

Day 9 Exercises

Exercise 1

Time: 20 mins.

1. Download and install Kubernetes
2. Enable dashboard
3. Obtain token
4. Login in dashboard

Answer

1. Goto the <https://microk8s.io/docs/install-alternatives> site or follow your instructor indications and install microk8s
2. Review that the service is started
3. use `microk8s start` if not or if you are in doubt
4. `microk8s enable dashboard` (in linux use `sudo`)
5. `microk8s dashboard-proxy`
6. copy the token
7. open the fiven URL as part of the response of the last command
8. If needed accept to continue due the self signed certificated
9. use the token value to login

Exercise 2

Time: 15 mins.

Creating your first service

1. List the active services for the system namespace
2. Create a service using the docker getting-started image
3. Make the service to expose a port to reach it
4. Create 3 pods on your services

Answer

```
microk8s kubectl get services --namespace kube-system
```

kube-system

Create from input Create from file Create from form

Enter YAML or JSON content specifying the resources to create to the currently selected namespace. [Learn more](#)

1

kube-public

Create from input Create from file Create from form

App name *
hello-minikube

Container image *
docker/getting-started

Number of pods *
3

Service *
External

Port * 3030 Target port * 80 Protocol * TCP

Port Target port Protocol *

Deploy Cancel Show advanced options

An 'app' label with this value will be added to the Deployment and Service that get deployed. [Learn more](#)

Enter the URL of a public image on any registry, or a private image hosted on Docker Hub or Google Container Registry. [Learn more](#)

A Deployment will be created to maintain the desired number of pods across your cluster. [Learn more](#)

Optionally, an internal or external Service can be defined to map an incoming Port to a target Port seen by the container. [Learn more](#)

10.152.183.30:3030/tutorial/

Most Visited Bienvenidos a Beisbó... Virtual University Packages Search - pk... Welcom

docker Labs Getting Started

Getting Started

kubernetes kube-system

Discovery and Load Balancing > Services

Workloads Cron Jobs Daemon Sets Deployments

Services

Name	Labels	Type	Cluster IP	Internal Endpoints
hello-minikube	k8s-app: hello-minikube	LoadBalancer	10.152.183.30	hello-minikube.kube-system:3030 TCP hello-minikube.kube-system:30911 TCP

Exercise 3

Time 60 min

1. Create the followin Dockerfile in your exercise directory

```
FROM alpine as HUGO
```

```
ENV HUGO_VERSION="0.81.0"
```

```
RUN apk add --update wget
```

```
# Install Hugo.
```

```
RUN wget --quiet https://github.com/gohugoio/hugo/releases/download/v${HUGO_VERSION}/  
hugo_${HUGO_VERSION}_Linux-64bit.tar.gz && \
```

```
tar -xf hugo_${HUGO_VERSION}_Linux-64bit.tar.gz && \
```

```
mv hugo /usr/local/bin/hugo && \
```

```
rm -rf hugo_${HUGO_VERSION}_Linux-64bit.tar.gz
```

```
COPY . /hugo-site
```

```
# Use Hugo to build the static site files.
```

```
RUN hugo -v --source=/hugo-site --destination=/hugo-site/public
```

```
FROM bitnami/nginx:latest
```

```
COPY --from=HUGO /hugo-site/public/ /opt/bitnami/nginx/html/
```

```
# The container will listen on port 8080 (non-privileged) using the TCP protocol.
```

```
EXPOSE 8080
```

2. build the docker image
3. review that the image is recognized by docker
4. Run instance in docker
5. Review the website is running

6. create the following deployment.yaml file

apiVersion: apps/v1

kind: Deployment

metadata:

name: hugo-site

labels:

app: hugo-site

tier: frontend

spec:

replicas: 2

selector:

matchLabels:

app: hugo-site

tier: frontend

template:

metadata:

labels:

app: hugo-site

tier: frontend

spec:

containers:

- name: hugosite

image: eduk8s-labs-w08-s119-registry.kube-prod-blue5-bf4a136.kubeacademy.esp.vmware.com/

hugo-site:v1

securityContext:

runAsNonRoot: true

ports:

- containerPort: 8080

7. create the kubernetes deployment

8. create the service using service.yaml

apiVersion: v1

kind: Service

metadata:

name: hugo-site

labels:

app: hugo-site

tier: frontend

spec:

type: ClusterIP

ports:

- protocol: TCP

port: 80

targetPort: 8080

selector:

app: hugo-site

tier: frontend

9. Create the service

10. list services

11. Access your site

12. List your endpoint

Answer

docker build -t hugo-site:v1 .

Docker images hugo-site:v1

docker run -d --rm --name hugo -p 8080:8080 hugo-site:v1

Go to the localhost:8080 on browser

Create the yaml file

microk8s kubectl apply -f deployment.yaml

Create service.yaml

```
microk8s kubectl apply -f service.yaml
```

```
microk8s kubectl get service
```

use the browser

```
microk8s kubectl get endpoints
```