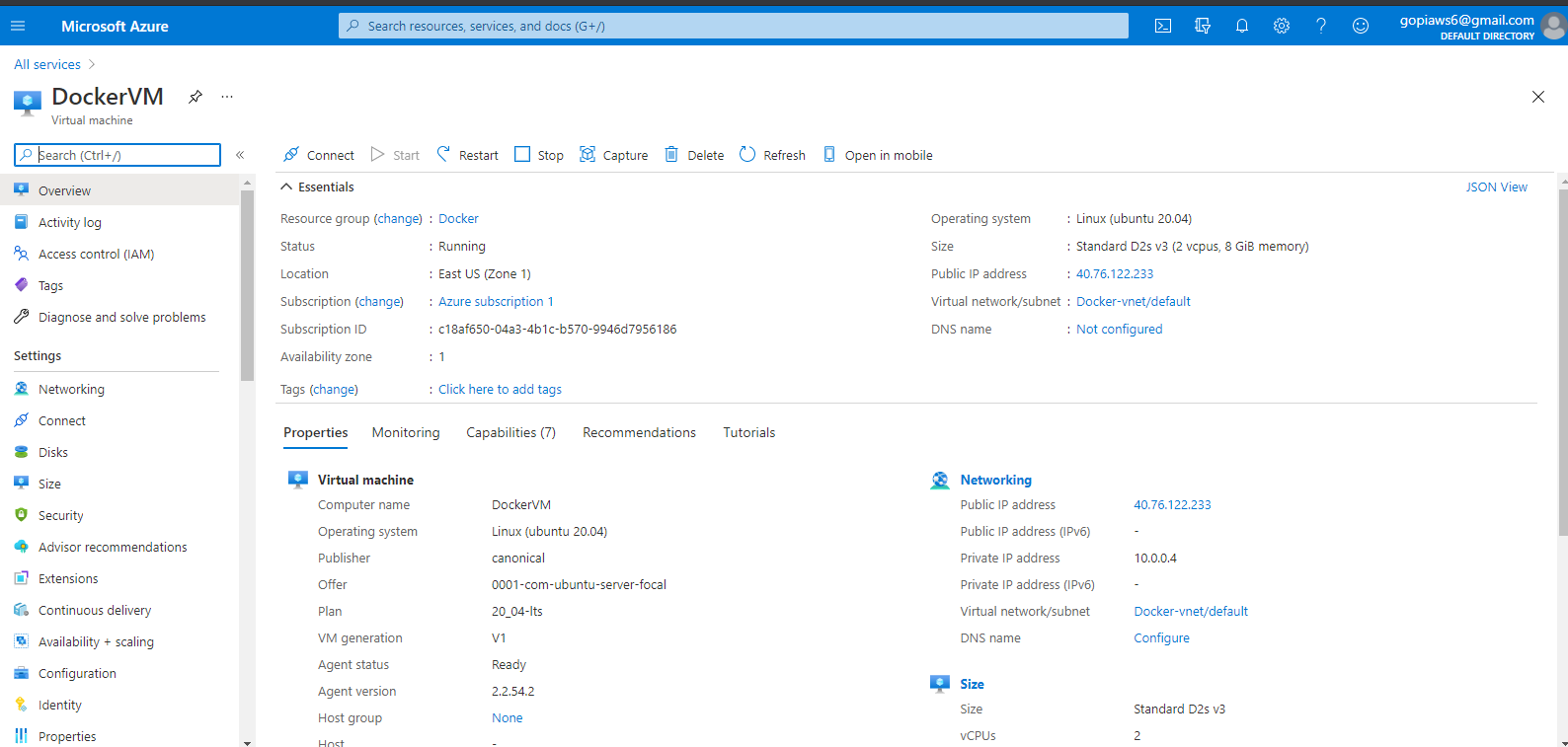
**BankiFi Lead DevOps Technical Test Documentation:**

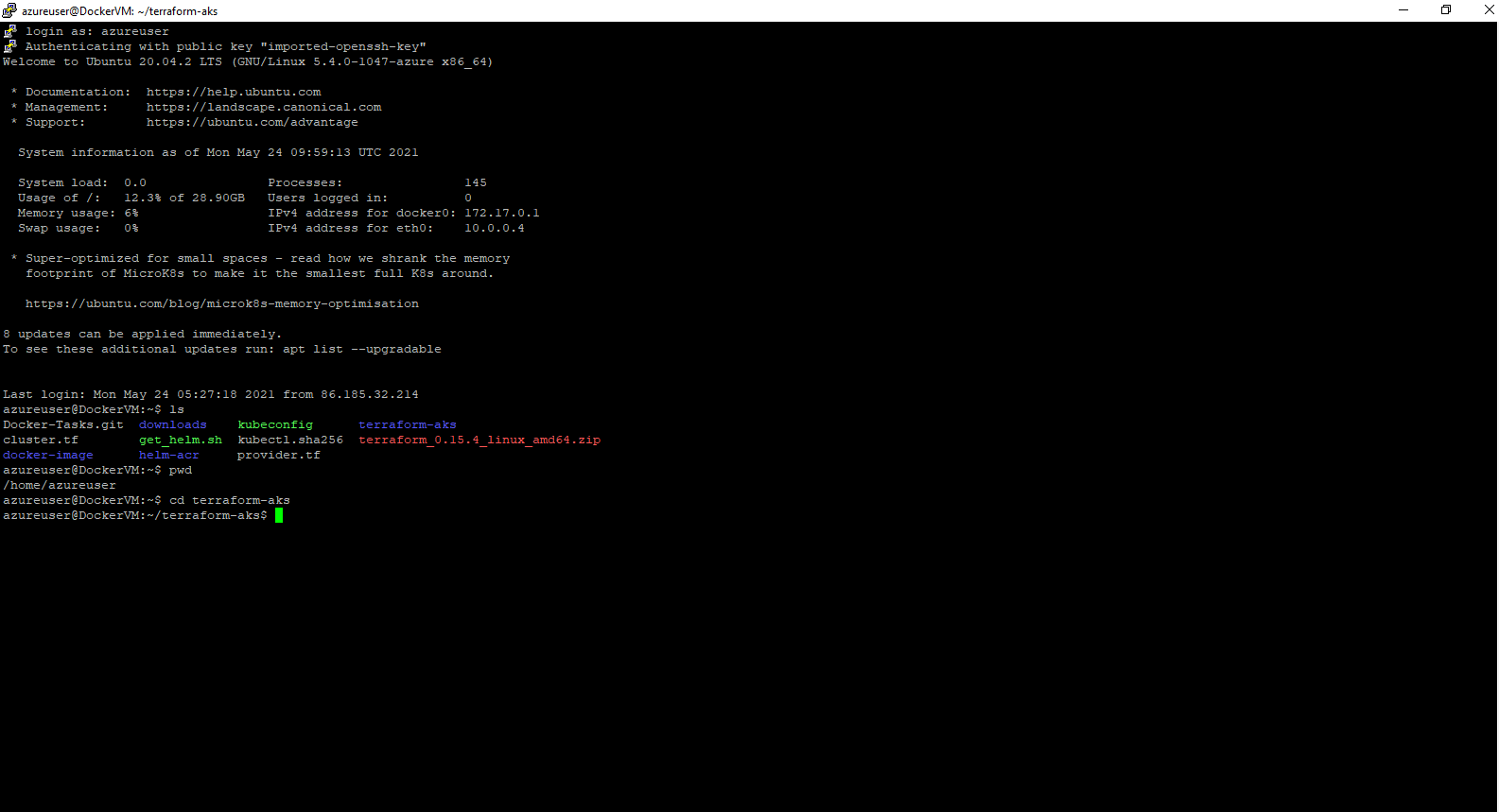
**Step1 :** First we need to create the VM instance and I created vm instance on the name Docker VM.

****

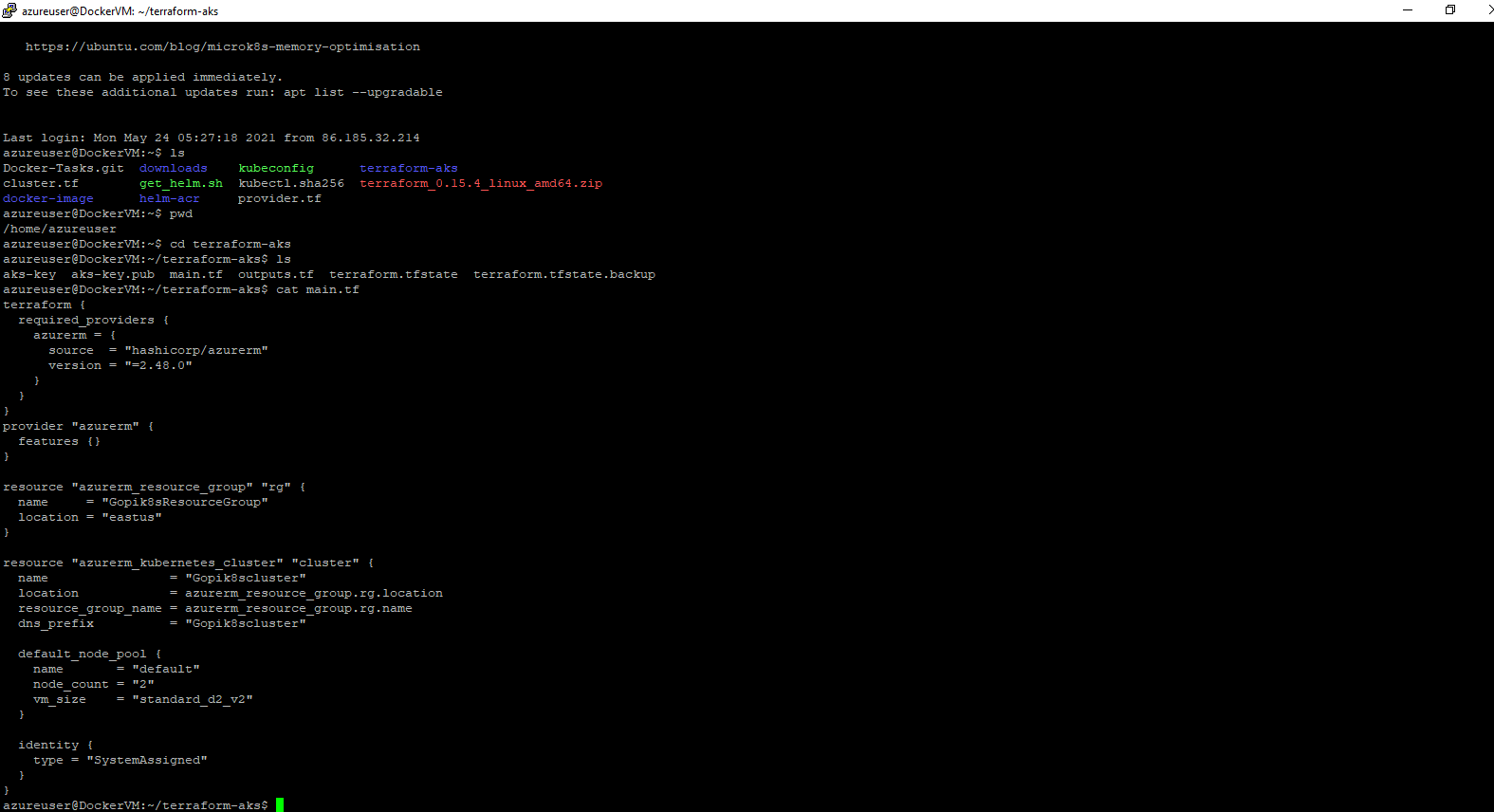
**Step2:** Once I created the instance we need to install the below packages:

1. Azure CLI
2. Kubectl
3. Terraform

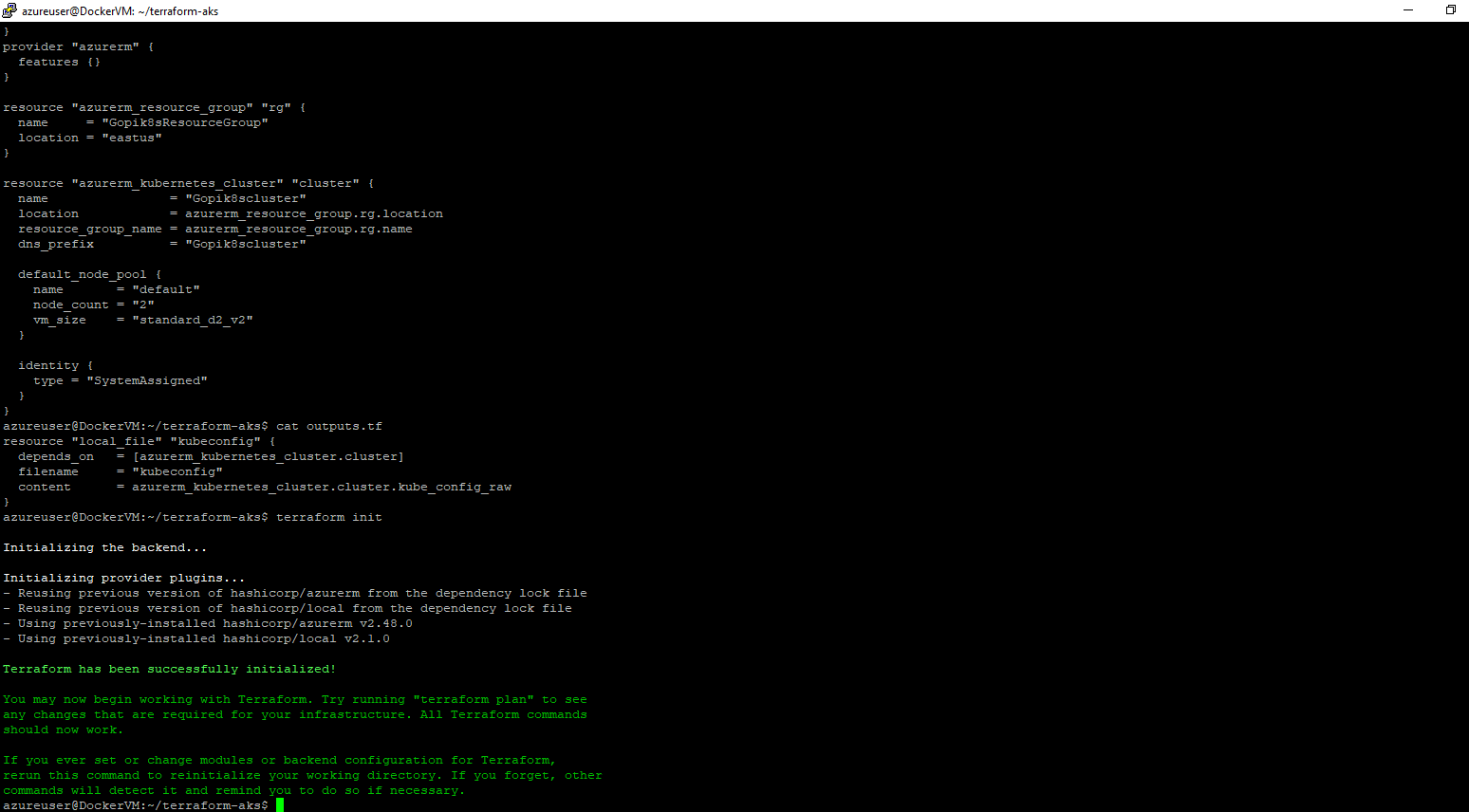
**Step3:** Once we install the three packages as prerequisites. We need to create directory as terraform-aks.

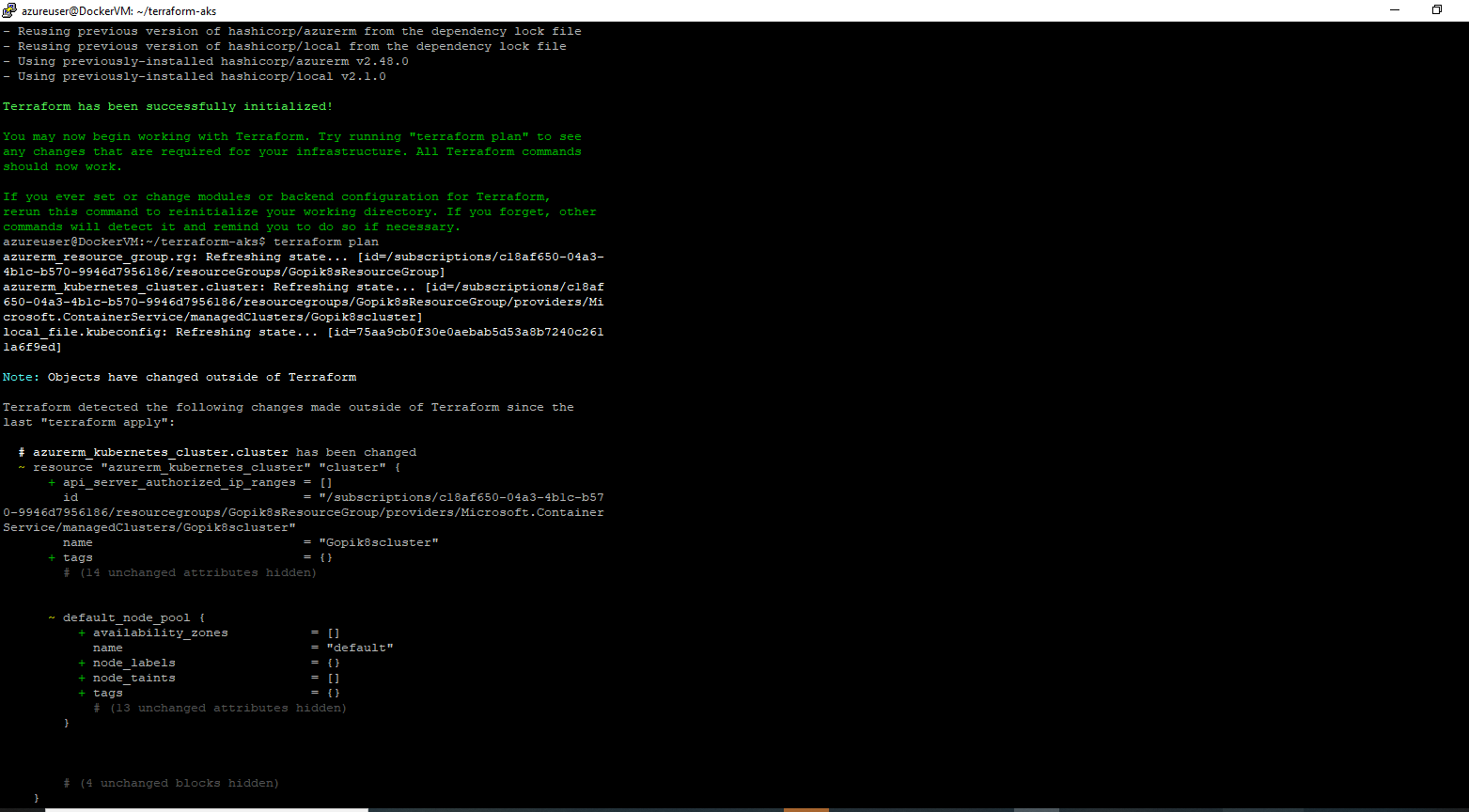


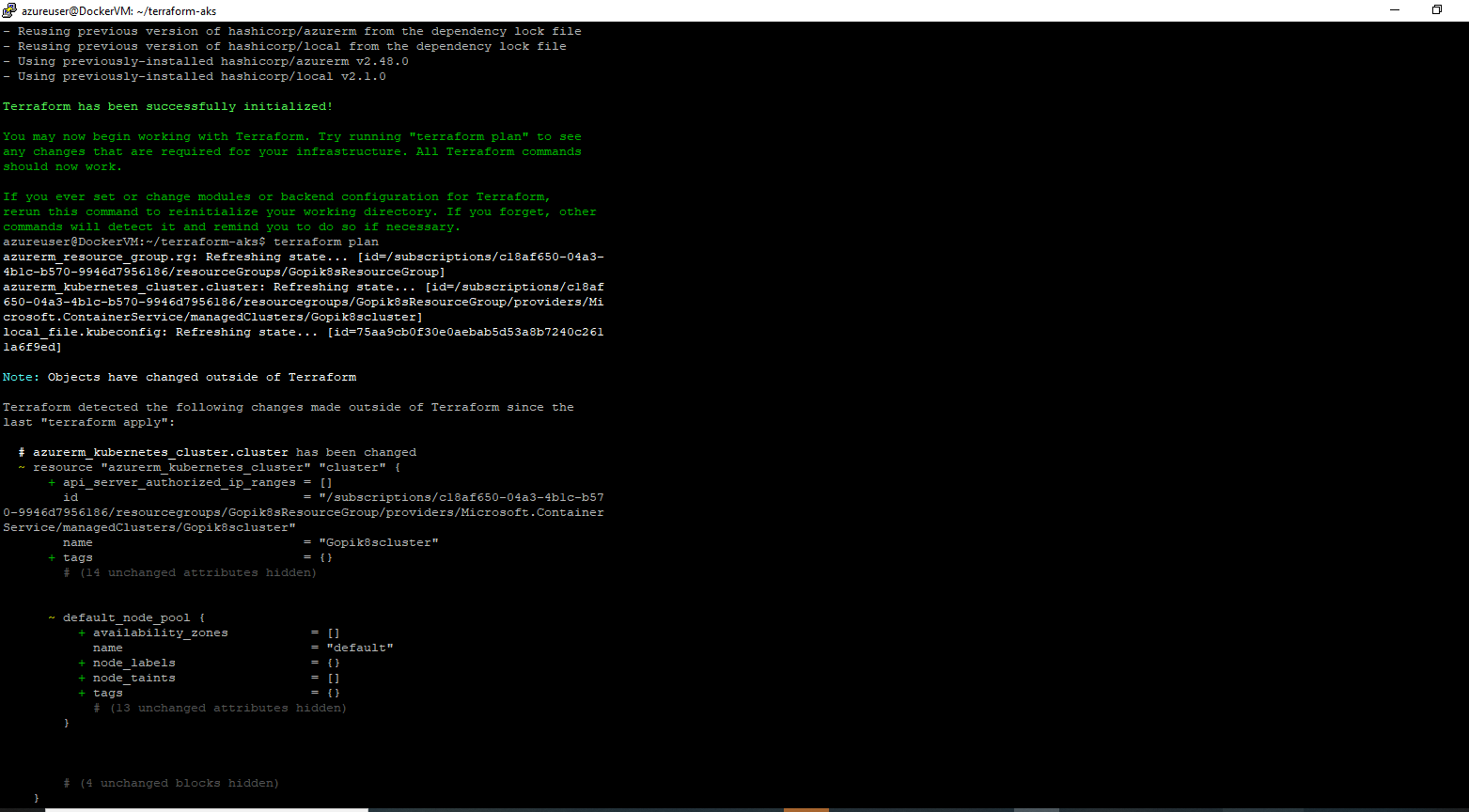
**Step4:** In that directory I created two files on the name of main.tf and outputs.tf. In stored my cluster code on the name of main.tf and in outputs.tf I used code where we need to store the kube.config file as default. We can see that kubeconfig is created.

****

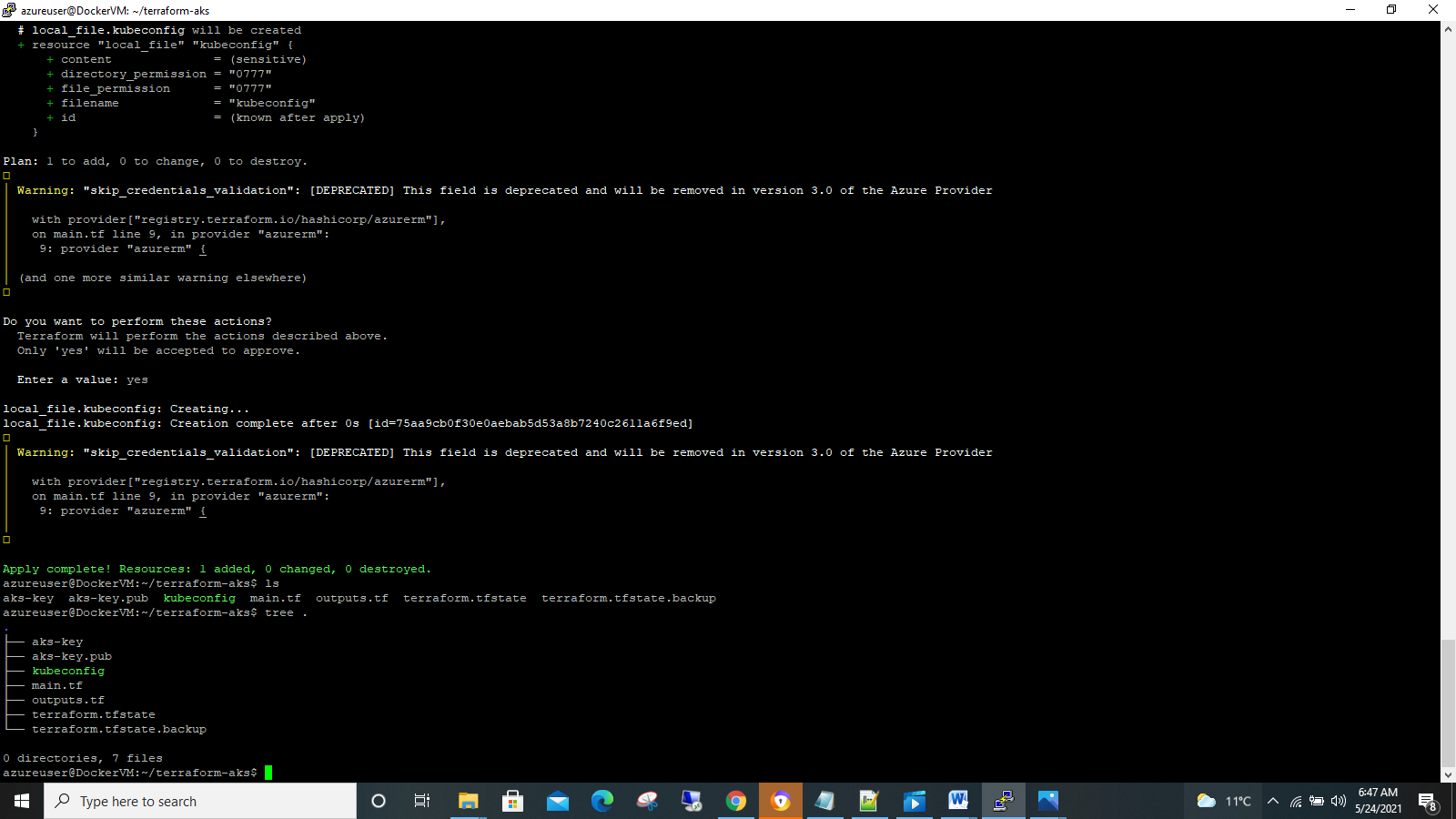
**Step5**: I used terraform init as dry runi am checking whether we have any error. Once we used terraform init we will use terraform plan by using it will check which resources need to add,chang and destroy. To truncate those changes I used the terraform apply.

****

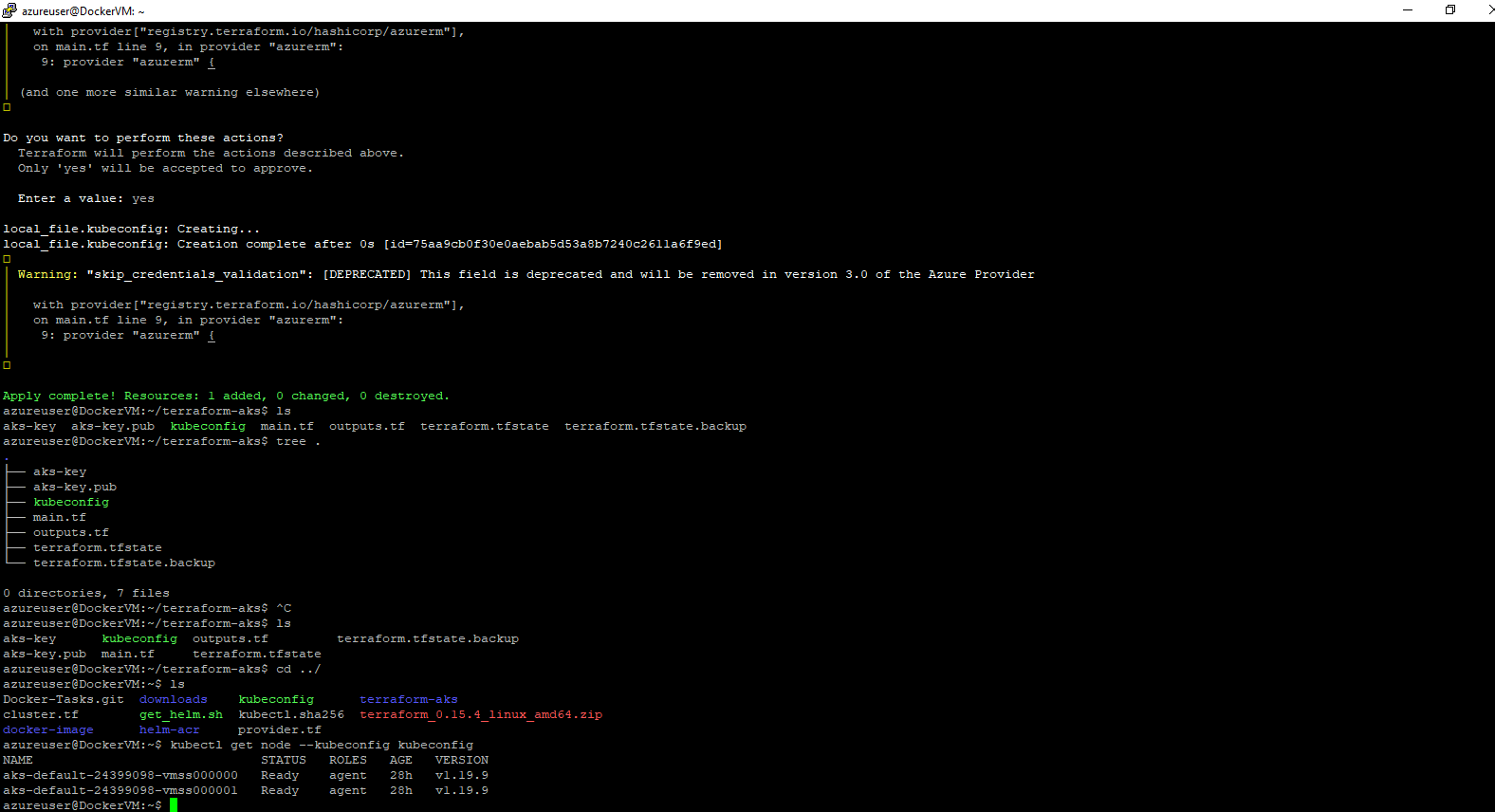
****

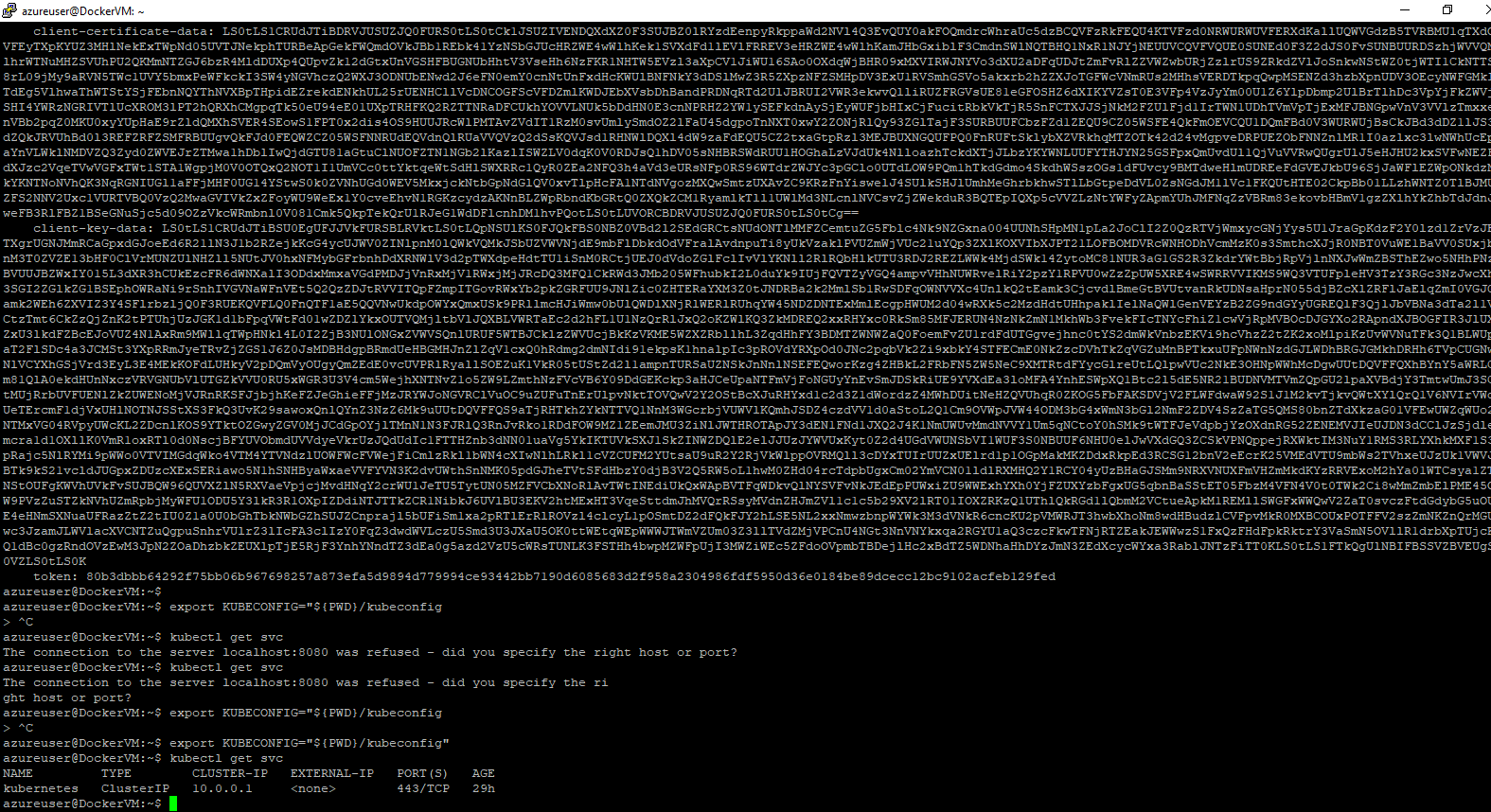
****

**Step6:** After using terraform apply by using tree. We can check which files are created we can see that terraform.tf state file and terraform.tfstate.backup files. This will two files will track which resources we created and also we kubeconfig file also created.

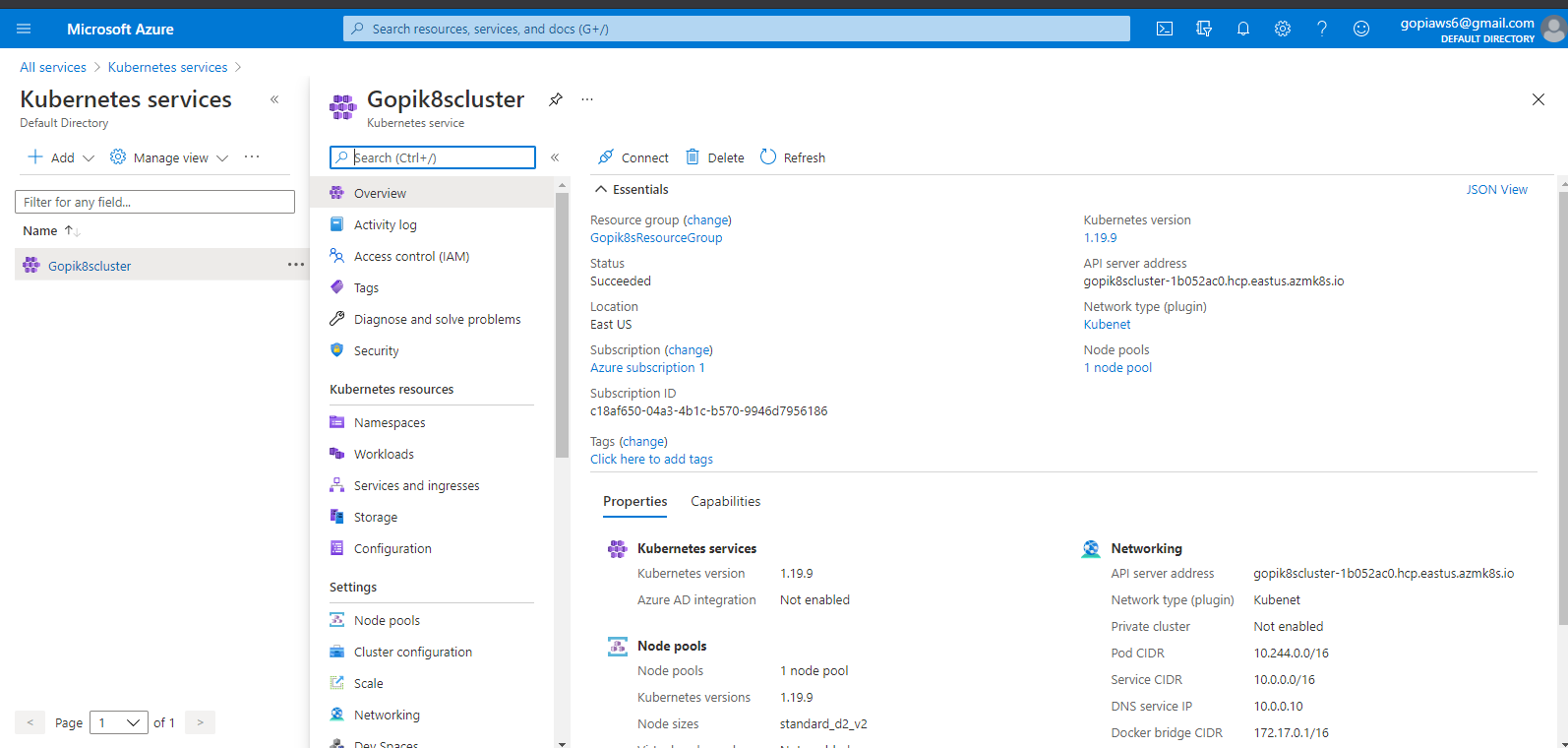


**Step7:** By using kubectl get node --kubeconfig kubeconfig we check how many nodes are created. Hear we can see that two nodes are created. By using kubectl get svc we can check the service.

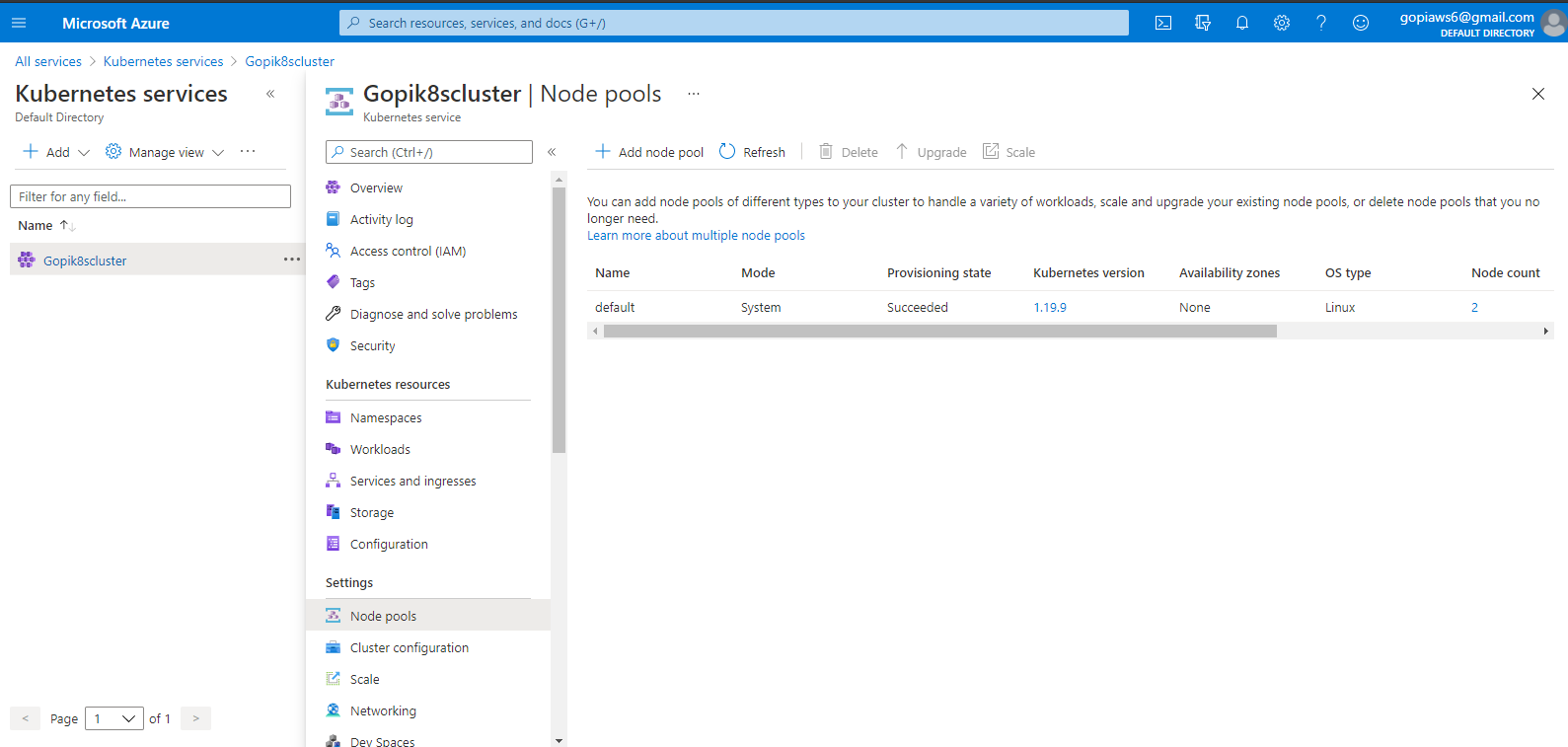
****

****

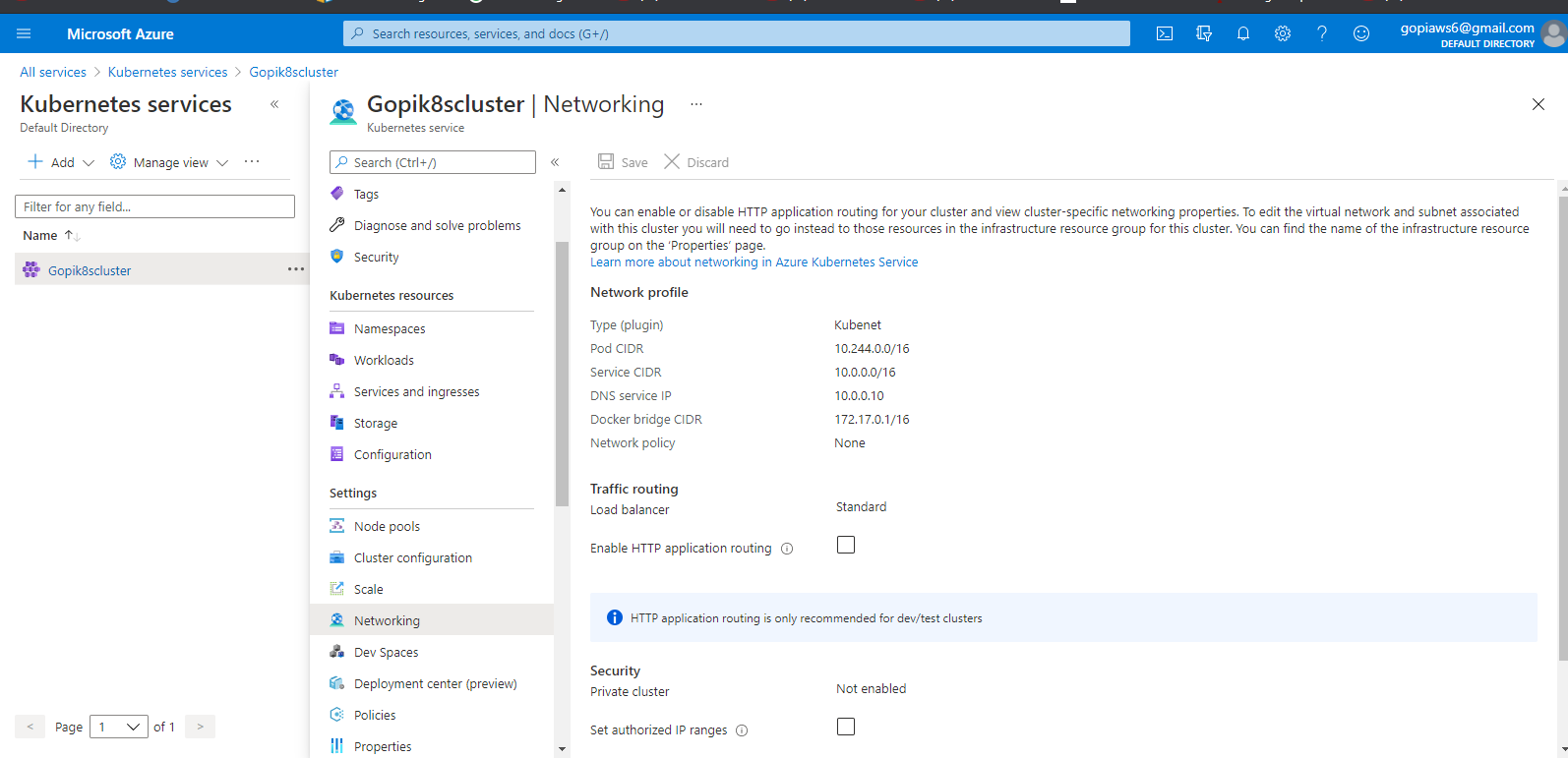
**Step8:** We can see on gui that on the name Gopi8cluster is created and we can check it in kubernates services.

****

**Step9:** We can also see that 2 nodes are created in Gopik8cluster| Node pools.

****

**Step10**: In Gopi8scluster we can check that default network is created and used ip range as pubic if you need to use private ip range there where we can use in Networking.



**Step11:** By this cmds we can install the helm which is mentioned below.

curl -fsSL -o get\_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3

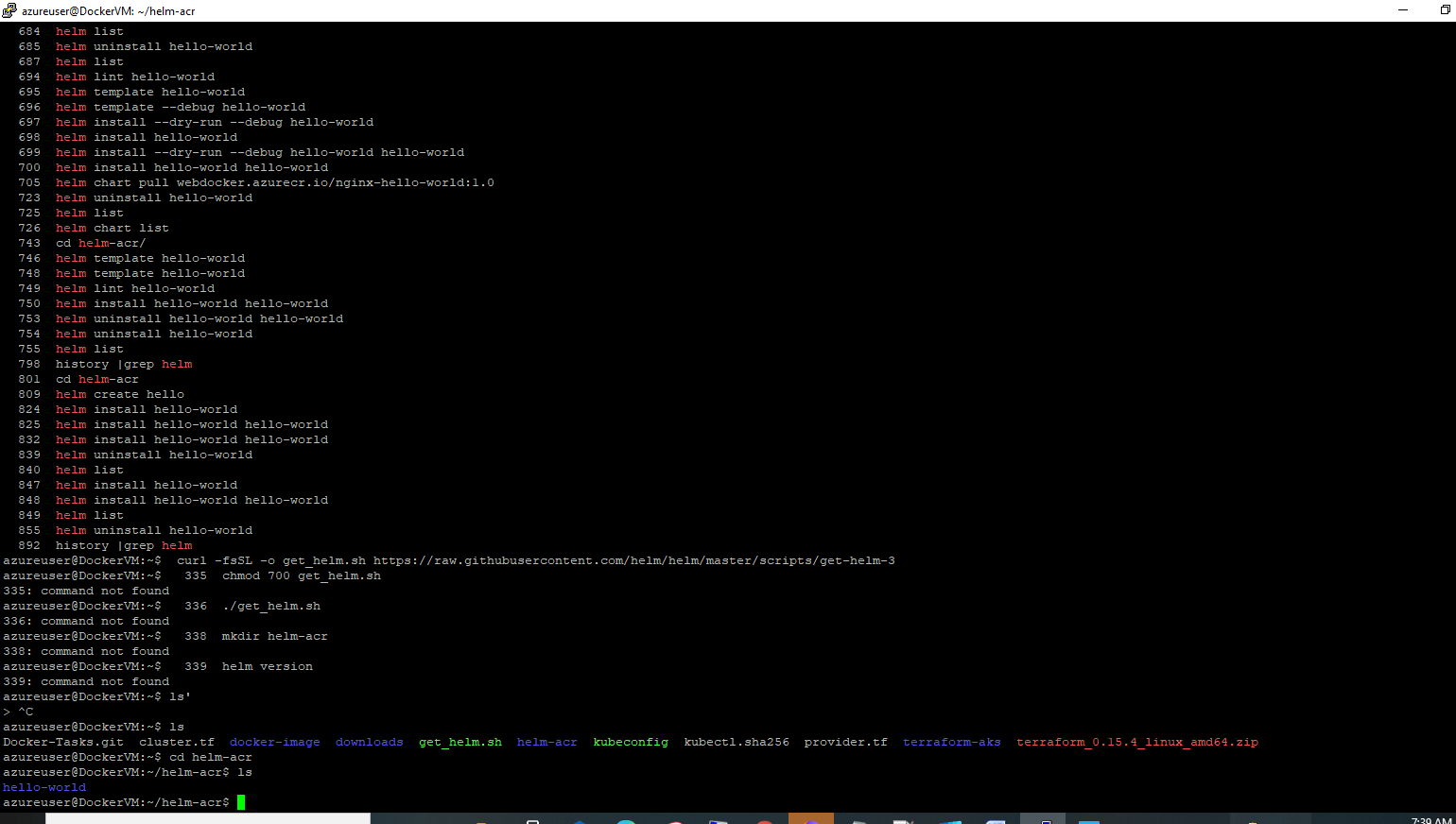
chmod 700 get\_helm.sh

./get\_helm.sh

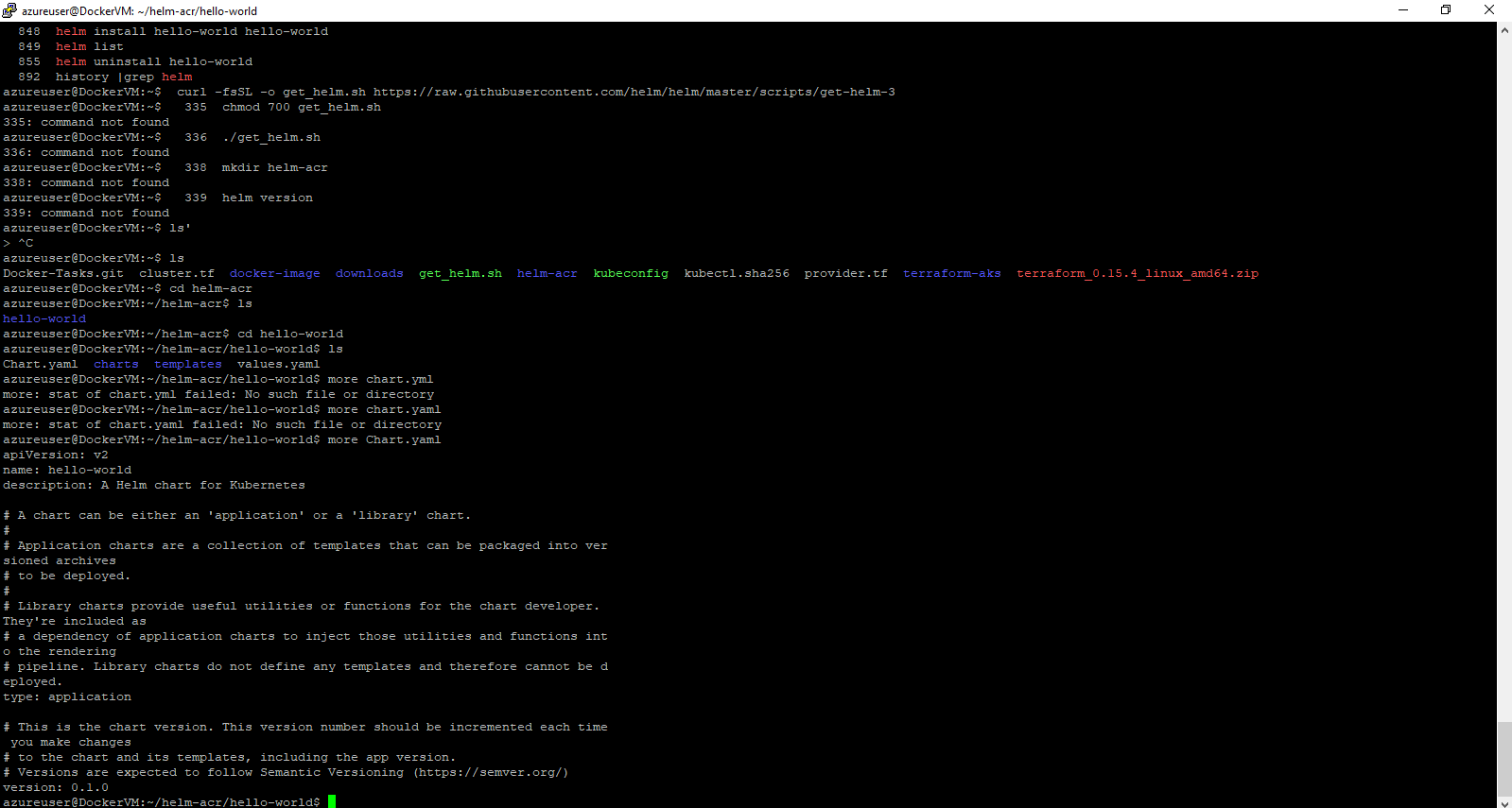
mkdir helm-acr

helm version

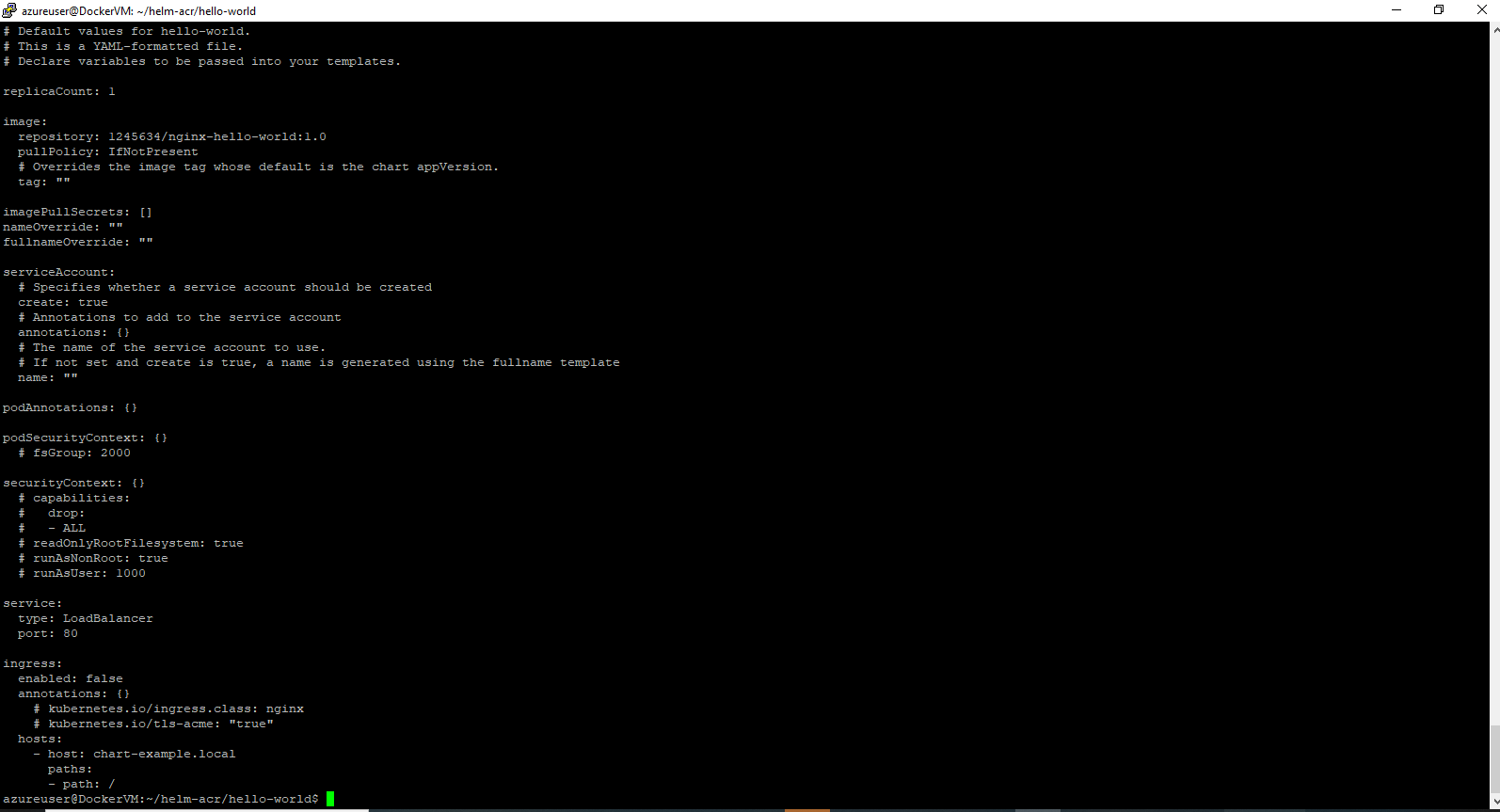
**Step12:** I created helm-acr as directory and by using cmd helm create hello-world I created folder on the name of hello-world directory.

****

**Step13:** Once I applied all changes in the charts where hello-world chart will be created.

****

**Step14:** In values.yaml file I provided my docker hub credentials and image name. We can see the type:LoadBalancer and port:80 are services.



Step15: In template folder I saved deployment.yaml file if we see in to it where I used port as:

- name: http

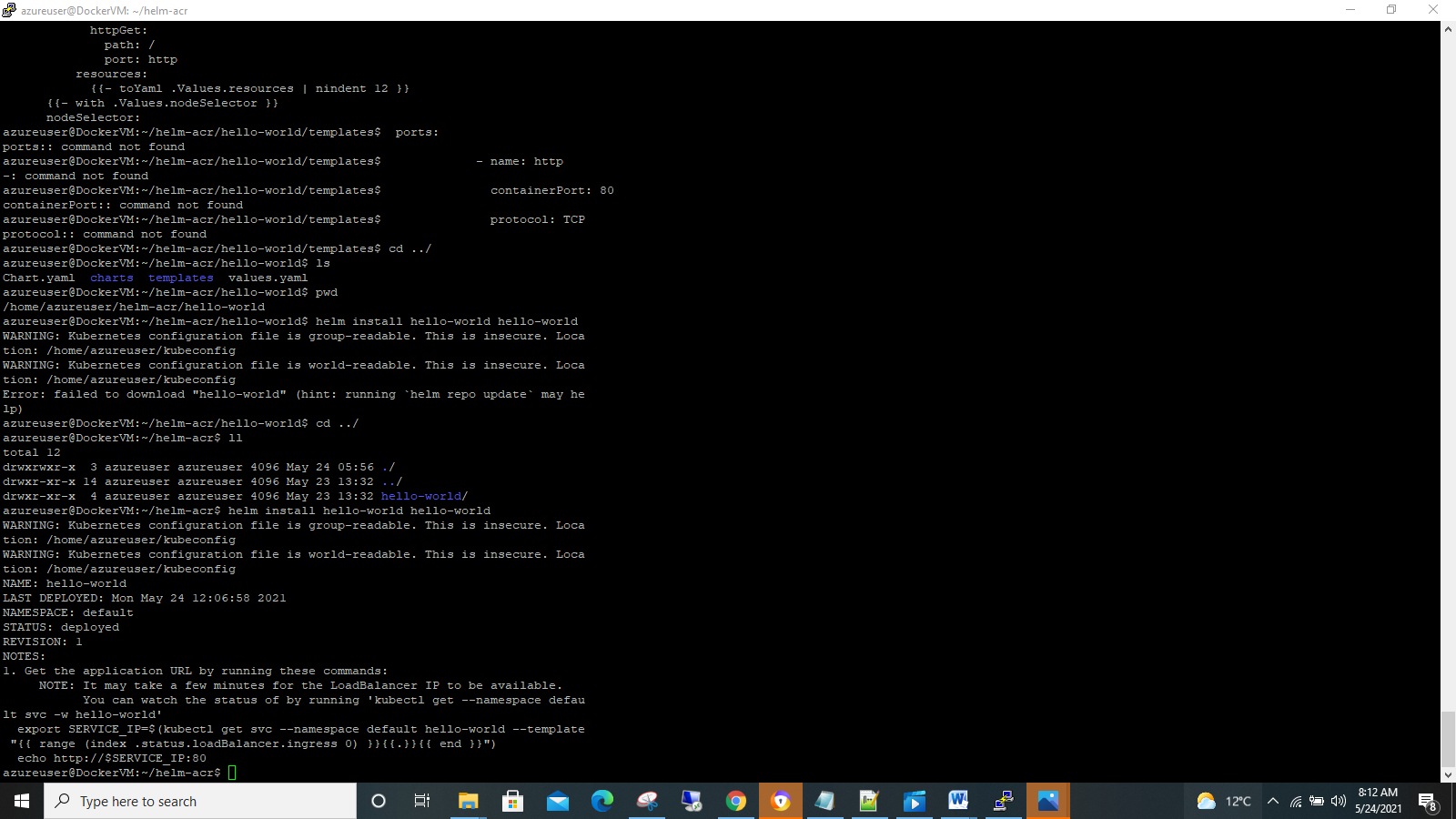
containerPort: 80

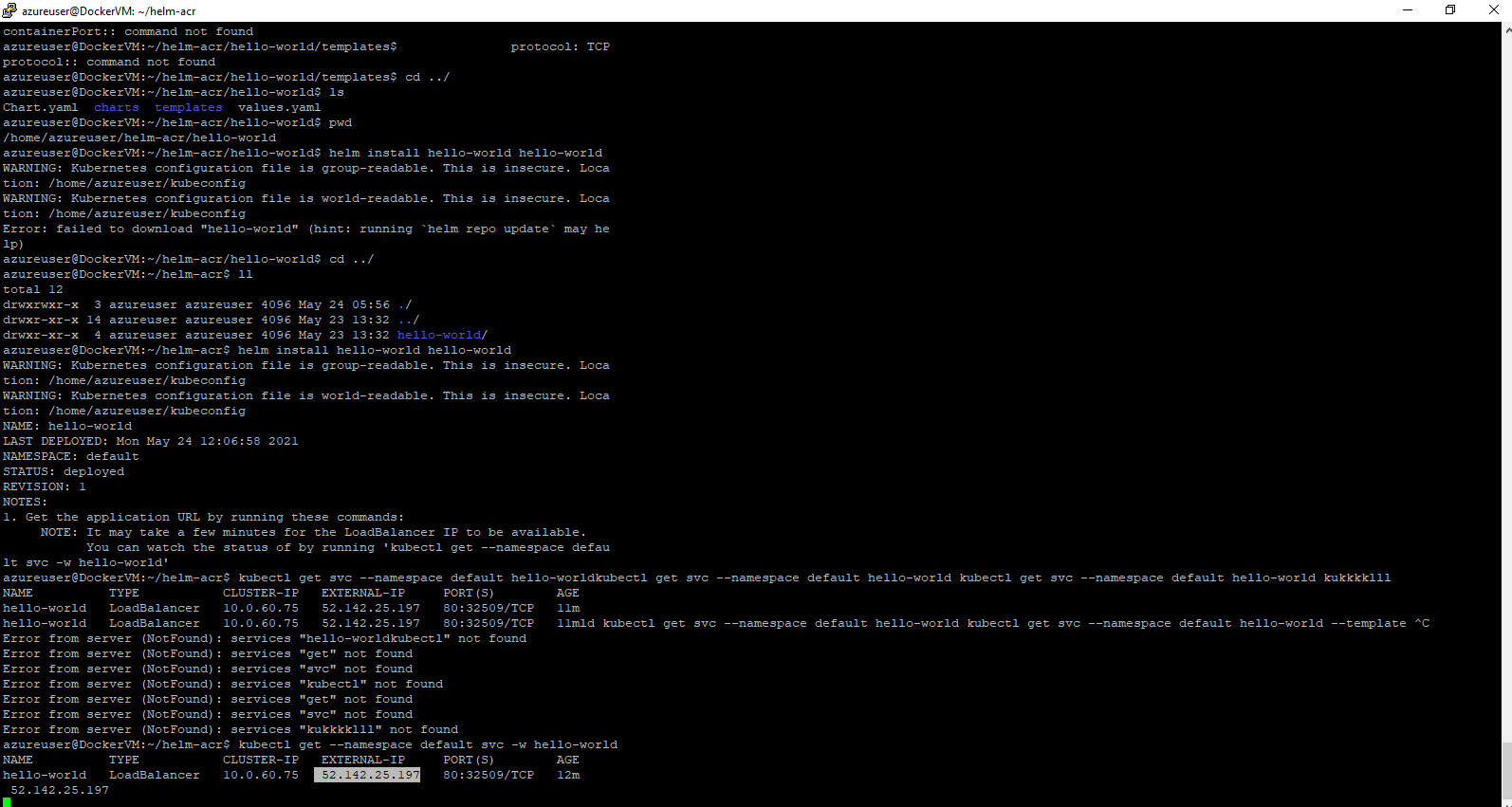
protocol: TCP

If we want we change to any other port as per requirement by using cmd s heml update cmd.



**Step16:** By using cmd helm install hello-world hello-world. We can see that helm is installed.

****

**Step17:** By using cmd kubectl get --namespace default svc –w hello-world we see the External-IP.

**Step18:**By using External-IP we can see that output

