

SIMPLE SITE HOSTING WITH LIGHTWEIGHT KUBERNETES

FEAT. MICROK8S, LETSENCRYPT, AND NGINX INGRESS

JAMES PANNACCIULLI

Kubernetes is a wildly extensible, configurable, enterprise grade orchestration platform which drives many of the largest presences on the web.

LET'S USE IT TO HOST A FEW SMALL STATIC SITES.

ADVANTAGES OF KUBERNETES

Extensibility

Each site runs whatever servers, configs, and apps needed in its own container
Containers can all share an IP via Ingress routing

Scalability

Increasing the number of replicas for a deployment is a one line change
If host resources are running low, more nodes can be added to the cluster

Automation

Automated LetsEncrypt certificate provisioning and updates

Resilience

Containers are automatically restarted in case of failure

Experience

Setting up static sites is a relatively easy way to get experience with Kubernetes

The ~~problem~~ solution space is

VAST

MICROK8S

Addons

Easily add common applications to your Kubernetes cluster

Automatic Updates

The MicroK8s snap package will automatically update for minor version releases

CNCF Compliant

Guarantees standard open source Kubernetes software stack

Ensures expected, versioned API compatibility

Simplicity

Easy to install, no special configuration needed

MICROK8S INSTALLATION

```
# Install snapd
sudo apt install snapd

# Install the latest microk8s snap
# (can be made explicit with --channel=latest/stable
# or use specific channel from `snap info microk8s`)
sudo snap install microk8s --classic

# Add your user to the microk8s group
usermod -a G microk8s $USER
newgrp microk8s
```

MICROK8S CONFIGURATION

```
# Check MicroK8s status and list available/enabled addons  
microk8s status
```

```
# Install the ingress and cert-manager addons  
microk8s enable ingress cert-manager
```

PUT A SIMPLE SITE IN AN NGINX CONTAINER

Or any container, but if you want transparent logging, redirect to stdout and stderr, which the default nginx container does for you!

```
FROM nginx  
COPY html /usr/share/nginx/html
```

BUILD AND TAG YOUR DOCKER IMAGE WITH A FAKE REGISTRY FOR SIDELOADING:

```
# build / tag
sudo docker build -t \
  microk8s.sideload/seagl00.k8s.jp.net:v000 seagl00/

# sideload import to microk8s
sudo docker save microk8s.sideload/seagl00.k8s.jp.net:v000 | \
  microk8s image import

# list images
microk8s ctr images list
```

EDIT SOME YAML AND APPLY!

Grab example yamls from [the example repo](#)

```
microk8s kubectl apply -f ./seagl00.yaml  
microk8s kubectl apply -f ./seagl01.yaml  
microk8s kubectl apply -f ./ingress.yaml
```

Same command with `replace` instead of `apply` for updates!

THANKS

Slides:

[JS seagl2023.slides.k8s.jp.net]

[PDF talk.jpnc.info/k8s-simple-sites.jpnc.seagl2023.pdf]

Example code:

git.jpnc.info/simple__sites.k8s.examples/tree/seagl2023

Example sites:

seagl00.k8s.jp.net seagl01.k8s.jp.net

FIND ME AT JPNC.INFO