**Part 3:** **Assignment-3.3: Simple Microservices on Kubernetes**

**Minikube and kubectl**

* If you haven’t yet installed Minikube on your local machine, it’s time to do it now. Use Minikube Installation instructions from the Class Materials.
* Start Minikube with command:

$ minikube start

A picture containing indoor, wall

Description automatically generated

* Set VirtualBox as Hypervisor

$ minikube config set vm-driver virtualbox

* Configure *kubectl* to talk to the cluster called *minikube*

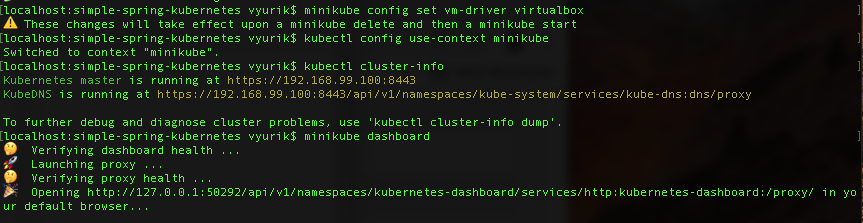
$ kubectl config use-context minikube

* Verify that kubectl properly interacts with the cluster

$ kubectl cluster-info

* Inspect the state o cluster with command:

$ minikube dashboard



**Microservices: simple-server and simple-client**

* You will develop two simple microservices. Let’s call them **simple-server** and **simple-client**.
* The **simple-server** provides the REST endpoint on port 8080, returning a *String* containing its hostname. The **simple-client** microservice provides the REST endpoint on port 8081. The **simple-client** microservice invokes the **simple-server** microservice endpoint and return its response.
* You will have to develop two simple classes. Let’s name them KubernetesSimpleServerApplication for the **simple-server** microservice and KubernetesSimpleClientApplication for the **simple-client** microservice.
* You will need to create one Dockerfile on the **simple-server** microservice and one Dockerfile on the **simple-client** microservice.
* If you have a directory structure as one shown below:

A screenshot of a cell phone

Description automatically generated

change to the **simple-spring-kubernetes** directory and run **mvn clean install.**

A screenshot of a computer

Description automatically generated

* To start the build process on the Docker host of the Minikube cluster you need to create SSH with command:

$ minikube ssh

* Change to **simple-spring-kubernetes/simple-server** directory and buildthe **simple-server** image from its **Dockerfile.**
* Change to **simple-spring-kubernetes/simple-client** directory and buildthe **simple-client** image from its **Dockerfile.**
* Exit the SSH shell using command

$ logout

* Now you have a choice of either using direct commands for deploying the microservices to Minikube or creating configuration files **simple-server.yaml** and **simple-client.yaml** for this purpose.

**Using direct commands for deploying the microservices to Minikube**

* Create a deployment named **simple-server** which is instantiated from image alsonamed **simple-server.**
* Create a deployment named **simple-client** which is instantiated from image alsonamed **simple-client.**
* Verify the deployments:

A screen shot of a clock

Description automatically generated

* Verify the pods:

A screen shot of a clock

Description automatically generated

* Create both services exposing the **simple-server** and **simple-client** deployments.
* Verify that the services were created

A clock mounted to the side

Description automatically generated

* Launch the **simple-server** service, that will open your default browser and will display the **simple-server** message.

$ minikube service simple-server

A screenshot of a cell phone

Description automatically generated

* Launch the **simple-client** service, that will open your default browser and will display the **simple-client** message.

$ minikube service simple-client

A screenshot of a social media post

Description automatically generated

* To visually observe what’s being done, launch the command

**$** minikube dashboard

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Using configuration files simple-server.yaml and simple-client.yaml for deploying microservices to Minikube**

* This approach is described in the **spring-cloud-kubernetes** application we’ve examined in class.
* Please refer to this application if you’ve selected this option.