pm9TrackingServer Deployment Instructions

pm9

November 2019

The following are the steps to deploy the application used for tracking. We have used Kubernetes engine on Google Cloud Platform (GCP) for deployment. A major requirement of the application is to easily scale it up or down according to the requirement. This is the reason for deploying the application on Kubernetes, it makes it easy to add components according to requirement and the ability to scale them individually. Kubernetes also allows us to define configuration with **manifests.yaml** file.

1. To create a Kubernetes cluster an account on www.console.cloud.google.com needs to be created. A project along with a Kubernetes cluster needs to be created. Google cloud SDK also needs to be installed and initialized. After that *kube* needs to be configured. To do all this steps please find the steps in the tutorial here.

   kube is the productive, open source way to manage containers and microservices, automating the time-consuming tasks of installing, patching, upgrading, and carrying out cluster health checks.

   In this step we have created a cluster and configured *kube* to connect to the cluster.

2. Tracking server is a Gradle project. Install Gradle using this. To build the Gradle project go the root for the project and use the following command

   ```
   $ ./gradlew bootWar
   ```

   This creates a war file: *./build/libs/trackingserver-0.0.1-SNAPSHOT.war*.

3. Now we need to build a docker image for this application. To build the docker image, we first need to install Docker engine. For this use the tutorial at https://docs.docker.com/install/linux/docker-ce/ubuntu/. This tutorial shows how to install Docker engine on Ubuntu. Use the following command to create an image: *docker image build . -t ⟨repository − name⟩/⟨application − name⟩:⟨version⟩*. This command is run at root of the tracking server application. Here a *Dockerfile* is present which has the code to build the Docker image of the application. An example is shown here:

   ```
   $ docker image build . -t imagine5am/pm9-tracking-server:0.0.9
   ```

   Before pushing this image, we need to first create an account on https://hub.docker.com. After creating the account, use a command like the following to push the image:

   ```
   $ docker push imagine5am/pm9-tracking-server:0.0.9
   ```

   Here *imagine5am* is the name of the account on https://hub.docker.com.

4. Now use the *kubemanifests.yaml* to deploy the application. This file is created using *docker-compose.yml* and a tool called 'kompose'.

   ```
   $ kompose convert -f docker-compose.yml -o kubemanifests.yaml
   ```

   After this file i.e. *kubemanifests.yaml* is created *'type: LoadBalancer'* is added under *'web'* of kind *'Service'*. This enables us to deploy the application on a public IP. See the sample file *kubemanifests-master.yaml*.

   Note that *docker-compose.yml* file currently uses the application version 0.0.9 of docker image at https://hub.docker.com/r/imagine5am/pm9-tracking-server. You can continue using it, in which case you can skip all theses steps except step 1 and directly go to the last step.

5. You now need to deploy the application to Kubernetes engine:

   ```
   # kubectl apply -f kubemanifests.yaml
   ```

You can view the pods running on Kubernetes engine:

```
# kubectl get pods
```

Results:

NAME	READY	STATUS	RESTARTS	AGE
mysql-85877c9f76-mb9tv	1/1	Running	0	28h
redis-85b98669-drl8v	1/1	Running	0	28h
web-65858bd449-9jft5	1/1	Running	0	28h

You can view the services running on Kubernetes engine:

```
# kubectl get svc
```

Results:

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.56.0.1	<none>	443/TCP	29h
mysql	ClusterIP	10.56.10.74	<none>	3306/TCP	28h
redis	ClusterIP	10.56.5.167	<none>	6379/TCP	28h
web	LoadBalancer	10.56.1.132	34.93.44.41	80:30631/TCP	28h

Your web service is now running at IP 34.93.44.41 MySQL and Redis Server are also running at the IPs as seen in the results. To view logs use commands like the following:

```
# kubectl logs web-65858bd449-9jft5
```

—x—