Deploy app over k8s – Hands on

**Prerequisites:**0. Docker knowledge

1. Account in Google cloud – New user gets 300 $ free   
   
<https://cloud.google.com>

2. Account in Docker hub   
  
<https://hub.docker.com>

3. Account in Mongo dB Atlas

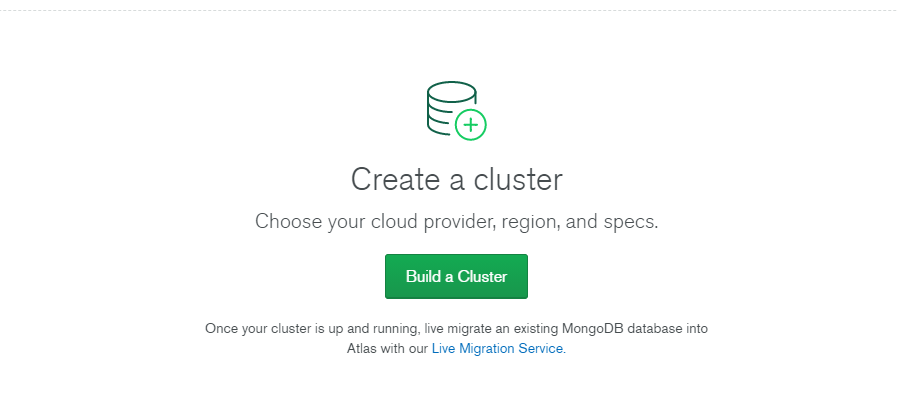
<https://www.mongodb.com/cloud/atlas>

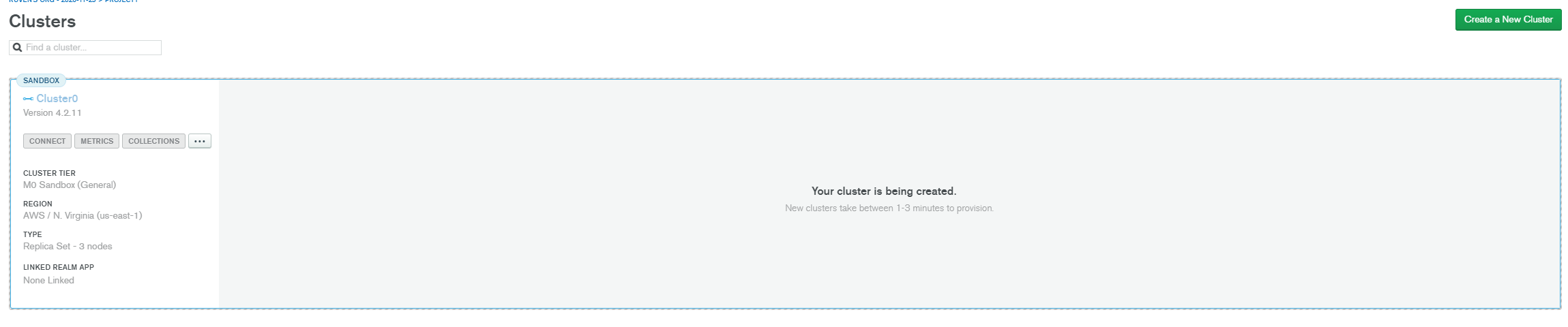
**What are we going to deploy?**   
App translate the client IP to morse code, present the result and the number of times the user visited in the web over the screen.

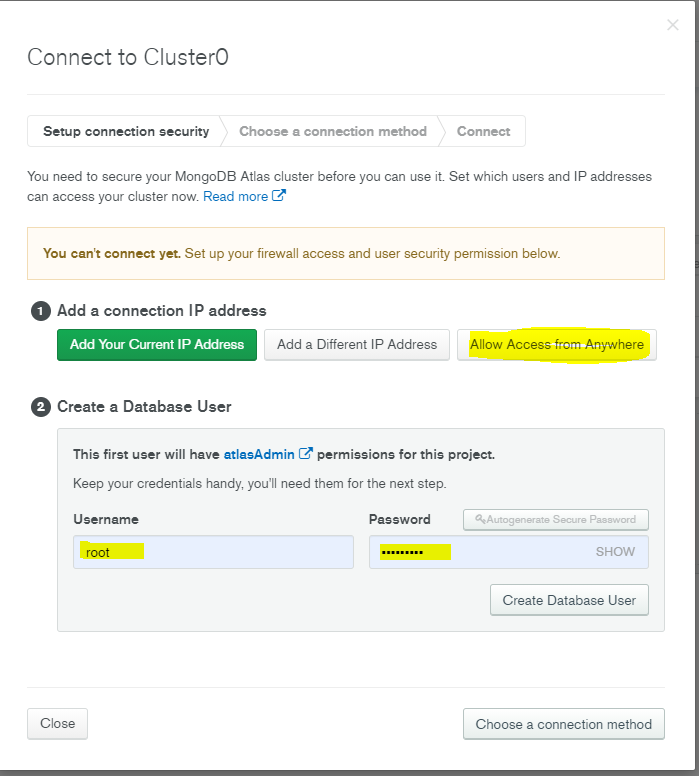
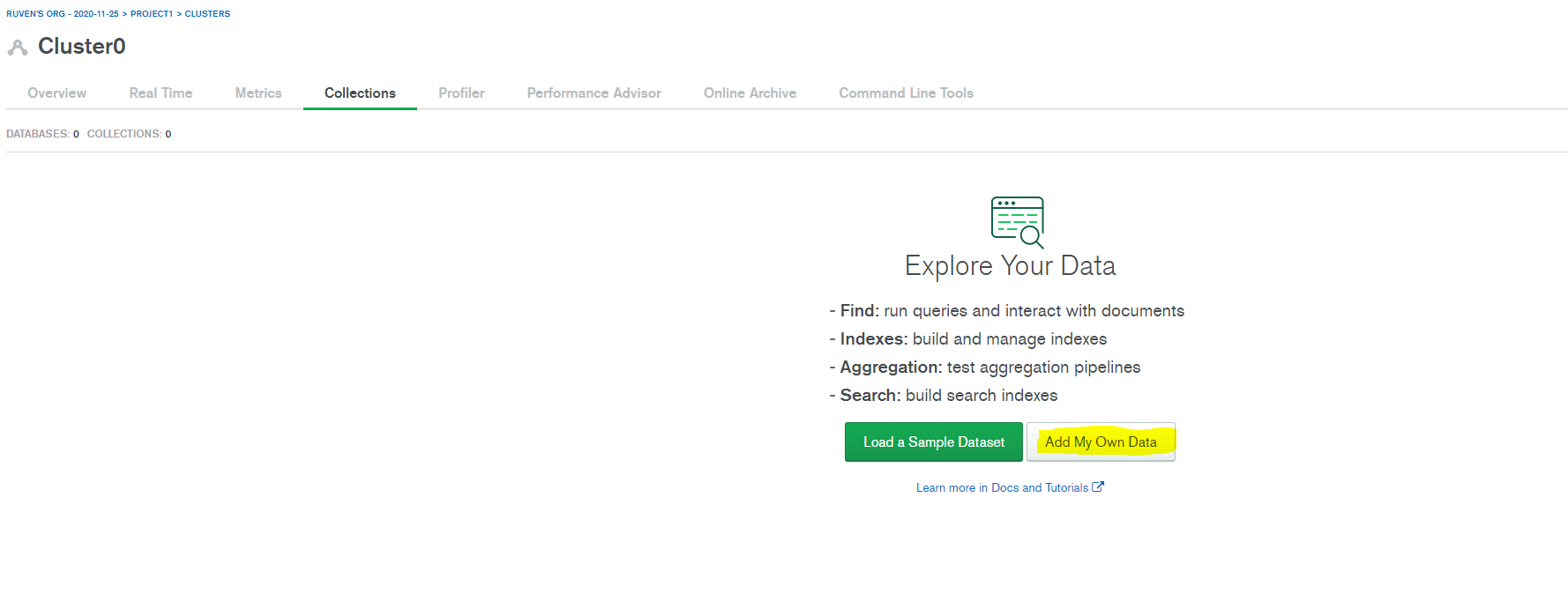
Which components are you going to create?

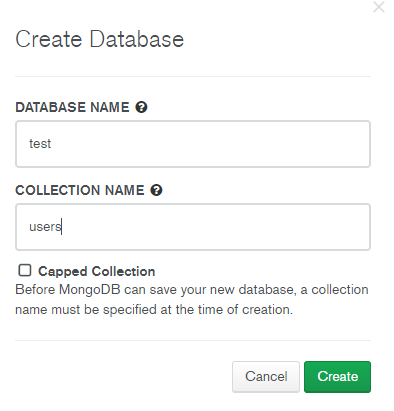
1. Docker image with the app (create the image and push the image to your Docker hub repository)
2. Create Mongo DB database for your app – for saving the user visits
3. Create Deployment Yaml file
4. Create Service Load Balancer Yaml file
5. Deploy the app like a ninja   
   
6. Teardown the App

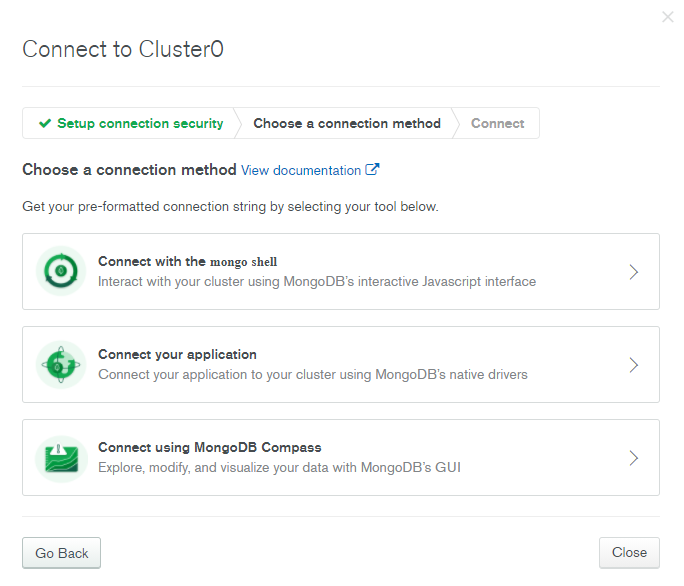
**Creating the DB:**

1. Go to <https://www.mongodb.com/cloud/atlas>
2. Click on the new project and create your first project!
3. Once you created the project, now it is the time to build a cluster – after you click on Build a cluster, then click on the free cluster creation button and then create your cluster with the default values.  
   
4. Now we need to wait till the cluster will be ready



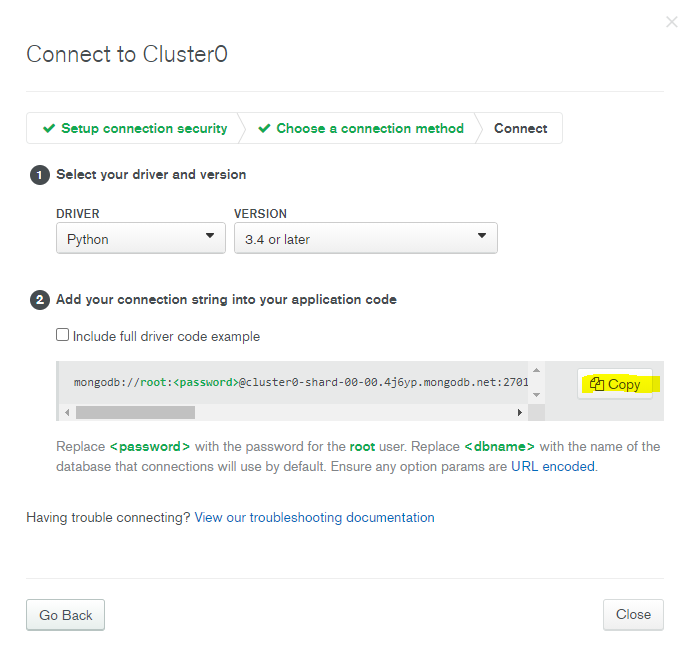
1. Let's create permission login to the DB - for simplicity please fill up like the picture below.  
   Please click on the connect button and then do like the picture – Don't forget to choose a password for the root user and allow access from anywhere  
   
2. Please click on collections button and then click on add my own data and then enter database name and collection name  
   



1. Adding mongo DB into your app – Please click on connect button and then Connect your application button.  
     
   

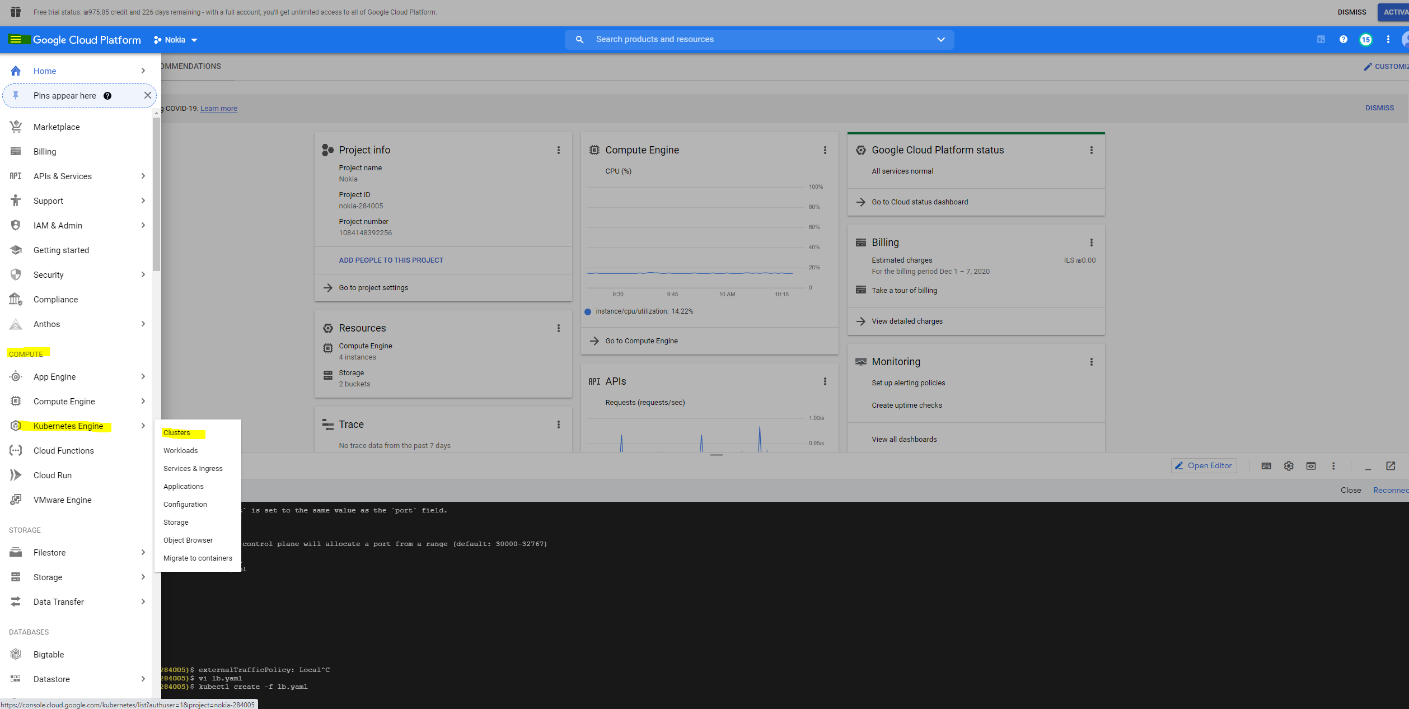
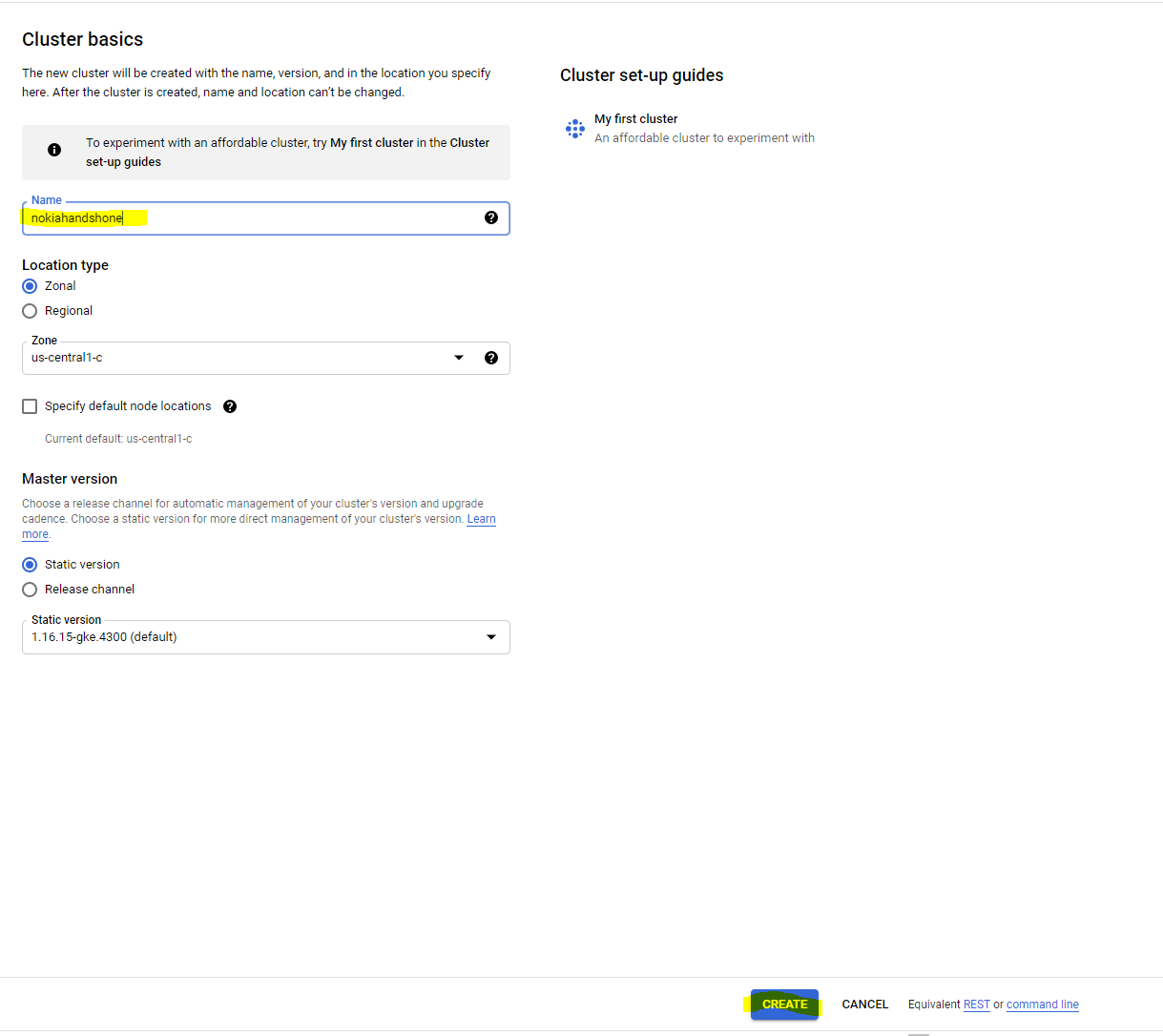


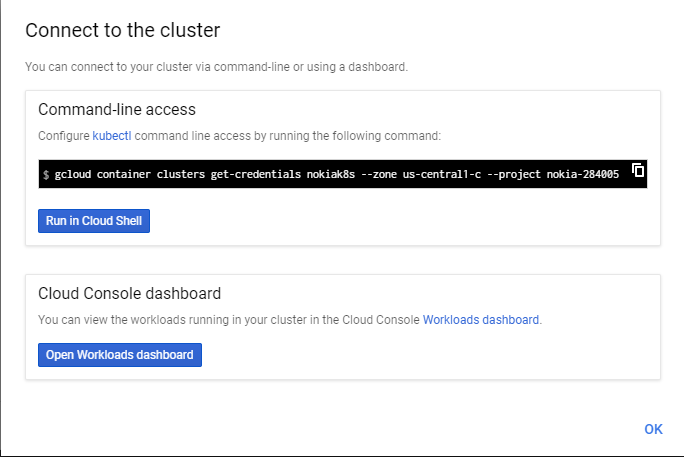
1. Copy the connection string and save it – Replace **<password>** with the password for the **root** user. Replace **<dbname>** with the name of the database that connections will use by default.





**Creating GKE – google Kubernetes engine:**

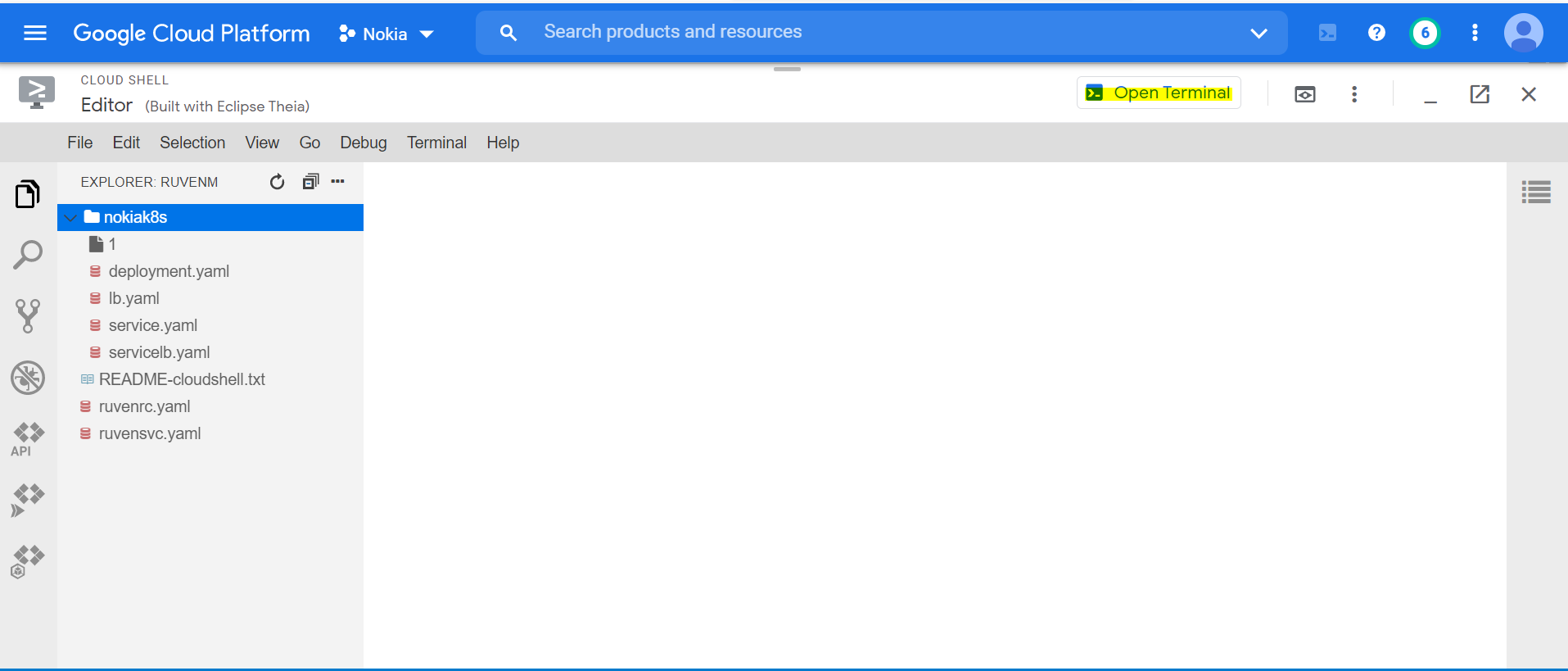
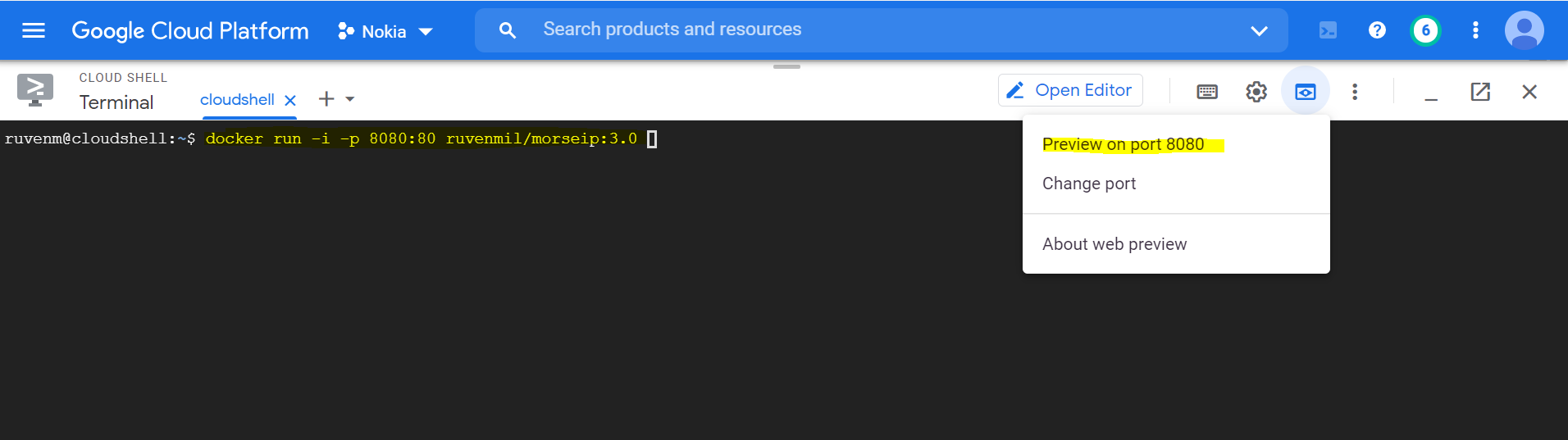
1. Login to your google cloud - <https://cloud.google.com>
2. Click on create cluster and then chose cluster name and then click on create   
   
3. Once your cluster is ready then click on connect button and then click on run in cloud shell button



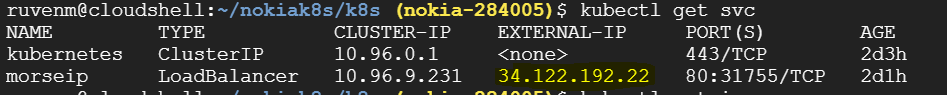


1. An internal terminal should open on your web, click on enter button and then test your cluster by running the command:  
     
   kubectl get nodes -o wide
2. Download the tgz file   
   <https://storage.googleapis.com/nokiak8shandson/nokiak8shandon.tgz>
3. Open the file : tar xzvf nokiak8shandon.tgz

**Build the app:**

1. In the folder there are app folder and k8s folder.  
   Please go to app folder and edit the app.py file – in the comments there are instructions what to do.  
   You can edit the python script by open editor button(instead of Open Terminal you will see Open editor):  
   
2. Now after we have finished to edit the python script, this is the time to build the container image:  
   *docker build . -t "your name in dockerhub/containerName:your\_tag"*
3. This is the time to check if the docker image is working   
     
   *docker run -i -p 8080:80 "your docker image"*
4. Once the container is running then you can try it by open web preview  
     
   
5. If you can see the web, then this is a good time to deploy the app over k8s 😊
6. Now we need to push the image we created to Docker hub registry but first we need to login.  
     
   docker push "youdockerimage:tag"
7. Go to k8s folder and edit the deployment yaml –

* Change to value of image with your image name
* Look on the replica number and on the selector key

1. Deploy the app -   
     
   kubectl create -f deployment.yaml
2. Check the pods status -   
     
   kubectl get pods
3. Once the pods are running let's check what happen if we are removing some pod.  
     
   kubectl delete pod "chose one pod from the list"  
     
   wait for 3 sec  
     
   kubectl get pods  
     
   **What happen?**
4. Deploy the network –   
     
   kubectl create -f lb.yaml
5. Check the load balancer service -   
     
   kubectl get svc   
     
   Please wait till an external network appear  
   
6. Please try to reach the website by running in the browser the external IP

**Tear down:**  
