Lab (1)

01) What is Jenkins used for?

Jenkins is an open-source automation server written in Java. It is used to implement continuous integration and continuous delivery (CI/CD) workflows. Jenkins can be used to automate a wide variety of tasks, including:

- Building and testing software
- Deploying software to production
- Managing releases
- Monitoring software
- Collecting metrics

Jenkins is a highly extensible tool, and there are hundreds of plugins available that can be used to extend its functionality.

Here are some of the benefits of using Jenkins:

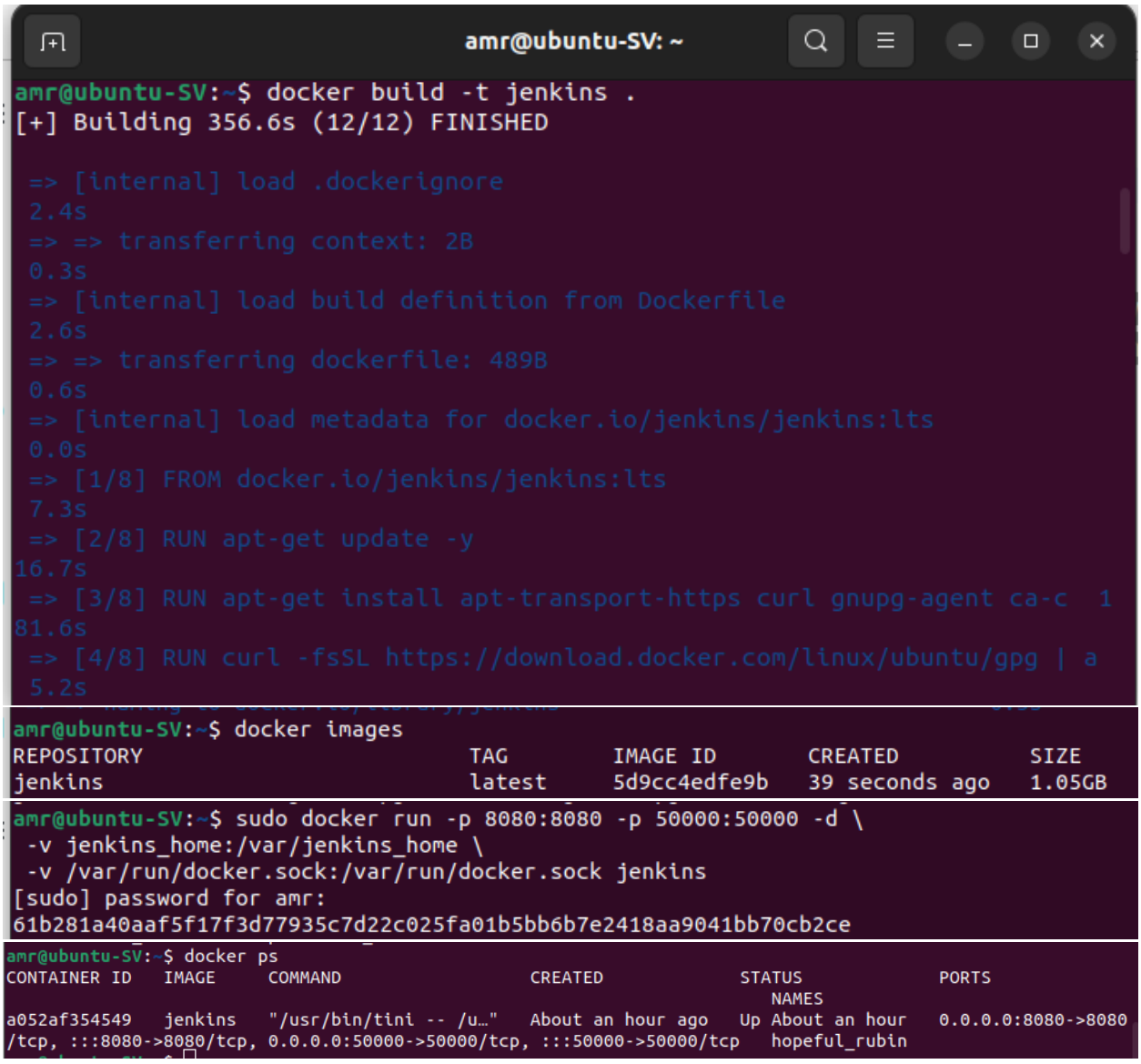
Increased productivity: Jenkins can help to automate many of the manual tasks involved in software development, which can free up developers to focus on more creative and strategic work.

Improved quality: Jenkins can help to ensure that software is built and tested consistently, which can help to improve the quality of the software.

Reduced risk: Jenkins can help to reduce the risk of introducing errors into software, as changes are automatically tested before they are deployed to production.

Improved visibility: Jenkins provides a central dashboard that can be used to track the progress of software builds and deployments. This can help to improve visibility into the software development process and identify potential problems early on.

2) Install jenkins with docker image.



3) What are plugins in Jenkins?

Jenkins plugins are software extensions that add new features or functionality to Jenkins. There are hundreds of plugins available, covering a wide range of use cases. Some of the most popular plugins include:

Build plugins: These plugins add support for building software using different build tools, such as Maven, Gradle, and Ant.

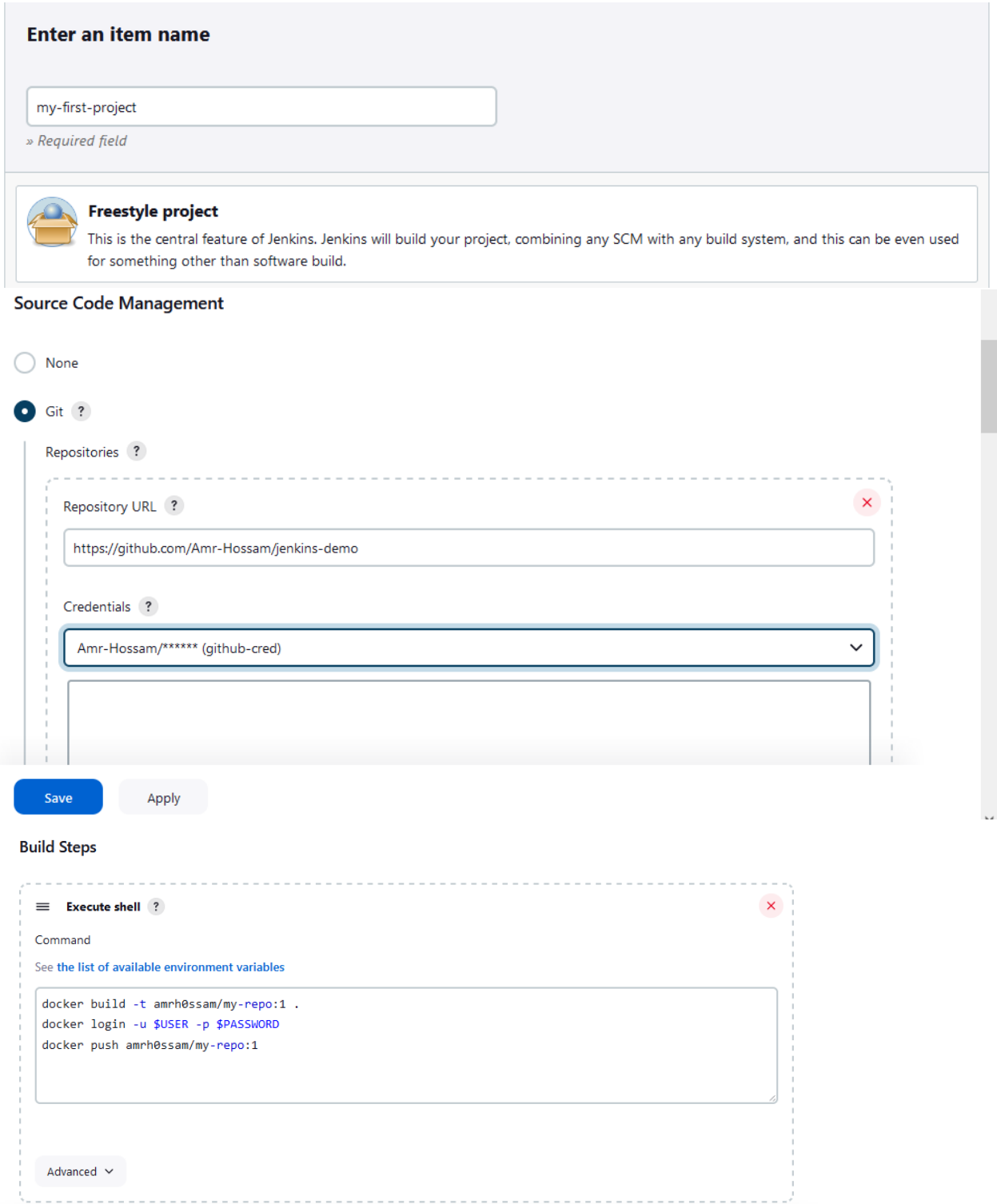
Testing plugins: These plugins add support for testing software using different testing frameworks, such as JUnit, TestNG, and Selenium.

Deployment plugins: These plugins add support for deploying software to different environments, such as production, staging, and development.

Monitoring plugins: These plugins add support for monitoring software builds and deployments, and collecting metrics.

Security plugins: These plugins add support for securing Jenkins, such as by requiring two-factor authentication.

4) Create free style project and link it to private git repo containing any dockerfile then build an image from this dockerfile and push it to private docker repo



Console Output:

```
Login Succeeded
+ docker push amrh0ssam/my-repo:1
The push refers to repository [docker.io/amrh0ssam/my-repo]
4c26417549a4: Preparing
6b5aaff44254: Preparing
53a0b163e995: Preparing
b626401ef603: Preparing
9b55156abf26: Preparing
293d5db30c9f: Preparing
03127cdb479b: Preparing
9c742cd6c7a5: Preparing
293d5db30c9f: Waiting
03127cdb479b: Waiting
9c742cd6c7a5: Waiting
b626401ef603: Mounted from library/openjdk
6b5aaff44254: Mounted from library/openjdk
9b55156abf26: Mounted from library/openjdk
53a0b163e995: Mounted from library/openjdk
4c26417549a4: Pushed
293d5db30c9f: Mounted from library/openjdk
9c742cd6c7a5: Mounted from library/openjdk
03127cdb479b: Mounted from library/openjdk
1: digest: sha256:fc08724293232279438b800eda929664e7d81a2cd81b53bf5828320c70ddeaa4 size: 2003
Finished: SUCCESS
```

Private Repo on Dockerhub: