**乱七八糟的命令**

Grep -Epatten1\|patten2

curl localhost:32080/graph -H "Host:grafana.myk8s.poc.com" 用-H指定域名来访问ingress

docker exec -u root -it 48929e1a0ad5 bash

K8s 进入容器进程的运行空间

nsenter -n -t $(docker inspect -f {{.State.Pid}} id)

删除一类标签的pod

kubectl delete po -l k8s-app=nimitz-openvswitch -n kube-system

kubectl delete po -l k8s-app=nimitz-netplugin -n kube-system

dd if=/dev/zero of=/dev/vdb bs=1M count=1

yum install --downloadonly --downloaddir=/tmp <package-name>

ssh -L

ssh -L 9090:192.168.0.48:9090 pre1

9090: 本地开启的端口

192.168.0.48:9090 目标端口

Pre1： 正常远程连接的账号

效果：访问本地9090 端口就是访问192.168.0.48:9090

容器内不显示不正常

解决

docker exec -it --env COLUMNS=`tput cols` --env LINES=`tput lines` 060f079ced6a bash

k8s 跨namespace 访问service

Https://serviename.namespacce:clusterPort/XXX

这里namespace需要加在servicename后面，端口是service的端口，不是nodeport

**一些可以看的知识**

官网：<https://jenkins.io/doc/book/pipeline/>

groovy语法： <http://groovy-lang.org/syntax.html#_shebang_line>

pipline 语法：<https://jenkins.io/doc/book/pipeline/syntax/>

step Sh插件语法：<https://jenkins.io/doc/pipeline/steps/>

skaffold: https://skaffold.dev/docs/

基于k8s的持续构建，热度非常高9.4k；可以用来替换我们isv当前的持续构建方案

restic: https://restic.net/

开源数据备份工具，kubedb的通用数据备份组件底层就是使用restic实现；目前我们的数据备份方案还比较简陋，后续可以使用该工具形成通用的数据备份组件

yq: https://github.com/mikefarah/yq

轻量级的ymal处理命令行工具（类似于jq）；可以用在需要yaml解析，调试的场景

osm: https://github.com/appscode/osm

通用的对象存储命令行工具（类似ossUtil）；可以适用于ceph文件上传场景（无需登录内网机器）