Containerized OpenStack

2016.03

目录

关于我

- Jeffrey Zhang (张雷)
- From 99cloud (九州云)
- Nickname/Github: Jeffrey4l
- Email: zhang.lei.fly@gmail.com
- OpenStack Kolla Core
- Blog: http://xcodest.me

九州云

跨越 OPENSTACK 企业落地最后一公里

开物成务 溯流求源 九州纵风 四海牧云

http://www.99cloud.net


九州云
Cloud
九州云99Cloud

OpenStack 简介

The OpenStack Mission: to produce the ubiquitous Open Source Cloud Computing platform that will meet the needs of public and private clouds regardless of size, by being simple to implement and massively scalable.

目标:成为一个易于实现的，可扩展的，可同时满足公有云和私有云需求的开源云平台

OpenStack 中和 Docker 相关的项目

- Magnum
  - Containers Service for OpenStack
- Kolla
  - Deploying OpenStack using Docker
- nova-docker
  - Docker driver for OpenStack Nova
- kuryr
  - Docker remote driver for OpenStack Neutron

Kolla 背景

- OpenStack 部署困难
- 运维 OpenStack 也很困难
- 当前的部署工具也很复杂
  - Fuel (mirantis)
  - openstack-ansible (hp)
  - puppet

Kolla 是什么

- “Kolla” is Greek for glue
- An OpenStack project hosted on
  - https://github.com/openstack/kolla
- provides
  - **production-ready** docker images
  - **deployment tools** for operating OpenStack clouds.

Kolla 目标

- 简化部署和运维
  - 通过 ansible 来做配置管理
- 使用容器
  - 提供直接可用的组件(Image 等)
  - 快速部署

Kolla 当前项目情况

#	Module	Commits
1	project-config	1758
2	nova	1302
3	openstack-manuals	1069
4	neutron	1064
5	cinder	789
6	kolla	769
7	fuel-library	748
8	fuel-web	734
9	horizon	706
10	heat	706

Showing 1 to 10 of 502 entries

First Previous 1 2 3 4 5 Next Last

#	Company	Commits
	Servosity	179
	Mirantis	119
	Red Hat	91
	Cisco Systems	79
	99cloud	77
	Oracle	45
	EasyStack	43
	Intel	26
	IBM	20
)	NEC	18

wing 1 to 10 of 22 entries

First Previous 1 2 3 Next Last

为什么选用 Docker

优点

- Immutable
- Portable
- Fast
- Massive community
- Branding
- Growth

缺点

- Green
  - Kolla is even greener
- Additional complexity
- Difficult to audit

Kolla 架构

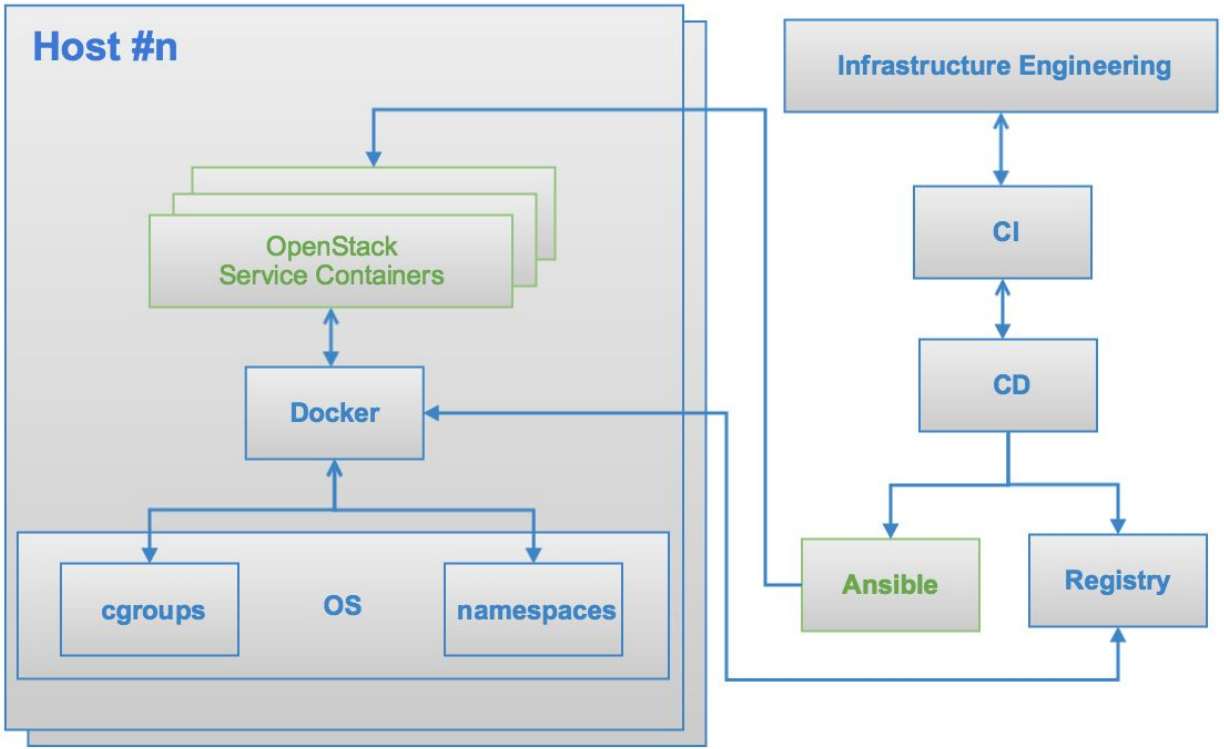
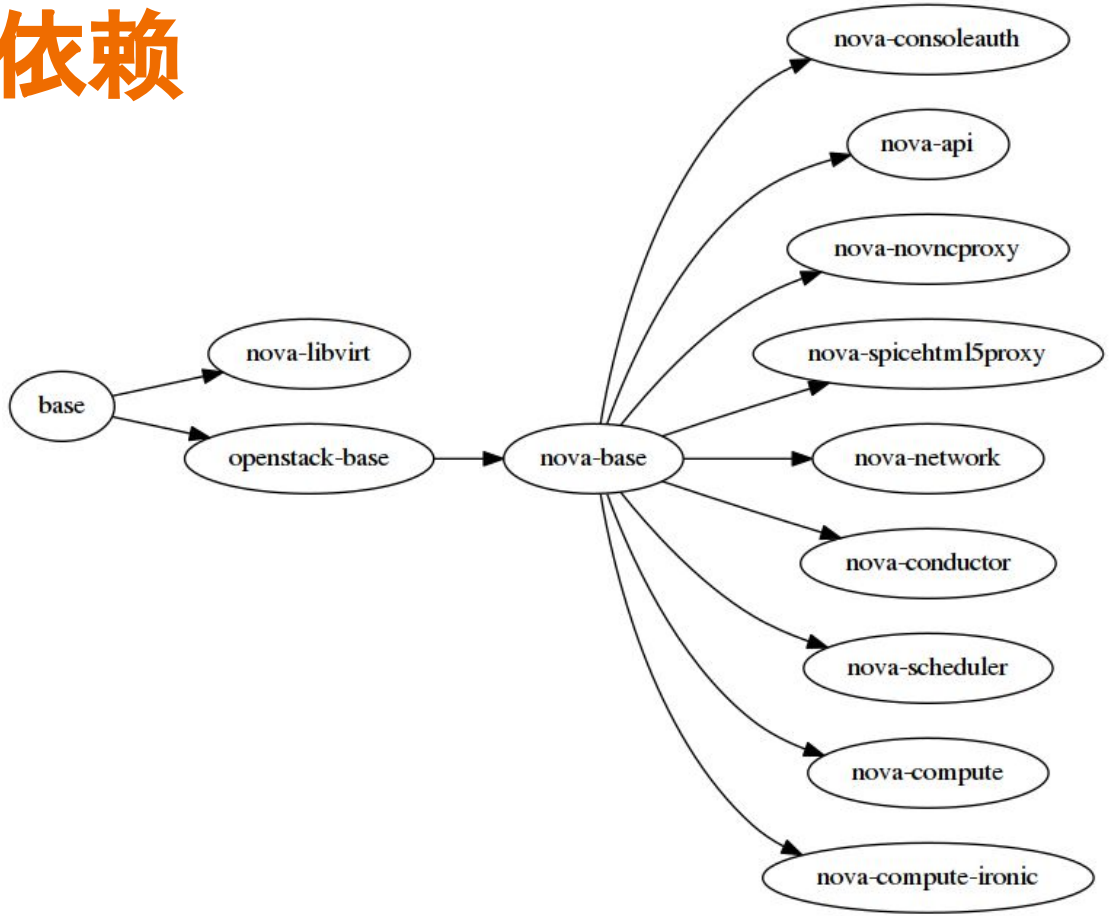

Host #n
OpenStack
Service Containers
Docker
cgroups
OS
namespaces
Infrastructure Engineering
CI
CD
Ansible
Registry

Image Build

- 使用镜像依赖，合理利用 Docker 的 cache 机制
- 使用 Jinja2 模板语言，增强 Dockerfile 语法
- 支持多种系统发行版本
  - CentOS/RHEL/Ubuntu
- 支持多种安装类型
  - 二进制安装包/源码安装

镜像依赖

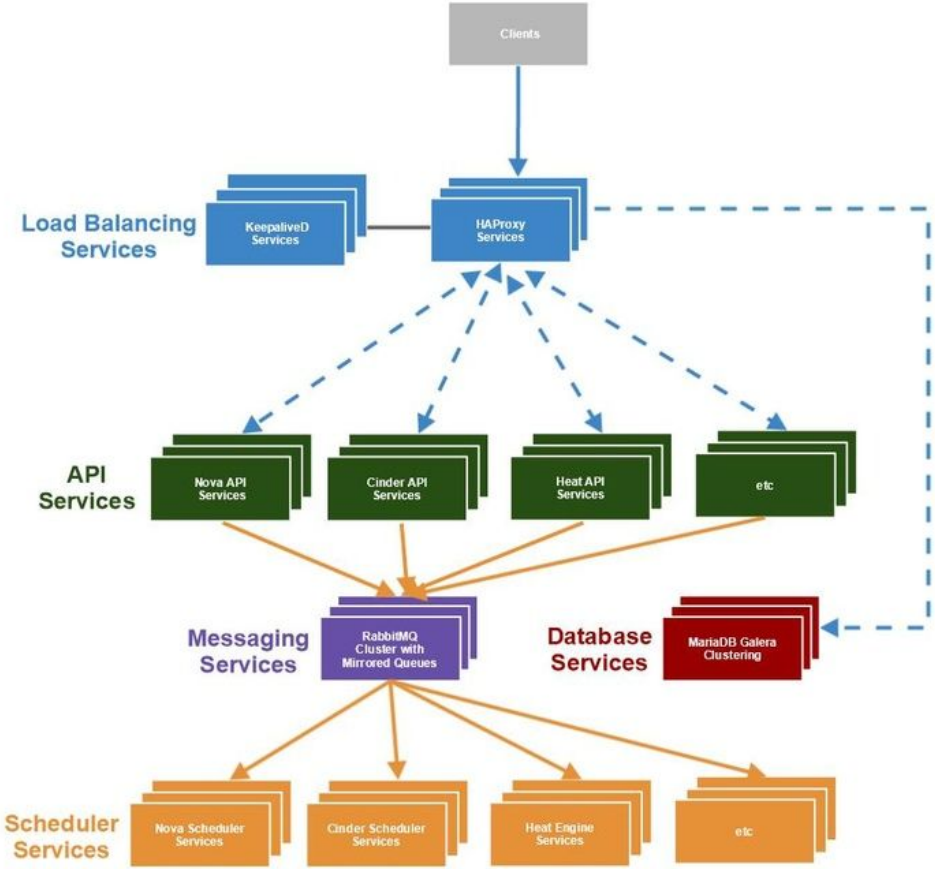

base
nova-libvirt
openstack-base
nova-base
nova-consoleauth
nova-api
nova-novncproxy
nova-spicehtml5proxy
nova-network
nova-conductor
nova-scheduler
nova-compute
nova-compute-ironic

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 %}
```

Kolla 部署

- 使用 Ansible 作为编排工具
  - 提供 `kolla-ansible` 脚本
  - 通过 `ansible` 的 `inventory` 来控制每台机器安装的服 务
  - 支持单机，多机部署
  - 支持 `rabbitmq/mariadb` 的集群部署
  - 支持整个 OpenStack 的 HA
- 直接使用 host 网络 `--net host`
- 非 root 用户运行，保存安全
- 有些 container 需要使用 `` `privileged` `` 权限，如 libvirt
- 将来会支持 mesos, k8s 等


Clients
Load Balancing Services
KeepaliveD Services
HAProxy Services
API Services
Nova API Services
Cinder API Services
Heat API Services
etc
Messaging Services
RabbitMQ Cluster with Mirrored Queues
Database Services
MariaDB Galera Clustering
Scheduler Services
Nova Scheduler Services
Cinder Scheduler Services
Heat Engine Services
etc

Container Bootstrap

在第一次启动容器前，往往需要进行某些初始化的工作

通过启动一个 bootstrap container 进行初始化，并在完成后销毁。

例如：通过 keystone_bootrap 初始化好 keystone 的 admin 账号和 identity 服务的 endpoint.

Demo

遇到的问题

1. 配置文件
2. Data Container VS Named Volume
3. 日志收集
4. restart_policy: always 的问题

问题:配置文件

Docker 推崇使用环境变量来进行配置，但是

- OpenStack 的配置项太多，通过环境变量控制根本不可能
- 难以修改
- 难以审计

解决:配置文件

- 配置文件统一存放在宿主机
- 通过 --volume 的方法映射到 Container 中 (/var/lib/kolla/config_files/)
- 用一个 json 文件，控制配置文件在 container 中的位置和权限，每次 container 启动时，先运行 `set_config.py` 脚本，把配置文件复制到目标位置，再启动真正的应用服务程序

nova-api.json

```
{
    "command": "nova-api",
    "config_files": [
        {
            "source": "/var/lib/kolla/config_files/nova.conf",
            "dest": "/etc/nova/nova.conf",
            "owner": "nova",
            "perm": "0600"
        }
    ]
}
```

问题：Data Container VS Named Volumes

Data Container 可能丢失数据

- image 如果发生改变的话，container 重启时数据会被删除
- Named Volume 没有这样的问题。
  - Named Volume 还支持多种插件**

** https://docs.docker.com/engine/extend/plugins/

问题:日志收集

- 同一服务打印出来的日志有多个，不能简单的打印到 stdout
- docker logs 出来的日志会有延时
- 有的服务不能打印到 stdout

解决:日志收集

- 使用 heka 来接收日志
  - heka 性能比 logstash 要好的多
- 所用服务的日志保存到同一个 volume 里面
- 对于只会写入 rsyslog 的服务，由 heka 来生成容器内的 /dev/log，对日志进行收集
  - 如 swift

问题：restart_policy: always

使用 restart_policy: always 后，Container 之前没有了编排功能，容易造成被依赖的容器启动没有完成，后面的容器出错

例如启用 rsyslog 容器时，container 虽然已经启动了，但是 /dev/log 没有初始化成功，导致后面的容器因为没有变法拿到 /dev/log 而启动出错

https://bugs.launchpad.net/kolla/+bug/1544545

问题：restart_policy: always

```
rm -rf /run/kolla/log
docker run -it --rm -v /run/kolla/log:/dev/log centos ls -alh /dev/log
OK, but /dev/log will be a folder

rm -rf /run/kolla/log
docker run -it --rm -v /dev:/dev centos ls -alh /dev/log
OK, the /dev/log will be the same with the host /dev/log ( just mount bind)

rm -rf /run/kolla/log
docker run -it --rm -v /dev:/dev -v /run/kolla/log:/dev/log centos ls -alh /dev/log
Error, this is the issue case.

rm -rf /run/kolla/log
touch /run/kolla/log
docker run -it --rm -v /dev:/dev -v /run/kolla/log:/dev/log centos ls -alh /dev/log
OK. the /dev/log is the same with the host /run/kolla/log( a normal file, just mount bind)
```

解决：restart_policy: always

- 不使用 restart_policy: always
- 使用 ansible 或其它编排工具来保证容器的正常启动

或

- 尽量避免使用上述的使用情况

Contributing to Kolla

- Join us on IRC : #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/kollaglue/

рахмат
danke
謝謝
ngiyabonga
teşekkür ederim
Баярлалаа
спасибо
thank you
gracias
dank je
tapadh leat
mochchakkeram
bedankt
dziękuję
go raibh maith agat
obrigado
sukriya
kop khun krap
arigatô
dakujem
мерси
grazie
terima kasih
merci
ευχαριστώ
감사합니다
ありがとう
hvala
mauruuru
sagolun
takk
taiku
mersi
kia ora
barka
welalin
tack
spas
faafetai lava
vinaka
blagodaram
misaotra
matondo
paldies
grazzi
mahalo
kiitos
dankie
dhanyavad
хвала
asante
manana
obrigada
tenki
murakoze
enkosi
djiere dieuf
tau
дякую
mamnun
chnorakaloutioun
gratias ago
gràcies
sulpáy
sobodi
dėkuji
mèsi
najis tuke
tanemirt
rahmet
trugarez
shukriya
mercê
dhanyavadagalu
diolch
xièxie
rahmat
kam sah hamnida
akun dankon
köszönöm