Day 1: Kubernetes

- : Kubernetes
- : Cluster, Node, Pod
- :
  - Kubernetes (minikube)
  - kubectl
  - Pod YAML
  - kubectl Pod

(40%)

1.

Docker

- :
- :
- :
- :
- :

(Container Orchestration) Kubernetes “”

2. Kubernetes

Kubernetes **(Control Plane Nodes) (Worker Nodes)**

a. (Control Plane) -

Pod

- **kube-apiserver**: Kubernetes API kubectl REST etcd
- **etcd**: “ (Source of Truth)” apiserver etcd
- **kube-scheduler**: **Pod** Pod
- **kube-controller-manager**: etcd

b. (Worker Node) -

- **kubelet**: containerd Pod apiserver
- **kube-proxy**: Kubernetes Service Pod
- **Container Runtime**: containerd, CRI-O Docker

3. Cluster, Node, Pod

- **Cluster ()**: Kubernetes
- **Node ()**: Pod
- **Pod**: **Kubernetes** Pod IPPod localhost

□ □□□□ (50%)

1. Kubernetes (minikube)

Minikube Kubernetes

```
#  minikube
# https://minikube.sigs.k8s.io/docs/start/

#  minikube
minikube start --driver=docker
```

2. kubectl

kubectl Kubernetes

```
#  kubectl
# https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/

# minikube start  kubectl
#  kubectl
kubectl cluster-info
#  Master  CoreDNS

#
kubectl get nodes
# NAME       STATUS   ROLES           AGE   VERSION
# minikube   Ready    control-plane   10m   v1.28.3
```

3. Pod

my-first-pod.yaml:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx-container
    image: nginx:1.25
    ports:
    - containerPort: 80
```

- apiVersion: K8s API
- kind: Pod
- metadata: (name) (labels)
- spec: Pod

kubectl Pod:

```
kubectl apply -f my-first-pod.yaml
# pod/nginx-pod created
```

4. Pod

```
#  Pod
kubectl get pods
# NAME        READY   STATUS    RESTARTS   AGE
# nginx-pod   1/1     Running   0          30s

#  IP
kubectl get pods -o wide

#  Pod
kubectl describe pod nginx-pod

#  Pod
kubectl logs nginx-pod

#  Pod  ( docker exec)
kubectl exec -it nginx-pod -- /bin/bash
# root@nginx-pod:/# ls
# root@nginx-pod:/# exit
```

5. Pod

```
kubectl delete -f my-first-pod.yaml
# pod "nginx-pod" deleted

#
kubectl delete pod nginx-pod
```

Go (10%)

: k8s-cluster-info

: Go client-go

1.

```
mkdir k8s-cluster-info
cd k8s-cluster-info
go mod init cluster.info.dev/me
go get k8s.io/client-go@v0.28.2 k8s.io/api@v0.28.2 k8s.io/apimachinery@v0.28.2
```

2. (main.go)

```
package main

import (
        "context"
        "fmt"
        "log"
        "os"
        "path/filepath"

        metav1 "k8s.io/apimachinery/pkg/apis/meta/v1"
        "k8s.io/client-go/kubernetes"
        "k8s.io/client-go/tools/clientcmd"
)

func main() {
        // 1.  kubeconfig
        userHomeDir, err := os.UserHomeDir()
        if err != nil {
                log.Fatalf(": %v", err)
        }
        kubeconfigPath := filepath.Join(userHomeDir, ".kube", "config")

        // 2.
        config, err := clientcmd.BuildConfigFromFlags("", kubeconfigPath)
        if err != nil {
                log.Fatalf(" kubeconfig : %v", err)
        }

        // 3.  clientset
        clientset, err := kubernetes.NewForConfig(config)
        if err != nil {
                log.Fatalf(" clientset : %v", err)
        }

        // 4.  clientset  API Server
        fmt.Println("--- Kubernetes Nodes ---")
        nodes, err := clientset.CoreV1().Nodes().List(context.TODO(), metav1.ListOptions{})
        if err != nil {
                log.Fatalf(": %v", err)
        }

        for _, node := range nodes.Items {
                fmt.Printf("- Name: %s\n", node.Name)
                fmt.Printf("  Kubelet Version: %s\n", node.Status.NodeInfo.KubeletVersion)
                fmt.Printf("  OS: %s\n", node.Status.NodeInfo.OperatingSystem)
                fmt.Println("--------------------")
        }
}
```

3.

```
go run main.go
# --- Kubernetes Nodes ---
# - Name: minikube
#   Kubelet Version: v1.28.3
#   OS: linux
# --------------------
```

- **kubectl** :
  - ~/.kube/config
  - minikube status
- **Pod Pending**:
  - kubectl describe pod <pod-name>
- **Pod ImagePullBackOff ErrImagePull**:
  - kubectl describe pod <pod-name> Tag

1. **Pod** : Pod Pending, Running, Succeeded, Failed, Unknown
2. **Pod**: my-first-pod.yaml Pod busybox image: busybox5 (command: ["/bin/sh", "-c", "while true; do date; sleep 5; done"] kubectl logs nginx-pod -c busybox-container busybox