Docker & Kubernetes Deployment Cheat Sheet

🐳 Docker Commands

Build & Tag Images

```
# Build image
docker build -t image-name:tag .
docker build -t my-app:latest .

# Tag existing image
docker tag old-name:tag new-name:tag
docker tag my-app:latest username/my-app:latest

# Build and tag in one command
docker build -t username/my-app:latest .
```

Manage Images

```
# List images
docker images
docker images | grep my-app

# Remove image
docker rmi image-name:tag

# Pull image
docker pull username/my-app:latest

# Push to Docker Hub
docker push username/my-app:latest
```

Run Containers

```
# Run container
docker run -p host-port:container-port image-name
docker run -p 3000:3000 my-app:latest

# Run in detached mode
docker run -d -p 3000:3000 my-app:latest

# Run with environment variables
docker run -p 3000:3000 -e NODE_ENV=production my-app:latest
```

Kubernetes Commands

Minikube Management

```
# Start minikube
minikube start

# Check status
minikube status

# Stop minikube
minikube stop

# Delete minikube cluster
minikube delete

# Get service URL
minikube service service-name --url
minikube service my-app-service --url
```

Apply & Delete Resources

```
# Apply single file
kubectl apply -f deployment.yaml
```

```
kubectl apply -f service.yaml

# Apply all files in directory
kubectl apply -f k8s/
kubectl apply -f .

# Delete resources
kubectl delete -f deployment.yaml
kubectl delete deployment my-app
kubectl delete service my-app-service
```

View Resources

```
# Get all resources
kubectl get all

# Get specific resources
kubectl get pods
kubectl get services
kubectl get deployments
kubectl get replicasets

# Get detailed info
kubectl get pods -o wide
kubectl get services -o wide

# Describe resources
kubectl describe pod pod-name
kubectl describe service service-name
kubectl describe deployment deployment-name
```

Logs & Debugging

```
# View logs
kubectl logs pod-name
kubectl logs deployment/deployment-name
```

```
kubectl logs -f pod-name  # Follow logs

# Execute commands in pod
kubectl exec -it pod-name -- /bin/bash
kubectl exec -it pod-name -- sh

# Port forwarding
kubectl port-forward pod/pod-name local-port:pod-port
kubectl port-forward service/service-name local-port:service-port
kubectl port-forward service/my-app-service 8080:3000
```

Scale Applications

```
# Scale deployment
kubectl scale deployment deployment-name --replicas=5
kubectl scale deployment my-app --replicas=3

# Check scaling
kubectl get deployments
```

📝 YAML File Examples

Deployment (deployment.yaml)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
  labels:
    app: my-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-app
  template:
```

```
  metadata:
    labels:
      app: my-app
  spec:
    containers:
    - name: my-app
      image: username/my-app:latest
      ports:
      - containerPort: 3000
      env:
      - name: NODE_ENV
        value: "production"
      resources:
        requests:
          cpu: "100m"
          memory: "128Mi"
        limits:
          cpu: "500m"
          memory: "512Mi"
```

Service (service.yaml)

```
apiVersion: v1
kind: Service
metadata:
  name: my-app-service
  labels:
    app: my-app
spec:
  selector:
    app: my-app
  ports:
  - protocol: TCP
    port: 3000
    targetPort: 3000
  type: NodePort  # or LoadBalancer for cloud
```

🚀 Deployment Script (deploy.sh)

```
#!/bin/bash
set -e

NAME="my-app"
USERNAME="your-username"
IMAGE="$USERNAME/$NAME:latest"

echo "Building Docker image..."
docker build -t $IMAGE .

echo "Pushing to Docker Hub..."
docker push $IMAGE

echo "Applying Kubernetes manifests..."
kubectl apply -f k8s/deployment.yaml
kubectl apply -f k8s/service.yaml

echo "Getting status..."
kubectl get pods
kubectl get services

echo "Getting service URL..."
minikube service my-app-service --url
```

Make script executable and run:

```
chmod +x deploy.sh
./deploy.sh
# or
npm run deploy  # if added to package.json
```

🛠️ Troubleshooting Commands

Check Pod Issues

```
# Describe pod for events
kubectl describe pod pod-name

# Check logs
kubectl logs pod-name

# Get events
kubectl get events --sort-by=.metadata.creationTimestamp

# Check resource usage
kubectl top pods  # Requires metrics-server
```

Common Issues & Solutions

```
# Image pull errors - check image name
kubectl describe pod pod-name

# Service not accessible - check labels match
kubectl get pods --show-labels
kubectl describe service service-name

# Pod stuck in Pending - check resources
kubectl describe pod pod-name

# Restart deployment
kubectl rollout restart deployment/deployment-name
```

📦 Package.json Scripts

```
{
  "scripts": {
    "dev": "node --watch index.js",
```

```
    "start": "node index.js",
    "docker:build": "docker build -t my-app:latest .",
    "docker:run": "docker run -p 3000:3000 my-app:latest",
    "k8s:deploy": "kubectl apply -f k8s/",
    "k8s:delete": "kubectl delete -f k8s/",
    "deploy": "sh deploy.sh"
  }
}
```

🔄 Complete Workflow

1. Develop your application
2. Build Docker image: `docker build -t username/app:latest .`
3. Push to registry: `docker push username/app:latest`
4. Deploy to K8s: `kubectl apply -f k8s/`
5. Access service: `minikube service app-service --url`
6. Monitor: `kubectl get pods`, `kubectl logs pod-name`
7. Update: Modify code → Build → Push → `kubectl rollout restart deployment/app`

🎯 Quick Commands Summary

```
# Full deployment pipeline
docker build -t username/app:latest . && \
docker push username/app:latest && \
kubectl apply -f k8s/ && \
kubectl get pods && \
minikube service app-service --url

# Quick status check
kubectl get all

# Quick cleanup
kubectl delete -f k8s/

# Emergency restart
```

```
kubectl rollout restart deployment/app-name
```