Table of Contents

https://K3S.io

kubectl alias to microk8s

```
sudo snap alias microk8s.kubectl kubectl
sudo usermod -aG microk8s $(whoami)
sudo microk8s enable dashboard
sudo microk8s enable dns
sudo microk8s enable registry
sudo microk8s enable ingress
```

access dashboard

```
microk8s dashboard-proxy
```

create namespace

```
kubectl create namespace test
```

Show kubernetes

microk8s kubectl get pod -o wide

Show yaml

microk8s kubectl get deployment nginx-deployment -o yaml

Show services

microk8s kubectl get service

Install istio ingress

```
microk8s enable community
microk8s enable istio
```

```
kubectl create namespace demo
kubectl label namespace demo istio-injection=enabled
```

test it

```
kubectl apply -f
https://raw.githubusercontent.com/istio/istio/release-1.16/samples/bookinfo/platform/kube/boo
kinfo.yaml -n demo
```

Install NGINX ingress

```
sudo microk8s enable ingress
sudo kubectl delete -A ValidatingWebhookConfiguration ingress-nginx-admission
```

test.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: test-ingress
spec:
  rules:
  - http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: [service-name]
            port:
              number: [service-port]
```

```
sudo kubectl apply -f test.yaml
kubectl expose deployment/nginx-deployment --type="NodePort" --port 80 --namespace=default
```

Deploy nginx

Create ConfigMap with HTML

```
apiVersion: v1
data:
  index.html: |
    <!DOCTYPE html>
    <html lang="en">
    <head>
        <title>A simple HTML document</title>
    </head>
    <body>
        <p>This is Deployment One!<p>
    </body>
    </html>
kind: ConfigMap
metadata:
  name: my-config1
  namespace: default
```

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

```
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.23.3
          ports:
            - containerPort: 80
          volumeMounts:
            - name: nginx-config
              mountPath: /usr/share/nginx/html #nginx specific
      volumes:
        - name: nginx-config
          configMap:
            name: my-config1
```

Create NGINX with custom conf

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: nginx-conf
data:
  nginx.conf: |
    user nginx;
    worker_processes  1;
    events {
      worker_connections  10240;
    }
    http {
      server {
          listen       80;
          server_name  localhost;
          location / {
            root   html;
            index  index.html index.htm;
        }
      }
    }

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
```

```
        - containerPort: 80
        volumeMounts:
            - name: nginx-conf
              mountPath: /etc/nginx/nginx.conf
              subPath: nginx.conf
              readOnly: true
      volumes:
      - name: nginx-conf
        configMap:
          name: nginx-conf
          items:
            - key: nginx.conf
              path: nginx.conf

---
apiVersion: v1
kind: Service
metadata:
  name: nginx
spec:
  type: NodePort
  ports:
  - port: 80
    protocol: TCP
    targetPort: 80
    nodePort: 30008
  selector:
    app: nginx
```

Proxmox / LXC

```
# Allow running Docker inside LXC
lxc.aa_profile = unconfined
lxc.cap.drop = mac_override sys_time sys_module sys_rawio
```

Note: this affect security!

Docker

Data directory

```
/var/lib/docker
/var/lib/docker/vfs/dir
/var/lib/docker/volumes
```

Show info

```
docker info
```

Get image from repository

```
docker pull centos
```

Run image in background

```
docker run -d -e VARIABLE=CONTENT -v /sys/fs/cgroup:/sys/fs/cgroup:ro -p 222:22 centos
--rm remove after
--name INSTANCE_NAME
-it interactive
```

Show running / images / stats

```
docker ps
docker ps -a
docker images
docker stats {container}
```

Return info about container in JSON

```
docker inspect {container}
```

Enter / Stop / Start Container

```
docker attach {container}
docker stop {container}
docker start {container}
```

Map directory

```
-v /home/user:/home/user:ro -u 500:500
```

From:
https://wiki.janforman.com/ - **wiki.janforman.com**

Permanent link:
https://wiki.janforman.com/kubernetes

Last update: **2025/03/15 18:40**