

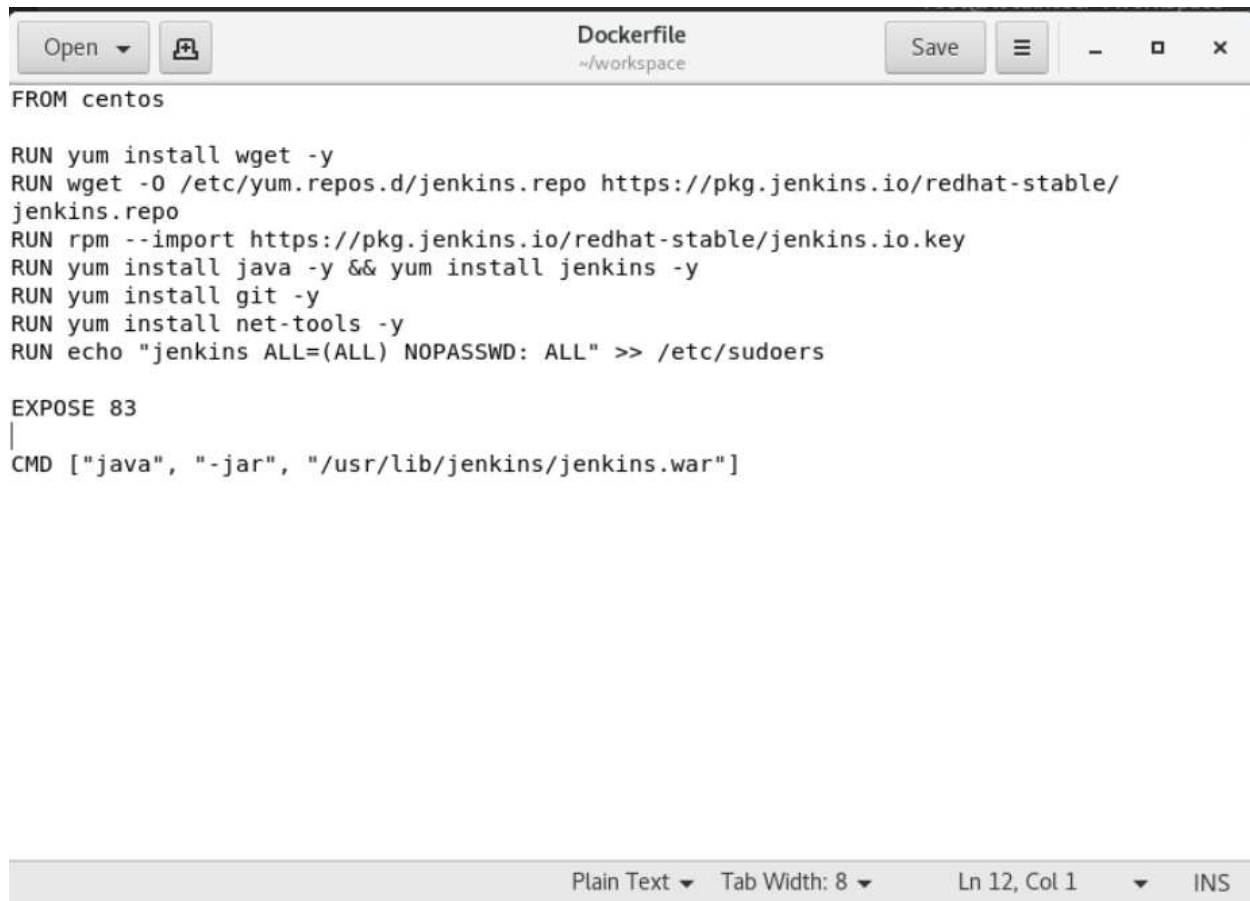


## Automation of Git-Jenkins-Docker

### Problem Statement :

1. Create a container image that's has Jenkins installed using Dockerfile.
2. When we launch this image, it should automatically start the Jenkins service in the container.
3. Create a job chain of job1, job2, job3, and job4 using the build pipeline plugin in Jenkins.
4. Job1: Pull the GitHub repo automatically when some developers push the repo to GitHub.
5. Job2: By looking at the code or program file, Jenkins should automatically start the respective language interpreter install image container to deploy code ( eg. If code is of PHP, then Jenkins should start the container that has PHP already installed ).
6. Job3: Test your app if it is working or not.
7. Job4: If the container where the app is running. fails due to any reason then this job should automatically start the container again.

First we can create the image as follows:-



The screenshot shows a text editor window titled "Dockerfile" with the path "~/workspace". The editor contains the following Dockerfile content:

```
FROM centos

RUN yum install wget -y
RUN wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/
jenkins.repo
RUN rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
RUN yum install java -y && yum install jenkins -y
RUN yum install git -y
RUN yum install net-tools -y
RUN echo "jenkins ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

EXPOSE 83
|
CMD ["java", "-jar", "/usr/lib/jenkins/jenkins.war"]
```

The editor's status bar at the bottom indicates "Plain Text", "Tab Width: 8", "Ln 12, Col 1", and "INS".

FROM: Used for the image to be used for container

RUN: Use this RUN commands to be executed while building the modified container

CMD: the CMD used here will keep the Jenkins live till the container is on and will start on container boot.

After this Build your own image:

Run the given command in the docker

```
docker build -t jenkins:latest
```

All the steps are done of building the images , after this, we have to use this image to make a container,

After is use you IP and 8765 to launch Jenkins:-

Getting Started


# Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/root/.jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

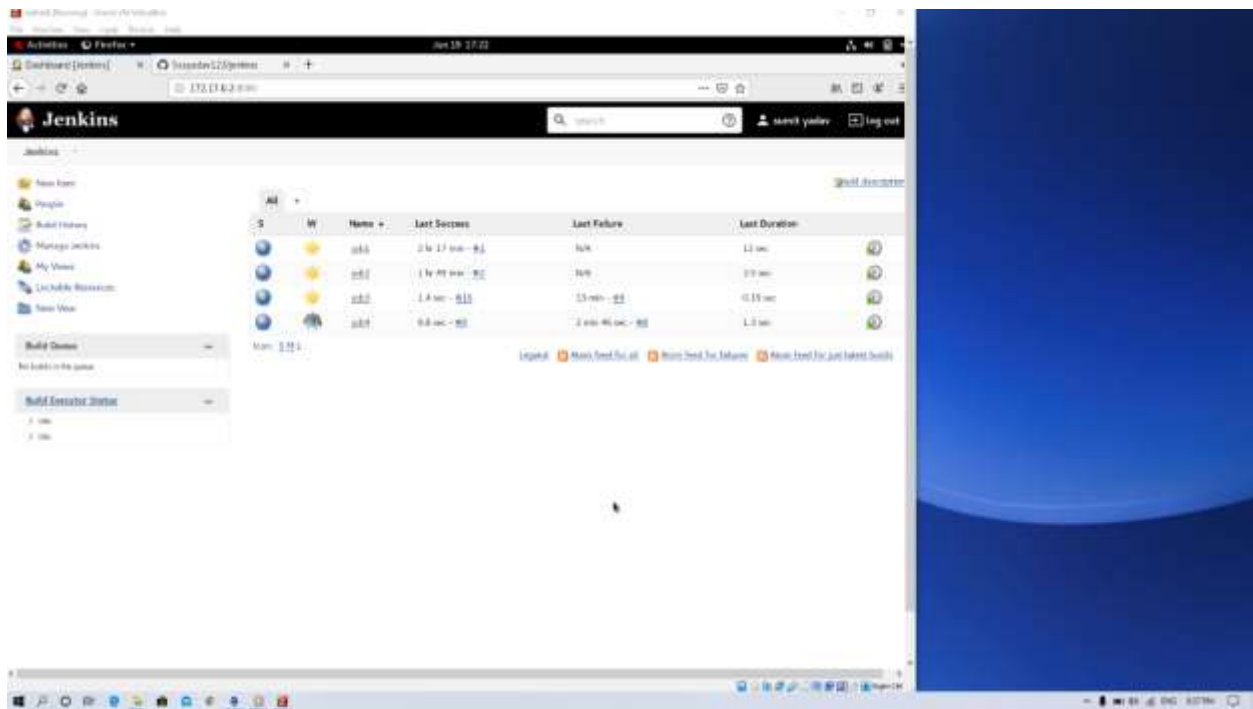


Continue

Paste the password in this and your Jenkins is ready to use:-

Now we will be moving on to our jobs.

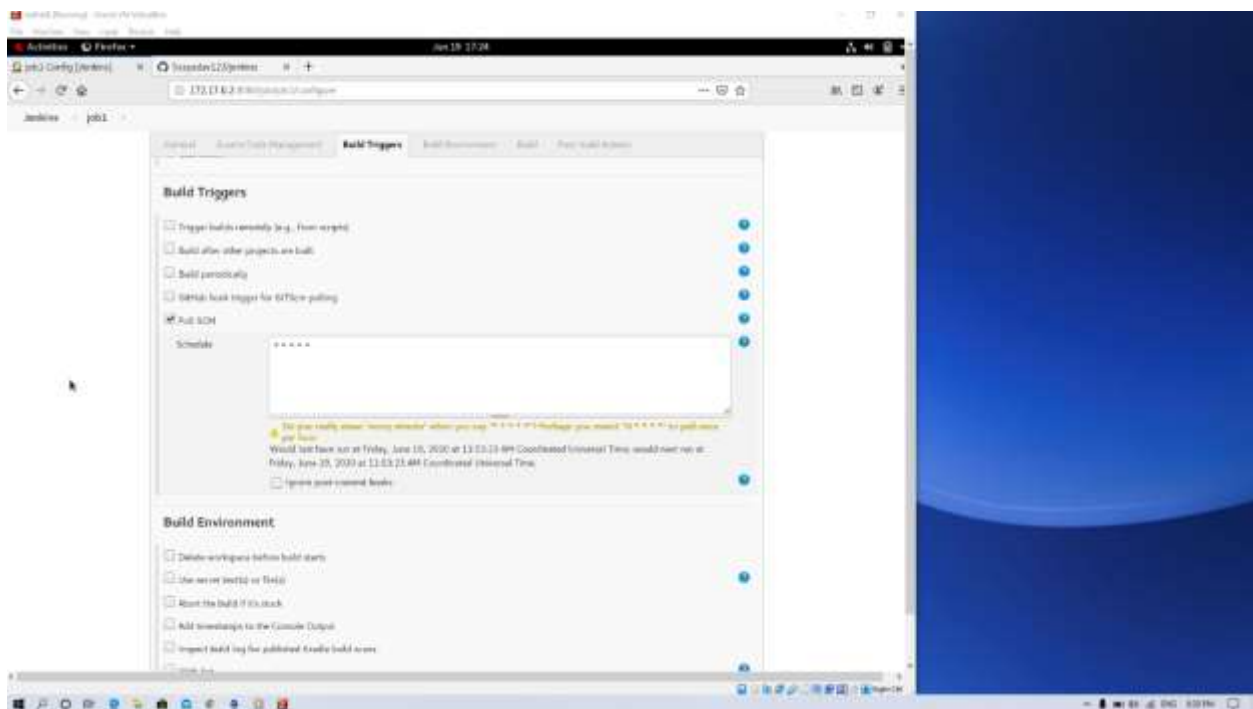
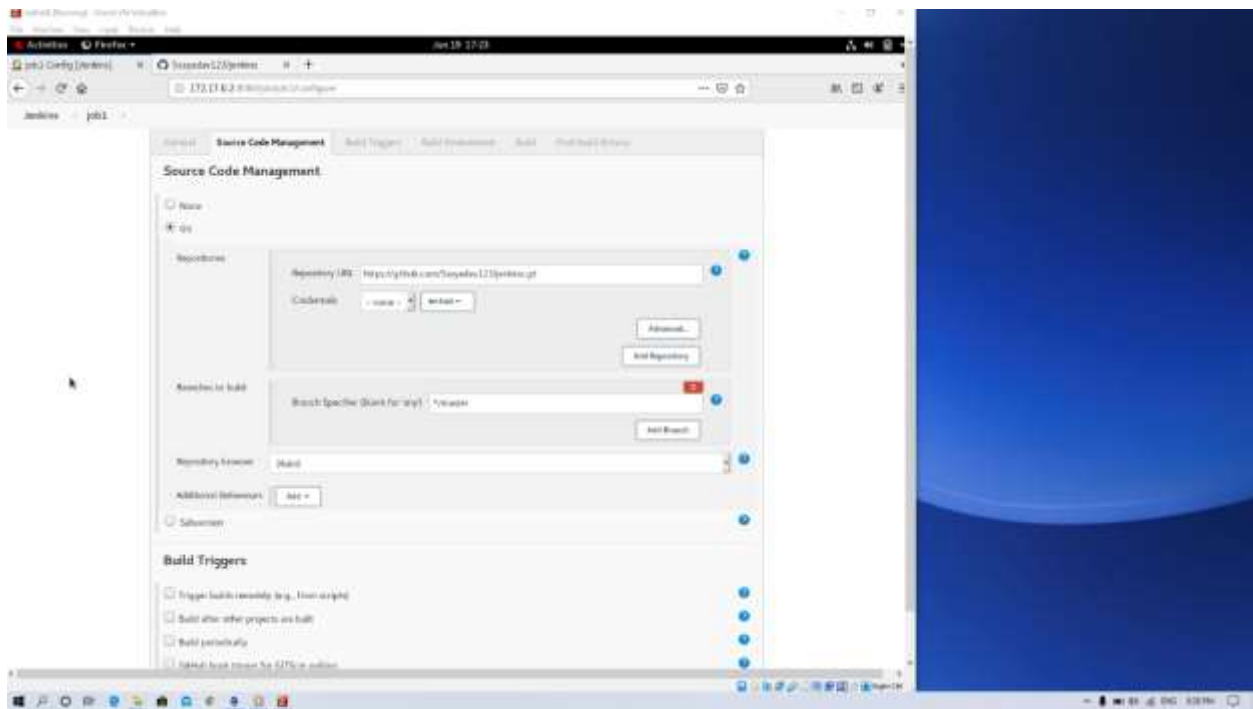
Jobs are created in this container and the necessary plugins like Build pipeline, Git, GitHub are installed.

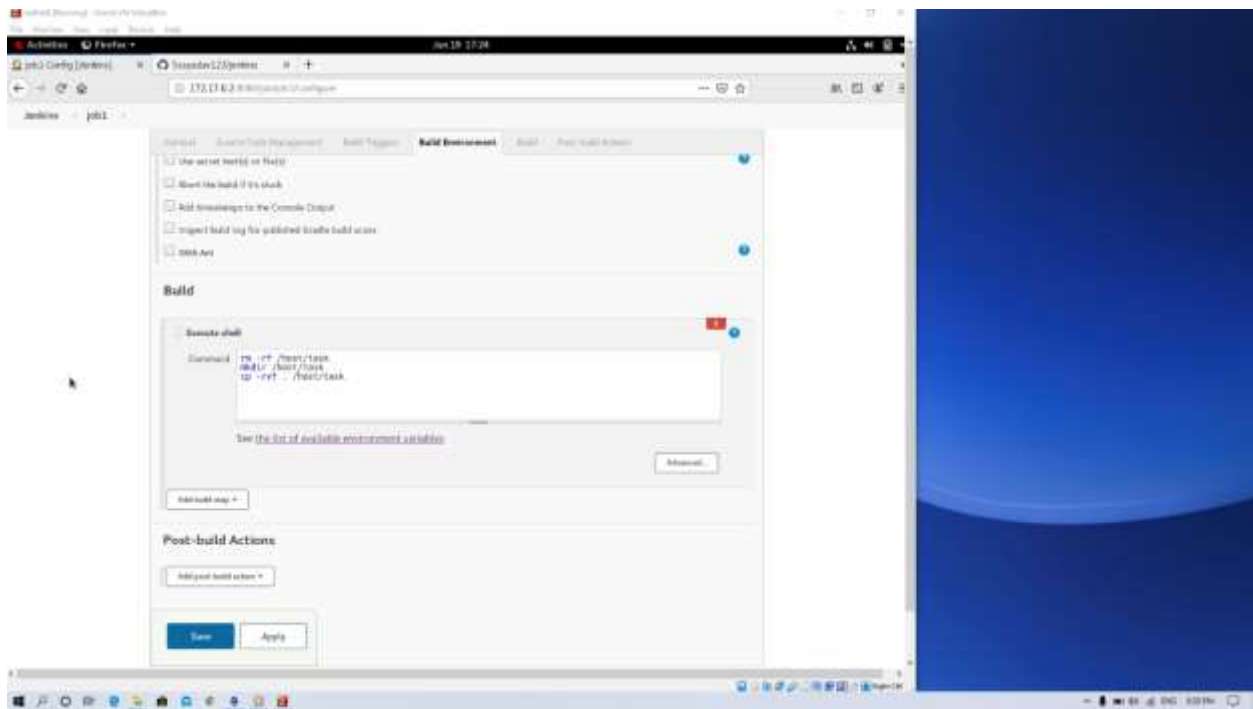


This is the dashboard of Jenkins

## 1. JOB 1

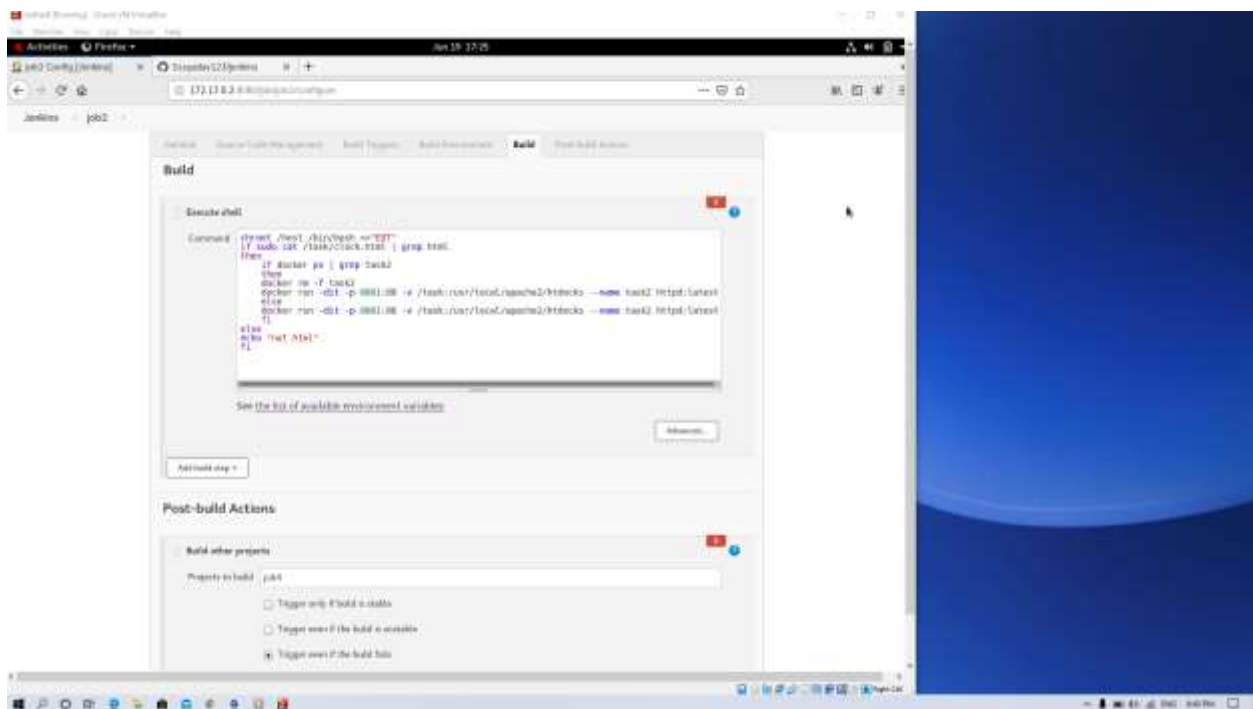
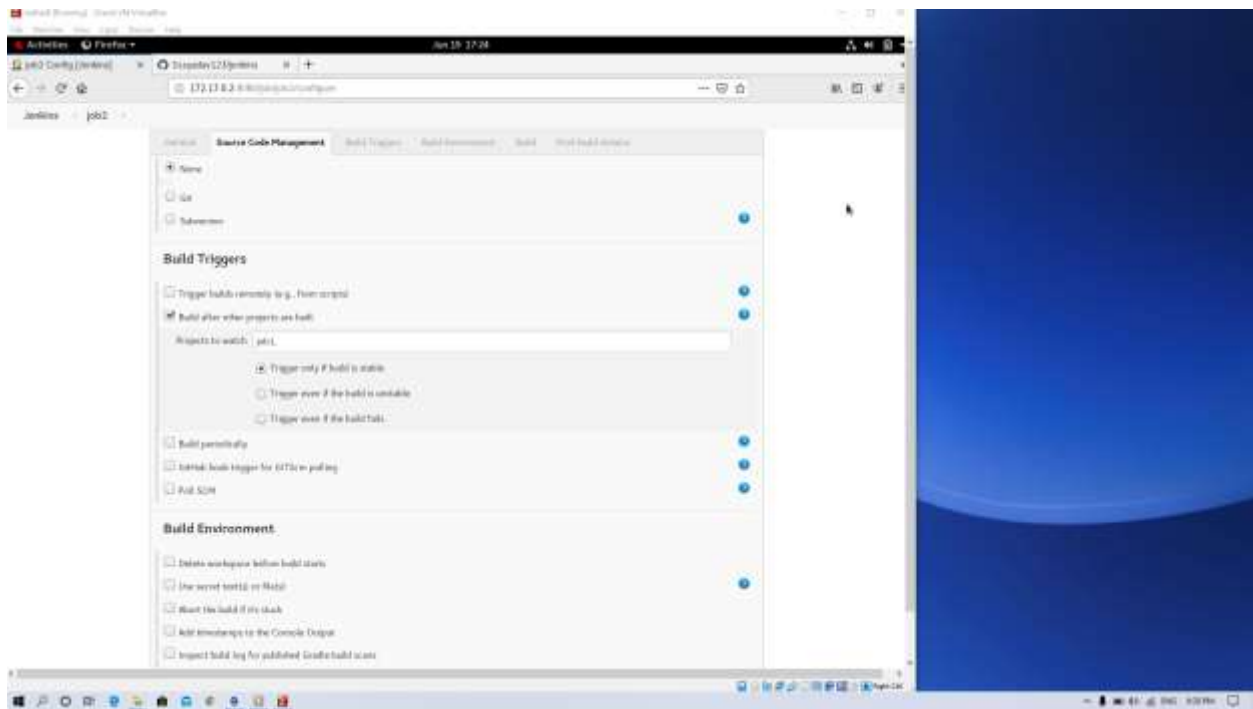
Pull the GitHub repo automatically when some developers push the repo to GitHub.





## 2. JOB 2:

By looking at the code or program file, Jenkins should automatically start the respective language interpreter install image container to deploy code ( eg. If code is of PHP, then Jenkins should start the container that has PHP already installed ).



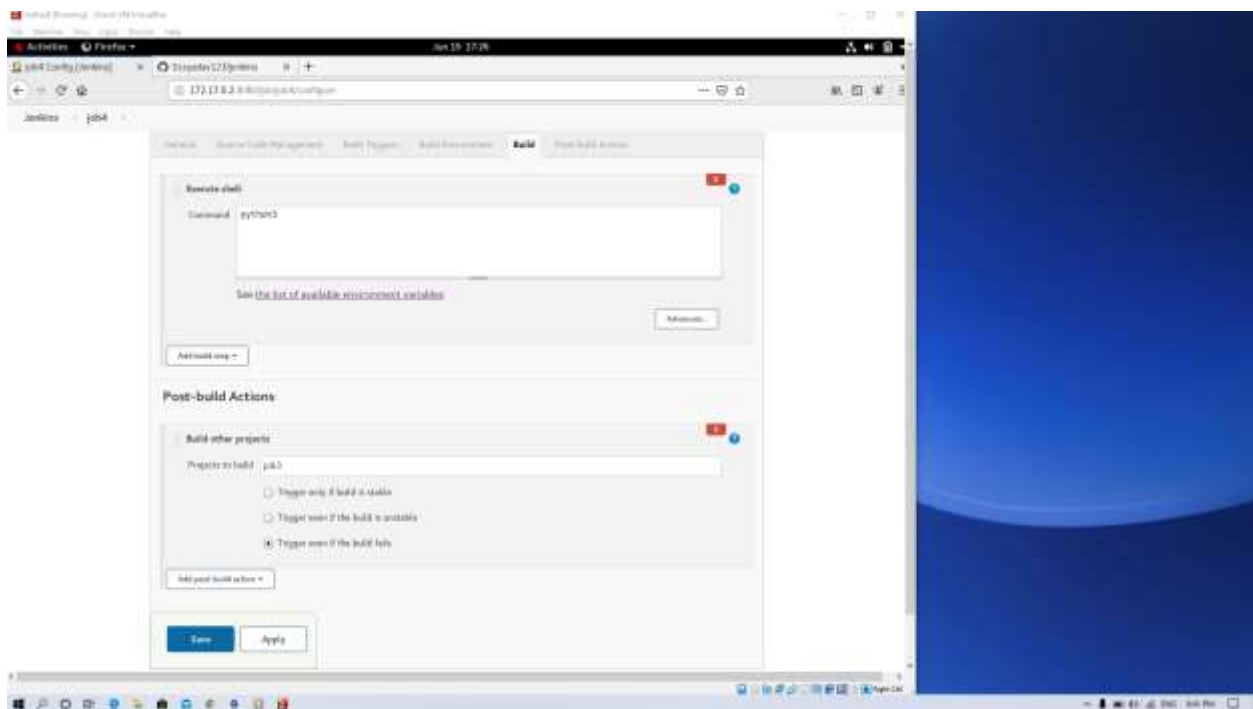
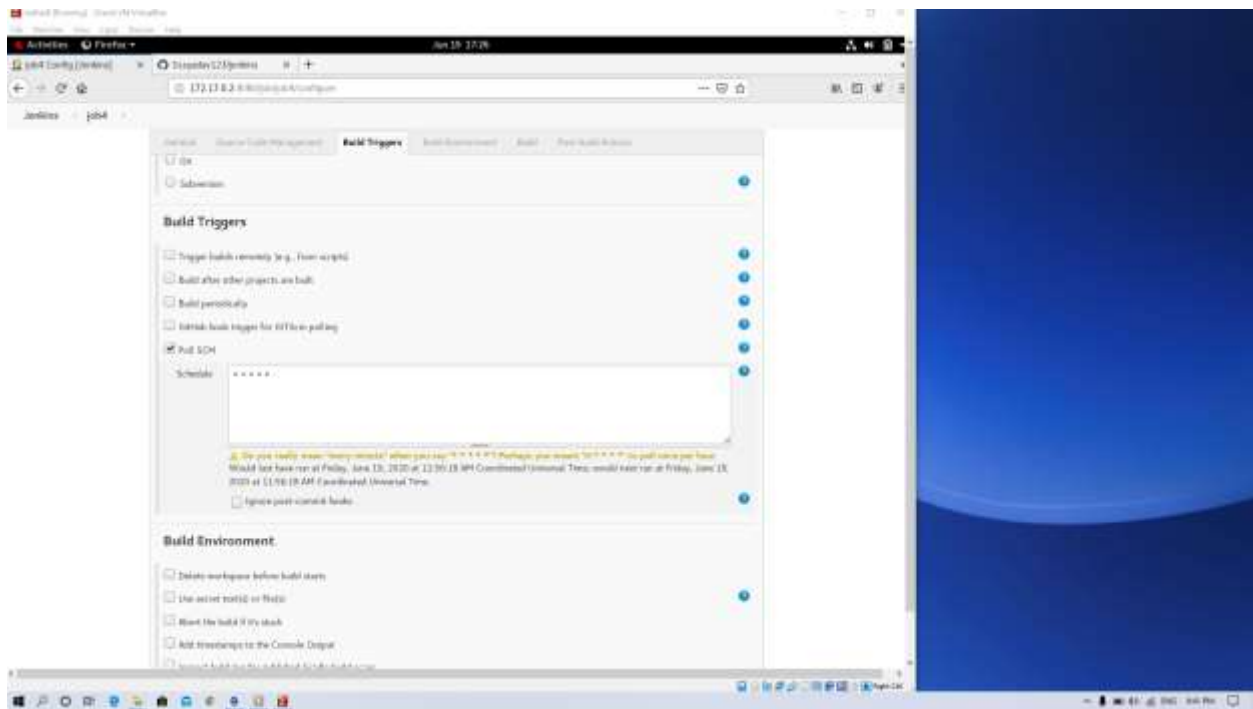
### 3. JOB 3

Test your app if it is working or not.









If the container where the app is running. fails due to any reason then this job should automatically start the container again.





After this install BUILD PIPELINE to look your all jobs systematic

### Build Pipeline

-  Run
-  History
-  Configure
-  Add Step
-  Delete
-  Manage

