

Tham Khảo Nhanh Kubernetes

kubectl, pods, deployments, services, configs, debugging

Cơ Bản kubectl

Thông Tin Cluster

```
kubectl cluster-info
kubectl get nodes
kubectl config current-context
kubectl config use-context my-cluster
```

Lệnh Thiết Yếu

kubectl get <resource>	Liệt kê tài nguyên
kubectl describe <resource> <name>	Thông tin chi tiết tài nguyên
kubectl create -f file.yaml	Tạo tài nguyên từ file
kubectl apply -f file.yaml	Tạo hoặc cập nhật tài nguyên
kubectl delete -f file.yaml	Xóa tài nguyên từ file
kubectl edit <resource> <name>	Chỉnh sửa tài nguyên tại chỗ
kubectl api-resources	Liệt kê tất cả loại tài nguyên

Định Dạng Output

-o wide	Thêm cột (IP, node)
-o yaml	Output YAML đầy đủ
-o json	Output JSON đầy đủ
-o jsonpath='{.spec}'	Trích xuất trường cụ thể
--sort-by=.metadata.name	Sắp xếp output theo trường

Pods

Thao Tác Pod

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

Pod YAML

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

Giá Trị Trạng Thái Pod

Running	Tất cả container đã khởi động
Pending	Chờ scheduling hoặc pull image
CrashLoopBackOff	Container liên tục crash và khởi động lại
ImagePullBackOff	Không thể pull container image
Completed	Đã chạy xong (Jobs)

Deployments

Deployment YAML

```
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
```

Lệnh Deployment

kubectl get deploy	Liệt kê deployments
kubectl scale deploy web --replicas=5	Scale replicas
kubectl set image deploy/web web=nginx:1.28	Cập nhật image (rolling)
kubectl rollout status deploy/web	Theo dõi tiến trình rollout
kubectl rollout undo deploy/web	Rollback về phiên bản trước
kubectl rollout history deploy/web	Xem lịch sử phiên bản

Services

Loại Service

ClusterIP	Chỉ nội bộ (mặc định)
NodePort	Expose trên IP mỗi node tại port cố định
LoadBalancer	Load balancer bên ngoài (cloud)
ExternalName	Alias DNS đến service bên ngoài

Service YAML

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

Expose Nhanh

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

ConfigMaps & Secrets

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
```

Dùng Trong Pods

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

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

Lệnh

kubectl get cm	Liệt kê ConfigMaps
kubectl get secret	Liệt kê Secrets
kubectl describe cm app-cfg	Hiển thị dữ liệu ConfigMap
kubectl get secret db-creds -o yaml	Hiển thị Secret (mã hóa base64)

Namespaces

Lệnh Namespace

kubectl get ns	Liệt kê namespaces
kubectl create ns staging	Tạo namespace
kubectl delete ns staging	Xóa namespace và tất cả tài nguyên
kubectl get pods -n staging	Liệt kê pods trong namespace
kubectl get pods -A	Liệt kê pods trên tất cả namespaces

Đặt Namespace Mặc Định

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

Volumes

PersistentVolumeClaim

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

Mount Trong Pod

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

Loại Volume

emptyDir	Thư mục tạm, bị xóa cùng pod
hostPath	Mount đường dẫn filesystem của host
persistentVolumeClaim	Storage bền vững (PVC)
configMap	Mount ConfigMap dưới dạng file
secret	Mount Secret dưới dạng file

Tham Khảo Nhanh Kubernetes

Ingress

Ingress YAML

```
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 }
```

Ghi Chú Ingress

Ingress Controller	Bắt buộc (nginx-ingress, traefik, v.v.)
pathType: Prefix	Khớp tiền tố URL
pathType: Exact	Khớp chính xác đường dẫn URL
TLS	Thêm phần tls : với tên secret

Debugging

Lệnh Chẩn Đoán

kubectl logs <pod>	stdout/stderr của container
kubectl logs <pod> -c <ctr>	Log của container cụ thể
kubectl logs <pod> --previous	Log từ container bị crash
kubectl describe pod <pod>	Events, điều kiện, trạng thái
kubectl exec -it <pod> -- sh	Shell vào container
kubectl port-forward <pod> 8080:80	Chuyển tiếp port cục bộ đến pod
kubectl top pods	Sử dụng CPU/memory (metrics-server)
kubectl get events --sort-by=.lastTimestamp	Dòng thời gian sự kiện cluster

Debug Pod

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

Mẫu Phổ Biến

Labels & Selectors

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

Giới Hạn Tài Nguyên

```
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 }
```

Công Thức Nhanh

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 files	kubectl cp file.txt pod:/tmp/
Restart deploy	kubectl rollout restart deploy/web