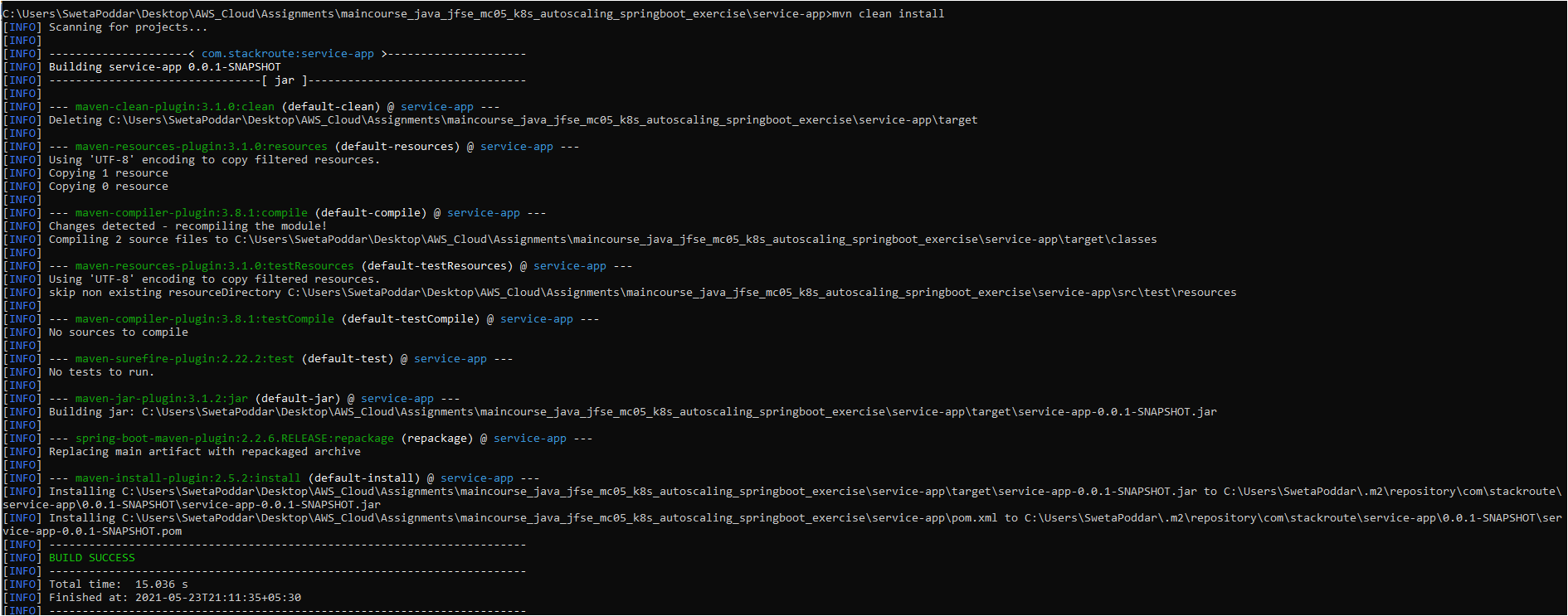
**Auto-Scaling on local system**

**1. Build the Spring Boot application.**



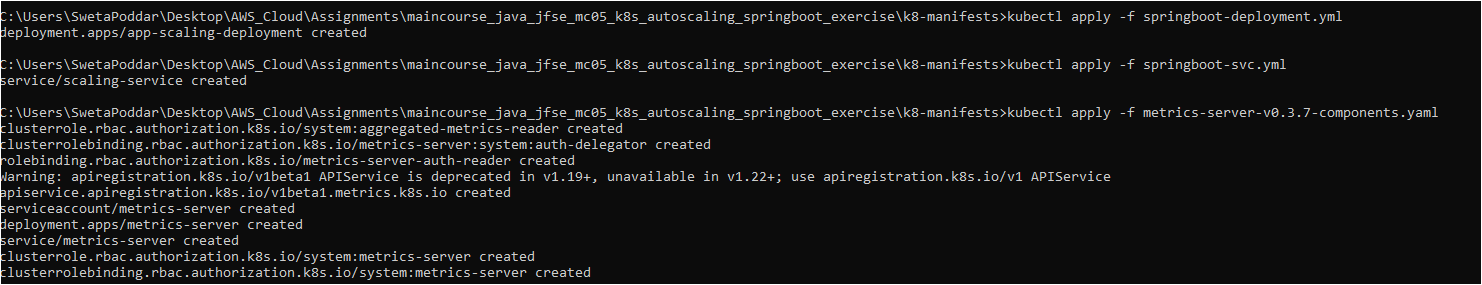
**2. Build the Docker image for the Spring Boot application.**

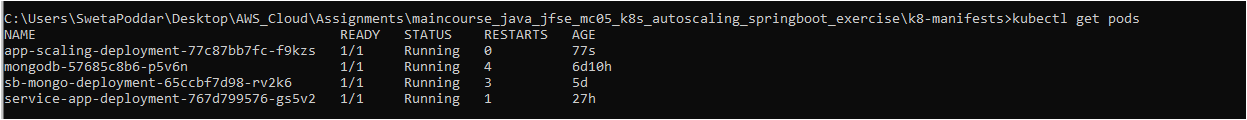
**3. Tag the docker image.**

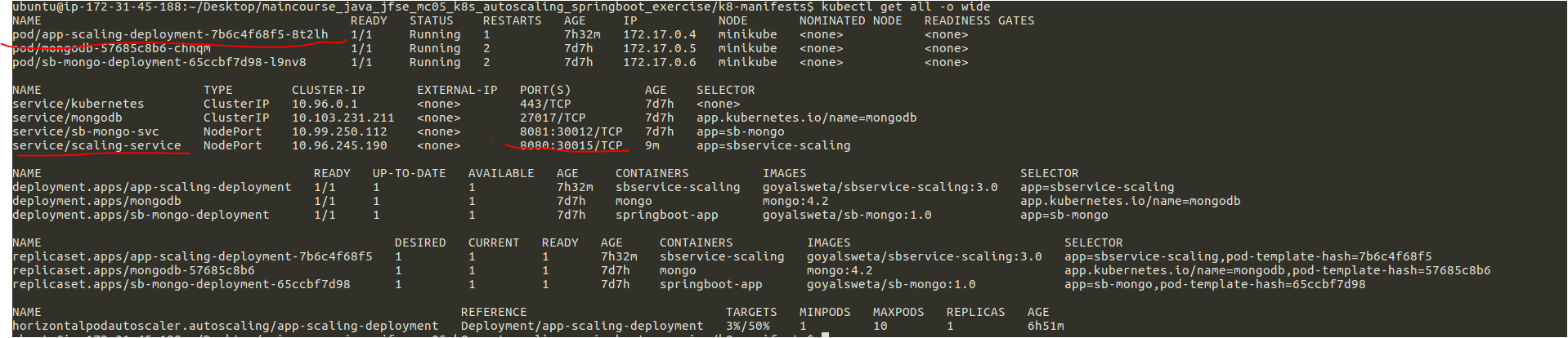
**4. Push the image to Docker Hub.**



**5. Deploy the manifest files springboot-deployment.yml and springboot-svc.yml**







**Execute the command to get minikube ip -> minikube ip**



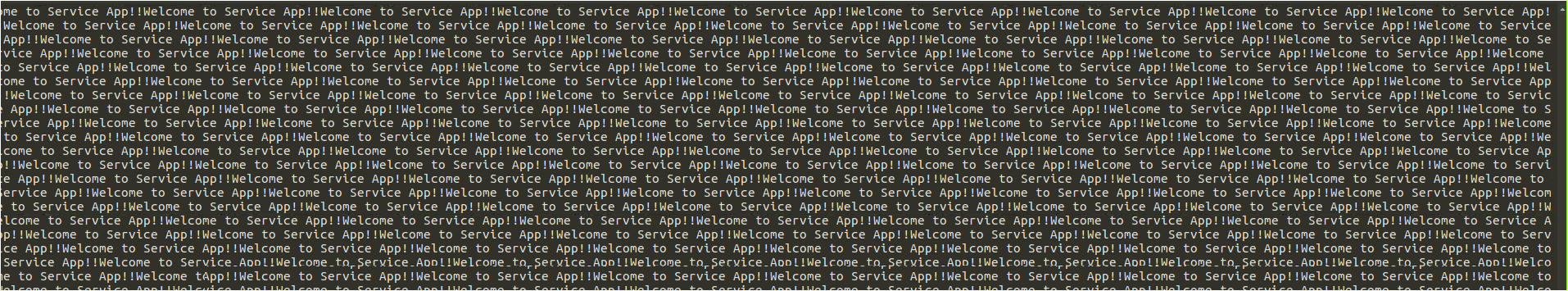
**Test the service in web browser with ip =192.168.49.2 and port = 30015**



**Execute the command -> kubectl run -it --rm load-generator --image=busybox /bin/sh` in a different terminal to start a new container.**



**Create load by running Execute the command -> while true; do wget -q -O- http://192.168.49.2:30015/api/v1/hello; done**

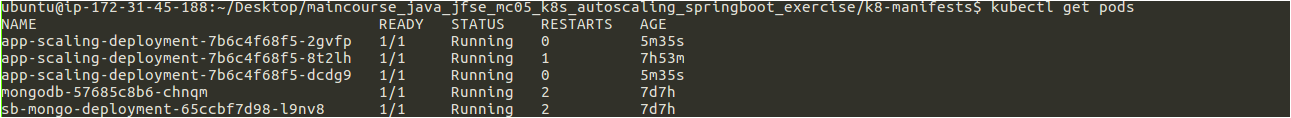


**Again, run the command in step 8 (kubectl get hpa) to check how the auto-scaler reacts to the increased load on the server.**



**Wait for few minutes, and observe Hpa Status, new pods created 🡪 3 replicas**





**Exit the terminals of step 9 and step 10 to terminate the loops sending requests to the server and stop the user load.**

**Run command `kubectl get pods` to see the pods scaling down.**

* **The number of pods got reduced to 1, after terminating the loops sending requests to the server.**

