Workshop Kubernetes

Getting started

1- Setup environment

- Install Docker
- Install kubectl
- Execute: $ kubectl completion bash
- Install kubens + kubectx

2- Start a k8s cluster

You're going to build the project in the Cloud. But if you never used Kubernetes, starting locally with Minikube may be easier.

With minikube:

Minikube is a local K8s cluster, with a single node.

Some usefull commands:

Start cluster:

```
$ minikube start --vm-driver virtualbox \
        --extra-config=apiserver.service-node-port-range=1-30000

$ minikube status
$ minikube ip
$ kubectl config current-context
$ kubectl config use-context minikube
```

Delete cluster:

```
$ minikube stop
$ minikube delete
```

In the Cloud:

Most IaaS offer Kubernetes as a Service.

- Digital Ocean: Discount code: https://m.do.co/c/1670fd3e0af6 ($50, 30 days)
- GCP (300$ free tier)

3- Take a look on your cluster

- Can you list nodes ?
- Can you get nodes IPs ?

- Can you list namespaces ?
- Can you switch your current namespace ?

- Can you list pods ?
- Can you list pods of “kube-system” namespace ?

4- Run Forest, run!

Create a pod with the following Docker image: samber/hello-world-nodejs (https://hub.docker.com/r/samber/hello-world-nodejs).

- Can you list running pods ?
- Can you print more details about this new pod ?
  (print as much information as possible: ip, node, labels, uptime...)
- Can you print your container logs ?
- Can you execute the command “$ date” in this running container ?
- Can you delete this pod ?

5- Communicate

Environment

Update your pod configuration and add the following environment variable: “PORT=8080”.

- Run this pod
- Can you execute “$ env” the pod ? Do you see the PORT=8080 ?

Networking

Now, can you ask Kubernetes to expose this port to the cluster ?

The port won’t be accessible from outside the Kubernetes cluster. Can you forward pod port to localhost ? Then execute “$ curl localhost:8080”.

- Can you print full details of this pod ?
- Do you see IP/port ?
- Can you drop this pod and create it again ?
- What is the new IP/port ? Is it static ? Dynamic ?

Addressing

Using IP/port is never the right way to communicate between 2 services in an infrastructure.
Find a way to create an internal DNS for your hello-world container.

- What is the pattern of kubernetes internal DNS ?
- Can you print details about this DNS and check it is linked to your container ?

Start another container with “$ kubectl run” and curl your hello-world DNS.

If the server replies with “Hello world!”, congratulations, you’ve done it!
\o/

6- Persist

Can you attach a 512MB volume to your container ?

7- Deployments

Take a look at Kubernetes “deployments”. Do you see the difference with pods ?

- Can you run the “hello-world” app in a deployment ?

Scale up!

In order to handle more requests on your hello-world API, can you start 3 instances of the app ?

- Can you list pods ?
- Can you check if your DNS points to those 3 pods ?
- Can you print logs of those pods in 3 terminals ?
- When you send an HTTP request to the DNS (with curl), do you see it in logs ?

Upgrade

The developers did great work, can you now upgrade the Docker image to samber/hello-world-nodejs:v2 ?

- Can you list pods ?
- Can you print the deployment history ?
- Can you rollback to the previous version (from command line, only) ?
- Can you upgrade again, but 1 container at a time ?

Scheduling

Now, spread your 3 containers on 3 different Kubernetes nodes.

- Can you list pods ?
- What do you see when a single node has been added to the cluster ?

Can you reserve 1GB of memory to the container ?
Can you reserve some CPU ?