Docker Workshop 101

February 8th, 2017

June 13th, 2017	**Nader**	**Add Workshop #2 (Risk-Service with Kubernetes)**

Overview

Welcome to to Docker workshop 101; Docker is a full development platform for creating containerized apps, and Docker for Mac is the best way to get started with Docker on a Mac.

Goals

1. Give the practical understanding of Docker containers
2. Provide you with basic skills of using Docker on Mac/Windows
3. Create basic multi-service application on Docker
4. Have some fun!!

Let's go ...

Installing the Required Software

1. **Install Docker on Mac/Windows**
   a. Install VirtualBox on your laptop
   b. Install Minikube on your laptop

```
brew install docker

brew install Caskroom/cask/minikube

brew install kubectl
```

Goto the next page >>>

Workshop #1 (Pure Docker - Sunbit Free App)

Let’s create an simple (useless) multi service app:

- Simple Python Application (Flask)
- Redis Service

The workshop source code is found here

ssh: git@github.com:sunbit-dev/docker-workshop.git

or

https://github.com/sunbit-dev/docker-workshop.git

Demo:

:)

1. Browse the following files with your favorite text editor and read the short code
2.
    - **app.py** : The main web service
    - **Dockerfile** : Explains to docker how to package the app
    - **docker-compose.yml** : Multi Service App description

3. Open your terminal and cd to the source code directory
4. Connect docker command with Minikube

```
eval $(minikube docker-env )
```

5. First, let’s build the application docker image:

```
docker build -t web .
```

6. Let’s run our multiservice app (Flask + Redis)

```
docker-compose up -d
```

Lets test our app:

```
curl $(docker-machine ip default):3000
```

Workshop #2 (Sunbit Risk Service)

Let’s dockerize the risk-service:

1. Go to the project docker directory:

   ```
   cd ~/workspace/risk-service/Docker/docker
   ```

2. Build the Project

   ```
   cd ../../ && ./gradlew \
   clean build unzipJar zipAndRename -x test -ProotFolder="." \
    && cd -
   ```

3. Copy the artifact to the Docker directory

   ```
   cp ../../build/distributions/risk-service .
   ```

4. Build the Docker Image (Review the Dockerfile first)

   ```
   docker build -t risk-service .
   ```

5. Tag the docker image with the ECR registry

   ```
   docker tag risk-service \
   312765344048.dkr.ecr.us-west-1.amazonaws.com/risk-service
   ```

Goto the next page >>>

6. Copy the following Kubernetes Manifest file into `risk-service-ecr.yaml`

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: risk-service
  labels:
    name: risk-service
spec:
  replicas: 2
  template:
    metadata:
      name: risk-service
      labels:
        name: risk-service
    spec:
      containers:
      - name: risk-service
        Image:
312765344048.dkr.ecr.us-west-1.amazonaws.com/risk-service:latest
        ports:
        - containerPort: 3080
        imagePullPolicy: IfNotPresent
      imagePullSecrets:
        - name: ecr.sunbit.com
---
apiVersion: v1
kind: Service
metadata:
  name: risk-service
spec:
  selector:
    name: risk-service
  type: NodePort
  ports:
    - port: 3080
      targetPort: 3080
```

7. Run the docker as Kubernetes Deployment

```
kubectl create -f  risk-service-ecr.yml
```

8. Open the Kubernetes dashboard (It will open in your default browser)

```
minikube dashboard
```

9. List Running services

   (Method #1) `minikube service list`

   (Method #2) `kubectl get all --all-namespaces`

10. Get Service Log (Standard output/Error)

```
kubectl logs -f po/risk-service-xxxxxxxxxx-yyyyy
```

11. Logging into the Kubernetes Pod

```
kubectl exec -it risk-service-xxxxxxxxxx-xxxxx -- bash
```

   (You may exit session by typing <CMD>+<D>)

12. Killing the Deployment

```
kubectl delete deploy risk-service --force --grace-period=0
```