

基于Kolla在48小时内完成300节点OpenStack多存储、多区域容器化部署的实践

浪潮电子信息产业股份有限公司 陈英南





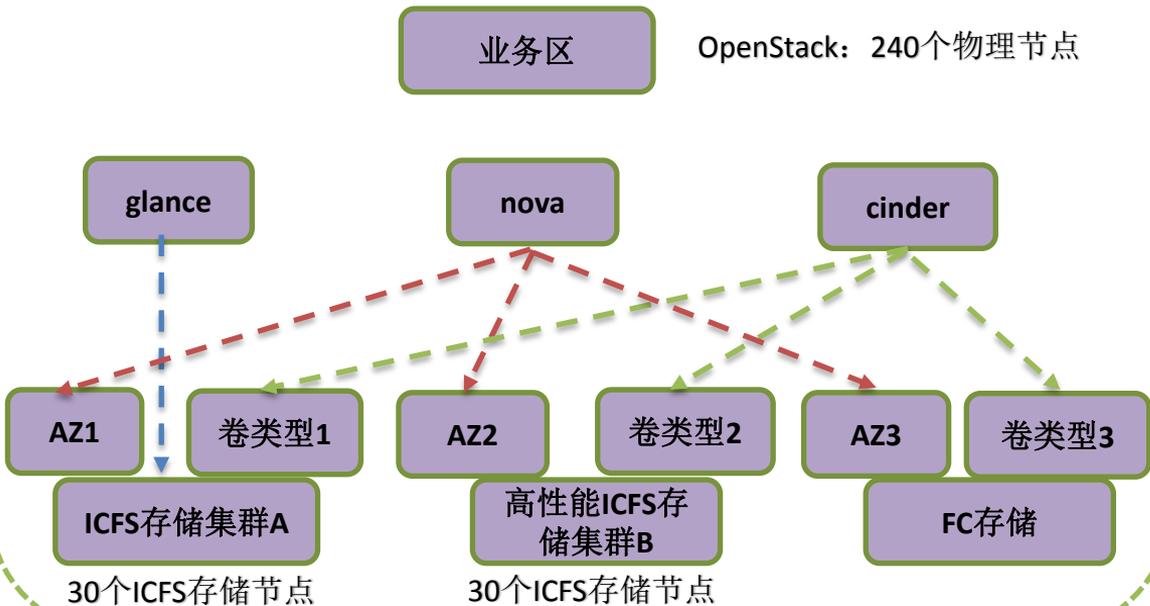
目录 CONTENTS

- 1 场景介绍
- 2 整体规划
- 3 容器化部署
- 4 容器化升级

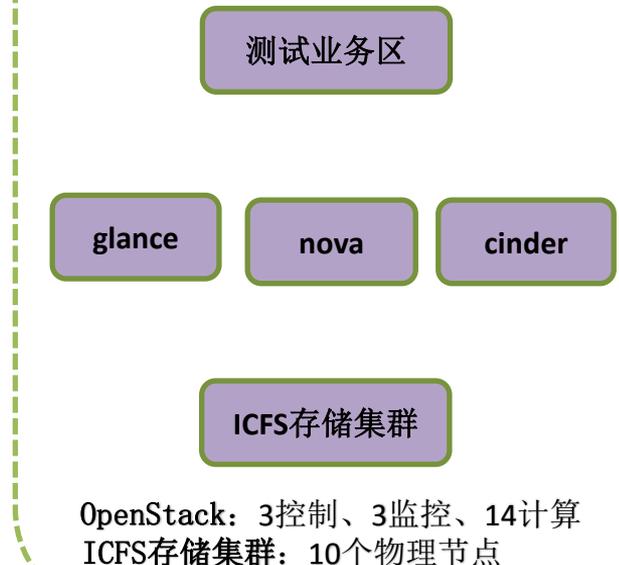


场景介绍

业务区 RegionOne

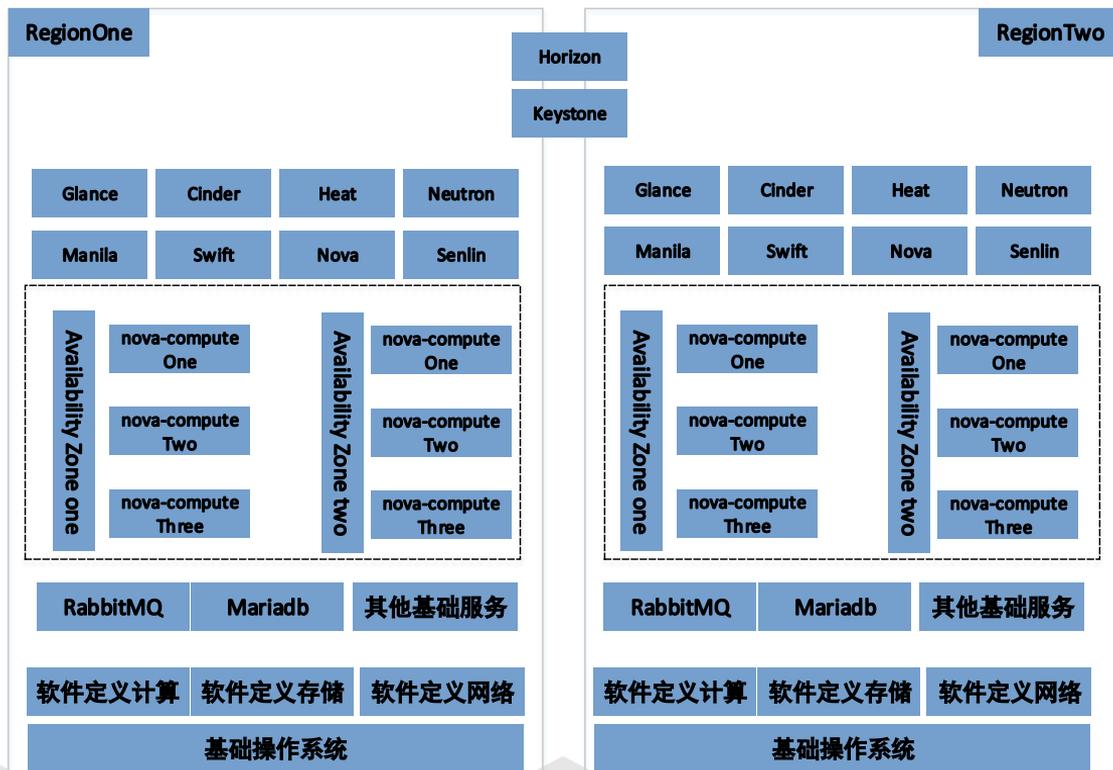


测试区 RegionTwo





整体规划-部署方案



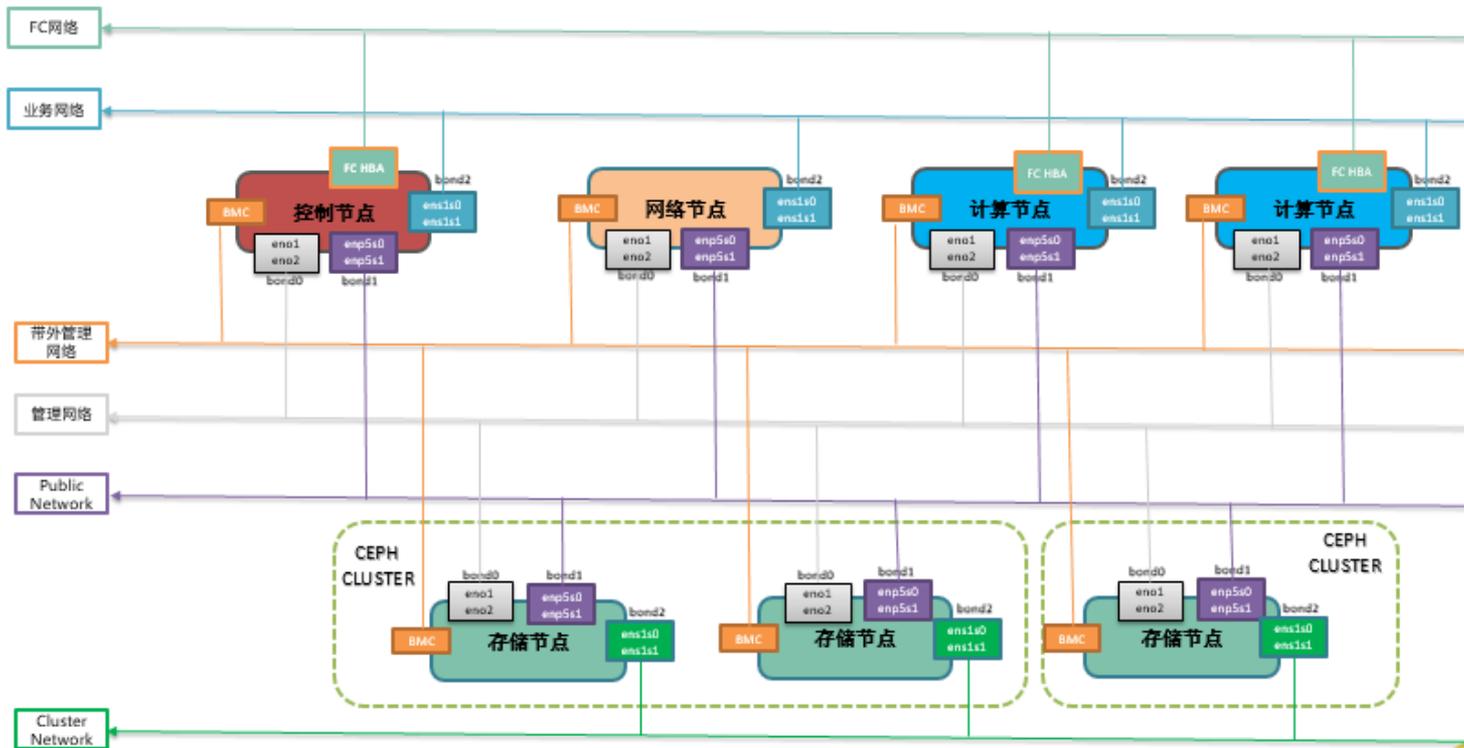


整体规划-RAID方案

节点类型	磁盘容量	RAID	磁盘用途
OpenStack	2*600G SAS盘	RAID1	操作系统+OpenStack
高性能ICFS存储	2*600G SAS盘 2*480G SSD盘	系统盘RAID1 数据盘单盘RAID0	操作系统+ICFS存储
ICFS存储	2*600G SAS盘 8*6T SATA盘	系统盘RAID1 数据盘单盘RAID0	操作系统+ICFS存储

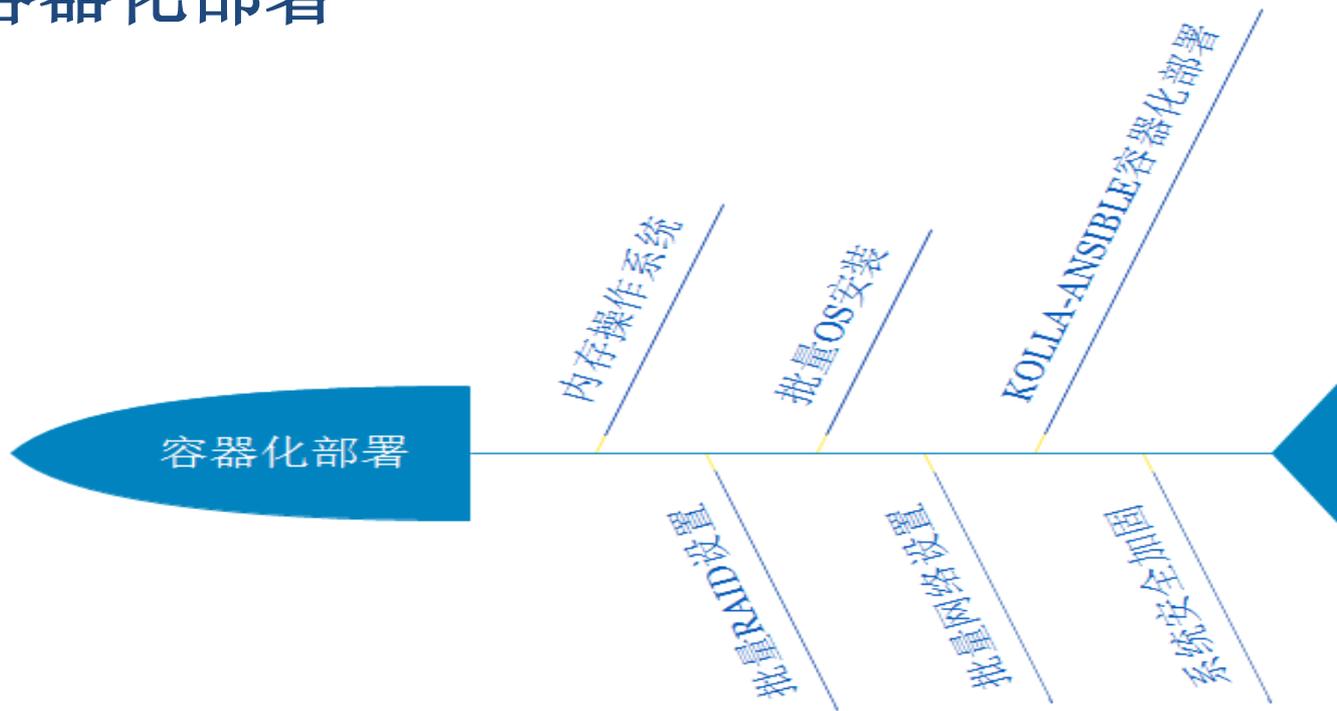


整体规划-网络方案





容器化部署





容器化部署-内存操作系统安装

- 部署节点提供容器化的cobbler服务，pxe装机支持三种启动方式
- 其中内存操作系统用于批量RAID

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。



容器化部署-批量RAID设置

节点类型	磁盘容量	RAID	磁盘用途
OpenStack	2*600G SAS盘	RAID1	操作系统+OpenStack
高性能ICFS存储	2*600G SAS盘 2*480G SSD盘	系统盘RAID1 数据盘单盘RAID0	操作系统+ICFS存储
ICFS存储	2*600G SAS盘 8*6T SATA盘	系统盘RAID1 数据盘单盘RAID0	操作系统+ICFS存储

- 内存系统固化了RAID管理配置工具Storcli及Arconf:

Storcli: 相对MegaCLI命令更加简洁, 并且兼容MegaCLI的命令

Arconf: 适用于adaptec公司的raid芯片管理工具



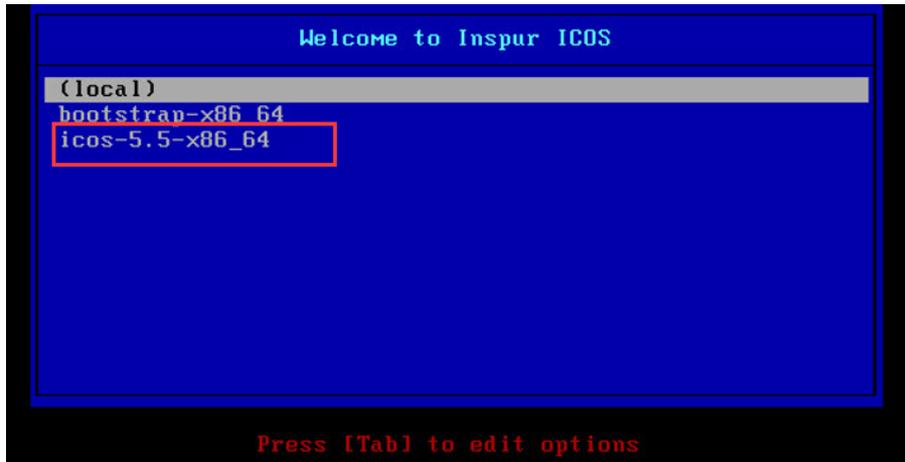
容器化部署-批量OS安装

- 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=
```

在这里设置系统的启动方式





容器化部署-域名解析及网络设置

• 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:

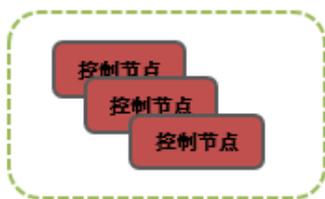
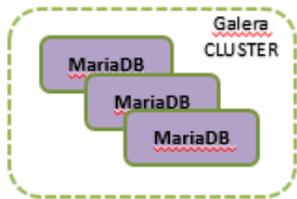
- ✓ 以机器序列号为识别标识
- ✓ sn_ip.file保存序列号、主机名及管理网ip间的映射关系
- ✓ network.info保存各网卡名bond名称及ip信息
- ✓ init_config.sh根据sn_ip.file对主机名进行设置
- ✓ config_bond.sh根据network.info对网络进行设置

```
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
```



容器化部署-角色分配

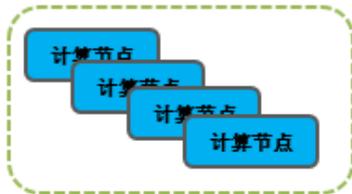
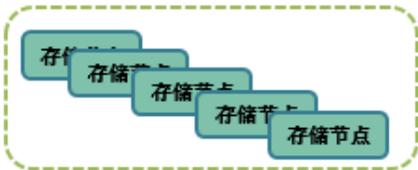
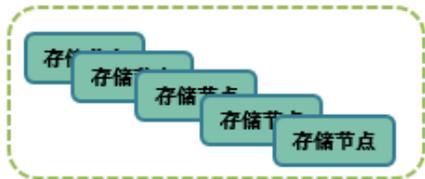


Nova-api
Glance-api
Cinder-api
Horizon
Neutron-Server
Telegraf
...

InfluxDB
Grafana
Telegraf
...



Neutron-OVS-Agent
Neutron-Vpnaas-Agent
Neutron-Lbaas-Agent
Neutron-Dhcp-Agent
Telegraf
...



Neutron-OVS-Agent
Nova-Compute
Nova-Libvirt
Telegraf
...



容器化部署-部署工具

部署工具

- 基于OpenStack开源项目：kolla、kolla-ansible
- 参数固化及自研模块的镜像构建及容器化部署支持

容器化部署优势

- Docker容器化技术将部署、升级操作的粒度减小到Service级别；
- 遵从幂等性原则，可以任意多次执行部署、升级等操作；
- 提高DevOps效率，快速开发迭代、快速部署；



- 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，额外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，部署之前修改角色分配、全局配置文件及自定义配置对接多存储后端，然后执行部署操作。

```
[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 }
```



容器化升级

容器化升级

- 能够在业务不中断的情况下完成平滑升级；
- Docker容器化技术将升级操作的粒度减小到Service级别；
- 表结构不变动的情况下，相当于升级OpenStack服务容器（功能模块代码升级），其中数据库增加表的情况下不影响升级；

Thank You

