**TITLE: Demonstrate the continuous integration and delivery by Dockerizing Jenkins Pipeline.**

Problem Statement Scenario:

You are a DevOps consultant in AchiStar Technologies. The company decided to implement DevOps to develop and deliver their products. Since it is an Agile organization, it follows Scrum methodology to develop the projects incrementally. You are working with multiple DevOps Engineers to Dockerize the Jenkins Pipeline. During the sprint planning, you agreed to take the lead on this project and plan on the requirements, system configurations, and track the efficiency. The tasks you are responsible for:

• Availability of the application and its versions in the GitHub.

◦ Track their versions every time a code is committed to the repository.

• Build the application in Docker and host it in Docker Hub.

• Pull the Docker image and run it again.

The company goal is to deliver the product frequently to the production with high-end quality.

**Step by step procedure**

1. Create the project folder and login to the folder.

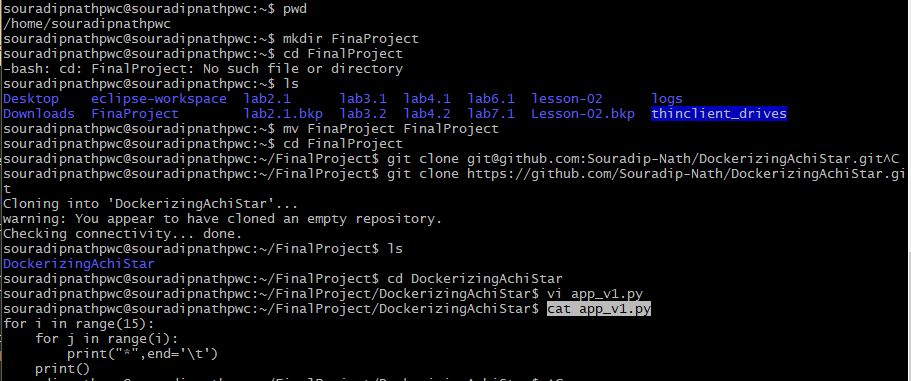
>> mkdir FinaProject

>> cd FinalProject

1. Cloning github repository

>> git clone <https://github.com/Souradip-Nath/DockerizingAchiStar.git>

>> cd DockerizingAchiStar

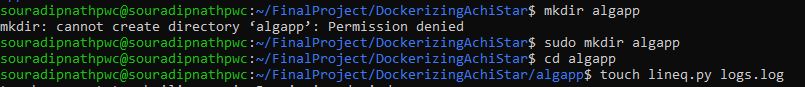


1. Creating application folder “algapp” and creating blank files

>> mkdir algapp

>> cd algapp

>> touch lineq.py logs.log

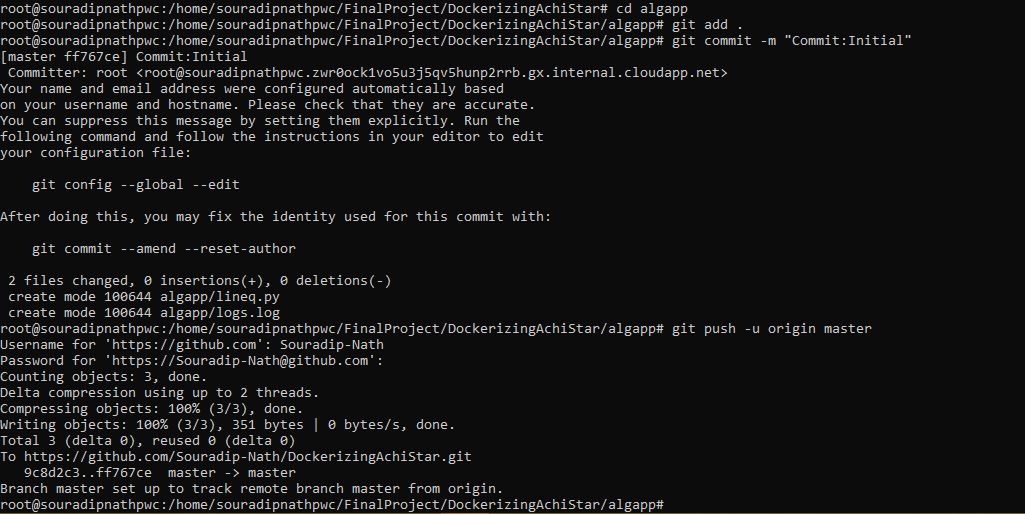


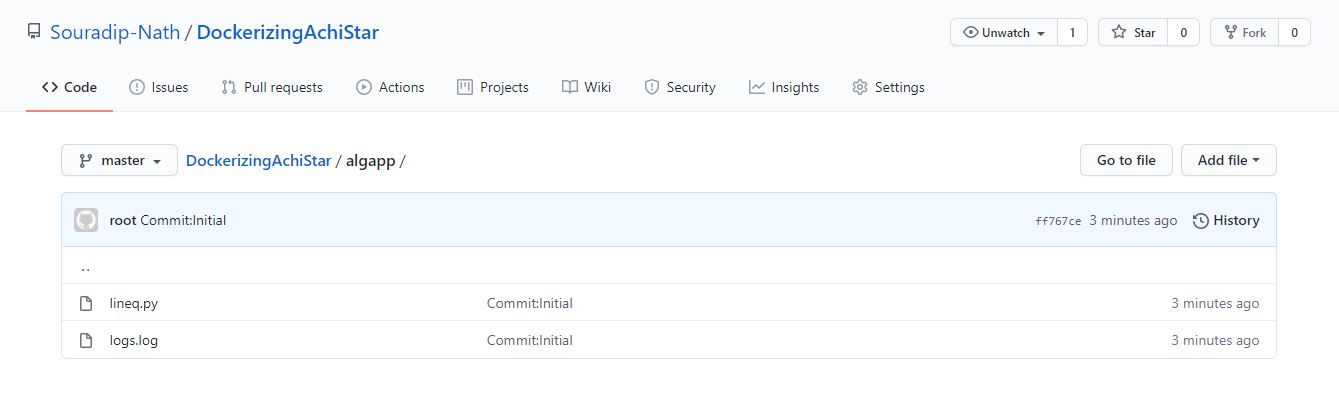
1. Pushing files to github.

>> git add .

>> git commit -m "Commit:Initial"

>> git push -u origin master

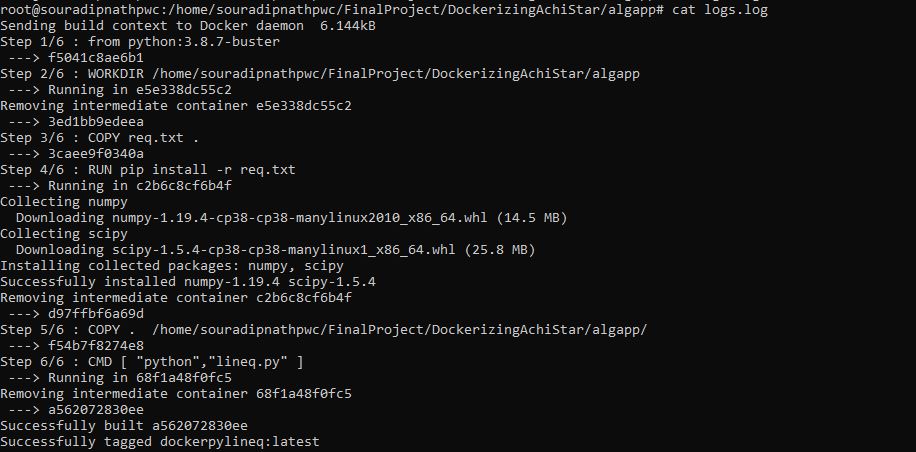




1. Create docker image

>> docker build -t dockerpylineq . > logs.log

>> cat logs.log



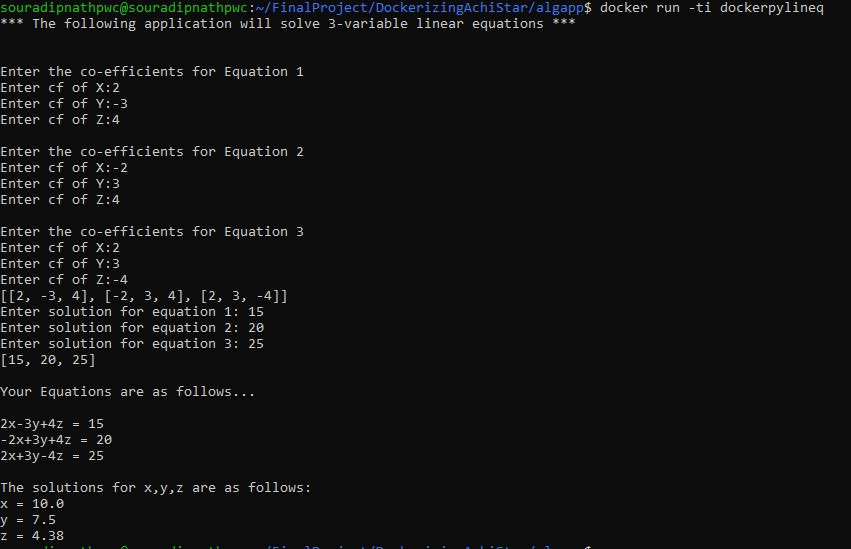
1. Checking for new docker image

>> docker images



1. Running the application from the newly created docker image

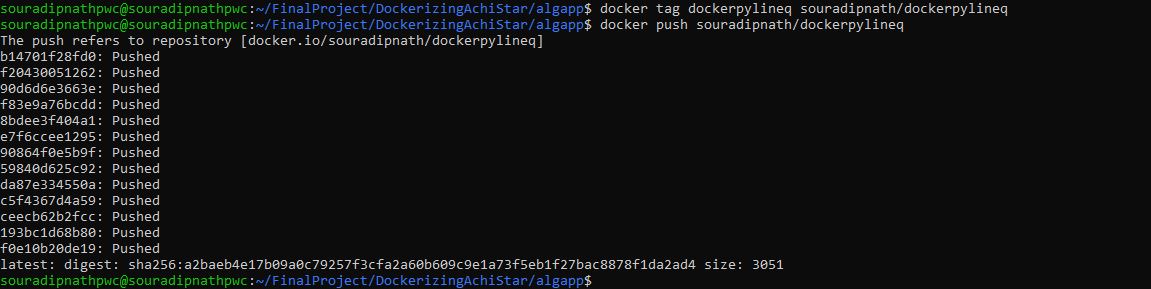
>> docker run -ti dockerpylineq



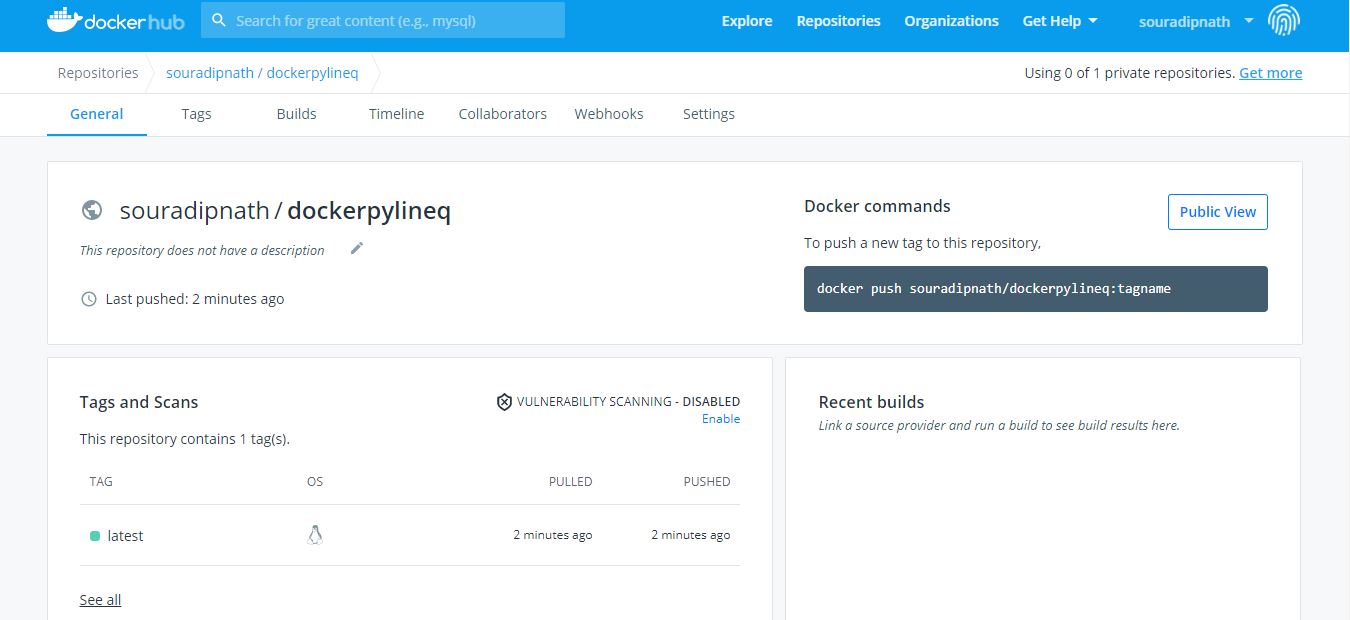
1. Pushing image to dockerhub

>> docker tag dockerpylineq souradipnath/dockerpylineq

>> docker push souradipnath/dockerpylineq

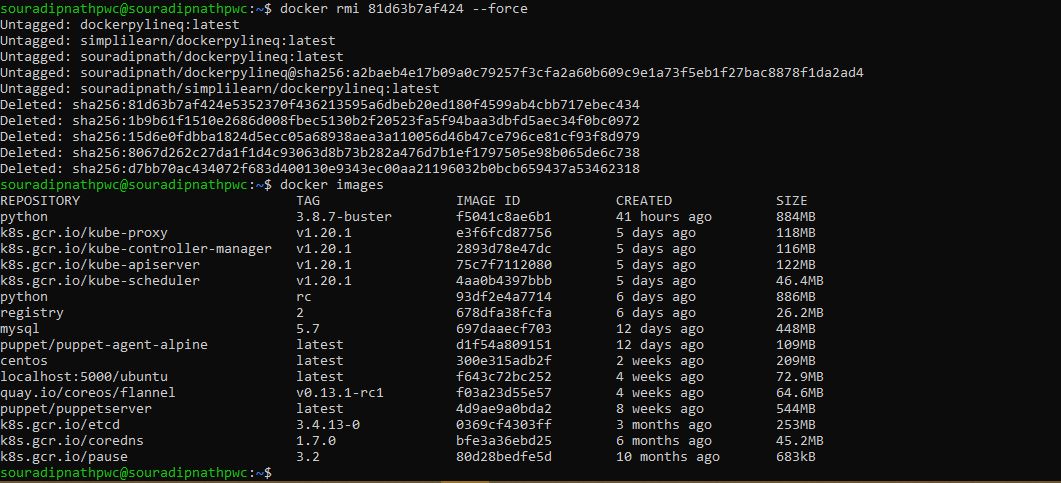


1. Checking the public image in <https://hub.docker.com/>



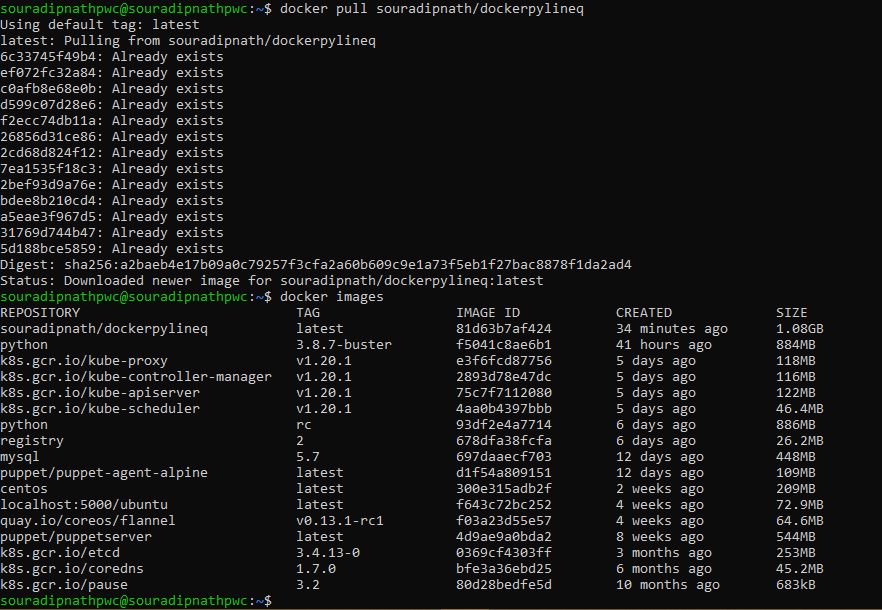
1. Removing the docker image from local repository

>> docker rmi <image id> --force



1. Pulling the image from dockerhub

>> docker pull souradipnath/dockerpylineq



1. Running the application from the new image

>> docker run -ti souradipnath/dockerpylineq

