**Prerequisite to run the below command:**

1. Install Docker on your local machine.
2. Install minikube on your local machine. {I am using cloud provider here due to limitation in memory on my laptop. My latop started having memory issues on various occasion on running both Docker and Minikube together. So I decided to use Kubernetes Cluster on a Cloud platform(Digital Ocean) .}
3. Create Kubernetes cluster on any cloud provider. { Here I am using Digital ocean as there is no extra chares for setting up Kubernetes Cluster as in case of AWS which charges $0.10 per hour per cluster that we create in AWS. With Digital Ocean I could utilize the promo code and compete this task without incurring any charge.}

**Containerize and deploy application on kubernetes cluster.**

1. Create a Dockerfile on your machine using the project script. Final Dockerfile added in github repository. { here I used ruby:alpine as base image as there was no dependency on the other OS component, so as to keep the image file size low.}
2. Create a docker image using the dockerfile.

Docker build . -t amuaamir/ruby-webserver:latest

1. I saved the docker image created on to Docker Hub

docker push amuaamir/ruby-webserver:latest

1. Now the Docker image is ready to be deployed on to Kubernetes Cluster.
2. Created the Deployment file for ruby-webserver. Script available at the github location.
3. The Deployment is created with 1 replica for high availability. We can scale up as needed. I have tested this by deleting the running pod of webserver. The deployment creates a new pod instantly.

File: web-server.yaml

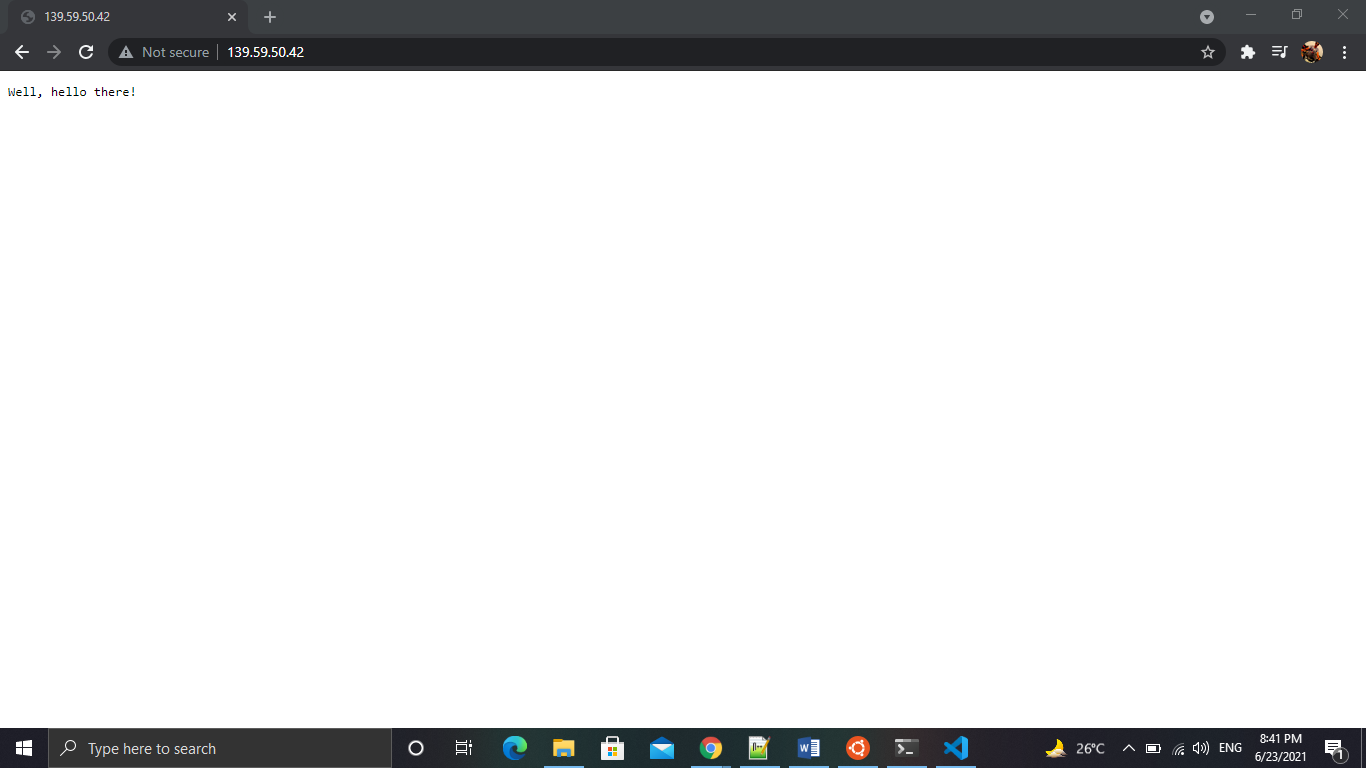
1. We have created a service in Kubernetes for LoadBalancer to expose the worker node to external IP, for us to access the web-server.

File: service-definition.yaml

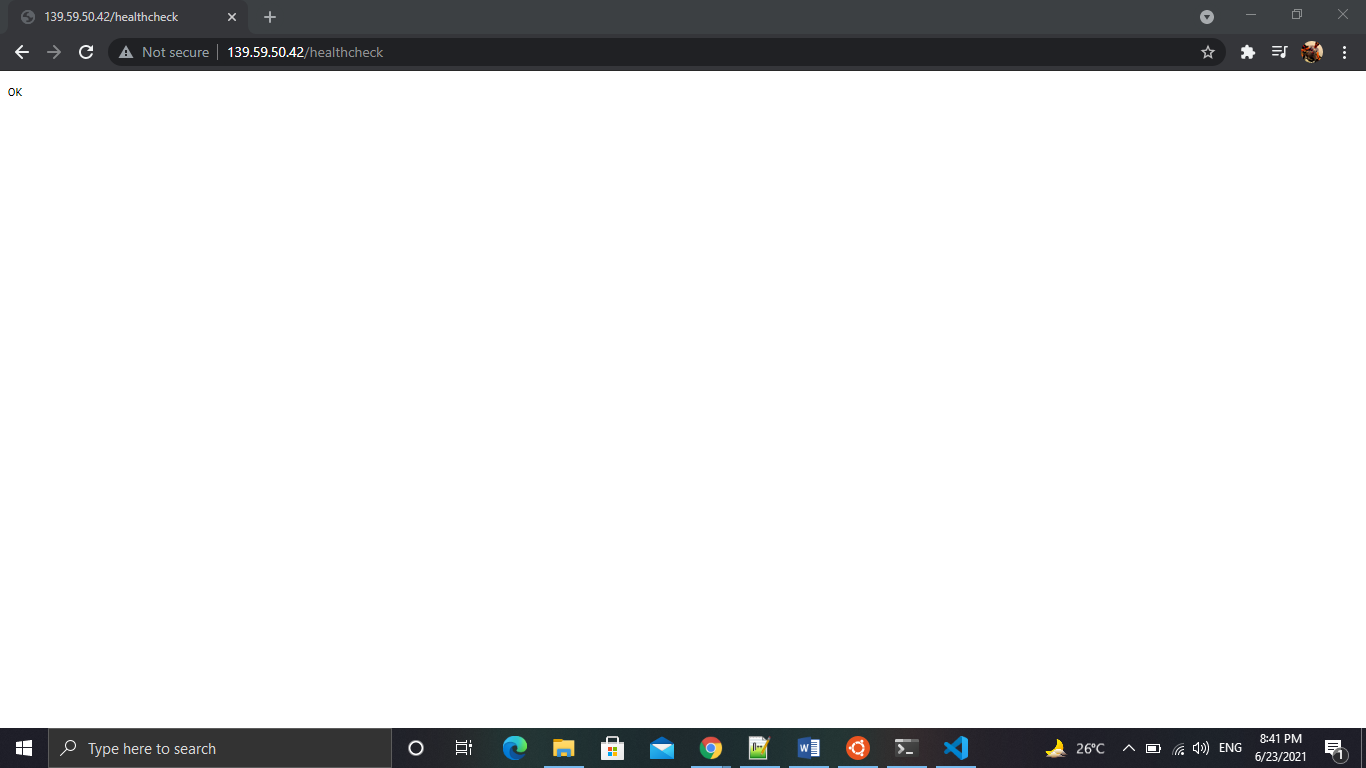
1. You can get the external IP address using the below command:

Kubectl get services

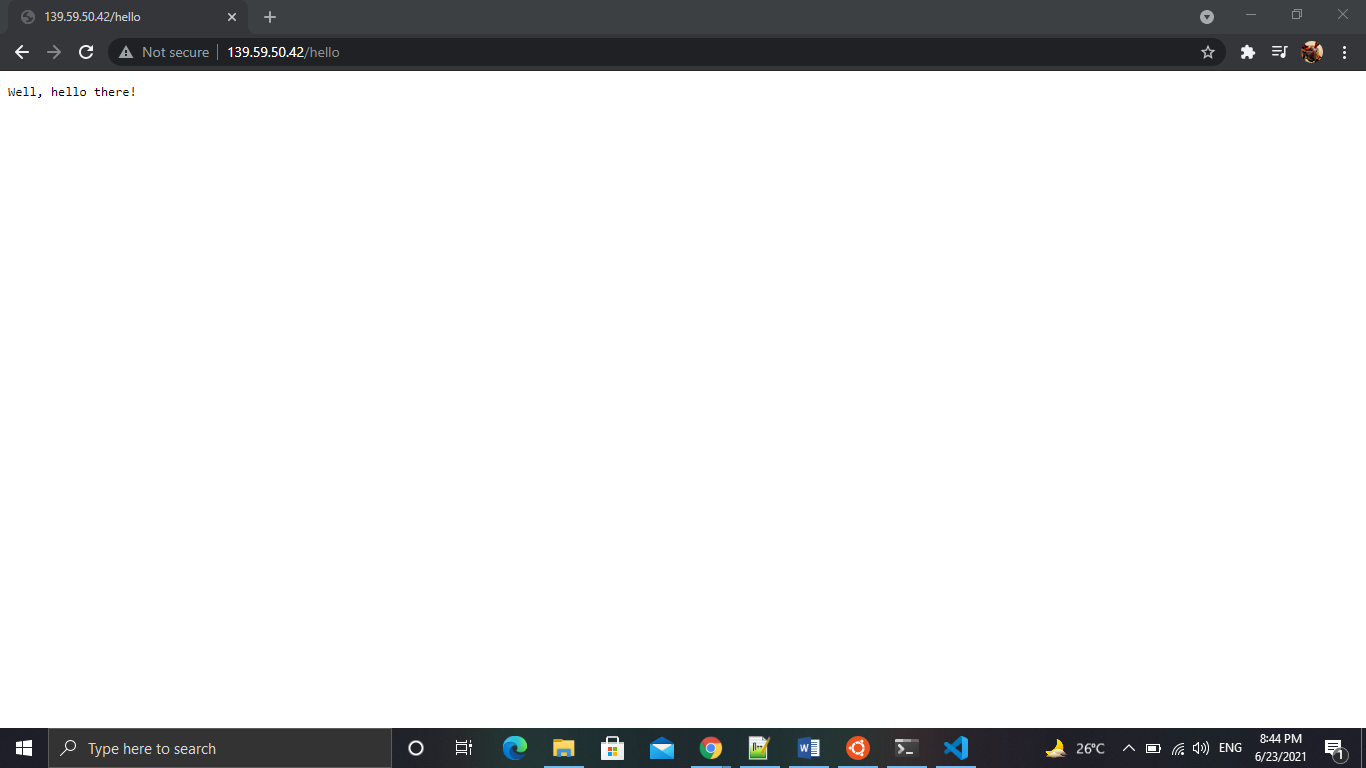
1. You can access the web-server from your browser using the above external Load Balancer IP.
2. Screenshot of the successful implementation attached.



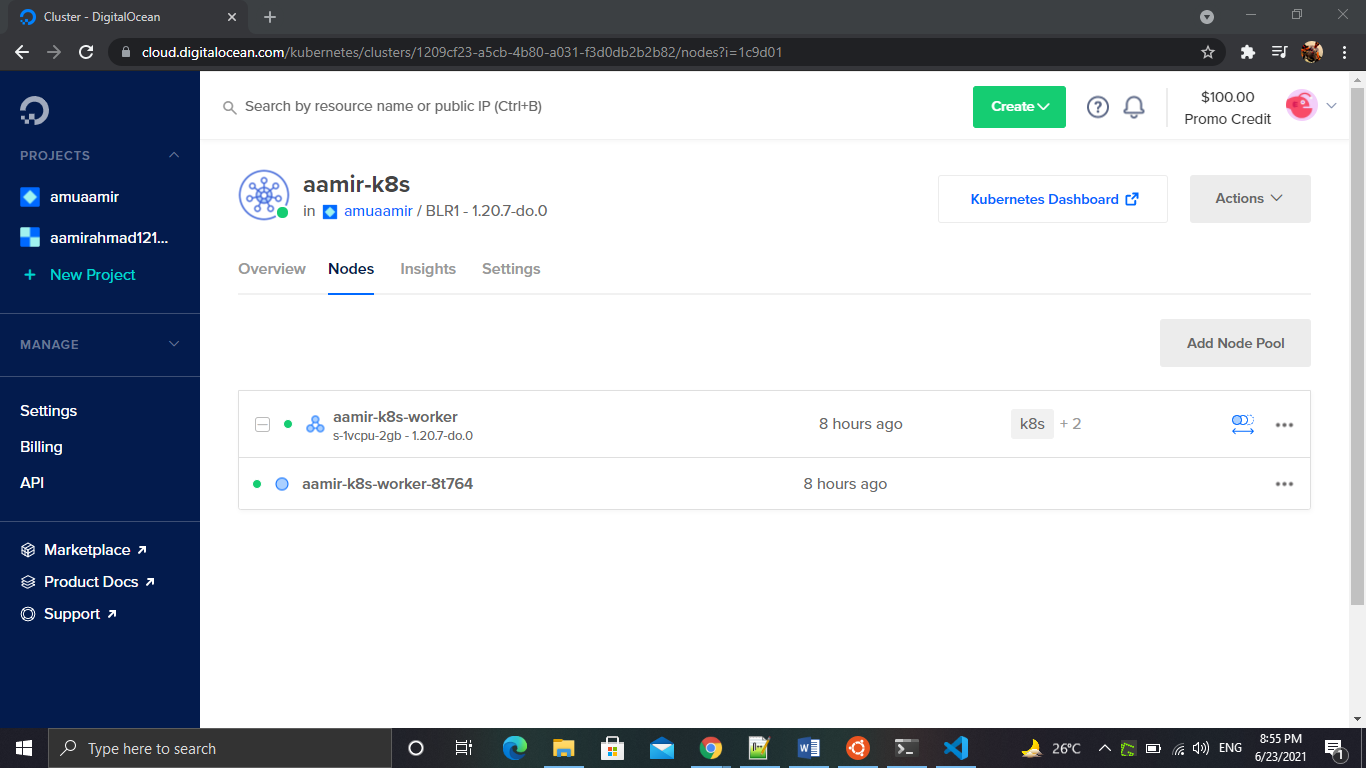
Checking the path **/healthcheck**: Expected Output: **OK**



All other path returns: **Well, hello there!**



Kubernetes Cluster details:



Load Balancer:

