****

**kolla-ansibe**

Contents

[一．安装准备 3](#_Toc535916578)

[1.1 主机配置 3](#_Toc535916579)

[1.2 下载依赖 3](#_Toc535916580)

[1.3 安装kolla-ansible 3](#_Toc535916581)

[二．docker配置 3](#_Toc535916582)

[2.1 admin节点docker安装 3](#_Toc535916583)

[2.2 docker registry配置 4](#_Toc535916584)

[2.3 构建镜像 4](#_Toc535916585)

[2.4 skip bug 5](#_Toc535916586)

[三．all-in-one安装 5](#_Toc535916587)

[# ~~3.1 初始化配置~~ 5](#_Toc535916588)

[四．mutil-node安装 6](#_Toc535916589)

[4.1 初始化配置 6](#_Toc535916590)

[4.2 部署 7](#_Toc535916591)

[五．附录： 7](#_Toc535916592)

[5.1参照： 7](#_Toc535916593)

[5.2 问题集： 8](#_Toc535916594)

[5.3 arm-debain docker installation and registry container startup 9](#_Toc535916595)

[5.4 error images 10](#_Toc535916596)

一．安装准备

1.1 主机配置

* 2个网卡
* 8G内存
* 40GB的磁盘空间

1.2 下载依赖

|  |
| --- |
| sudo apt update  sudo apt -y install python-pip  sudo apt -y install python-dev libffi-dev gcc libssl-dev python-selinux python-setuptools python-virtualenv  # 确保ansible版本为2.4及以上  pip install ansible |

1.3 安装kolla-ansible

|  |
| --- |
| virtualenv ./virtualenv  source virtualenv/bin/activate  pip install kolla-ansible  # sudo cp -rf virtualenv/share/kolla-ansible/etc\_examples/kolla/ /etc  # cp -rf virtualenv/share/kolla-ansible/ansible/ kolla-ansible  # 编辑inventory  cd virtualenv/share/kolla-ansible/ansible/inventory |

二．docker配置

2.1 admin节点docker安装

|  |
| --- |
| # doccker安装  sudo apt-get remove docker docker-engine docker.io  sudo apt-get update  sudo apt-get install apt-transport-https ca-certificates curl software-properties-common  curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  sudo apt-key fingerprint 0EBFCD88  sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"  sudo apt-get update  sudo apt-get install docker-ce  sudo systemctl status docker |

2.2 docker registry配置

|  |
| --- |
| # registry配置，可参照附录部分进行docker-registry的安装和registry的重新构建  sudo docker pull registry  sudo docker run -d -p 5000:5000 -v /home/yang/dockerrepo/:/var/lib/registry --restart=always --name registry registry  sudo vim /etc/docker/docker.json  …  {  "insecure-registries" : ["192.168.10.4:5000"]  }  …  sudo systemctl restart docker  # 浏览器查看  http:// 192.168.10.4:5000/v2/\_catalog |

2.3 构建镜像

|  |
| --- |
| # kolla-ansible中所有的镜像都可以使用kolla来构建，可以使用pip来进行安装  pip install kolla  pip install tox  git clone https://github.com/openstack/kolla.git  cd kolla  #生成配置文件  tox -e genconfig  sudo cp etc/kolla/kolla-build.conf /etc/kolla/  # 编译所有镜像  sudo kolla-build  # 将镜像上传到本地仓库  sudo kolla-build --push --regisgry localhost:5000 --tag master --namespace test  # 指定使用的docker基础image  kolla-build -b ubuntu  或修改kolla-build.conf文件的base=CENTOS, 可选的基础Image有这些，默认为centos：centos,debian,oraclelinux,rhel,Ubuntu  # 选定编译的镜像  kolla-build keystone nova  可选的镜像在kolla-build.conf的profiles部分定义；  该部分还把各个服务进行了分类，可以使用默认的分组或新增分组的方式来选择要编译的镜像：  kolla-build --profile magnum  # 镜像的编译方式有两种，source模式和binary模式，binary模式是使用apt/yum来安装各个服务，而source则是从源码编译服务，可以使用-t选项进行指定:  kolla-build -t source  或修改配置文件kolla-build.con中的install\_type = binary；  使用source进行安装的时候，可以进行如下配置  [zaqar-base]  # Source location type (string value)  # Possible values:  # local – 本地的.tar.gz源码压缩包  # git – 使用git从git仓库下载  # url -  #type = url  # The location for source install (string value)  #location = $tarballs\_base/zaqar/zaqar-master.tar.gz  # Git reference to pull, commit sha, tag or branch name (string value)  #reference = <None> |
|  |

~~2.4 skip bug~~

|  |
| --- |
| * RUNNING HANDLER [mariadb : Waiting for master mariadb]   kolla-ansible/ansible/roles/mariadb/handlers/main.yaml文件注释job - name: Waiting for master mariadb |

三．all-in-one安装

# ~~3.1 初始化配置~~

|  |
| --- |
| ansible -i inventory/all-in-one all -m ping  sudo kolla-genpwd # 生成密码文件/etc/kolla/ passwords.yml，所有服务的密码都在该文件中，默认为都是空的，如果没有设置则生成随机密码，使用该命令可生成随机密码。  # 配置globals.yml |

四．mutil-node安装

4.1 初始化配置

|  |
| --- |
| # admin节点docker配置，同all-in-one  # admin节点docker registry配置  # ssh配置，安装ssh并启动sshd服务  sudo apt install ssh  sudo systemctl start sshd  # admin节点hosts解析配置  yang@admin:~$ sudo cat /etc/hosts  …  192.168.10.5 compute1  192.168.10.6 compute2  192.168.10.7 control1  192.168.10.8 monitor1  192.168.10.9 network1  192.168.10.10 storage1  192.168.10.11 storage2  192.168.10.12 storage3  …  # virtualenv/share/kolla-ansible/ansible  # 配置intentory文件  …  [control]  control1 ansible\_become=true  [network]  network1 ansible\_become=true  [external-compute]  compute1 ansible\_become=true  compute2 ansible\_become=true  [monitoring]  monitor1 ansible\_become=true  [storage]  storage1 ansible\_become=true  storage2 ansible\_become=true  storage3 ansible\_become=true  # ssh无密登陆配置  ssh-keygen  ssh-copy-id [datemon-host-name]  # ansible连接测试  ansible -i inventory/multinode all -m ping  # globals.xml配置  kolla\_base\_distro: "centos"  kolla\_install\_type: "binary"  openstack\_release: "master"  kolla\_internal\_vip\_address: "192.168.10.100"  docker\_registry: "192.168.10.4:5000"  docker\_namespace: "test"  network\_interface: "enp1s0"  neutron\_external\_interface: "enp6s0"  enable\_ceph: "yes"  enable\_ceph\_mds: "yes"  enable\_ceph\_rgw: "yes"  enable\_cinder: "yes"  enable\_cinder\_backend\_lvm: "yes"  enable\_fluentd: "no"  glance\_backend\_ceph: "yes"  glance\_backend\_file: "no"  ceph\_rgw\_compatibility: "True"  ceph\_osd\_store\_type: "bluestore"  # prepare osd for koll on storage nodes  # sudo parted [device] -s -- mklabel gpt mkpart KOLLA\_CEPH\_OSD\_BOOTSTRAP\_BS 1 -1  sudo apt install parted  sudo parted /dev/sdb -s -- mklabel gpt mkpart KOLLA\_CEPH\_OSD\_BOOTSTRAP\_BS 1 -1 |

4.2 部署

|  |
| --- |
| kolla-ansible -i inventory/multinode bootstrap-servers  kolla-ansible -i inventory/multinode prechecks  kolla-ansible -i inventory/multinode deploy |

五．附录：

5.1参照：

【1】https://docs.openstack.org/kolla-ansible/latest/

【2】https://github.com/openstack/kolla-ansible

【3】https://docs.openstack.org/kolla/newton/image-building.html#guide

【4】https://github.com/openstack/kolla/tree/master/docker

5.2 问题集：

|  |
| --- |
| 问题1：('Connection aborted.', error(13, 'Permission denied')  解决1：  inventory文件中ansible\_become=true  问题2：  [ERROR] Could not open mysql.plugin table. Some plugins may be not loaded  问题3：  u"Table 'keystone.project' doesn't exist")  问题4：  :No package nfs-ganesha-ceph available.  问题5：  "msg": "The conditional check '(ceph\_files\_json.stdout | from\_json).changed' failed. The error was: No JSON object could be decoded"  问题6：  Failed to get D-Bus connection: Operation not permitted  问题7：  libcollection-0.7.0-32.el7.aarch64.rpm librados2-12.2.5-0.el7.aarch64.rpm  Operation too slow. Less than 1000 bytes/sec transferred the last 30 seconds  问题8：  rpc error: code = 2 desc = containerd: container not found  解决：重启docker服务  sudo systemctl restart docker  问题9：  roles/nova/tasks/simple\_cell\_setup.yml  fatal: [control1 -> control1]: FAILED! => {"changed": false, "cmd": ["docker", "exec", "nova\_api", "nova-manage", "cell\_v2", "map\_cell0"],  问题10：  oslo\_config.cfg.NoSuchOptError: no such option nfast-user in group [DEFAULT] |

* 1. arm-debain docker installation and registry container startup

|  |
| --- |
| sudo apt-get remove docker docker-engine docker.io  sudo apt update  sudo apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common  sudo curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add –  sudo apt-key fingerprint 0EBFCD88  sudo echo "deb [arch=armhf] https://download.docker.com/linux/debian \  $(lsb\_release -cs) stable" | \  sudo tee /etc/apt/sources.list.d/docker.list  sudo apt update  sudo apt install docker-ce  sudo systemctl status docker  sudo docker run hello-world  # install registry with deb package  wget http://ftp.de.debian.org/debian/pool/main/d/docker-registry/docker-registry\_2.6.2~ds1-2\_arm64.deb  sudo dpkg -i docker-registry\_2.6.2~ds1-2\_arm64.deb  sudo systemctl status docker-registry  # build registry image with Dockerfile  git clone https://github.com/docker/distribution-library-image.git  cd distribution-library-image/arm64  cat Dockerfile  …  FROM alpine:latest  ..  sudo docker build -t registry:latest .  # test registry image  sudo docker run -d –p 5000:5000 –v /home/yang/registry/registry-repo/ --name registry registry  # view in web browser with url http://ip-address:5000/v2/\_catalog  sudo docker tag hello-world localhost:5000/hello-world:v1.0  sudo docker push localhost:5000/hello-world  # view again in web. |

* 1. error images

|  |
| --- |
| INFO:kolla.common.utils:===========================  INFO:kolla.common.utils:Images that failed to build  INFO:kolla.common.utils:===========================  ERROR:kolla.common.utils:almanach-api Failed with status: matched  ERROR:kolla.common.utils:almanach-collector Failed with status: matched  ERROR:kolla.common.utils:bifrost-deploy Failed with status: matched  ERROR:kolla.common.utils:blazar-api Failed with status: matched  ERROR:kolla.common.utils:blazar-manager Failed with status: matched  ERROR:kolla.common.utils:dragonflow-controller Failed with status: matched  ERROR:kolla.common.utils:dragonflow-metadata Failed with status: matched  ERROR:kolla.common.utils:dragonflow-publisher-service Failed with status: matched  ERROR:kolla.common.utils:elasticsearch Failed with status: error  ERROR:kolla.common.utils:freezer-api Failed with status: matched  ERROR:kolla.common.utils:freezer-scheduler Failed with status: matched  ERROR:kolla.common.utils:grafana Failed with status: error  ERROR:kolla.common.utils:helm-repository Failed with status: error  ERROR:kolla.common.utils:influxdb Failed with status: error  ERROR:kolla.common.utils:kafka Failed with status: error  ERROR:kolla.common.utils:karbor-api Failed with status: matched  ERROR:kolla.common.utils:karbor-operationengine Failed with status: matched  ERROR:kolla.common.utils:karbor-protection Failed with status: matched  ERROR:kolla.common.utils:kibana Failed with status: error  ERROR:kolla.common.utils:kube-apiserver Failed with status: error  ERROR:kolla.common.utils:kube-controller-manager Failed with status: error  ERROR:kolla.common.utils:kube-discovery Failed with status: error  ERROR:kolla.common.utils:kube-proxy Failed with status: error  ERROR:kolla.common.utils:kube-scheduler Failed with status: error  ERROR:kolla.common.utils:kuryr-libnetwork Failed with status: matched  ERROR:kolla.common.utils:logstash Failed with status: error  ERROR:kolla.common.utils:monasca-agent Failed with status: matched  ERROR:kolla.common.utils:monasca-api Failed with status: matched  ERROR:kolla.common.utils:monasca-log-api Failed with status: matched  ERROR:kolla.common.utils:monasca-notification Failed with status: matched  ERROR:kolla.common.utils:monasca-persister Failed with status: matched  ERROR:kolla.common.utils:mongodb Failed with status: error  ERROR:kolla.common.utils:opendaylight Failed with status: error  ERROR:kolla.common.utils:ovsdpdk-db Failed with status: matched  ERROR:kolla.common.utils:ovsdpdk-vswitchd Failed with status: matched  ERROR:kolla.common.utils:searchlight-api Failed with status: matched  ERROR:kolla.common.utils:searchlight-listener Failed with status: matched  ERROR:kolla.common.utils:sensu-client Failed with status: error  ERROR:kolla.common.utils:solum-api Failed with status: matched  ERROR:kolla.common.utils:solum-conductor Failed with status: matched  ERROR:kolla.common.utils:solum-deployer Failed with status: matched  ERROR:kolla.common.utils:solum-worker Failed with status: matched  ERROR:kolla.common.utils:telegraf Failed with status: error  ERROR:kolla.common.utils:zun-api Failed with status: matched  ERROR:kolla.common.utils:zun-compute Failed with status: matched  ERROR:kolla.common.utils:zun-wsproxy Failed with status: matched |

* 1. configure debian NIC

|  |
| --- |
| # This file describes the network interfaces available on your system  # and how to activate them. For more information, see interfaces(5).  source /etc/network/interfaces.d/\*  # The loopback network interface  auto lo  iface lo inet loopback  # The primary network interface  allow-hotplug enp1s0  # iface enp1s0 inet dhcp  iface enp1s0 inet static  address 192.168.10.5  netmask 255.255.255.0  # gateway 192.168.0.1  allow-hotplug enp6s0  iface enp6s0 inet dhcp  allow-hotplug enp6s0  iface enp6s0 inet static  address 192.168.10.6  netmask 255.255.255.0 |