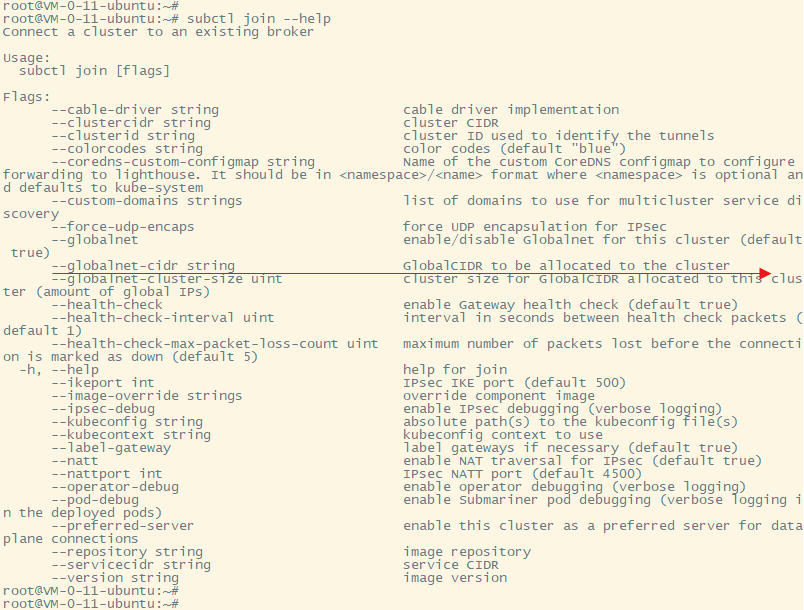
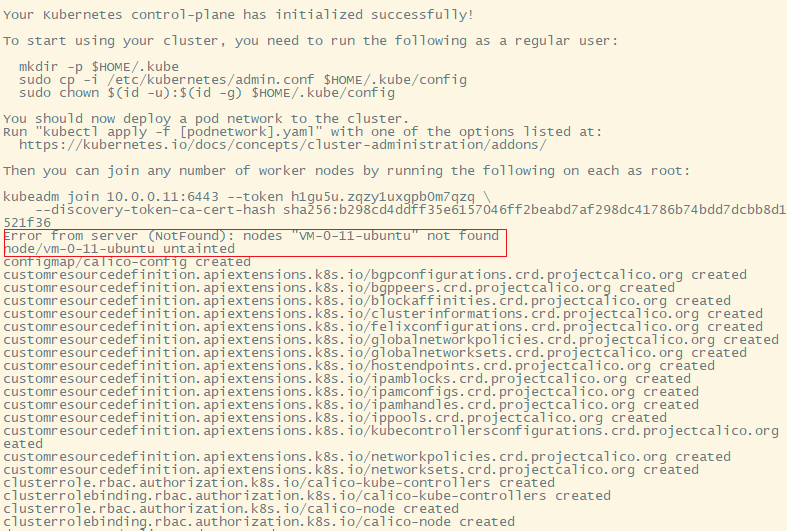
globalnet-CIDR网段冲突问题

20210712





查看service的yaml文档，虽然在创建的时候不适用的yaml文件，而是命令行，但是其实是有的

kubectl get service <service name> -o yaml

kubectl get svc nginx -o yaml

查看pod的yaml文件

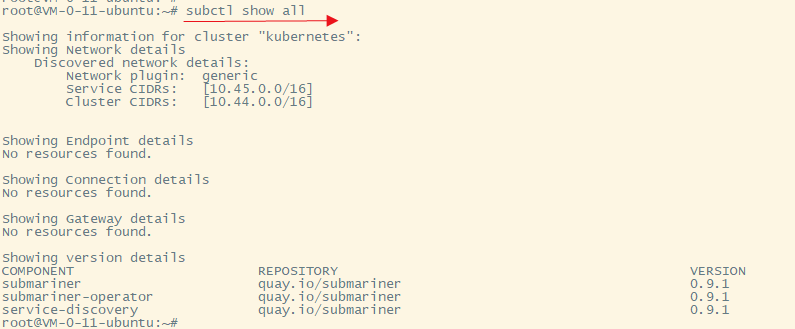
kubectl get po <pod name> -n <namespace> -o yaml

kubectl get po coredns-f9fd979d6-jd8b2 -n kube-system -o yaml

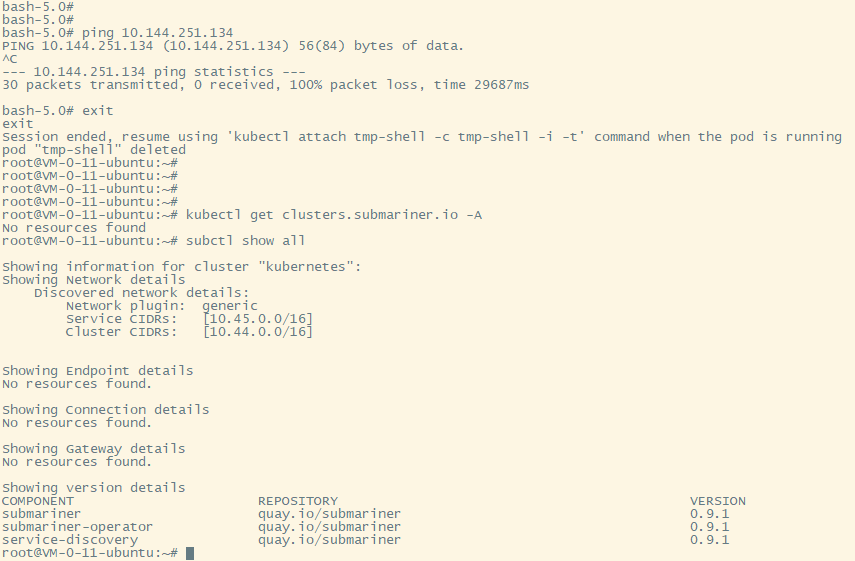
submariner查看部署的结果

subctl show all

从图中可以看到，submariner并没有检测到globalnet-CIDR，说明部署失败



自己申请虚拟机搭建集群，发现使用以前的脚本cluster-a5.sh和cluster-b5.sh用不了了，可是我明明已经控制变量了，怀疑是subctl的版本问题（排查日志发现是hostname的问题）



globalnet 网段冲突问题，参考链接

https://www.cnblogs.com/walkersss/p/11739098.html

<https://blog.csdn.net/weixin_40864891/article/details/106172717>

https://blog.csdn.net/h2728677716/article/details/107003558/

https://submariner.io/getting-started/architecture/globalnet/

https://submariner.io/getting-started/quickstart/openshift/globalnet/

https://submariner.io/operations/deployment/calico/

<https://docs.projectcalico.org/archive/v3.19/maintenance/kubernetes-upgrade#upgrading-an-installation-that-uses-the-kubernetes-api-datastore>

https://github.com/submariner-io/submariner/issues/544

https://github.com/submariner-io/shipyard/commit/466d9506424bb6f8e66881e77e38643a856101d0

https://github.com/submariner-io/submariner/issues/913

https://github.com/submariner-io/shipyard/blob/1b2d412bf0cd86905c5f15912f8cc0cb165b4c34/test/e2e/tcp/connectivity.go

https://github.com/submariner-io/shipyard/blob/3ca03fbbaa064dd7f1e3f3cc4b32340859176e4f/scripts/shared/lib/deploy\_funcs

https://github.com/submariner-io/shipyard/search?q=global+cidr

https://github.com/submariner-io/shipyard/blob/2411d131f223fa6c279390dfaf893b25519dae39/scripts/shared/lib/deploy\_operator

https://github.com/submariner-io/submariner/issues/652

https://github.com/flannel-io/flannel

https://stackoverflow.com/questions/53610641/how-to-delete-remove-calico-cni-from-my-kubernetes-cluster

重新开始搭建环境

集群的网络参数

|  |  |  |  |
| --- | --- | --- | --- |
| Ubuntu 18.04; VPC CIDR: not available; Singapore; POD CIDR: 10.44.0.128/26; SERVICE CIDR: 10.45.0.192/26; iptables; 242.254.144.0/24 | | | |
| Ubuntu 18.04; VPC CIDR: not available; Seoul; POD CIDR: 10.44.0.0/16; SERVICE CIDR: 10.45.0.0/16; iptables; 242.254.145.0/24 | | | |
| kubernetes version | pod/pod connectivity across cluster | service discovery | note |
| 1.19.1 | not pupported original | ok | flannel v0.14.0; subctl v0.10.0-rc0 |

用虚拟机搭建环境和之前的步骤大致相同，只是需要注意以下两点

subctl 的版本不要用v0.9.0，这个版本好像有问题，在与开发者的交流中，开发者也建议使用新版本

curl -Ls https://get.submariner.io | VERSION=0.10.0-rc0 bash

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

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

CNI插件建议使用flannel，calico不太好用，因为需要配置IPPool，如果已经下载了calicoctl，可以先用which命令查看calicoctl的位置，然后再删除该文件即可

which calicoctl

rm /usr/bin/calicoctl

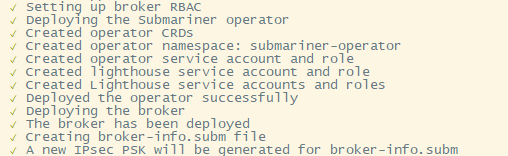
下载flannel的话不需要单独下载，因为相关的命令已经在脚本里面配置好了

关于globalnet-CIDR的脚本，和以前有一些区别

subctl部署的时候，globalnet默认是不开启的，这里需要明确开启

subctl deploy-broker --globalnet enabled

deploy成功的话会得到下图的结果

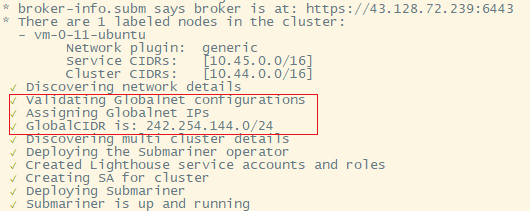


subctl在join的时候需要制定globalnet-CIDR

subctl join broker-info.subm --clusterid <give a cluster name> --globalnet-cidr <give a global-CIDR>

subctl join broker-info.subm --clusterid cluster-a --globalnet-cidr 242.254.144.0/24

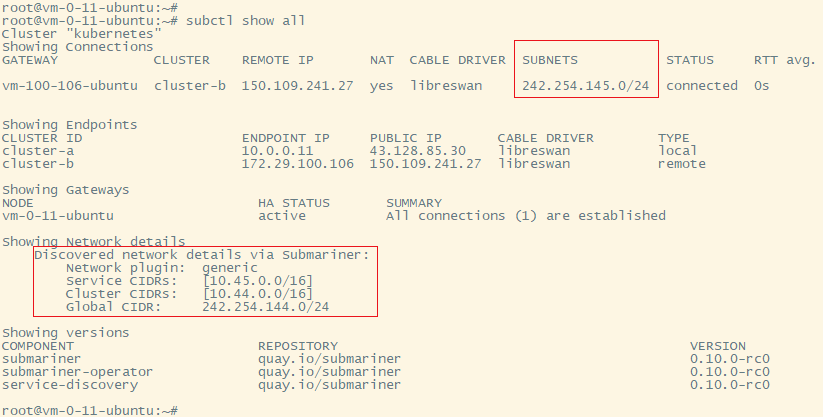
join成功的话会得到下图的结果，红色框的内容就是globalnet-CIDR起作用了



其他的和以前的脚本保持一致

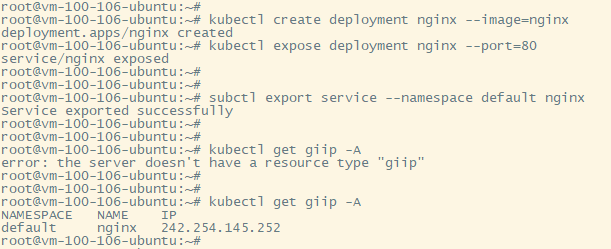
cluster-b的脚本修改也和这个差不多

可以看到，globalnet-CIDR已经生效了，并且集群之间连接成功



在cluster-b部署nginx之后，获取nginx的globalnet IP，注意这里的globalnet IP是service IP而不是pod IP

kubectl get giip -A



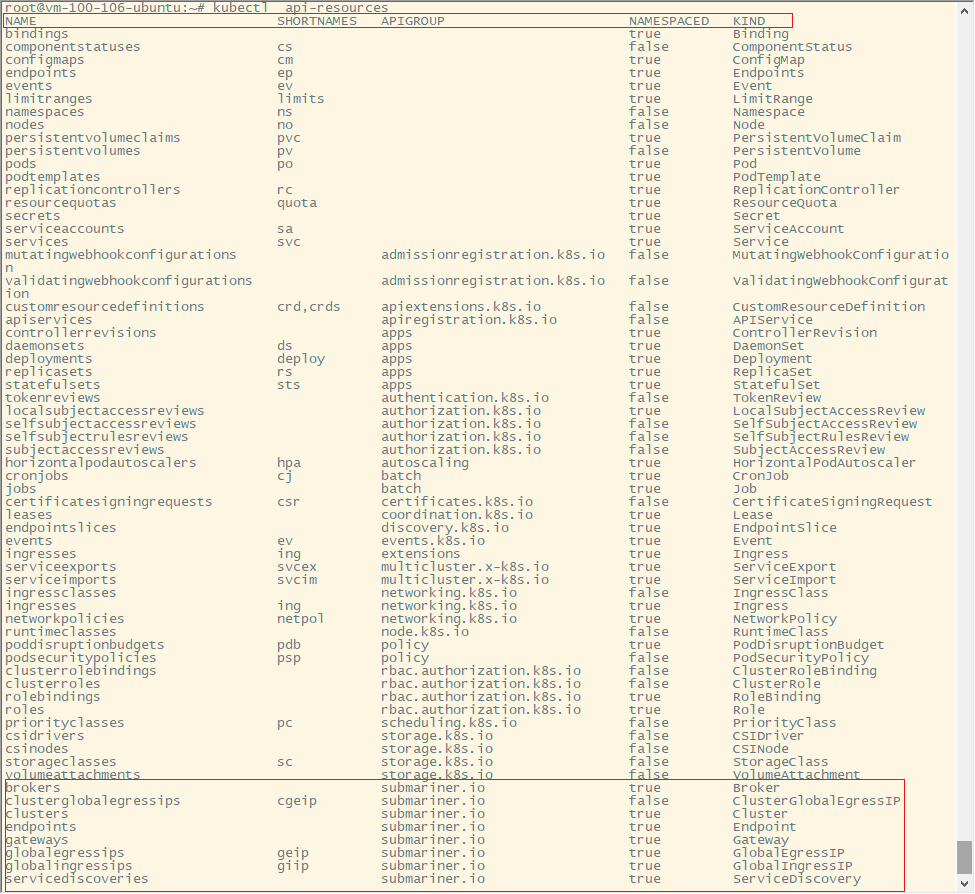
上\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*上\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*上\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*上\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*上\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*上

giip是submariner自己定义的resource，它的全称是globalingressips，这个resource是由submariner.io创建的

kubectl api-resources | grep giip

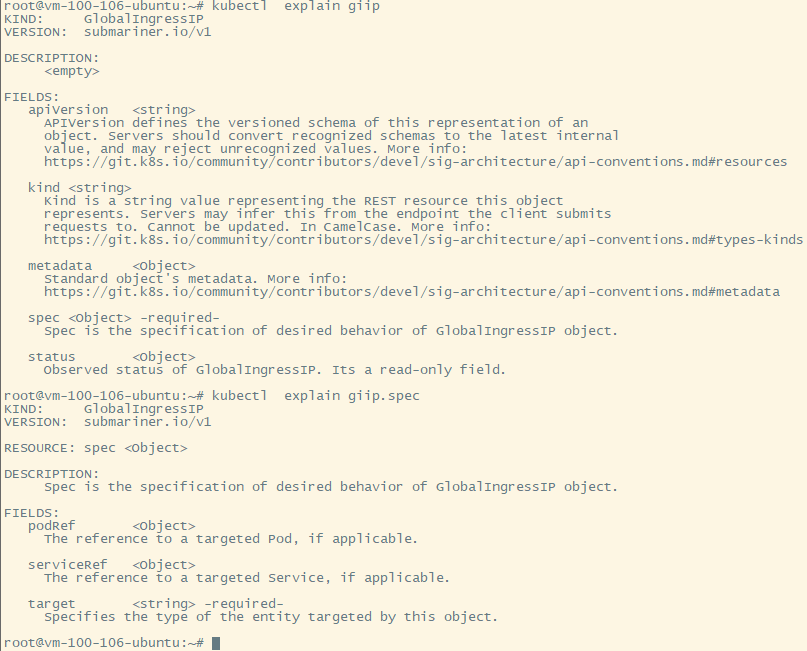


可以看到，submariner.io还创建了很多resource，如下图所示



关于globalingressips，可以用kubectl explain语句查看进一步的信息

kubectl explain <resource name>



Ingress 简单的理解就是你原来需要改配置，然后配置各种域名对应哪个 Service (理解为网络映射关系就好了)，现在把这个动作抽象出来，变成一个 Ingress 对象，你可以用 yaml 创建，每次不要去改配置了，直接改 yaml 然后创建/更新就行了；

实际上Ingress也是Kubernetes API的标准资源类型之一，它其实是一组基于DNS名称（host）或URL路径把请求转发到指定的Service资源的规则。用于将集群外部的请求流量转发到集群内部完成的服务发布。我们需要明白的是，Ingress资源自身不能进行“流量穿透”，仅仅是一组规则的集合，这些集合规则还需要其他功能的辅助，比如监听某套接字，然后根据这些规则的匹配进行路由转发，这些能够为Ingress资源监听套接字并将流量转发的组件就是Ingress Controller。

下\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*下\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*下\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*下\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*下\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*下

可以看到，cluster-a的pod可以ping通cluster-b的service

注意，submariner开发者已经指出，在globalnet-CIDR的模式下，cluster-a的pod无法ping通cluster-b的pod，因为(1)pod的IP地址可能会变动(pod漂移问题)，(2) globalnet-CIDR的模式下，并没有为pod分配globalne IP



