

容器交付OpenStack



OpenStack交付工具

红帽RDO和TripleO (puppet)

Ubuntu Maas和juju (juju)

Suse Crowbar (Chef)

Mirantis Fuel (puppet)

HP Hellion (ansible)

Rackspace (chef, salt, ansible)

IBM PowerVC (shell)

Devstack (shell)

配置管理工具

Puppet 市场占有率很高

Chef 最早支持OpenStack的工具，基本要死掉

Ansible 发展速度的最快，OpenStack项目里也包含

Salt 可能会东山再起

面临挑战

升级很困难

项目太多，半年周期太短

OpenStack很复杂，

Puppet学习曲线陡峭

维护代价很高

依赖厂商很严重

新项目支持不足

企业需求

Cost（成本）

Security（安全）

Reliability（可靠）

Scalability（扩展）

Flexibility（灵活）

How to choose
OpenStack
vendors?

What is different
between
vendors?



How to Avoid
vendor lock-in?

Simple solution to complex problem

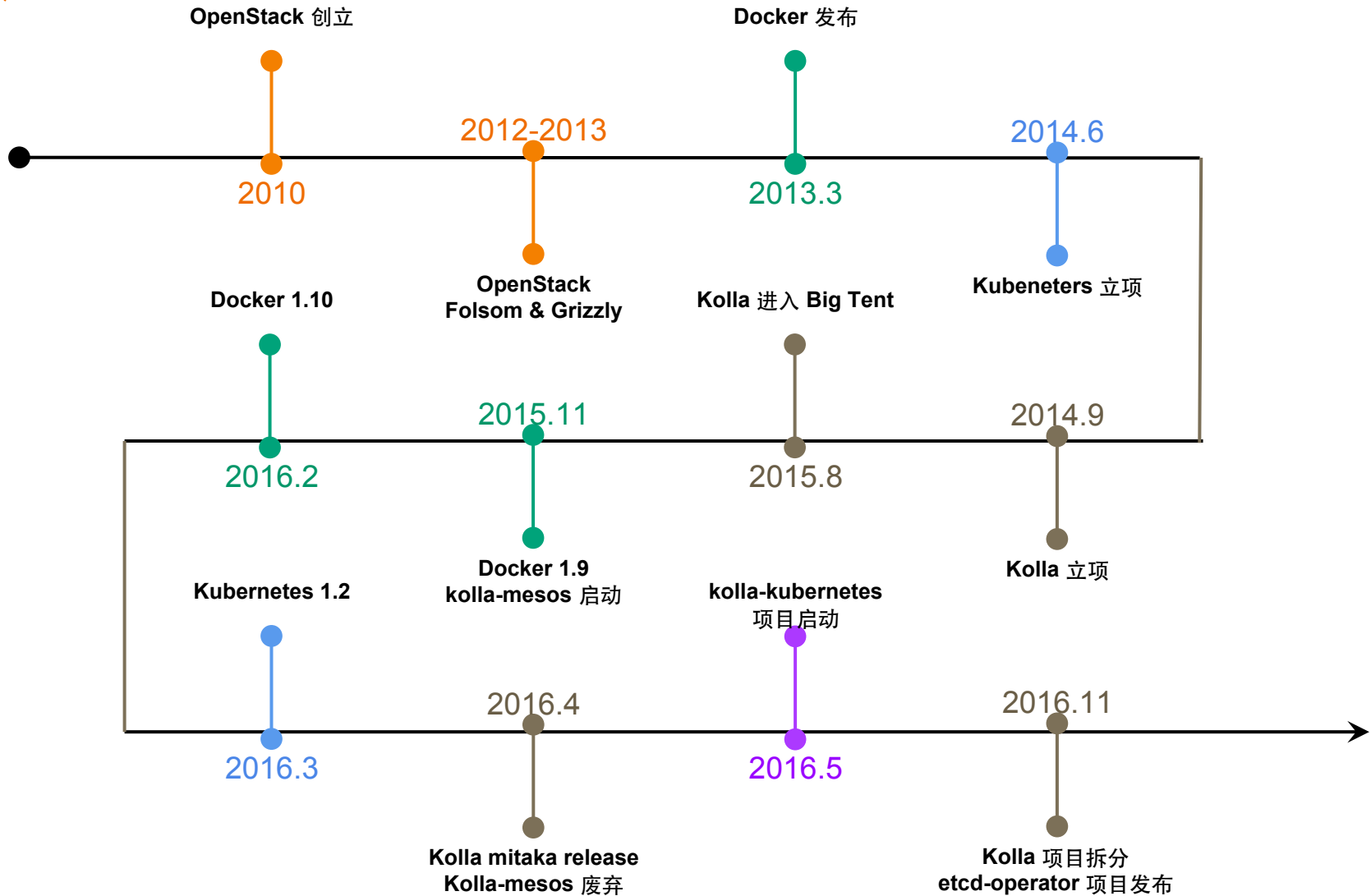
Multiplicity of Goods



Multiplicity of methods for transporting/storing



机遇



容器交付OpenStack

Rackspace的OpenStack-ansible (LXC)

Kolla项目 (liberty, POC阶段)

TCP Cloud (salt+docker, kilo和liberty)

企业IT内部实现OpenStack容器化



IT大咖说
不止于技术

PAST



容器交付OpenStack

Kolla Mitaka版本，Product Ready

Rackspace的OpenStack-ansible遇到麻烦（LXC缺乏镜像管理）

红帽产品支持计算节点容器化验证（采用Kolla）

Mirantis收购TCP Cloud

Mirantis Fuel CCP（K8s管OpenStack）艰难推进

Kolla

- 孵化于 TripleO
- 创立于2014年9月
- 2015年8月加入OpenStack Big Tent
- Liberty 发布第一个版本
- Mitaka / Newton 版本加入了更多的功能
- 100%容器化



DaoCloud

IT大咖说
不止于技术

Layer 4: 应用

Octavia

Trove

Sahara

Zaqar

Murano

Layer 3: 服务

Mistral

Magnum

Cloudkitty

Manila

Solum

Heat

Tacker

Senlin

Freezer

Karbor

Layer 2: IaaS 扩展

Ironic

Designate

Barbican

Telemetry

Kuryr

Zun

Dragonflow

SearchLight

Monasca

Layer 1: 基础IaaS服务

Nova

Glance

Cinder

Keyston

Neutron

Swift

Horizon

Congress

Watch

Rally

Vitrage

OVS

Libvirt

Qemu

RabbitMQ

E (Fluentd) K

NTP

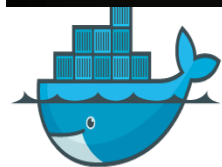
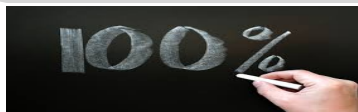
HAProxy

Keepalived

Ceph

MariaDB

MongoDB



docker



ANSIBLE



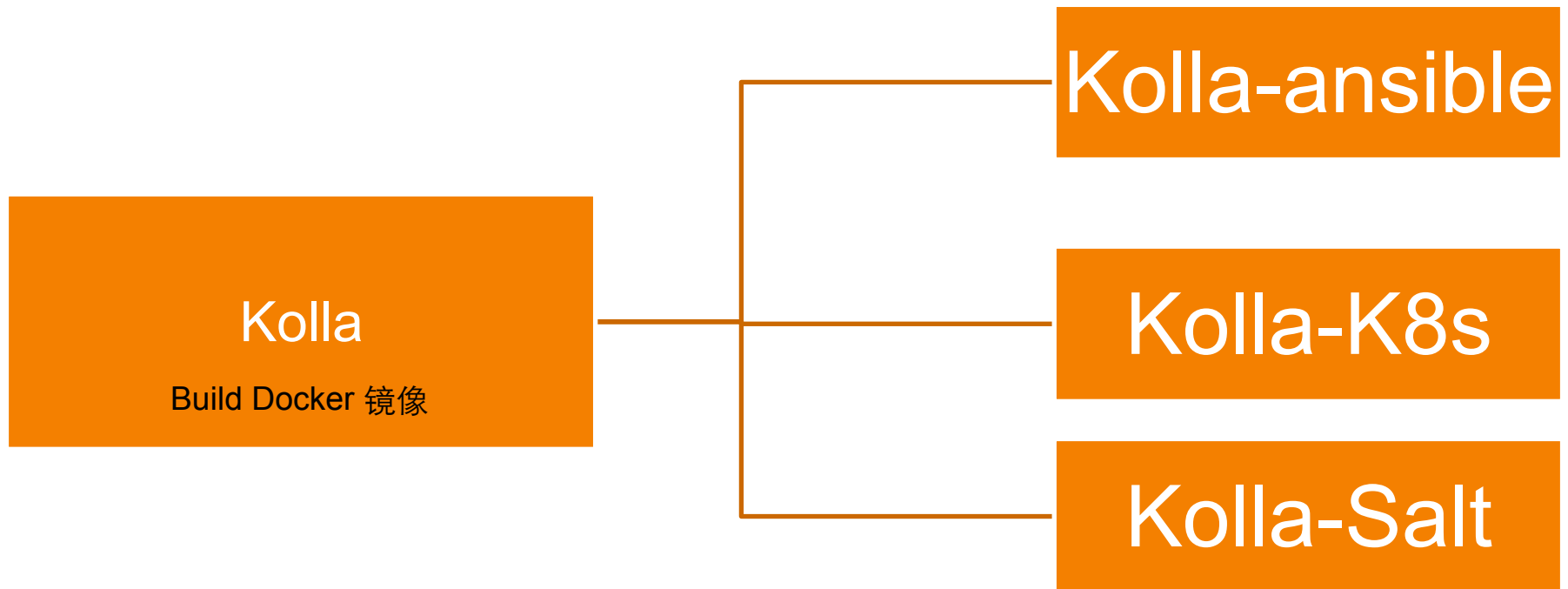
Kolla 的目标

- 简化部署和运维
- 提供
 - 生产级别的 Docker 镜像
 - 用于镜像部署的运维工具
- 在100台物理节点的规模下做到开箱即用

特性

- 所有服务的高可用
- 支持 ceph 做后端存储，glance/nova/cinder/gnocchi
- 支持多个 Linux 发行版
 - CentOS / OracleLinux / Ubuntu / RHEL
- 同时支持源码安装和二进制包安装
- 物理机上只依赖 Docker Engine 和 docker-py
- 支持原子升级

Kolla项目组成



实现

- 使用 Dockerfile + jinja2 构建镜像
- 使用镜像依赖
 - 更快的构建速度，更小的镜像
- 使用 Ansible 进行部署
- 容器化一切服务
 - libvirt / openvswitch / neutron
- 一个容器只有一个进程
- 使用 --net host 网络模型

实现

- 使用 Dockerfile + jinja2 构建镜像
- 使用镜像依赖
 - 更快的构建速度，更小的镜像
- 使用 Ansible 进行部署
- 容器化一切服务
 - libvirt / openvswitch / neutron
- 一个容器只有一个进程
- 使用 --net host 网络模型

Dockerfile.j2

```
FROM {{ namespace }}/{{ image_prefix }}nova-base:{{ tag }}
MAINTAINER {{ maintainer }}

{% if install_type == 'binary' %}
    {% if base_distro in ['centos', 'fedora', 'oraclelinux', 'rhel'] %}

RUN yum -y install openstack-nova-api \
    && yum clean all

    {% elif base_distro in ['ubuntu'] %}

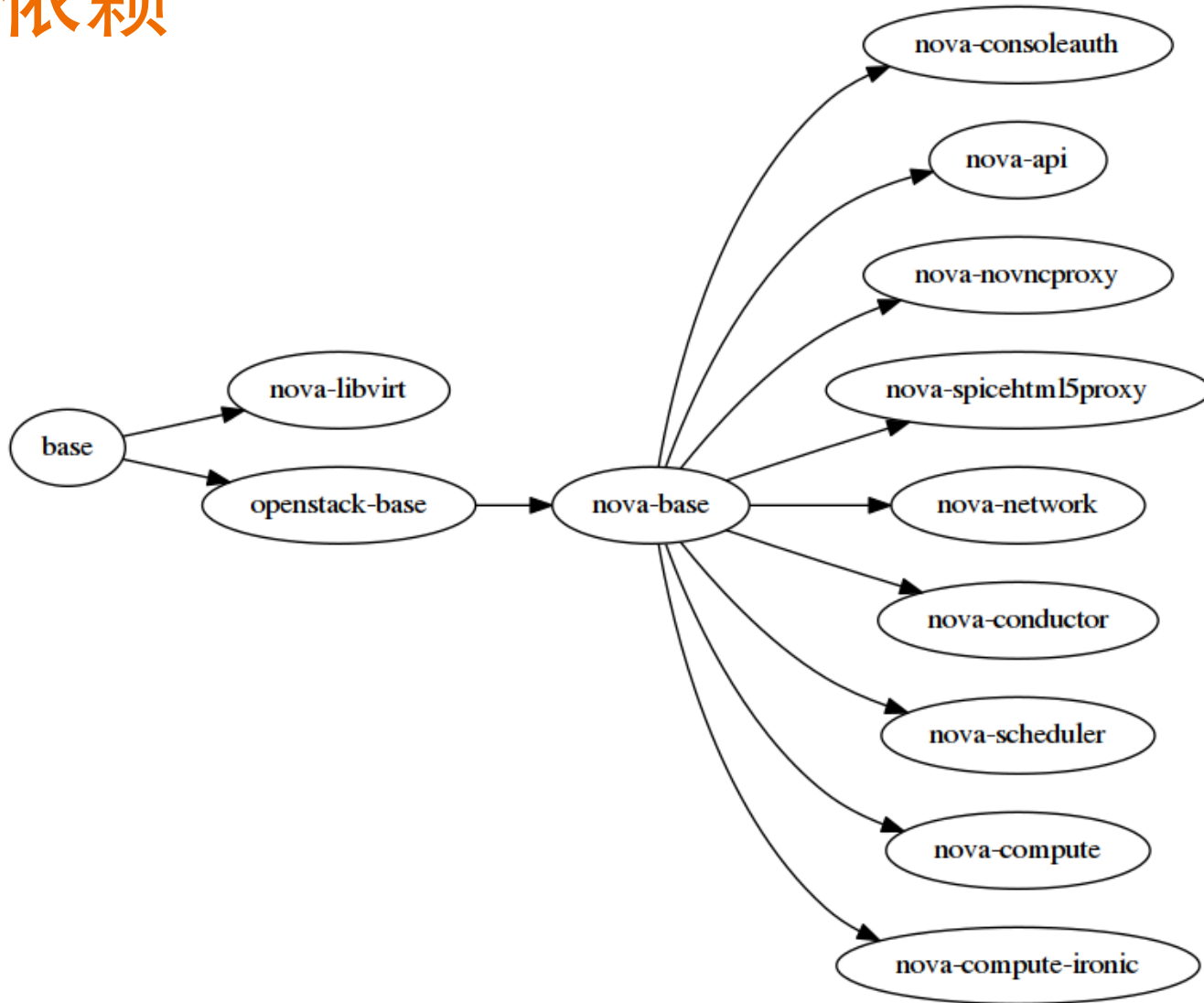
RUN apt-get install -y --no-install-recommends \
    nova-api \
    python-memcache \
    && apt-get clean

    {% endif %}
{% endif %}
```

实现

- 使用 Dockerfile + jinja2 构建镜像
- 使用镜像依赖
 - 更快的构建速度，更小的镜像
- 使用 Ansible 进行部署
- 容器化一切服务
 - libvirt / openvswitch / neutron
- 一个容器只有一个进程
- 使用 --net host 网络模型

镜像依赖



实现

- 使用 Dockerfile + jinja2 构建镜像
- 使用镜像依赖
 - 更快的构建速度，更小的镜像
- 使用 Ansible 进行部署
- 容器化一切服务
 - libvirt / openvswitch / neutron
- 一个容器只有一个进程
- 使用 --net host 网络模型

Kolla Deploy

使用 Ansible 做服务编排

提供 `kolla-ansible` 脚本

使用 Ansible Inventory 控制服务在哪些机器上启动

同时支持单机部署和多机部署

支持所有组件的高可用，包括 `rabbitmq` 和 `mariadb`

容器内非 `root` 用户运行，保证安全

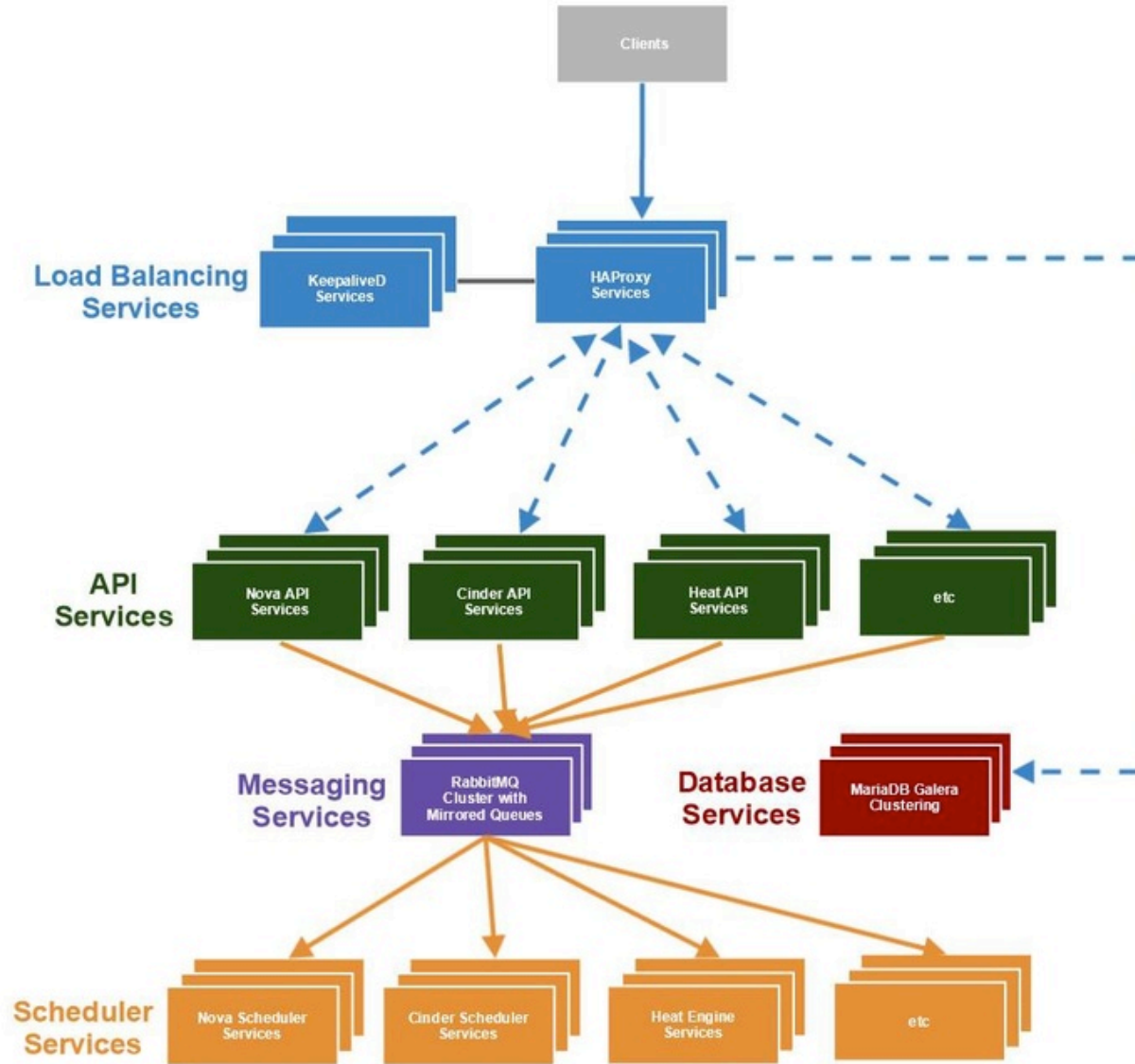
部分容器需要使用 `--privileged` 权限

支持 `kubernetes`

kolla-ansible

Commands :

prechecks	Do pre-deployment checks for hosts
mariadb_recovery	Recover a completely stopped mariadb cluster
bootstrap-servers	bootstrap servers with kolla deploy dependencies
destroy	Destroy Kolla containers, volumes and host configuration
deploy	Deploy and start all kolla containers
deploy-bifrost	Deploy and start bifrost container
deploy-servers	Enroll and deploy servers with bifrost
post-deploy	Do post deploy on deploy node
pull	Pull all images for containers
reconfigure	Reconfigure OpenStack service
certificates	Generate self-signed certificate for TLS *For Development Only*
upgrade	Upgrades existing OpenStack Environment
genconfig	Generate configuration files for enabled OpenStack services



实现

- 使用 Dockerfile + jinja2 构建镜像
- 使用镜像依赖
 - 更快的构建速度，更小的镜像
- 使用 Ansible 进行部署
- 容器化一切服务
 - libvirt / openvswitch / neutron
- 一个容器只有一个进程
- 使用 --net host 网络模型

容器化一切服务

Image	COMMAND	NAMES
centos-source-horizon:3.0.0	"kolla_start"	horizon
centos-source-neutron-metadata-agent:3.0.0	"kolla_start"	neutron_metadata_agent
centos-source-neutron-l3-agent:3.0.0	"kolla_start"	neutron_l3_agent
centos-source-neutron-dhcp-agent:3.0.0	"kolla_start"	neutron_dhcp_agent
centos-source-neutron-openvswitch-agent:3.0.0	"kolla_start"	neutron_openvswitch_agent
centos-source-neutron-server:3.0.0	"kolla_start"	neutron_server
centos-source-openvswitch-vswitchd:3.0.0	"kolla_start"	openvswitch_vswitchd
centos-source-openvswitch-db-server:3.0.0	"kolla_start"	openvswitch_db
centos-source-nova-compute:3.0.0	"kolla_start"	nova_compute
centos-source-nova-libvirt:3.0.0	"kolla_start"	nova_libvirt
centos-source-nova-conductor:3.0.0	"kolla_start"	nova_conductor
centos-source-nova-scheduler:3.0.0	"kolla_start"	nova_scheduler
centos-source-nova-novncproxy:3.0.0	"kolla_start"	nova_novncproxy
centos-source-nova-consoleauth:3.0.0	"kolla_start"	nova_consoleauth
centos-source-nova-api:3.0.0	"kolla_start"	nova_api
centos-source-glance-api:3.0.0	"kolla_start"	glance_api
centos-source-glance-registry:3.0.0	"kolla_start"	glance_registry
centos-source-keystone:3.0.0	"kolla_start"	keystone
centos-source-rabbitmq:3.0.0	"kolla_start"	rabbitmq
centos-source-mariadb:3.0.0	"kolla_start"	mariadb
centos-source-memcached:3.0.0	"kolla_start"	memcached
centos-source-keepalived:3.0.0	"kolla_start"	keepalived
centos-source-haproxy:3.0.0	"kolla_start"	haproxy
centos-source-ceph-osd:3.0.0	"kolla_start"	ceph_osd_2
centos-source-ceph-mon:3.0.0	"kolla_start"	ceph_mon
centos-source-cron:3.0.0	"kolla_start"	cron
centos-source-kolla-toolbox:3.0.0	"kolla_start"	kolla_toolbox
centos-source-heka:3.0.0	"kolla_start"	heka

实现

- 使用 Dockerfile + jinja2 构建镜像
- 使用镜像依赖
 - 更快的构建速度，更小的镜像
- 使用 Ansible 进行部署
- 容器化一切服务
 - libvirt / openvswitch / neutron
- 一个容器只有一个进程
- 使用 `--net host` 网络模型

Kolla 实践

```
kolla-build --base ubuntu/centos/oraclelinux --type source/  
binary --push
```

```
kolla-genpwd
```

```
vim /etc/kolla/globals.yml
```

```
kolla-deploy -i <inventory> pre-check
```

```
kolla-deploy -i <inventory> pull
```

```
kolla-deploy -i <inventory> deploy
```

```
kolla-deploy -i <inventory> post-deploy
```

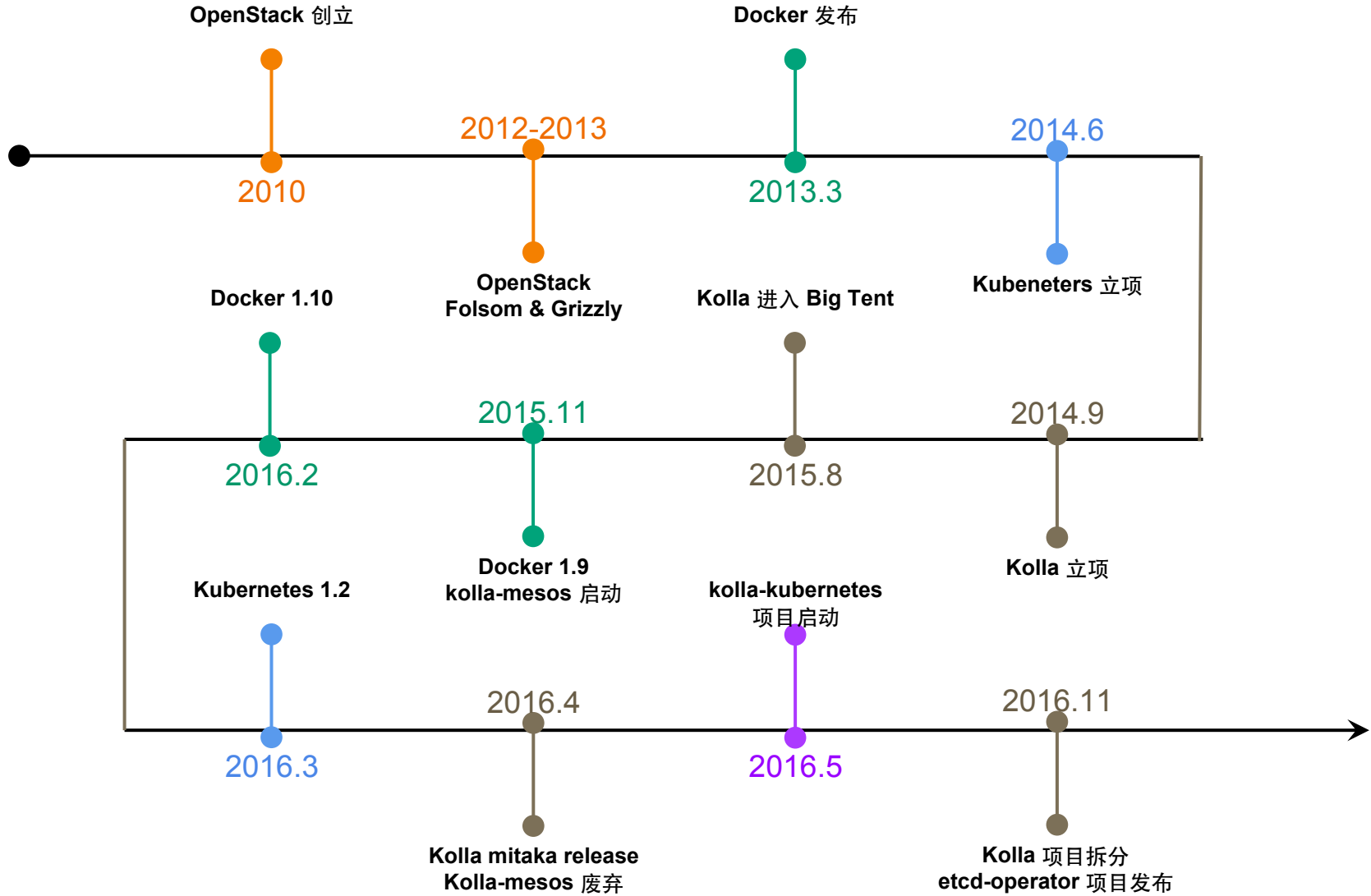
优点

- 资源隔离
- 快速部署
 - 容器的毫秒级启动
 - 单机部署在5分钟内
- 运行环境不变性
- 便携性
 - 镜像总大小 < 3GB
- 升级
 - 原子升级
 - 单服务升级
 - 失败回滚

缺点

- Docker 的成熟度
- 增加了额外的复杂度

机遇



挑战

- 日志的收集
- 配置文件管理
- container 编排
- 集成测试
- 镜像的灵活性
- 容器化后 docker 本身的问题
- 数据持久化

Contributing to Kolla

Join us on IRC : #openstack-kolla on Freenode

Mailing List: OpenStack dev list, prefix with [kolla]

Launchpad Project : <https://launchpad.net/kolla>

Features/Blueprints: <https://blueprints.launchpad.net/kolla>

Bug Tracker: <https://bugs.launchpad.net/kolla>

Github Repo : <https://github.com/openstack/kolla>

Docker Hub Images: <https://hub.docker.com/u/kolla>

容器交付OpenStack



openstack™
CLOUD SOFTWARE

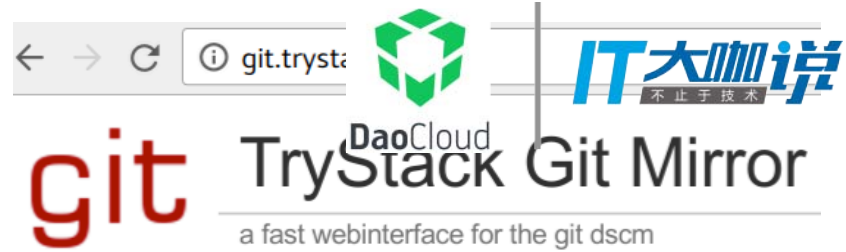
Kolla-Kubernetes
Fuel-ccp



kubernetes

by Google

广告时间



git.trystack.cn

git TryStack DaoCloud Git Mirror

a fast webinterface for the git dscm

images.trystack.cn

index about

TryStack Images Mirror

Cirros

- [cirros-0.3.4-x86_64-disk.img](#)

CentOS

- [CentOS-6-x86_64-Gener](#)
- [CentOS-7-x86_64-Gener](#)

Ubuntu

- [precise-server-cloudimg-a](#)
- [trusty-server-cloudimg-ar](#)
- [xenial-server-cloudimg-a](#)

Name

default

git/spice/spice-html5
kanaka/noVNC

openstack

openstack/aeromancer
openstack/ailuropoda
openstack/almanach

r

⊖-role-bindep

⊖-role-cloud-launcher

⊖-role-diskimage-builder

⊖-role-jenkins

⊖-role-jenkins-job-builder

⊖-role-jenkins-plugins

⊖-role-nginx

⊖-role-nodenvol

TryStack Kolla Images

How to use

Now, we only support centos+source deployment.

Add following parameter into `/etc/kolla/globals.yml` file.

```
kolla_base_distro: "centos"
kolla_install_type: "source"
openstack_release: "3.0.1"
docker_registry: "hub.trystack.cn"
docker_namespace: "kolla"
```