올인원 패키지 Online.

CKA 쿠버네티스 자격증 과정

PART1 | CKA 자격증 소개
자격증 등록과 Hands-On 실습 환경 만들기

PART2 | Kubernetes 아키텍쳐
Etcd 백업/복구, Kubernetes 업그레이드, RBAC 인증 실습

PART3 | Workloads & Scheduling
Application deploy 후 scale/rollback, pod 스케줄링 실습

PART4 | Services & Networking
Service 운영과 접근제한(NetworkPolicy),Ingress 운영 실습

PART5 | Storage
StorageClass 적용한 PV, PVC 생성하여 pod에 적용하기 실습

PART6 | Troubleshooting
controller component 설정 정보 수정 및 node/pod 문제해결

PART7 | 실전문제풀이
실전 문제를 직접 풀어 보고, 답안 리뷰 확인

Part 6 Troubleshooting

01 Monitor, log(1)
02 Monitor, log(2)
03 app/cluster/network troubleshooting(1)
04 app/cluster/network troubleshooting(2)

Part 6 Troubleshooting

01 Monitor, log(1)

Application Log 모니터링

- 지정한 Pod 내의 특정 컨테이너 애플리케이션 로그 확인
  kubectl logs PODNAME -c CONTAINER_NAME

  kubectl run web --image=nginx
  kubectl get pods

  kubectl describe pod web

  Kubectl logs web

문제1:
Application Log 추출하기

■ 작업 클러스터 : kubectl config use-context **hk8s**

- Pod custom-app의 로그 모니터링 후 'file not found' 오류가 있는 로그 라인 추출(Extract)해서 /var/CK2022/CUSTOM-LOG001 파일에 저장하시오.

```
$ kubectl config use-context hk8s

$ kubectl logs custom-app | grep -i 'file not found' > /var/CKA2022/CUSTOM-LOG001

$ cat /var/CKA2022/CUSTOM-LOG001
```

Part 6 Troubleshooting

02 Monitor, log(2)

클러스터 리소스 모니터링

- Pod가 사용하는 CPU나 Memory 리소스 정보 보기
  $ kubectl top pods --sort-by=cpu

- Node가 사용하는 CPU나 Memory 리소스 정보 보기
  $ kubectl top nodes --sort-by=cpu

- Json 포맷을 기준으로 특정 리소스 sort 해서 보기
  $ kubectl get pod -o json

  $ kubectl get pods --sort-by=.metadata.name

  $ kubectl get pv --sort-by=.spec.capacity.storage

문제2:
Persistent Volume 정보 보기

■ 작업 클러스터 : **hk8s**

- 클러스터에 구성된 모든 PV를 capacity별로 sort하여 /var/CKA2022/my-pv-list 파일에 저장하시오.
- PV 출력 결과를 sort하기 위해 kubectl 명령만 사용하고, 그 외 리눅스 명령은 적용하지 마시오.

```
$ kubectl config use-context hk8s

$ kubectl get pv -o json pv0001

$ kubectl get pv  --sort-by='{.spec.capacity.storage}' > /var/CKA2022/my-pv-list

$ cat /var/CKA2022/my-pv-list
```

문제3:
클러스터 리소스 정보 보기

■ 작업 클러스터 : kubectl config use-context **hk8s**

- 'name=overloaded-cpu' 레이블을 사용하는 Pod들 중 CPU 소비율이 가장 높은 Pod의 이름을 찾아서 /var/CKA2022/custom-app-log에 기록하시오.

```
$ kubectl get pods -o wide --show-labels --all-namespaces | grep -i name=overloaded-cpu

# 해당 pod CPU만 추출하여 /var/CKA2022/custom-app-log 넣기
$ kubectl top pods  --sort-by=cpu | grep -e campus-01 -e fast-01

$ echo 'campus-01' > /var/CKA2022/custom-app-log
```

Part 6 Troubleshooting

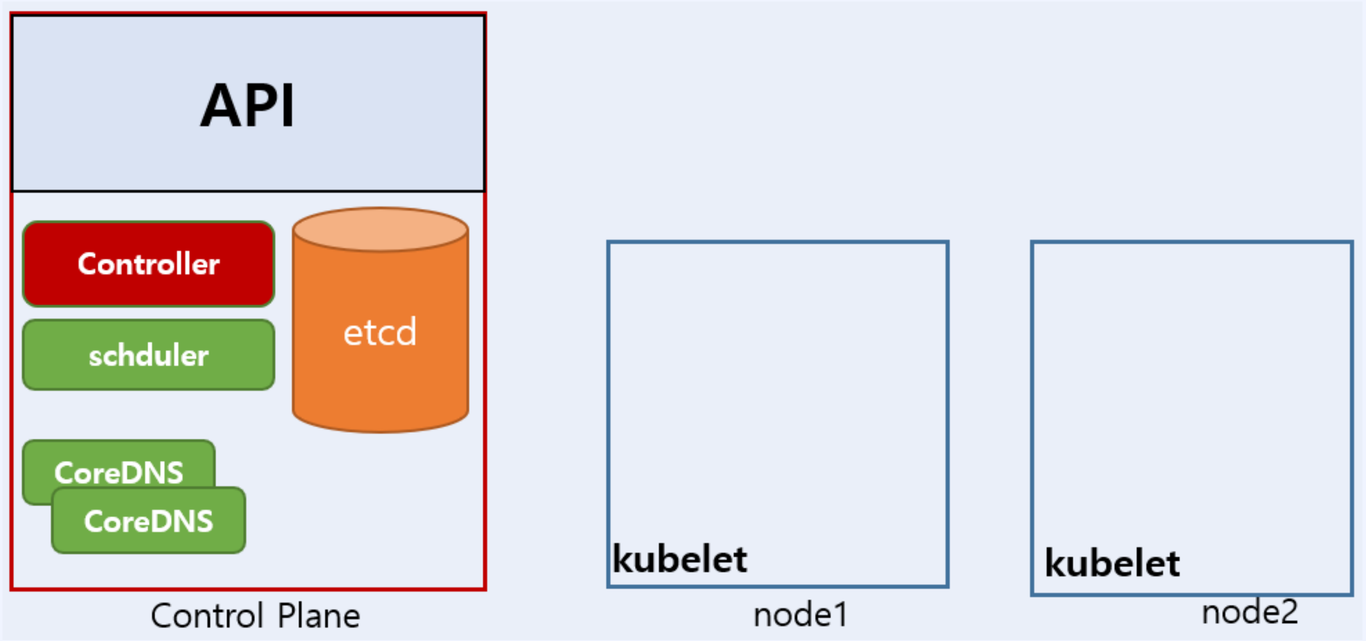
03 app/cluster/network troubleshooting(1)

Worker Node 동작

- Installing runtime

Runtime	Path to Unix domain socket
Docker Engine	/var/run/dockershim.sock
containerd	/run/containerd/containerd.sock
CRI-O	/var/run/crio/crio.sock

- kubelet
- kubeproxy
- cni



문제4:
Worker Node 동작 문제 해결

■ 작업 클러스터 : kubectl config use-context **hk8s**

- Worker Node 동작 문제 해결
- hk8s-w2라는 이름의 worker node가 현재 NotReady 상태에 있습니다. 이 상태의 원인을 조사하고 hk8s-w2 노드를 Ready 상태로 전환하여 영구적으로 유지되도록 운영하시오.

```
$ kubectl get nodes

$ ssh hk8s-w2
$ sudo -i
# docker ps
# systemctl status docker
# systemctl status kubelet
# systemctl enable --now kubelet
# systemctl status kubelet
# exit
$ exit

$ kubectl get nodes
```

Part 6 Troubleshooting

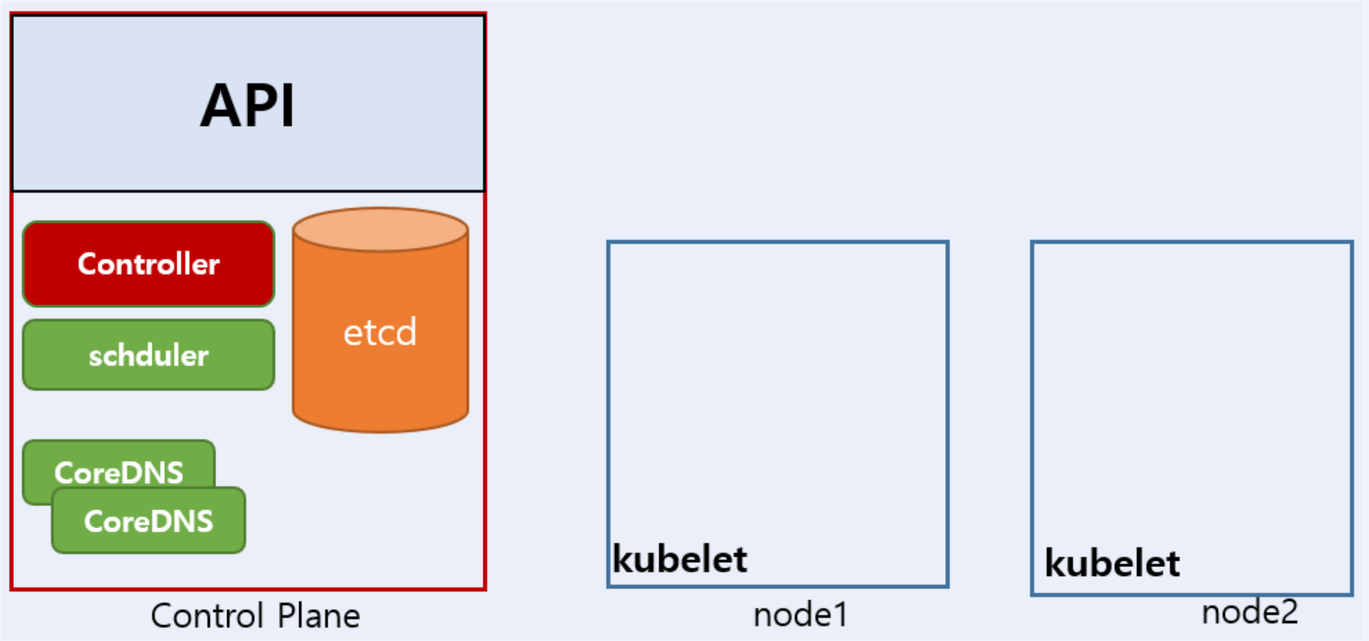
04 app/cluster/network troubleshooting(2)

Worker Node 동작

- Container engine

Runtime	Path to Unix domain socket
Docker Engine	/var/run/dockershim.sock
containerd	/run/containerd/containerd.sock
CRI-O	/var/run/crio/crio.sock

- kubelet
- kubeproxy
- cni



문제5:
Worker Node 동작 문제 해결

■ 작업 클러스터 : kubectl config use-context **hk8s**

- Worker Node 동작 문제 해결
- hk8s-w2라는 이름의 worker node가 현재 NotReady 상태에 있습니다. 이 상태의 원인을 조사하고 hk8s-w2 노드를 Ready 상태로 전환하여 영구적으로 유지되도록 운영하시오.

```
$ kubectl get nodes

$ ssh hk8s-w2
$ sudo -i
# docker ps
# systemctl status docker
# systemctl enable --now docker
# systemctl status docker

# systemctl status kubelet
# exit
$ exit

$ kubectl get nodes
```