


IT WORKS ON MY CLUSTER

Deploying a Groovy kubernetes application to Okteto

PRESENTATION

- Jorge Aguilera
- jorge.aguilera@puravida-software.com
-  @jorge@jvm.social
- Independent software developer

KNOW YOUR AUDIENCE

- Who knows Kubernetes (a.k.a. k8s) ?
- deploying in k8s?
- Knowledge of deployment, services, configmaps, ingress ...
- Cluster, namespace,
- AWS, GCP, Azure, ...

THIS TALK

- A kind of "guide" for first steps with Kubernetes
- Deploy a Groovy application in Internet using Okteto
- "Requirements":
 - Java/Groovy basic knowledge
 - Docker+docker-compose basic knowledge

DEPLOYMENT CYCLE

- write code (and write test obviously)
- build an image and push it to a registry
- redeploy updating the image id
- customers find a bug
- and start again

COLLATZ'S CONJECTURE

```
Map index(BigInteger n){
  def ret = [n:n, count : 0]
  while (!n.equals(BigInteger.ONE)) {
    if (n.mod(new BigInteger("2")) == BigInteger.ZERO) {
      n = n.divide(new BigInteger("2"))
    } else {
      n = n.multiply(new BigInteger("3"))
      n = n.add(new BigInteger("1"))
    }
    ret.count++
  }
  ret
}
```

MICRONAUT APPLICATION

```
mn create-app \  
  --features kubernetes \  
  --features postgres \  
  --features jdbc-data \  
  --lang groovy collatz
```

```
@Controller  
  ...  
  @Get('{n}')  
  Map index(BigInteger n){  
    ....  
  }
```

DOCKER-COMPOSE

```
services:  
  db:  
    image: postgres  
    ports:  
      - 5432:5432  
    environment:  
      POSTGRES_DB: collatz  
      POSTGRES_USER: postgres  
      POSTGRES_PASSWORD: postgres  
  app:  
    build:  
      context: ./
```


OKTETO

- requirements: Github account ... but business account :(
- you have a free namespace with 5Gb storage, 8Gb memory and 10 pods
- <https://cloud.okteto.com/login/>
- `curl https://get.okteto.com -sSfL | sh`

I'm not affiliate to Okteto, only a big fan

LIVE SESSION

if something can go wrong it will

OKTETO.YAML , DEPLOY

```
deploy:  
  compose: docker-compose.yml  
  
endpoints:  
  - path: /  
    service: app  
    port: 8080
```

\$ okteto deploy --build

WE ARE LIVE



<https://collatz-apacheconf-pvidasoftware.cloud.okteto.net/>

.. BUT WAIT, IT'S K8S

```
$ kubectl --context cloud_okteto_com-context get pods  
..  
$ k9s --context cloud_okteto_com-context get pods
```

OKTETO.YAML , DEV

```
dev:  
  collatz:  
    image: gradle:8.2.0-jdk17  
    autocreate: true  
    command: bash  
    sync:  
      - ./home/gradle  
    forward:  
      - 8080:8080  
      - 5005:5005
```

\$ okteto up

CLEAN

\$ okteto down --volumes

\$ okteto destroy --volumes

RECAP

We've deployed:

- deployments
- services
- ingress

without knowledge of Kubernetes!!!

NEXT STEPS

- let app use the database
- play with kubectl, configmap, secret, helm, ...
- deploy your pet project

OKTETO FOR MY PET PROJECTS

- Telegram/Slack/Whats App Bots
- Microservices + Apache Apisix
- ... Basically learn k8s

Q&A

- Form feedback



SLIDES & REPO

<https://slides.aguilera.soy/apacheconf/2023/collatz/>

