

Kubernetes Command Cheat Sheet & Node.js Deployment Guide

Cluster Basics

```
kubectl version
kubectl cluster-info
kubectl config view
kubectl get nodes
kubectl describe node <node-name>
```

Pod Management

```
kubectl run <pod-name> --image=<image>
kubectl get pods
kubectl describe pod <pod-name>
kubectl delete pod <pod-name>
kubectl logs <pod-name>
kubectl exec -it <pod-name> -- /bin/bash
```

From YAML:

```
kubectl apply -f pod.yaml
kubectl delete -f pod.yaml
```

Deployments & ReplicaSets

```
kubectl create deployment <name> --image=<image>
kubectl get deployments
kubectl describe deployment <name>
kubectl scale deployment <name> --replicas=<num>
kubectl rollout status deployment <name>
kubectl rollout undo deployment <name>
kubectl delete deployment <name>
```

Services

```
kubectl expose pod <pod-name> --port=80 --type=NodePort
kubectl expose deployment <deployment-name> --type=LoadBalancer --port=80
kubectl get services
kubectl describe svc <svc-name>
kubectl delete svc <svc-name>
```

Namespaces

```
kubectl get namespaces
kubectl create namespace <name>
kubectl delete namespace <name>
kubectl get pods -n <namespace>
kubectl run <pod> --image=nginx -n <namespace>
```

Kubernetes Command Cheat Sheet & Node.js Deployment Guide

ConfigMaps & Secrets

```
kubectl create configmap <name> --from-literal=key=value
kubectl get configmaps
kubectl describe configmap <name>

kubectl create secret generic <name> --from-literal=password=mypassword
kubectl get secrets
kubectl describe secret <name>
kubectl get secret <name> -o yaml
```

Volumes

```
emptyDir example:
volumes:
  - name: myvol
    emptyDir: {}

volumeMounts:
  - name: myvol
    mountPath: /data
```

Probes

```
livenessProbe:
  httpGet:
    path: /health
    port: 3000
  initialDelaySeconds: 5
  periodSeconds: 10
```

Node Scheduling

```
Taints:
kubectl taint nodes <node> key=value:NoSchedule
kubectl describe node <node>
```

```
Tolerations:
tolerations:
  - key: "key"
    operator: "Equal"
    value: "value"
    effect: "NoSchedule"
```

```
Node Affinity:
affinity:
  nodeAffinity:
```

Kubernetes Command Cheat Sheet & Node.js Deployment Guide

```
requiredDuringSchedulingIgnoredDuringExecution:
  nodeSelectorTerms:
  - matchExpressions:
    - key: disktype
      operator: In
      values:
      - ssd
```

RBAC

```
kubectl create role devreader --verb=get,list --resource=pods -n dev
kubectl create rolebinding devbind --role=devreader --user=<user> -n dev
kubectl get roles -n dev
kubectl get rolebindings -n dev
```

Helm

```
helm repo add bitnami https://charts.bitnami.com/bitnami
helm search repo nginx
helm install mynginx bitnami/nginx
helm list
helm upgrade mynginx bitnami/nginx
helm uninstall mynginx
```

Imperative Object Creation

```
kubectl create configmap myconf --from-file=app.config
kubectl create secret generic mysecret --from-env-file=secret.env
kubectl create deployment myapp --image=myimage
kubectl expose deployment myapp --port=8080 --target-port=8080
kubectl set image deployment/myapp myapp=myimage:v2
```

Deploying a Node.js App to Kubernetes

```
1. server.js:
const express = require('express');
const app = express();
const PORT = 3000;
app.get('/', (req, res) => res.send('Hello from Kubernetes!'));
app.listen(PORT, () => console.log(`Listening on port ${PORT}`));
```

```
2. Dockerfile:
FROM node:18
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
```

Kubernetes Command Cheat Sheet & Node.js Deployment Guide

EXPOSE 3000

CMD ["node", "server.js"]

3. Build & Push:

docker build -t <your-dockerhub-username>/node-kube-app .

docker push <your-dockerhub-username>/node-kube-app

4. node-deployment.yaml:

apiVersion: apps/v1

kind: Deployment

metadata:

name: node-kube-app

spec:

replicas: 2

selector:

matchLabels:

app: node-kube-app

template:

metadata:

labels:

app: node-kube-app

spec:

containers:

- name: node-kube-container

image: <your-dockerhub-username>/node-kube-app

ports:

- containerPort: 3000

5. node-service.yaml:

apiVersion: v1

kind: Service

metadata:

name: node-kube-service

spec:

selector:

app: node-kube-app

ports:

- protocol: TCP

port: 80

targetPort: 3000

type: NodePort

6. Deploy:

kubectl apply -f node-deployment.yaml

kubectl apply -f node-service.yaml

kubectl get pods

kubectl get svc