

REFERENSI CEPAT KUBERNETES

kubectl, pod, deployment, service, config, debugging

Dasar kubectl

Info Cluster
kubectl cluster-info
kubectl get nodes
kubectl config current-context
kubectl config use-context my-cluster

Perintah Utama

kubectl get <resource> Tampilkan daftar resource
kubectl describe <resource> <name> Info detail resource
kubectl create -f file.yaml Buat resource dari file
kubectl apply -f file.yaml Buat atau perbarui resource
kubectl delete -f file.yaml Hapus resource dari file
kubectl edit <resource> <name> Edit resource di tempat
kubectl api-resources Tampilkan semua tipe resource

Format Output

-o wide Kolom tambahan (IP, node)
-o yaml Output YAML lengkap
-o json Output JSON lengkap
-o jsonpath='{.spec}' Ekstrak field tertentu
--sort-by=<metadata.name> Urutkan output berdasarkan field

Pod

Operasi Pod

kubectl get pods
kubectl get pods -A # all namespaces
kubectl run nginx --image=nginx # quick pod
kubectl delete pod nginx

YAML Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp
  labels: { app: myapp }
spec:
  containers:
  - name: app
    image: nginx:1.27
    ports:
    - containerPort: 80
```

Nilai Status Pod

Running Semua container sudah berjalan
Pending Menunggu penjadwalan atau pull image
CrashLoopBackOff Container terus crash dan restart
ImagePullBackOff Gagal mengunduh image container
Completed Selesai berjalan (Jobs)

Deployment

YAML Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web
spec:
  replicas: 3
  selector:
    matchLabels: { app: web }
  template:
    metadata:
      labels: { app: web }
    spec:
      containers:
      - name: web
        image: nginx:1.27
        ports:
        - containerPort: 80
```

Perintah Deployment

kubectl get deploy Tampilkan deployment
kubectl scale deploy web --replicas=5 Skala replica
kubectl set image deploy/web web=nginx:1.28 Update image (rolling)
kubectl rollout status deploy/web Pantau progres rollout
kubectl rollout undo deploy/web Rollback ke revisi sebelumnya
kubectl rollout history deploy/web Lihat riwayat revisi

Service

Jenis Service
ClusterIP Internal saja (default)
NodePort Ekspos di IP setiap node pada port statis
LoadBalancer Load balancer eksternal (cloud)
ExternalName Alias DNS ke layanan eksternal

YAML Service

```
apiVersion: v1
kind: Service
metadata:
  name: web-svc
spec:
  type: ClusterIP
  selector: { app: web }
  ports:
  - port: 80
    targetPort: 80
```

Ekspos Cepat

kubectl expose deploy web --port=80 --type=ClusterIP
kubectl expose deploy web --port=80 --type=NodePort
kubectl get svc

ConfigMap & Secret

ConfigMap
kubectl create configmap app-cfg \ --from-literal=DB_HOST=db.example.com \ --from-file=config.ini

Secret

kubectl create secret generic db-creds \ --from-literal=username=admin \ --from-literal=password=s3cret

Penggunaan dalam Pod

```
# As environment variables
envFrom:
- configMapRef: { name: app-cfg }
- secretRef: { name: db-creds }

# As volume mount
volumes:
- name: cfg
  configMap: { name: app-cfg }
```

Perintah

kubectl get cm Tampilkan ConfigMap
kubectl get secret Tampilkan Secret
kubectl describe cm app-cfg Tampilkan data ConfigMap
kubectl get secret db-creds -o yaml Tampilkan Secret (base64-encoded)

Namespace

Perintah Namespace

kubectl get ns Tampilkan namespace
kubectl create ns staging Buat namespace
kubectl delete ns staging Hapus namespace dan semua resource
kubectl get pods -n staging Tampilkan pod di namespace
kubectl get pods -A Tampilkan pod di semua namespace

Atur Namespace Default

kubectl config set-context --current \ --namespace=staging

Volume

PersistentVolumeClaim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: data-pvc
spec:
  accessModes: [ReadWriteOnce]
  resources:
    requests: { storage: 10Gi }
```

Mount dalam Pod

```
volumes:
- name: data
  persistentVolumeClaim:
    claimName: data-pvc
containers:
- volumeMounts:
  - name: data
    mountPath: /app/data
```

Jenis Volume

emptyDir Direktori sementara, dihapus bersama pod
hostPath Mount path filesystem host
persistentVolumeClaim Penyimpanan persisten (PVC)
configMap Mount ConfigMap sebagai file
secret Mount Secret sebagai file

Ingress

YAML Ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: web-ingress
spec:
  rules:
  - host: app.example.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: web-svc
            port: { number: 80 }
```

Catatan Ingress

Ingress Controller Wajib ada (nginx-ingress, traefik, dll.)
pathType: Prefix Cocokkan awalan URL
pathType: Exact Cocokkan path URL tepat
TLS Tambahkan bagian `tls:` dengan nama secret

Debugging

Perintah Diagnostik

kubectl logs <pod> stdout/stderr container
kubectl logs <pod> -c <ctz> Log container tertentu

kubectl logs <pod> --previous

Log dari container yang crash

kubectl describe pod <pod>

Event, kondisi, status Masuk shell ke container

kubectl exec -it <pod> -- sh

Forward port lokal ke pod

kubectl port-forward <pod> 8080:80

Penggunaan CPU/memori (metrics-server)

kubectl get events --sort-by=.lastTimestamp

Timeline event cluster

Pod Debug

kubectl run debug --rm -it --image=busybox -- sh
or attach ephemeral container
kubectl debug -it <pod> --image=busybox

Pola Umum

Label & Selector

kubectl get pods -l app=web
kubectl get pods -l 'env in (prod,staging)'
kubectl label pod myapp env=prod

Batas Resource

```
resources:
  requests: { cpu: 100m, memory: 128Mi }
  limits: { cpu: 500m, memory: 256Mi }
```

Liveness & Readiness

```
livenessProbe:
  httpGet: { path: /healthz, port: 8080 }
  initialDelaySeconds: 5
  periodSeconds: 10
readinessProbe:
  httpGet: { path: /ready, port: 8080 }
```

Resep Cepat

Dry run `kubectl apply -f file.yaml --dry-run=client`
Generate YAML `kubectl create deploy web --image=nginx --dry-run=client -o yaml`
Watch `kubectl get pods -w`
Copy file `kubectl cp file.txt pod:/tmp/`
Restart deploy `kubectl rollout restart deploy/web`