



Kubernetes Tutorial

Kubernetes on OpenSuse Leap 15

Youssef Hbali(c) 2019

Version 1.0, 08.03.2019

Table of Contents

1. Summary	1
2. Docker	2
2.1. Docker installation	2
2.2. Start & check docker	2
2.3. Create an image	2
2.4. Build the image	2
2.5. Run and test the buit image	3
3. Kubernetes	4
3.1. Basic Architecture	4
3.2. Requirement : Virtual Box installation	4
3.3. Minikube	4
3.4. Kubectl	5
3.5. Starting Minikube	5

1. Summary

2. Docker

2.1. Docker installation

```
sudo zypper ar -cf \
http://download.opensuse.org/repositories/home:/janhebler:/kubi/openSUSE_Leap_15.0/ Docker

sudo zypper in docker
sudo zypper in docker-bash-completion
```

2.2. Start & check docker

```
sudo systemctl start docker
sudo systemctl status docker

# Add the current user to the docker group
sudo usermod -aG docker $USER

# Log out from the session or restart the OS in case you're using a VM

docker run hello-world
```

2.3. Create an image

```
touch Dockerfile

# Add
FROM node:7
ADD app.js /app.js
ENTRYPOINT ["node", "app.js"]

touch app.js

# Add
const http = require('http');
const os = require('os');
console.log("Kubia server starting...");
var handler = function(request, response) {
  console.log("Received request from " + request.connection.remoteAddress);
  response.writeHead(200);
  response.end("You've hit " + os.hostname() + "\n");
};
var www = http.createServer(handler);
www.listen(8080);
```

2.4. Build the image

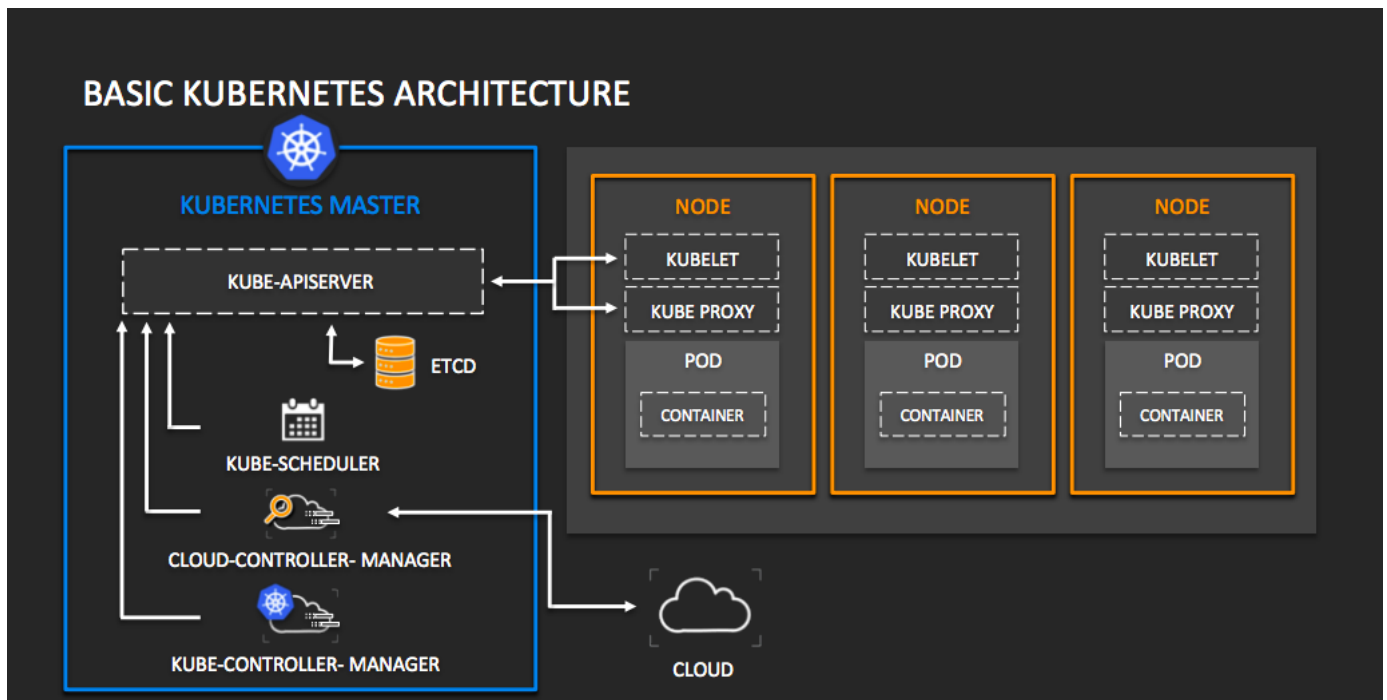
```
docker build -t image-name .
```

2.5. Run and test the built image

```
docker run --name container-name -p 8080:8080 -d image-name  
curl localhost:8080
```

3. Kubernetes

3.1. Basic Architecture



3.2. Requirement : Virtual Box installation

```
sudo zypper ar -cf \
http://download.opensuse.org/update/leap/15.0/oss oss

sudo zypper in virtualbox
sudo usermod -aG vboxusers $USER
# relog

sudo zypper install virtualbox-host-source kernel_devel
sudo /sbin/vboxconfig

sudo systemctl enable vboxdrv
sudo systemctl start vboxdrv
sudo systemctl status vboxdrv
```

3.3. Minikube

The simplest and quickest path to a fully functioning Kubernetes cluster is by using Minikube. Minikube is a tool that sets up a single-node cluster that's great for both testing Kubernetes and developing apps locally. Although we can't show certain Kubernetes features related to managing apps on multiple nodes, the single-node cluster should be enough for exploring most topics discussed in this book.

```
curl -Lo minikube https://storage.googleapis.com/minikube/releases/v0.35.0/minikube-linux-amd64 && chmod +x minikube  
&& sudo mv minikube /usr/local/bin/
```

3.4. Kubectl

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s  
https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl && chmod +x kubectl &&  
sudo mv kubectl /usr/local/bin/
```

3.5. Starting Minikube

```
youssef@tokyo:Kubernetes-training$ minikube start  
🐳 minikube v0.35.0 on linux (amd64)  
💡 Tip: Use 'minikube start -p <name>' to create a new cluster, or 'minikube delete' to delete this one.  
🔄 Restarting existing virtualbox VM for "minikube" ...  
⌚ Waiting for SSH access ...  
📶 "minikube" IP address is 192.168.99.100  
🔧 Configuring Docker as the container runtime ...  
⚙️ Preparing Kubernetes environment ...  
📦 Pulling images required by Kubernetes v1.13.4 ...  
🔄 Relaunching Kubernetes v1.13.4 using kubeadm ...  
⌚ Waiting for pods: apiserver proxy etcd scheduler controller addon-manager dns  
🔧 Updating kube-proxy configuration ...  
🏡 Verifying component health .....  
💕 kubectl is now configured to use "minikube"  
🎉 Done! Thank you for using minikube!
```