

kubectl 快速入门

kubectl 是 Kubernetes 的命令行工具。Kubernetes 是一个开源的容器编排系统，用于自动化部署、扩展和管理容器化应用程序。

1. 安装 kubectl

命令	说明
<code>kubectl cluster-info</code>	显示 Kubernetes 集群信息
<code>kubectl get nodes</code>	显示集群中的节点
<code>kubectl describe node <node-name></code>	显示节点的详细信息
<code>kubectl version</code>	显示 kubectl 的版本信息

2. 创建 Namespace

命令	说明
<code>kubectl get ns</code>	显示集群中的 Namespace
<code>kubectl create ns <namespace></code>	创建新的 Namespace
<code>kubectl delete ns <namespace></code>	删除 Namespace
<code>kubectl config set-context --current --namespace=<namespace></code>	将当前上下文切换到指定的 Namespace

3. Pod 管理

命令	说明
<code>kubectl get pods</code>	显示集群中的 Pod
<code>kubectl get pods -A</code>	显示所有 Namespace 中的 Pod
<code>kubectl describe pod <pod-name></code>	显示 Pod 的详细信息

명령어	설명
<code>kubectl logs <pod-name></code>	Pod 로그 보기
<code>kubectl logs -f <pod-name></code>	Pod 로그 실시간 모니터링 (tail -f)
<code>kubectl exec -it <pod-name> -- /bin/sh</code>	Pod 내부에 Shell 접속
<code>kubectl delete pod <pod-name></code>	Pod 삭제
<code>kubectl port-forward <pod-name> <local-port>:<pod-port></code>	Pod 포트와 로컬 포트 연결

4. Deployment

명령어	설명
<code>kubectl get deployments</code>	Deployments 목록 보기
<code>kubectl describe deployment <deployment-name></code>	Deployment 상세 정보 보기
<code>kubectl apply -f deployment.yaml</code>	Deployment YAML 파일 적용
<code>kubectl delete deployment <deployment-name></code>	Deployment 삭제
<code>kubectl scale deployment <deployment-name> --replicas=3</code>	Pod 수 조정
<code>kubectl rollout status deployment <deployment-name></code>	Deployment 상태 확인
<code>kubectl rollout undo deployment <deployment-name></code>	Deployment 롤백

5. Service

명령어	설명
<code>kubectl get svc</code>	Service 목록 보기
<code>kubectl describe svc <service-name></code>	Service 상세 정보 보기
<code>kubectl expose deployment <deployment-name> --type=NodePort --port=80</code>	Service 생성 (NodePort 타입)
<code>kubectl delete svc <service-name></code>	Service 삭제

6. ConfigMap & Secret

命令	作用
kubectl get configmaps	显示 ConfigMap
kubectl create configmap <name> --from-file=<file>	创建 ConfigMap
kubectl get secrets	显示 Secrets
kubectl create secret generic <name> --from-literal=key=value	创建 Secret

7. 集群管理

命令	作用
kubectl top pod	显示 Pod 资源使用
kubectl top node	显示 Node 资源使用
kubectl get events --sort-by=.metadata.creationTimestamp	显示事件
kubectl cp <pod-name>:<pod-path> <local-path>	从 Pod 复制文件
kubectl run debug-pod --image=busybox --restart=Never --rm -it --sh	在 Pod 中运行调试容器

8. 部署管理

命令	作用
kubectl api-resources	显示 Kubernetes API 资源
kubectl explain <resource>	显示 YAML 资源说明
kubectl edit <resource> <name>	编辑资源 (例如: kubectl edit deployment nginx)
kubectl apply -f <directory>	应用 YAML 文件
kubectl wait --for=condition=Ready pod/<pod-name>	等待 Pod 就绪

kubectl logs 查看日志

kubectl logs 查看 Kubernetes 容器的日志



1. Pod

```
kubectl logs <pod-name>
```

2. Pod



```
kubectl logs <pod-name> -c <container-name>
```



```
kubectl logs -f <pod-name>
```



```
kubectl logs --timestamps <pod-name>
```



N



```
kubectl logs --tail=20 <pod-name> # 20
```



```
kubectl logs --since=1h <pod-name> # 1
```

```
kubectl logs --since-time="2023-01-02T15:04:05Z" <pod-name>
```



```
kubectl logs -p <pod-name>
```



1. Deployment Pod

```
kubectl logs deploy/<deployment-name>
```

2. Pod

```
kubectl logs -l app=nginx
```

3.

```
kubectl logs <pod-name> > pod.log
```

4. Shell

```
kubectl exec -it <pod-name> -- cat /var/log/myapp.log
```



1.

```
kubectl logs <pod-name> | highlight
```

2.

```
kubectl logs <pod-name> | grep "ERROR"
```

3.

kubectl logs <pod-name> --all-containers=true

4. **JSON**

kubectl logs <pod-name> -f --tail=10 | jq



1. Pod
2. Pod
3. Kubernetes
4.



Pod
kubectl describe pod <pod-name>

kubectl get pod <pod-name> -o jsonpath='{.status.containerStatuses[0].ready}'

Kubernetes



- **kubectl get**
- **kubectl describe**
- **kubectl logs**
- **kubectl exec** Pod
- **kubectl apply/delete** /
- **kubectl rollout** Deployment

Kubernetes