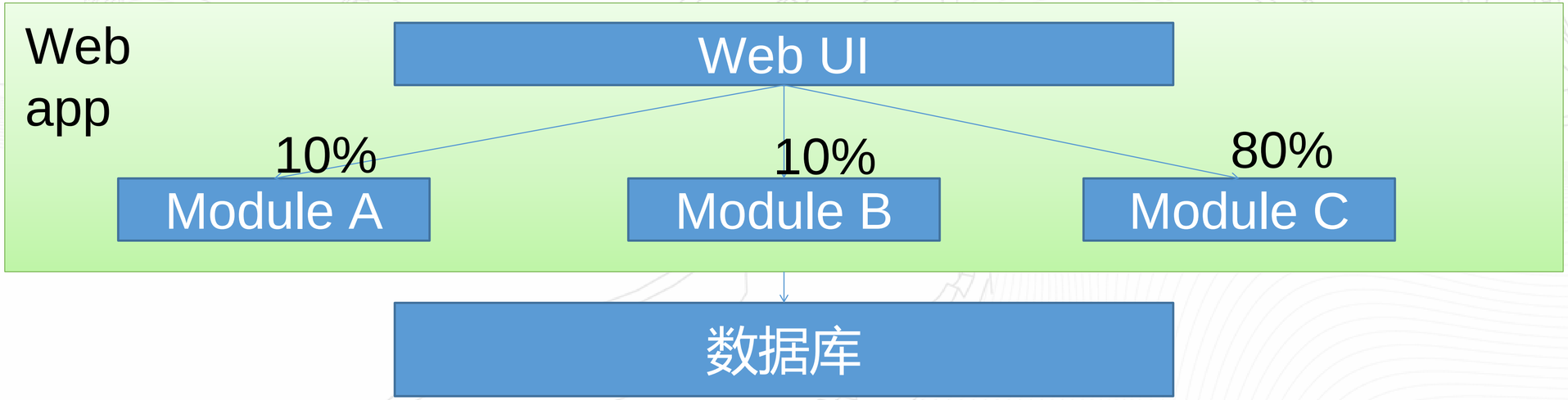
微服务与传统服务区别

传统的WEB应用核心分为业务逻辑、适配器以及API或通过UI访问WEB界面。其业务逻辑定义业务流程、业务规则以及各领域实体。类似打车软件、叫餐软件。尽管也是遵循模块化开发，但最终它们会打包并部署为单体式应用居多。例如Java应用程序会被打包成WAR，部署在Tomcat。之间存在优缺，但是随着市场的发展，用户数、数据量、业务模式的增长，传统模式存在一些缺点：

1、部署不灵活：构建时间长，任何小修改必须重新构建整个项目；

2、稳定性不高：一个对象内存释放不及时等问题，可能会导致整个应用挂掉；

3、扩展性不够：高并发情况下的业务需求，在扩展性方面比较麻烦；



微服务与传统服务区别

现在主流的设计一般会采用微服务架构。其思路不是开发一个巨大的单体式应用，而是将应用分解为小的、互相连接的微服务，每个业务逻辑都被分解为一个微服务，微服务之间通过REST API通信。微服务优点，比传统的应用程序更有效地利用计算资源。这是因为它们通过扩展组件来处理功能瓶颈问题。

- 一种软件架构模式
- 复杂应用解耦为小而众的服务
- 各服务精而专
- 服务间通信通过API完成
- 更快且更容易更新等

而为了满足如上几大优点或者说规则微服务细化带来工作维护麻烦性等，一般使用**docker**技术结合来提供微服务的优势和运维工作的高效性。

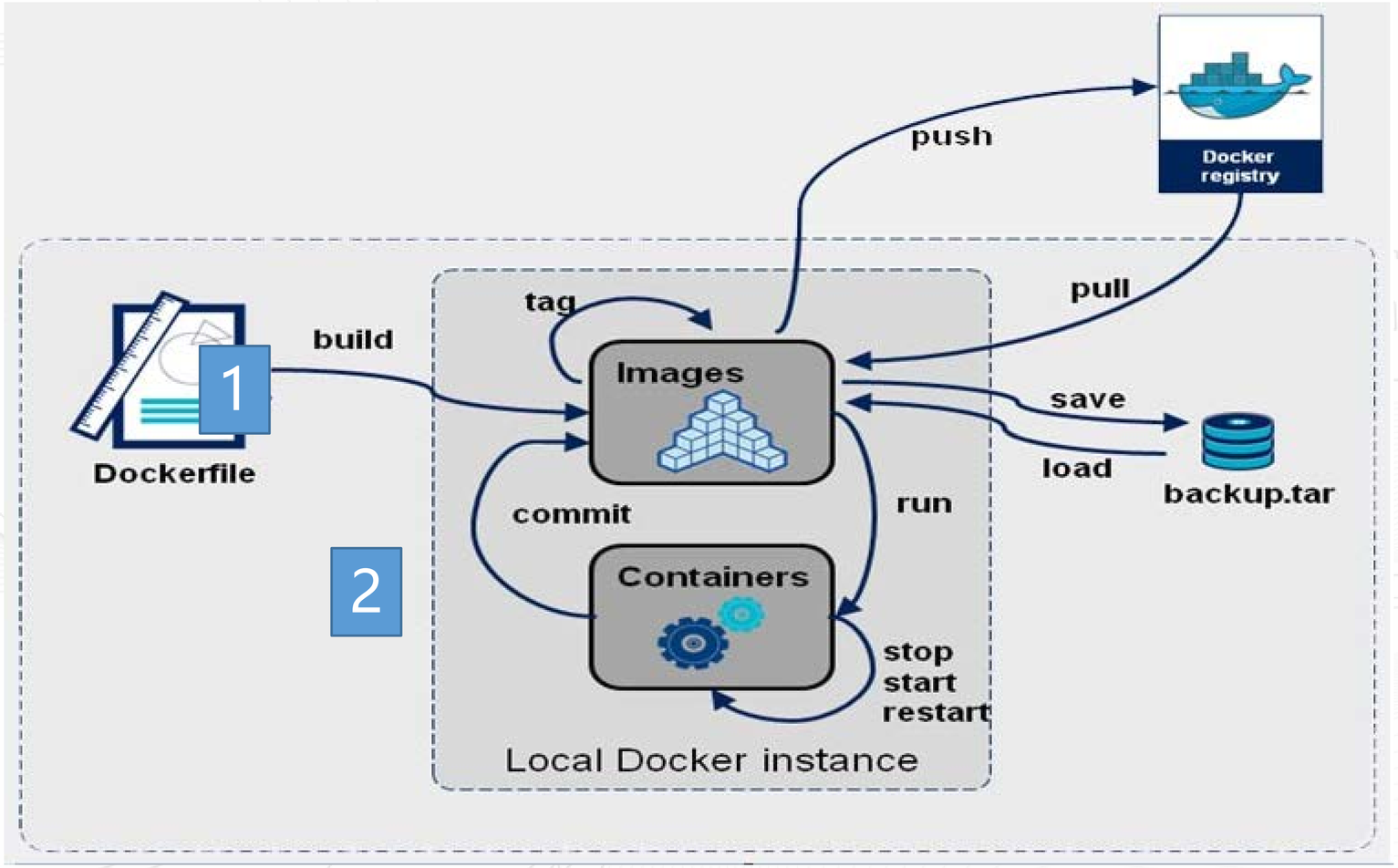
Docker重要概念&生命周期

Docker*重要概念包括*:

- 镜像 image
- 容器 Container
- 仓库 Repository

Docker生命周期包括:

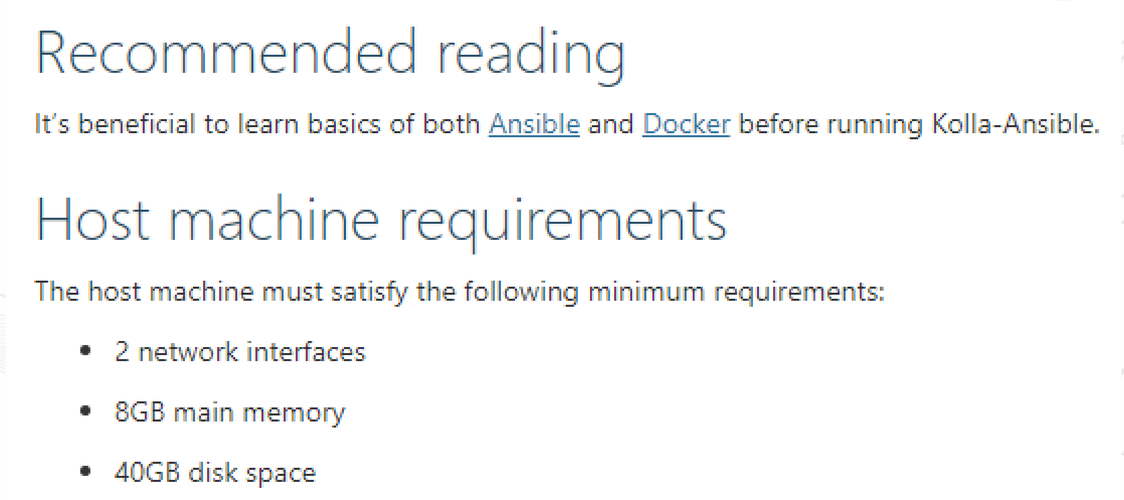
- 容器的启动
- 容器的停止
- 容器的关闭

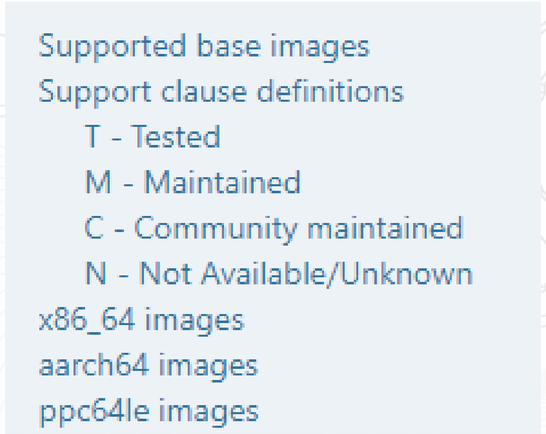


OpenStack组件的微服务化项目（kolla）

Kolla是OpenStack下用于自动化部署的一个项目，它基于docker和ansible来实现，其中docker主要负责镜像制作和容器管理，ansible主要负责环境的部署和管理。

Kolla实际上分为两部分：Kolla部分提供了生产环境级别的镜像，涵盖了OpenStack用到的各个服务；Kolla-ansible部分提供了自动化的部署。最开始这两部分是在一个项目中的（即Kolla），OpenStack从O开头的版本开始被独立开来，这才有了用于构建所有服务镜像的Kolla项目，以及用于执行自动化部署的Kolla-ansible。

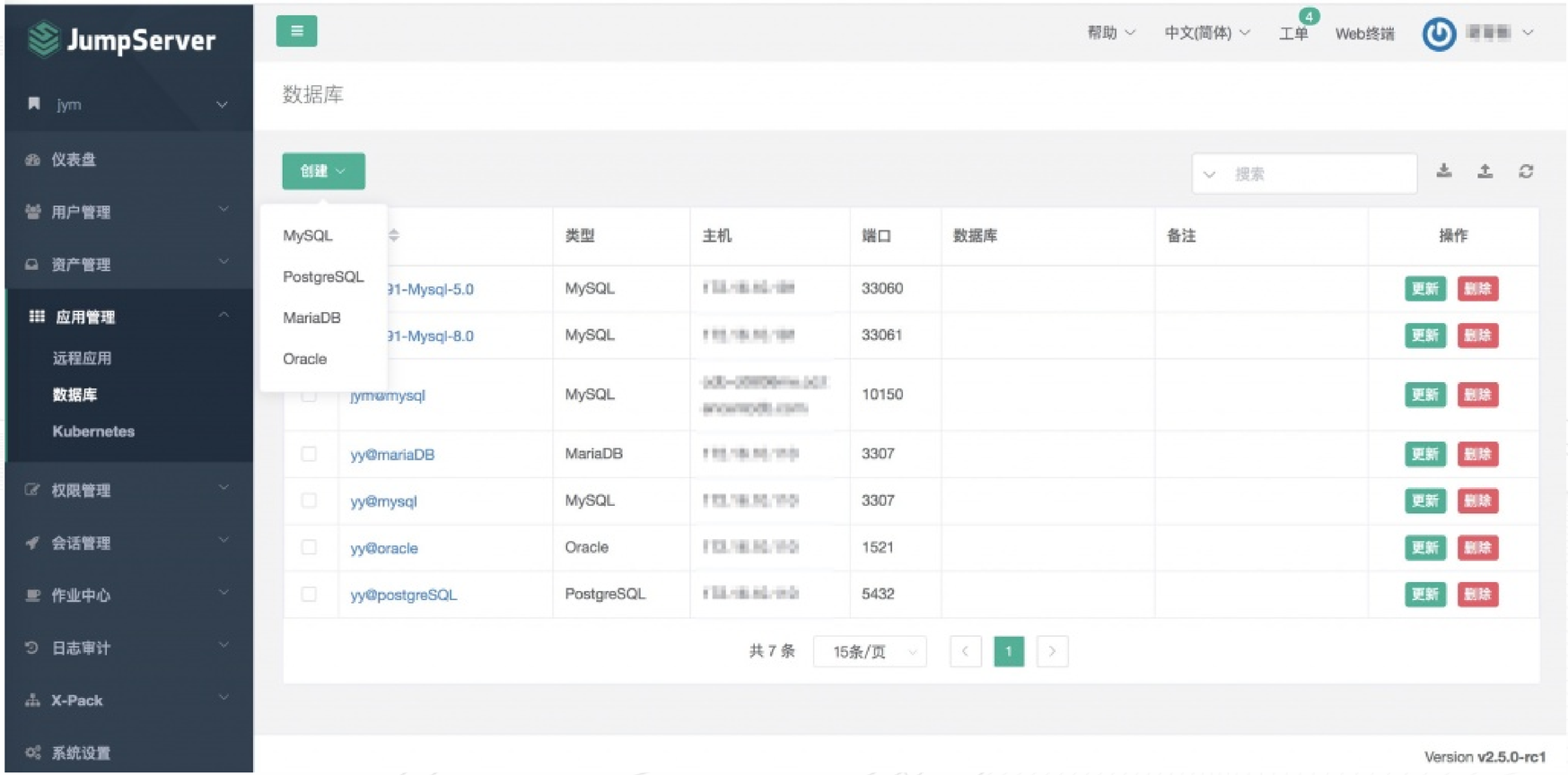




https://docs.openstack.org/kolla/latest/support_matrix

微服务化交付案例

仅需两步快速安装 JumpServer：
1、准备一台 4核8G （最低）且可以访问互联网的 64 位 Debian 主机；
2、以 root 用户执行如下命令一键安装 JumpServer：
curl -sSL https://github.com/jumpserver/jumpserver/releases/download/v2.6.0/quick_start.sh | sh



微服务小结

Docker是一个开源应用容器（当然目前也分为CE和EE版本，不完全开源化，也存在收费版本），让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的Linux机器上，也可以实现虚拟化。容器是完全使用沙箱机制，相互之间不会有任何接口。

Docker作为容器工具可以把业务逻辑容器、数据库容器、储存容器、队列容器使得软件可以拆分成若干个标准化容器，然后像搭积木一样组合起来，让彼此通信，从而形成微服务。

因此微服务很适合用Docker容器实现，每个容器承载一个服务，一台计算机同时运行多个容器，从而就能很轻松地模拟出复杂的微服务架构。

微服务化后可以通过Kubernetes等技术提供应用运行平台，从而实现运维自动化，快速部署应用、弹性伸缩和动态调整应用环境资源，提高研发运营效率。

华云数据、时速云、灵雀云、Openshift、Rancher

三、常用工具与云平台的实操

工具

虚拟化工具：VMware Workstation Pro

远程管理工具：SecureCRT 8.5

测试工具：restclient-ui-3.5-jar-with-dependencies

远程桌面工具：TeamViewer

玩转VMware Workstation Pro

1、查看电脑配置情况

2、VMware Workstation Pro

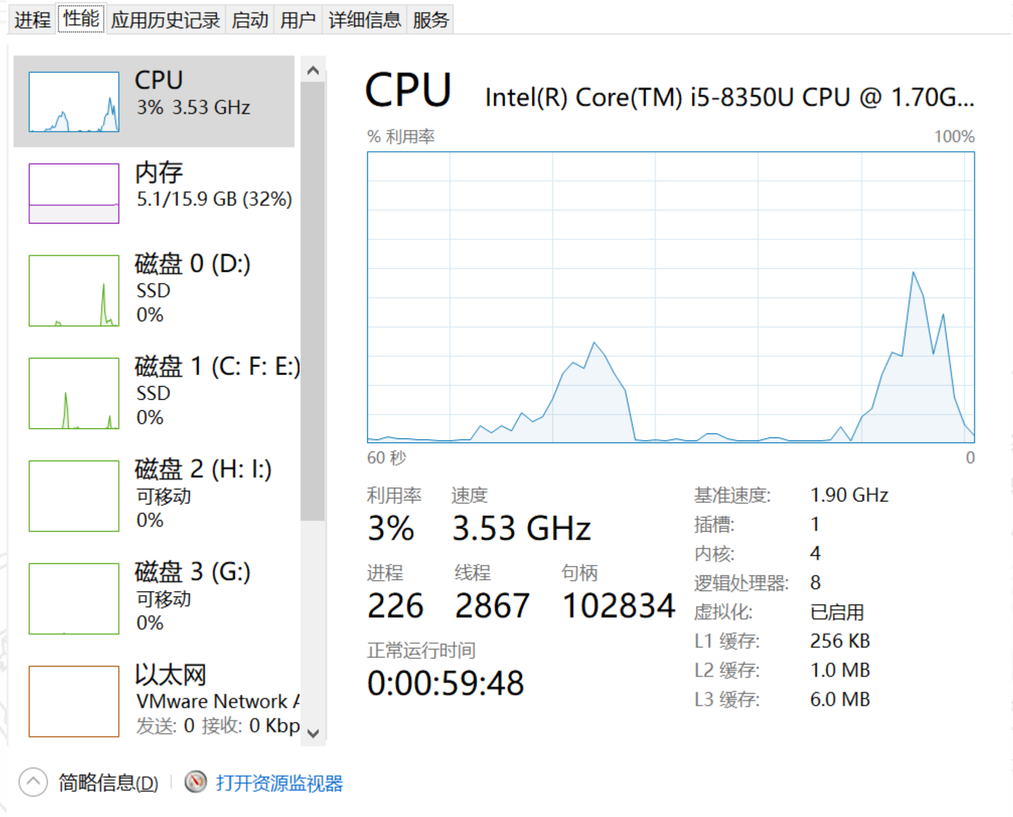
3、CentOS系统

4、配置网络（回环网卡）

windwos键+R 打开运行，输入hdwwiz

5、远程管理工具（专业管理工具）

SecureCRT 8.5



脚本文件

- 部署完成后，运行以下命令可以生成一个openrc文件
- kolla-ansible post-deploy
- 界面下载RC文件再传到虚拟机中

```
[root@control01 ~]# source admin-openrc.sh
Please enter your OpenStack Password for project admin as user admin:
[root@control01 ~]#
```

简单验证环境

- 执行脚本
- openstack service list
- openstack endpoint list
- openstack network agent list
- openstack compute service list

```
[root@control01 ~]# openstack network agent list
+--------------------------------------+--------------------+----------+-------------------+-------+-------+---------------------------+
| ID                                   | Agent Type         | Host     | Availability Zone | Alive | State | Binary                    |
+--------------------------------------+--------------------+----------+-------------------+-------+-------+---------------------------+
| 5605acda-b23f-420b-bf0a-2b68a17c17cc | Metadata agent     | control01 | None              | True  | UP    | neutron-metadata-agent    |
| 6d06f22a-86f0-472a-a80a-e749dd930f9a | L3 agent           | control01 | nova              | True  | UP    | neutron-l3-agent          |
| 7389ad2b-0e57-4933-bf69-0fe35b07afb3 | DHCP agent         | control01 | nova              | True  | UP    | neutron-dhcp-agent        |
| fbde0437-d177-4033-9b9b-da9228eebbec | Open vSwitch agent | control01 | None              | True  | UP    | neutron-openvswitch-agent |
+--------------------------------------+--------------------+----------+-------------------+-------+-------+---------------------------+
[root@control01 ~]# openstack compute service list
+----+------------------+-----------+----------+---------+-------+----------------------------+
| ID | Binary           | Host      | Zone     | Status  | State | Updated At                 |
+----+------------------+-----------+----------+---------+-------+----------------------------+
|  1 | nova-scheduler   | control01 | internal | enabled | up    | 2018-04-05T03:36:02.000000 |
|  5 | nova-conductor   | control01 | internal | enabled | up    | 2018-04-05T03:36:06.000000 |
|  6 | nova-consoleauth | control01 | internal | enabled | up    | 2018-04-05T03:36:07.000000 |
|  7 | nova-compute     | control01 | nova     | enabled | up    | 2018-04-05T03:36:09.000000 |
+----+------------------+-----------+----------+---------+-------+----------------------------+
```

创建网络（外网）

- openstack network create --external --provider-physical-network physnet1 \
- --provider-network-type flat public1
- openstack subnet create --no-dhcp \
- --allocation-pool ${EXT_NET_RANGE} --network public1 \
- --subnet-range ${EXT_NET_CIDR} --gateway ${EXT_NET_GATEWAY}
- public1-subnet

创建网络（内网）

- openstack network create --provider-network-type vxlan demo-net
- openstack subnet create --subnet-range 10.0.0.0/24 --network demo-net \
- --gateway 10.0.0.1 --dns-nameserver 8.8.8.8 demo-subnet

创建路由并绑定接口

- openstack router create demo-router
- openstack router add subnet demo-router demo-subnet
- openstack router set --external-gateway public1 demo-router

上传镜像

- openstack image create --disk-format qcow2 --container-format bare --public \
- --file ./${IMAGE} ${IMAGE_NAME}

创建模版

- openstack flavor create --id 1 --ram 512 --disk 1 --vcpus 1 m1.tiny
- openstack flavor create --id 2 --ram 2048 --disk 20 --vcpus 1 m1.small
- openstack flavor create --id 5 --ram 16384 --disk 160 --vcpus 8 m1.xlarge

创建vm

- DEMO_NET_ID=$(openstack network list | awk '/ demo-net / {print $2}')
- openstack server create \\
- --image ${IMAGE_NAME} \\
- --flavor m1.tiny \\
- --nic net-id=${DEMO_NET_ID} \\
- demo1

防火墙的配置

- openstack project list | awk '/ admin / {print $2}'
- f8d3113e6d8d43c394a387f91a2cba7e
- openstack security group list --project ${ADMIN_PROJECT_ID} | awk '/ default / {print $2}‘
- openstack security group list --project f8d3113e6d8d43c394a387f91a2cba7e | awk '/ default / {print $2}'
- 3b63a919-a977-4581-9273-8eb5676cc2a9
- openstack security group rule create --ingress --ethertype IPv4 \
- --protocol icmp ${ADMIN_SEC_GROUP}
- openstack security group rule create --ingress --ethertype IPv4 \
- --protocol tcp --dst-port 22 ${ADMIN_SEC_GROUP}

验证网络

- openstack server list
- ip netns
- ip netns exec qdhcp-2bc929d9-a012-4efa-af35-7365dad7ecce ping 10.0.0.15

```
[root@control01 tools]# openstack server list
+--------------------------------------+------+--------+----------------------------+------------+
| ID                                   | Name | Status | Networks                   | Image Name |
+--------------------------------------+------+--------+----------------------------+------------+
| ad798bfc-4251-408a-9946-bce2f8d25139 | a    | ACTIVE | 10=10.0.0.15, 192.168.0.17 | cirros     |
+--------------------------------------+------+--------+----------------------------+------------+
```

```
[root@control01 tools]# ip netns exec qdhcp-2bc929d9-a012-4efa-af35-7365dad7ecce ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
17: taped4e164c-bd: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN qlen 1000
    link/ether fa:16:3e:30:54:77 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.10/24 brd 10.0.0.255 scope global taped4e164c-bd
       valid_lft forever preferred_lft forever
    inet 169.254.169.254/16 brd 169.254.255.255 scope global taped4e164c-bd
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe30:5477/64 scope link
       valid_lft forever preferred_lft forever
```

命令行登陆

- ip netns exec qdhcp-2bc929d9-a012-4efa-af35-7365dad7ecce ssh cirros@10.0.0.15

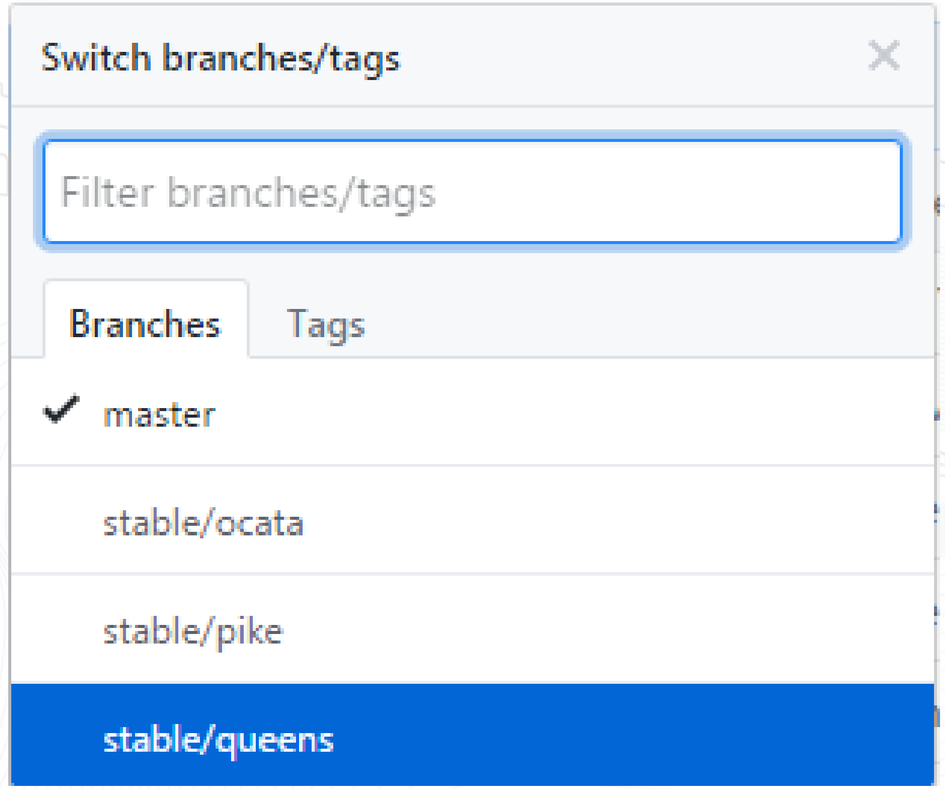
```
[root@control01 tools]# ip netns exec qdhcp-2bc929d9-a012-4efa-af35-7365dad7ecce ssh cirros@10.0.0.15
Warning: Permanently added '10.0.0.15' (RSA) to the list of known hosts.
cirros@10.0.0.15's password:
$ ls
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc pfifo_fast qlen 1000
    link/ether fa:16:3e:de:ab:c6 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.15/24 brd 10.0.0.255 scope global eth0
    inet6 fe80::f816:3eff:fede:abc6/64 scope link
       valid_lft forever preferred_lft forever
$
```

初始化脚本

- /root/kolla-ansible-4.0.3.dev36/tools/init-runonce
- 需要编辑外网的子网
  - EXT_NET_CIDR='10.0.2.0/24'
  - EXT_NET_RANGE='start=10.0.2.150,end=10.0.2.199'
  - EXT_NET_GATEWAY='10.0.2.1'
- 只能执行一次
- cd /root/kolla-ansible-4.0.3.dev36/tools/
- ./init-runonce

代码的下载

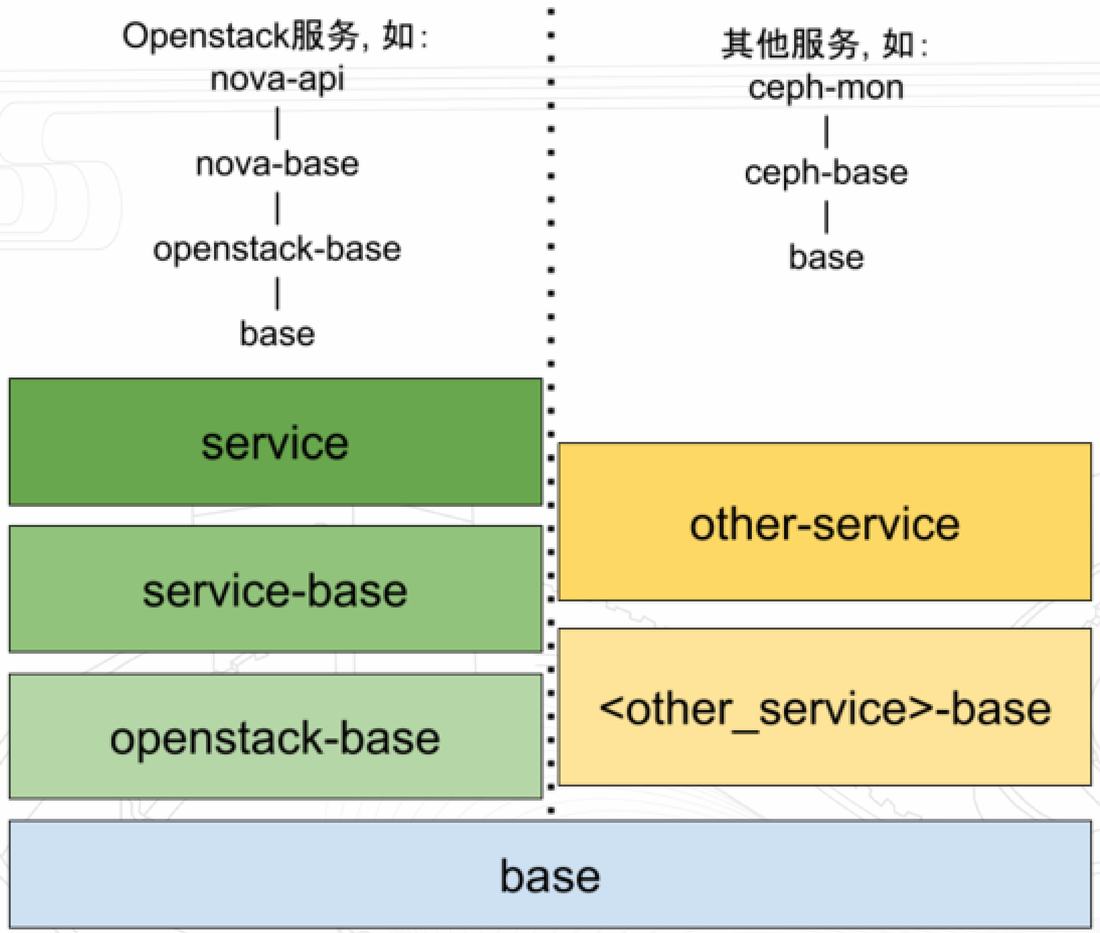
- kolla项目只用来做docker images build
- http://www.github.com/openstack/kolla
- 部署工作由kolla-ansible
- https://github.com/openstack/kolla-ansible
- 下载代码
- git clone 地址 –b 分支



分为四层

分为四层

- base
- openstack-base
- <service>-base
- <service>



Yaml文件

- YML文件格式是YAML(YAML Aint Markup Language)编写的文件格式，YAML是一种直观的能够被电脑识别的数据序列化格式，容易和脚本语言交互的，可以被支持YAML库的不同的编程语言程序导入，比如： C/C++, Ruby, Python, Java, Perl, C#, PHP等。在all.yml作用是提供文件目录、以及各种配置信息（如IP地址、端口号、进程id等等）

.json.j2

- Jinja2是python的一种模板语言，以Django的模板语言为原本，和Django的模板语言有很多相似之处，同时Jinja本身也是一种系统的、完整的Python模板语言。

开发工具

- pycharm
- vim
- java

踩坑记录

- 1、Win10 nvme磁盘建议 VMware Workstation Pro16的版本
- 2、开启Vmware的虚拟化
- 3、网络及接口的配置
- 4、 pip install python3-openstackclient
- pip install python-openstackclient

个人建议篇

研究生的三年怎么过?

1、研究生是一个全新的开始;

2、以什么样的心态开始新的生活;

3、问的问题一定要经过自己的思考之后再问。

- 123640869@qq.com

03江苏省应用创新培训中心介绍

江苏省信息技术应用创新培训中心由江苏信创技术适配攻关基地有限公司牵头，联合江苏省信息技术应用创新测试中心、华云数据控股集团有限公司、蓝深远望科技股份有限公司、江苏新世纪信息科技有限公司、江苏软件产业人才发展基金会共同建设运营。

指导单位：电子工业标准化研究院（电子四院）、无锡市工业和信息化局、无锡市委机要局。

培训中心将携手信创产业生态厂商，进行广泛的社会和市场调研，以市场需求为导向，开展分阶段，分层次的培训、实践和推广，为我省信创产业快速稳定发展输送人才。

江苏省信息技术应用创新培训中心

共建单位	入园企业	指导单位
6家	40家	3家