Node Scaduling(Troubleshooting)

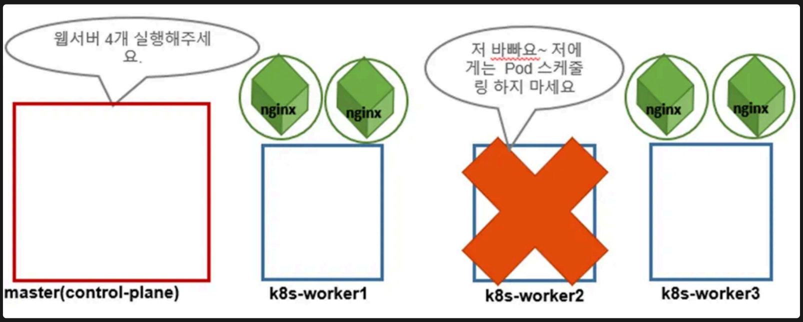
1. Node scaduling

Node 스케줄링 중단 및 허용

- 컨테이너를 포함한 pod는 스케줄러를 통해 node에 assign되어 실행됨
- 특정 노드로 스케줄링 중단(cordon) 및 허용(uncordon)

  kubectl cordon <Node_Name>

  kubectl uncordon <Node_Name>



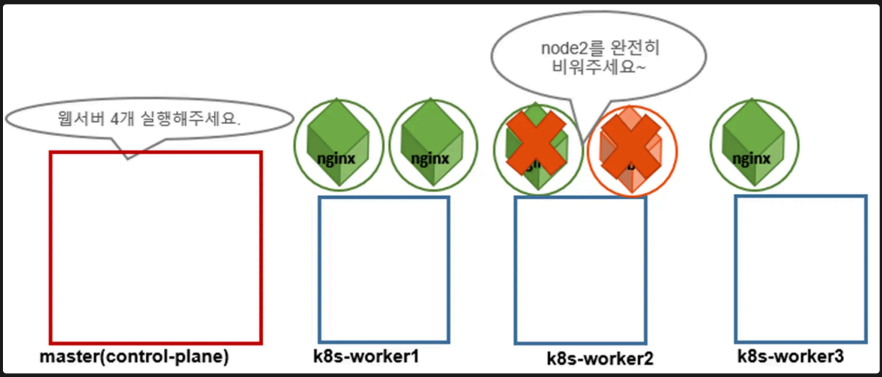
▶ 실습

Node 비우기(drain)

- 노드에서 실행 중인 모든 Pod를 삭제(re-scheduling)하고, 스케쥴링 금지

  kubectl drain <Node_Name> --ignore-daemonsets --force

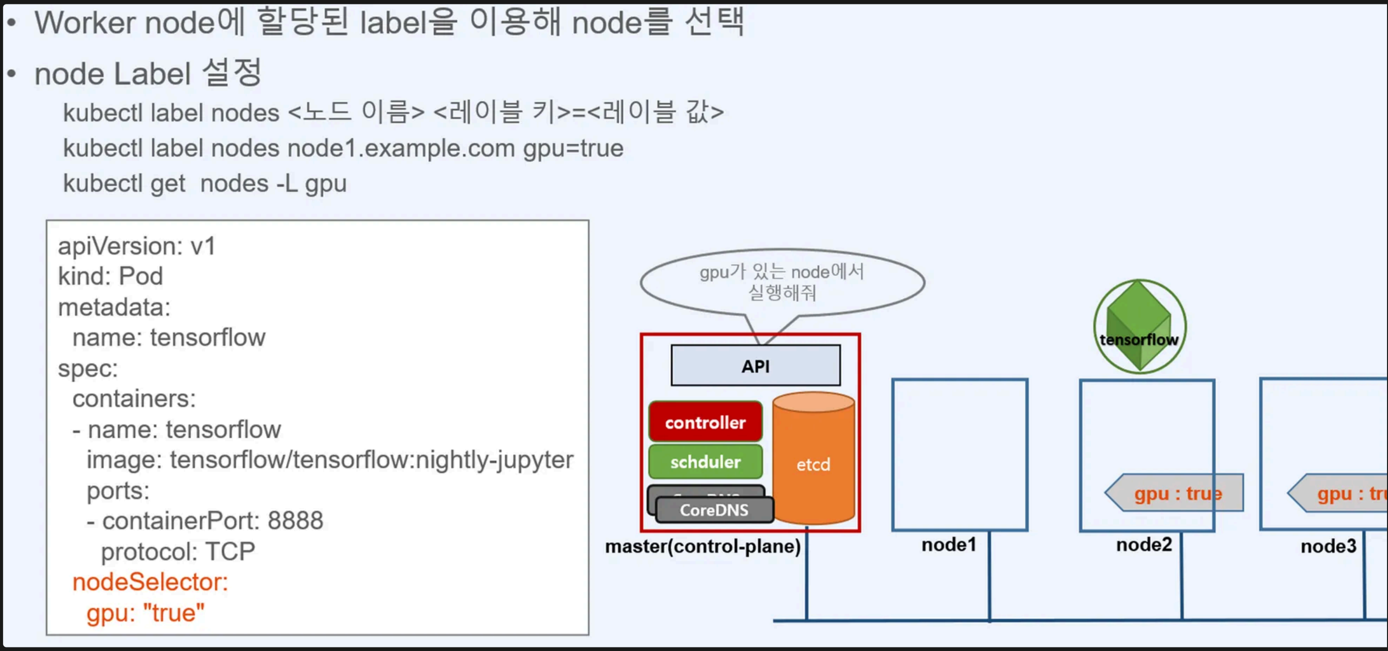
  ```
  --ignore-daemonsets          DaemonSet-managed pod들은 ignore.
  --force=false               RC, RS, Job, DaemonSet 또는 StatefulSet에서 관리하지 않는 Pod까지 제거.
  ```



- ▶ 실습

2. NodeSelector

- NodeSelector



- Worker node에 할당된 label을 이용해 node를 선택
- node Label 설정

```
kubectl label nodes <노드 이름> <레이블 키>=<레이블 값>
kubectl label nodes node1.example.com gpu=true
kubectl get  nodes -L gpu
```

```
apiVersion: v1
kind: Pod
metadata:
  name: tensorflow
spec:
  containers:
  - name: tensorflow
    image: tensorflow/tensorflow:nightly-jupyter
    ports:
    - containerPort: 8888
      protocol: TCP
  nodeSelector:
    gpu: "true"
```

- ▶ Label
- ▶ 실습

3. Taints / Tolerations

https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/

- taints : node에 설정
    - kubectl taint node {nodename} {key}={value}:{option}

```
kubectl taint nodes node1 key1=value1:NoSchedule
```

- tolerations : pod에 설정

```
tolerations: - key: "key1" operator: "Equal" value: "value1" effect: "NoSchedule"
```

- k8s cluster에서 taint 확인

  {key}={value}:{option}

```
# node에 설정된 taints 확인 kubectl describe nodes k8s-master kubectl describe nodes k8s-master |
grep -i taint Taints: node-role.kubernetes.io/control-plane:NoSchedule kubectl describe nodes
k8s-worker1 | grep -i taint Taints: <none> kubectl describe nodes k8s-worker2 | grep -i taint
Taints: <none> # toleration을 설정한 pod 생성 # nginx 파드는 k8s-master, k8s-worker1, k8s-worker2를 대
상으로 스케줄링 된다. cat pod.yaml apiVersion: v1 kind: Pod metadata: name: nginx labels: env: test
spec: containers: - name: nginx image: public.ecr.aws/nginx/nginx:1.26 tolerations: - key:
"node-role.kubernetes.io/control-plane" operator: "Exists" effect: "NoSchedule"
```

- LAB

```
# 테인트 설정 ## control-plane에 설정된 taint 삭제하기 kubectl taint node k8s-master node-
role.kubernetes.io/control-plane:NoSchedule- ## control-plane에 설정된 taint 다시 적용하기 kubectl
taint node k8s-master node-role.kubernetes.io/control-plane:NoSchedule kubectl describe node
k8s-worker1 | grep -i -A 2 taint ## k8s-worker1에 taint 설정후 daemonSet기반의 pod 실행하기 kubectl
taint node k8s-worker1 key1=value1:NoSchedule kubectl describe node k8s-worker1 | grep -i -A 2
taint Taints: key1=value1:NoSchedule # daemonSet example: cat <<END > ds.yaml apiVersion:
apps/v1 kind: DaemonSet metadata: name: daemonset-nginx spec: selector: matchLabels: app: webui
template: metadata: name: nginx-pod labels: app: webui spec: containers: - name: nginx-container
image: public.ecr.aws/nginx/nginx:1.26 END kubectl apply -f ds.yaml # 어느 node에 배포되었나요? k8s-
master, k8s-worker1에 배치되지 않는 이유는? kubectl get pods -o wide kubectl delete -f ds.yaml #
toleration을 넣어서 k8s-worker1을 포함해서 스케쥴링 되도록 구성 cat <<END > ds.yaml apiVersion: apps/v1
kind: DaemonSet metadata: name: daemonset-nginx spec: selector: matchLabels: app: webui
template: metadata: name: nginx-pod labels: app: webui spec: containers: - name: nginx-container
image: nginx:1.14 tolerations: - key: key1 operator: Equal value: value1 effect: NoSchedule END
kubectl apply -f ds.yaml # 어느 node에 배포되었나요? k8s-worker1,k8s-worker2에 스케쥴링 되었는지 확인하세요.
kubectl get pods -o wide kubectl delete -f ds.yaml
```

3. 기출 문제 풀이

❓ 1. Ready Node Count(모니터링)

- 작업 클러스터 : k8s 7%
    - Check to see **how many nodes** are ready (**not including nodes tainted** NoSchedule) and write the number to `/var/CKA2022/ready-nodes`

▶ 답

❓ 2. Schedule a pod

- 작업 클러스터 : **k8s**
- Schedule a **pod** as follows:
  - Name: `eshop-store`
  - Image: `public.ecr.aws/nginx/nginx:1.26`
  - Node selector: `disktype=ssd`

📌 Search Key: NodeSelector (https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes/)

▶ 답