Q1: 请记录所有安装步骤的指令，并简要描述其含义

以下是所有安装步骤的总结，以备自己日后所需

下述内容仅对安装K8S和创建K8S集群部分的指令进行阐释

安装docker

```
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg lsb-release
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-
by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
#安装一些必要的系统工具
apt-get -y install apt-transport-https ca-certificates curl software-properties-
common
```

配置docker

```
# 添加 docker 配置 /etc/docker/daemon.json
{
  "exec-opts": [
    "native.cgroupdriver=systemd"
  ],
  "bip":"172.12.0.1/24",
  "registry-mirrors": [
    "http://docker-registry-mirror.kodekloud.com"
  ]
}
```

需要设置cgroupdriver=systemd，否则会无法正常初始化。因为kubelet的cgroup driver是systemd，docker默认的 cgroup driver是cgroupfs，两者不一致导致kubelet启动失败。

```
#之后验证kubelet的cgroup driver是systemd
root@cloud1:/etc/docker# cat /var/lib/kubelet/config.yaml |grep group
cgroupDriver: systemd
```

安装K8S

- 在每台机器上执行(需要sudo权限)

```
# install dependency
apt-get update
apt-get install -y apt-transport-https ca-certificates curl
```

```
# add key
curl https://mirrors.aliyun.com/kubernetes/apt/doc/apt-key.gpg | apt-key add -

# add apt source
cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
deb https://mirrors.aliyun.com/kubernetes/apt/ kubernetes-xenial main
EOF

# install kube* and enable kubelet
apt-get update
apt-get install -y kubelet kubeadm kubectl
systemctl enable kubelet
```

创建K8S集群

在master(Cloud1)上执行

```
# 关闭swap，否则将导致K8S无法初始化
swapoff -a

# init and start master
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-
address=192.168.1.7 --image-repository registry.aliyuncs.com/google_containers -
-kubernetes-version 1.23.6

# To start using your cluster, you need to run the following as a regular user:
# copy config file
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

# install weave in the k8s cluster
kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl
version | base64 | tr -d '\n')"
```

- 此时集群已经创建好，kubectl get pod --all-namespaces可以查看组件运行成功

```
root@cloud1:/etc/docker# kubectl get nodes
NAME     STATUS   ROLES                  AGE     VERSION
cloud1   Ready    control-plane,master   86m     v1.23.6
```

- 在需要加入集群的node（cloud2）上执行

```
// worker node joins into cluster
kubeadm join 192.168.1.7:6443 --token fub2tb.48r4esila0zfm170 \
        --discovery-token-ca-cert-hash
sha256:71b5b16771bec25d0f3fb19f30e547e60003eea15a3701825df07845af2c34e2
```

现在执行kubectl get pod --all-namespaces可以查看集群运行成功

```
root@cloud1:/etc/docker# kubectl get nodes
NAME     STATUS   ROLES                  AGE     VERSION
cloud1   Ready    control-plane,master   86m     v1.23.6
cloud2   Ready    <none>                 2m38s   v1.23.6
```

Q2: 在两个节点上分别使用 ps aux | grep kube 列出所有和k8s相关的进程，记录其输出，并简要说明各个进程的作用

master结点（cloud1）：

```
root@cloud1:/etc/docker# ps aux | grep kube
root       38253  4.1  3.9 1111044 317520 ?      Ssl  16:19   6:27 kube-apiserver --advertise-address=192.168.1.7 --allow-privileged=true --authorization-mode=Node,RBAC --client-ca-file=/etc/kubernetes/pki/ca.crt --enable-admission-plugins=NodeRestriction --enable-bootstrap-token-auth=true --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key --etcd-servers=https://127.0.0.1:2379 --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key --requestheader-allowed-names=front-proxy-client --requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt --requestheader-extra-headers-prefix=X-Remote-Extra- --requestheader-group-headers=X-Remote-Group --requestheader-username-headers=X-Remote-User --secure-port=6443 --service-account-issuer=https://kubernetes.default.svc.cluster.local --service-account-key-file=/etc/kubernetes/pki/sa.pub --service-account-signing-key-file=/etc/kubernetes/pki/sa.key --service-cluster-ip-range=10.96.0.0/12 --tls-cert-file=/etc/kubernetes/pki/apiserver.crt --tls-private-key-file=/etc/kubernetes/pki/apiserver.key
root       38304  0.2  0.6 754548 53180 ?        Ssl  16:19   0:25 kube-scheduler --authentication-kubeconfig=/etc/kubernetes/scheduler.conf --authorization-kubeconfig=/etc/kubernetes/scheduler.conf --bind-address=127.0.0.1 --kubeconfig=/etc/kubernetes/scheduler.conf --leader-elect=true
root       38374  1.3  0.7 11215036 62124 ?      Ssl  16:19   2:00 etcd --advertise-client-urls=https://192.168.1.7:2379 --cert-file=/etc/kubernetes/pki/etcd/server.crt --client-cert-auth=true --data-dir=/var/lib/etcd --initial-advertise-peer-urls=https://192.168.1.7:2380 --initial-cluster=cloud1=https://192.168.1.7:2380 --key-file=/etc/kubernetes/pki/etcd/server.key --listen-client-urls=https://127.0.0.1:2379,https://192.168.1.7:2379 --listen-metrics-urls=http://127.0.0.1:2381 --listen-peer-urls=https://192.168.1.7:2380 --name=cloud1 --peer-cert-file=/etc/kubernetes/pki/etcd/peer.crt --peer-client-cert-auth=true --peer-key-file=/etc/kubernetes/pki/etcd/peer.key --peer-trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt --snapshot-count=10000 --trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt
root       38387  1.2  1.3 825432 108172 ?       Ssl  16:19   1:55 kube-controller-manager --allocate-node-cidrs=true --authentication-kubeconfig=/etc/kubernetes/controller-manager.conf --authorization-kubeconfig=/etc/kubernetes/controller-manager.conf --bind-address=127.0.0.1 --client-ca-file=/etc/kubernetes/pki/ca.crt --cluster-cidr=10.244.0.0/16 --cluster-name=kubernetes --cluster-signing-cert-file=/etc/kubernetes/pki/ca.crt --cluster-signing-key-file=/etc/kubernetes/pki/ca.key --controllers=*,bootstrapsigner,tokencleaner --kubeconfig=/etc/kubernetes/controller-manager.conf --leader-elect=true --requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt --root-ca-file=/etc/kubernetes/pki/ca.crt --service-account-private-key-file=/etc/kubernetes/pki/sa.key --service-cluster-ip-range=10.96.0.0/12 --use-service-account-credentials=true
root       38619  2.5  1.3 2011480 110364 ?      Ssl  16:19   3:57 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/lib/kubelet/config.yaml --network-plugin=cni --pod-infra-container-image=registry.aliyuncs.com/google_containers/pause:3.6
root       38960  0.0  0.4 748436 39108 ?        Ssl  16:19   0:01 /usr/local/bin/kube-proxy --config=/var/lib/kube-proxy/config.conf --hostname-override=cloud1
root       41410  0.0  0.3 732188 24536 ?        Sl   16:24   0:00 /home/weave/kube-utils -run-reclaim-daemon -node-name=cloud1 -peer-name=6a:a4:9b:cd:b2:ff -log-level=debug
root       92589  0.0  0.0   8160  2404 pts/0    S+   18:53   0:00 grep --color=auto kube
```

- kube-apiserver：负责与kubelet，controller，etcd，scheduler等组件交互，提供api接口，也是唯一与etcd进行直接交互的组件
- kube-scheduler：在kubectl发送消息想要创建pod后，根据一些判断依据选择调度将pod在哪个node创建
- etcd：是一个分布式一致性键值存储系统，存储K8S所需的各种信息，用于共享配置和服务发现等
- kube-controller-manager：作为集群内部的管理控制中心，负责集群内的Node、Pod、Namespace等的管理，当出现什么意外时，Controller Manager会及时发现并执行自动化修复流程，确保集群始终处于预期的工作状态。
- kubelet：与apiserver进行交互，获取pod相关的数据，监控当前的Pod变化的事件；操作当前node的资源信息，并启动Pod等。
- kube-proxy：是管理service的访问入口，管理包括集群内Pod到Service的访问和集群外访问service
- kube-utils：管理pod的网络插件

worker结点（cloud2）：

```
root@cloud2:~# ps aux | grep kube
root       39835  1.6  1.2 1937236 101056 ?      Ssl  17:43   1:08 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/lib/kubelet/config.yaml --network-plugin=cni --pod-infra-container-image=registry.aliyuncs.com/google_containers/pause:3.6
root       40351  0.0  0.4 748436 38796 ?        Ssl  17:43   0:01 /usr/local/bin/kube-proxy --config=/var/lib/kube-proxy/config.conf --hostname-override=cloud2
root       41024  0.0  0.2 732188 24380 ?        Sl   17:44   0:00 /home/weave/kube-utils -run-reclaim-daemon -node-name=cloud2 -peer-name=d2:34:f0:19:17:aa -log-level=debug
root       64364  0.0  0.0   8160  2576 pts/0    S+   18:54   0:00 grep --color=auto kube
```

- kubelet：与apiserver进行交互，获取pod相关的数据，监控当前的Pod变化的事件；操作当前node的资源信息，并启动Pod等。
- kube-proxy：是管理service的访问入口，管理包括集群内Pod到Service的访问和集群外访问service
- kube-utils：管理pod的网络插件

Q3: 在两个节点中分别使用 docker ps 显示所有正常运行的docker容器，记录其输出，并简要说明各个容器所包含的k8s组件，以及哪些k8s组件未运行在容器中

master结点（cloud1）：

```
root@cloud1:/etc/docker# docker ps
CONTAINER ID   IMAGE                                                COMMAND                  CREATED       STATUS       PORTS     NAMES
c0c4ebc97a7b   a4ca41631cc7                                         "/coredns -conf /etc…"   3 hours ago   Up 3 hours             k8s_coredns_coredns-6d8c4cb4d-f48
qr_kube-system_727e9c44-e4c8-48bf-9186-af78697feb38_0
ae16bd5b3c44   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 3 hours ago   Up 3 hours             k8s_POD_coredns-6d8c4cb4d-f48qr_k
ube-system_727e9c44-e4c8-48bf-9186-af78697feb38_1
5f27206b2467   a4ca41631cc7                                         "/coredns -conf /etc…"   3 hours ago   Up 3 hours             k8s_coredns_coredns-6d8c4cb4d-wzk
5c_kube-system_ea0bc7a0-9684-4c84-aa2e-de2576da5b27_0
3bc51b82c3b4   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 3 hours ago   Up 3 hours             k8s_POD_coredns-6d8c4cb4d-wzk5c_k
ube-system_ea0bc7a0-9684-4c84-aa2e-de2576da5b27_1
b950c417d6e5   df29c0a4002c                                         "/home/weave/launch.…"   3 hours ago   Up 3 hours             k8s_weave_weave-net-vlf6j_kube-sy
stem_2bcd2efb-d772-4eb5-9095-2e31b741f6d0_1
bcf01e055d02   ghcr.io/weaveworks/launcher/weave-npc                "/usr/bin/launch.sh"     3 hours ago   Up 3 hours             k8s_weave-npc_weave-net-vlf6j_kub
e-system_2bcd2efb-d772-4eb5-9095-2e31b741f6d0_0
f8ac74023828   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 3 hours ago   Up 3 hours             k8s_POD_weave-net-vlf6j_kube-syst
em_2bcd2efb-d772-4eb5-9095-2e31b741f6d0_0
8bc6a8943f98   4c0375452406                                         "/usr/local/bin/kube…"   3 hours ago   Up 3 hours             k8s_kube-proxy_kube-proxy-5tkmp_k
ube-system_fed46f88-ad1e-44a4-871f-0c28143fcf40_0
cf8784d144db   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 3 hours ago   Up 3 hours             k8s_POD_kube-proxy-5tkmp_kube-sys
tem_fed46f88-ad1e-44a4-871f-0c28143fcf40_0
b569cfd69c19   df7b72818ad2                                         "kube-controller-man…"   3 hours ago   Up 3 hours             k8s_kube-controller-manager_kube-
controller-manager-cloud1_kube-system_1d619d2830fcd8355dbeaa3cb11717b9_2
80518788d6ab   25f8c7f3da61                                         "etcd --advertise-cl…"   3 hours ago   Up 3 hours             k8s_etcd_etcd-cloud1_kube-system_
1a7a8fe18f0b41fea46fe533f36e7dea_1
25eb894e2d85   595f327f224a                                         "kube-scheduler --au…"   3 hours ago   Up 3 hours             k8s_kube-scheduler_kube-scheduler
-cloud1_kube-system_b34dbe7b194cf50b73d35c5d64c7e926_2
79301513cefe   8fa62c12256d                                         "kube-apiserver --ad…"   3 hours ago   Up 3 hours             k8s_kube-apiserver_kube-apiserver
-cloud1_kube-system_794097a1953a618d205b0bfafcba9406_1
fc141c14903e   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 3 hours ago   Up 3 hours             k8s_POD_kube-scheduler-cloud1_kub
e-system_b34dbe7b194cf50b73d35c5d64c7e926_0
c5a220a3f043   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 3 hours ago   Up 3 hours             k8s_POD_kube-controller-manager-c
loud1_kube-system_1d619d2830fcd8355dbeaa3cb11717b9_0
741e1130d8bd   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 3 hours ago   Up 3 hours             k8s_POD_kube-apiserver-cloud1_kub
e-system_794097a1953a618d205b0bfafcba9406_0
d05804796abd   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 3 hours ago   Up 3 hours             k8s_POD_etcd-cloud1_kube-system_1
a7a8fe18f0b41fea46fe533f36e7dea_0
```

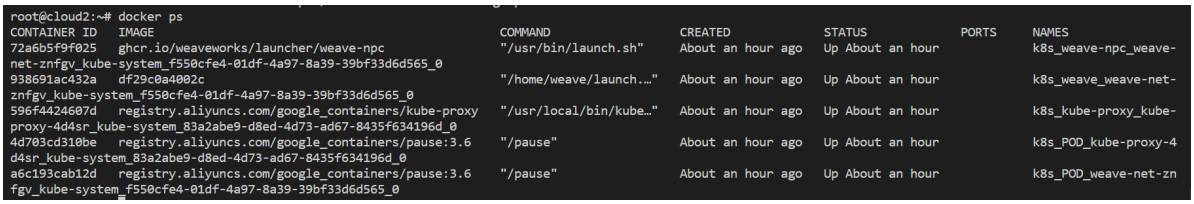
包含了K8S组件的容器有：

容器名	组件
k8s_kube-controller-manager_kube-controller-manager-cloud1_kube-system_1d619d2830fcd8355dbeaa3cb11717b9_2	controller-manager
k8s_etcd_etcd-cloud1_kube-system_1a7a8fe18f0b41fea46fe533f36e7dea_1	etcd
k8s_kube-scheduler_kube-scheduler-cloud1_kube-system_b34dbe7b194cf50b73d35c5d64c7e926_2	kube-scheduler
k8s_kube-apiserver_kube-apiserver-cloud1_kube-system_794097a1953a618d205b0bfafcba9406_1	apiserver
k8s_kube-proxy_kube-proxy-5tkmp_kube-system_fed46f88-ad1e-44a4-871f-0c28143fcf40_0	kube-proxy
k8s_weave_weave-net-vlf6j_kube-system_2bcd2efb-d772-4eb5-9095-2e31b741f6d0_1和k8s_weave-npc_weave-net-vlf6j_kube-system_2bcd2efb-d772-4eb5-9095-2e31b741f6d0_0	采用的weave网络插件
k8s_coredns_coredns-6d8c4cb4d-f48qr_kube-system_727e9c44-e4c8-48bf-9186-af78697feb38_0	提供DNS服务的coredns插件

未运行在容器中的K8S组件：

kubectl

worker结点（cloud2）：



```
root@cloud2:~# docker ps
CONTAINER ID   IMAGE                                                COMMAND                  CREATED             STATUS             PORTS     NAMES
72a6b5f9f025   ghcr.io/weaveworks/launcher/weave-npc                "/usr/bin/launch.sh"     About an hour ago   Up About an hour             k8s_weave-npc_weave-
net-znfgv_kube-system_f550cfe4-01df-4a97-8a39-39bf33d6d565_0
938691ac432a   df29c0a4002c                                         "/home/weave/launch.…"   About an hour ago   Up About an hour             k8s_weave_weave-net-
znfgv_kube-system_f550cfe4-01df-4a97-8a39-39bf33d6d565_0
596f4424607d   registry.aliyuncs.com/google_containers/kube-proxy   "/usr/local/bin/kube…"   About an hour ago   Up About an hour             k8s_kube-proxy_kube-
proxy-4d4sr_kube-system_83a2abe9-d8ed-4d73-ad67-8435f634196d_0
4d703cd310be   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 About an hour ago   Up About an hour             k8s_POD_kube-proxy-4
d4sr_kube-system_83a2abe9-d8ed-4d73-ad67-8435f634196d_0
a6c193cab12d   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 About an hour ago   Up About an hour             k8s_POD_weave-net-zn
fgv_kube-system_f550cfe4-01df-4a97-8a39-39bf33d6d565_0
```

包含了K8S组件的容器有：

容器名	组件
k8s_weave-npc_weave-net-znfgv_kube-system_f550cfe4-01df-4a97-8a39-39bf33d6d565_0和k8s_weave_weave-net-znfgv_kube-system_f550cfe4-01df-4a97-8a39-39bf33d6d565_0	采用的weave网络插件
k8s_kube-proxy_kube-proxy-4d4sr_kube-system_83a2abe9-d8ed-4d73-ad67-8435f634196d_0	kube-proxy

未运行在容器中的K8S组件：

kubelet

Q4: 请采用声明式接口对Pod进行部署，并将部署所需的yml文件记录在实践文档中

```
apiVersion: v1
kind: Pod
metadata:
  name: hw-pod
  labels:
    name: hw-pod

spec:
  containers:
    - name: viewer
      image: dplsming/nginx-fileserver:1.0
      command:
      args:
      workingDir:
      volumeMounts:
        - name: nginx-volume
          mountPath: /usr/share/nginx/html/files

      ports:
        - name: viewer-port
          containerPort: 80
          hostPort: 80
          protocol: TCP

    - name: downloader
      image: dplsming/aria2ng-downloader:1.0
      command:
      args:
```

```
      workingDir:
      volumeMounts:
        - name: downloader-volume
          mountPath: /data

      ports:
        - name: down-port1
          containerPort: 6800
          hostPort: 6800
          protocol: TCP
        - name: down-port2
          containerPort: 6880
          hostPort: 6880
          protocol: TCP

  volumes:
    - name: nginx-volume
      hostPath:
        path: /hw
    - name: downloader-volume
      hostPath:
        path: /hw
```

Q5: 请在worker节点上，在部署Pod的前后分别采用 docker ps 查看所有运行中的容器并对比两者的区别。请将创建该Pod所创建的全部新容器列举出来，并一一解释其作用

部署前：

```
root@cloud2:~# docker ps
CONTAINER ID   IMAGE                                                COMMAND                  CREATED         STATUS         PORTS     NAMES
72a6b5f9f025   ghcr.io/weaveworks/launcher/weave-npc                "/usr/bin/launch.sh"     30 hours ago    Up 30 hours              k8s_weave-npc_weave-net-znfgv_
kube-system_f550cfe4-01df-4a97-8a39-39bf33d6d565_0
938691ac432a   df29c0a4002c                                         "/home/weave/launch.…"   30 hours ago    Up 30 hours              k8s_weave_weave-net-znfgv_kube
-system_f550cfe4-01df-4a97-8a39-39bf33d6d565_0
596f4424607d   registry.aliyuncs.com/google_containers/kube-proxy   "/usr/local/bin/kube…"   30 hours ago    Up 30 hours              k8s_kube-proxy_kube-proxy-4d4s
r_kube-system_83a2abe9-d8ed-4d73-ad67-8435f634196d_0
4d703cd310be   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 30 hours ago    Up 30 hours              k8s_POD_kube-proxy-4d4sr_kube-
system_83a2abe9-d8ed-4d73-ad67-8435f634196d_0
a6c193cab12d   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 30 hours ago    Up 30 hours              k8s_POD_weave-net-znfgv_kube-s
ystem_f550cfe4-01df-4a97-8a39-39bf33d6d565_0
```

部署后：

```
root@cloud2:~# docker ps
CONTAINER ID   IMAGE                                                COMMAND                  CREATED         STATUS         PORTS     NAMES
2b427bf50f96   dplsming/aria2ng-downloader                          "/conf-copy/start.sh"    2 minutes ago   Up 2 minutes             k8s_downloader_hw-pod_defaul
t_25a39314-2236-4b55-8176-9676f46a5a6c_0
14786550d1f9   dplsming/nginx-fileserver                            "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes             k8s_viewer_hw-pod_default_25
a39314-2236-4b55-8176-9676f46a5a6c_0
2d82bb254c50   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 2 minutes ago   Up 2 minutes             k8s_POD_hw-pod_default_25a39
314-2236-4b55-8176-9676f46a5a6c_0
72a6b5f9f025   ghcr.io/weaveworks/launcher/weave-npc                "/usr/bin/launch.sh"     30 hours ago    Up 30 hours              k8s_weave-npc_weave-net-znfg
v_kube-system_f550cfe4-01df-4a97-8a39-39bf33d6d565_0
938691ac432a   df29c0a4002c                                         "/home/weave/launch.…"   30 hours ago    Up 30 hours              k8s_weave_weave-net-znfgv_ku
be-system_f550cfe4-01df-4a97-8a39-39bf33d6d565_0
596f4424607d   registry.aliyuncs.com/google_containers/kube-proxy   "/usr/local/bin/kube…"   30 hours ago    Up 30 hours              k8s_kube-proxy_kube-proxy-4d
4sr_kube-system_83a2abe9-d8ed-4d73-ad67-8435f634196d_0
4d703cd310be   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 30 hours ago    Up 30 hours              k8s_POD_kube-proxy-4d4sr_kub
e-system_83a2abe9-d8ed-4d73-ad67-8435f634196d_0
a6c193cab12d   registry.aliyuncs.com/google_containers/pause:3.6    "/pause"                 30 hours ago    Up 30 hours              k8s_POD_weave-net-znfgv_kube
-system_f550cfe4-01df-4a97-8a39-39bf33d6d565_0
```

新增容器：

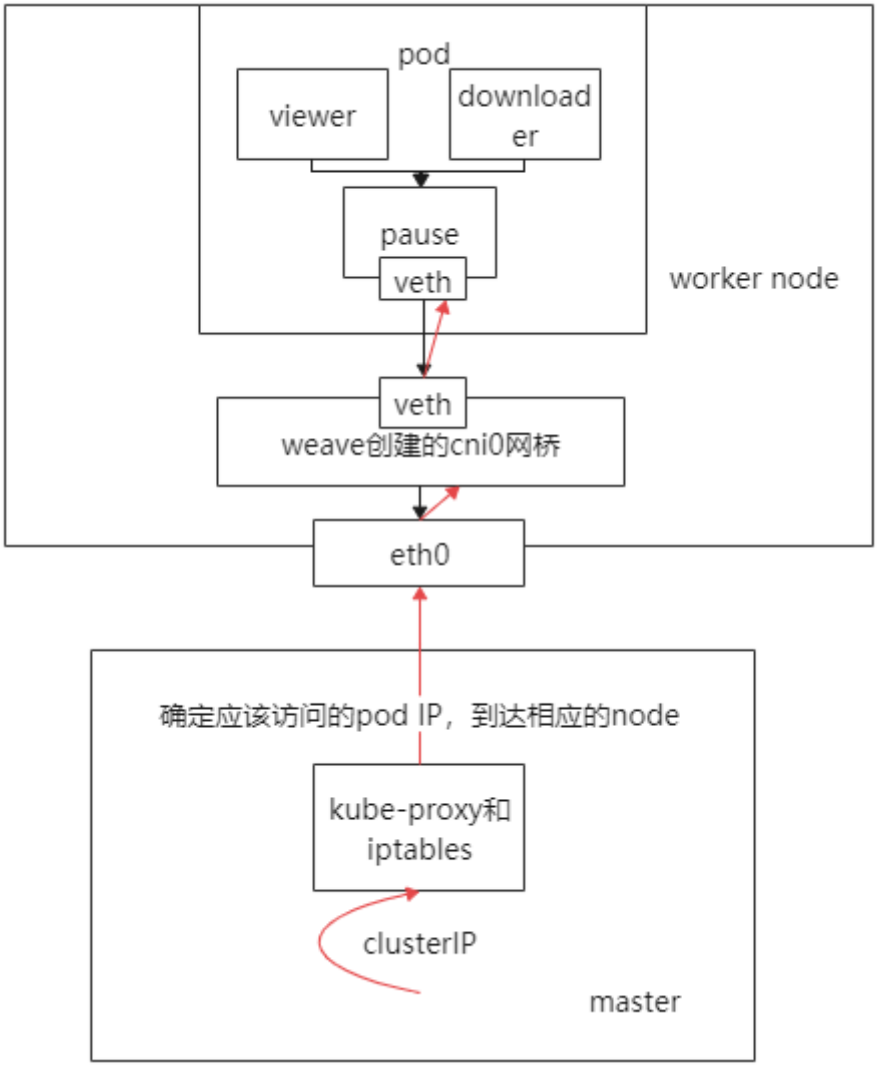
容器名	镜像	作用
k8s_downloader_hw-pod_default_25a39314-2236-4b55-8176-9676f46a5a6c_0	dplsming/aria2ng-downloader	系实验主动创建容器，提供文件目录共享服务。利用nginx服务器，将主要目录下的所有文件以http协议对外共享
k8s_viewer_hw-pod_default_25a39314-2236-4b55-8176-9676f46a5a6c_0	dplsming/nginx-fileserver	系实验主动创建的容器件，提供下载服务。基于开源项目轻微改动而来， 利用aria2完成下载功能，并利用dark-httpd 和aria2-ng为aria2提供 了一个基于web页面的 简单UI，
k8s_POD_hw-pod_default_25a39314-2236-4b55-8176-9676f46a5a6c_0	registry.aliyuncs.com/google_containers/pause:3.6	系pod内自动创建的pause容器，为pod中的其他容器提供网络。 其它的容器再采用共享container的网络模式来共享这个pause的网络即可与外部进行通信。

Q6: 请结合博客 https://blog.51cto.com/u_15069443/4043930 的内容，将容器中的veth与host机器上的veth匹配起来，并采用 ip link 和 ip addr 指令找到位于host机器中的所有网络设备及其之间的关系。结合两者的输出，试绘制出worker节点中涉及新部署Pod的所有网络设备和其网络结构，并在图中标注出从master节点中使用cluster ip访问位于worker节点中的Pod的网络路径

```
root@cloud2:~# docker exec -it c7c4bbf7724b /bin/bash
bash-5.1# ip link show eth0
15: eth0@if16: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1376 qdisc noqueue state UP
    link/ether 0a:5e:95:5f:f7:66 brd ff:ff:ff:ff:ff:ff
bash-5.1# exit
exit
root@cloud2:~# ip link show | grep 16
    link/ether fa:16:3e:ab:89:4f brd ff:ff:ff:ff:ff:ff
16: vethwepl52d4196@if15: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1376 qdisc noqueue master weave state UP mode DEFAULT group default
```

可以看出容器的veth为15，与之对应的host机器上的veth为16

画出worker中网络设备结构如下图所示，其中master使用cluster IP访问pod的路径如红线所示：



Q7: 请采用声明式接口对Service进行部署，并将部署所需的yml文件记录在实践文档中

```
apiVersion: v1
kind: Service
metadata:
  name: hw-service
spec:
  selector:
    name: hw-pod
  ports:
    - name: service-port1
      protocol: TCP
      port: 80
      targetPort: 80
    - name: service-port2
      protocol: TCP
      port: 6800
      targetPort: 6800
    - name: service-port3
      protocol: TCP
      port: 6880
      targetPort: 6880
```

Q8: 请在master节点中使用 iptables-save 指令输出所有的iptables规则，将其中与Service访问相关的 iptable规则记录在实践文档中，并解释网络流量是如何采用基于iptables的方式被从对Service的cluster IP的访问定向到实际的Pod中的

与Service访问相关的 iptable规则记录：

主要是存在于filter表和nat表中的与`KUBE-SERVICES`(cluster ip类型的service)，`KUBE-NODEPORTS`(node port类型的service)，`KUBE-SVC-XXX`，`KUBE-SEP-XXX`相关的一些规则

```
*filter
-A INPUT -m comment --comment "kubernetes health check service ports" -j KUBE-
NODEPORTS
-A INPUT -m conntrack --ctstate NEW -m comment --comment "kubernetes externally-
visible service portals" -j KUBE-EXTERNAL-SERVICES
-A FORWARD -m conntrack --ctstate NEW -m comment --comment "kubernetes service
portals" -j KUBE-SERVICES
-A FORWARD -m conntrack --ctstate NEW -m comment --comment "kubernetes
externally-visible service portals" -j KUBE-EXTERNAL-SERVICES
-A OUTPUT -m conntrack --ctstate NEW -m comment --comment "kubernetes service
portals" -j KUBE-SERVICES

*nat
-A KUBE-SEP-3DU66DE6VORVEQVD -s 10.32.0.3/32 -m comment --comment "kube-
system/kube-dns:dns" -j KUBE-MARK-MASQ
-A KUBE-SEP-3DU66DE6VORVEQVD -p udp -m comment --comment "kube-system/kube-
dns:dns" -m udp -j DNAT --to-destination 10.32.0.3:53
-A KUBE-SEP-DBE6WNYPSNVGPT5K -s 10.32.0.2/32 -m comment --comment "default/hw-
service:service-port1" -j KUBE-MARK-MASQ
-A KUBE-SEP-DBE6WNYPSNVGPT5K -p tcp -m comment --comment "default/hw-
service:service-port1" -m tcp -j DNAT --to-destination 10.32.0.2:80
-A KUBE-SEP-LD2WAJ7CCWMNRHXI -s 10.32.0.2/32 -m comment --comment "default/hw-
service:service-port3" -j KUBE-MARK-MASQ
-A KUBE-SEP-LD2WAJ7CCWMNRHXI -p tcp -m comment --comment "default/hw-
service:service-port3" -m tcp -j DNAT --to-destination 10.32.0.2:6880
-A KUBE-SEP-S4MK5EVI7CLHCCS6 -s 10.32.0.3/32 -m comment --comment "kube-
system/kube-dns:dns-tcp" -j KUBE-MARK-MASQ
-A KUBE-SEP-S4MK5EVI7CLHCCS6 -p tcp -m comment --comment "kube-system/kube-
dns:dns-tcp" -m tcp -j DNAT --to-destination 10.32.0.3:53
-A KUBE-SEP-SSOXY7UWNUZWWPGH -s 192.168.1.7/32 -m comment --comment
"default/kubernetes:https" -j KUBE-MARK-MASQ
-A KUBE-SEP-SSOXY7UWNUZWWPGH -p tcp -m comment --comment
"default/kubernetes:https" -m tcp -j DNAT --to-destination 192.168.1.7:6443
-A KUBE-SEP-SWLOBIBPXYBP7G2Z -s 10.32.0.2/32 -m comment --comment "kube-
system/kube-dns:metrics" -j KUBE-MARK-MASQ
-A KUBE-SEP-SWLOBIBPXYBP7G2Z -p tcp -m comment --comment "kube-system/kube-
dns:metrics" -m tcp -j DNAT --to-destination 10.32.0.2:9153
-A KUBE-SEP-SZZ7MOWKTWUFXIJT -s 10.32.0.2/32 -m comment --comment "kube-
system/kube-dns:dns" -j KUBE-MARK-MASQ
-A KUBE-SEP-SZZ7MOWKTWUFXIJT -p udp -m comment --comment "kube-system/kube-
dns:dns" -m udp -j DNAT --to-destination 10.32.0.2:53
-A KUBE-SEP-UJJNLSZU6HL4F5UO -s 10.32.0.2/32 -m comment --comment "kube-
system/kube-dns:dns-tcp" -j KUBE-MARK-MASQ
-A KUBE-SEP-UJJNLSZU6HL4F5UO -p tcp -m comment --comment "kube-system/kube-
dns:dns-tcp" -m tcp -j DNAT --to-destination 10.32.0.2:53
```

```
-A KUBE-SEP-YNJHIKEFEQVKXLYN -s 10.32.0.2/32 -m comment --comment "default/hw-service:service-port2" -j KUBE-MARK-MASQ
-A KUBE-SEP-YNJHIKEFEQVKXLYN -p tcp -m comment --comment "default/hw-service:service-port2" -m tcp -j DNAT --to-destination 10.32.0.2:6800
-A KUBE-SEP-ZCHNBYOGFZRFKYMA -s 10.32.0.3/32 -m comment --comment "kube-system/kube-dns:metrics" -j KUBE-MARK-MASQ
-A KUBE-SEP-ZCHNBYOGFZRFKYMA -p tcp -m comment --comment "kube-system/kube-dns:metrics" -m tcp -j DNAT --to-destination 10.32.0.3:9153
-A KUBE-SERVICES -d 10.96.0.1/32 -p tcp -m comment --comment "default/kubernetes:https cluster IP" -m tcp --dport 443 -j KUBE-SVC-NPX46M4PTMTKRN6Y
-A KUBE-SERVICES -d 10.96.0.10/32 -p udp -m comment --comment "kube-system/kube-dns:dns cluster IP" -m udp --dport 53 -j KUBE-SVC-TCOU7JCQXEZGVUNU
-A KUBE-SERVICES -d 10.97.138.148/32 -p tcp -m comment --comment "default/hw-service:service-port3 cluster IP" -m tcp --dport 6880 -j KUBE-SVC-BW2GGV47LHXQWV6M
-A KUBE-SERVICES -d 10.96.0.10/32 -p tcp -m comment --comment "kube-system/kube-dns:dns-tcp cluster IP" -m tcp --dport 53 -j KUBE-SVC-ERIFXISQEP7F7OF4
-A KUBE-SERVICES -d 10.96.0.10/32 -p tcp -m comment --comment "kube-system/kube-dns:metrics cluster IP" -m tcp --dport 9153 -j KUBE-SVC-JD5MR3NA4I4DYORP
-A KUBE-SERVICES -d 10.97.138.148/32 -p tcp -m comment --comment "default/hw-service:service-port1 cluster IP" -m tcp --dport 80 -j KUBE-SVC-ENU43L6F5DHOZE5C
-A KUBE-SERVICES -d 10.97.138.148/32 -p tcp -m comment --comment "default/hw-service:service-port2 cluster IP" -m tcp --dport 6800 -j KUBE-SVC-T3JNK3PIHWND75ML
-A KUBE-SERVICES -m comment --comment "kubernetes service nodeports; NOTE: this must be the last rule in this chain" -m addrtype --dst-type LOCAL -j KUBE-NODEPORTS
-A KUBE-SVC-BW2GGV47LHXQWV6M ! -s 10.244.0.0/16 -d 10.97.138.148/32 -p tcp -m comment --comment "default/hw-service:service-port3 cluster IP" -m tcp --dport 6880 -j KUBE-MARK-MASQ
-A KUBE-SVC-BW2GGV47LHXQWV6M -m comment --comment "default/hw-service:service-port3" -j KUBE-SEP-LD2WAJ7CCWMNRHXI
-A KUBE-SVC-ENU43L6F5DHOZE5C ! -s 10.244.0.0/16 -d 10.97.138.148/32 -p tcp -m comment --comment "default/hw-service:service-port1 cluster IP" -m tcp --dport 80 -j KUBE-MARK-MASQ
-A KUBE-SVC-ENU43L6F5DHOZE5C -m comment --comment "default/hw-service:service-port1" -j KUBE-SEP-DBE6WNYPSNVGPT5K
-A KUBE-SVC-ERIFXISQEP7F7OF4 ! -s 10.244.0.0/16 -d 10.96.0.10/32 -p tcp -m comment --comment "kube-system/kube-dns:dns-tcp cluster IP" -m tcp --dport 53 -j KUBE-MARK-MASQ
-A KUBE-SVC-ERIFXISQEP7F7OF4 -m comment --comment "kube-system/kube-dns:dns-tcp" -m statistic --mode random --probability 0.50000000000 -j KUBE-SEP-UJJNLSZU6HL4F5UO
-A KUBE-SVC-ERIFXISQEP7F7OF4 -m comment --comment "kube-system/kube-dns:dns-tcp" -j KUBE-SEP-S4MK5EVI7CLHCCS6
-A KUBE-SVC-JD5MR3NA4I4DYORP ! -s 10.244.0.0/16 -d 10.96.0.10/32 -p tcp -m comment --comment "kube-system/kube-dns:metrics cluster IP" -m tcp --dport 9153 -j KUBE-MARK-MASQ
-A KUBE-SVC-JD5MR3NA4I4DYORP -m comment --comment "kube-system/kube-dns:metrics" -m statistic --mode random --probability 0.50000000000 -j KUBE-SEP-SWLOBIBPXYBP7G2Z
-A KUBE-SVC-JD5MR3NA4I4DYORP -m comment --comment "kube-system/kube-dns:metrics" -j KUBE-SEP-ZCHNBYOGFZRFKYMA
-A KUBE-SVC-NPX46M4PTMTKRN6Y ! -s 10.244.0.0/16 -d 10.96.0.1/32 -p tcp -m comment --comment "default/kubernetes:https cluster IP" -m tcp --dport 443 -j KUBE-MARK-MASQ
```

```
-A KUBE-SVC-NPX46M4PTMTKRN6Y -m comment --comment "default/kubernetes:https" -j
KUBE-SEP-SSOXY7UWNUZWWPGH
-A KUBE-SVC-T3JNK3PIHWND75ML ! -s 10.244.0.0/16 -d 10.97.138.148/32 -p tcp -m
comment --comment "default/hw-service:service-port2 cluster IP" -m tcp --dport
6800 -j KUBE-MARK-MASQ
-A KUBE-SVC-T3JNK3PIHWND75ML -m comment --comment "default/hw-service:service-
port2" -j KUBE-SEP-YNJHIKEFEQVKXLYN
-A KUBE-SVC-TCOU7JCQXEZGVUNU ! -s 10.244.0.0/16 -d 10.96.0.10/32 -p udp -m
comment --comment "kube-system/kube-dns:dns cluster IP" -m udp --dport 53 -j
KUBE-MARK-MASQ
-A KUBE-SVC-TCOU7JCQXEZGVUNU -m comment --comment "kube-system/kube-dns:dns" -m
statistic --mode random --probability 0.50000000000 -j KUBE-SEP-SZZ7MOWKTWUFXIJT
-A KUBE-SVC-TCOU7JCQXEZGVUNU -m comment --comment "kube-system/kube-dns:dns" -j
KUBE-SEP-3DU66DE6VORVEQVD
```

网络流量基于iptables对cluster IP访问定向到实际的Pod中的方式

在 `PREROUTING` 的 chain里将经过 `PREROUTING` 里的数据包重定向到 `KUBE-SERVICES` 中。在自定义链 `KUBE-SERVICES` 里找到dst为目标地址的ip，找到对应的 `KUBE-SVC-XXX` (ps:只有有对应endpoint信息的才有 `KUBE-SVC-XXX`)，并找到 `KUBE-SVC-XXX` 对应的 `KUBE-SEP-XXX` 。在endpoint target(`KUBE-SEP-XXX`) 里实现了 `DNAT` ，也就是将目标地址cluster ip转化为实际的pod的ip。

Q9: kube-proxy组件在整个service的定义与实现过程中起到了什么作用？请自行查找资料，并解释在 iptables模式下，kube-proxy的功能

Kube-proxy是一个简单的网络代理和负载均衡器，它的作用主要是负责Service的实现，具体来说，kube-proxy其实就是管理service的访问入口，包括集群内Pod到Service的访问和集群外访问service。kube-proxy管理sevice的Endpoints，该service对外暴露一个Virtual IP，也成为Cluster IP, 集群内通过访问这个Cluster IP:Port就能访问到集群内对应的serivce下的Pod。

在 iptables模式，每台机器上都运行一个 kube-proxy 服务，它监听 API server 中 service 和 endpoint 的变化情况，并通过 iptables 的nat转发来为service配置负载均衡。

Q10: 请采用声明式接口对Deployment进行部署，并将Deployment所需要的yml文件记录在文档中

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hw-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      name: hw-pod
  template:
    metadata:
      labels:
        name: hw-pod
    spec:
```

```
      containers:
        - name: viewer
          image: dplsming/nginx-fileserver:1.0
          command:
          args:
          workingDir:
          volumeMounts:
            - name: nginx-volume
              mountPath: /usr/share/nginx/html/files

          ports:
            - name: viewer-port
              containerPort: 80
              protocol: TCP

        - name: downloader
          image: dplsming/aria2ng-downloader:1.0
          command:
          args:
          workingDir:
          volumeMounts:
            - name: downloader-volume
              mountPath: /data

          ports:
            - name: down-port1
              containerPort: 6800
              protocol: TCP
            - name: down-port2
              containerPort: 6880
              protocol: TCP

      volumes:
        - name: nginx-volume
          hostPath:
            path: /hw
        - name: downloader-volume
          hostPath:
            path: /hw
```

Q11: 请在master节点中使用 iptables-save 指令输出所有的iptables规则，将其中与Service访问相关 的iptable规则记录在实践文档中，并解释网络流量是如何采用基于iptable的方式被从对Service的 cluster ip的访问，以随机且负载均衡的方式，定向到Deployment所创建的实际的Pod中的?

与Service访问相关的 iptable规则记录:

主要是存在于filter表和nat表中的与`KUBE-SERVICES`(cluster ip类型的service)，`KUBE-NODEPORTS`(node port类型的service)，`KUBE-SVC-XXX`，`KUBE-SEP-XXX`相关的一些规则

```
*filter
-A INPUT -m conntrack --ctstate NEW -m comment --comment "kubernetes externally-visible service portals" -j KUBE-EXTERNAL-SERVICES
-A FORWARD -m conntrack --ctstate NEW -m comment --comment "kubernetes service portals" -j KUBE-SERVICES
```

```
-A FORWARD -m conntrack --ctstate NEW -m comment --comment "kubernetes
externally-visible service portals" -j KUBE-EXTERNAL-SERVICES
-A OUTPUT -m conntrack --ctstate NEW -m comment --comment "kubernetes service
portals" -j KUBE-SERVICES

*nat
-A PREROUTING -m comment --comment "kubernetes service portals" -j KUBE-SERVICES
-A OUTPUT -m comment --comment "kubernetes service portals" -j KUBE-SERVICES
-A KUBE-SEP-3DU66DE6VORVEQVD -s 10.32.0.3/32 -m comment --comment "kube-
system/kube-dns:dns" -j KUBE-MARK-MASQ
-A KUBE-SEP-3DU66DE6VORVEQVD -p udp -m comment --comment "kube-system/kube-
dns:dns" -m udp -j DNAT --to-destination 10.32.0.3:53
-A KUBE-SEP-7ELNUO3E5TS2ZAJS -s 10.32.0.4/32 -m comment --comment "default/hw-
service:service-port3" -j KUBE-MARK-MASQ
-A KUBE-SEP-7ELNUO3E5TS2ZAJS -p tcp -m comment --comment "default/hw-
service:service-port3" -m tcp -j DNAT --to-destination 10.32.0.4:6880
-A KUBE-SEP-C2ZYYM2WITXS564E -s 10.32.0.4/32 -m comment --comment "default/hw-
service:service-port2" -j KUBE-MARK-MASQ
-A KUBE-SEP-C2ZYYM2WITXS564E -p tcp -m comment --comment "default/hw-
service:service-port2" -m tcp -j DNAT --to-destination 10.32.0.4:6800
-A KUBE-SEP-DBE6WNYPSNVGPT5K -s 10.32.0.2/32 -m comment --comment "default/hw-
service:service-port1" -j KUBE-MARK-MASQ
-A KUBE-SEP-DBE6WNYPSNVGPT5K -p tcp -m comment --comment "default/hw-
service:service-port1" -m tcp -j DNAT --to-destination 10.32.0.2:80
-A KUBE-SEP-I6DCNKFZ3NEMJCDK -s 10.32.0.3/32 -m comment --comment "default/hw-
service:service-port2" -j KUBE-MARK-MASQ
-A KUBE-SEP-I6DCNKFZ3NEMJCDK -p tcp -m comment --comment "default/hw-
service:service-port2" -m tcp -j DNAT --to-destination 10.32.0.3:6800
-A KUBE-SEP-LD2WAJ7CCWMNRHXI -s 10.32.0.2/32 -m comment --comment "default/hw-
service:service-port3" -j KUBE-MARK-MASQ
-A KUBE-SEP-LD2WAJ7CCWMNRHXI -p tcp -m comment --comment "default/hw-
service:service-port3" -m tcp -j DNAT --to-destination 10.32.0.2:6880
-A KUBE-SEP-Q7XPNW24XPUIDDXY -s 10.32.0.4/32 -m comment --comment "default/hw-
service:service-port1" -j KUBE-MARK-MASQ
-A KUBE-SEP-Q7XPNW24XPUIDDXY -p tcp -m comment --comment "default/hw-
service:service-port1" -m tcp -j DNAT --to-destination 10.32.0.4:80
-A KUBE-SEP-S4MK5EVI7CLHCCS6 -s 10.32.0.3/32 -m comment --comment "kube-
system/kube-dns:dns-tcp" -j KUBE-MARK-MASQ
-A KUBE-SEP-S4MK5EVI7CLHCCS6 -p tcp -m comment --comment "kube-system/kube-
dns:dns-tcp" -m tcp -j DNAT --to-destination 10.32.0.3:53
-A KUBE-SEP-SSOXY7UWNUZWWPGH -s 192.168.1.7/32 -m comment --comment
"default/kubernetes:https" -j KUBE-MARK-MASQ
-A KUBE-SEP-SSOXY7UWNUZWWPGH -p tcp -m comment --comment
"default/kubernetes:https" -m tcp -j DNAT --to-destination 192.168.1.7:6443
-A KUBE-SEP-SWLOBIBPXYBP7G2Z -s 10.32.0.2/32 -m comment --comment "kube-
system/kube-dns:metrics" -j KUBE-MARK-MASQ
-A KUBE-SEP-SWLOBIBPXYBP7G2Z -p tcp -m comment --comment "kube-system/kube-
dns:metrics" -m tcp -j DNAT --to-destination 10.32.0.2:9153
-A KUBE-SEP-SZZ7MOWKTWUFXIJT -s 10.32.0.2/32 -m comment --comment "kube-
system/kube-dns:dns" -j KUBE-MARK-MASQ
-A KUBE-SEP-SZZ7MOWKTWUFXIJT -p udp -m comment --comment "kube-system/kube-
dns:dns" -m udp -j DNAT --to-destination 10.32.0.2:53
-A KUBE-SEP-UJJNLSZU6HL4F5UO -s 10.32.0.2/32 -m comment --comment "kube-
system/kube-dns:dns-tcp" -j KUBE-MARK-MASQ
-A KUBE-SEP-UJJNLSZU6HL4F5UO -p tcp -m comment --comment "kube-system/kube-
dns:dns-tcp" -m tcp -j DNAT --to-destination 10.32.0.2:53
-A KUBE-SEP-XL2NAUUCNIGJ67EE -s 10.32.0.3/32 -m comment --comment "default/hw-
service:service-port1" -j KUBE-MARK-MASQ
```

```
-A KUBE-SEP-XL2NAUUCNIGJ67EE -p tcp -m comment --comment "default/hw-
service:service-port1" -m tcp -j DNAT --to-destination 10.32.0.3:80
-A KUBE-SEP-YHFAT6YJ6ANM6C6K -s 10.32.0.3/32 -m comment --comment "default/hw-
service:service-port3" -j KUBE-MARK-MASQ
-A KUBE-SEP-YHFAT6YJ6ANM6C6K -p tcp -m comment --comment "default/hw-
service:service-port3" -m tcp -j DNAT --to-destination 10.32.0.3:6880
-A KUBE-SEP-YNJHIKEFEQVKXLYN -s 10.32.0.2/32 -m comment --comment "default/hw-
service:service-port2" -j KUBE-MARK-MASQ
-A KUBE-SEP-YNJHIKEFEQVKXLYN -p tcp -m comment --comment "default/hw-
service:service-port2" -m tcp -j DNAT --to-destination 10.32.0.2:6800
-A KUBE-SEP-ZCHNBYOGFZRFKYMA -s 10.32.0.3/32 -m comment --comment "kube-
system/kube-dns:metrics" -j KUBE-MARK-MASQ
-A KUBE-SEP-ZCHNBYOGFZRFKYMA -p tcp -m comment --comment "kube-system/kube-
dns:metrics" -m tcp -j DNAT --to-destination 10.32.0.3:9153
-A KUBE-SERVICES -d 10.96.0.10/32 -p tcp -m comment --comment "kube-system/kube-
dns:dns-tcp cluster IP" -m tcp --dport 53 -j KUBE-SVC-ERIFXISQEP7F7OF4
-A KUBE-SERVICES -d 10.96.0.10/32 -p tcp -m comment --comment "kube-system/kube-
dns:metrics cluster IP" -m tcp --dport 9153 -j KUBE-SVC-JD5MR3NA4I4DYORP
-A KUBE-SERVICES -d 10.107.25.193/32 -p tcp -m comment --comment "default/hw-
service:service-port1 cluster IP" -m tcp --dport 80 -j KUBE-SVC-ENU43L6F5DHOZE5C
-A KUBE-SERVICES -d 10.107.25.193/32 -p tcp -m comment --comment "default/hw-
service:service-port2 cluster IP" -m tcp --dport 6800 -j KUBE-SVC-
T3JNK3PIHWND75ML
-A KUBE-SERVICES -d 10.107.25.193/32 -p tcp -m comment --comment "default/hw-
service:service-port3 cluster IP" -m tcp --dport 6880 -j KUBE-SVC-
BW2GGV47LHXQWV6M
-A KUBE-SERVICES -d 10.96.0.1/32 -p tcp -m comment --comment
"default/kubernetes:https cluster IP" -m tcp --dport 443 -j KUBE-SVC-
NPX46M4PTMTKRN6Y
-A KUBE-SERVICES -d 10.96.0.10/32 -p udp -m comment --comment "kube-system/kube-
dns:dns cluster IP" -m udp --dport 53 -j KUBE-SVC-TCOU7JCQXEZGVUNU
-A KUBE-SERVICES -m comment --comment "kubernetes service nodeports; NOTE: this
must be the last rule in this chain" -m addrtype --dst-type LOCAL -j KUBE-
NODEPORTS
-A KUBE-SVC-BW2GGV47LHXQWV6M ! -s 10.244.0.0/16 -d 10.107.25.193/32 -p tcp -m
comment --comment "default/hw-service:service-port3 cluster IP" -m tcp --dport
6880 -j KUBE-MARK-MASQ
-A KUBE-SVC-BW2GGV47LHXQWV6M -m comment --comment "default/hw-service:service-
port3" -m statistic --mode random --probability 0.33333333349 -j KUBE-SEP-
LD2WAJ7CCWMNRHXI
-A KUBE-SVC-BW2GGV47LHXQWV6M -m comment --comment "default/hw-service:service-
port3" -m statistic --mode random --probability 0.50000000000 -j KUBE-SEP-
YHFAT6YJ6ANM6C6K
-A KUBE-SVC-BW2GGV47LHXQWV6M -m comment --comment "default/hw-service:service-
port3" -j KUBE-SEP-7ELNUO3E5TS2ZAJS
-A KUBE-SVC-ENU43L6F5DHOZE5C ! -s 10.244.0.0/16 -d 10.107.25.193/32 -p tcp -m
comment --comment "default/hw-service:service-port1 cluster IP" -m tcp --dport 80
-j KUBE-MARK-MASQ
-A KUBE-SVC-ENU43L6F5DHOZE5C -m comment --comment "default/hw-service:service-
port1" -m statistic --mode random --probability 0.33333333349 -j KUBE-SEP-
DBE6WNYPSNVGPT5K
-A KUBE-SVC-ENU43L6F5DHOZE5C -m comment --comment "default/hw-service:service-
port1" -m statistic --mode random --probability 0.50000000000 -j KUBE-SEP-
XL2NAUUCNIGJ67EE
-A KUBE-SVC-ENU43L6F5DHOZE5C -m comment --comment "default/hw-service:service-
port1" -j KUBE-SEP-Q7XPNW24XPUIDDXY
```

```
-A KUBE-SVC-ERIFXISQEP7F7OF4 ! -s 10.244.0.0/16 -d 10.96.0.10/32 -p tcp -m
comment --comment "kube-system/kube-dns:dns-tcp cluster IP" -m tcp --dport 53 -j
KUBE-MARK-MASQ
-A KUBE-SVC-ERIFXISQEP7F7OF4 -m comment --comment "kube-system/kube-dns:dns-tcp"
-m statistic --mode random --probability 0.50000000000 -j KUBE-SEP-
UJJNLSZU6HL4F5UO
-A KUBE-SVC-ERIFXISQEP7F7OF4 -m comment --comment "kube-system/kube-dns:dns-tcp"
-j KUBE-SEP-S4MK5EVI7CLHCCS6
-A KUBE-SVC-JD5MR3NA4I4DYORP ! -s 10.244.0.0/16 -d 10.96.0.10/32 -p tcp -m
comment --comment "kube-system/kube-dns:metrics cluster IP" -m tcp --dport 9153 -
j KUBE-MARK-MASQ
-A KUBE-SVC-JD5MR3NA4I4DYORP -m comment --comment "kube-system/kube-dns:metrics"
-m statistic --mode random --probability 0.50000000000 -j KUBE-SEP-
SWLOBIBPXYBP7G2Z
-A KUBE-SVC-JD5MR3NA4I4DYORP -m comment --comment "kube-system/kube-dns:metrics"
-j KUBE-SEP-ZCHNBYOGFZRFKYMA
-A KUBE-SVC-NPX46M4PTMTKRN6Y ! -s 10.244.0.0/16 -d 10.96.0.1/32 -p tcp -m comment
--comment "default/kubernetes:https cluster IP" -m tcp --dport 443 -j KUBE-MARK-
MASQ
-A KUBE-SVC-NPX46M4PTMTKRN6Y -m comment --comment "default/kubernetes:https" -j
KUBE-SEP-SSOXY7UWNUZWWPGH
-A KUBE-SVC-T3JNK3PIHWND75ML ! -s 10.244.0.0/16 -d 10.107.25.193/32 -p tcp -m
comment --comment "default/hw-service:service-port2 cluster IP" -m tcp --dport
6800 -j KUBE-MARK-MASQ
-A KUBE-SVC-T3JNK3PIHWND75ML -m comment --comment "default/hw-service:service-
port2" -m statistic --mode random --probability 0.33333333349 -j KUBE-SEP-
YNJHIKEFEQVKXLYN
-A KUBE-SVC-T3JNK3PIHWND75ML -m comment --comment "default/hw-service:service-
port2" -m statistic --mode random --probability 0.50000000000 -j KUBE-SEP-
I6DCNKFZ3NEMJCDK
-A KUBE-SVC-T3JNK3PIHWND75ML -m comment --comment "default/hw-service:service-
port2" -j KUBE-SEP-C2ZYYM2WITXS564E
-A KUBE-SVC-TCOU7JCQXEZGVUNU ! -s 10.244.0.0/16 -d 10.96.0.10/32 -p udp -m
comment --comment "kube-system/kube-dns:dns cluster IP" -m udp --dport 53 -j
KUBE-MARK-MASQ
-A KUBE-SVC-TCOU7JCQXEZGVUNU -m comment --comment "kube-system/kube-dns:dns" -m
statistic --mode random --probability 0.50000000000 -j KUBE-SEP-SZZ7MOWKTWUFXIJT
-A KUBE-SVC-TCOU7JCQXEZGVUNU -m comment --comment "kube-system/kube-dns:dns" -j
KUBE-SEP-3DU66DE6VORVEQVD
```

网络流量被iptable定向的方式：

在 `PREROUTING` 的 chain里将经过 `PREROUTING` 里的数据包重定向到 `KUBE-SERVICES` 中。在自定义链 `KUBE-SERVICES` 里找到dst为目标地址的ip，找到对应的 `KUBE-SVC-XXX` (ps:只有有对应endpoint信息的才有 `KUBE-SVC-XXX`)。

而在 `KUBE-SVC-XXX` 对应的 `KUBE-SEP-XXX` 多个条目中，会随机跳转到其中一个条目中，在这里实现 `DNAT` ，即将目标地址cluster ip转化为实际的pod的ip。

Q12: 在该使用Deployment的部署方式下，不同Pod之间的文件是否共享？该情况会在实际使用文件下载与共享服务时产生怎样的实际效果与问题？应如何解决这一问题？

如果pod处在不同的物理机下，则他们之间的文件不能共享。

这可能会导致这样的情况：用户明明已将文件上传到共享服务中，但该文件存放在node1中，用户想要下载文件时，访问到的却是node2里的pod，导致原本上传好的文件无法被正常下载。

解决方式：再分布于不同node的pod之间开启实时增量同步，确保他们的文件状态保持一致。

519021910594

陶昱丞