TP Openwhisk

Goal: Deploy Openwhisk, define an action, configure a trigger and a rule binding the action and the trigger.

Connection

The first step is the connections to your servers which is similar to what you did with Kubernetes LabWork.

Kubernetes installation

This step is similar to the installation of Kubernetes. You should reinstall Kubernetes if not installed.

Openwisk installation

Once Kubernetes installed you have to install OpenWhisk using Helm (for more info about helm, you can search online.)

Helm installation

On the master node:

```
sudo bash
curl https://raw.githubusercontent.com/helm/helm/master/scripts/get > get_helm.sh
chmod +x get_helm.sh
./get_helm.sh
helm init
curl https://raw.githubusercontent.com/helm/helm/master/scripts/get > get_helm.sh
kubectl get pods -n kube-system
git clone https://github.com/apache/incubator-openwhisk-deploy-kube.git
cd incubator-openwhisk-deploy-kube/
```

You have to configure helm deployment files.

Modify the mycluster.yam (in the current directory) and set the apiHostName attribute with your master node IP address.

To get your master node Ip address your can use the command Hostname -I

Modify the file ./helm/openwisk/Chart.yaml file and change the attribute apiVersion from v2 to v1

Now you can install

```
kubectl label nodes --all openwhisk-role=invoker
helm install ./helm/openwhisk --namespace=openwhisk --name=owdev -f mycluster.yaml
```

You should wait until install-packages in openwhisk namespace move to completed. This operation can take sometime.

To test your deployment do:

```
helm test owdev --cleanup
```

The next step is to install wsk which is OpenWhick client. On the master node, you should

```
exit #As a non root
wget https://github.com/apache/incubator-openwhisk-cli/releases/download/0.10.0-incubating
/OpenWhisk_CLI-0.10.0-incubating-linux-amd64.tgz
tar -xvf OpenWhisk_CLI-0.10.0-incubating-linux-amd64.tgz
sudo mv wsk /usr/bin/wsk
```

You can check your deployment by invoking a hello word action.

```
wsk property set --apihost $(Hostanme -I):31001
wsk list -v -i
wsk -i action invoke /whisk.system/samples/greeting --result
```

Deploy a JS script, an action, create a trigger and bind both of them.

On the master node (or on the slave node if your decide to install wsk on a different node than the master) do:

Create a js file, named hello.js   with the following content.

```
function main({name}) {
  var msg = 'you did not tell me who you are.';
  if (name) {
    msg = `hello ${name}!`
  }
  return {body: `<html><body><h3>${msg}</h3></body></html>`}
}
```

Creation of the action

```
wsk package create demo -i
wsk action create /guest/demo/hello hello.js --web true -i
```

Check if your action is created

```
wsk action list -i
```

Get the url of your web action

```
wsk action get /guest/demo/hello --url -i
```

You can check your js script execution by using the URL with curl command

```
curl https://(masterIP):31001/api/v1/web/guest/demo/hello.http?name=test --insecure
```

You can check the activation list of action with

```
 wsk activation list --limit 4 -i
```

You can see from the list that your hello script was invoked.

You can also invoke by using wsk

```
wsk action invoke demo/hello --result -i --param name World
```

Now we create a trigger

```
wsk trigger create test -i
wsk trigger list -i
```

Now create the rule

```
wsk rule create testRule test demo/hello -i
```

And then trigger using the trigger name

```
wsk trigger fire test --param name toto -i
```

You can check the execution output by getting the execution ID and use it to check the result

```
wsk activation list --limit 2 -i # Get the ID and use it in the next command
wsk activation result IDFromPreviousCommand
```