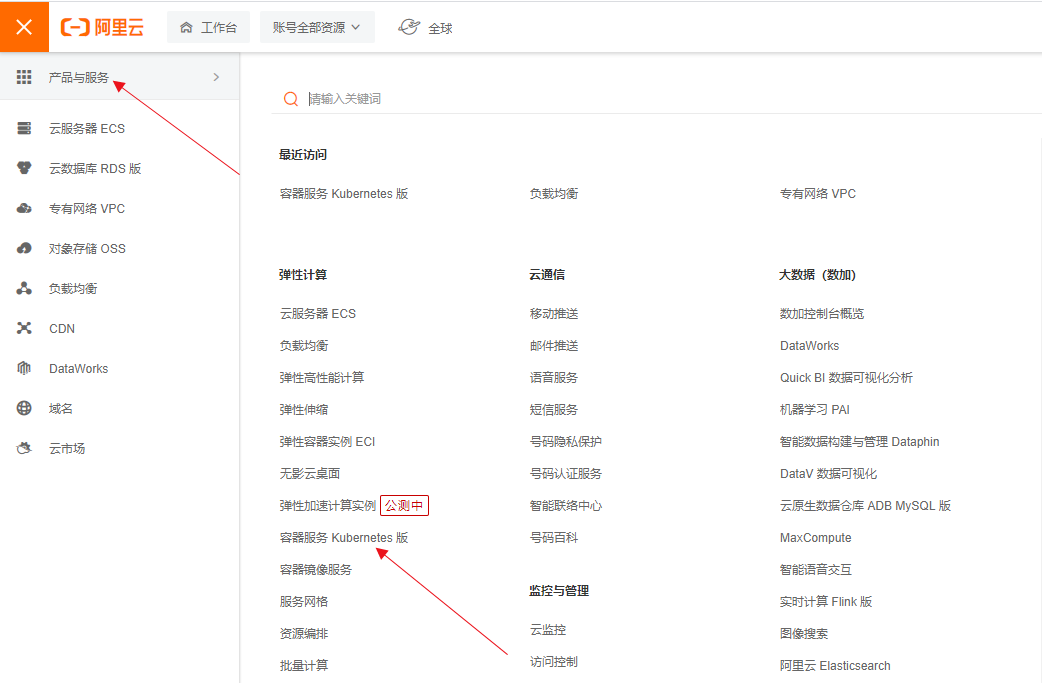
在AKS公有云建立集群加入TKE

20210629

一、在阿里云上创建集群

（1.1）登录阿里云账户，先点击页面（第一张图）的左上角的三条横线，然后在弹出的页面中（第二张图）选择产品与服务 🡪 容器服务kubernetes





（1.2）然后点击创建集群（图片太长，我截成了两段）





（截断之前的图）



（1.3）选择ACK专有版，集群的区域最好选择境外的服务器



网络插件选择flannel，并且记下pod CIDR和service CIDR



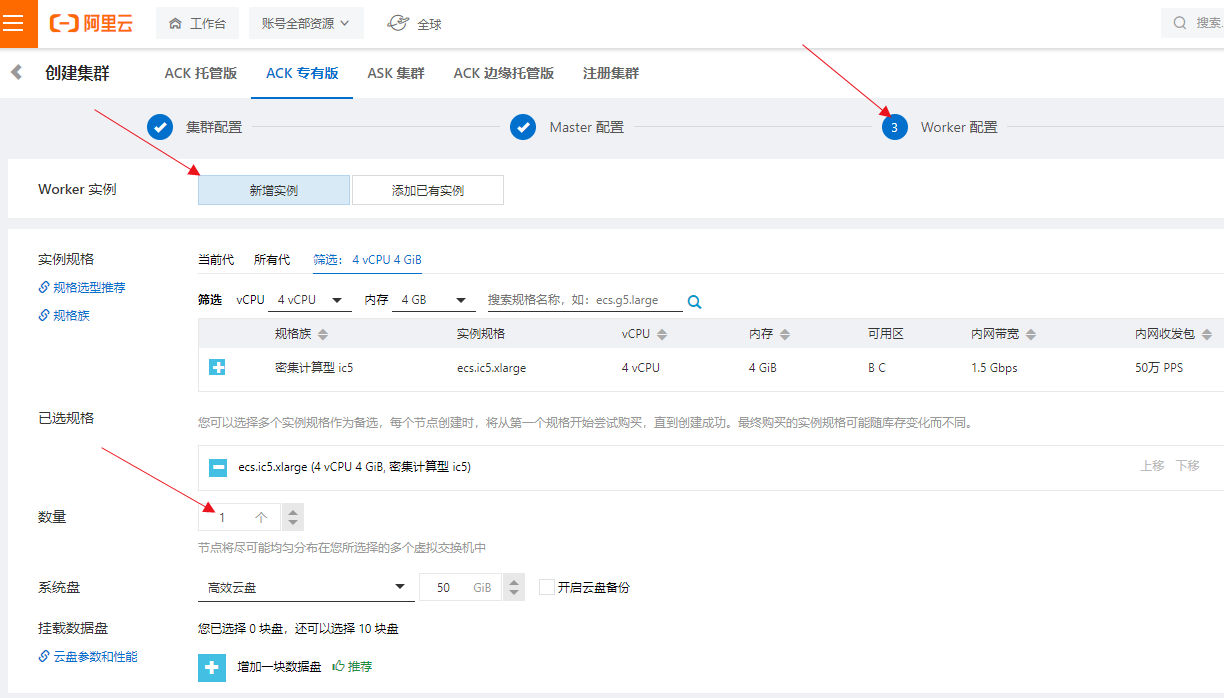
开启各种公网访问



在“高级”选项中选择iptables（默认是ipvs，所以要记得改）



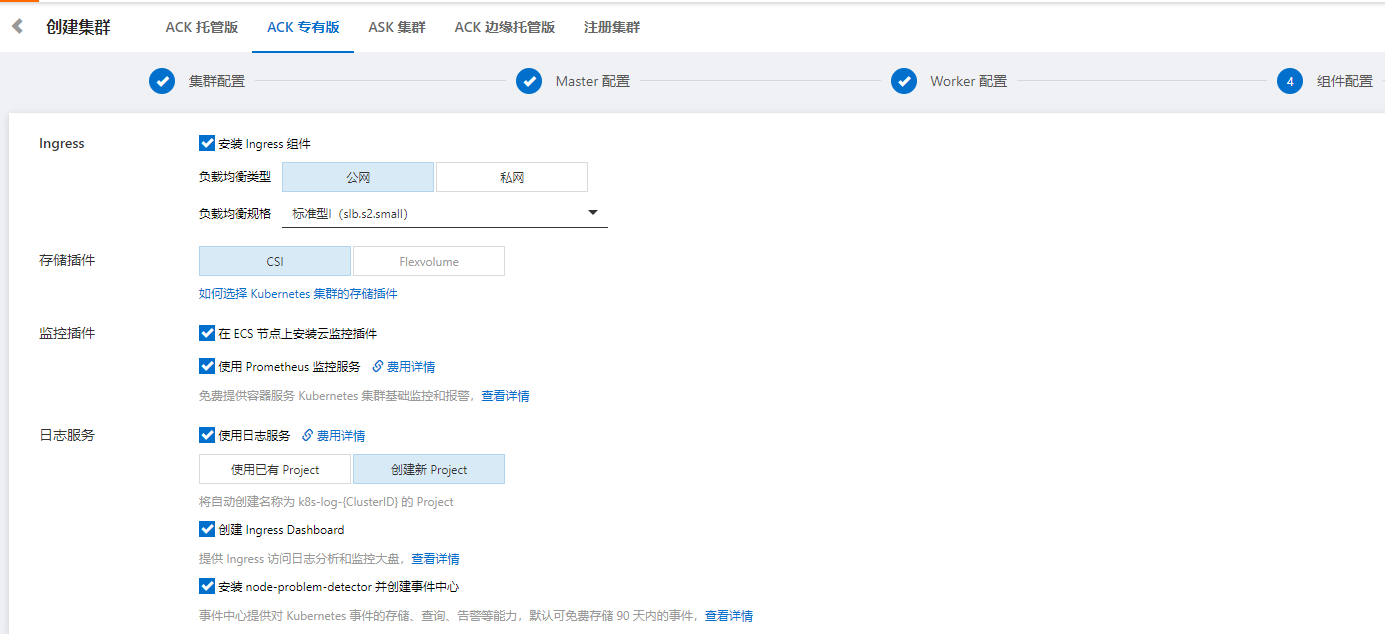
（1.4）配置worker节点，这里worker节点的数目可以为0（我在图片里是1，其实这一个也可以不用）



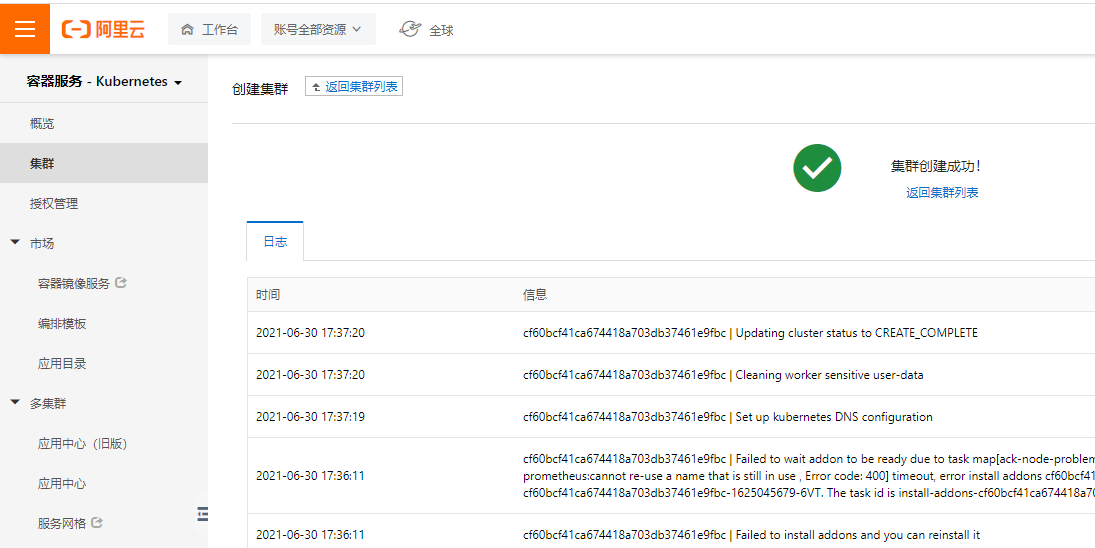
给worker节点配置账号密码（如果没有worker节点的话自然也就不用配置账号密码了）



（1.5）组件配置



所有的都没问题后，点击确认，创建集群，等待大约15分钟就可以了



回到集群界面发现集群已经创建好了（图片太长，我截成了两段）

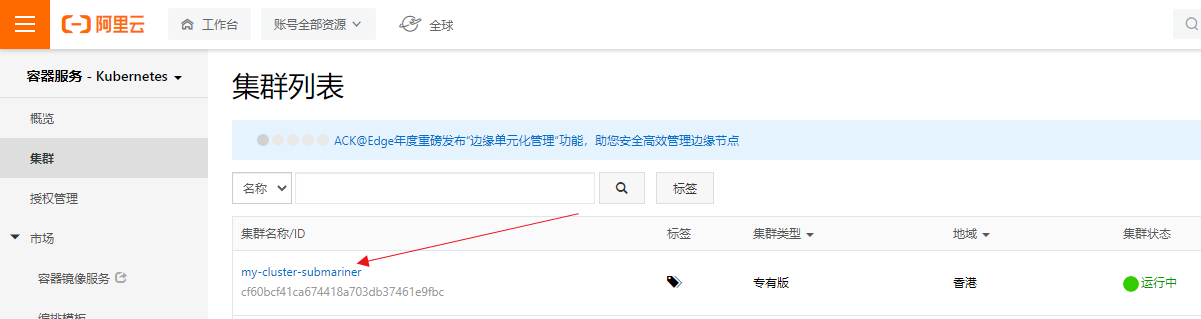




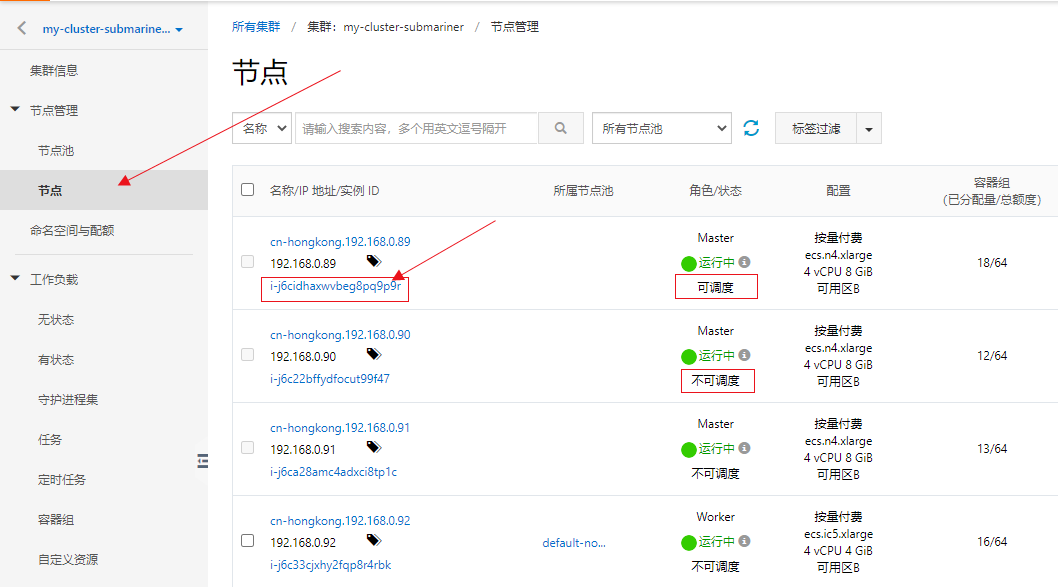


（1.6）配置公网ip

点进集群



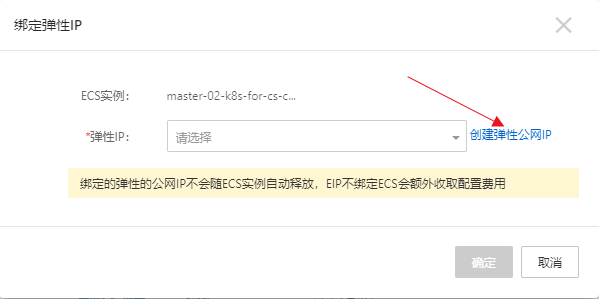
然后点击“节点”，点击节点对应的实例（因为有的节点已经被我禁用了，所以显示“不可调度”，如何禁用节点后面会讲）



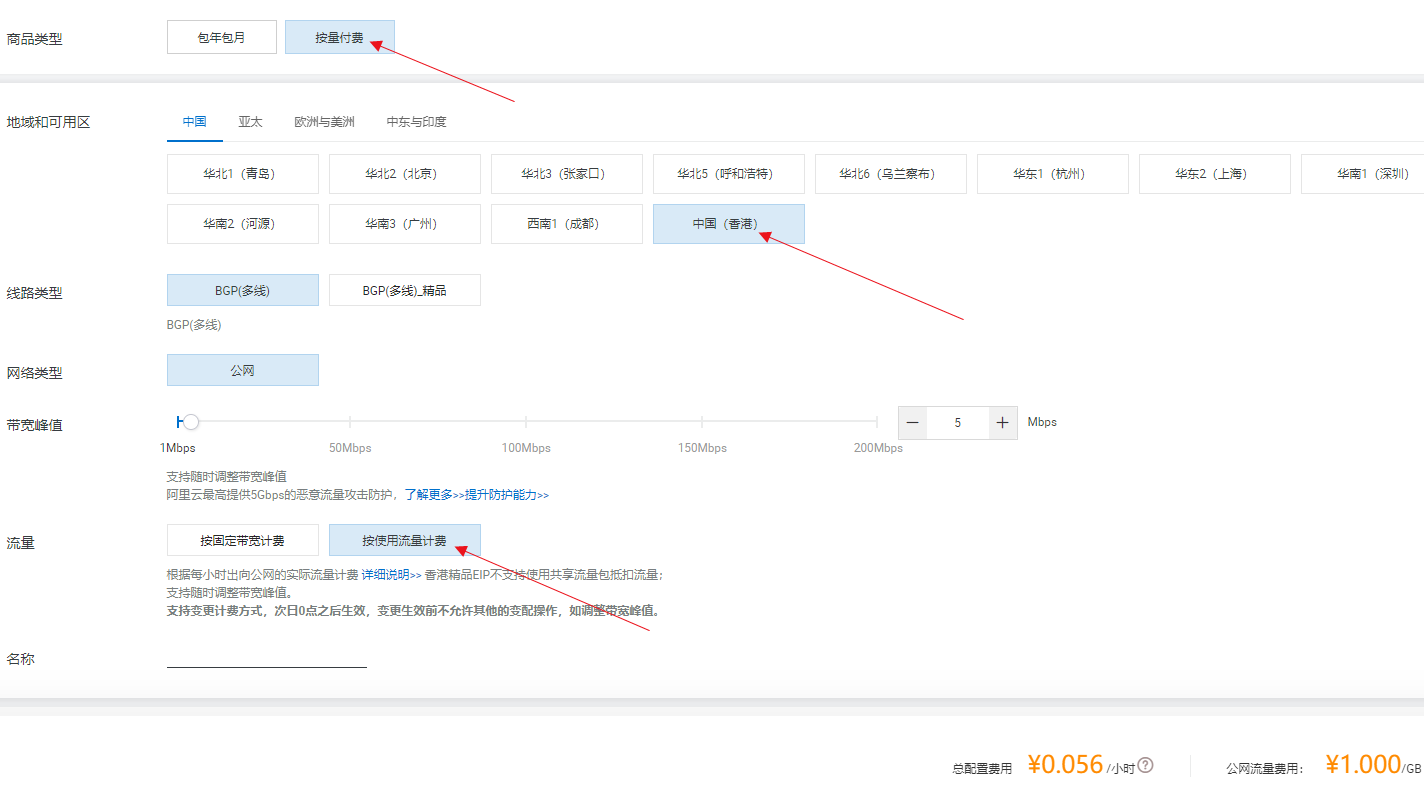
点进去发现这个节点是没有公网ip的，这里必须绑定公网ip，点击“绑定弹性ip”



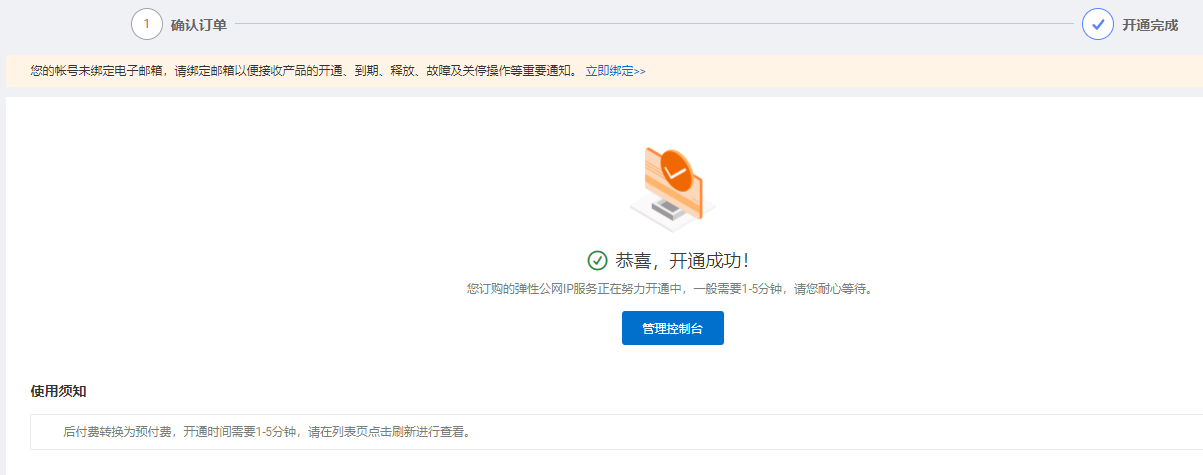
这时候弹出来一个窗口绑定公网ip，如果你没有公网ip，点击“创建弹性公网ip”（其实就是买一个公网ip）



在购买公网ip的时候要注意，如果你的集群在香港，那么你就要购买香港的公网ip



公网ip购买成功



然后再回到刚才的绑定弹性ip的界面，选择刚刚购买的公网ip，绑定就可以了。绑定好之后，会弹出一个窗口让你重启实例，先别急着重启，先配置好实例的账号和密码（如果你已经配置了的话就不用重复配置），然后点击“重启”，等几分钟就好了。

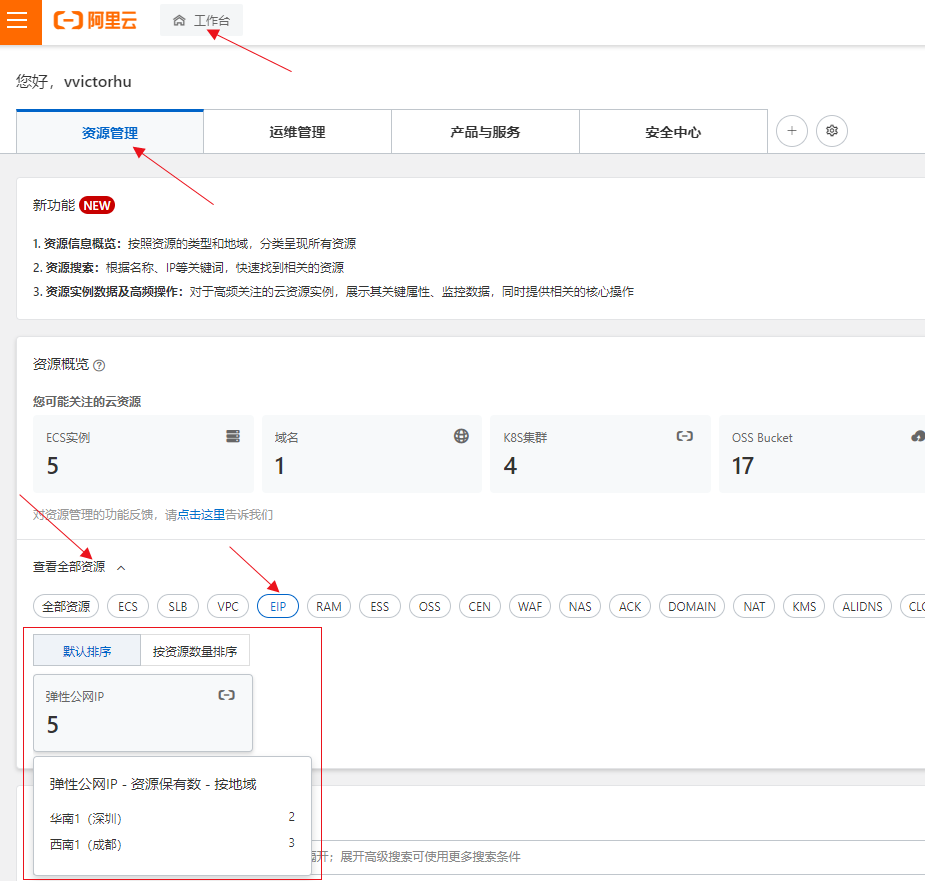
注意，因为ip是另外购买的，所以当你删除了集群，你的公网ip还会被保留，并且继续收费，所以还需要手动删除你购买的公网ip。



可以看到，公网ip已经绑定成功了



删除弹性公网ip，“工作台” 🡪 “资源管理” 🡪 “查看全部资源”，然后选择EIP就可以看到拟购买的EIP了，点进去可以删除指定的EIP，再回到刚才的页面，可以看到你的EIP少了一个，说明你删除成功了



（1.7）配置安全组



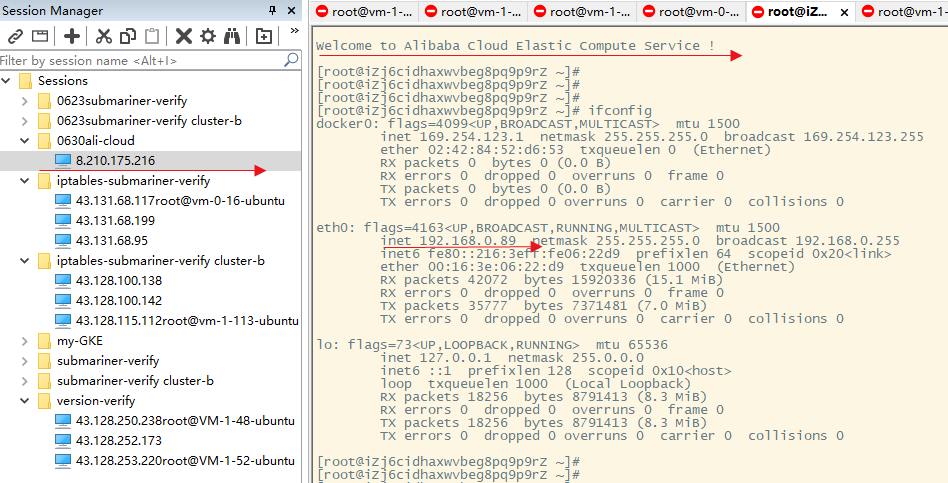
增加一条安全组策略（图片太长，我截成了两段）

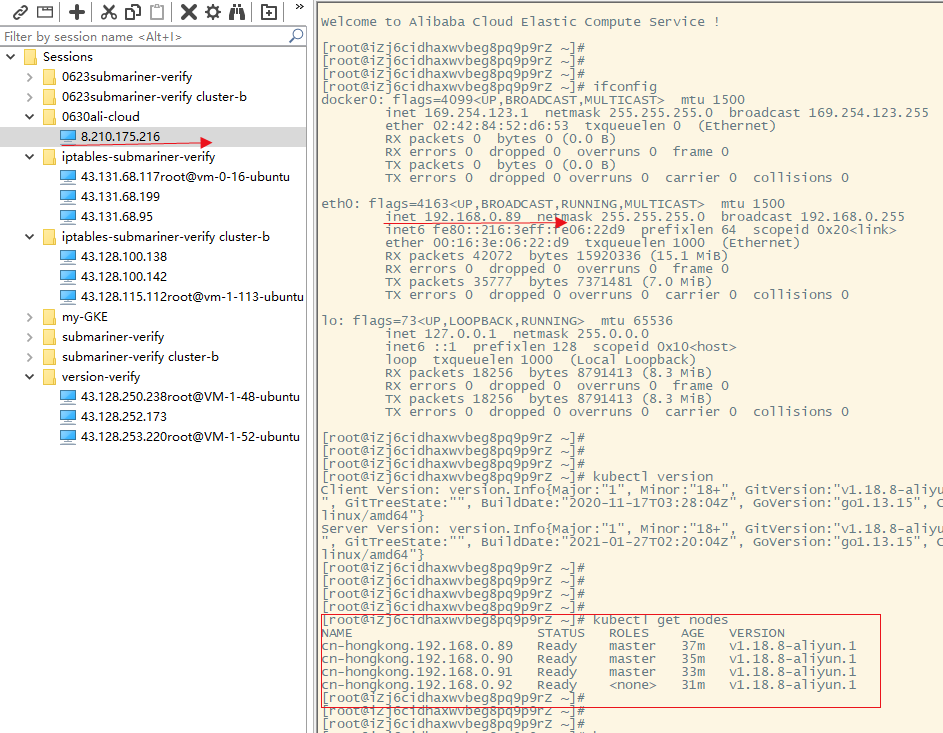






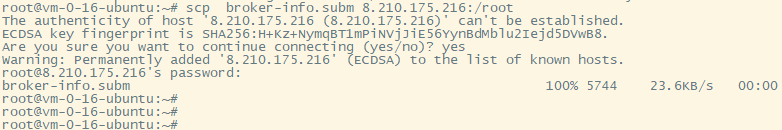
到这里阿里云集群就搭建好了，用secure CRT成功连接





二、cluster-d的准备工作（这里的步骤和之前的验证脚本类似，所以不详细些了）

（2.1）cluster-a把broker.subm文件scp发给cluster-d

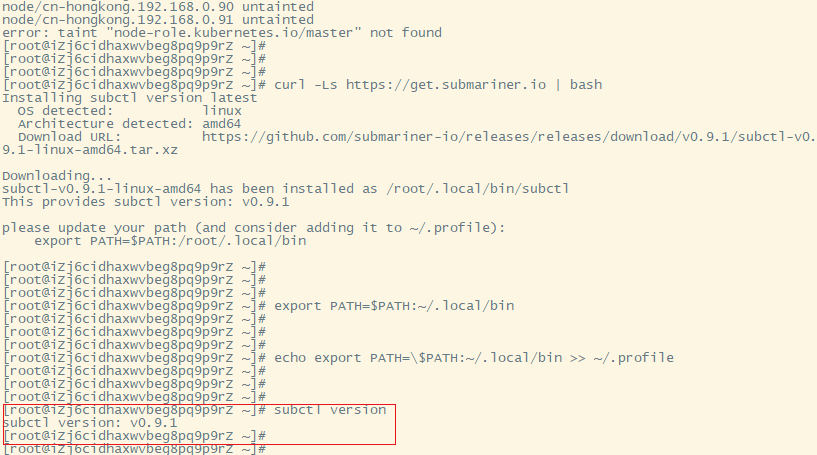


（2.2）安装subctl

curl -Ls https://get.submariner.io | bash

export PATH=$PATH:~/.local/bin

echo export PATH=\$PATH:~/.local/bin >> ~/.profile

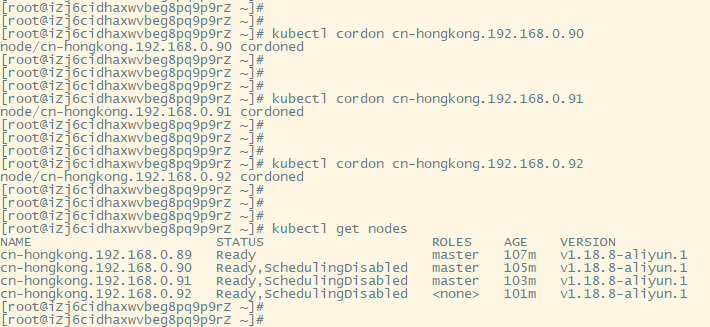


（2.3）# 查看节点

kubectl get nodes

# 把其他node隔离，cordon意思是封锁

kubectl cordon <node name>



节点被禁用，在控制台也是可以看到的，（1.6）讲到过

（2.4）# 把其中一个节点标记为submariner的网关未来部署的节点（这里只需要写一个，因为其他node被禁用了）

# kubectl label node <node节点的名字> submariner.io/gateway=true

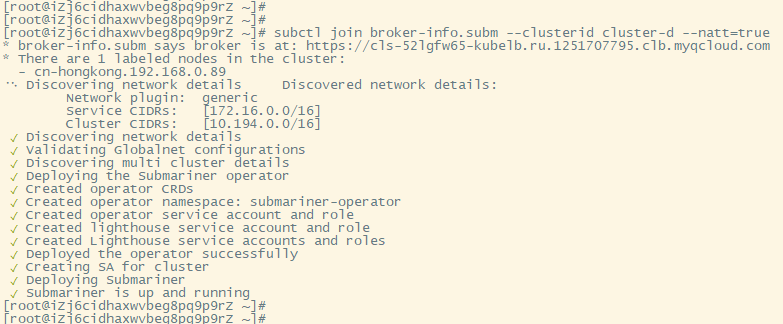
（2.5）# 默认情况下，出于安全原因，集群不会在control-plane节点上调度pod。如果非要在control-plane节点上调度pod

# 参见网页链接https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/

kubectl taint nodes --all node-role.kubernetes.io/master-

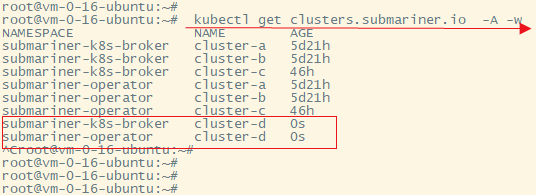
（2.6）# cluster-d以cluster-d的名义加入broker

subctl join broker-info.subm --clusterid cluster-d --natt=true



（2.7）在cluster-a查看集群是否加入

kubectl get clusters.submariner.io -A -w



（2.8）在cluster-d安装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。

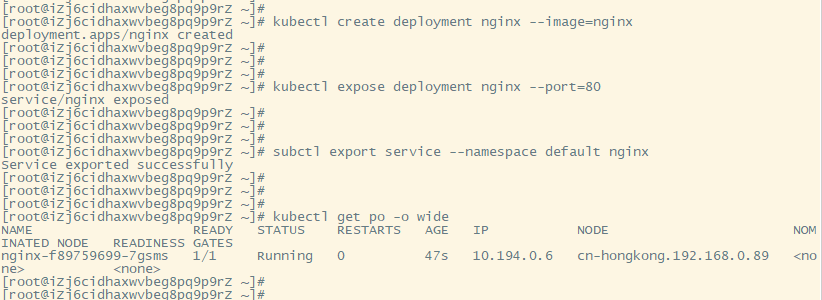
kubectl expose deployment nginx --port=80

#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）

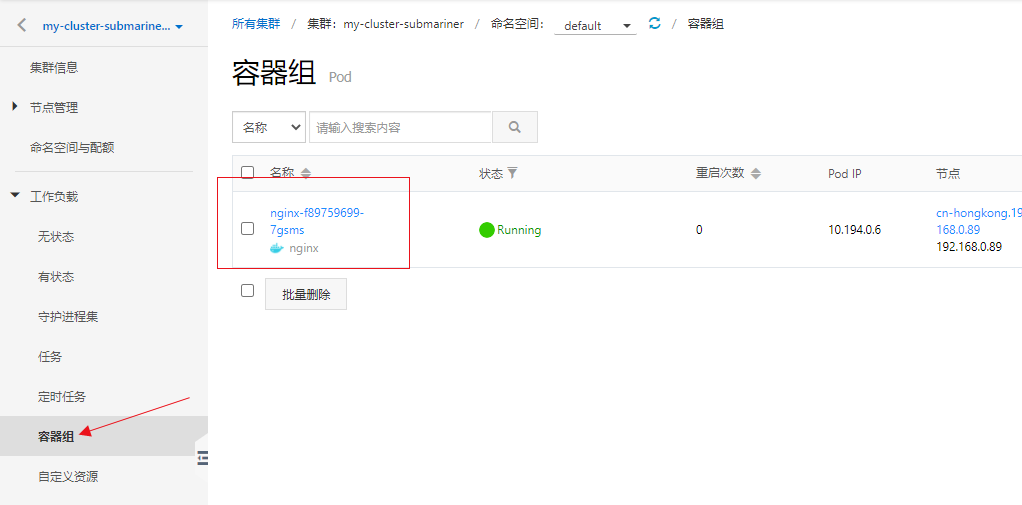
subctl export service --namespace default nginx

# 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



或者在控制台上也可以看到nginx这个pod的相关信息



三、在cluster-a验证cluster-d的nginx是否连通

# 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 \

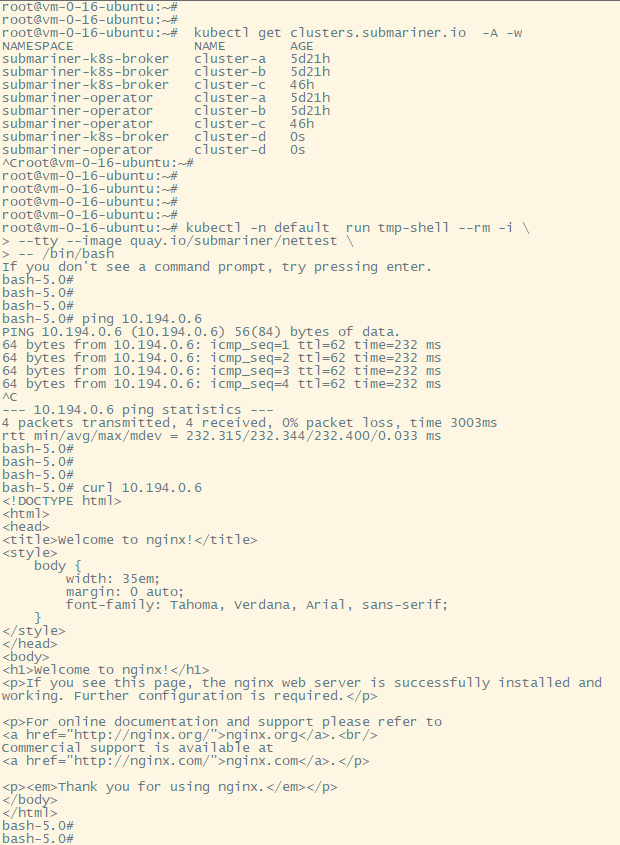
--tty --image quay.io/submariner/nettest \

-- /bin/bash

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

ping <cluster-b显示的nginx的IP>

curl nginx.default.svc.clusterset.local



四、数据总结

|  |  |  |  |
| --- | --- | --- | --- |
| ubuntu 18.04; VPC CIDR: 10.0.0.0/16; Moscow; POD CIDR: 10.0.0.0/24; SERVICE CIDR: 10.45.0.0/24; iptables | | | |
| CentOS 7.9-64bit; VPC CIDR: xxx; Hong Kong; POD CIDR: 10.194.0.0/16; SERVICE CIDR: 172.16.0.0/16; iptables | | | |
| kubernetes version | pod/pod connectivity across cluster | service discovery | note |
| 1.18.4; 1.18.8-aliyun | ok | ok | 独立集群部署；subctl v0.9.0 |