

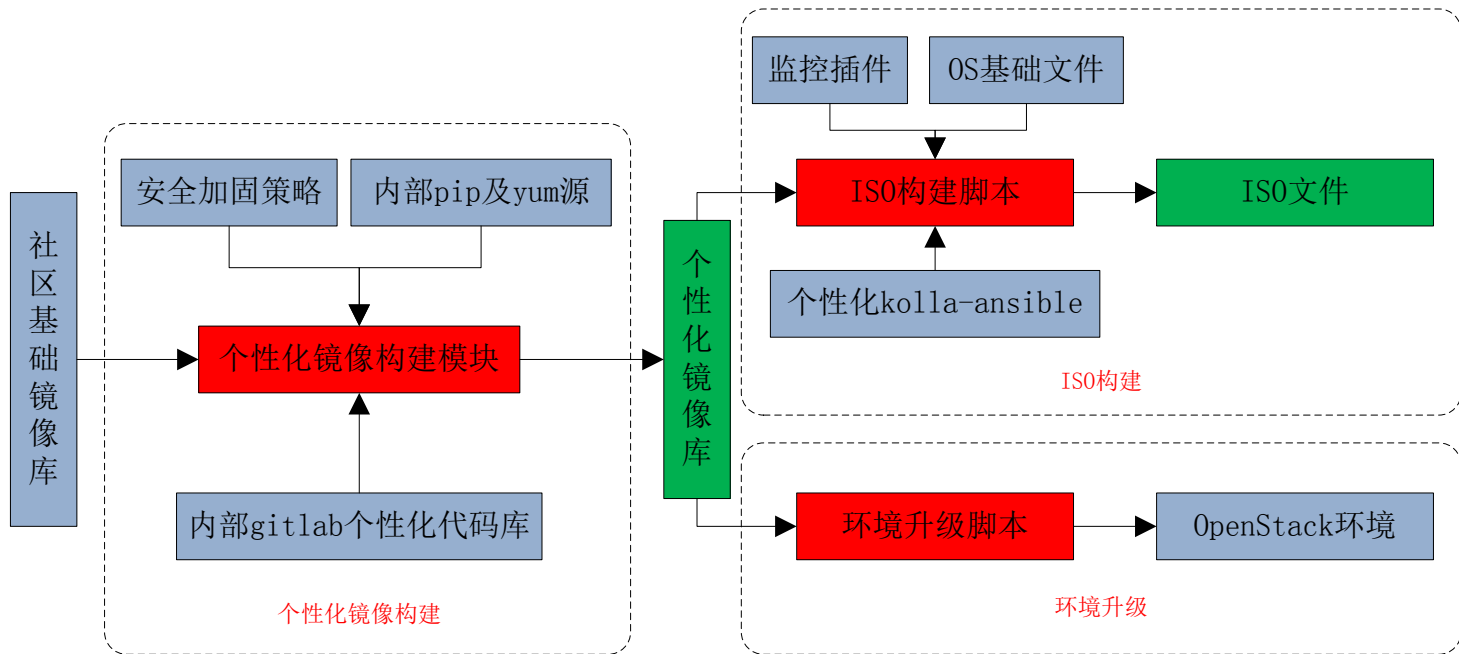


目 录 CONTENTS

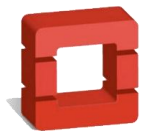
- 1 个性化镜像构建及环境升级
- 2 OpenStack组件容器化部署
- 3 Kolla-kubernetes
- 4 Openshift+kolla
- 5 Pipeline流程



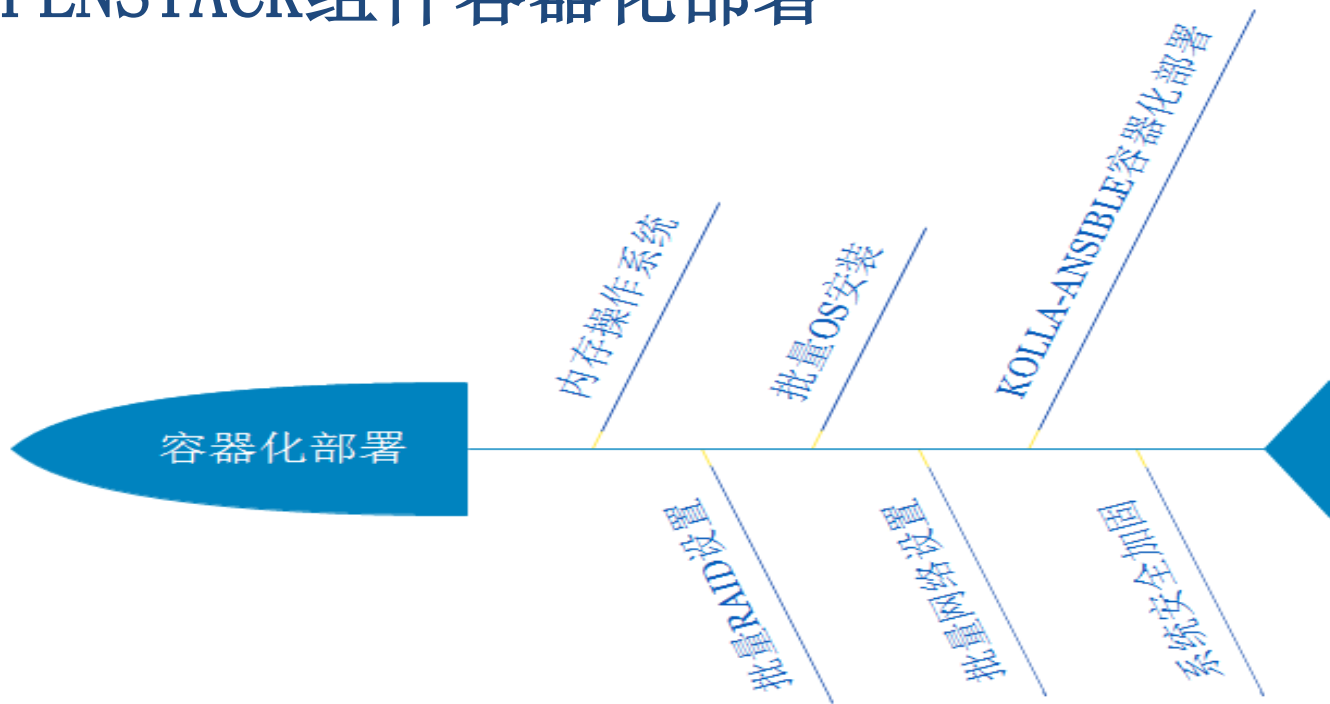
个性化镜像构建及环境升级



每日构建，每日更新，提高产品研究的敏捷程度。



OPENSTACK组件容器化部署





OPENSTACK组件容器化部署

- 内存操作系统

经过优化的icos-cobbler支持的启动方式:

- ✓ 本地硬盘启动
- ✓ 内存操作系统启动
- ✓ 本地硬盘安装并启动



此外，我们构建的iso，不仅支持BIOS类型的主板，还支持UEFI类型的主板。



OPENSTACK组件容器化部署

- 批量raid设置

得益于icos-cobbler对内存操作系统的支持，我们可以在内存操作系统中对raid进行批量设置。

```
1 #!/bin/bash
2 # clear original configurations.
3 /opt/MegaCli164 -cfgclr -a0
4 # get device id
5 device= ./MegaCli164 -PDList -a0 | grep 'Enclosure Device ID' | awk '{print $4}' | sort -u
6
7 #create raid1, build by disk 0-1
8 #/opt/MegaCli164 CfgLdAdd -r1[${device}:0,${device}:1] WB Direct -a0
9
10 # create raid10, build by disk 0-3
11 /opt/MegaRAID/MegaCli/Megacli164 -CfgSpanAdd -r10 -Array0[${device}:0,${device}:1] -Array1[${device}:2,${device}:3] WB Direct -a0
12
13 # create raid0
14 /opt/MegaCli164 CfgLdAdd -r0[${device}:4] WB Direct -a0
15 /opt/MegaCli164 CfgLdAdd -r0[${device}:5] WB Direct -a0
16 /opt/MegaCli164 CfgLdAdd -r0[${device}:6] WB Direct -a0
17 /opt/MegaCli164 CfgLdAdd -r0[${device}:8] WB Direct -a0
18 /opt/MegaCli164 CfgLdAdd -r0[${device}:9] WB Direct -a0
19 /opt/MegaCli164 CfgLdAdd -r0[${device}:10] WB Direct -a0
20 /opt/MegaCli164 CfgLdAdd -r0[${device}:11] WB Direct -a0
21 /opt/MegaCli164 CfgLdAdd -r0[${device}:12] WB Direct -a0
22 /opt/MegaCli164 CfgLdAdd -r0[${device}:13] WB Direct -a0
23 /opt/MegaCli164 CfgLdAdd -r0[${device}:14] WB Direct -a0
24 /opt/MegaCli164 CfgLdAdd -r0[${device}:15] WB Direct -a0
25 /opt/MegaCli164 CfgLdAdd -r0[${device}:16] WB Direct -a0
26 /opt/MegaCli164 CfgLdAdd -r0[${device}:17] WB Direct -a0
27 /opt/MegaCli164 CfgLdAdd -r0[${device}:18] WB Direct -a0
```



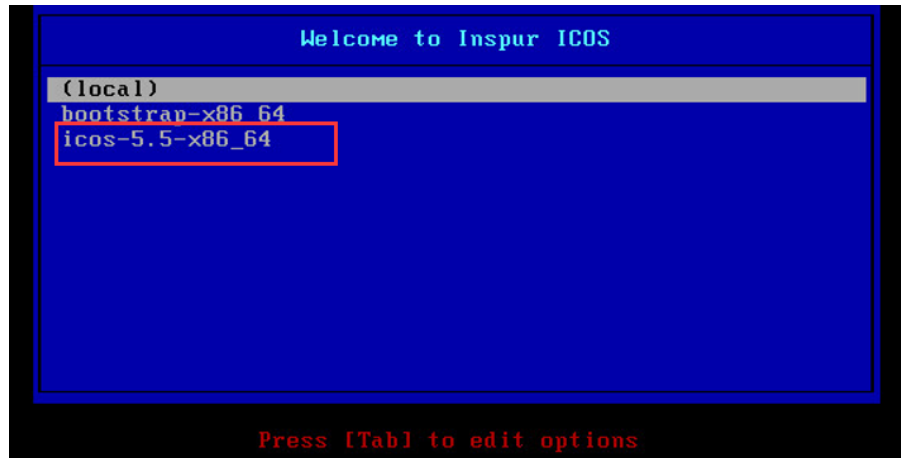
OPENSTACK组件容器化部署

- 批量操作系统安装

操作系统安装完成后，各系统之间自动互信。

```
1 server_ip
2 SERVER_IP=10.106.1.220
3
4 #cobbler default user password of kickstart(ignore)
5 ROOT_PASSWORD=password
6
7 #dhcp ip range
8 DHCP_RANGE=10.106.1.220 10.106.1.223
9
10 #dhcp ip subnet
11 DHCP_SUBNET=10.106.1.0
12
13 #dhcp gateway
14 DHCP_ROUTER=10.106.1.1
15
16 #dhcp dns
17 DHCP_DNS=8.8.8.8
18
19 #http server port
20 HTTP_PORT=82
21
22 #default menu selection of cobbler 1 local 2 bootstrap 3 icos-5.2-x86_64(danger!!!)
23 PXE_DEFAULT_MENU=3
```

在这里设置系统的启动方式





OPENSTACK组件容器化部署

• 批量网络设置

```
[root@localhost set_net_ssh]# ls
add_hosts  config_bond.sh  init_config.sh  main.sh  network.info  node.yml  PRE_DEPLOY  README  sn_ip.file
[root@localhost set_net_ssh]#
```

开发了一些工具对网络等进行设置:

- ✓ 以机器序列号为识别标识
- ✓ sn_ip.file保存序列号、主机名及管理网ip间的映射关系
- ✓ network.info保存各网卡、bond及ip信息
- ✓ init_config.sh根据sn_ip.file对主机名进行设置
- ✓ config_bond.sh根据network.info对网络进行设置

```
1 232439995129e69193838eca043dacd con01 200.10.1.21
2 4232fb3db50fc47957d1314cee6a6d5c con02 200.10.1.22
3 42329e637a9cc6981ce6d61428beddb7 con03 200.10.1.23
4 42323e28b46da603649ec855e9349400 com01 200.10.1.31
```

```
4 #device_name device_name bond_name ipaddr-prefix gateway
5 # | | | |
6 #eno1 eno2 bond0 100.100.1.1 100.100.1.254
7 ens33 ens9999 admin 100.2.30 100.2.30.1
8 ens34 ens9999 private 192.168.2
9 ens35 ens9999 public
10 ens36 ens9999 storage 192.168.3
```




OPENSTACK组件容器化部署

- Kolla-ansible容器化部署

我们目前的部署流程与社区的部署流程大体相同，不同的是，我们在部署的流程中增加了我们自研模块的部署过程，并进行了一些参数的固化。

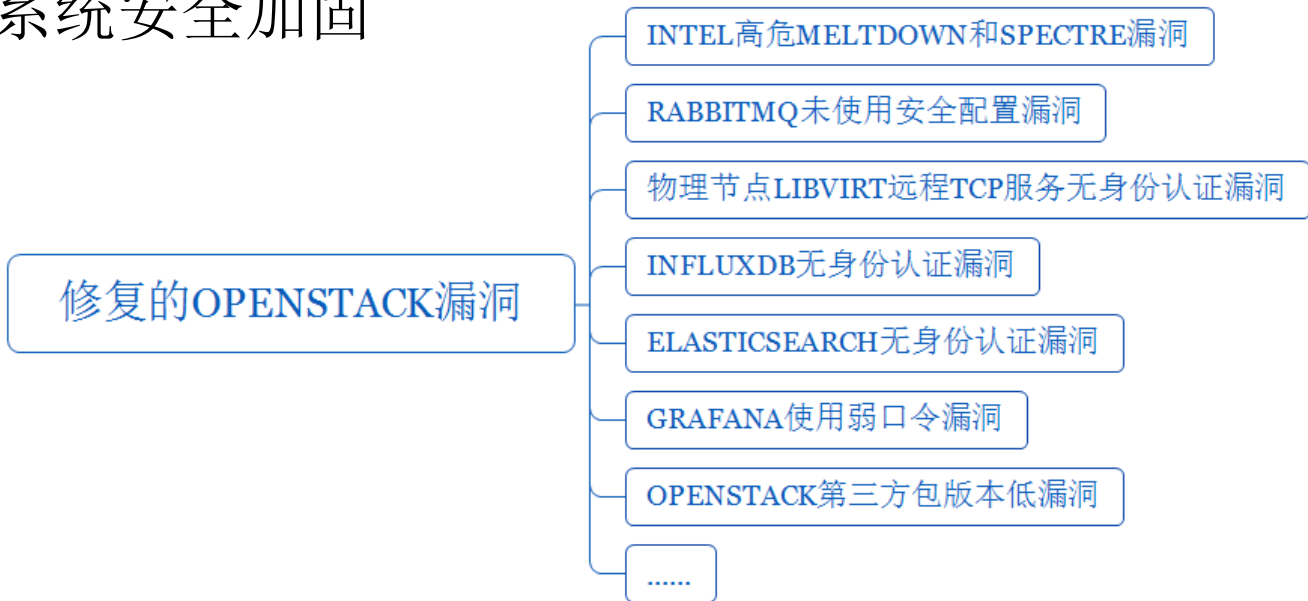
```
[root@localhost ~]# kolla-ansible deploy -i /root/multinode
```

```
583 - name: Apply role leo
584   hosts: leo
585   serial: '{{ serial|default("0") }}'
586   roles:
587     - { role: leo,
588         tags: leo,
589         when: enable_leo | bool }
590
591 - name: Apply role pisces
592   hosts: pisces
593   serial: '{{ serial|default("0") }}'
594   roles:
595     - { role: pisces,
596         tags: pisces,
597         when: enable_pisces | bool }
598
599 - name: Apply role kapacitor
600   hosts: kapacitor
601   serial: '{{ serial|default("0") }}'
602   roles:
603     - { role: kapacitor,
604         tags: kapacitor,
605         when: enable_kapacitor | bool }
606
607 - name: Apply role pluto
608   hosts: pluto
609   serial: '{{ serial|default("0") }}'
610   roles:
611     - { role: pluto,
612         tags: pluto,
613         when: enable_pluto | bool }
```



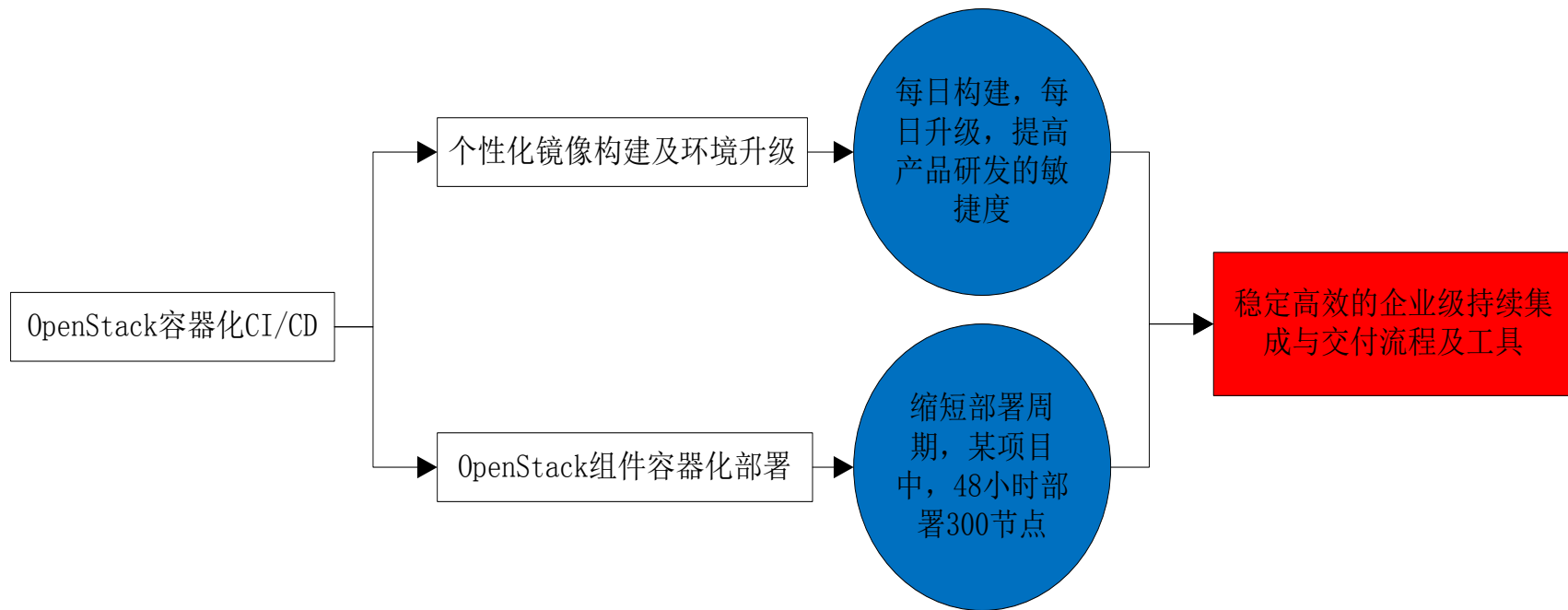
OPENSTACK组件容器化部署

- 系统安全加固



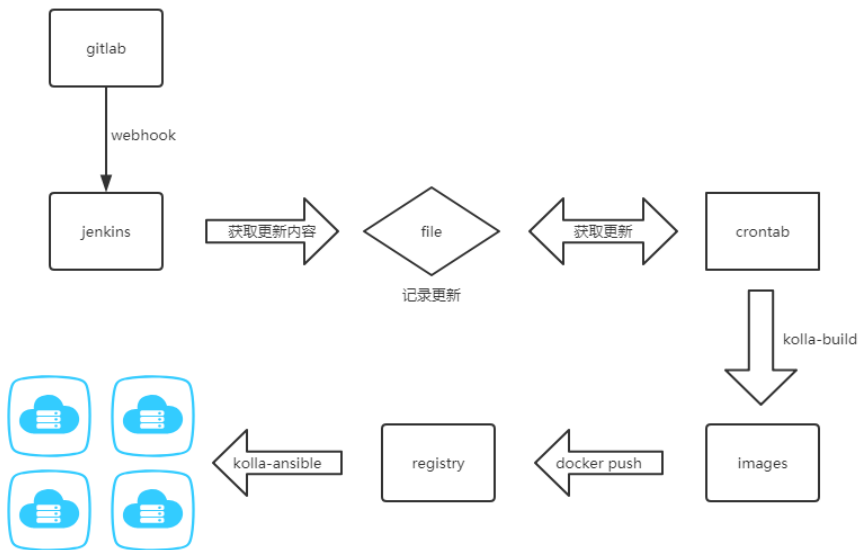


OPENSTACK组件容器化部署





传统方案



问题:

- 组织松散、跨平台
- 没有统一的规范
- 构建部署触发依赖定时任务
- 升级困难
- 缺少监控和运维



KOLLA-KUBERNETES

Kubernetes:

容器云管理平台，提供多种自动化部署策略，支持多种复杂条件的节点调度，满足生产环境中复杂的应用场景，支持弹性伸缩功能，可根据业务需求调整实例数量，同时可以借助第三方系统实现应用的持续集成。

Kolla-kubernetes:

将容器化openstack与kubernetes结合，OpenStack看作是运行在kubernetes集群上一个应用程序，和其他应用一样使用k8s提供的服务；k8s的集群特性可以为OpenStack提供运维支持；同时OpenStack的Ironic 还可以为k8s集群提供裸机管理服务。



OPENSIFT+KOLLA

- 灵活、高度可定制化
- Pipeline自动化程度高
 - 使用定制的 Jenkins 镜像，其中包含 Jenkins 插件，目的是将 Jenkins 中的构建和编排操作可以被 openshift 纳管
 - 私有的STI构建流程, 使用S2I builder镜像构建，保证构建一致性
- 内置镜像仓库、监控
 - 支持HTTPS类型的私有镜像仓库，可以自定义安全证书，如同访问 DockerHub一样Pull、push操作



部署过程

- 修改默认RBAC设置
- 安装部署Helm
- 下载安装kolla-ansible kolla-kubernetes, 自定义配置文件
- 生成secret和config map并注册到openshift/k8s
- 编译Helm使用的微服务、元数据、openstack服务(tgz)
- 使用Helm部署openstack
- 安装部署openstack客户端

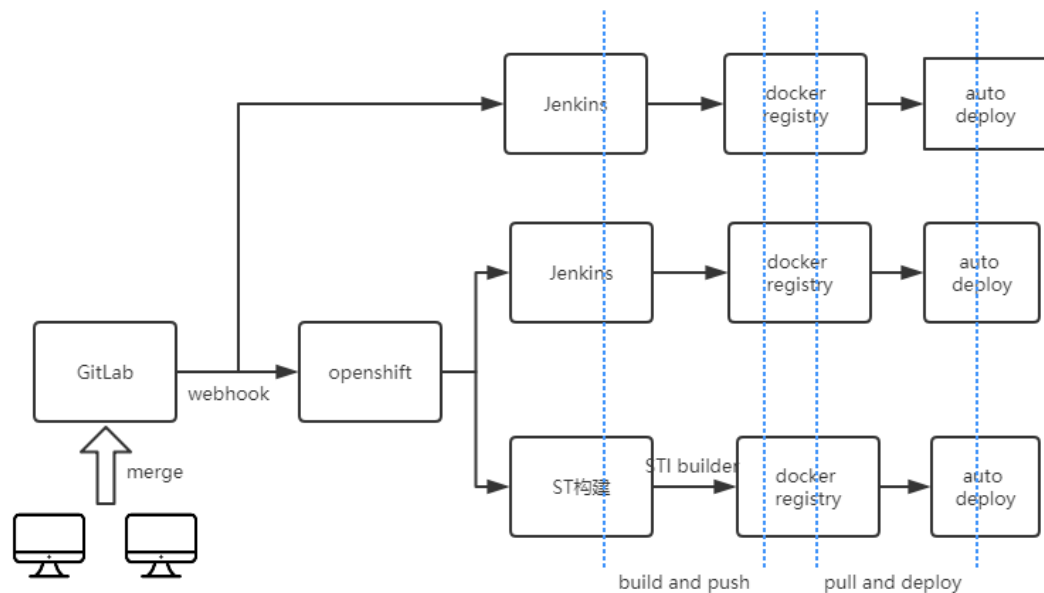


部署注意事项

- 在k8s中设置标签，指定controller和compute
- Kubernetes \geq 1.6.4 docker \geq 1.12.6 helm \geq 2.4.1
- 权限问题，kolla很多容器需要在root权限下执行,kubernetes默认是非root用户启动容器。
- 一些配置文件配置错误，导致容器不能启动成功，需要修改对应配置项。
- Neutron server 启动失败，缺少配置文件。
- Neutron网络类型目前只支持vswitch



流程图



优点:

- 所有操作/资源统一由openshift管理
 - 配置文件 - configmap
 - 用户密码 - secret
- PaaS平台提供运维支持
- 自带监控系统可监控openstack服务
- Helm版本管理，升级和回滚方便

缺点:

- 需要成熟的PaaS平台支持，增加了额外的运维工作
- Helm文件需要开发、维护
- STI构建镜像需要额外开发



CHINA
OpenInfra Days

CHINA
OpenInfra Days

IT大咖说
知识共享平台

Thank You

