**APPLICATION CONTAINERIZATION**

**LAB EXPERIMENT 8**

**SUBMITTED BY:**

NITISH KUMAR SINGLA

500071913

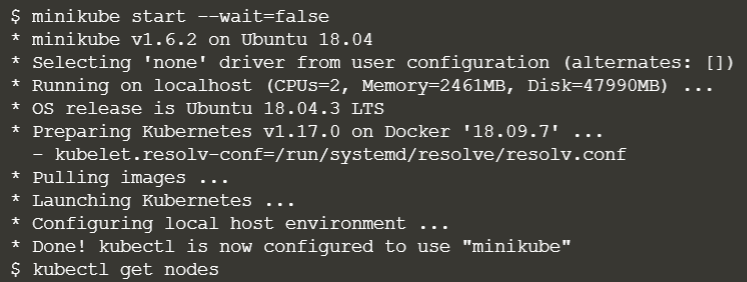
R171218126

**SUBMITTED TO:**

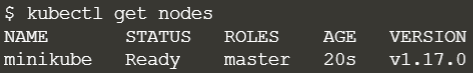
HITESH KUMAR SHARMA SIR

**Kubernetes – Minikube Installation and Fundamentals**

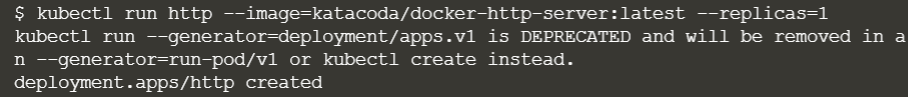
Execute the command below to start the cluster components and download the Kubectl CLI.



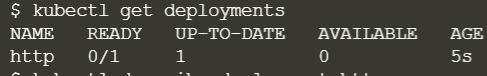
Wait for the Node to become Ready by checking



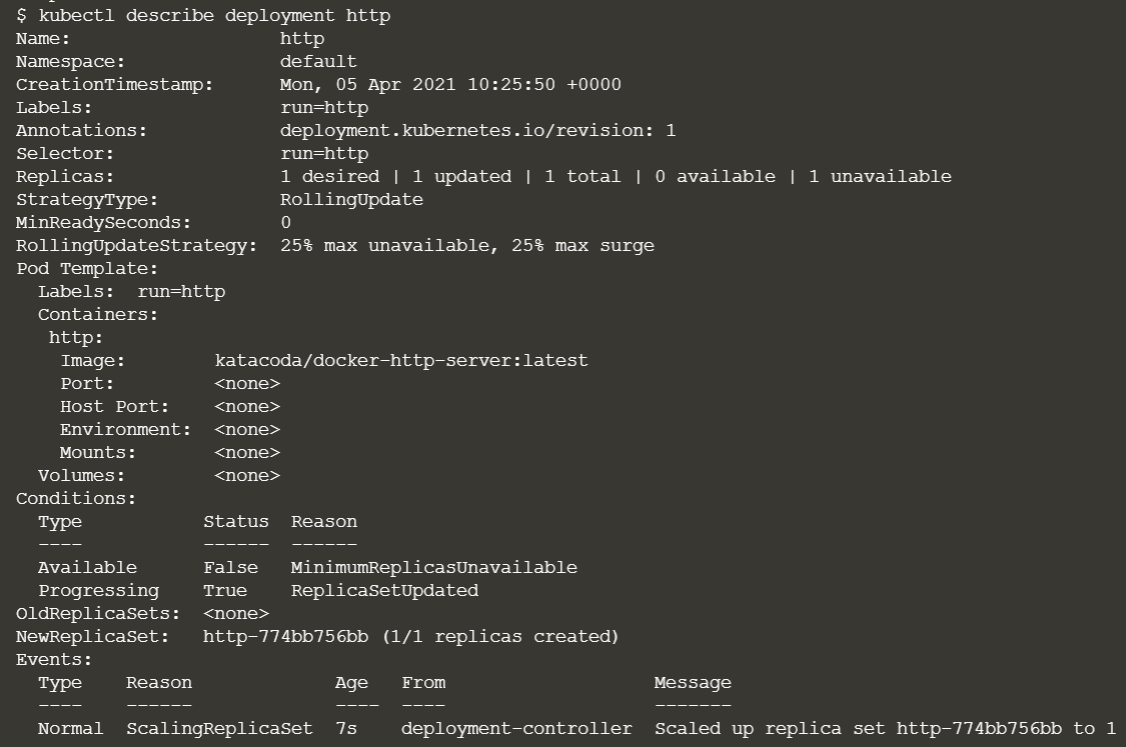
The following command will launch a deployment called http which will start a container based on the Docker Image katacoda/docker-http-server:latest.



You can then use kubectl to view the status of the deployments



To find out what Kubernetes created you can describe the deployment process



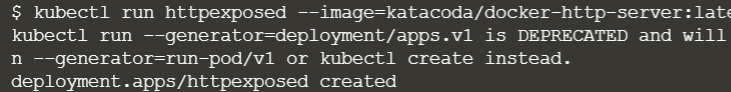
Use the following command to expose the container port 80 on the host 8000 binding to the external-ip of the host



You will then be able to ping the host and see the result from the HTTP service.



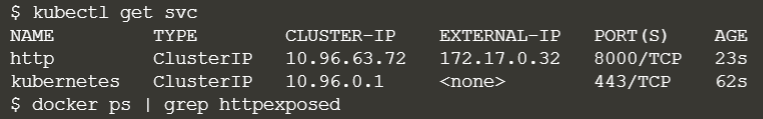
Use the command command to create a second http service exposed on port 8001.



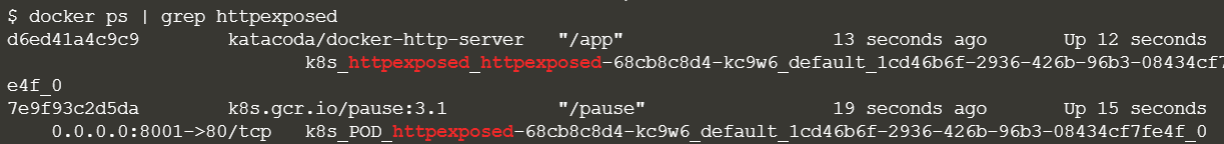
You should be able to access it using



Under the covers, this exposes the Pod via Docker Port Mapping. As a result, you will not see the service listed using



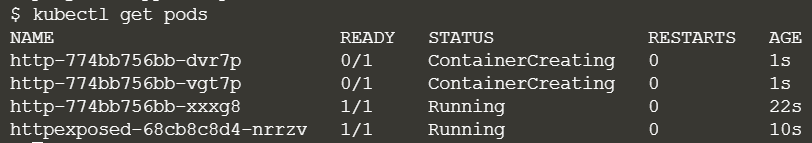
To find the details you can use



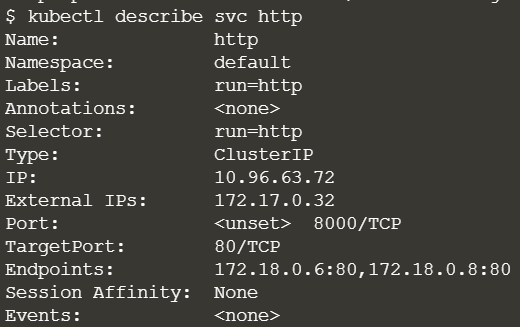
The command kubectl scale allows us to adjust the number of Pods running for a particular deployment or replication controller.



Listing all the pods, you should see three running for the http deployment



Once each Pod starts it will be added to the load balancer service. By describing the service you can view the endpoint and the associated Pods which are included.



Making requests to the service will request in different nodes processing the request

