Kubernetes实践

本实践旨在通过部署一个简单的Kubernetes集群，并在其中对一系列容器进行编排，从而加深对Kubernetes内各种概念和实现的理解。请利用JCloud平台所提供的计算资源创建2台可以互相联通的，运行有JCloud提供的Ubuntu20.04镜像的虚拟机，之后在该2台机器所组成的集群中完成下列操作，并回答对应问题。注意，由于许多问题和操作过程联系紧密，在进行每一个小节时，请先阅读完整个操作过程和相应问题后，再开始实际操作

使用kubeadm安装并部署k8s集群

kubeadm是k8s所提供的方便使用者在集群上快速部署Kubernetes的工具，请参照其官方教程（https://kubernetes.io/zh/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/），利用kubeadm在JCloud提供的2台机器组成的集群中部署k8s，使得部署完成后的两台机器中一台机器为master节点，另一台机器为普通节点（后文称之为worker节点）。

在部署过程中，请注意以下事项：

- 请采用weave作为Pod网络插件，安装方式可使用如下命令：

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

- 安装过程中若出现由于各种因素导致的网络过慢，可采用阿里云提供的国内镜像进行加速。其中，安装k8s相关组件时可采用：将apt-key地址替换为 https://mirrors.aliyun.com/kubernetes/apt/doc/apt-key.gpg，将apt源地址设置为 https://mirrors.aliyun.com/kubernetes/apt/ ；使用kubeadm init初始化容器是可添加如下参数`--image-repository='registry.cn-hangzhou.aliyuncs.com/google_containers'`来应用镜像地址。
- 确认kube-proxy采用iptables模式运转（若未手动安装ipvs相关组件，此项应为默认配置，无需主动修改）

部署完成后，请使用kubectl get nodes在制定为master的节点上打印部署结果，打印的结果应如下：

```
NAME       STATUS   ROLES                  AGE   VERSION
k8s-hw-1   Ready    control-plane,master   17h   v1.23.6
k8s-hw-2   Ready    <none>                 16h   v1.23.6
```

请基于部署过程，回答下列问题：

Q1: 请记录所有安装步骤的指令，并简要描述其含义

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

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，以及那些k8s组件未运行在容器中

部署Pod

小明希望在刚刚部署的k8s集群中部署一个简单的文件下载与共享服务，涉及到的容器镜像、使用到的端口和主要目录如下表：

容器镜像	简要描述	主要端口	主要目录
dplsming/nginx-fileserver:1.0	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	80：http服务器访问端口	/usr/share/nginx/html/files：nginx向外共享的根目录
dplsming/aria2ng-downloader:1.0	文件下载服务。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	6800：aria2c的rpc控制端口，6880：webUI访问端口	/data：默认的下载文件存储目录

请将上述两个镜像部署到同一个Pod中，暴露nginx-fileserver镜像所对应容器（后简称viewer）的80端口和aria2ng-downloader镜像所对应容器（后简称downloader）的6800、6880端口，并使得viewer的目录`/usr/share/nginx/html/files`和downloader的目录`/data`下的所有文件均为两个容器所共享的。基于部署Pod的过程，回答下述问题：

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

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

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

部署Service

为了方便访问Pod，小明决定在k8s集群中部署Service，用以对外界提供访问刚刚部署好的负责文件下载与共享服务的Pod的方法。请为上述Pod部署clusterIP类型的service，暴露80、6800、6880端口。基于Service的部署过程，回答下述问题：

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

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

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

使用Deployment为Pod创建备份

为了提高Pod的可用性，小明决定采用deployment的功能，在worker节点上，为文件下载与共享服务创建备份。请将前文所述Pod修改为采用Deployment的方式进行部署，使得deployment内部具有3份不同的备份。完成后，请将前文所述的service修改为closureIP类型，并采用该Deployment所提供的所有Pod随机的提供服务。基于部署过程，回答下述问题：

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

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

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？