

# Guia Completo de Kubernetes - Teoria, Prática e Preparação para CKA

# 1. Fundamentos do Kubernetes

Kubernetes é um orquestrador de contêineres que automatiza deploy, escalonamento, monitoramento e gerenciamento de aplicações em contêineres.

## Arquitetura

- **Control Plane**: responsável por tomar decisões globais (kube-apiserver, etcd, scheduler, controller-manager).
- **Node**: executa os contêineres (kubelet, kube-proxy).

## Componentes

- **kube-apiserver**: interface de comunicação com o cluster.
- **etcd**: banco de dados chave-valor que guarda o estado do cluster.
- **kubelet**: agente que garante a execução dos Pods nos Nodes.
- **kube-proxy**: gerencia o roteamento de rede nos Nodes.

![[Cluster Arquitetura]](<https://kubernetes.io/images/docs/components-of-kubernetes.svg>)

## 2. Gerenciamento de Objetos

### Pods

Unidade mínima executável que agrupa um ou mais contêineres.

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
  name: nginx-pod
```

```
spec:
```

```
  containers:
```

```
    - name: nginx
```

```
      image: nginx
```

### ReplicaSet

Garante que um número específico de Pods esteja sempre rodando.

### Deployment

Gerencia ReplicaSets, permite atualizações declarativas e rollback.

### Services

Exposição de Pods de forma estável via ClusterIP, NodePort ou LoadBalancer.

### ConfigMaps e Secrets

- ConfigMap: configs não sensíveis.

- Secret: informações sensíveis (base64).

### Namespaces e Labels

Organizam e categorizam recursos para gestão isolada.

### 3. Volumes e Persistência

Volumes

Usados para armazenar dados persistentes.

PersistentVolume (PV) e PersistentVolumeClaim (PVC)

PV é o recurso de armazenamento.

PVC é a solicitação feita pelo usuário.

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: exemplo-pvc

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 1Gi

StatefulSet

Gerencia Pods com identidade estável e volumes persistentes.

## 4. Escalabilidade

Horizontal Pod Autoscaler (HPA)

Escala automaticamente o número de Pods com base no uso de CPU/memória.

```
apiVersion: autoscaling/v2
```

```
kind: HorizontalPodAutoscaler
```

```
...
```

Vertical Pod Autoscaler (VPA)

Ajusta recursos de CPU e memória de cada Pod.

Cluster Autoscaler

Ajusta a quantidade de nodes no cluster.

## 5. Rede

Service Discovery

- DNS interno para comunicação entre Pods e Services.

Ingress

Controla acesso HTTP externo baseado em regras.

```
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
```

```
...
```

NetworkPolicy

Define regras de comunicação entre Pods com base em labels.

## 6. Segurança

### RBAC

Controle de acesso baseado em papéis.

### Roles e RoleBindings

Permissões específicas por namespace.

### Secrets

Gerenciamento seguro de senhas, tokens, certificados.

### ServiceAccounts

Identidade para aplicações acessarem a API do cluster.

## 7. Diagnóstico e Monitoramento

### Probes

- **Liveness**: reinicia container com problema.
- **Readiness**: indica se o container pode receber tráfego.
- **Startup**: usada para apps que demoram a subir.

### Ferramentas

- `kubectl logs`, `kubectl describe`, `kubectl exec`
- `k9s`, `stern`, `metrics-server`



## 8. Deploys e Ciclo de Vida

Estratégias de Deploy

- RollingUpdate
- Recreate

DaemonSet

Executa um Pod por Node.

Jobs e CronJobs

Executam tarefas únicas ou agendadas.

## 9. Helm e GitOps

Helm

Gerenciador de pacotes com Charts e templates.

bash

```
helm install nome ./meu-chart -f values.yaml
```

ArgoCD

Ferramenta GitOps para deploy automático baseado em Git.

```
apiVersion: argoproj.io/v1alpha1
```

```
kind: Application
```

```
...
```

## 10. Extensões e CSI

CSI Drivers

Montagem de volumes externos ou segredos.

# Azure Key Vault

Segredos são montados em tempo de execução no Pod.

# Dynatrace OneAgent

Binários injetados automaticamente por CSI.

CRDs e Operators

Estendem a API com recursos personalizados.

## 11. Gerenciamento de Cluster

Ferramentas

- kubeadm
- minikube
- kind

Backup e Restore do etcd

Ferramenta oficial para preservar o estado do cluster.

Atualizações

Recomenda-se testar upgrade em ambiente de homologação antes da produção.

## 12. Avaliação CKA e Simulado

### Requisitos da Prova

- 2 horas
- Open Book (documentação oficial permitida)
- 66% de acerto

### Dicas

- Use `kubectl explain`, `kubectl -h`
- Decore atalhos e estruturas YAML
- Pratique muito com minikube/kind