
Improve the OpenStack Deployment and Upgrade by Using Docker

— 2016.07 —

About me

- 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>

Aganda

- What is Kolla
- Feature
- Implementation
- Advantance
- Disadvantage
- Future

OpenStack and Docker

- Magnum
 - Containers Service for OpenStack
- Kolla
 - Deploying OpenStack using Docker
- nova-docker
 - Docker driver for OpenStack Nova
- kuryr
 - Docker remote driver for OpenStack Neutron
- Solum
 - make cloud services easier to consume and integrate into your application development process.



What's Kolla

- Come from TripleO
- Launch at Sep 17, 2014
- Join OpenStack Big Tent at Aug 5, 2015
- First release at Liberty
- Second release at Mitaka with more feature

What's Kolla: Goals

- Simplify the Deployment and Operation
- Provides
 - Production-ready docker images
 - Deployment tools for operating OpenStack clouds.
- Deploy the big tent at 100 nodes scale

Feature

- All active high availability
- Ceph backend storage
- Support multi Linux distro
 - CentOS / OracleLinux / Ubuntu
- Build from packages or build from source
- Small runtime dependency footprint, only need docker-py and docker-engine
- Docker container for atomic upgrades

Implementation

- Use Dockerfile + Jinja2 to build image
- Use Image dependency
 - build faster and smaller size
- Ansible-playbooks as deployment tool
- Containerized everything
 - libvirt / openvswitch / neutron
- Each container has only one process
- Use host network
- Better configuration management

Implementation

- Use Dockerfile + Jinja2 to build image
- Use Image dependency
 - build faster and smaller size
- Ansible-playbooks as deployment tool
- Containerized everything
 - libvirt / openvswitch / neutron
- Each container has only one process
- Use host network
- Better configuration management

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'] %}

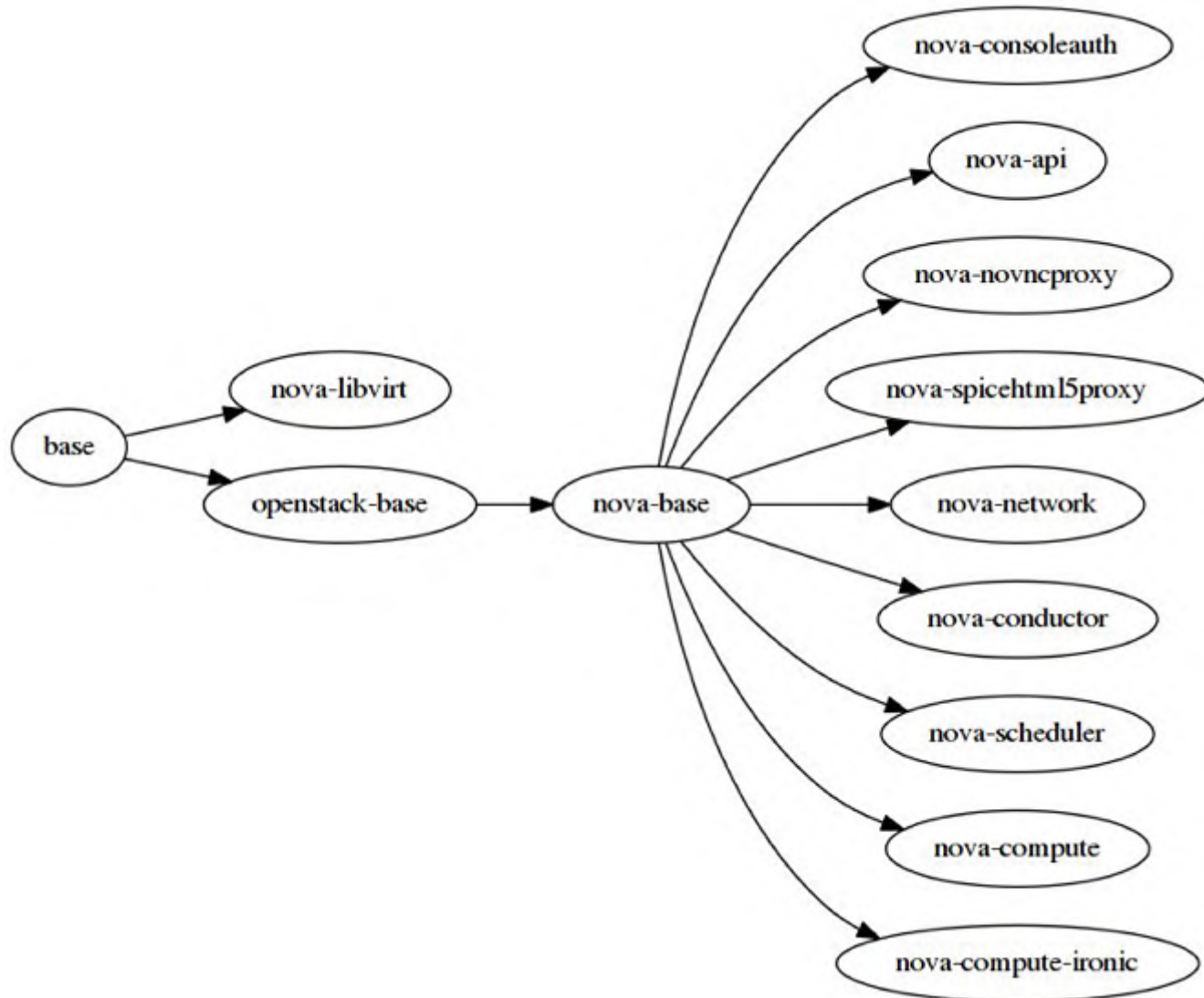
RUN apt-get install -y --no-install-recommends \
    nova-api \
    python-memcache \
    && apt-get clean

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

Implementation

- Use Dockerfile + Jinja2 to build image
- Use Image dependency
 - build faster and smaller size
- Ansible-playbooks as deployment tool
- Containerized everything
 - libvirt / openvswitch / neutron
- Each container has only one process
- Use host network
- Better configuration management

Image Dependency



Implementation

- Use Dockerfile + Jinja2 to build image
- Use Image dependency
 - build faster and smaller size
- Ansible-playbooks as deployment tool
- Containerized everything
 - libvirt / openvswitch / neutron
- Each container has only one process
- Use host network
- Better configuration management

Kolla Deploy

- use Ansible for orchestration
 - Provide `kolla-ansible` script
 - Use ansible inventory to control the running services on host
 - Support AIO and multi-node env
 - Support all active cluster include `rabbitmq` and `mariadb`
- use host network `--net host`
- non-root user for security
- need `--privileged` in some container, like libvirt
- Support k8s in future

kolla-ansible

prechecks	Do pre-deployment checks for hosts
mariadb_recovery	Recover a completely stopped mariadb cluster
deploy	Deploy and start all kolla containers
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
upgrade	Upgrades existing OpenStack Environment

Implementation

- Use Dockerfile + Jinja2 to build image
- Use Image dependency
 - faster and smaller
- Ansible-playbooks as deployment tool
- Containerized everything
 - libvirt / openvswitch / neutron
- Each container has only one process
- Use host network
- Better configuration management

Containerized Everything

IMAGE	COMMAND	NAMES
centos-source-horizon:2.0.1	"kolla_start"	horizon
centos-source-cinder-backup:2.0.1	"kolla_start"	cinder_backup
centos-source-cinder-volume:2.0.1	"kolla_start"	cinder_volume
centos-source-cinder-scheduler:2.0.1	"kolla_start"	cinder_scheduler
centos-source-cinder-api:2.0.1	"kolla_start"	cinder_api
centos-source-neutron-metadata-agent:2.0.1	"kolla_start"	neutron_metadata_agent
centos-source-neutron-l3-agent:2.0.1	"kolla_start"	neutron_l3_agent
centos-source-neutron-dhcp-agent:2.0.1	"kolla_start"	neutron_dhcp_agent
centos-source-neutron-openvswitch-agent:2.0.1	"kolla_start"	neutron_openvswitch_agent
centos-source-neutron-server:2.0.1	"kolla_start"	neutron_server
centos-source-openvswitch-vswitchd:2.0.1	"kolla_start"	openvswitch_vswitchd
centos-source-openvswitch-db-server:2.0.1	"kolla_start"	openvswitch_db
centos-source-nova-conductor:2.0.1	"kolla_start"	nova_conductor
centos-source-nova-scheduler:2.0.1	"kolla_start"	nova_scheduler
centos-source-nova-novncproxy:2.0.1	"kolla_start"	nova_novncproxy
centos-source-nova-consoleauth:2.0.1	"kolla_start"	nova_consoleauth
centos-source-nova-api:2.0.1	"kolla_start"	nova_api
centos-source-glance-api:2.0.1	"kolla_start"	glance_api
centos-source-glance-registry:2.0.1	"kolla_start"	glance_registry
centos-source-keystone:2.0.1	"kolla_start"	keystone
centos-source-rabbitmq:2.0.1	"kolla_start"	rabbitmq
centos-source-mariadb:2.0.1	"kolla_start"	mariadb
centos-source-memcached:2.0.1	"kolla_start"	memcached
centos-source-kibana:2.0.1	"kolla_start"	kibana
centos-source-keepalived:2.0.1	"kolla_start"	keepalived
centos-source-haproxy:2.0.1	"kolla_start"	haproxy
centos-source-elasticsearch:2.0.1	"kolla_start"	elasticsearch
centos-source-ceph-osd:2.0.1	"kolla_start"	ceph_osd_8
centos-source-ceph-osd:2.0.1	"kolla_start"	ceph_osd_2
centos-source-ceph-mon:2.0.1	"kolla_start"	ceph_mon
centos-source-cron:2.0.1	"kolla_start"	cron
centos-source-kolla-toolbox:2.0.1	"/bin/sleep infinity"	kolla_toolbox
centos-source-heka:2.0.1	"kolla_start"	heka

Implementation

- Use Dockerfile + Jinja2 to build image
- Use Image dependency
 - faster and smaller
- Ansible-playbooks as deployment tool
- Containerized everything
 - libvirt / openvswitch / neutron
- Each container has only one process
- Use host network
- Better configuration management

Implementation

- Use Dockerfile + Jinja2 to build image
- Use Image dependency
 - faster and smaller
- Ansible-playbooks as deployment tool
- Containerized everything
 - libvirt / openvswitch / neutron
- Each container has only one process
- Use host network
- Better configuration management

Advantage

- Resource Isolation
- Rapid deployment
 - Containers can boot extremely fast (in milliseconds!)
 - Deploy a AIO environment in 5 minute
- Immutable
- Portable
 - Total images size < 3GB
 - No more need big repository
- Upgrade
 - Atomic Upgrade
 - Single Service Upgrade
 - Rollback

Disadvantage

- Docker is green
 - Kolla is even greener
- Additional complexity

Demo

Future

- ~~kolla-mesos~~
- kolla-kubernetes
 - <https://github.com/openstack/kolla-kubernetes>
- kolla-host
 - manage bare metal mechina
- plugin
 - support more and more service
- better configuration merge

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>

Thanks