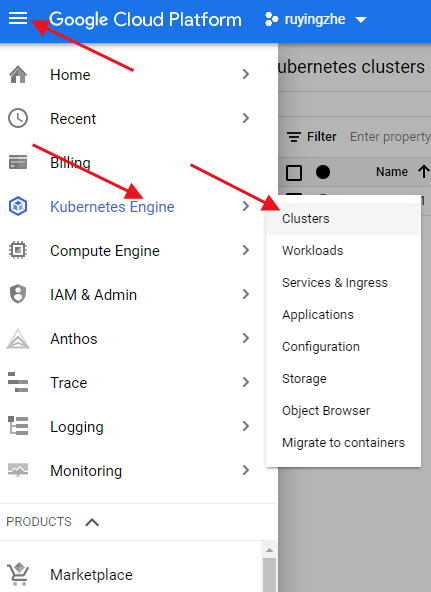
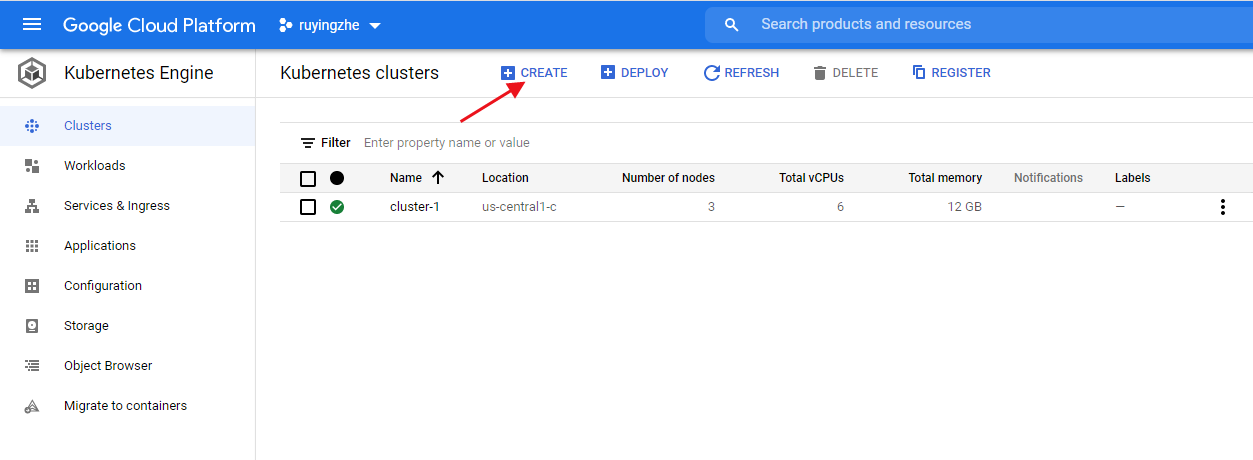
在GKE公有云建立集群加入TKE

20210629

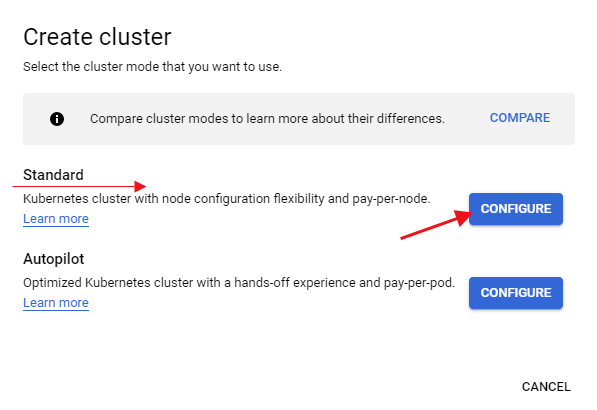
1. 在google cloud上创建集群

（1.1）登陆你的google账户，然后点击google cloud platform 🡪 kubernetes engine 🡪 cluster，然后得到第二张图，点击create

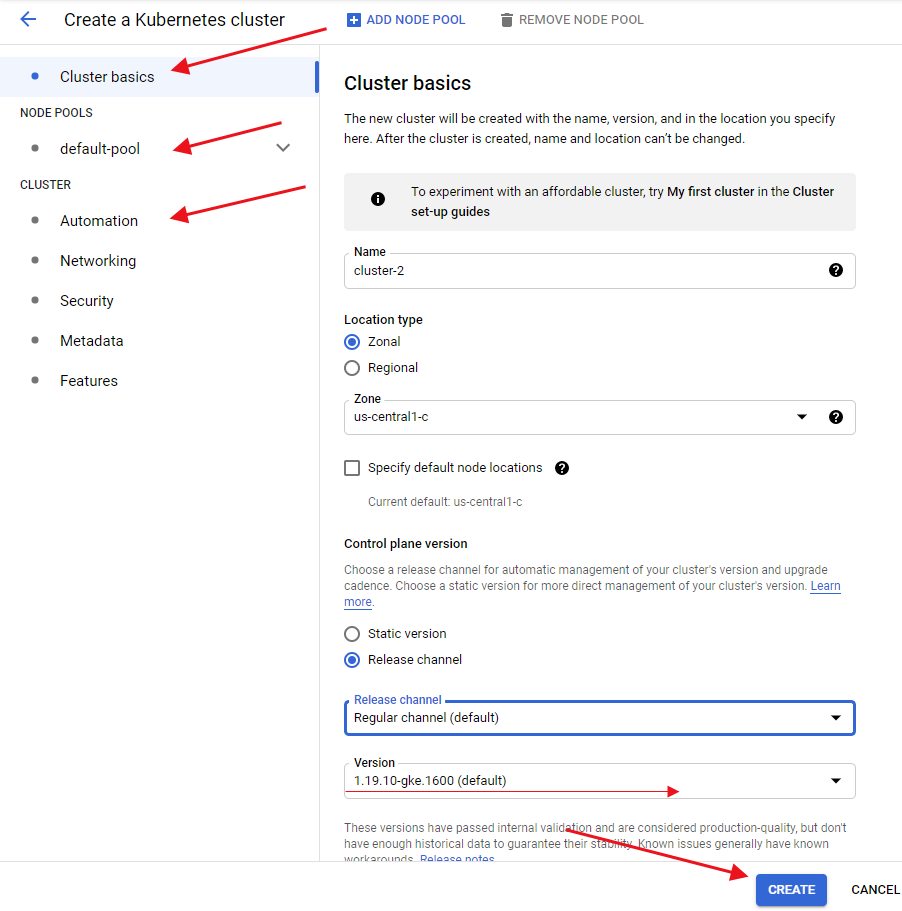




选择standrd，点击configure

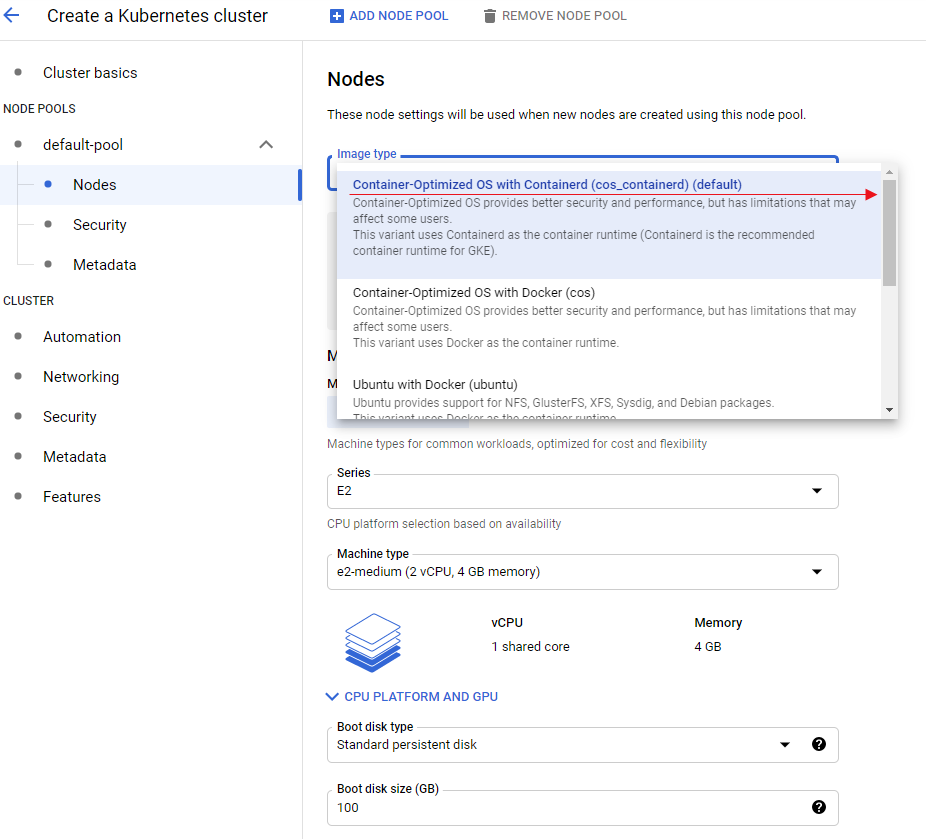


这里有一些选项可以修改，只不过我都是默认，把cluster name和kubernetes的version复制出来，等下用得上，配置好之后点击create就可以了



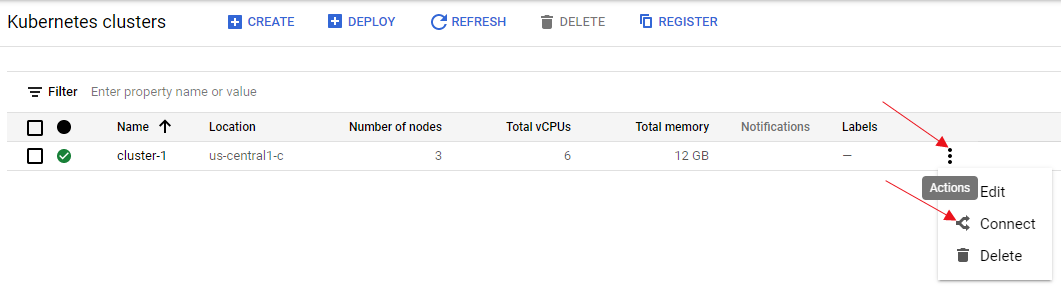
例如这里的操作系统我选择的是默认的container optimized OS（它的底层还是Linux），google利用自家开源Chromium OS作为基础，推出了一套为执行容器应用程序所生的操作系统Container-VM Image，该操作系统正式在GCP上线，并且改名为Container-Optimized OS，许多Google的服务都采用该操作系统，例如Google Cloud SQL、Google容器引擎。

关于container optimized OS，参考链接https://cloud.google.com/container-optimized-os/docs/concepts/security#filesystem

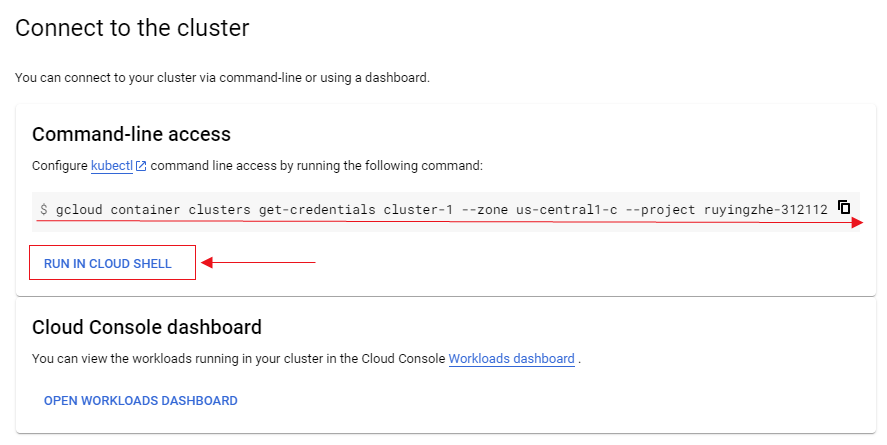


1. 搭建kubernetes集群

（2.1）等待几分钟，集群就创建好了，然后依次点击，找到connect



点击connect之后就会得到这个，可以自己执行画横线的命令或者点击“run in cloud shell”按钮



（2.2）点击“run in cloud shell”之后，会在当前页面的下半部分生成一个命令行窗口，窗口里已经帮你把需要的命令输入好了，直接回车执行就可以了，然后页面还可能弹出一个窗口让你授权登录，点击授权就可以了。如果觉得当前的视图太小了，可以点击web review按钮，在新标签页中打开命令行窗口（图片太长了，我截成了两段）



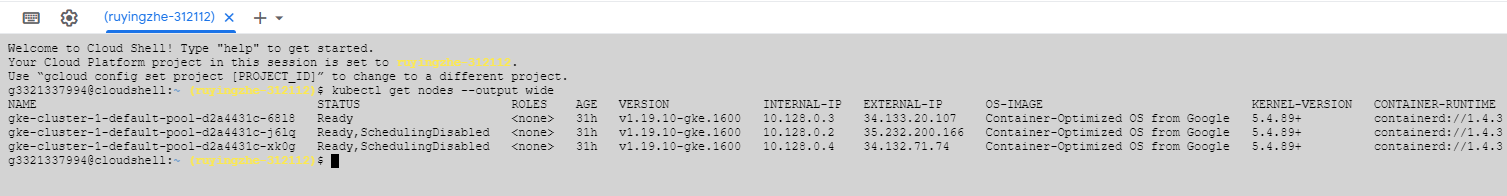


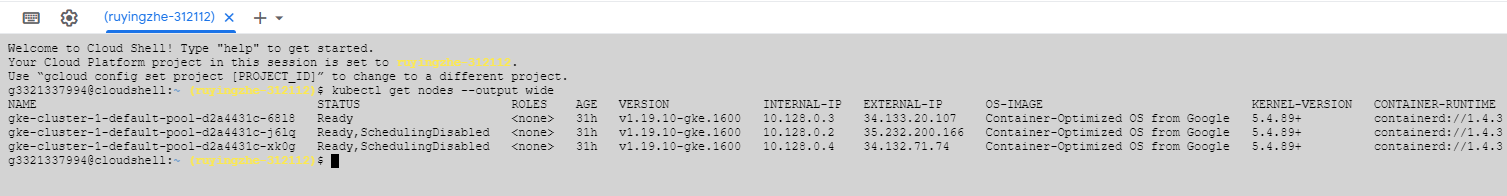
（截成两段前的原图）



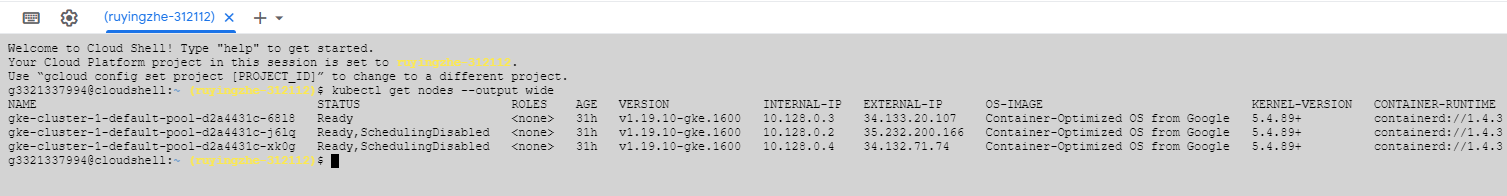
获取node节点的公网ip

g3321337994@cloudshell:~ (ruyingzhe-312112)$ kubectl get nodes --output wide





（截成两段前的图）



（2.3）# 把其他node隔离，在（2.2）中的第四张图可以看到效果

kubectl cordon gke-cluster-1-default-pool-d2a4431c-j6lq

kubectl cordon gke-cluster-1-default-pool-d2a4431c-xk0g

（2.4）下载broker-info.subm文件和subctl文件

# 因为没法从linux主机传文件给LB或者cluster-c中的node节点，

# 所以在LB或者cluster-c中的node节点，使用scp从指定的linux主机下载文件（需要输入linux主机的密码）

scp root@43.131.68.117:/root/broker-info.subm .

scp root@43.131.68.117:/root/.local/bin/subctl /tmp

（2.5）# 把其中一个节点标记为submariner的网关未来部署的节点

kubectl label node gke-cluster-1-default-pool-d2a4431c-68l8 submariner.io/gateway=true

（2.6）# GKE的集群以cluster-c的名字加入broker

./subctl join /tmp/broker-info.subm --clusterid cluster-c --natt=true

（2.7）# cluster-c创建nginx

# 先创建一个deployment对象，对象的名字叫做nginx，然后这个deployment对象运行nginx镜像

kubectl create deployment nginx --image=nginx

# 为deployment对象的nginx创建service，并通过Service的80端口转发至容器的80端口上。

kubectl expose deployment nginx --port=80

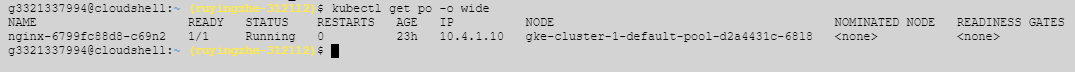
#使用subctl创建ServiceExport对象，使service可见，使得submariner中的其他集群可以发现这个service（执行这条指令可能会卡住大概10秒）

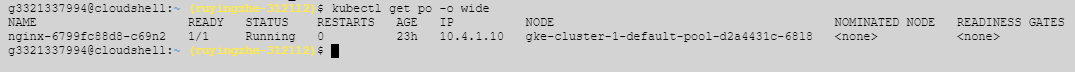
subctl export service --namespace default nginx

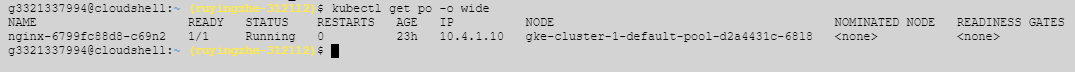
# 获取当前运行的namespace所有pods的信息（比如获取nginx的IP），包括pod运行在哪个节点上

kubectl get po -o wide

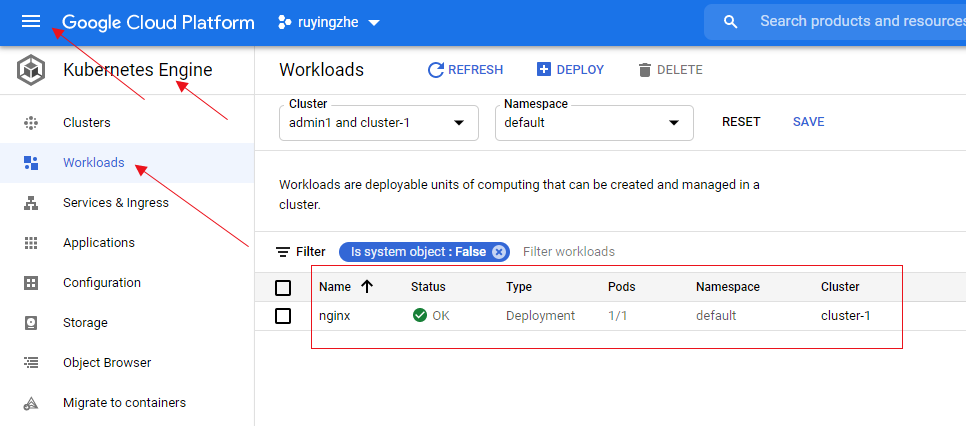
可以看到，nginx已经部署成功了







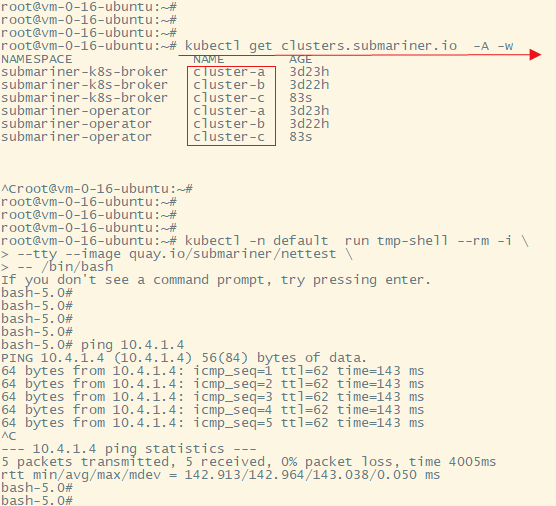
或者在网页的控制面板也可以看见nginx这个pod



三、回到cluster-a，验证submariner能否正常工作

（3.1）# 获取当前运行的namespace所有pods的信息（比如获取nginx的IP），包括pod运行在哪个节点上

kubectl get po -o wide



（3.2）# 如果你的cluster-a没安装tmp-shell（执行这条指令可能会卡住大概10秒）

kubectl -n default run tmp-shell --rm -i \

--tty --image quay.io/submariner/nettest \

-- /bin/bash

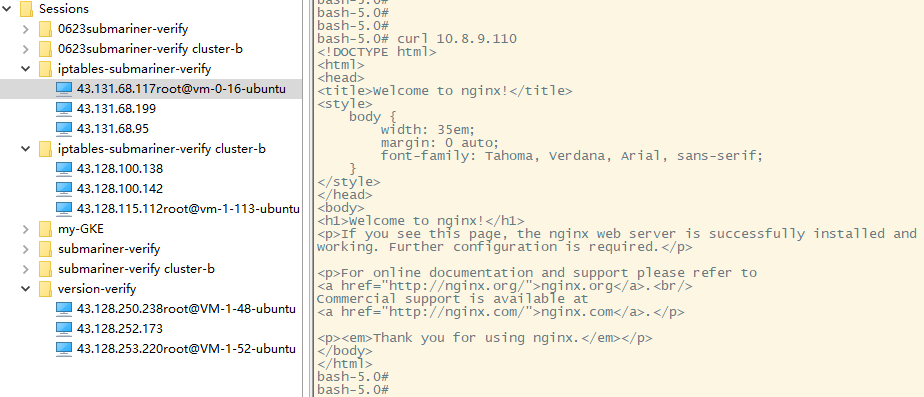
（3.3）# 在cluster-a上检查能否连接上cluster-b的nginx这个pod

ping <cluster-c显示的nginx的IP>

curl nginx.default.svc.clusterset.local

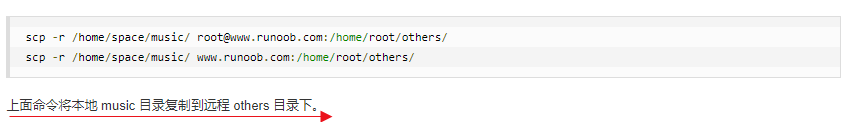
或者

curl <cluster-c显示的nginx的IP>



四、可能遇到的问题

（4.1）在本地的Linux主机连接GKE的cluster的LB节点



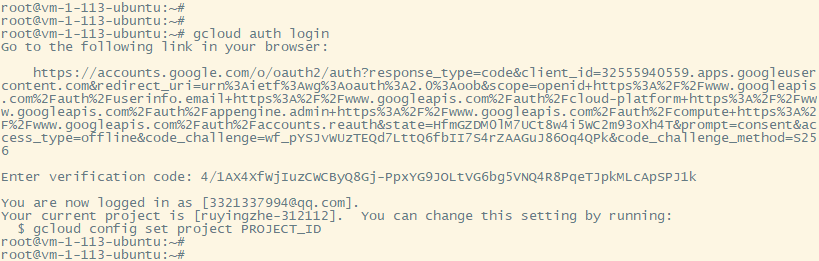
（4.1.1）# 下载gcloud

https://cloud.google.com/anthos/clusters/docs/on-prem/1.7/how-to/ssh-cluster-node

（4.1.2）# 安装之后需要登录认证（很多家的云产品都有类似的功能，比如腾讯云会让你扫描腾讯云助手的微信小程序，获取动态验证码验证登录）

gcloud auth login

如下图所示，terminal窗口会提示你把链接复制到浏览器打开，然后你会在浏览器得到一串验证码，复制会terminal窗口就可以了



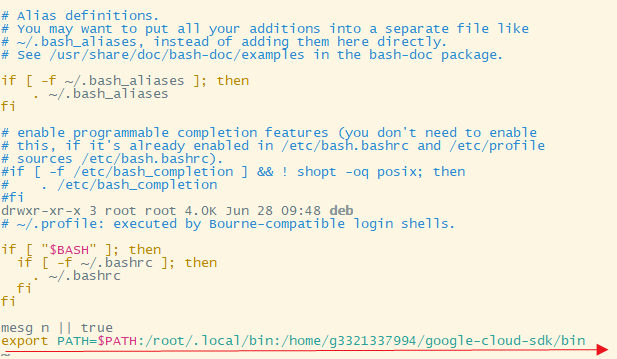
（4.1.3）把gcloud加入path路径

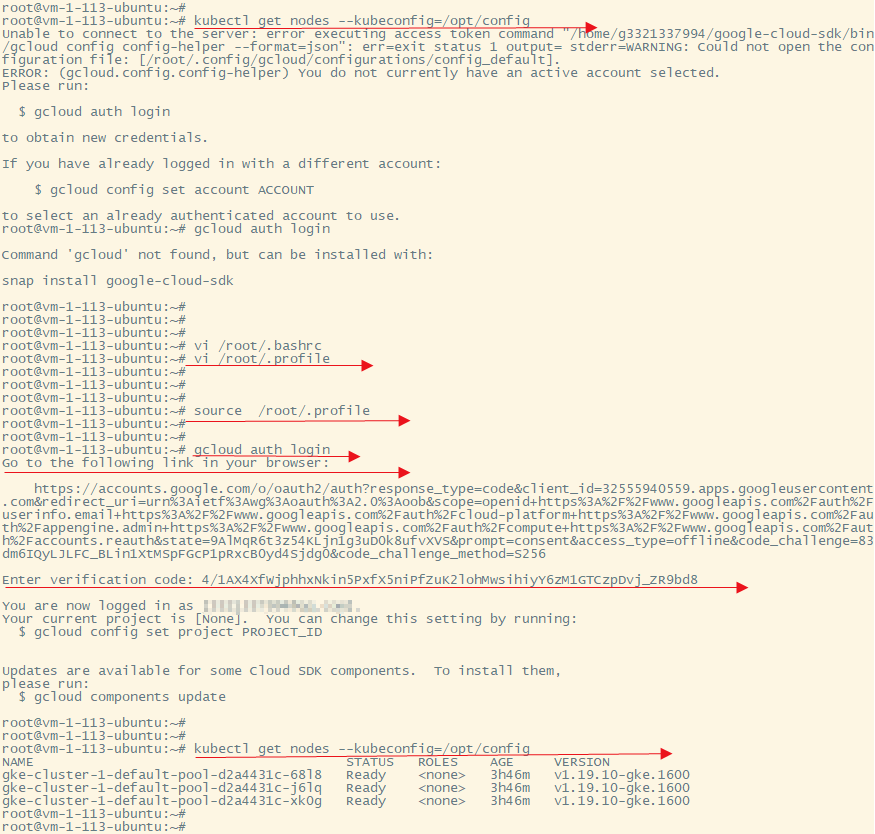
# 修改.profile文件

vi /root/.profile

# 刷新.profile文件

source /root/.profile





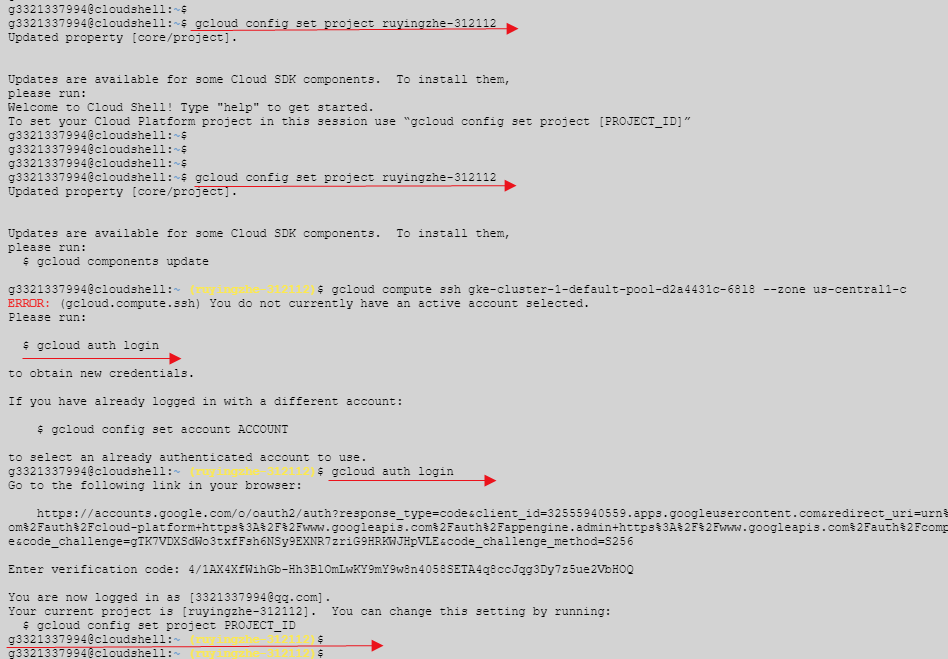
（4.2）在网页命令行窗口连接指定的node节点

（4.2.1）# 配置好PROJECT\_ID

gcloud config set project <PROJECT\_ID>

gcloud config set project ruyingzhe-312112

如果你之前用gcloud登陆过，那么你就可以直接连接集群（其实连接的是LB）；如果你之前没登陆过，那么你需要像（4.1.2）一样登录。登陆成功后ternimal的前缀从灰色变成了黄色



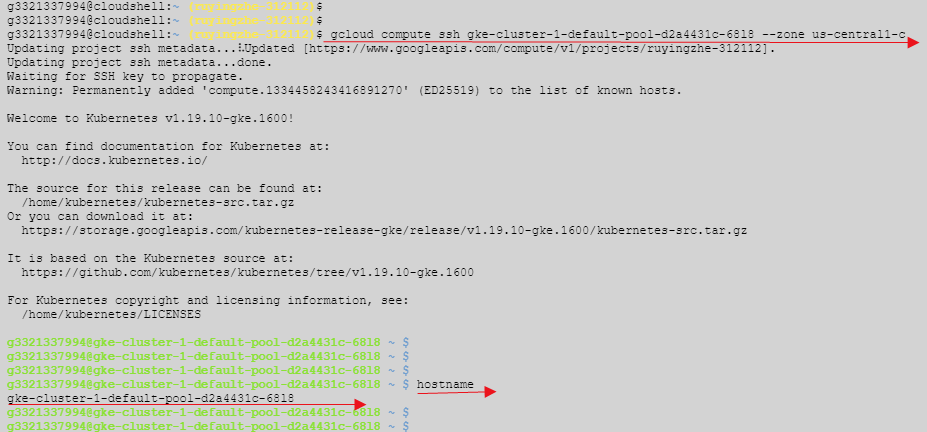
（4.2.2）登录LB之后，可以继续登录指定的node

# gcloud命令连接指定的node节点

gcloud compute ssh <NODE\_NAME> --zone <ZONE>

gcloud compute ssh gke-cluster-1-default-pool-d2a4431c-68l8 --zone us-central1-c

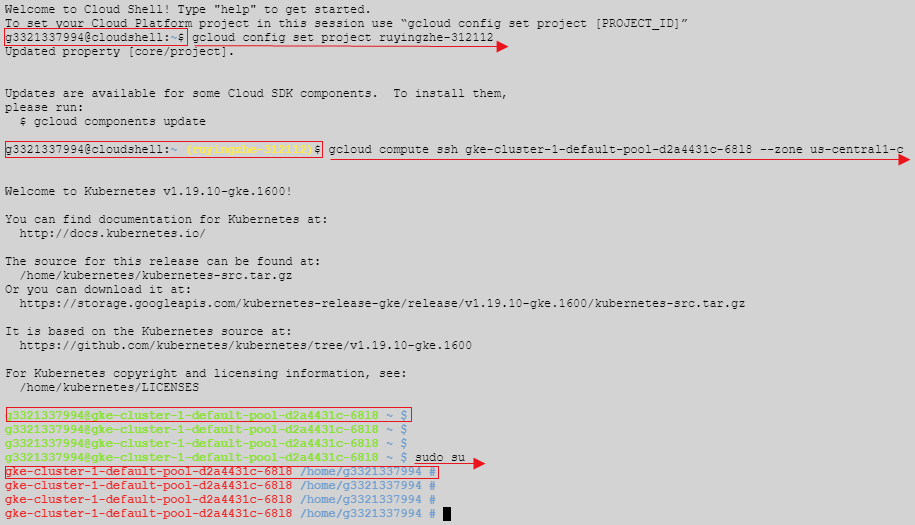
登录成功后ternimal的前缀从黄色变成了绿色，执行命令host，可以看到我们确实进入了指定的node



# 绿色的前缀只是表示你已经进入了node，如果需要获取node的root权限，还需要执行命令

sudo su

这时ternimal的前缀从绿色变成了红色，可以执行命令whoami查看当前的用户组





（4.3）container optimized OS的文件系统权限列表

获取node节点的root权限之后，需要下载文件，执行命令等，需要了解container optimized OS的文件系统（这是谷歌定制的操作系统，基于chromium os，chromium中文译为铬），参见网页

https://cloud.google.com/container-optimized-os/docs/concepts/security#filesystem



也就是说，即使你获取了root权限，在安装程序，执行程序时也需要在具有对应权限的的路径文件，否则程序是无法安装或者执行的

（4.4）执行kubectl命令时可能会出现报错，没有权限执行kubectl，切换到LB的命令行窗口执行该语句

The connection to the server localhost:8080 was refused - did you specify the right host or port?

红色的框框似乎表明，GKE默认支持的是iptables，结合网页文档https://cloud.google.com/kubernetes-engine/docs/concepts/network-overview

