kubernetes Cheatsheet

version 2020-11-15 EN

infraBuilder
infrastructure devops

CLI tools

```
# CLI setup
https://ibd.sh/kubectl
https://ibd.sh/kubectx
https://ibd.sh/kubetail
https://ibd.sh/helm
```

contexts

```
# List available contexts
kubectx

# Change context to dev
kubectx dev
```

namespaces

```
# List namespaces
kubens

# Change namespace to prod
kubens prod

# Create namespace test
kubectl create ns test
```

ingress

```
# Ingress manifest example
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: test-ingress
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /testpath
        backend:
          service:
            name: test
            port:
              number: 80
  tls:
  - hosts:
    - foo.bar.com
    secretName: foobar-tls
```

deploy / expose

```
# Create a deployment named web, using image nginx into prod namespace
kubectl create -n prod deploy web --image=nginx

# Expose port 80 of deployment web with an internal service named front
kubectl expose deploy/web --port=80 --name=front

# Retrieve logs of pods with tag app=web
kubetail -l app=web

# Open a tunnel listening on 127.0.0.1:8080 to the port 80 of a pod related to deployment web
kubectl port-forward deploy/web 8080:80

# Create a Yaml manifest, without sending it to the cluster
kubectl create deploy web --image=nginx --dry-run=client -o yaml > web.yml

# Edit deployment web
kubectl edit deploy/web
```

help / debug

```
# Retrieve detailed state of pod test
kubectl describe pod test

# Get all possible attributes of a resource
kubectl explain pod --recursive

# Open a bash terminal in pod app
kubectl exec -it app -- bash

# NB : The flag --help provide help of any command
```

storage / volume

```
# PVC manifest example
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: web-data
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 42Gi
```

configuration

```
# Use the config file /path/to/config rather than ~/.kube/config
export KUBECONFIG=/path/to/config

# Merge two configuration files config1 and config2 in one file config
KUBECONFIG=config1:config2 kubectl config view --flatten > config

# Export only the current context configuration to file config
kubectl config view --minify --flatten > config
```

Command cheasheet
forr Kubernetes 1.19
by Alexis DUCASTEL
Web : infraBuilder.com