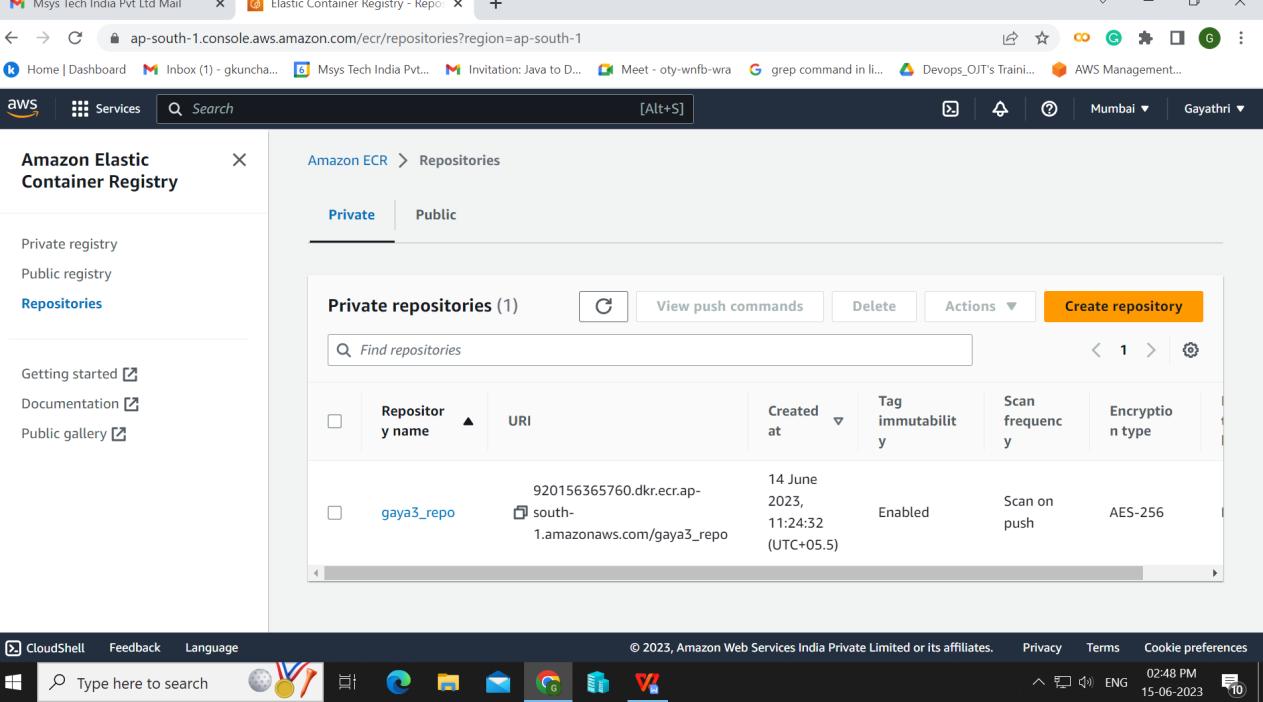
# **USING ANSIBLE TO DEPLOY SAMPLE NGINX/PYTHON APPLICATION INTO KUBERNETES CLUSTER**

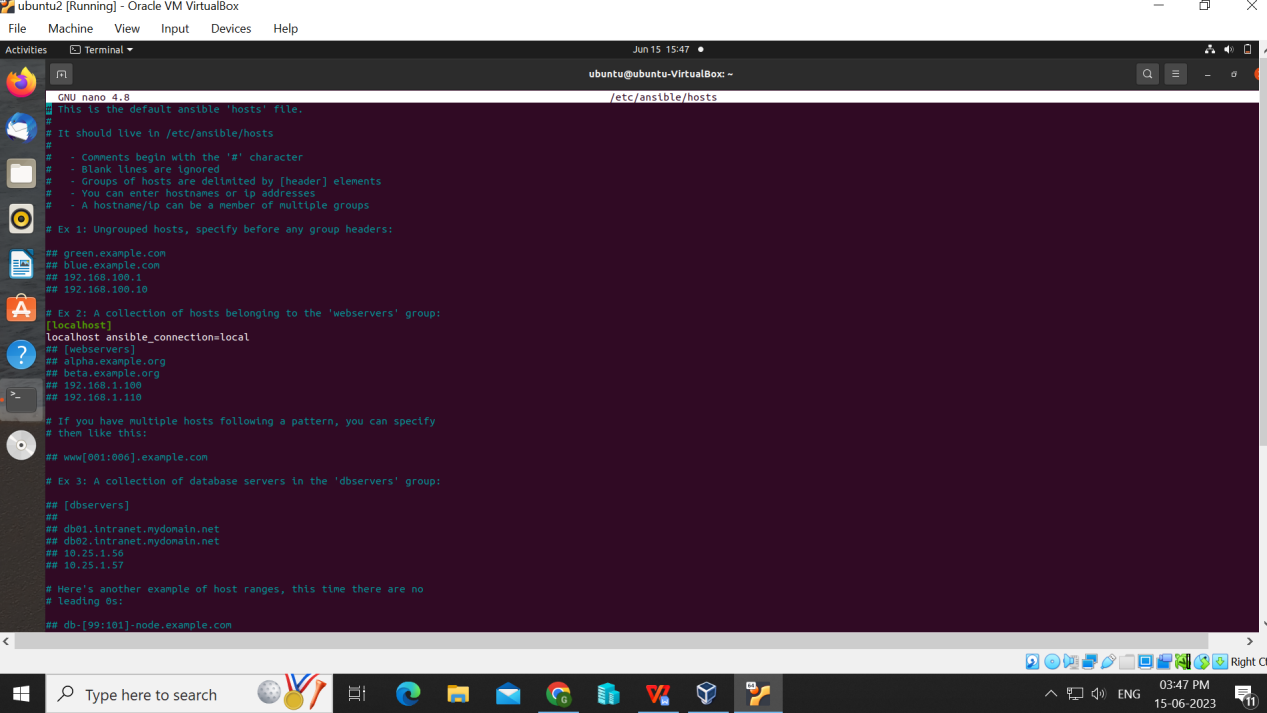
## Build docker image for sample nginx/python and push to AWS elastic container registry(ECR) using docker cli

* Created a docker file in the EC2 instance, and written a nginx latest version to pull the nginx image and set expose to port 80.
* Built a docker file using docker build command to create docker image, also set image name and tag.
* Simultaneously created ECR repository and inside that we select image push command.
* Creating IAM with administrative access permission and also creating access key for AWS configuration.
* Pushed the image to ECR by following the push commands after finishing the AWS configuration.

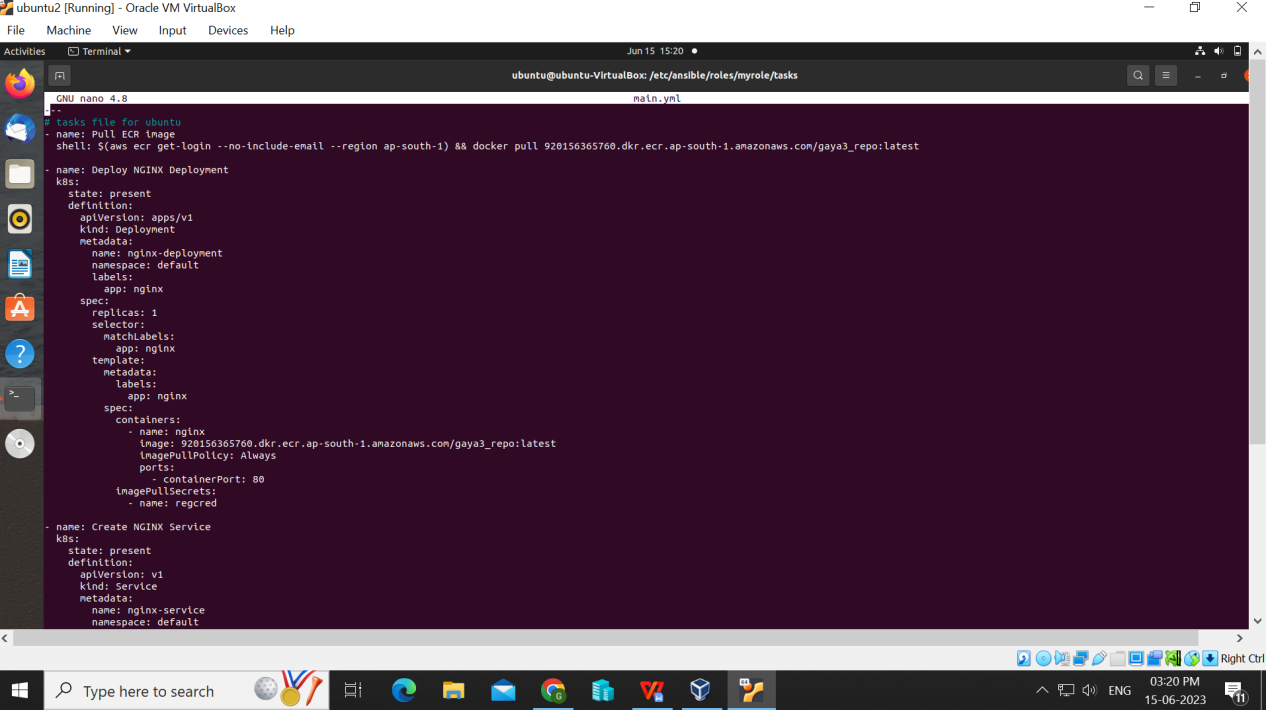


## Create ansible role to deploy nginx application into kubernetes cluster

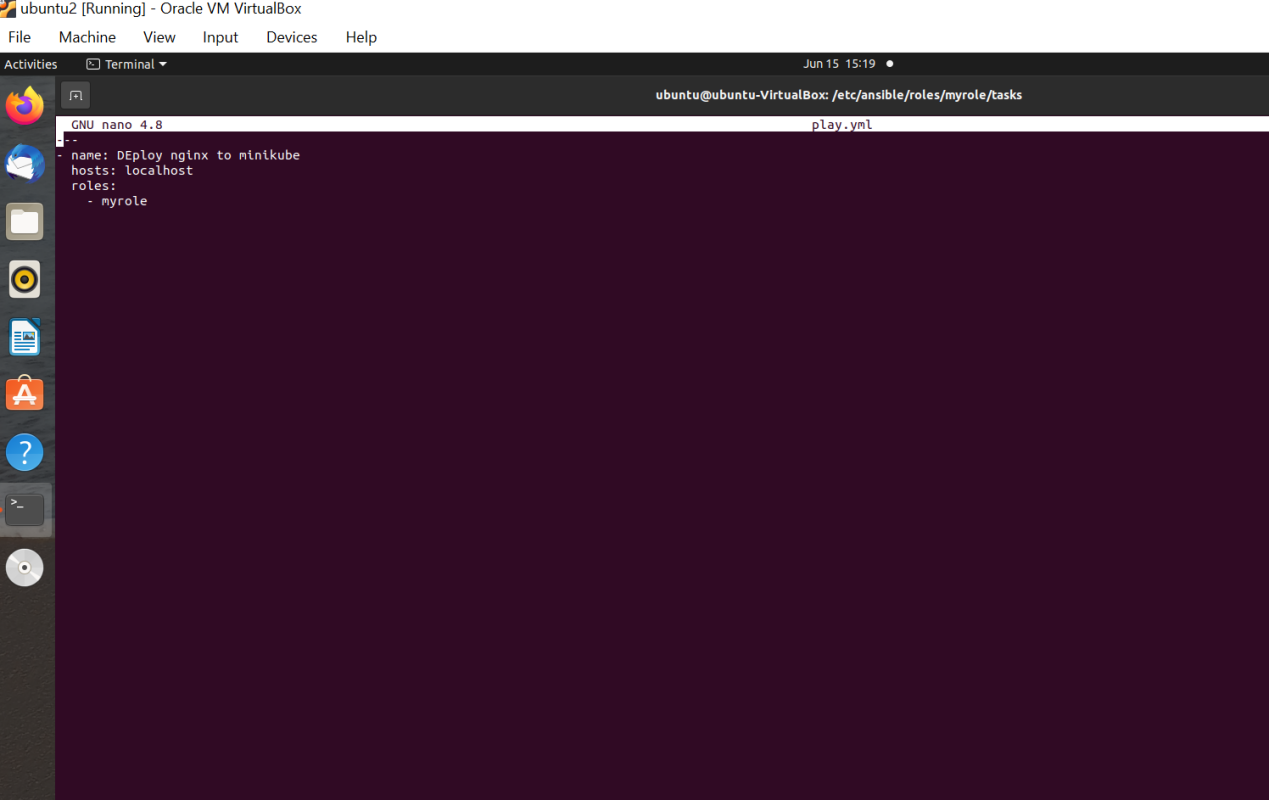
* Created inventory inside the etc/ansible/hosts.
* Created ansible role inside the etc/ansible/roles using “ansible-galaxy init <rolename>”and inside the tasks.
* Created a yml file with three tasks such as pulling image from ECR, deployment, service.
* Inventory File Configure:



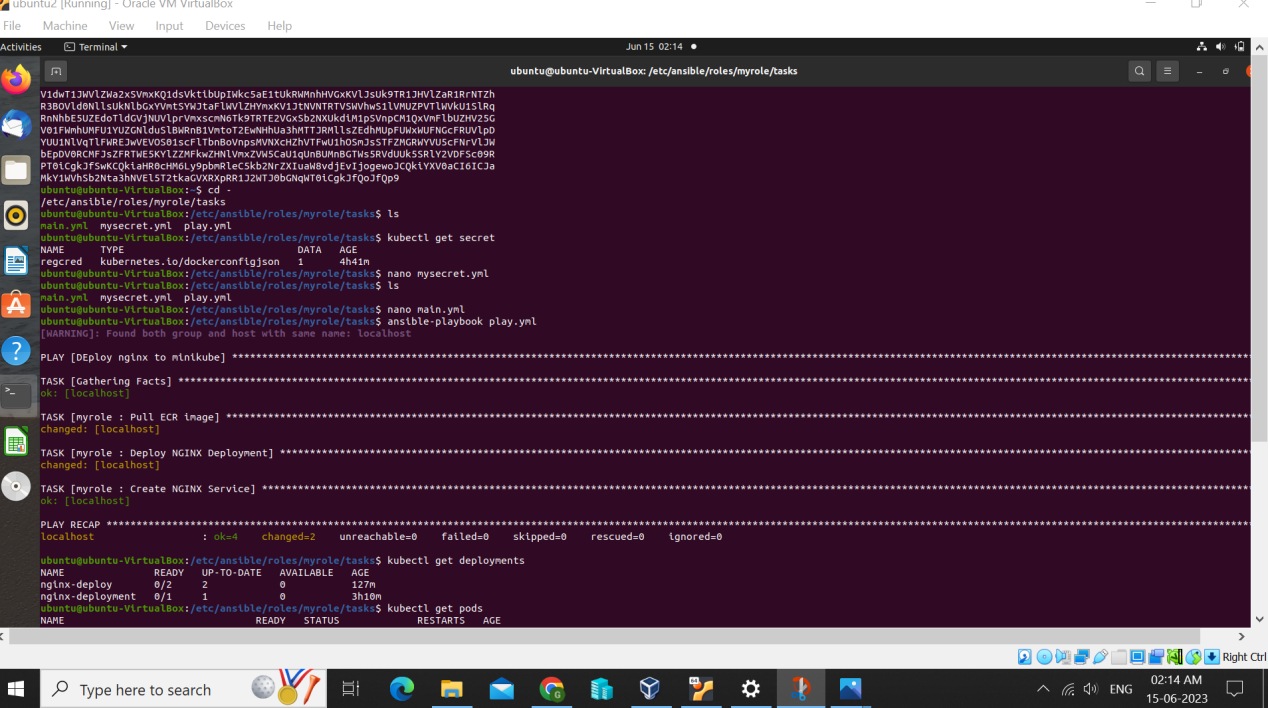
* Ansible Role:



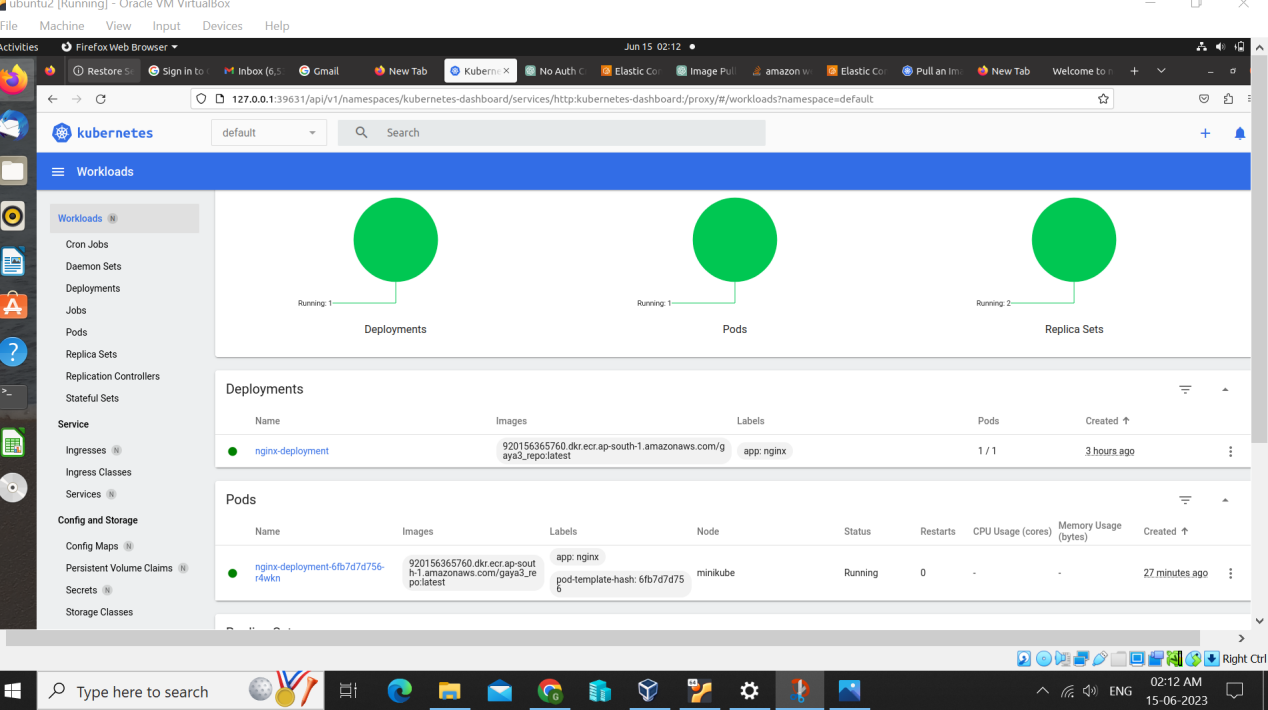
* Ansible Playbook File:



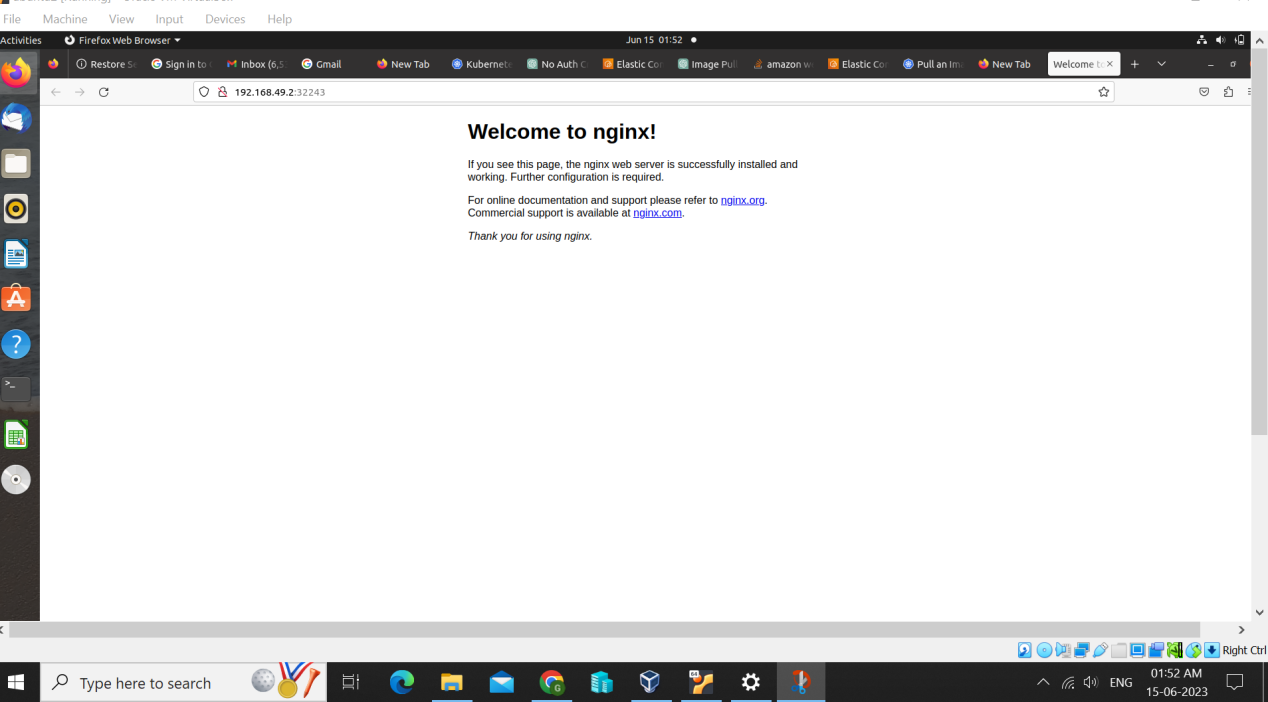
* Ansible Playbook Running:



* Running the minikube and setting based on our application
* Running the playbook using ansible-playbook <playbook name.yml> command.
* Checking the playbook health using kubectl gets pods commands.
* Checking the deployment health and service health using “kubectl get deployments” and “kubectl get svc”
* Pod Status:



## Outcome is application deployed and run in k8s cluster. Able to access nginx website with url



* GitHub Url:

https://github.com/Gaya3-2000/AnsibleK8sDeploy