06.06.2019

Azure Kubernetes Best Practices Workshop

Workbook Team 1

Preparation

```
# clone the hands-on repository
git clone https://github.com/kubernetes-workshop/hands-on.git
cd hands-on

# please checkout your branch: team1
git checkout team1
```

Exercise #1

```
# deploy to kubernetes
kubectl run hello --image nginx --labels=app=hello --port 80 --namespace team1
kubectl get pod -n team1
kubectl port-forward pod/hello-xxxxxxxxxx-xxxxx 8000:80
visit http://localhost:8000

# create service with ClusterIP
kubectl expose deployments hello --port 80 --type ClusterIP -n team1
kubectl get services -n team1
kubectl port-forward service/hello 8000:80
visit http://localhost:8000

# set team1 as default namespace
kubectl config set-context $(kubectl config current-context) --namespace=team1

# validate it
kubectl config view
kubectl get pod
kubectl get pod -n team1
kubectl get services
kubectl get services -n team1
```

Exercise #2

```
# set environment variables (use your Dockerhub username)
$version="v1"
$username="your-dockerhub-username"

# build and publish to DockerHub
cd podinfo
docker build . --tag $username/podinfo:$version
docker login --username $username --password xxxxxxxx
docker push $username/podinfo:$version

# get current manifests from 'hello' and create deployment.yaml and service.yaml
kubectl get deployment hello -o yaml
kubectl get service hello -o yaml

# replace labels, selectors and images according to the new application:
"name: podinfo"
"image: your-dockerhub-username/podinfo:v1"

# run in kubernetes
kubectl apply -f deployment.yaml
kubectl apply -f service.yaml
kubectl get all

# port forward and visit http://localhost:8000
kubectl port-forward service/podinfo 8000:80

# troubleshoot => ask for help
kubectl describe pod/podinfo-xxxxxxxx-xxxx
kubectl logs pod/podinfo-xxxxxxxx-xxxx
```

Exercise #3

```
# create ingress, first take a look into the file and try to make sense of it
kubectl apply -f ingress.yaml

# give it a minute to install
visit http://team1.ddnss.de/
visit http://team1.ddnss.de/podinfo

# play around (subdomain, wildcard, regex)
1) host: "podinfo.team1.ddnss.de"
2) path: /*
3) path: /foo/bar/[A-Z0-9]{3}

# troubleshoot
kubectl port-forward service/hello 8001:80
kubectl port-forward service/podinfo 8002:80
visit http://localhost:8001
visit http://localhost:8002
```

Exercise #4

```
# deploy fibo application
kubectl run fibo --image=fnbk/fibo --requests=cpu=200m --expose --port=80
kubectl autoscale deployment fibo --cpu-percent=50 --min=2 --max=10
kubectl get all

# for each command open a new powershell (see scaling in action)
kubectl get hpa --watch
kubectl get pod --watch --selector run=fibo

# loadtest manual
kubectl run --rm -it manual-loadtest --image=fnbk/loadtest /bin/bash
Curl http://fibo.team1.svc.cluster.local # check DNS resolution
/app/hey -z 3s -c 64 -m GET http://fibo.team1.svc.cluster.local # make 64 requests in 3 seconds

# use a job to create an automated load test
kubectl apply -f loadtest.yaml

# inspect, see what happens, see scaling in action
kubectl get all
kubectl describe job.batch/loadtest
kubectl logs pod/loadtest-job-xxxxx

# cleanup loadtest job
kubectl get all
kubectl delete -f ./loadtest.yaml
```