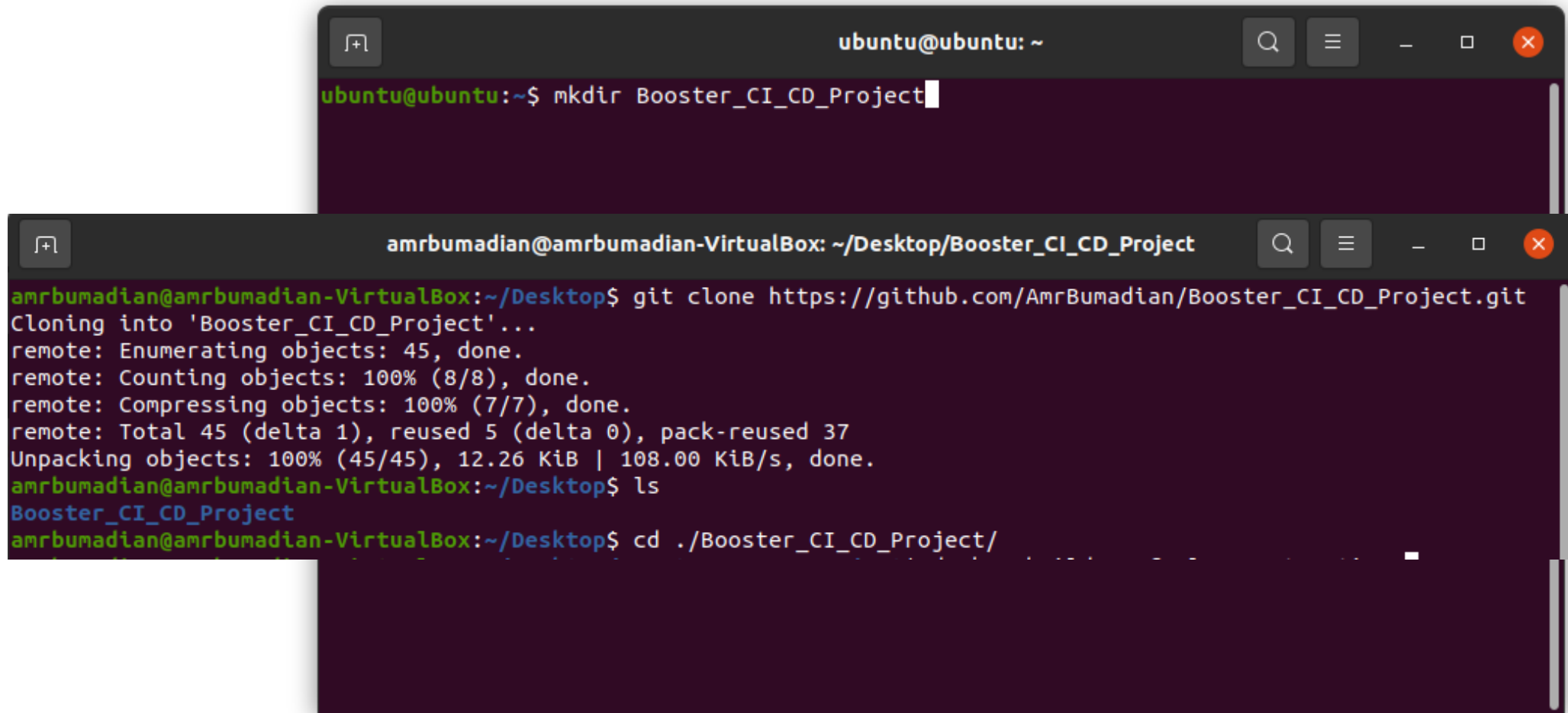
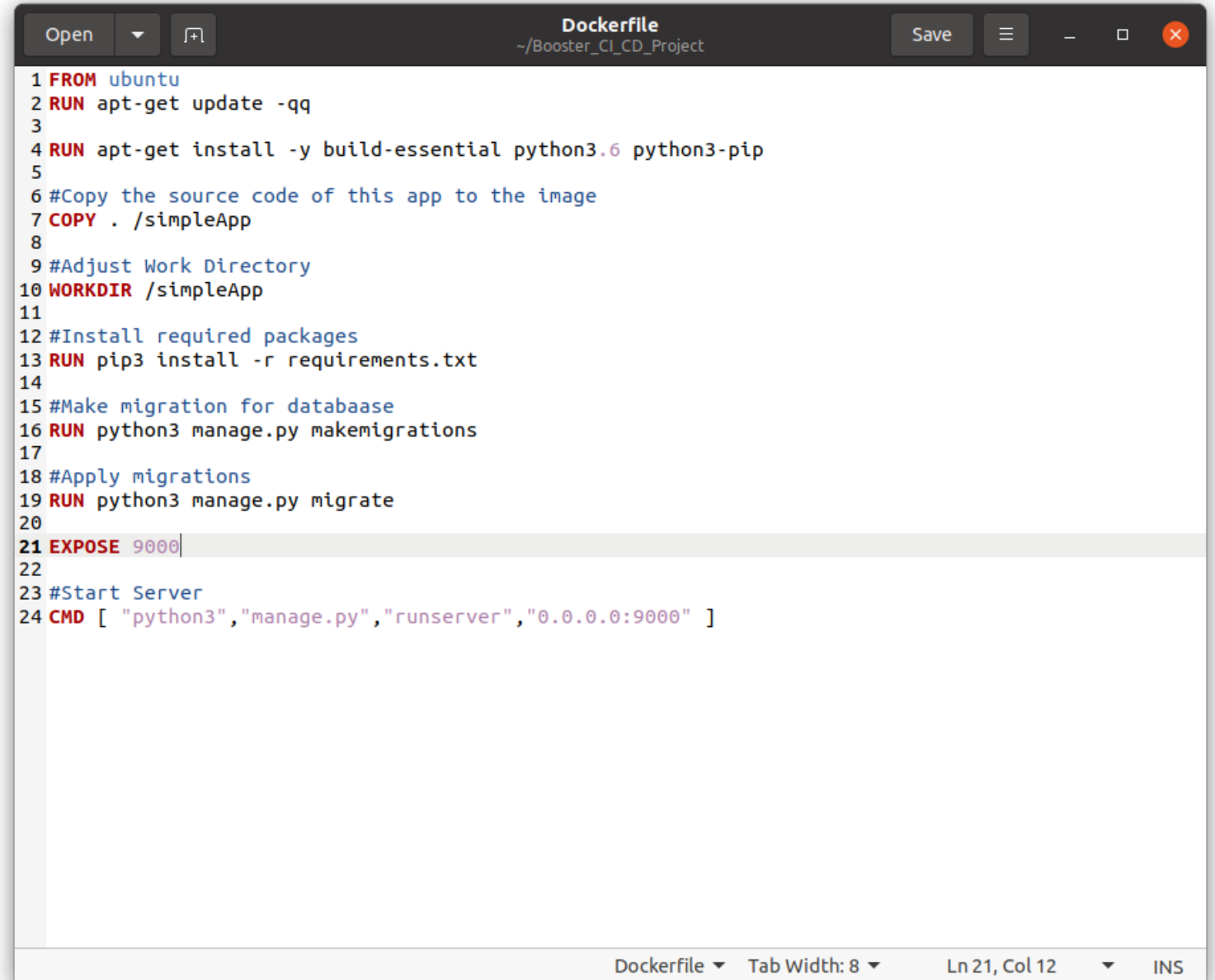


1-Create a new directory for the project, then clone the forked repo form the GitHub repository to the directory.



```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ mkdir Booster_CI_CD_Project  
  
amrbumadian@amrbumadian-VirtualBox: ~/Desktop/Booster_CI_CD_Project  
amrbumadian@amrbumadian-VirtualBox:~/Desktop$ git clone https://github.com/AmrBumadian/Booster_CI_CD_Project.git  
Cloning into 'Booster_CI_CD_Project'...  
remote: Enumerating objects: 45, done.  
remote: Counting objects: 100% (8/8), done.  
remote: Compressing objects: 100% (7/7), done.  
remote: Total 45 (delta 1), reused 5 (delta 0), pack-reused 37  
Unpacking objects: 100% (45/45), 12.26 KiB | 108.00 KiB/s, done.  
amrbumadian@amrbumadian-VirtualBox:~/Desktop$ ls  
Booster_CI_CD_Project  
amrbumadian@amrbumadian-VirtualBox:~/Desktop$ cd ./Booster_CI_CD_Project/
```

2-Create a new file “DockerFile” in the project directory for building the image.



```
1 FROM ubuntu
2 RUN apt-get update -qq
3
4 RUN apt-get install -y build-essential python3.6 python3-pip
5
6 #Copy the source code of this app to the image
7 COPY . /simpleApp
8
9 #Adjust Work Directory
10 WORKDIR /simpleApp
11
12 #Install required packages
13 RUN pip3 install -r requirements.txt
14
15 #Make migration for databaase
16 RUN python3 manage.py makemigrations
17
18 #Apply migrations
19 RUN python3 manage.py migrate
20
21 EXPOSE 9000
22
23 #Start Server
24 CMD [ "python3", "manage.py", "runserver", "0.0.0.0:9000" ]
```

Dockerfile ▾ Tab Width: 8 ▾ Ln 21, Col 12 ▾ INS

3-Run latest Jenkins image on port 8080 mapped to port 9090, then wait for the image to be pulled, if it is not already downloaded, and run.

```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ docker run -dp 9090:8080 jenkins/jenkins:lts  
Unable to find image 'jenkins/jenkins:lts' locally  
lts: Pulling from jenkins/jenkins  
955615a668ce: Downloading 13.48MB/54.93MB  
cd7e184186ab: Downloading 14.53MB/52.74MB  
0fb38fc696f5: Download complete  
b3586488ea64: Download complete  
671c689e6abc: Download complete  
37ba6b191a9b: Verifying Checksum  
2f8e68e4f802: Download complete  
814bd5b7da6c: Downloading 101.8kB/375.4kB  
2993200d5c23: Waiting  
d53cd42987dd: Waiting  
8e1cc48a9d14: Waiting  
bd7c6ba71c68: Waiting  
e284d32302ab: Pulling fs layer  
8a16ae8d151f: Waiting  
00796c524746: Waiting  
f64a98782baa: Waiting  
60f379d61558: Waiting  
█
```

4-In the browser, open localhost:9090. That shall start Jenkins, login and create admin account. Then proceed to the dashboard.

Dashboard

New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

New View

Build Queue

No builds in the queue.

Build Executor Status

Manage Jenkins

Building on the controller node can be a security issue. You should set up distributed builds. See [the documentation](#).

Set up agentSet up cloudDismiss

System Configuration

Configure System

Configure global settings and paths.

Global Tool Configuration

Configure tools, their locations and automatic installers.

Manage Plugins


Add, remove, disable or enable plugins that can extend the functionality of Jenkins.


Manage Nodes and Clouds


Add, remove, control and monitor the various nodes that Jenkins runs jobs on.


Security


5-In the plugin manager, go to available and search for “Role-Based Authorization Strategy”, check install, download now and install after restart, wait for installing to finish then refresh the page.

 **Jenkins**





 **1**


 **Amr Bumadian Mahmoud**

 **log out**

Dashboard > Plugin Manager

 **Back to Dashboard**

 **Manage Jenkins**

 **Update Center**

Updates

Available

Installed

Advanced

Install ↑	Name	Version	Released
<input checked="" type="checkbox"/>	<div>Role-based Authorization Strategy</div> <div><div>Security</div><div>Authentication and User Management</div></div> <div>Enables user authorization using a Role-Based strategy. Roles can be defined globally or for particular jobs or nodes selected by regular expressions.</div>	3.2.0	2 mo 7 days ago
<input type="checkbox"/>	<div>CloudBees AWS Credentials</div> <div><div>aws</div></div> <div>Allows storing Amazon IAM credentials within the Jenkins Credentials API. Store Amazon IAM access keys (AWSAccessKeyId and AWSSecretKey) within the Jenkins Credentials API. Also support IAM Roles and IAM MFA Token.</div>	1.32	6 days 22 hr ago

Install without restart

Download now and install after restart

Update information obtained: 16 min ago

Check now

6-From the side panel chose new item > multibranch pipeline > choose git then add the link to the GitHub repo.
(Before that add 2 other branches to the repo, development and production)

7-Proceed and wait till the scan finished, the green check mark shall appear. If it took long to finish refresh the page it may be stuck.

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view creates a separate namespace, so you can have multiple things of the same name as long as they

Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.

Jenkins

search

?

1

Amr Bumadian Mahmoud

log out

Dashboard

Booster_Project

Scan Multibranch Pipeline

Up

Status

Configure

Scan Multibranch Pipeline Now

Scan Multibranch Pipeline Log

View as plain text

Multibranch Pipeline Events

Delete M

People

New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

Scan Multibranch Pipeline Log

Started
[Fri Oct 08 12:54:52 UTC 2021] Starting branch indexing...
> git --version # timeout=10
> git --version # 'git version 2.30.2'
> git ls-remote --symref -- https://github.com/AmrBumadian/Booster_CI_CD_Project.git # timeout=10
Creating git repository in /var/jenkins_home/caches/git-44a5d80d4ae322625580f13f8f00d4e6
> git init /var/jenkins_home/caches/git-44a5d80d4ae322625580f13f8f00d4e6 # timeout=10
Setting origin to https://github.com/AmrBumadian/Booster_CI_CD_Project.git
> git config remote.origin.url https://github.com/AmrBumadian/Booster_CI_CD_Project.git # timeout=10
Fetching & pruning origin...
Listing remote references...
> git config --get remote.origin.url # timeout=10

Jenkins

search

?

1

Amr Bumadian Mahmoud

log out

All

+

add description

S	W	Name ↓	Last Success	Last Failure	Last Duration
		Booster_Project	6 min 1 sec - log	N/A	23 sec

Icon: S M L

Legend

Atom feed for all

Atom feed for failures

Atom feed for just latest builds

8-Create a slave automation file to help automate the process of setting up the Jenkins environment and add the docker client in the slave to be able to use docker, instead of manually setting up the Jenkins environment.

```
1 FROM ubuntu
2 USER root
3
4 # setup jenkins environment
5 RUN mkdir -p jenkins_home
6 RUN chmod 777 jenkins_home
7 RUN apt-get update -qq
8 RUN apt-get install -y apt-utils
9
10 #install open jdk
11 RUN apt-get install openjdk-8-jdk -qq
12
13 #install ssh server
14 RUN apt-get install openssh-server -qq
15 RUN useradd -ms /bin/bash jenkins
16
17 # install docker client
18 RUN apt-get install -qqy \
19     apt-transport-https \
20     ca-certificates \
21     curl \
22     gnupg \
23     software-properties-common
24 RUN curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -
25 RUN add-apt-repository \
26     "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
27     $(lsb_release -cs) \
28     stable"
29 RUN apt-get update -qq \
30     && apt-get install -y docker-ce docker-ce-cli containerd.io
31 RUN usermod -aG docker jenkins
32
33
34 WORKDIR jenkins_home
35 CMD [ "/bin/bash" ]
36
```

Continue step 8:

-Build the image.

-Run the image to start at bash.

-cd ~

-mkdir jenkins

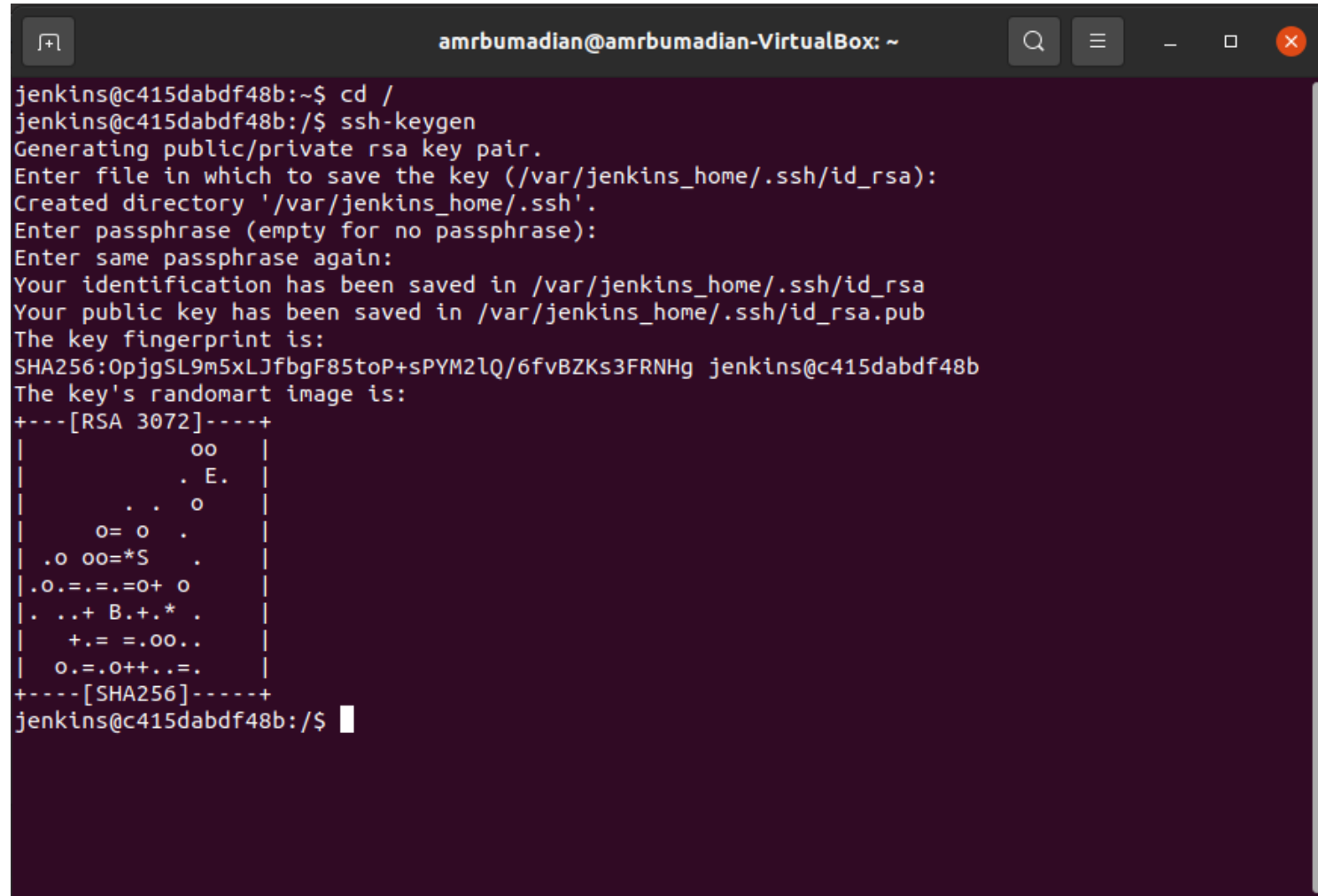
```
amrbumadian@amrbumadian-VirtualBox: ~/Desktop/Booster_CI_CD_Project
amrbumadian@amrbumadian-VirtualBox:~/Desktop/Booster_CI_CD_Project$ docker build . -f slave_automation
-t slave_automation:latest
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/build?buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=slave_automation&labels=%7B%7D&memory=0&memswap=0&networkmode=default&rm=1&shmsize=0&t=slave_automation%3Alatest&target=&ulimits=null&version=1": dial unix /var/run/docker.sock: connect: permission denied
amrbumadian@amrbumadian-VirtualBox:~/Desktop/Booster_CI_CD_Project$ sudo docker build . -f slave_automation
-t slave_automation:latest
```

```
root@0a959c2d517e: ~
amrbumadian@amrbumadian-VirtualBox:~/Desktop/Booster_CI_CD_Project$ sudo docker run -it -v /var/run/docker.sock:/var/run/docker.sock slave_automation bash
root@0a959c2d517e:/jenkins_home# cd ~
root@0a959c2d517e:~# mkdir jenkins
root@0a959c2d517e:~#
```


Continue step 8:

-cd /

-run ssh-keygen to generate public and private keys, then press “Enter” 3 times.

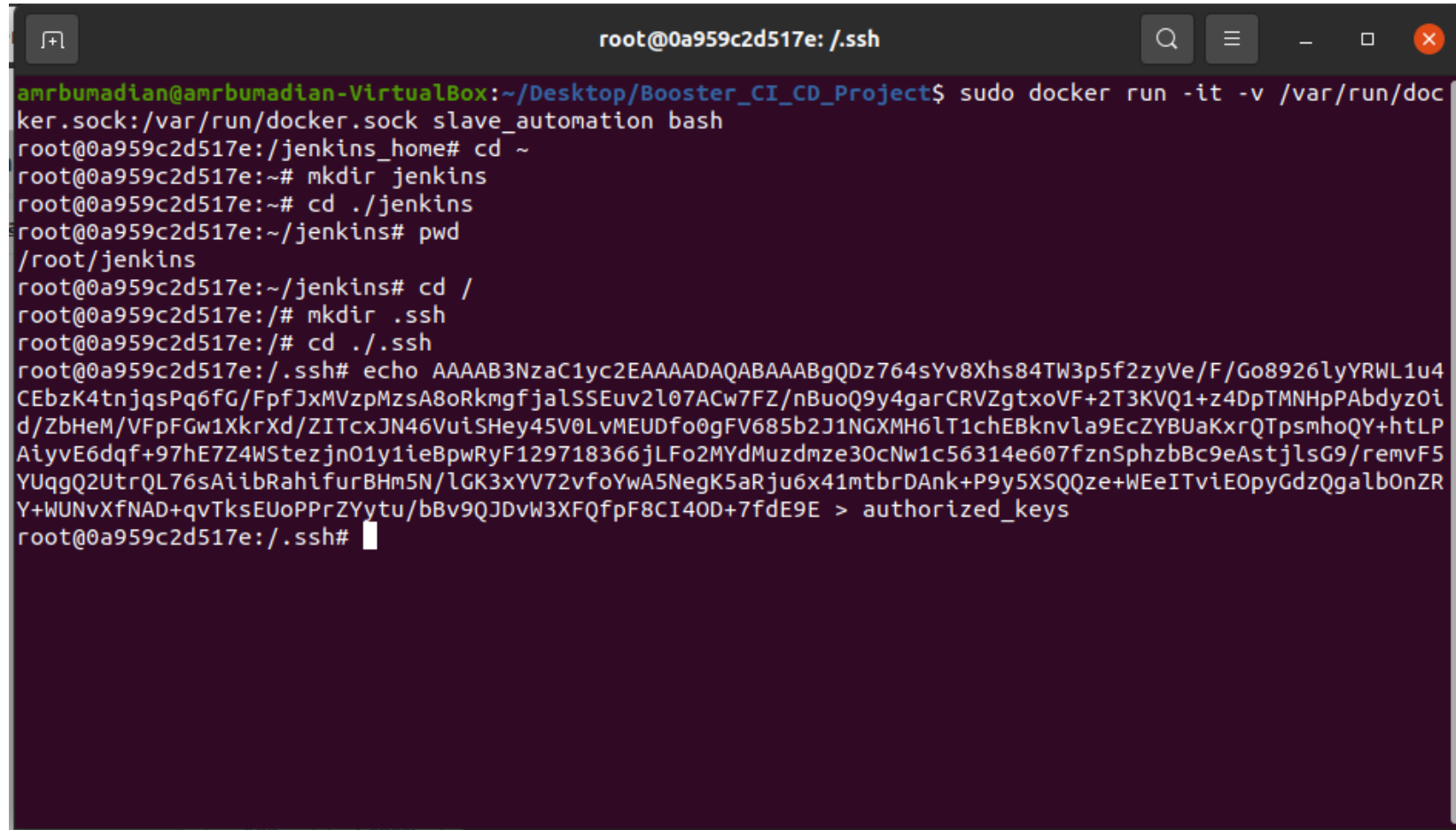
A terminal window titled 'amrbumadian@amrbumadian-VirtualBox: ~' with standard window controls. The terminal shows the execution of 'ssh-keygen' to generate an RSA key pair. The user is prompted to enter a file path, a passphrase, and confirm the passphrase. The output shows the key is saved in '/var/jenkins_home/.ssh/id_rsa' and the public key in '/var/jenkins_home/.ssh/id_rsa.pub'. The key fingerprint is displayed as 'SHA256:OpjgSL9m5xLJfbgF85toP+sPYM2lQ/6fvBZKs3FRNHg jenkins@c415dabdf48b'. A randomart image for the RSA 3072 key is shown, consisting of a grid of characters. The terminal ends with the prompt 'jenkins@c415dabdf48b:/\$' and a cursor.

```
jenkins@c415dabdf48b:~$ cd /
jenkins@c415dabdf48b:/$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/var/jenkins_home/.ssh/id_rsa):
Created directory '/var/jenkins_home/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/jenkins_home/.ssh/id_rsa
Your public key has been saved in /var/jenkins_home/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:OpjgSL9m5xLJfbgF85toP+sPYM2lQ/6fvBZKs3FRNHg jenkins@c415dabdf48b
The key's randomart image is:
+---[RSA 3072]-----+
|           oo       |
|          . E.      |
|         . .  o     |
|        o= o   .    |
|       .o oo=*S     |
|      .o.=.=.o+ o   |
|     . .+ B.+.*    |
|      +.= =.oo..   |
|     o.=.o++..=    |
+---[SHA256]-----+
jenkins@c415dabdf48b:/$
```

Continue step 8:

Open a new terminal and run bash in the slave automation container already running.


- cd ~
- mkdir Jenkins
- cd ./Jenkins
 - Pwd (we will need this path in creating a node)
- cd /
- mkdir .ssh
- cd ./ssh
- Echo "The public ssh key" > authorized_keys





```
root@0a959c2d517e: /.ssh
amrbumadian@amrbumadian-VirtualBox: ~/Desktop/Booster_CI_CD_Project$ sudo docker run -it -v /var/run/docker.sock:/var/run/docker.sock slave_automation bash
root@0a959c2d517e:/jenkins_home# cd ~
root@0a959c2d517e:~# mkdir jenkins
root@0a959c2d517e:~# cd ./jenkins
root@0a959c2d517e:~/jenkins# pwd
/root/jenkins
root@0a959c2d517e:~/jenkins# cd /
root@0a959c2d517e:/# mkdir .ssh
root@0a959c2d517e:/# cd ./ssh
root@0a959c2d517e:/ssh# echo AAAAB3NzaC1yc2EAAAADAQABAAQGDz764sYv8Xhs84TW3p5f2zyVe/F/Go8926lyYRWL1u4CEbzK4tnjqsPq6fG/FpfJxMVzpMzsA8oRkmgfjaLSSEuv2l07ACw7FZ/nBuoQ9y4garCRVZgtxoVF+2T3KVQ1+z4DpTMNHpPAbdyzOid/ZbHeM/VFpFGw1XkrXd/ZITcxJN46VuiSHey45V0LvMEUDfo0gFV685b2J1NGXMH6LT1chEBknvLa9EcZYBUaKxrQTpsmhoQY+htLPAiyvE6dqf+97hE7Z4WStezjn01y1ieBpwRyF129718366jLFo2MYdMuzdmze30cNw1c56314e607fznSphzbBc9eAstjlsG9/remvF5YUqgQ2UtrQL76sAibRahifurBHm5N/LGK3xYV72vfoYwA5NegK5aRju6x41mtbrDAnk+P9y5XSQQze+WEEITviEOpyGdzQgalb0nZRY+WUNvXfNAD+qvTksEUoPPrZYytU/bBv9QJDvW3XFQfpF8CI40D+7fdE9E > authorized_keys
root@0a959c2d517e:/ssh#
```


9-From the dashboard choose manage nodes and clouds, the from the side panel choose add new node.


System Configuration


**Configure System**
Configure global settings and paths.


**Global Tool Configuration**
Configure tools, their locations and automatic installers.


**Manage Plugins**
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.


**Manage Nodes and Clouds**
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.



 [Back to Dashboard](#)

 [Manage Jenkins](#)



 [New Node](#)

 [Configure Clouds](#)

 [Node Monitoring](#)

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free
	master	Linux (amd64)	In sync	 940.29 MB	
	Data obtained	2 min 19 sec	2 min 19 sec	2 min 21 sec	

Build Queue (3) ^

part of Booster_Project » production #1  

Continue step 9:

- Docker inspect <id of slave_automation container>
- Copy the IpAddress that will appear below. (will be needed in configuring the node)

```
amrbumadian@amrbumadian-VirtualBox:~$ ^C
amrbumadian@amrbumadian-VirtualBox:~$ sudo docker inspect 0a959c2d517e
[
  {
    "Id": "0a959c2d517efd8a43018e5447d2e7ea797dff9f14280827fe99ec395c03cb7e",
    "Created": "2021-10-08T13:11:33.473652111Z",
    "Path": "bash",
    "Args": [],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 25435,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2021-10-08T13:11:34.894752464Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "NetworkSettings": {
      "Bridge": "veth",
      "Gateway": "172.17.0.1",
      "IPAddress": "172.17.0.3",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "MacAddress": "02:42:ac:11:00:03",
      "DriverOpts": null
    }
  }
]
```

Continue step 9:

- Name the node
- Add the copied IpAddress in the host then choose add credentials.
- Configure the rest as shown.

Name?

Docker

Description?

Number of executors?

1

Launch method?

Launch agents via SSH

Host?

172.17.0.3

⊘

 The Host must be specified

Credentials?

- none -

⚙️

 Add

⊘

 The selected credentials cannot be found

Host Key Verification Strategy?

Known hosts file Verification Strategy

Continue step 9:

- Add the credentials, copy the generated private key and add it in the key input box

- Add the save the node.

Add Credentials

Domain

Global credentials (unrestricted)

Kind

SSH Username with private key

Scope

Global (Jenkins, nodes, items, all child items, etc)

ID

slave

Description

Username

root

☐ Treat username as secret

Private Key

☒ Enter directly

Key

No Stored Value

Add

Passphrase

Add Cancel

Continue step 9:

- Check that the node is successfully authenticated and connected.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	docker	Linux (amd64)	In sync	14.24 GB	317.38 MB	14.24 GB	1432ms 
	master	Linux (amd64)	In sync	14.24 GB	316.38 MB	14.24 GB	0ms 
Data obtained		13 min	13 min	13 min	13 min	13 min	13 min
Refresh status							


10-Write the Jenkins file, we will stage it as following: preparation, build, push, deploy, and notification for slack.




```
1 pipeline{
2   agent {label "docker"}
3
4   stages {
5
6     stage('preparation'){
7       steps {
8         git 'https://github.com/AmrBumadian/Booster_CI_CD_Project.git'
9       }
10
11     }
12
13     stage('build image'){
14       steps {
15         sh 'docker build . -f Dockerfile -t amrbumadian/booster_django_app:latest'
16       }
17     }
18
19     stage('push image'){
20       steps {
21         withCredentials([usernamePassword(credentialsId:"dockerhub",usernameVariable:"USERNAME",passwordVariable:"PASSWORD")]) {
22           sh """
23             docker login -u ${USERNAME} -p ${PASSWORD}
24             docker push amrbumadian/booster_django_app:latest
25             """
26         }
27       }
28     }
29
30     stage('deploy'){
31       steps {
32         sh 'docker run -d -p 9000:9000 amrbumadian/booster_django_app:latest'
33       }
34       post {
35         success {
36           slackSend (color: '#00FF00', message: "SUCCESSFUL")
37         }
38       }
39     }
40   }
41 }
```


Continue step 10:

-Add login credentials for the DockerHub in the global credentials.

-Build the pipeline.


 **Advanced Image Management**
View all your images and tags in this repository, clean up unused content, recover untagged images. Available for Pro and Team accounts. [View preview](#)



 **amrbumadian / booster_django_app**
This repository does not have a description 
 Last pushed: a minute ago

Docker commands [Public View](#)
To push a new tag to this repository,

```
docker push amrbumadian/booster_django_app:tagname
```

Tags and Scans
This repository contains 1 tag(s).

 **VULNERABILITY SCANNING - DISABLED**
[Enable](#)

TAG	OS	PULLED	PUSHED
 latest		a minute ago	a minute ago

Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available on Pro and Team plans.
[Upgrade to Pro](#) [Learn more](#)

django

[View release notes](#) for Django 3.0

The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

11-Connect to the website at localhost:8000

12-Add Slack plugin

The screenshot displays the Jenkins configuration interface. At the top, a breadcrumb trail shows 'Dashboard' followed by 'configuration'. The main configuration area is divided into sections. The 'E-mail Notification' section includes fields for 'SMTP server' and 'Default user e-mail suffix', each with a help icon. Below these is a checkbox for 'Test configuration by sending test e-mail' and an 'Advanced...' button. The 'Slack' section follows, with a 'Workspace' field containing 'sprintsglobal'. Below this is a 'Credential' dropdown menu set to 'Slack' with an 'Add' button. The 'Default channel / member id' field contains 'djangoapp'. At the bottom of this section is a checkbox for 'Custom slack app bot user'. To the right of the configuration fields is a 'Default Triggers...' button. Below the configuration area, a chat preview shows a message from the 'jenkins' app at 4:17 PM: 'Slack/Jenkins plugin: you're all set on <http://localhost:9090/>'. Below the message are two speech bubbles: 'Hello, team!' and 'Let's use this channel for...'. At the bottom of the chat preview is a text input field with the placeholder 'Send a message to #djangoapp' and a rich text editor toolbar.

Dashboard > configuration

Content Token Reference ? Default Triggers...

E-mail Notification

SMTP server

Default user e-mail suffix ?

☐ Test configuration by sending test e-mail Advanced...

Slack

Workspace ?

sprintsglobal

Credential ?

Slack Add

Default channel / member id

djangoapp

☐ Custom slack app bot user

Save Apply

jenkins APP 4:17 PM
Slack/Jenkins plugin: you're all set on <http://localhost:9090/>

Hello, team! Let's use this channel for... ×

Send a message to #djangoapp

B *I*

ALL DONE