2024.07.19

오전

calico 안에 manifest yaml파일 구성

마스터노드에 kube 스케쥴러와 kube 컨트롤러 kube api(kube proxy, etcd)

worknode kubelet,kube0proxy

conatiner program CRI(cuntainer runtime interface)
-containerd
-cri-o

sudo crictl images

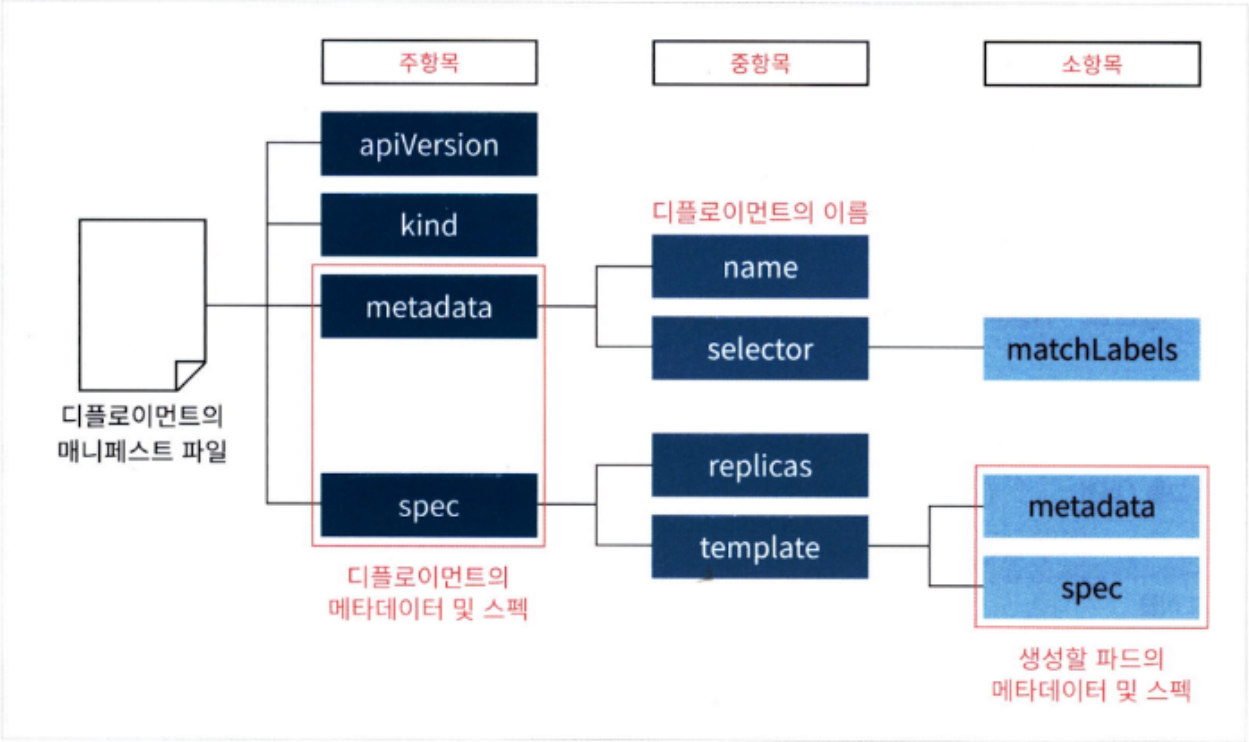


그림 8-5-11 디플로이먼트 매니페스트 파일의 기재 항목 및 내용

```
externalname 파드에서 서비스를 통해 외부로 난가기
```

self healing 파드의 자동복구기술...

k8s 대안 minikube

microk8s ubuntu만든회사꺼

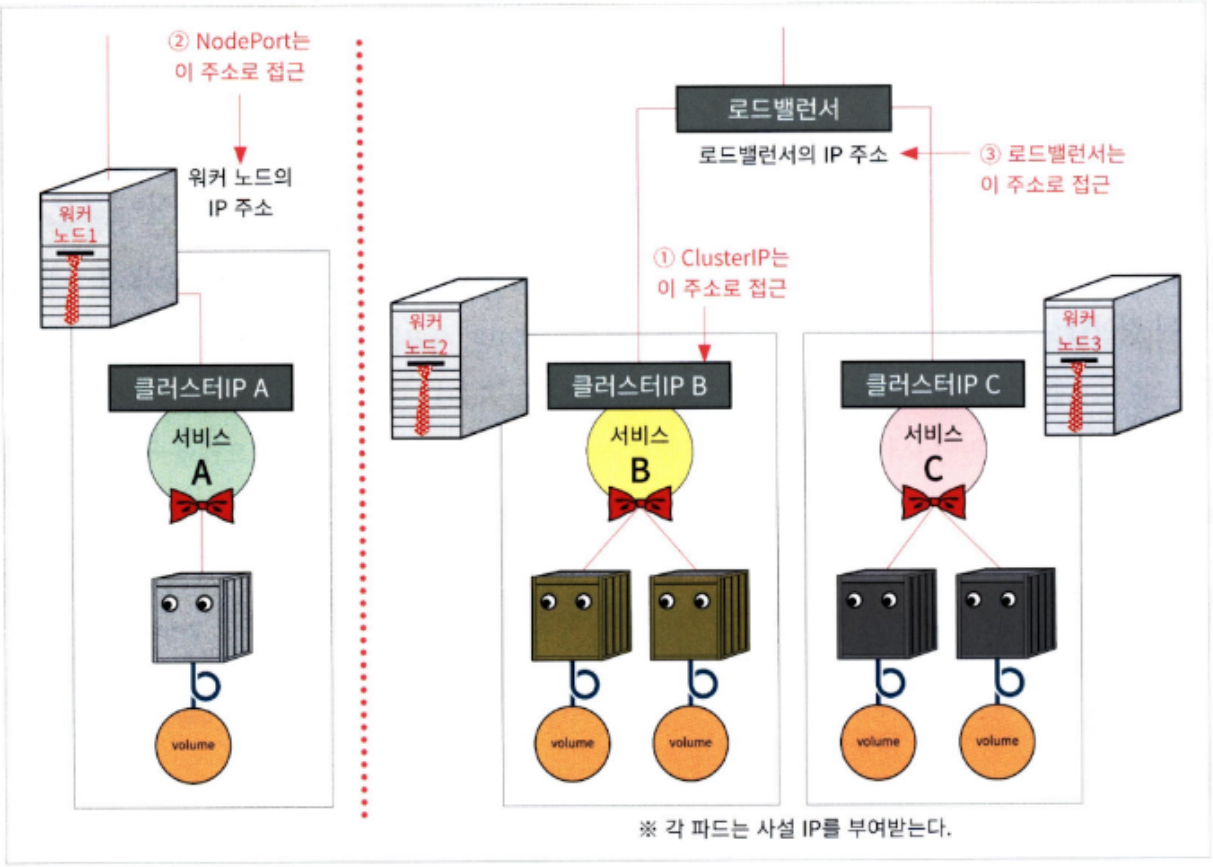
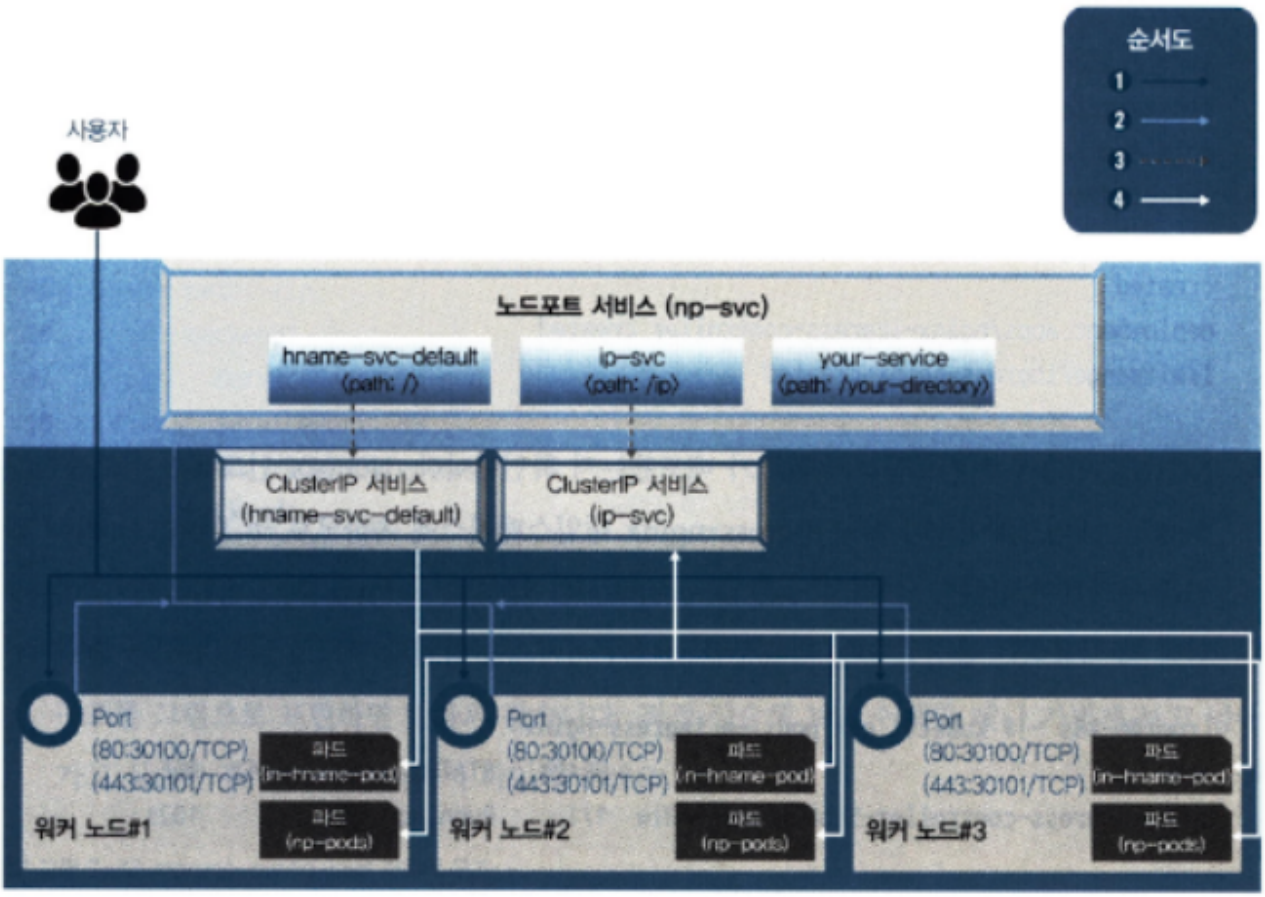


그림 8-5-13 유형 설정 항목과 의미

1253 k delete deployment in-hname-pod
1254 k delete deployment in-ip-pod
1255 k get all
1256 k delete services hname-svc-default
1257 k delete services ip-svc
k get service -A

❯ 그림 3-41 NGINX 인그레스 컨트롤러 서비스 구성도



kubectl create deployment in-hname-pod --image=sysnet4admin/echo-hname
kubectl create deployment in-ip-pod --image=sysnet4admin/echo-ip

베어매탈서버- 물리적인서버, 리얼머신

로컬에서 가상머신으로 구현하거나 baremetal 구성으로 되어 있는 경우 아래 코드 사용
**

wget https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.8.2/deploy/static/provider/baremetal/deploy.yaml

kubectl apply -f deploy.yaml**

kubectl get pods -n ingress-nginx

config
**

3. config

- 해당 파일은 들어오는 주소 값과 포트에 따라 노출된 서비스를 연결하는 역할을 설정
- 외부에서 주소 값과 노드포트를 가지고 들어오는 것은 hname-svc-default 서비스와 연결된 파드로 넘김
    - 외부에서 들어오는 주소 값, 노드포트와 함께 뒤에 /ip를 추가한 주소 값은 ip-svc 서비스와 연결된 파드로 접속하게 설정

****apiVersion: networking.k8s.io/v1**
kind: Ingress
metadata:
name: ingress-nginx
annotations:
nginx.ingress.kubernetes.io/rewrite-target: /
kubernetes.io/ingress.class: "nginx"
spec:
rules:
- http:
paths:
- path: /
pathType: Prefix
backend:
service:
name: hname-svc-default
port:
number: 80
- path: /ip
pathType: Prefix
backend:
service:
name: ip-svc
port:
number: 80

immutabble 뜰시

```
kubectl delete job ingress-nginx-admission-create -n ingress-nginx kubectl delete job ingress-nginx-admission-patch -n ingress-nginx
```

다시 kubectl apply -f deploy.yaml

에러날시

강제삭제..
kubectl get namespace ingress-nginx 종료중이면 강제셧다운

안먹힐시
kubectl get namespace ingress-nginx 상태확인
terminating? 중인가?? 그러면

kubectl get pods -n ingress-nginx 확인해보기

1289 kubectl delete job ingress-nginx-admission-create -n ingress-nginx
1290 kubectl delete job ingress-nginx-admission-patch -n ingress-nginx

k delete -f deploy.yaml
에러가 뜨는가??

kubectl get namespace ingress-nginx

kubectl delete namespace ingress-nginx --grace-period=0 --force
finalizer이 제거안됨.

클러스터가 안건강하다고?

kubectl get ValidatingWebhookConfiguration
여기에 ingress-nginx-admission 1 20m 이자식이있음..
kubectl delete -A ValidatingWebhookConfiguration ingress-nginx-admission

잡이 업는데
kubectl get jobs -n ingress-nginx

kubectl get namespace ingress-nginx -o json | jq '.spec.finalizers'
이게 안되는가?? sudo apt get upgrade
sudo apt install jq
kubectl get namespace ingress-nginx -o json | kubectl patch namespace ingress-nginx -p
'{"spec":{"finalizers":[]}}' --type=merge
임의로 안에걸 초기화해보자
1354 kubectl get ValidatingWebhookConfiguration
1355 kubectl delete -A ValidatingWebhookConfiguration ingress-nginx-admission

1436 kubectl get validatingwebhookconfigurations
1437 kubectl get jobs -n ingress-nginx
1438 kubectl describe job ingress-nginx-admission-create -n ingress-nginx
1439 kubectl get pods -n ingress-nginx
1440 kubectl get validatingwebhookconfigurations
1441 ubectl delete validatingwebhookconfiguration ingress-nginx-admission

1442 kubectl delete validatingwebhookconfiguration ingress-nginx-admission
1443 kubectl get validatingwebhookconfigurations
1444 kubectl get svc -n ingress-nginx
1445 kubectl apply -f deploy.yaml
1456 kubectl get pods -n ingress-nginx
1457 kubectl get all -n ingress-nginx
1458 kubectl get configmaps -n ingress-nginx
1459 kubectl get services -n ingress-nginx
1460 kubectl get deployments -n ingress-nginx
1461 kubectl get jobs -n ingress-nginx

이어서 진행하기

config

- 해당 파일은 들어오는 주소 값과 포트에 따라 노출된 서비스를 연결하는 역할을 설정
- 외부에서 주소 값과 노드포트를 가지고 들어오는 것은 hname-svc-default 서비스와 연결된 파드로 넘김
- 외부에서 들어오는 주소 값, 노드포트와 함께 뒤에 /ip를 추가한 주소 값은 ip-svc 서비스와 연결된 파드로 접속하게 설정
  apiVersion: networking.k8s.io/v1
  kind: Ingress
  metadata:
  name: ingress-nginx
  annotations:
  nginx.ingress.kubernetes.io/rewrite-target: /
  kubernetes.io/ingress.class: "nginx"
  spec:
  rules:
  - http:
  paths:
  - path: /
  pathType: Prefix
  backend:
  service:
  name: hname-svc-default
  port:
  number: 80
  - path: /ip

pathType: Prefix
backend:
service:
name: ip-svc
port:
number: 80

kubectl apply -f ingress-config.yaml
kubectl get ingress -o yaml

**## ingress

- 외부에서 nginx 인그레스 컨트롤러에 접속할 수 있게 노드포트 서비스로 nginx 인그레스 컨트롤러를 외부에 노출
- 30100번 포트로 들어온 요청을 80번 포트로 넘기고, https를 처리하기 위해 30101번 포트로 들어온 것을 443번 포트로 넘김
- 네임스페이스를 ngixn ingress controlller가 위치하는 ingress-nginx로 지정
- nginx ingress controller의 요구 사항에 따라 셀럭터를 ingress-nginx로 지정
- ingress.yaml**
  apiVersion: v1
  kind: Service
  metadata:
  name: nginx-ingress-controller
  namespace: ingress-nginx
  spec:
  ports:
  - name: http
  protocol: TCP
  port: 80
  targetPort: 80
  nodePort: 30100
  - name: https
  protocol: TCP
  port: 443
  targetPort: 443
  nodePort: 30101
  selector:
  app.kubernetes.io/name: ingress-nginx
  type: NodePort

kubectl apply -f ingress.yaml
kubectl get services -n ingress-nginx

port expose

- expose 명령어로 디폴로이먼트도 서비스에 노출
- 외부와 통신하기 위해 클러스터 내부에서만 사용하는 파드를 클러스터 외부에 노출할 수 있는 구역으로 옮기는 것
- 내부와 외부 네트워크를 분리해 관리하는 DMZ 설정
  kubectl expose deployment in-hname-pod --name=hname-svc-default --port=80,443
  kubectl expose deployment in-ip-pod --name=ip-svc --port=80,443
  **

스토리지 볼륨

볼륨-파일보존

컨테이너는 파일내부의 시스템-일시적 죽으면 소멸
pod가 재실행이 되어도 데이터를 보존해야하는데....
이걸하는게 스토리지볼륨
노드 내부의 디스크공간을 파드와 공유
외부스토리지와 시스템 연결

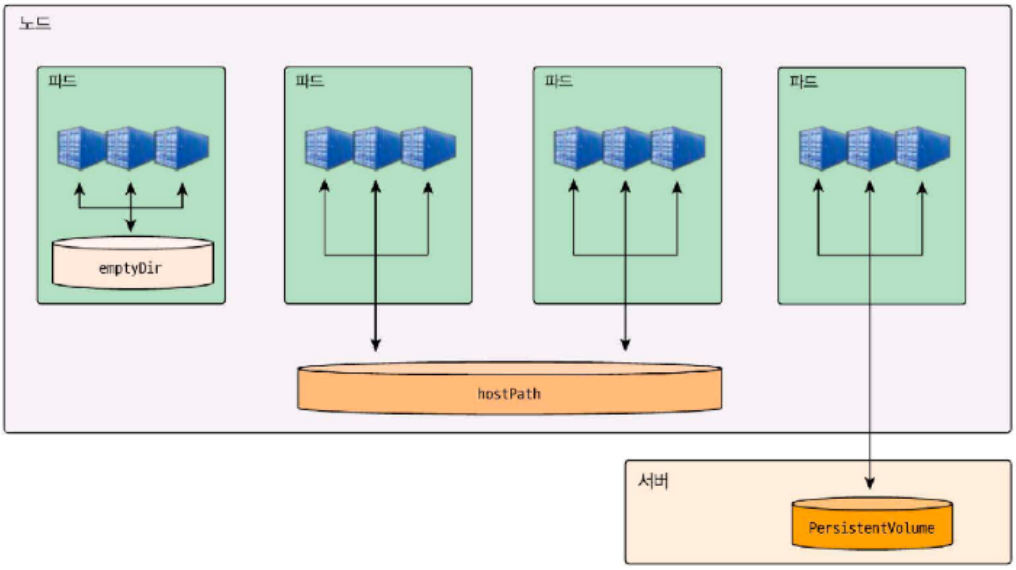


그림 9-22 쿠버네티스 스토리지 볼륨 개념

- emptyDir : 파드 내부에서 임시적으로 사용하는 볼륨
hostPath : 노드 내에 데이터를 저장할 수 있는 볼륨
PersistenVolume: 외부 서버에 데이터를 저장하는 볼륨
\emptyDir
파드가 노드에 할당할 때 처음으로 생성 파드가 노드에서 실행되는 동안만 존재
파드에 존재하는 모든 컨테이너는 emptyDir 볼륨이 존재하는 동안에는 동일한 파일을 읽고 쓸 수있다.
파드가 노드에서 삭제되면 emptyDir 내부에 존재하는 데이터도 삭제**

```
apiVersion: v1
kind: Pod
metadata:
name: nginx-volume-01
spec:
containers:
- name: nginx-test01
image: nginx:latest
volumeMounts:
- name: empty-test01
mountPath: /mount01
volumes:
- name: empty-test01
emptyDir: {}
```

mountPath란 볼륨을 mount할 경로로 작성
name은 볼륨이름, 나중에 생성할 볼륨이름과 일치!
empytydir은 볼륨타입선언
k exec -it nginx-volume-01 -- /bin/bash
그러면 저 안에 mount01이 생성.

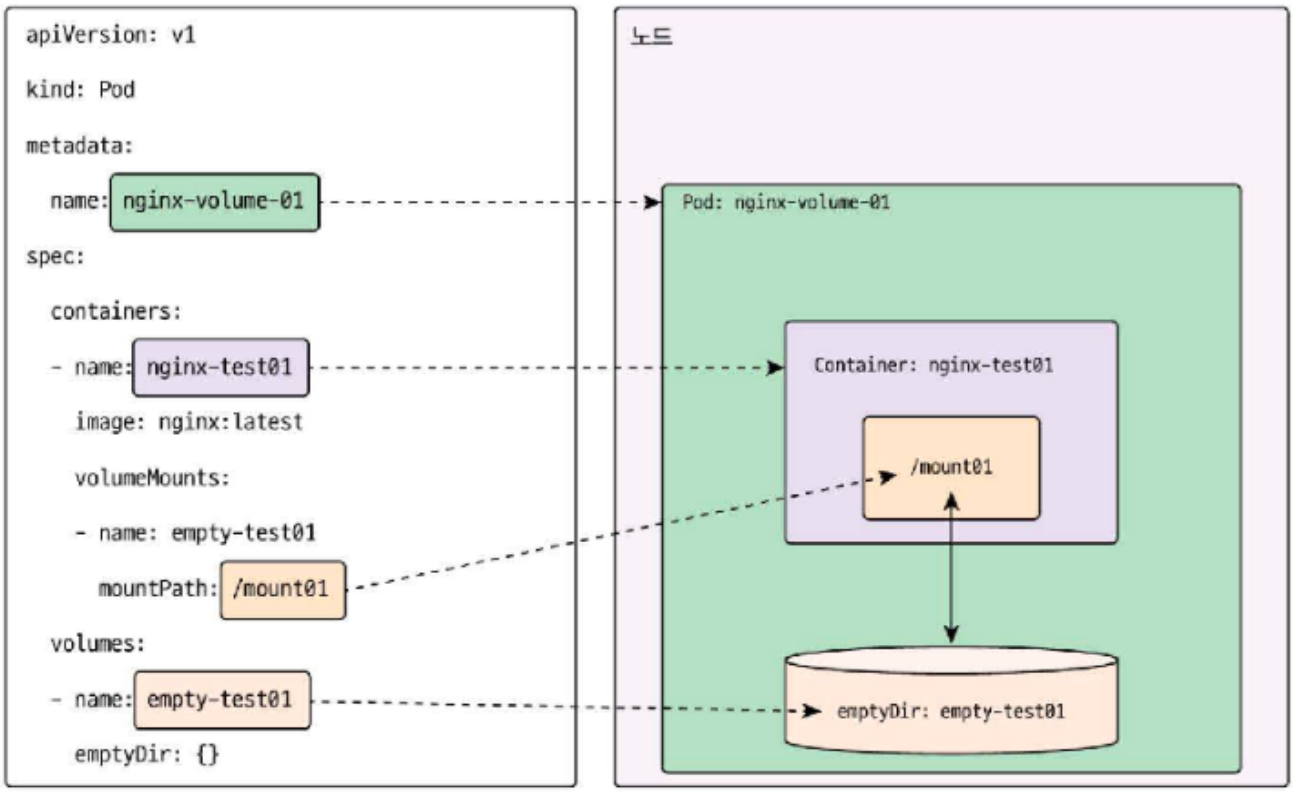


그림 9-25 yml 파일과 실습 내용

k delete -f volume-test01.yaml
다시 들어가면 파일이 있을까? 당연히없지 내부성이었으니깐..

hostPath
노드의 파일시스템ㅇ로부터 파일을 마운트하는것의미.
서로 다른 파드가 동일한 데이터를 보관할 수 있다.
파드가 죽어도 노드에 살아있다
노드가 죽으면 사라진다.

apiVersion: v1
kind: Pod
metadata:
name: nginx-volume-02
spec:
nodeSelector:
kubernetes.io/hostname: myserver03
containers:

- name: nginx-test01
  image: nginx:latest
  volumeMounts:
    - name: hostpath-test01
      mountPath: /mount01
      volumes:
- name: hostpath-test01
  hostPath:
  path: /home/encore/work/volhost01
  type: DirectoryOrCreate

생성이된다...myserver03에서

pv, pvc
persistency volume 외부스토리지
pv를 사용하려며 pv pvc(persistency volume claim)
pv는 스토리지 자원
pvc 스토리지 동적으로바인딩하기 위한 요청객체.
사용자느 pvc를 통해 pv 요청
크러스터가 pv 생성

nfs network file system
원격서버저장소
원격 서버에 존재하는 파일들을 마치 내 컴퓨터 처럼 사용할수 있게 하는 것

3개 전부다
sudo apt update
sudo apt install nfs-common

myserver03만
sudo apt install nfs-kernel-server
sudo systemctl status nfs-server
root권한으로 들어가기.
sudo -i
cd /tmp
mkdir k8s-pv
환경설정하기
nfs서버가 공유하는 디렉토리 설정
/tmp/k8s-pv ip(rw,no_root_squash)
/tmp/k8s-pv 폴더를 ip한테 허용

- rw(읽고, 쓰기)

- no_root_squash -> 클라이언트의 루트 권한을 인정해줄께
  --> 이 옵션을 쓰지 않으면 클라이언트가 해당 디렉토리에 루트 권한으로 쓰기가 안됨
  ** **sudo systemctl restart nfs-server**

다시 myserver01
**apiVersion: v1
kind: PersistentVolume
metadata:
name: pv-01
spec:
accessModes:

- ReadWriteOnce
  capacity:
  storage: 100Mi
  persistentVolumeReclaimPolicy: Retain
  storageClassName: pv-test-01
  nfs:
  server: ip
  path: /tmp/k8s-pv**

accessmodes는 하나의 노드만이 읽기쓰기를 위해 마운트 하는것
**

capacity:
storage: 100Mi

- 용량을 100메가로 설정 이진접미사
  - **persistentVolumeReclaimPolicy: Retain
- pv와 pvc 연결에 대한 정책을 결정하는 옵션**
  - Retain : PVC가 삭제되어도 PV 내부의 데이터는 그대로 유지하는 옵션
  - Delete : PVC가 삭제될 때 PV 역시 삭제되고 연계되어 있는 외부 스토리지 데이터도 삭제
  - Recycle : PV 내부 파일을 삭제할때 사용(지금 사용하지 않음)

**apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: pvc-01
spec:
accessModes:

- ReadWriteOnce
  resources:
  requests:
  storage: 30Mi
  storageClassName: pv-test-01**

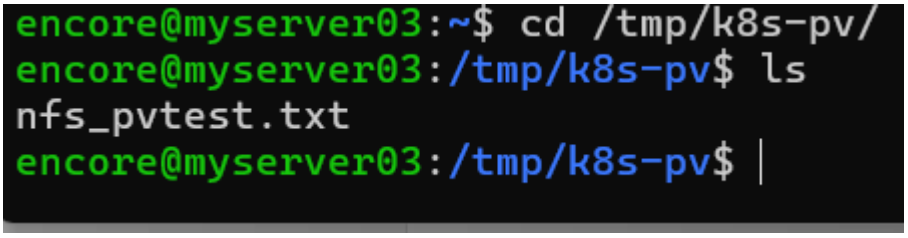
**apiVersion: v1
kind: Pod
metadata:
name: nginx-volume-04
spec:
nodeSelector:
kubernetes.io/hostname: myserver02
containers:

- name: nginx-test01
  image: nginx:latest
  volumeMounts:
  - name: nfs-pv-01
  mountPath: /mount01
  volumes:
- name: nfs-pv-01
  persistentVolumeClaim:
  claimName: pvc-01**

k apply -f volume-test04-1-pv.yaml
먼저 pv를 설계하고
k apply -f volume-test04-2-pvc.yaml
k apply -f volume-test04-3-pod.yaml

k get pv pvc bind되는지 확인
방금 생성된 pod안에 들어가기
cd mount
echo "hi" > ./nfs_pvtest.txt

이제 myserver03들엉가서.



pod는 node2에 생겼어 pvc로 pv에 대여해달라고 해서 binding했고 실제 저장은 nfs-server인 곳에 저장

지울때에는 반대로 pod pvc pv
k get pvc,pv

aws

pem으로 들어간거..
ssh -i encore_ec2_test.pem ubuntu@ip

logging 들어가서 바탕화면에 로그 출력 printable output

auth grdentials에서 private key
public 유동
private 고정
탄력적ip

인바운드규칙

putty

puttygen 에서 pem을 ppk pirvatekey로 save하기
그리고 putty에서 그 key를 넣고 ip와 ㅏ port 22를 친다음 들어가기
logins user ubuntu