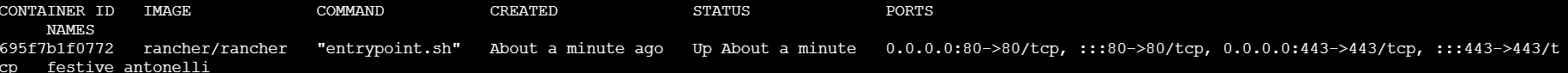
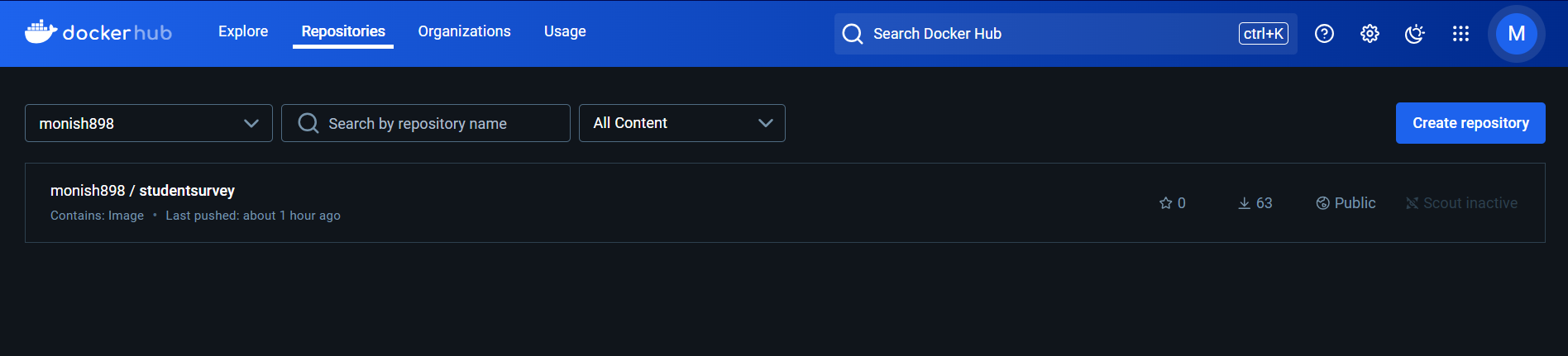
HW2 SWE 645

Monish Dhilipkumar (G01411482)

**Docker image creation**

1. In order to deal with docker image creation first we need to navigate to the folder where the site is created and open a terminal from that directory.
2. Run jar -cvf <war\_filename>.war\*
3. This will create a war file after which you need to build the image and need to run the command  
   docker build –tag<image\_name>
4. Containerize your image docker run -d -p 8080:80 my-survey
5. Then you can type docker ps and check if the container is running and it will show you the id  
   
6. Now the next step involvers pushing to docker hub
7. So first you need to login to docker using the command docker login -u <username>
8. Tag the image docker tag my-survey dockerhub-username/my-survey
9. Push it to docker hub using the command docker push my-survey
10. Verify if the image is pushed properly in docker hub

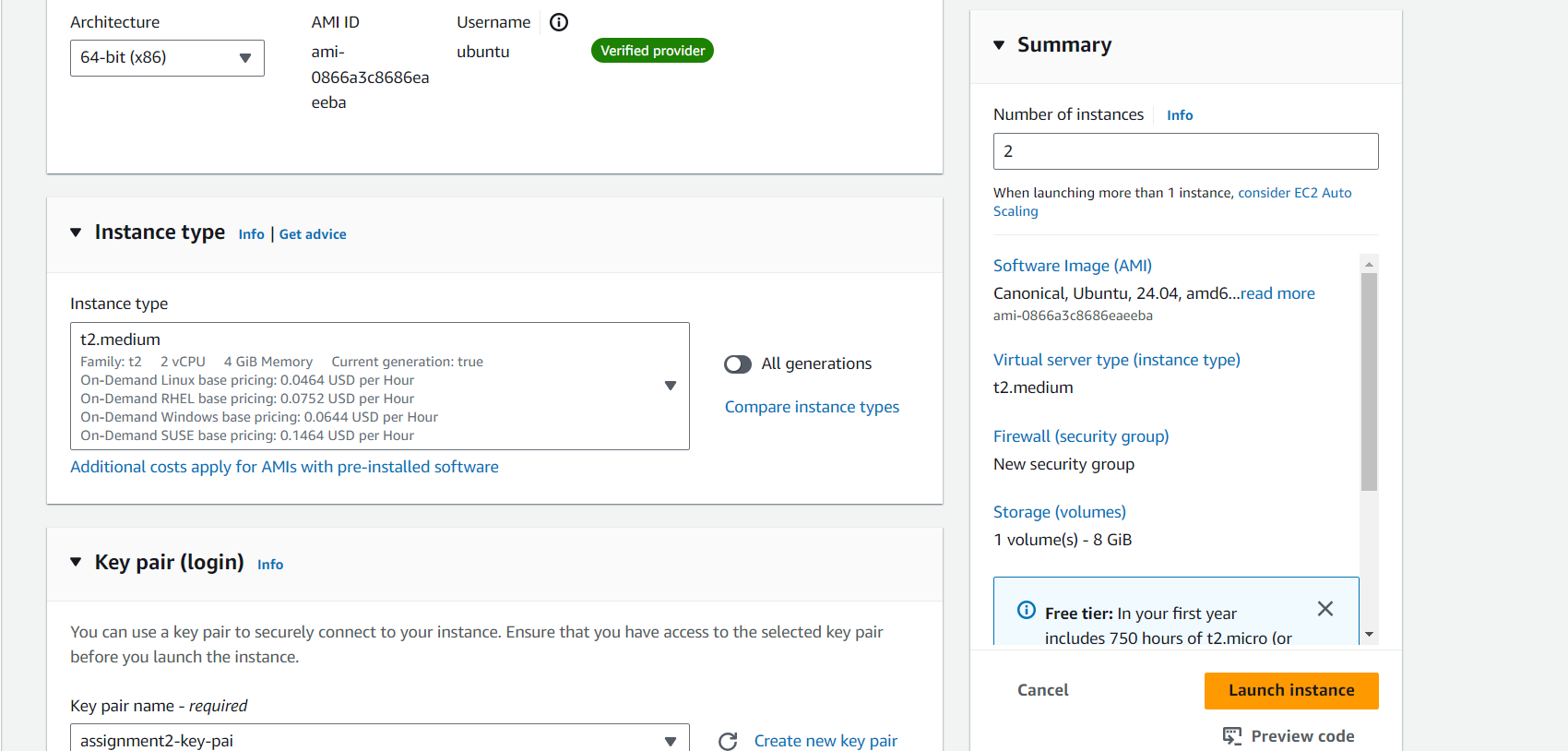


1. Login into the AWS console and login into your account and choose EC2  
2. Add a name for the server for eg rancher   
3. Choose the ubuntu server 24.04 LTS and 64-bit (x86) architecture

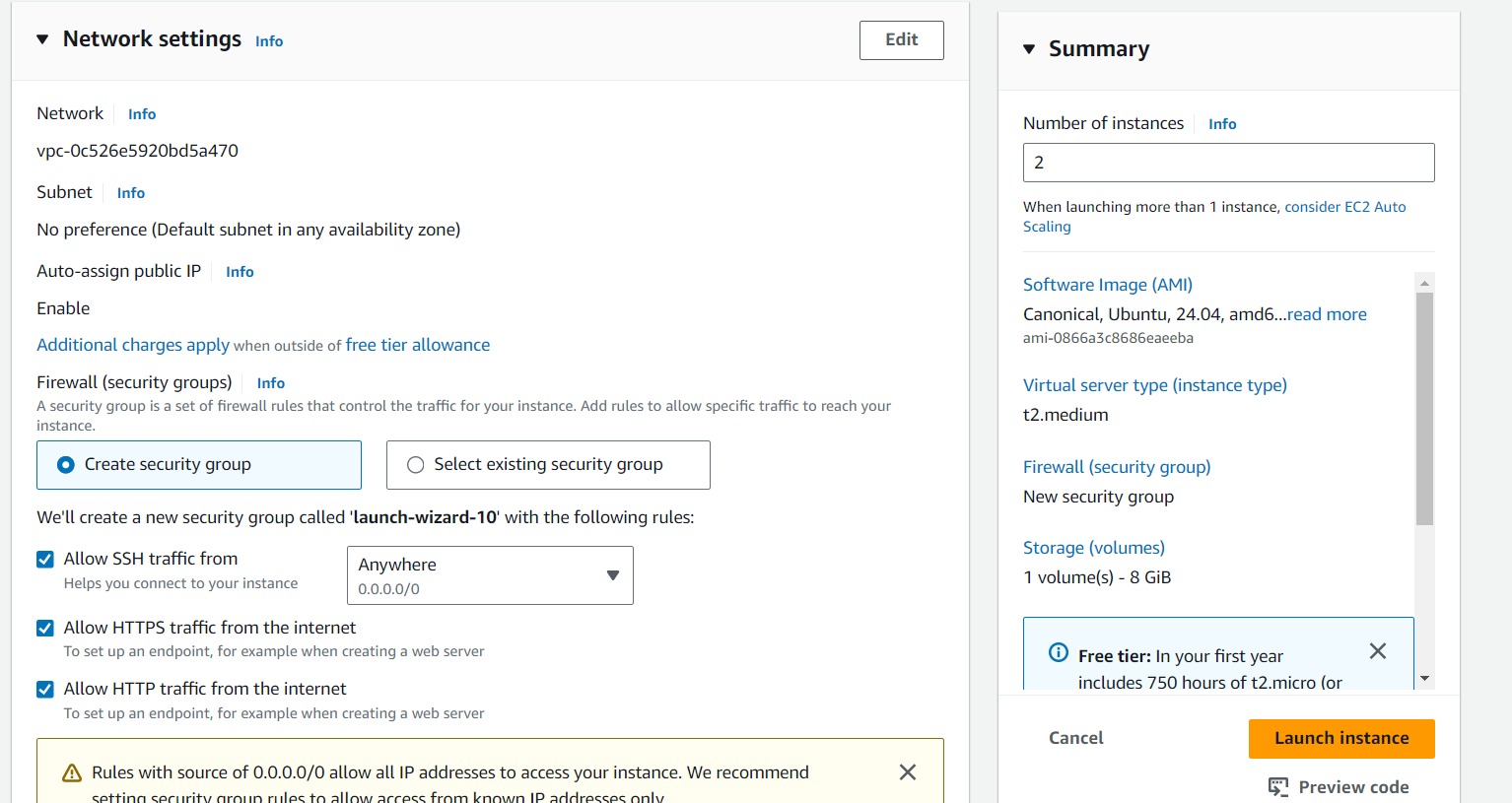
4. change the instance type to t2.medium or t2.large as this will be apt for this assignment. T2.micro will not be suitable.  
  
  


5. Change the number of instances to 2

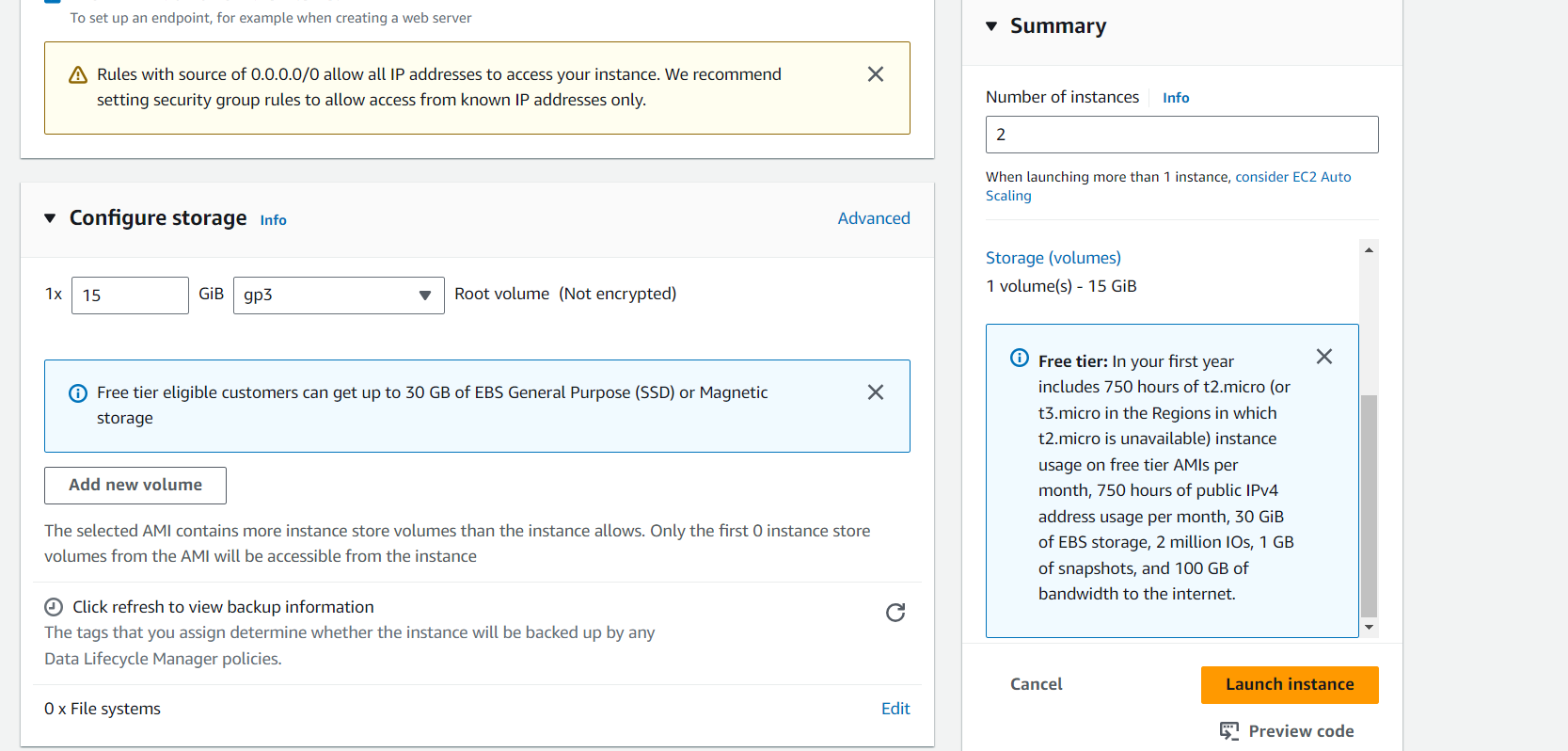
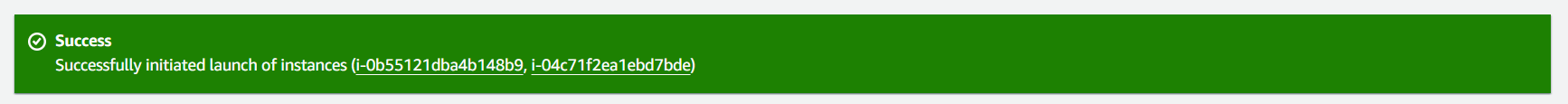
6. choose or create a key-pair and download and save it to your local machine

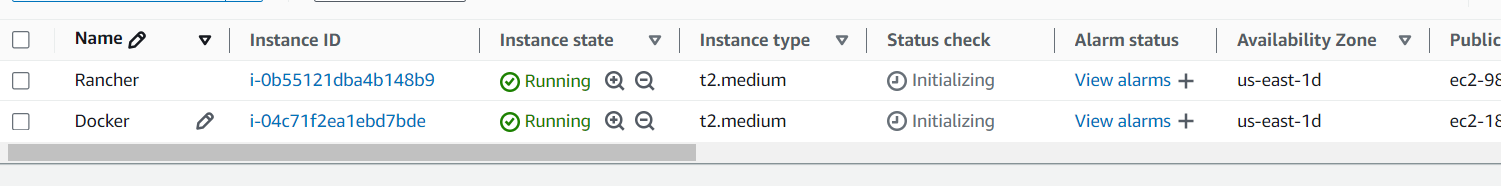


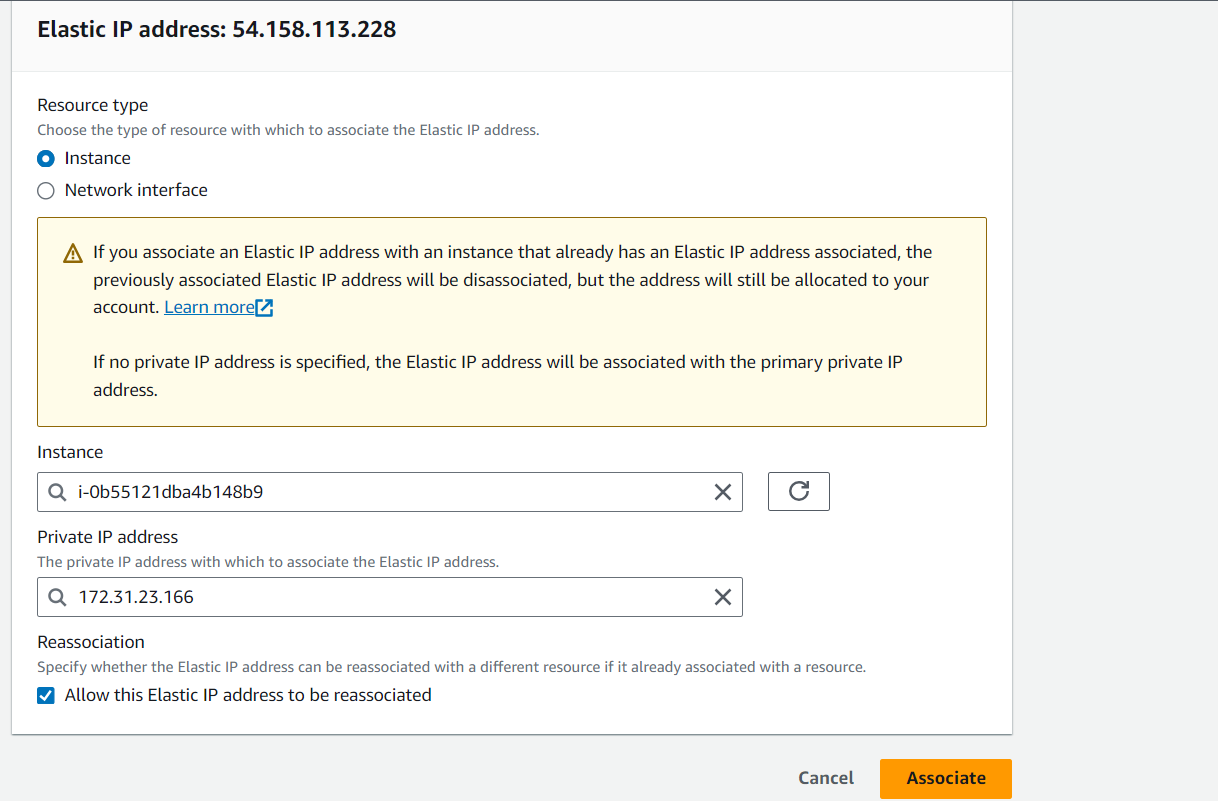
7. Under network settings create a security group in which you need to allow the networks http(80), https(443) and custom TCP (8080) and allow it from anywhere 0.0.0.0/0

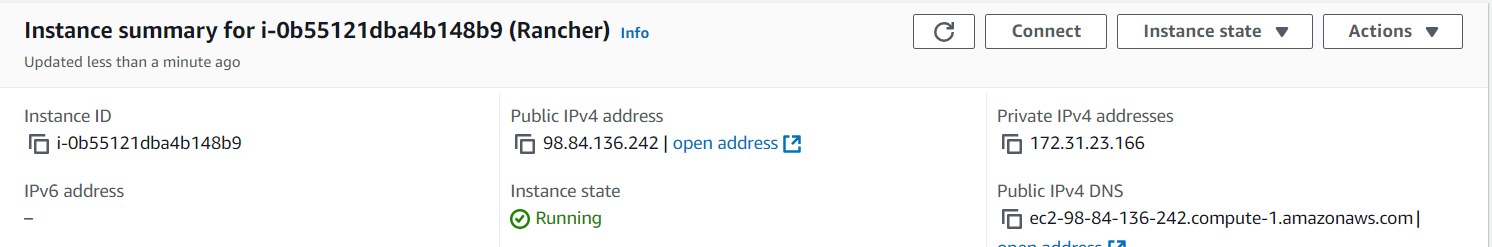


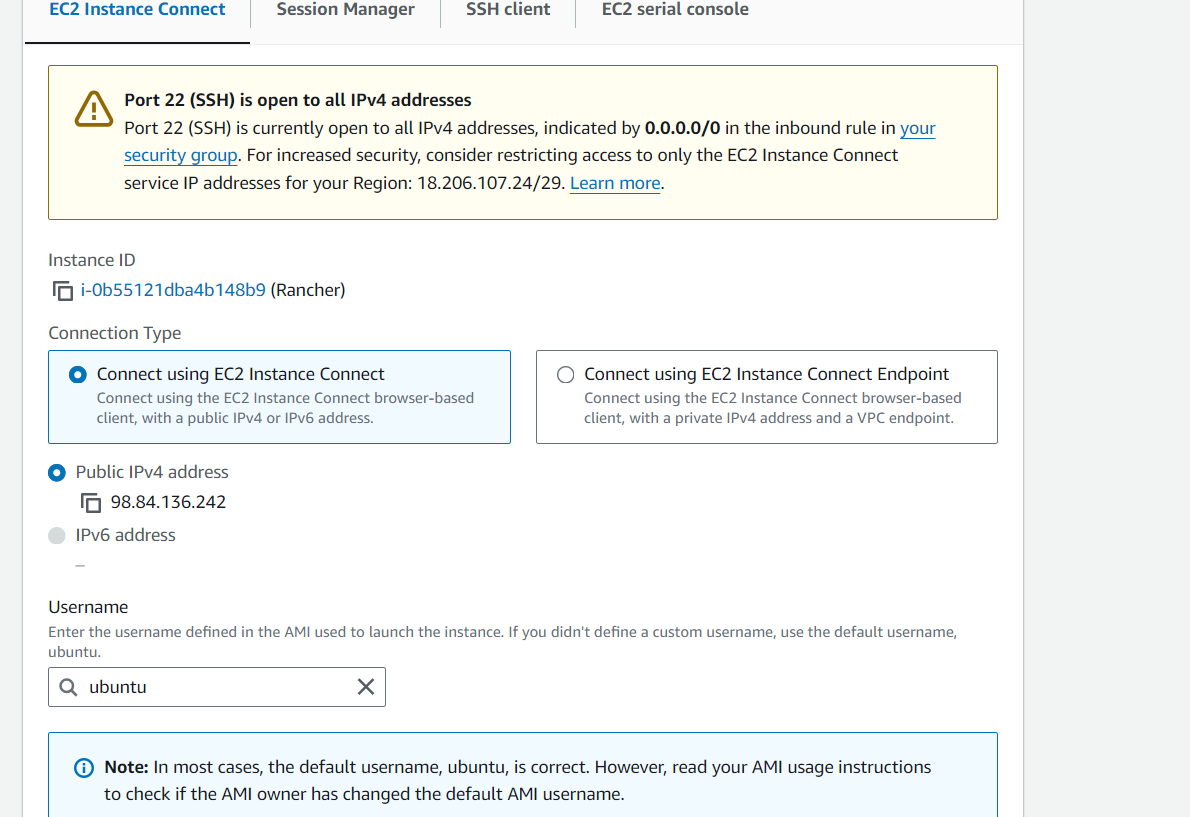
8. Add more storage as mentioned around 15GiB

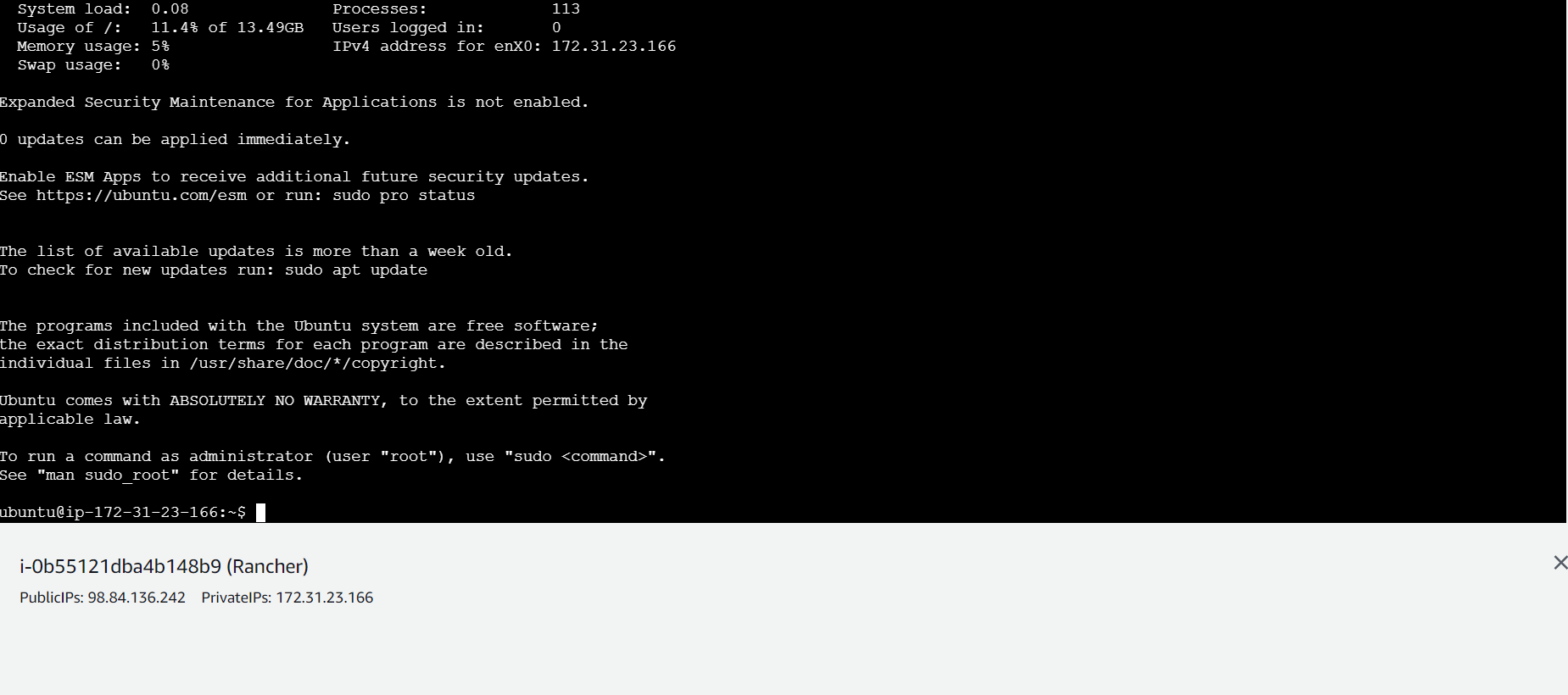
  
  
9. Now you can see that two instances are created and you need to name both the names as rancher as the first instance and the second instance as docker as in the first one you launch rancher and install docker in the second one  
  


  
  
10. I have also associated an elastic ip address for both the instances in order to make accessibility much more convenient by being much more convenient and highly available.

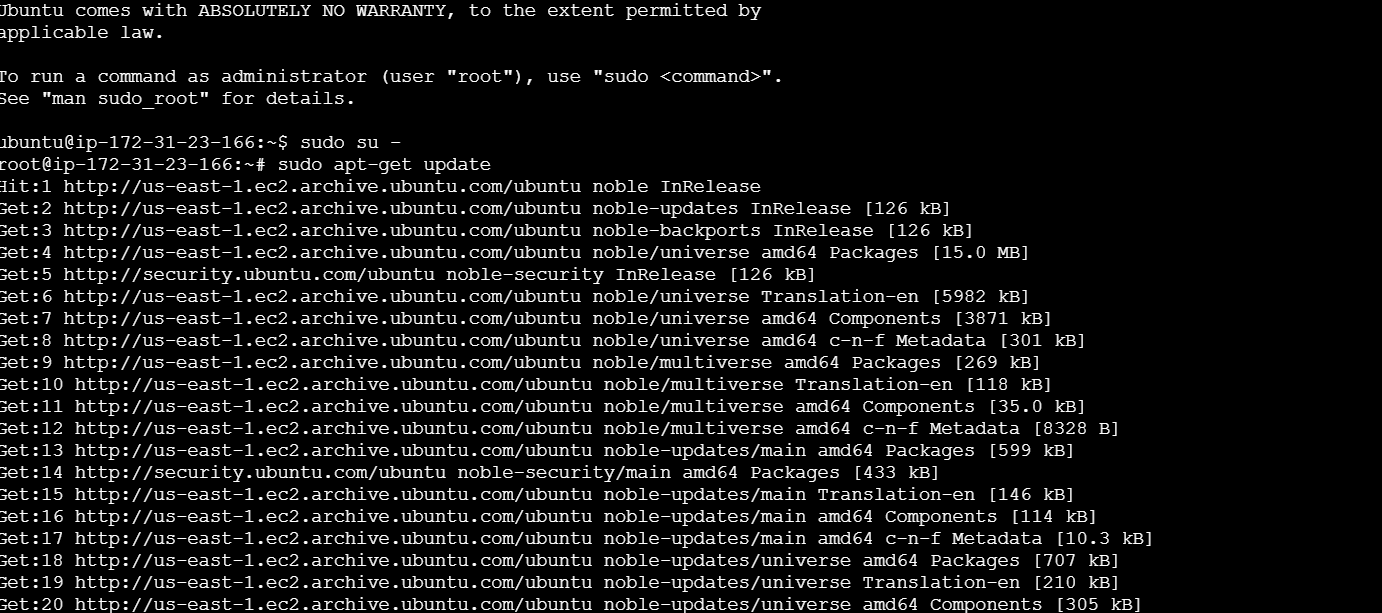


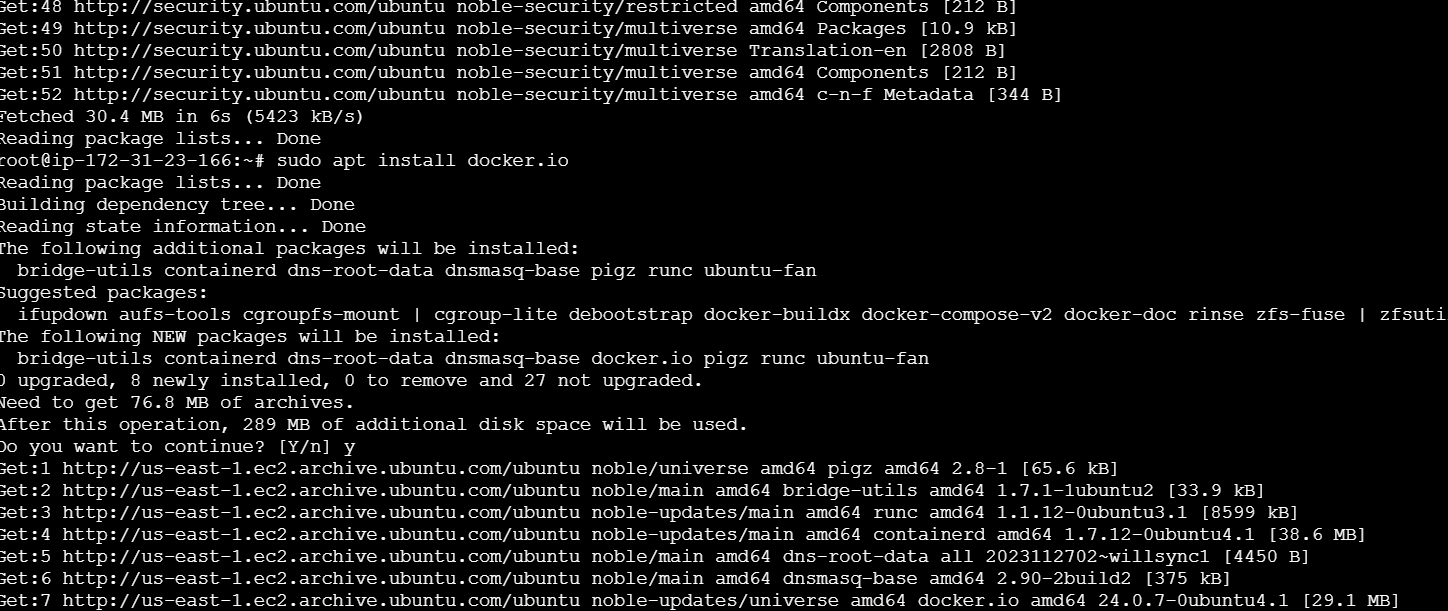
11. Now connect to the EC2 instance  


12. Now click on ec2 instance connect and connect using ec2 instance connect and go to the instance control command line  


13. This is how you connect to ec2 instance connect   
  
  
  
**Now lets move onto the next step installing docker**

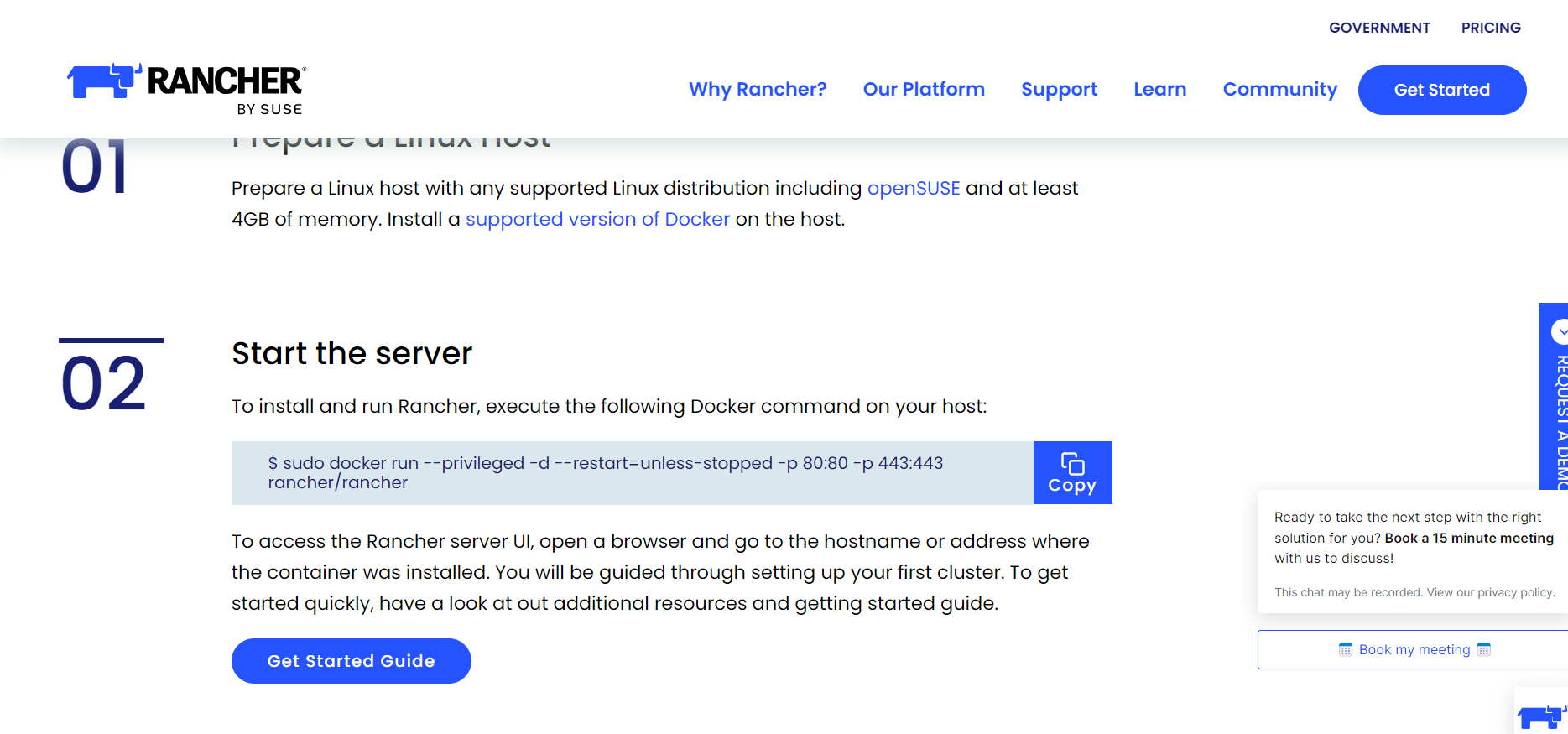
14. Now run the command sudo su – and sudo apt-get update

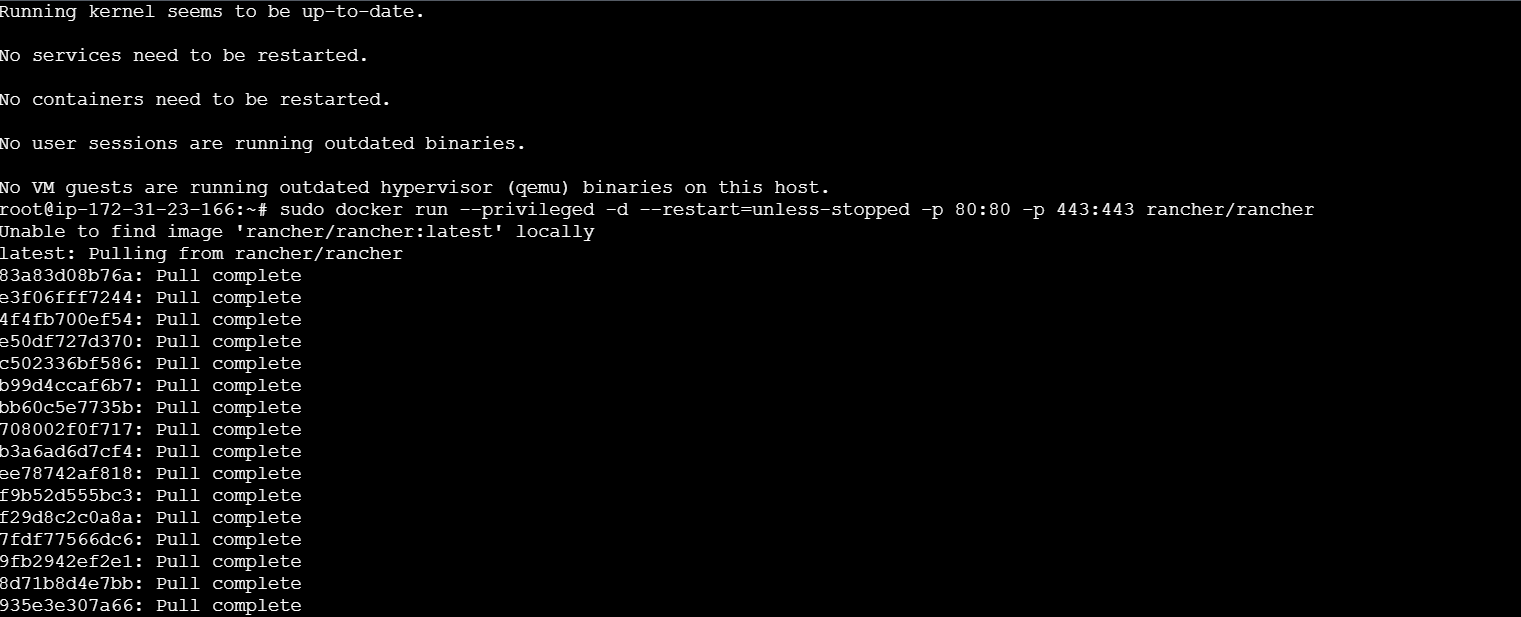


15. Now run sudo apt install docker.io and click on Y when prompted  


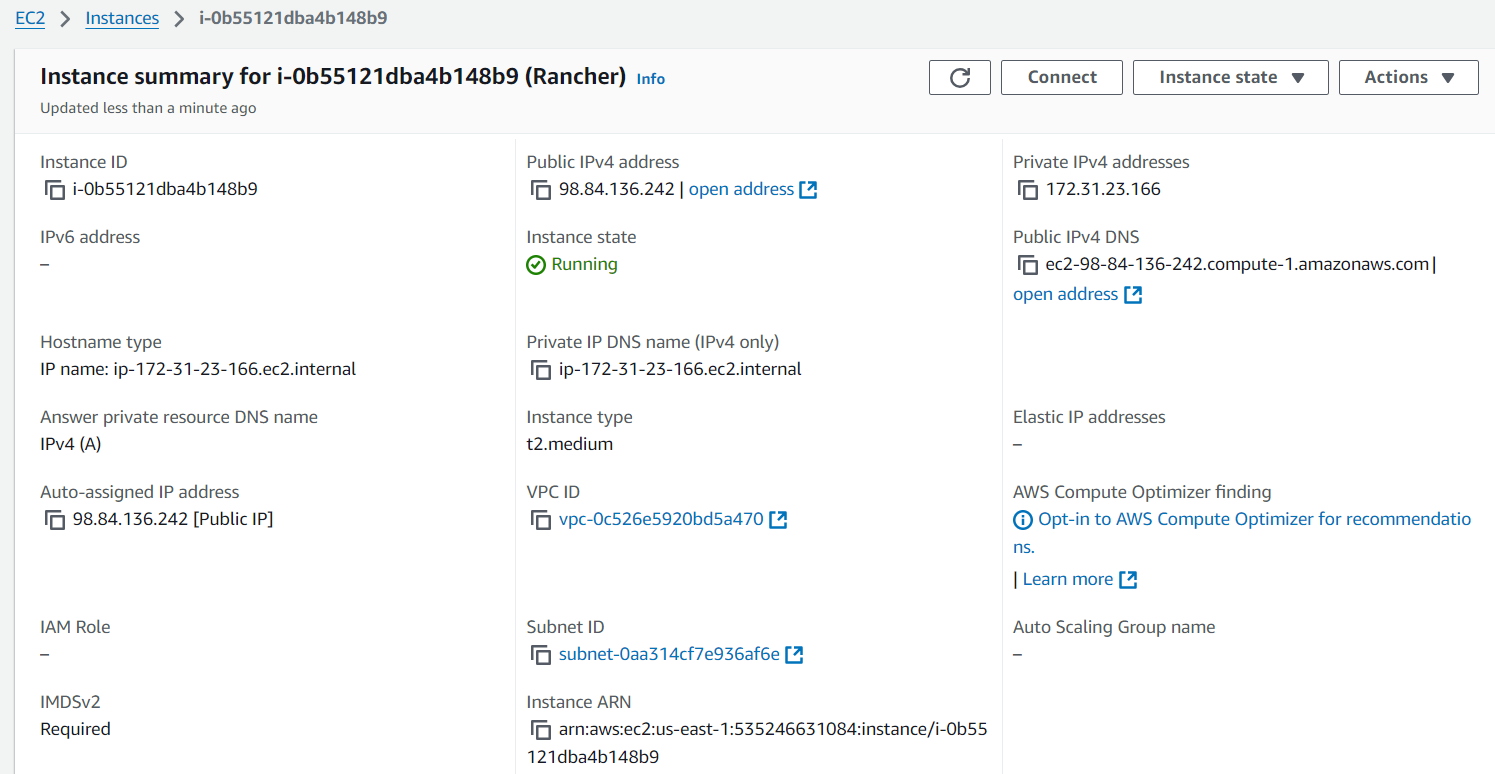
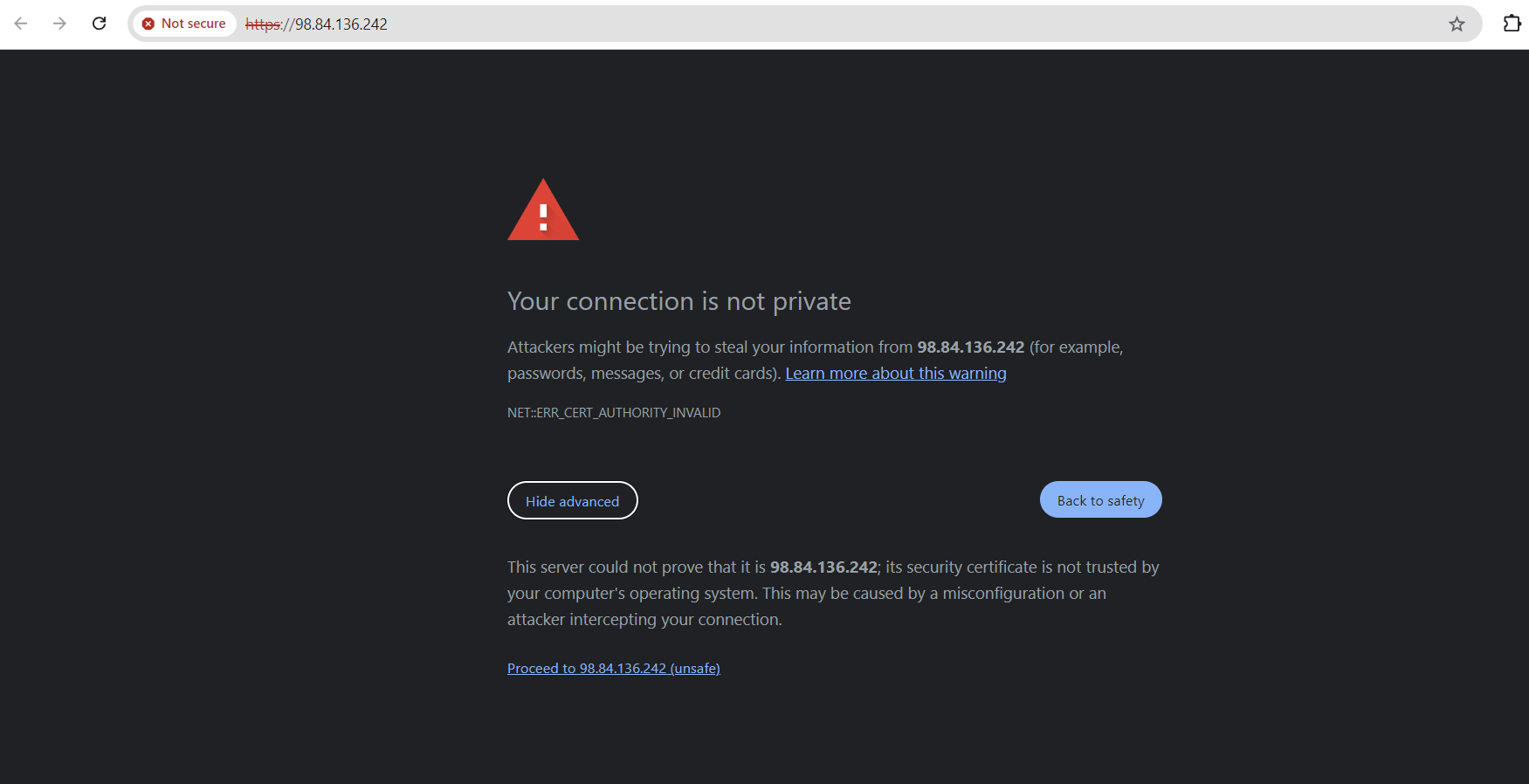
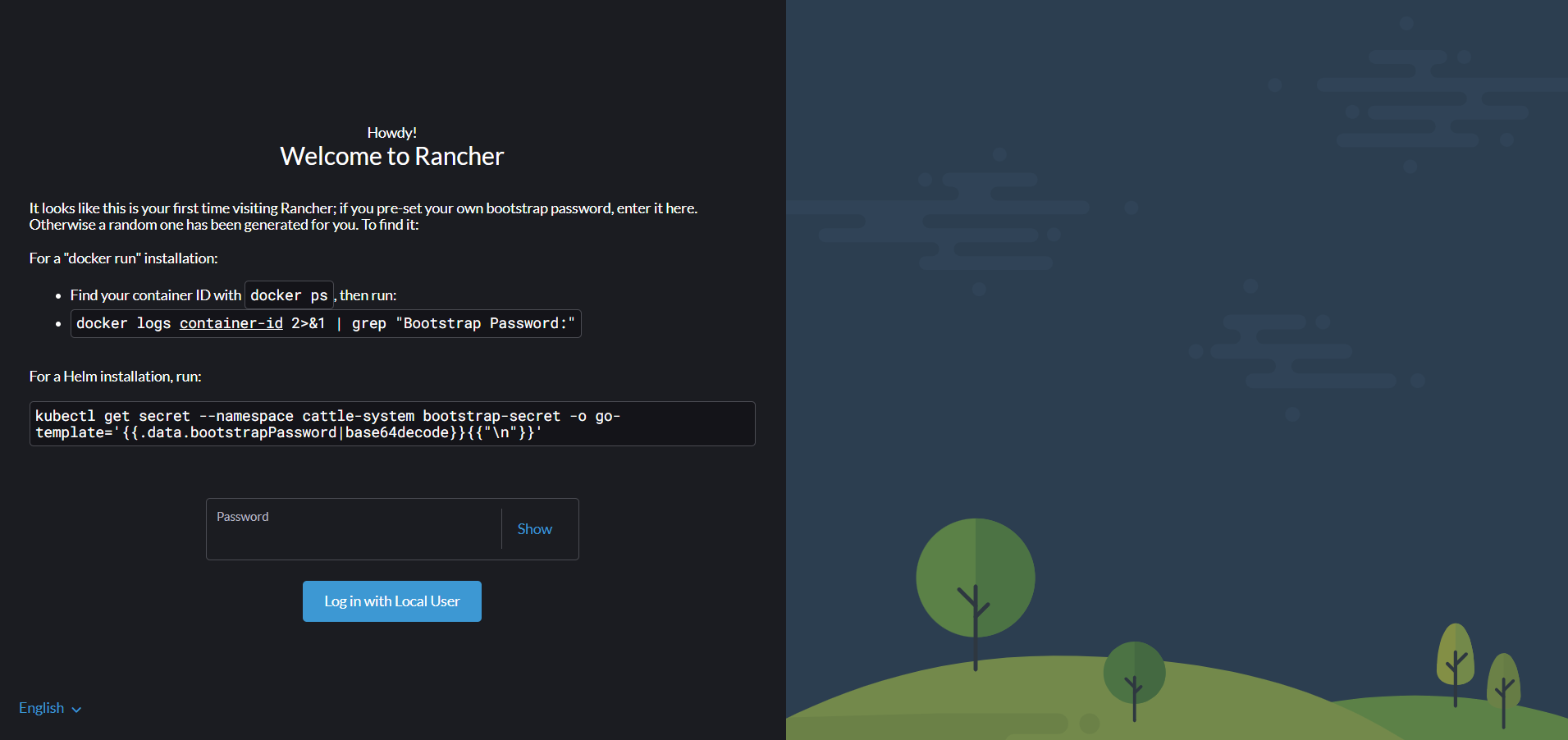
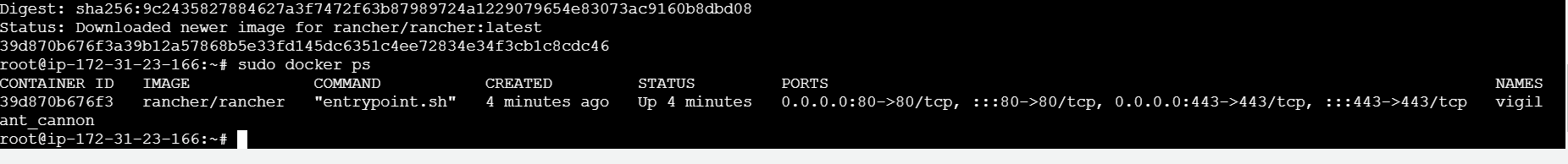
16. Now repeat the same for the second instance as well

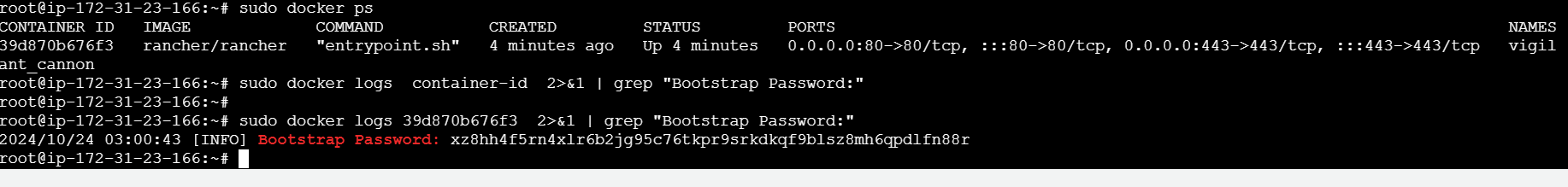
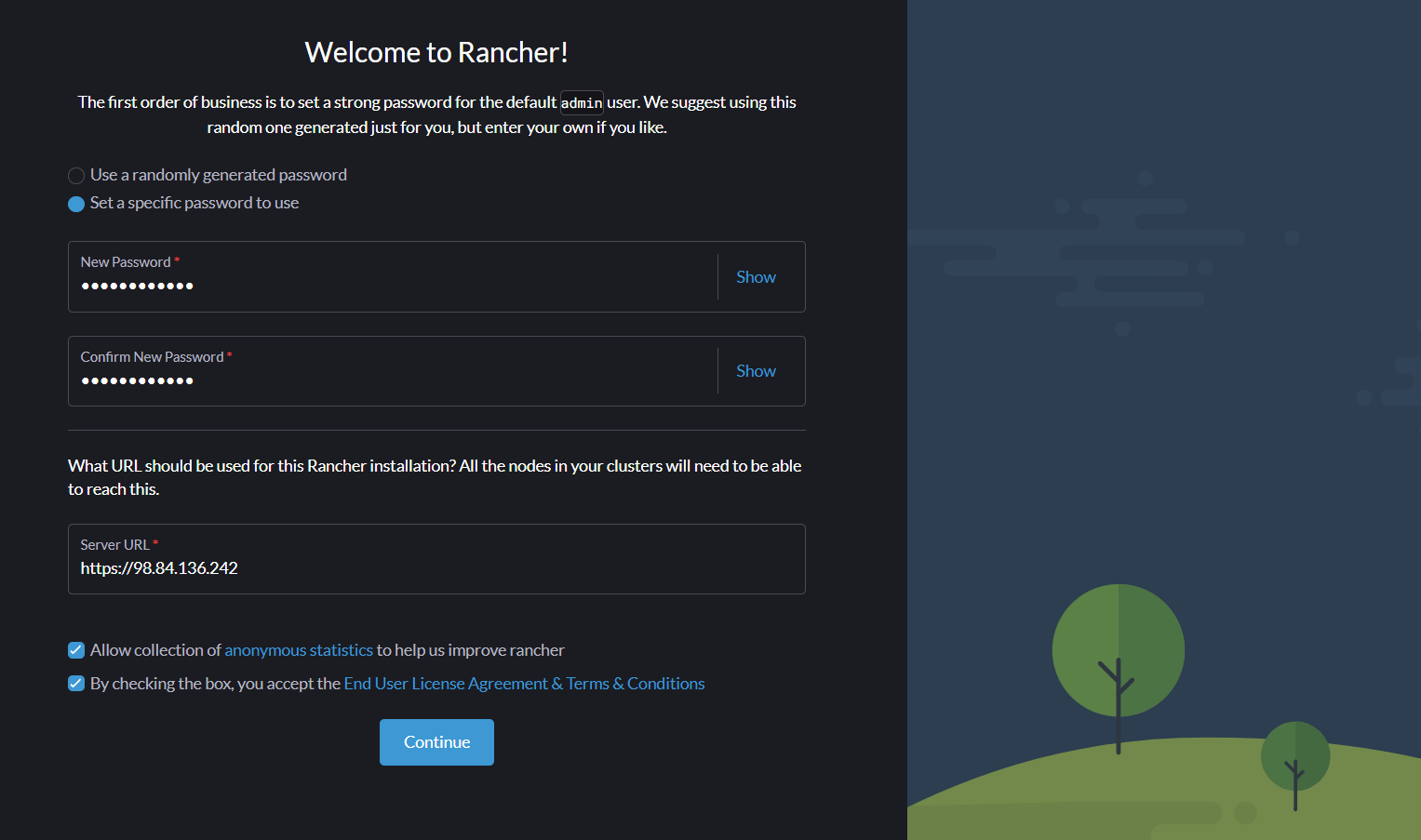
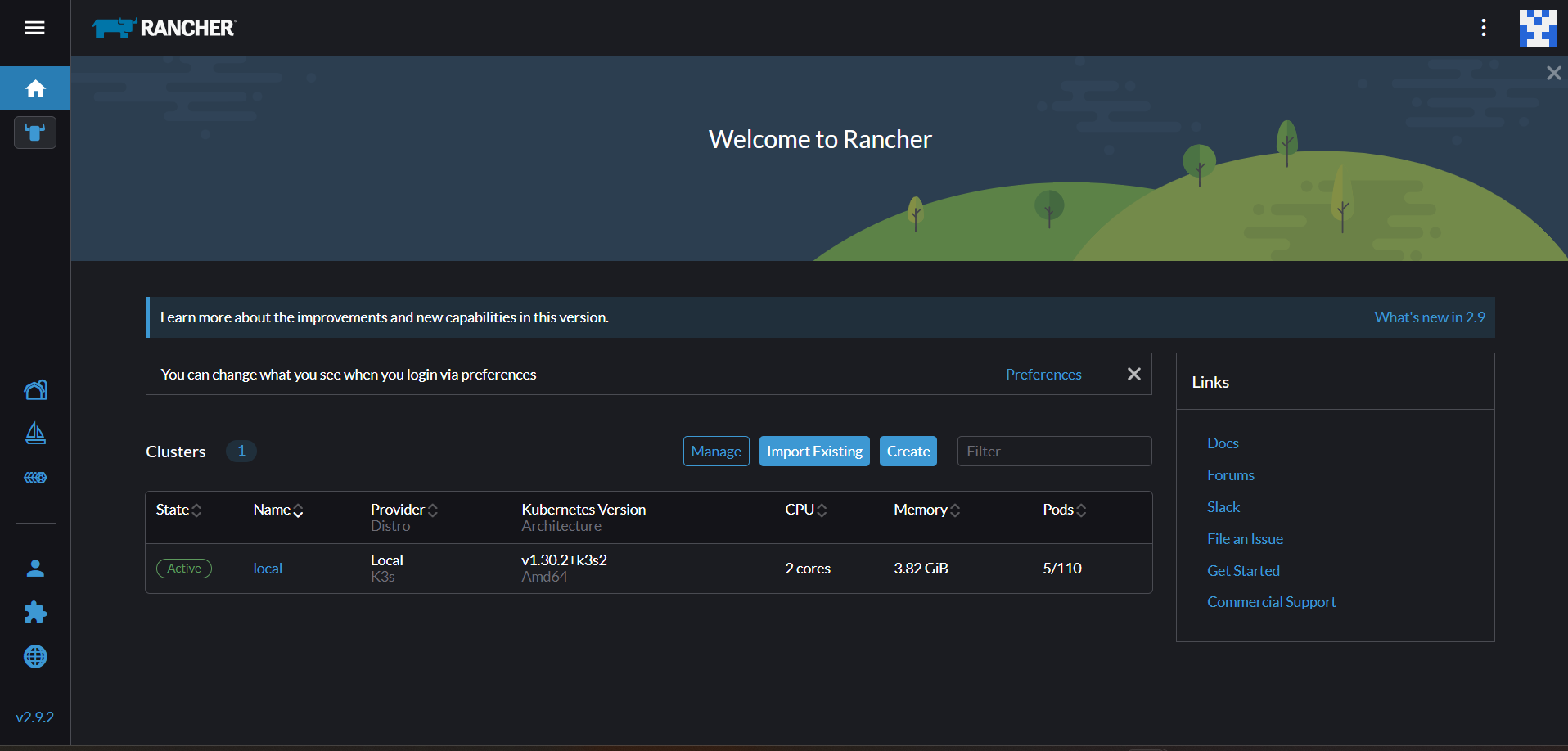
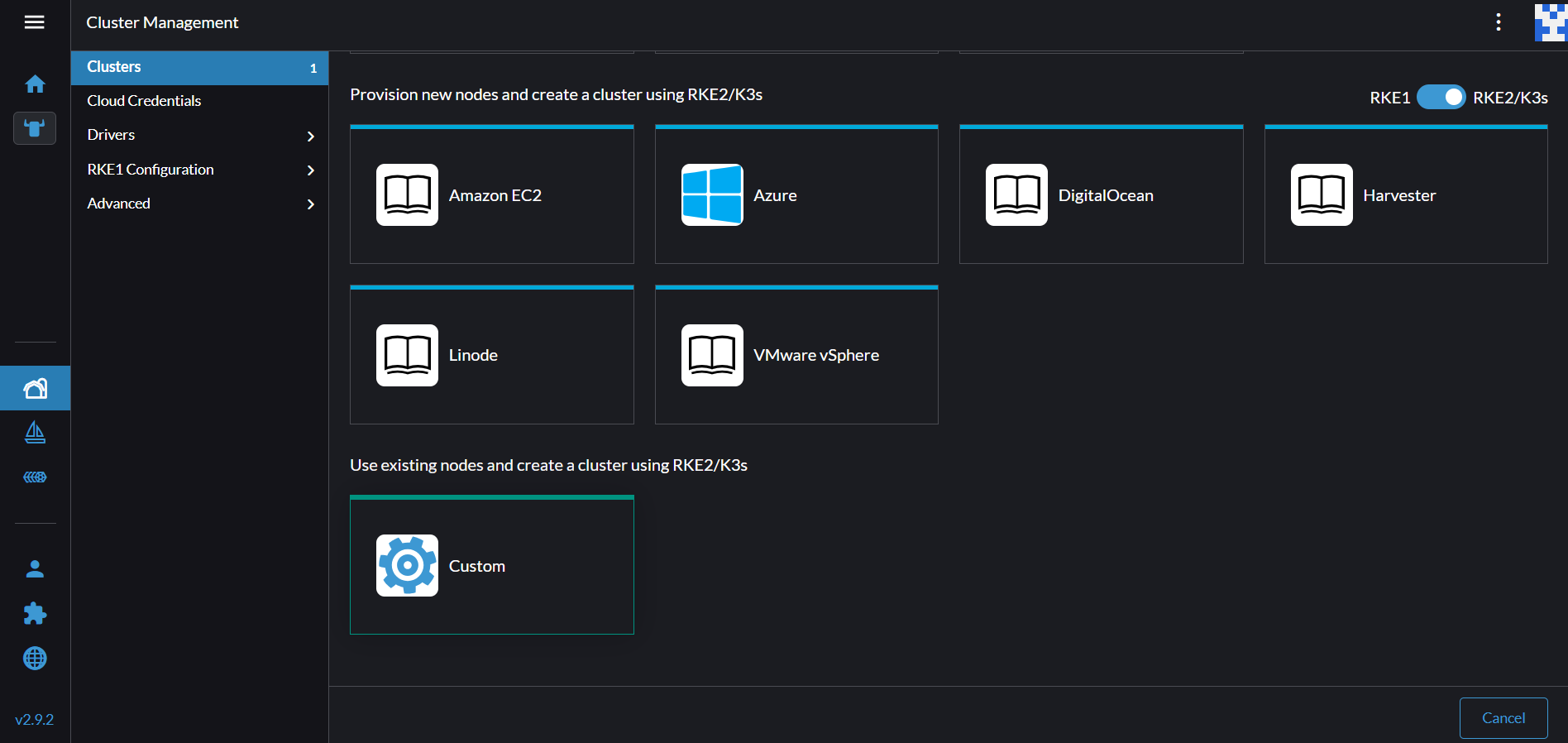
**Now we need to initialize rancher**

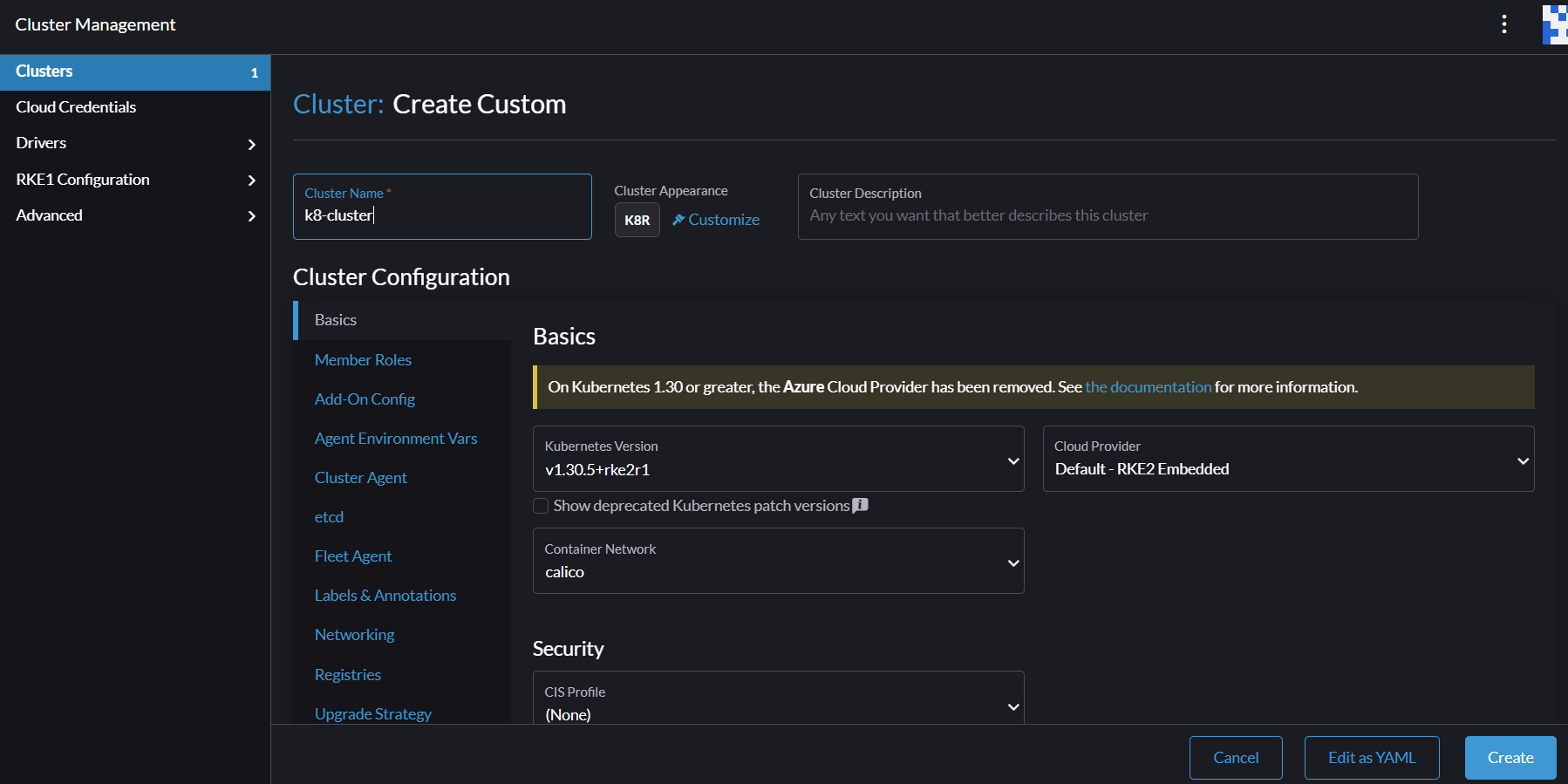
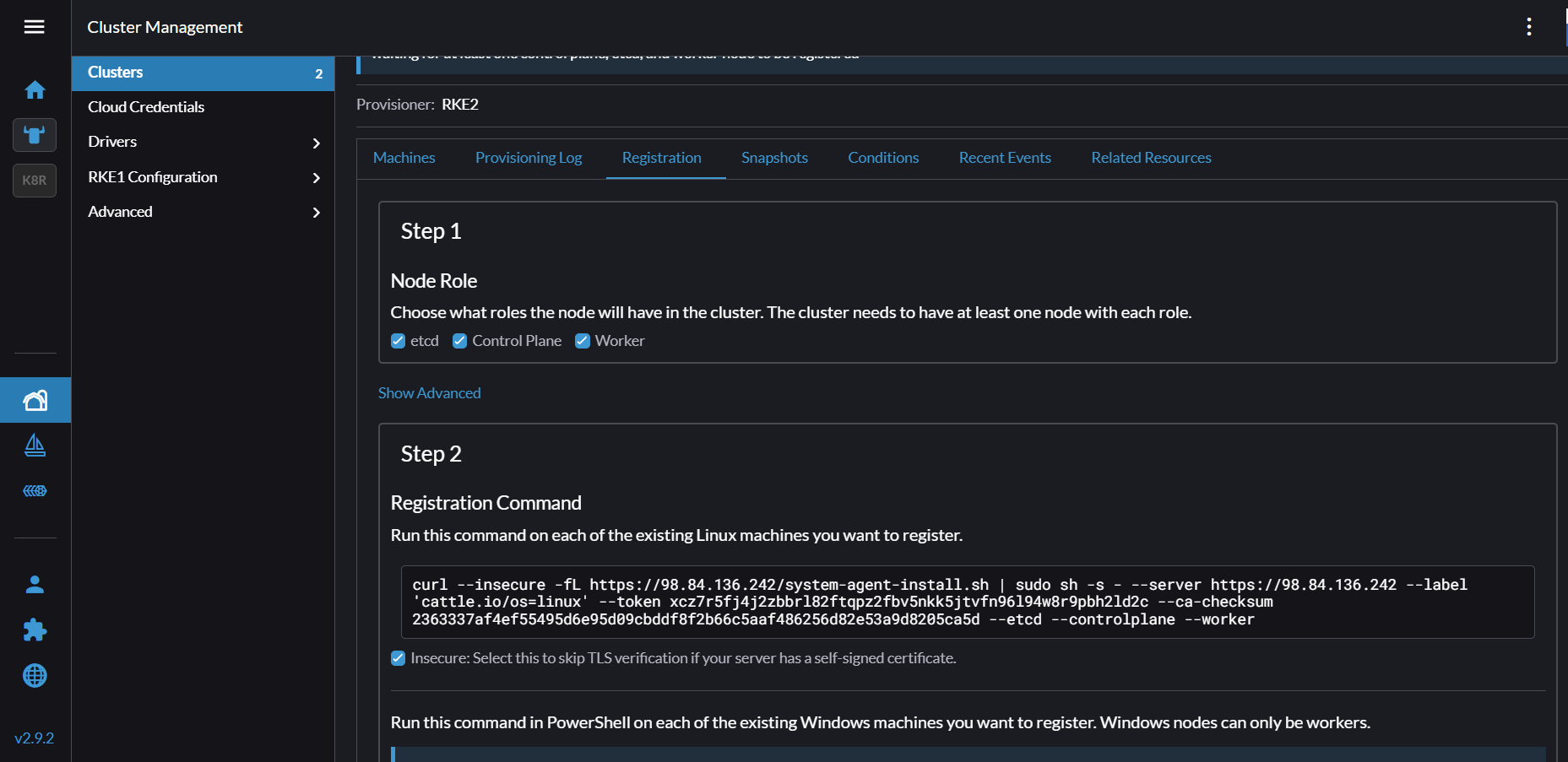
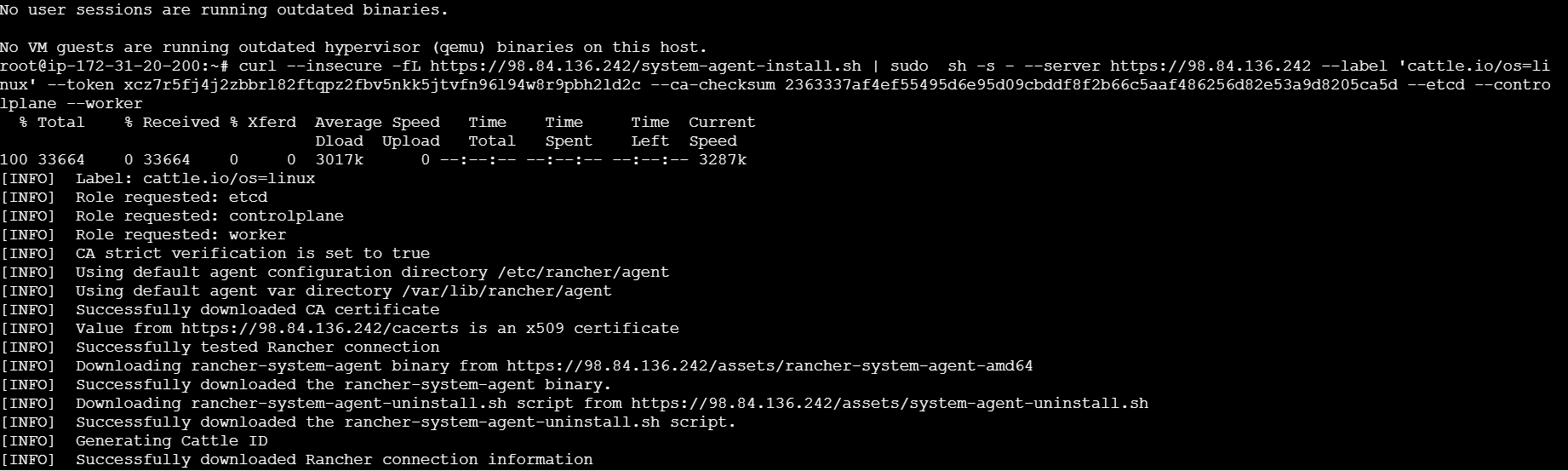
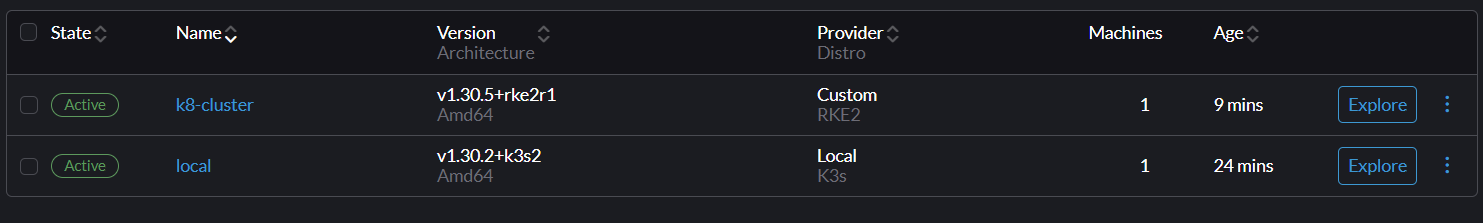
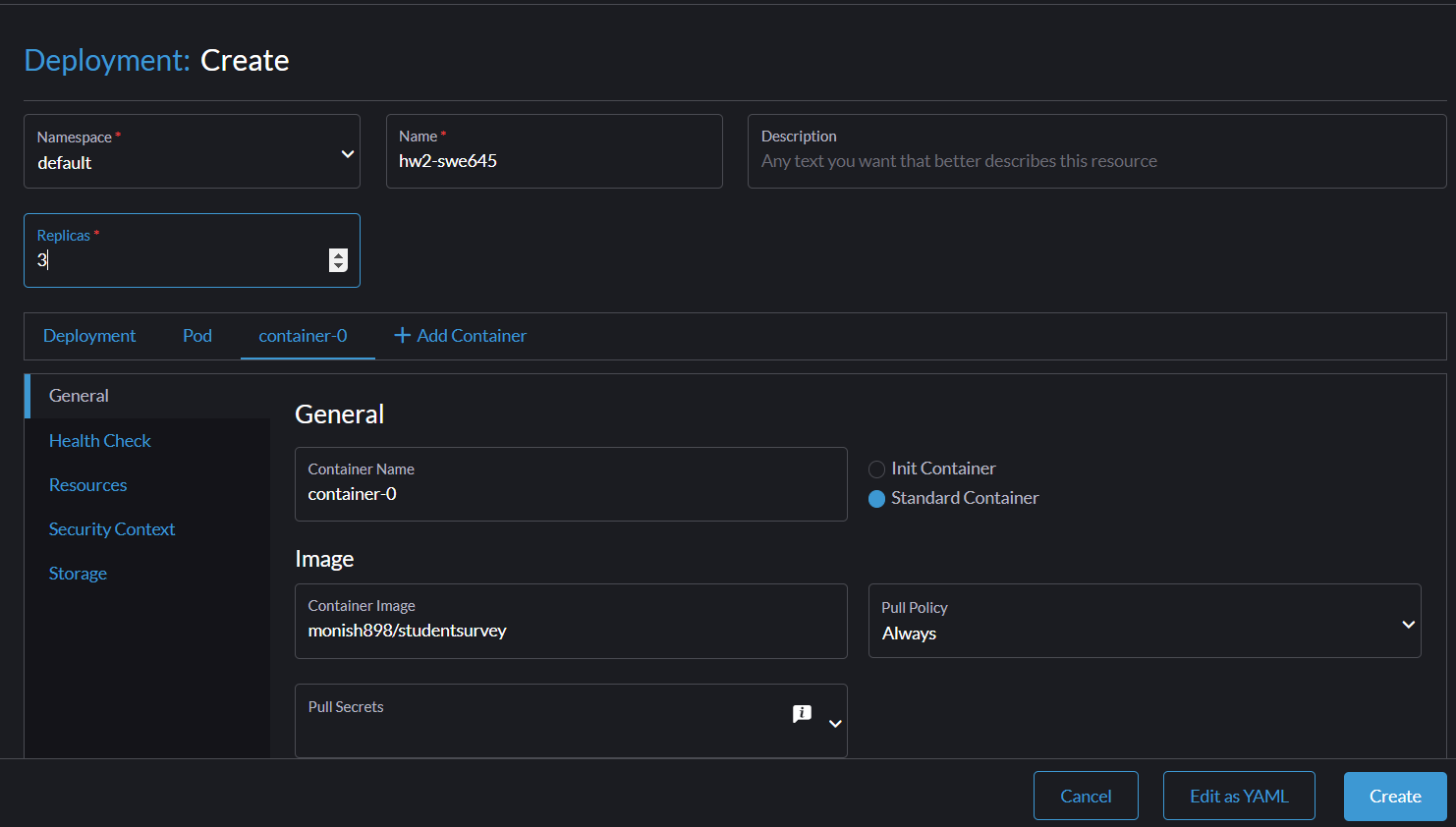
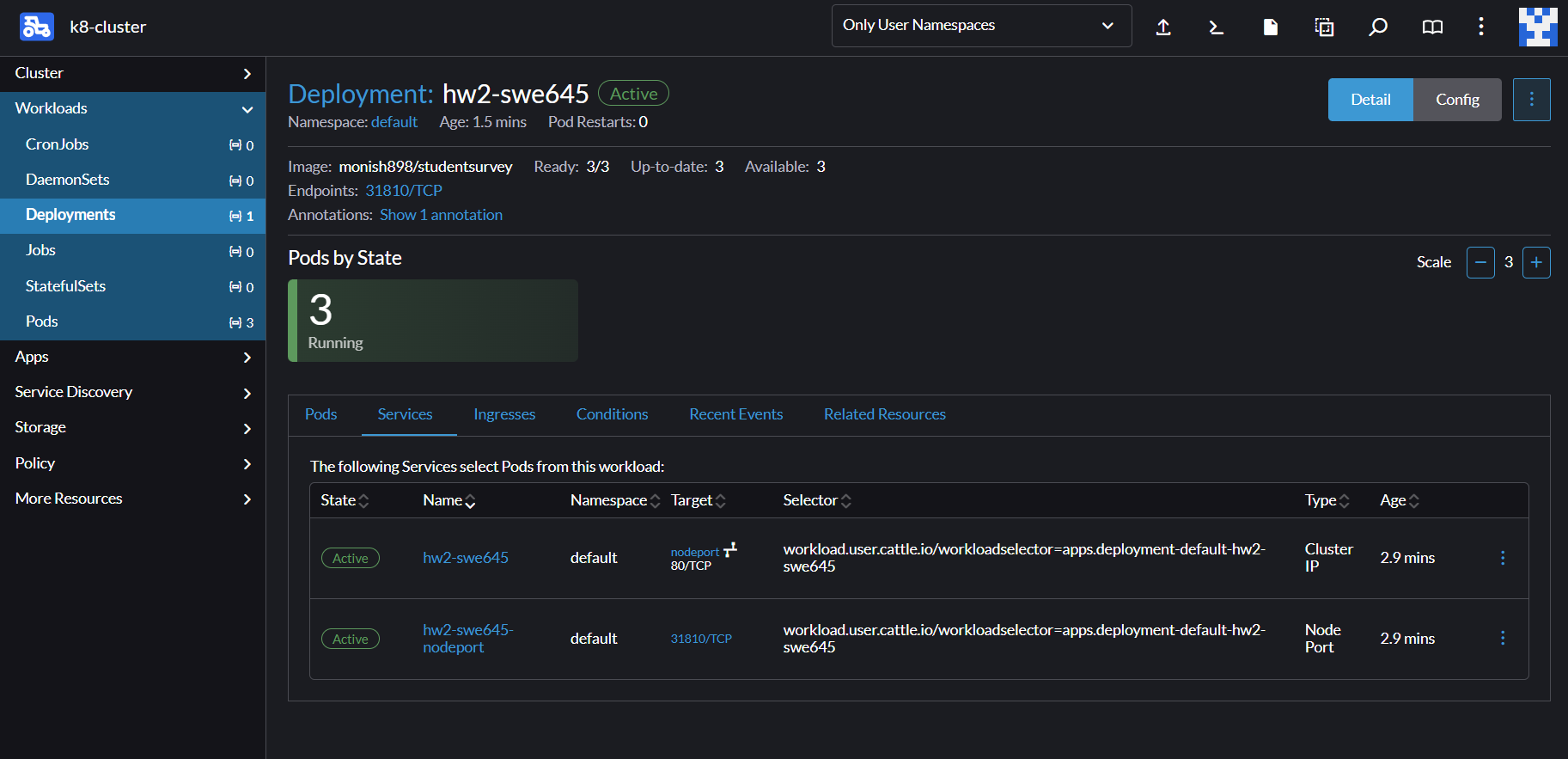
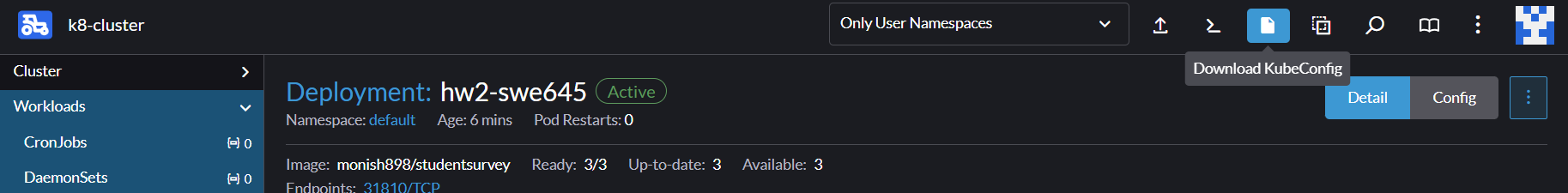
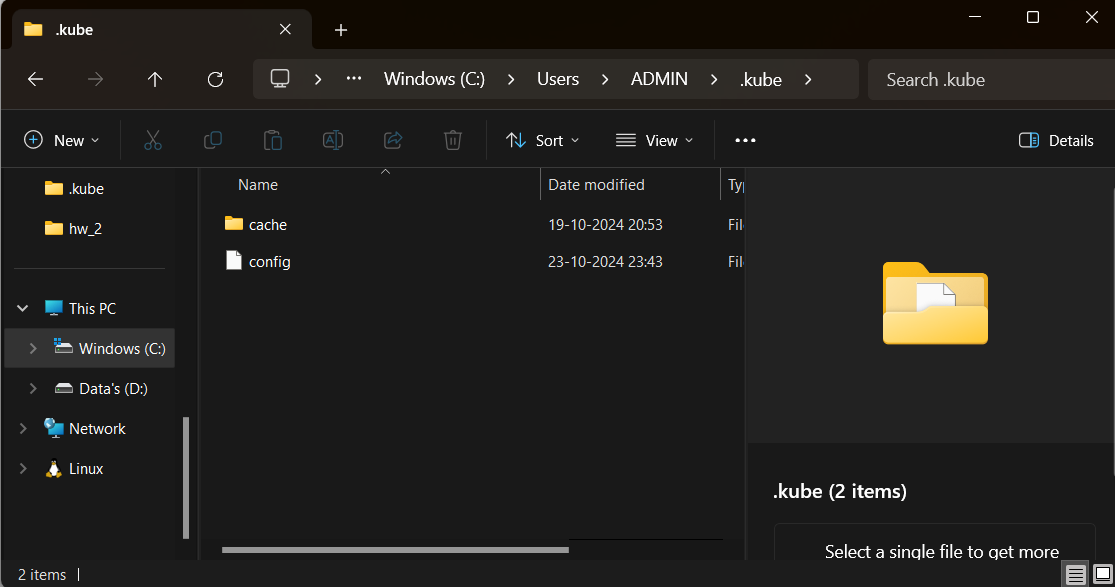
17. now go on to the link rancher.io and click on get started and scroll down to this page and copy the command.  
  


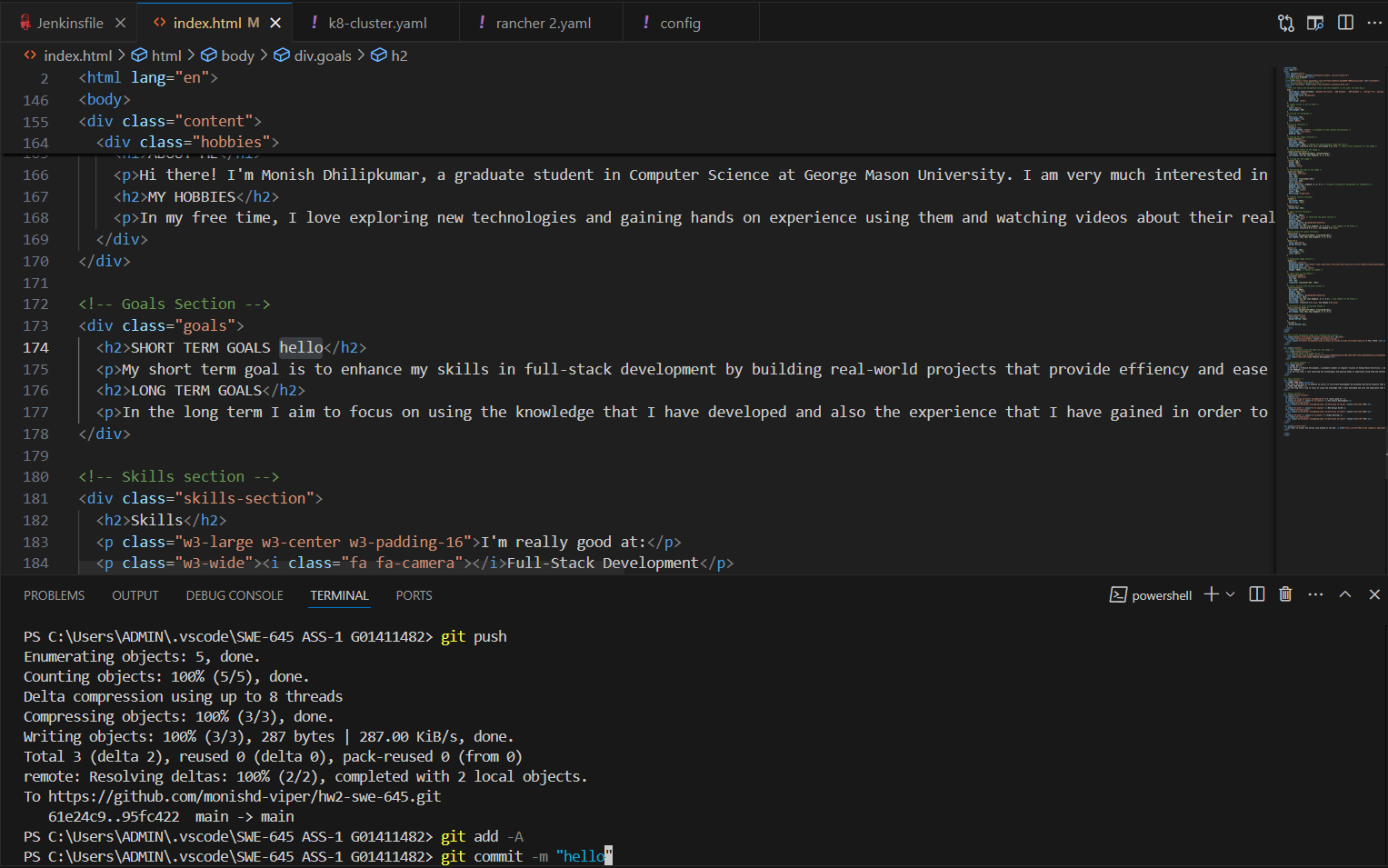
18. go to the first instance and run the command $ sudo docker run --privileged -d --restart=unless-stopped -p 80:80 -p 443:443 rancher/rancher  
  




19. now click on the public ipv4 address   
  
  
20. Proceed to the link by clicking on the link and bypass the warning  
  
  
  
21. It will open a tab with rancher   
  
   
  
22. Run sudo docker ps and check if rancher is initialized and check the ports  
  
  
  
23. Copy the command from rancher and paste it and run the command with the right container ID and you will get the bootstrap password and paste it and login in with local user

  
  
24. Set a password and tick the check box (by checking the box) and click on continue  
  
  
25. Now you will be able to access the rancher site.  
  
  
  
26. Now click on create cluster options and select custom.  
  
  
  
27. Now give a name for the cluster and click on create and the other options remain the same.

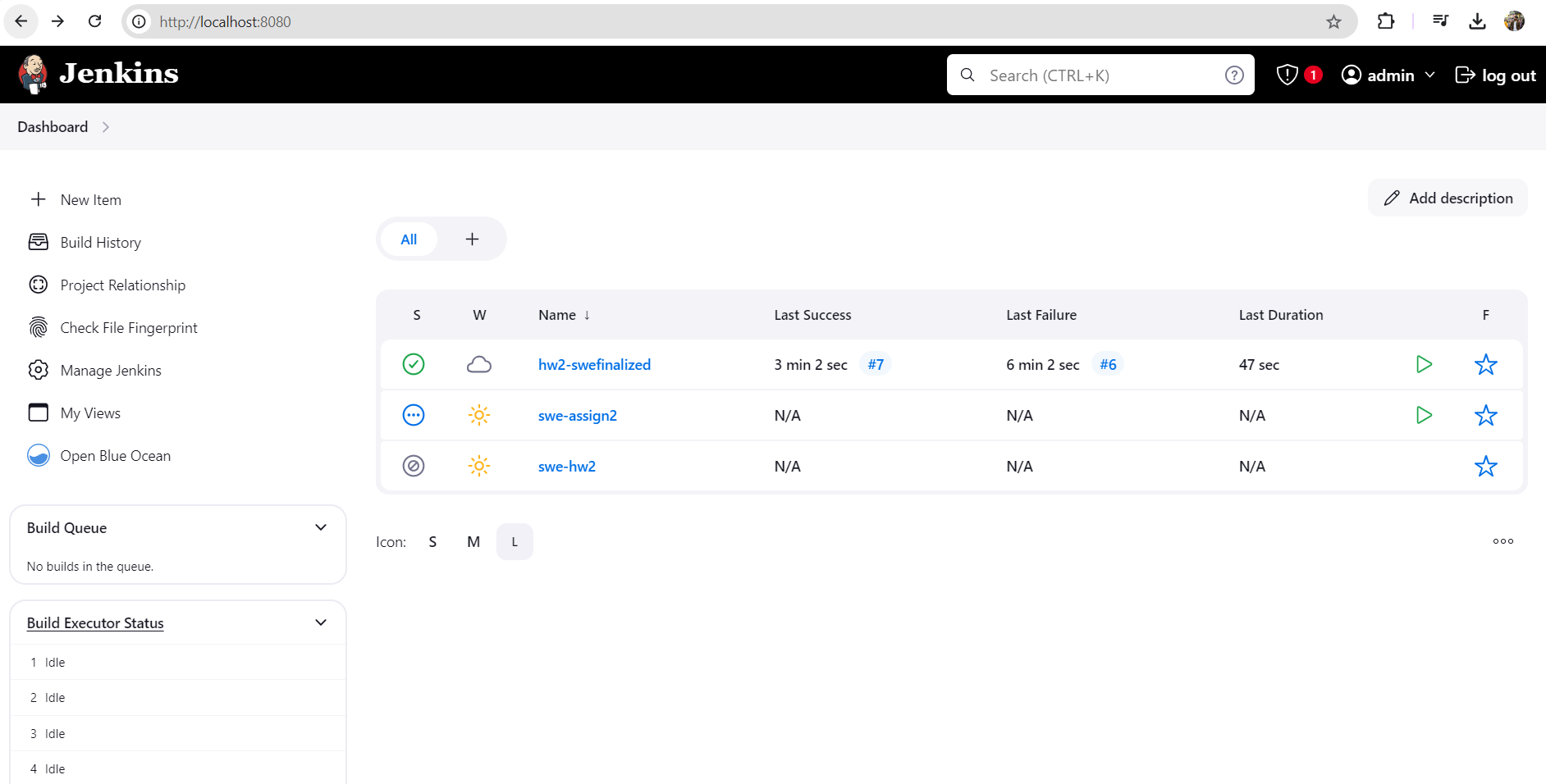
  
  
28. Now the k8 cluster is created and it will be updating as you need to click on all 3 nodes and click on the insecure checkbox and then copy the curl command and paste it in instance 2  
  
  
  
  
  
  
29. It will create a cluster and it will be in the provisioning stage until it gets active   
  
  
  
30. After it turns active click on explore  
  
  
  
31. Go to cluster and the workloads and then to the deployment page and create and then in the next page you need to set the namespace as default, the name as hw2-swe645 and set the replicas as 3 and also the container image exactly as pushed in docker hub  
  
  
  
32. Now under networking in ports add a nodeport and give the port number as 80. It will automatically fill the listening port and then click on create and wait for it to create a cluster.  
  
33. Now in the cluster go to services and click on the nodeport 80 and launch the link now you can see that our website has been established.  
  
  
  
  
  
34. Download the kubeconfig and you will download a file called k8-cluster.yaml and open the file and copy the content and paste it into the config file in .kube  
  
  


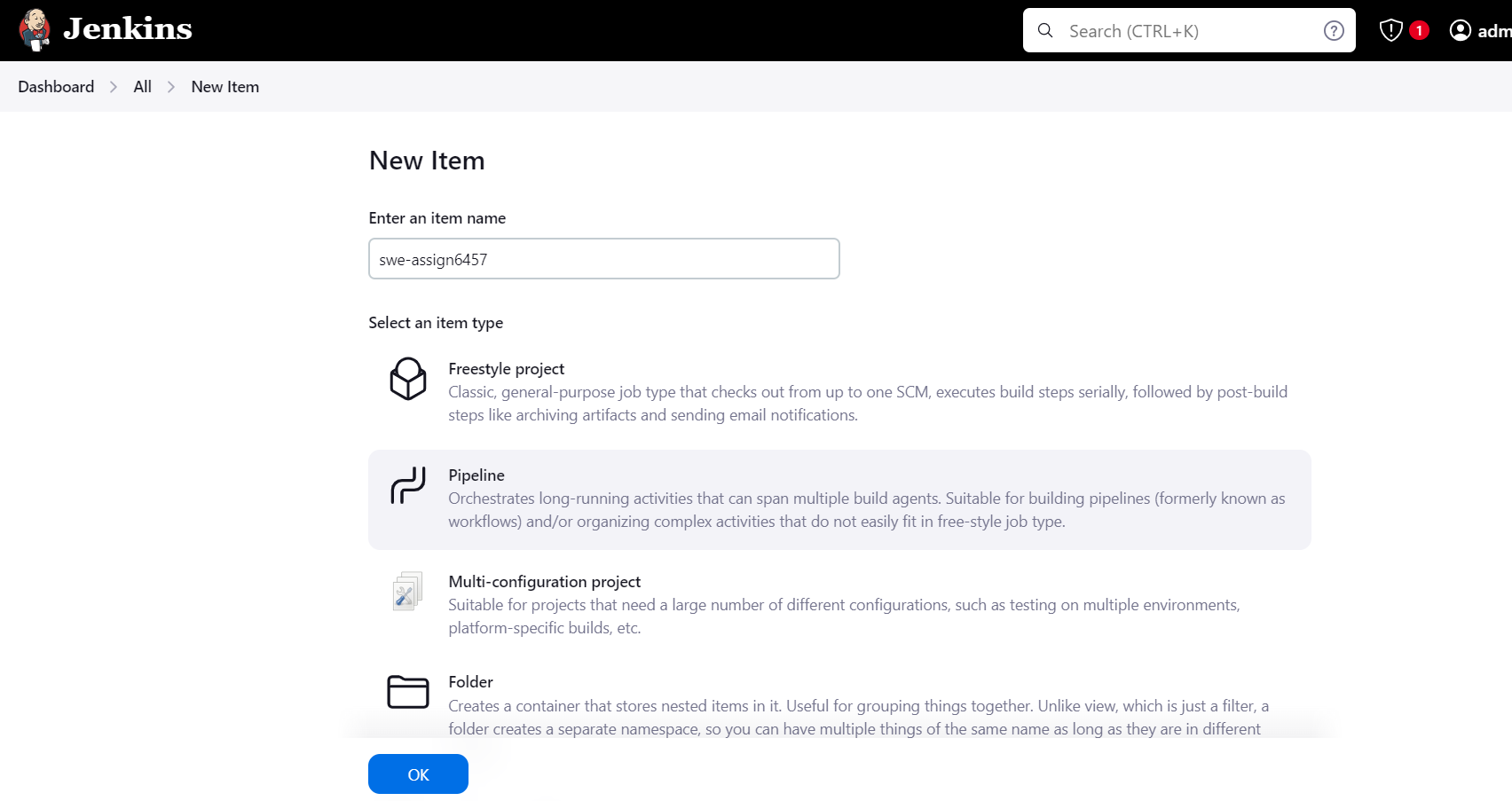
35. Now you can make a change in the index.html file and run the commands git add -A, git commit -m "hello" and git push  
  
  
  
  
  
  
  
**Jenkins**Installation of Jenkins  
1. Download war file from this link  
<http://www.jenkins.io/download/>

2.Open command prompt and go to the folder where Jenkins is present

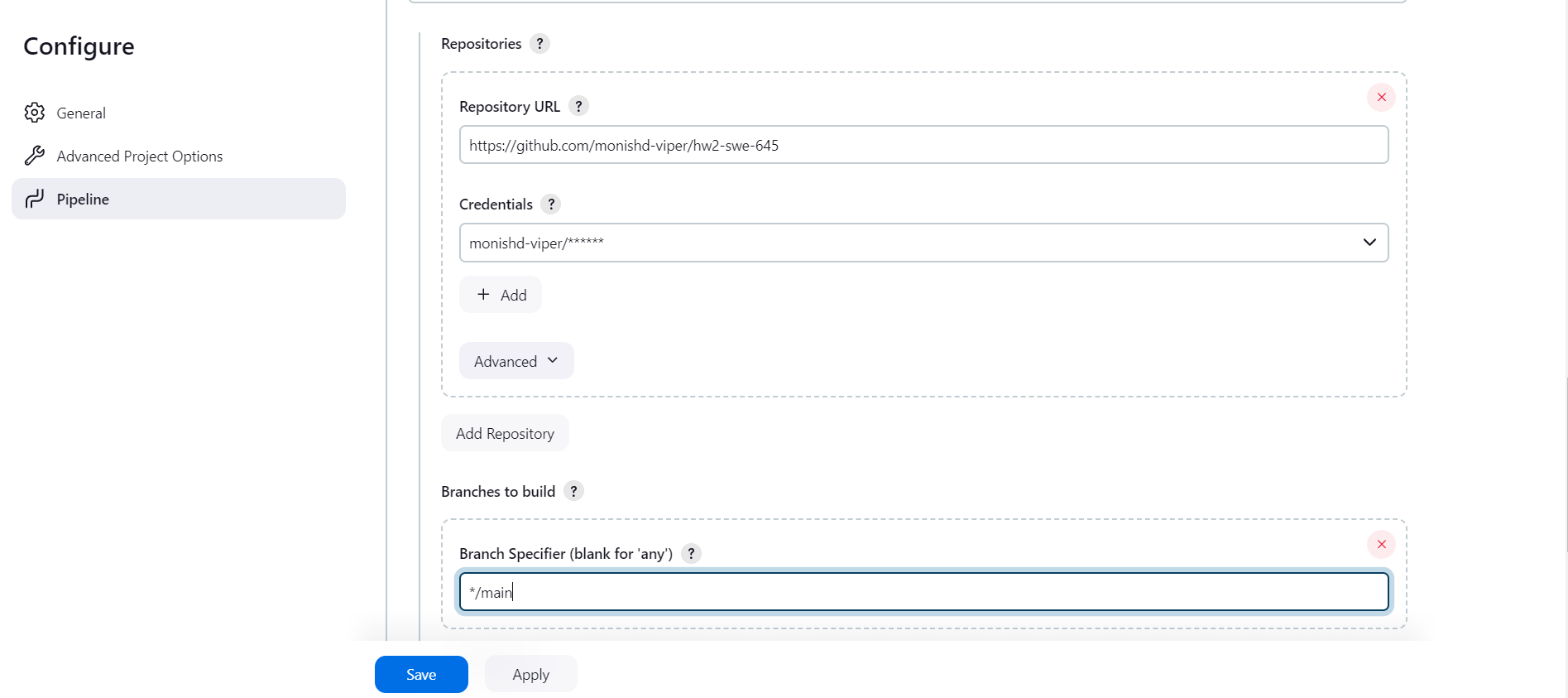
3. Run the command java -jar jenkins.war

4. it will run and give a password and copy it  
5. Go to localhost:8080 and wait until the Jenkins page appears and paste the password on the page and sign up  
6. Need to create a Jenkinsfile in order to automate this project or you can give it directly while creating a pipeline

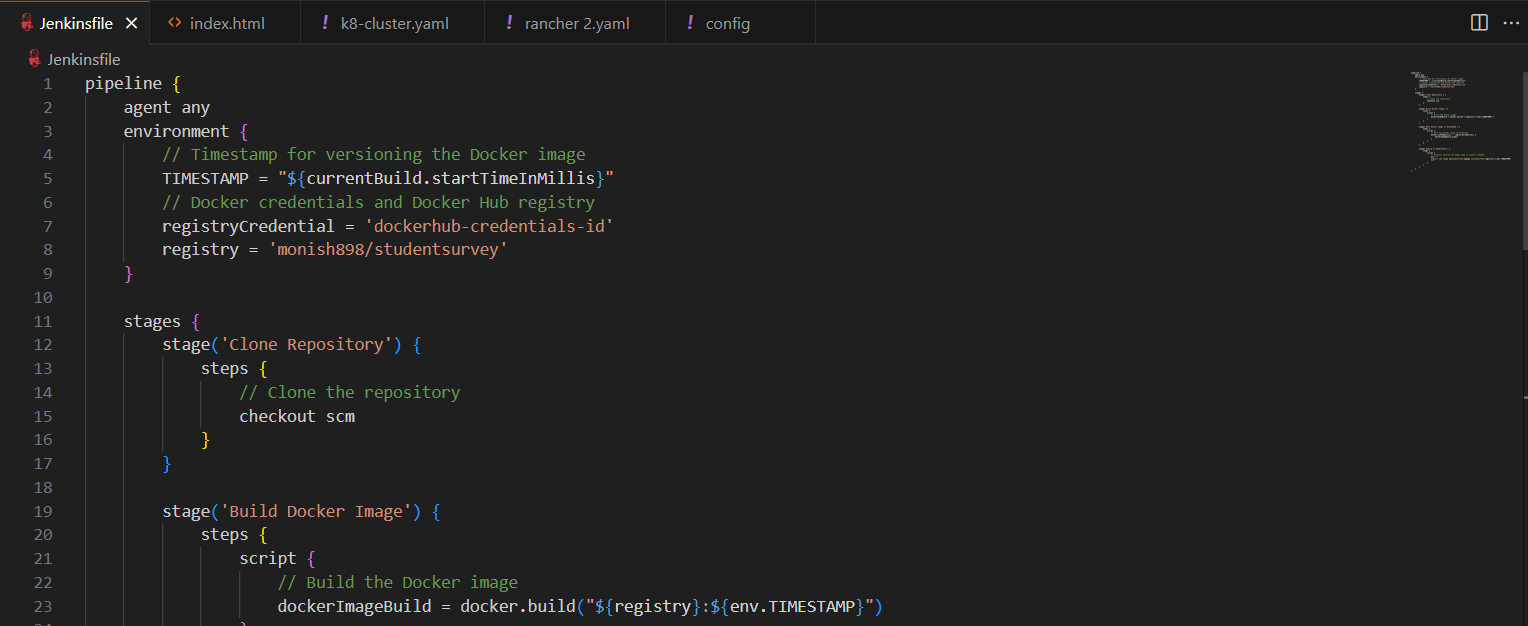
7. Browse to the localhost:8080 and wait until the Jenkins page appears and click on new item

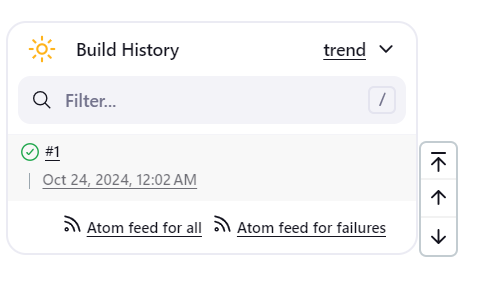
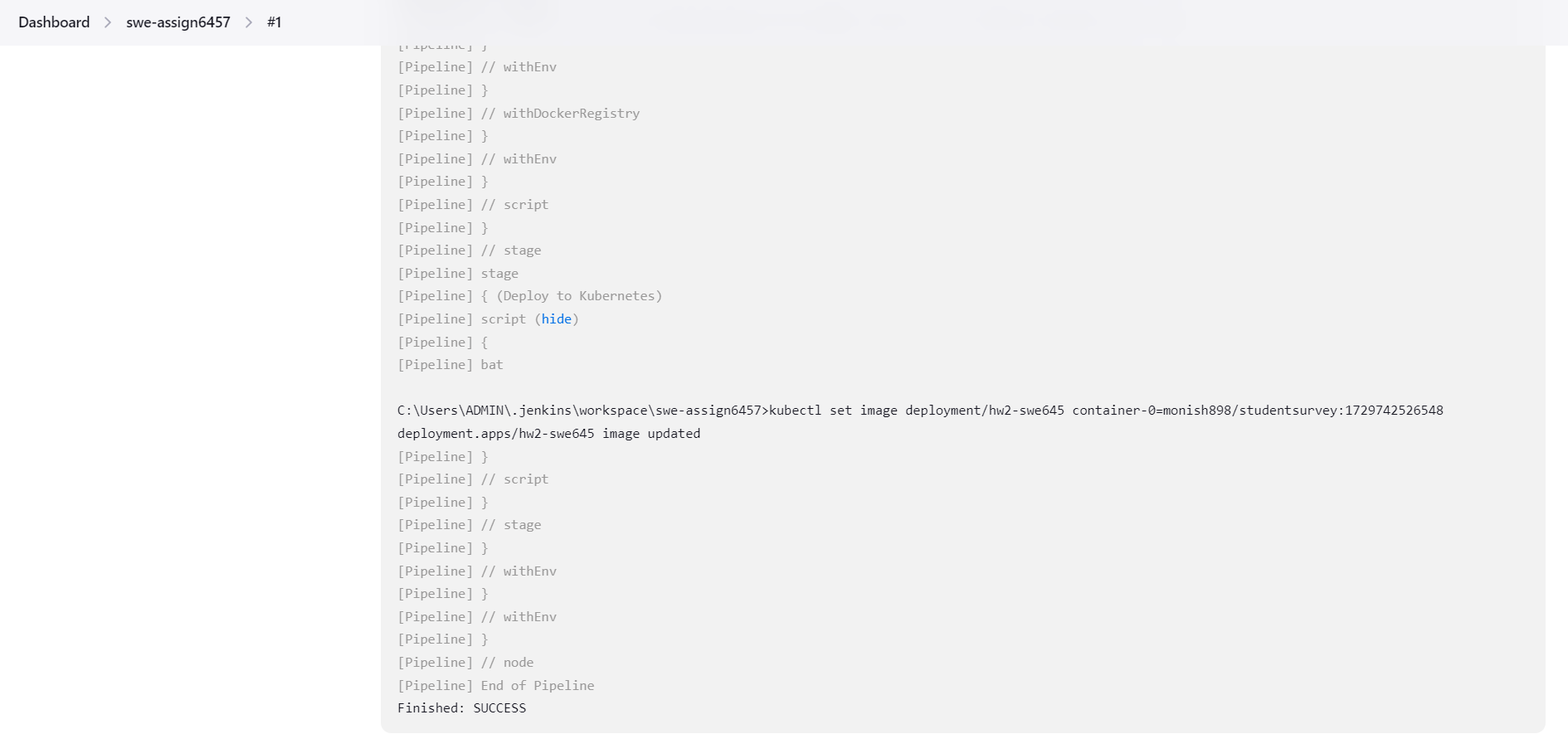
8. Assign a name and then choose pipeline and click on ok  


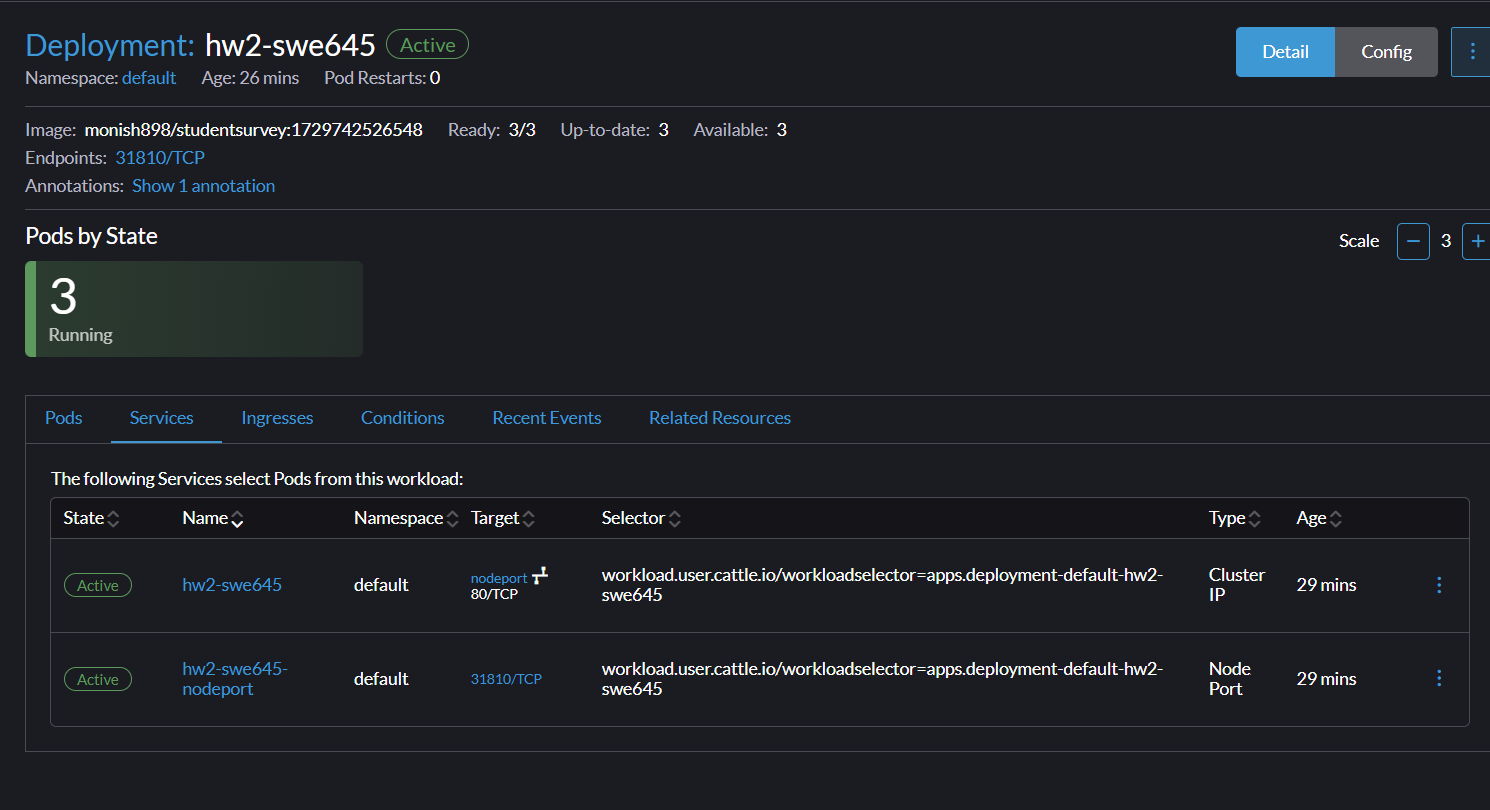
9. Scroll down to build triggers and select poll SCM and type “\* \* \* \* \*”. It means it will check the build for every minute whether changes or not.  
  

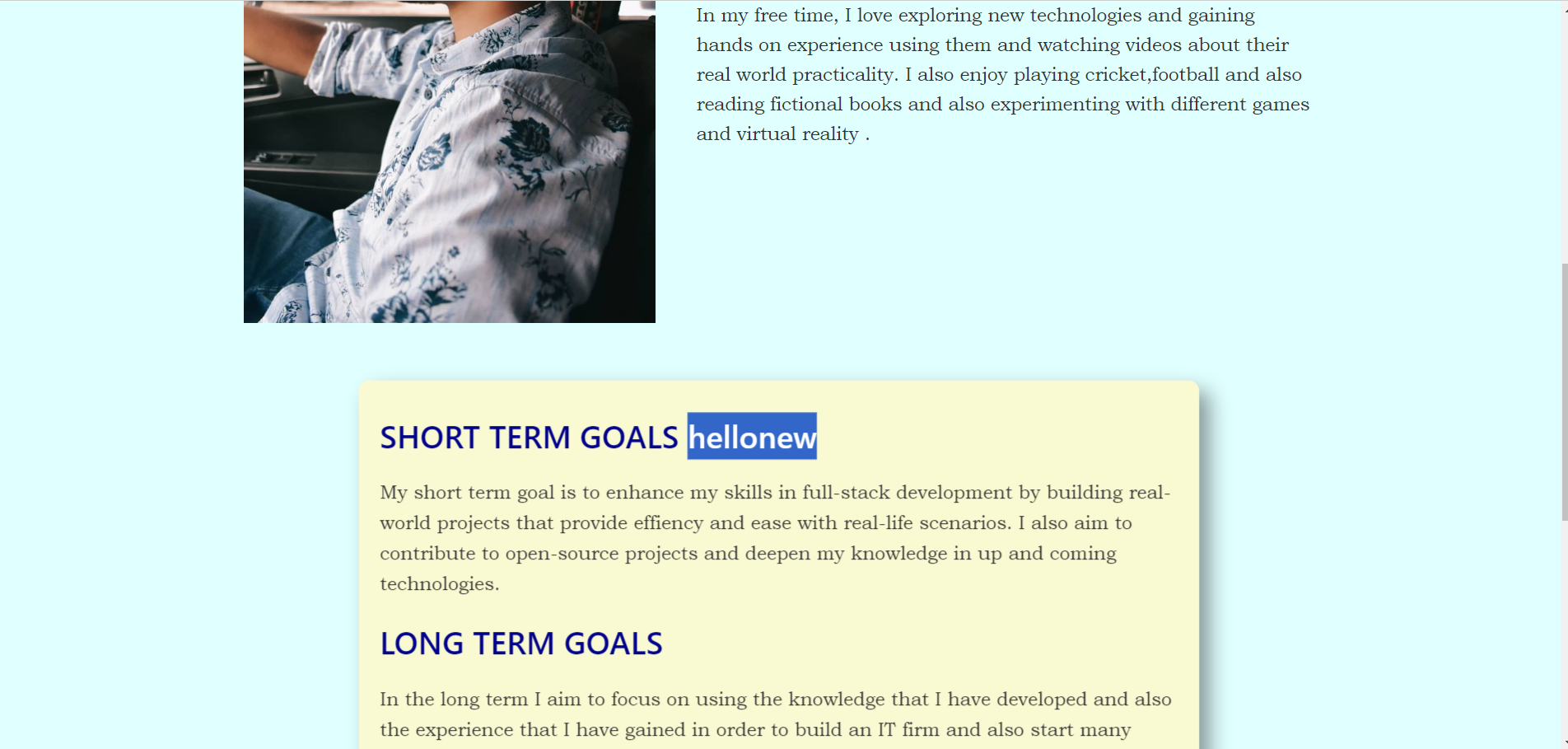

10. Scroll down to pipeline and select definition pipeline script from SCM. Select SCM as Git as I have deployed the project in Git. Now enter the git repository link and make sure that you have added git and docker credentials in Jenkins and select the git credentials and also change master to main and check if the path is given as Jenkinsfile and click on save.  
  


11. Go to the project folder and create a Jenkins file  
The four steps include cloning the repository, building the docker image where it will create a image with a tag.  
Then it will push the image to Dockerhub  
In deploying to pod it will update the image in the deployment.



12. Also make sure that you have downloaded the kubeconfig.yaml file from rancher and paste the data to the config file or create a config file.  
  
13. After you enter the specifications in the jenkins file you can see the build history and check the build status and after you get a green tick the build has succeeded.  
  
  
  
14. Under console output you can view that build has finished and the output is a SUCCESS.  
  


15. Now you can go to services and open nodeport   
  


16. Now you can see that the changes have been implemented.  
  


So this is the end of the assignment and the entire steps in the assignment have been mentioned throughout.