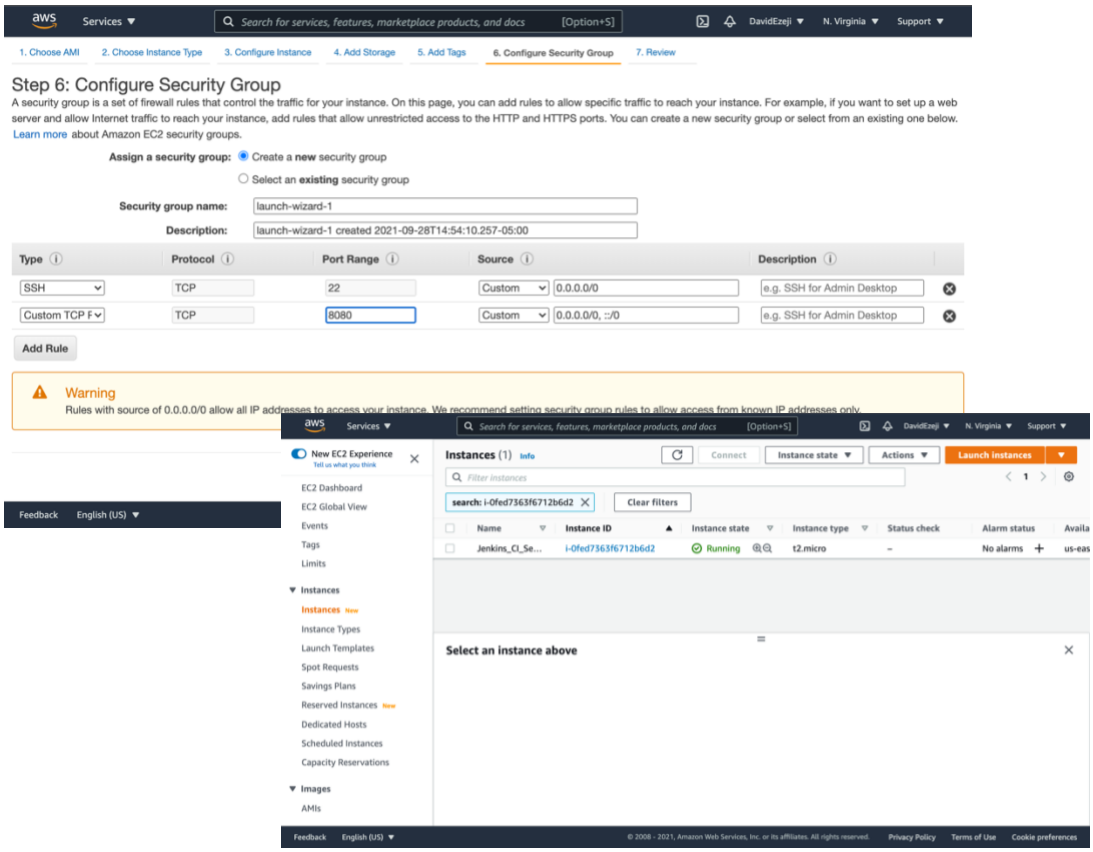
David Ezeji
11/04/2019

Jenkins Project

Overview: Spun up two Linux instances on AWS and installed Jenkins on the first instance so that I could automate the processes of building my application and deploying it to the second instance. On the second instance I created a Docker container to host my application.

1. Created the first Linux instance which will host Jenkins

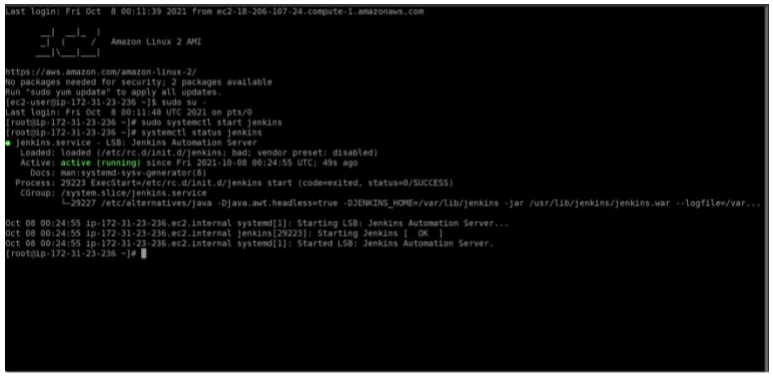


2. Installed Java onto the instance

3. Created a Java path within the .bash_profile directory, so that Jenkins would be able to find it once downloaded.

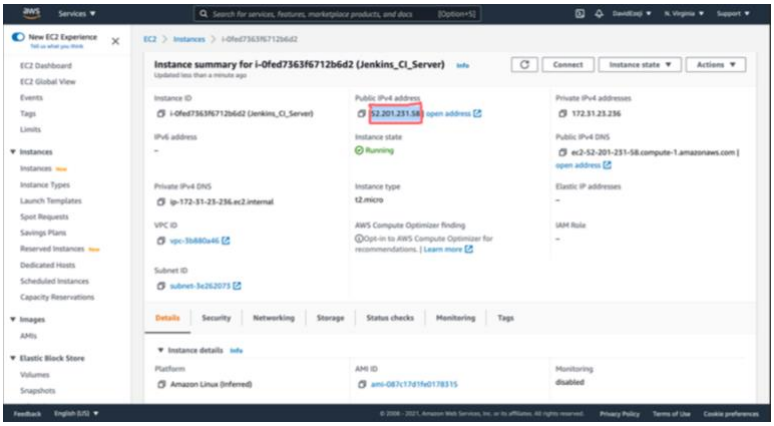
4. Installed Apache Maven onto my EC2 instance and set a path for it (Jenkins won't install without Maven being present)

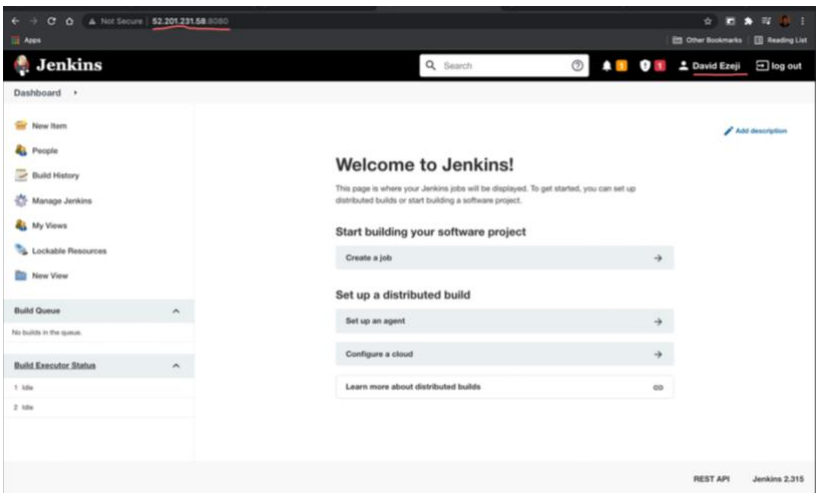
5. Installed Jenkins and started the Jenkins server



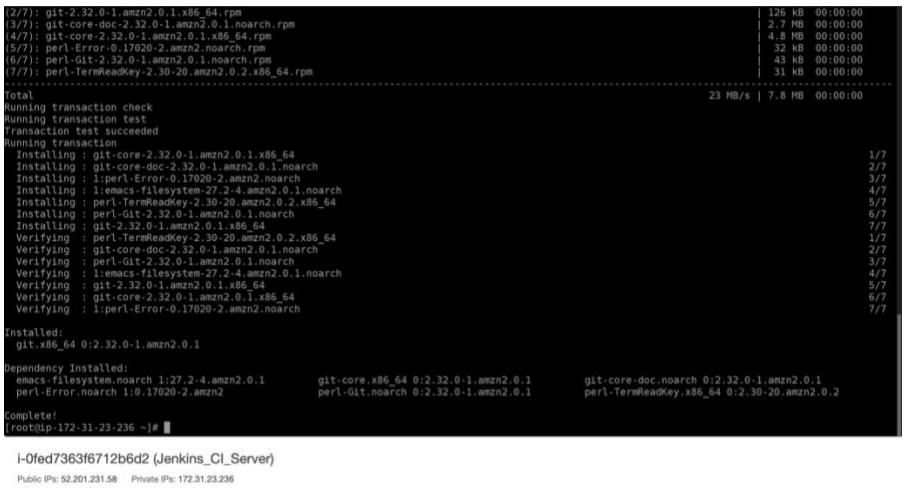
i-0fed7363f6712b6d2 (Jenkins_CI_Server)

Public IPs: 52.201.231.58 Private IPs: 172.31.23.236

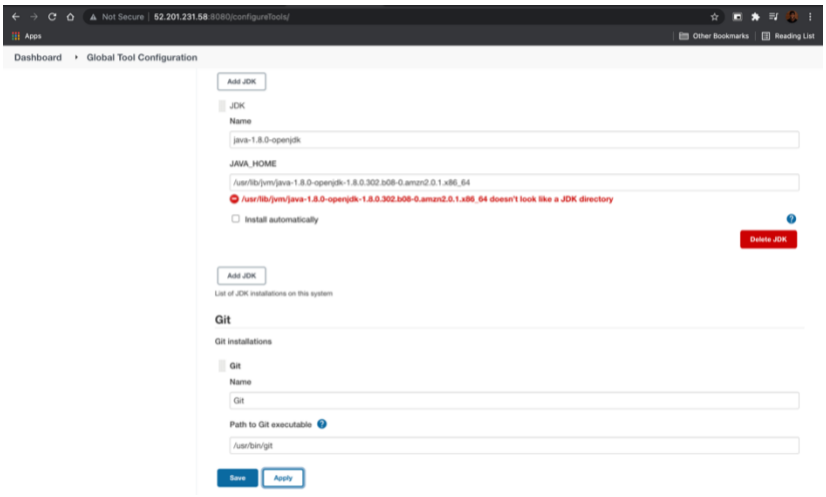


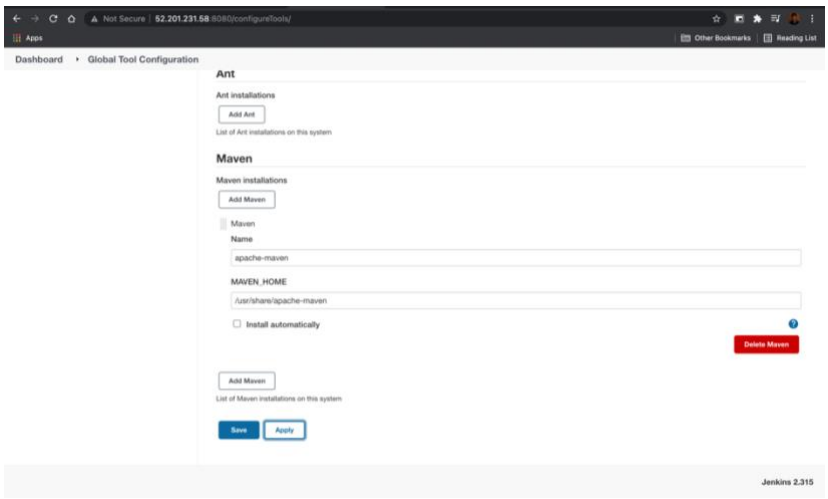


6. Installed Git on the EC2 instance

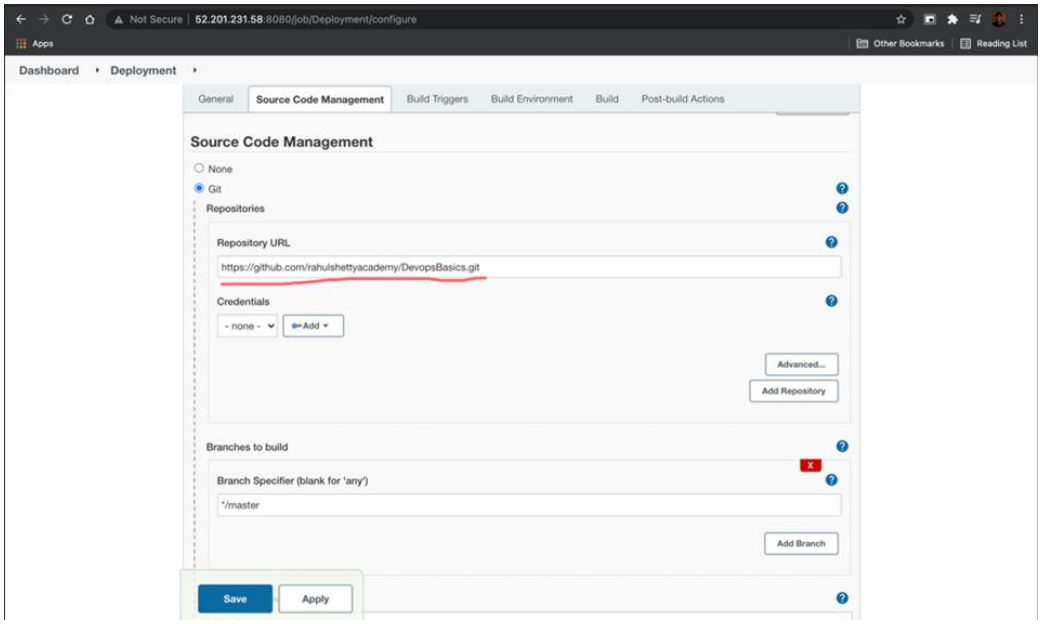


7. Created a new job for Jenkins that will automatically deploy apps. I then configured Jenkins using path variables.

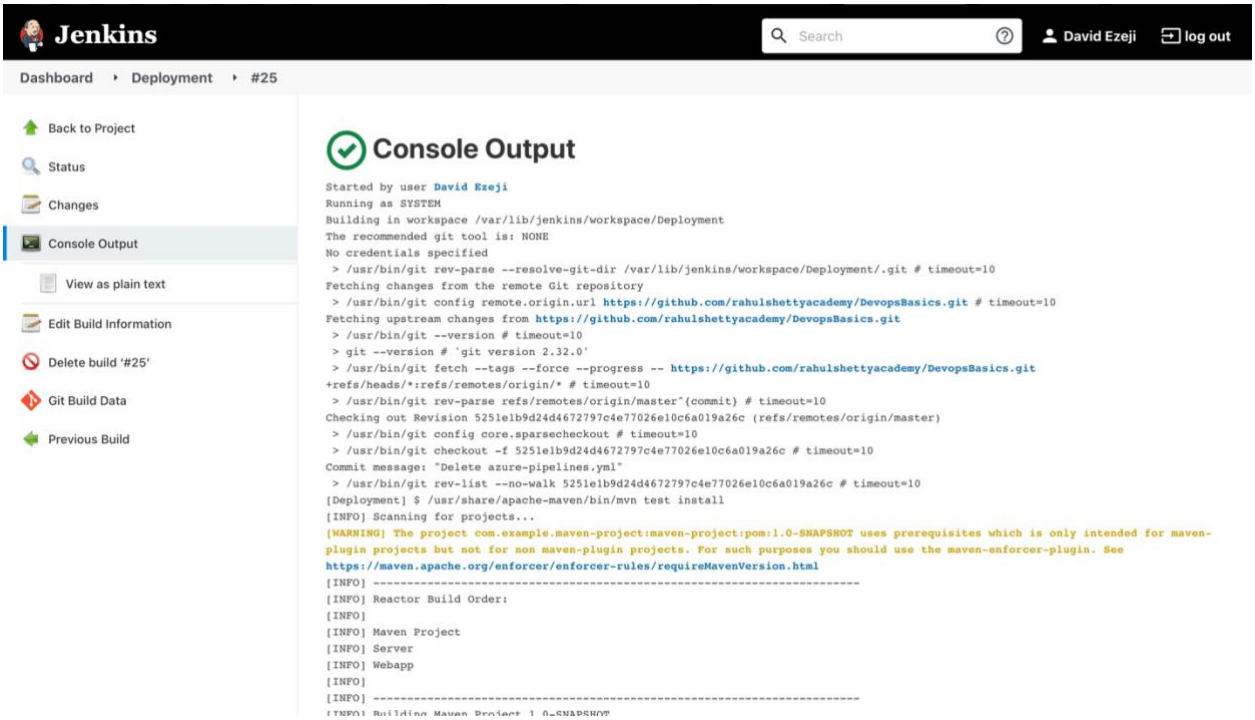




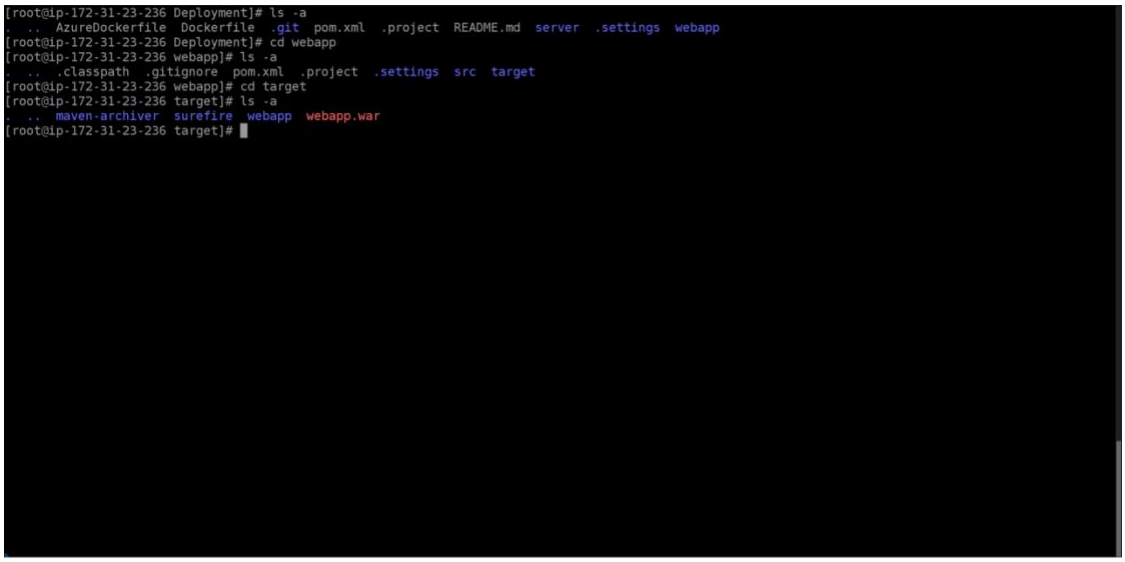
8. Sourced code that will be built/tested from a Github public repository



9. Built the project



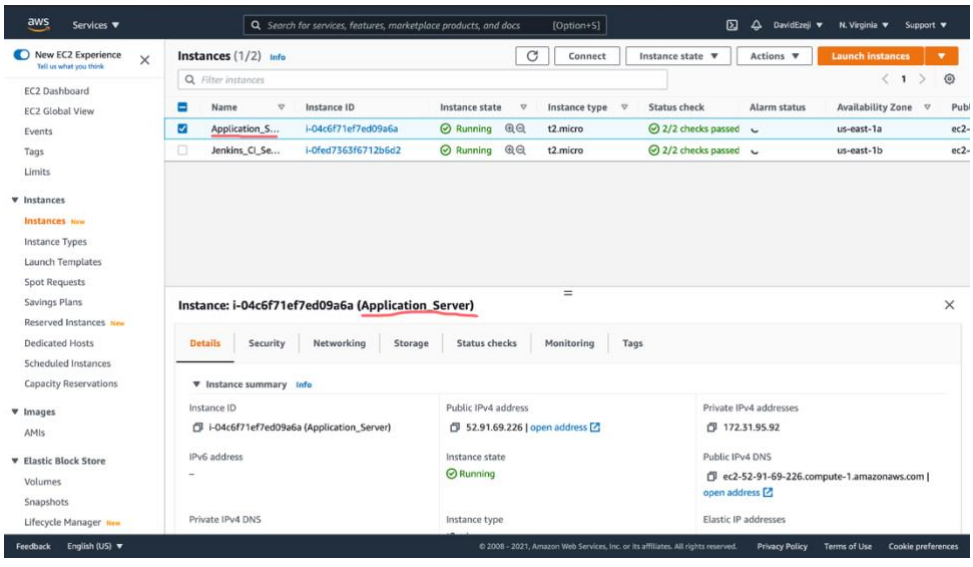
10. Jenkins project is now installed onto the ec2 instance and I have created a .war file



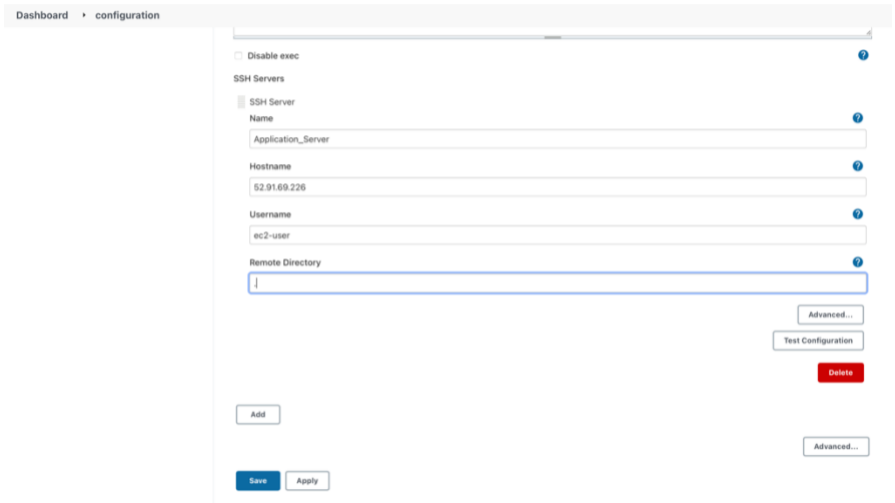
i-0fed7363f6712b6d2 (Jenkins_CI_Server)

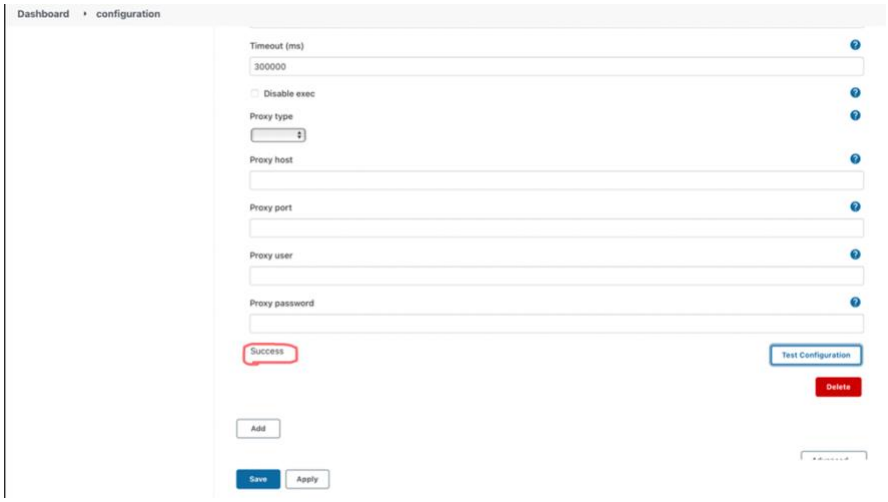
Public IPs: 52.201.231.58 Private IPs: 172.31.23.236

11. Spun up a second EC2 instance which will receive apps deployed from the first instance

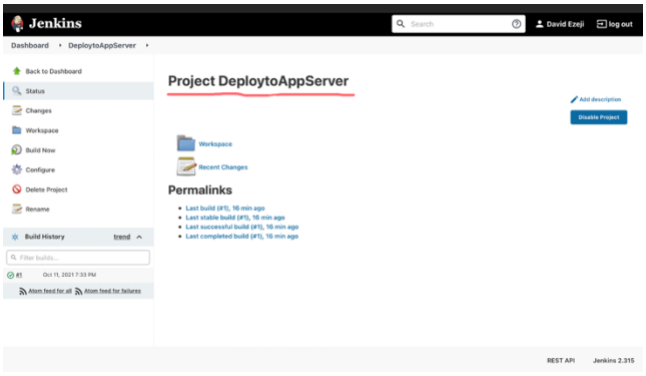


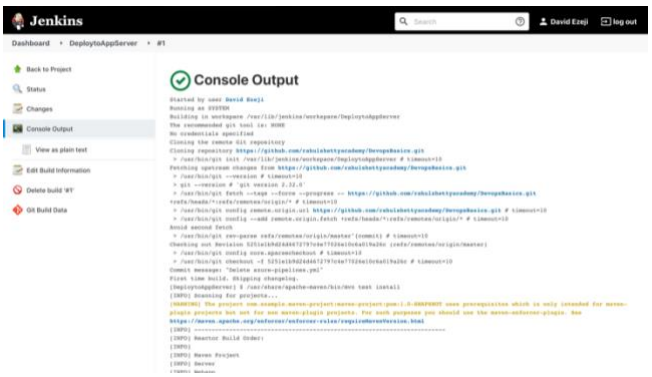
12. Connected the first instance (Jenkins_CI_Server) to the second instance (Application_Server) via Jenkins





13. Successfully deployed the application to the “Application_Server” instance by sending over the webapp.war file





```
Last login: Mon Oct 11 19:02:15 2021 from ec2-18-206-107-24.compute-1.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 15 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-95-92 ~]$ sudo su .
Last login: Mon Oct 11 19:02:20 UTC 2021 on pts/1
Last failed login: Mon Oct 11 19:43:12 UTC 2021 from 200.70.56.204 on ssh:notty
There were 4 failed login attempts since the last successful login.
[root@ip-172-31-95-92 ~]# ls -a
.  ..  .bash_logout  .bash_profile  .bashrc  .cshrc  .ssh  .tcshrc
[root@ip-172-31-95-92 ~]# sudo su ec2-user
[ec2-user@ip-172-31-95-92 root]$ cd ..
[ec2-user@ip-172-31-95-92 /]$ ls -a
.  ..  .autorelabel  bin  boot  dev  etc  home  lib  lib64  local  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[ec2-user@ip-172-31-95-92 /]$ cd root
bash: cd: root: Permission denied
[ec2-user@ip-172-31-95-92 /]$ cd ~
[ec2-user@ip-172-31-95-92 ~]$ ls -a
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc  .ssh  webapp.war
[ec2-user@ip-172-31-95-92 ~]$
```

i-04c6f71ef7ed09a6a (Application_Server)

Public IPs: 52.91.69.226 Private IPs: 172.31.95.92

14. Installed Docker onto the “Application_Server” instance

```
(4/5): runc-1.0.0-2.amzn2.x86_64.rpm                                                                | 3.3 MB  00:00:00
(5/5): docker-20.10.7-3.amzn2.x86_64.rpm                                                            |  42 MB  00:00:01
--------------------------------------------------------------------------------------------------------------------
Total                                                                                66 MB/s |  69 MB  00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : runc-1.0.0-2.amzn2.x86_64                                                                      1/5
  Installing : containerd-1.4.6-3.amzn2.x86_64                                                                2/5
  Installing : libcgroup-0.41-21.amzn2.x86_64                                                                 3/5
  Installing : pigz-2.3.4-1.amzn2.0.1.x86_64                                                                  4/5
  Installing : docker-20.10.7-3.amzn2.x86_64                                                                  5/5
  Verifying  : docker-20.10.7-3.amzn2.x86_64                                                                  1/5
  Verifying  : containerd-1.4.6-3.amzn2.x86_64                                                                2/5
  Verifying  : pigz-2.3.4-1.amzn2.0.1.x86_64                                                                  3/5
  Verifying  : runc-1.0.0-2.amzn2.x86_64                                                                      4/5
  Verifying  : libcgroup-0.41-21.amzn2.x86_64                                                                 5/5

Installed:
  docker.x86_64 0:20.10.7-3.amzn2

Dependency Installed:
  containerd.x86_64 0:1.4.6-3.amzn2      libcgroup.x86_64 0:0.41-21.amzn2      pigz.x86_64 0:2.3.4-1.amzn2.0.1      runc.x86_64 0:1.0.0-2.amzn2

Complete!
[root@ip-172-31-95-92 ~]# usermod -aG docker ec2-user
[root@ip-172-31-95-92 ~]# sudo su ec2-user
[ec2-user@ip-172-31-95-92 root]$ cd ..
[ec2-user@ip-172-31-95-92 /]$ cd home
[ec2-user@ip-172-31-95-92 home]$ cd ec2-user
[ec2-user@ip-172-31-95-92 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-172-31-95-92 ~]$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[ec2-user@ip-172-31-95-92 ~]$
```

i-04c6f71ef7ed09a6a (Application_Server)

Public IPs: 52.91.69.226 Private IPs: 172.31.95.92

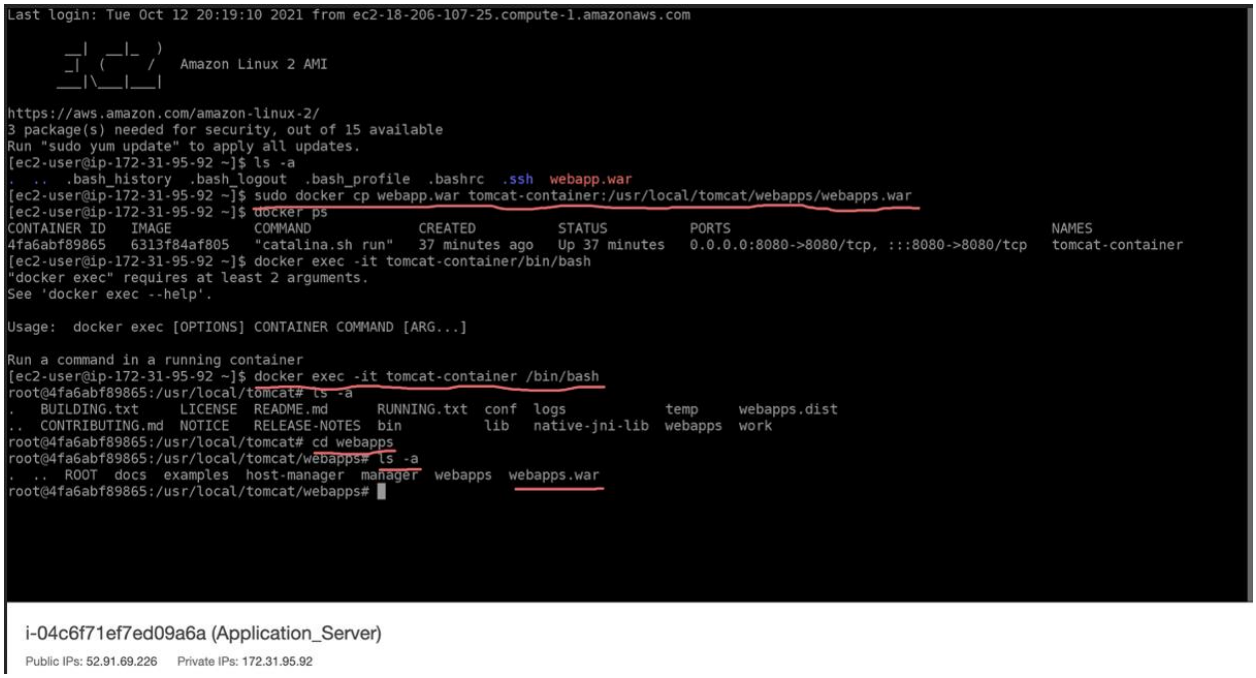
15. Created container from Tomcat server Image for the sake of deployment

```
Last login: Tue Oct 12 19:16:52 2021 from ec2-18-206-107-24.compute-1.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 15 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-95-92 ~]$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
tomcat       latest    6313f84af805   6 days ago   680MB
[ec2-user@ip-172-31-95-92 ~]$ docker run -d -p 8080:8080 --name tomcat-server 6313f84af805
6da0361f0c7cff2a44c91677c052d7b42ad12fd1d32d715e81ed8efa2786c5d2
[ec2-user@ip-172-31-95-92 ~]$ docer ps
-bash: docer: command not found
[ec2-user@ip-172-31-95-92 ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND             CREATED          STATUS          PORTS                                       NAMES
6da0361f0c7c   6313f84af805   "catalina.sh run"   10 seconds ago   Up 8 seconds    0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   tomcat-server
[ec2-user@ip-172-31-95-92 ~]$
```

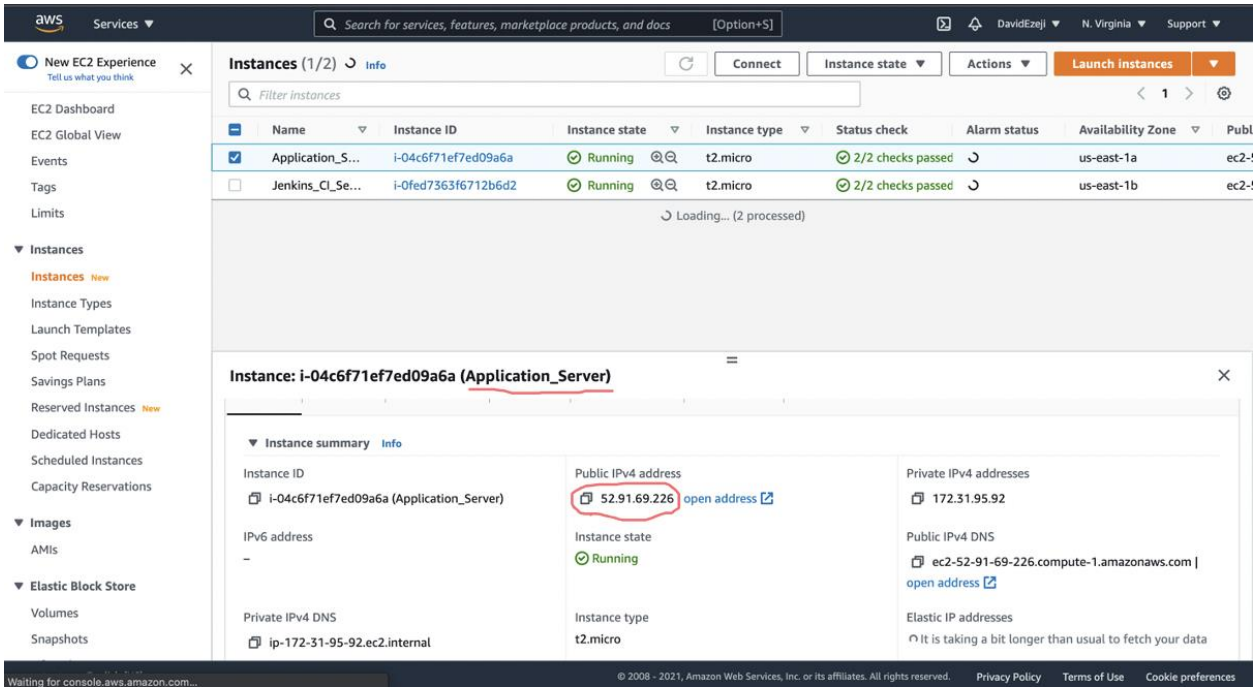
i-04c6f71ef7ed09a6a (Application_Server)

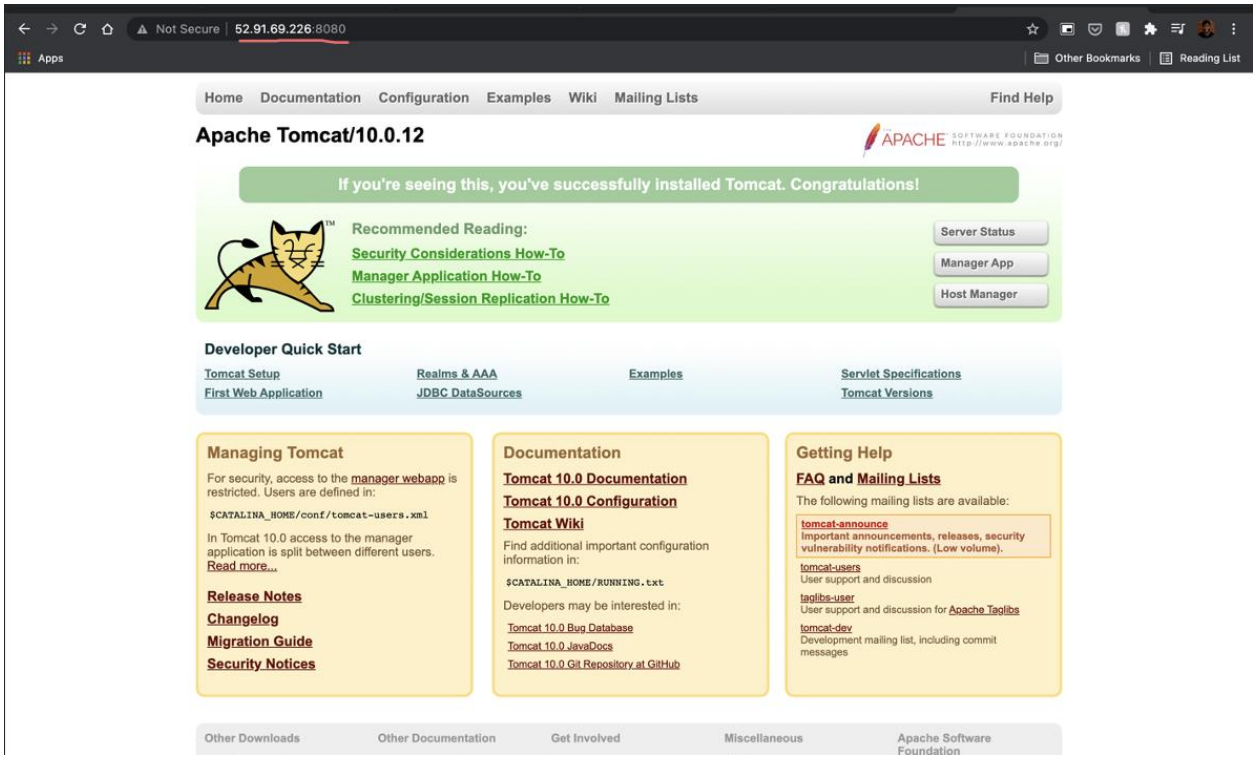
Public IPs: 52.91.69.226 Private IPs: 172.31.95.92

16. Moved the webapps.war file (this file exists on the Linux instance) into the webapps folder that is in the tomcat container.



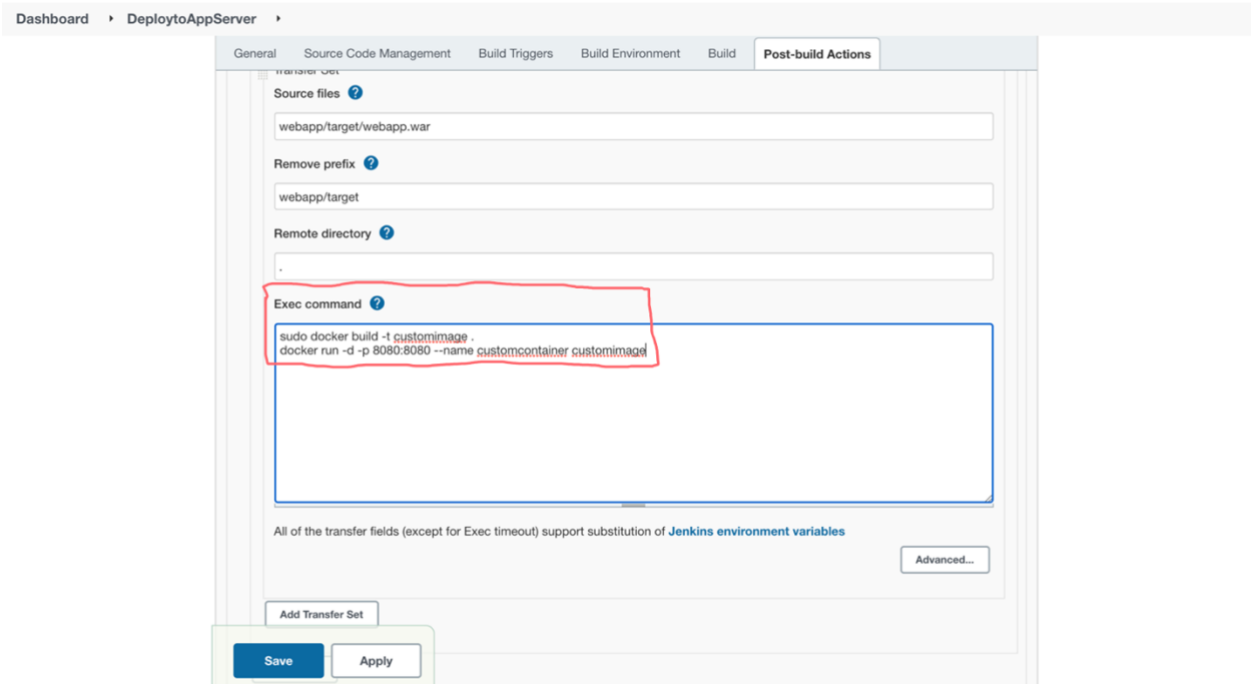
17. The web application is successfully deployed and readily available on the "Application_Server" instance

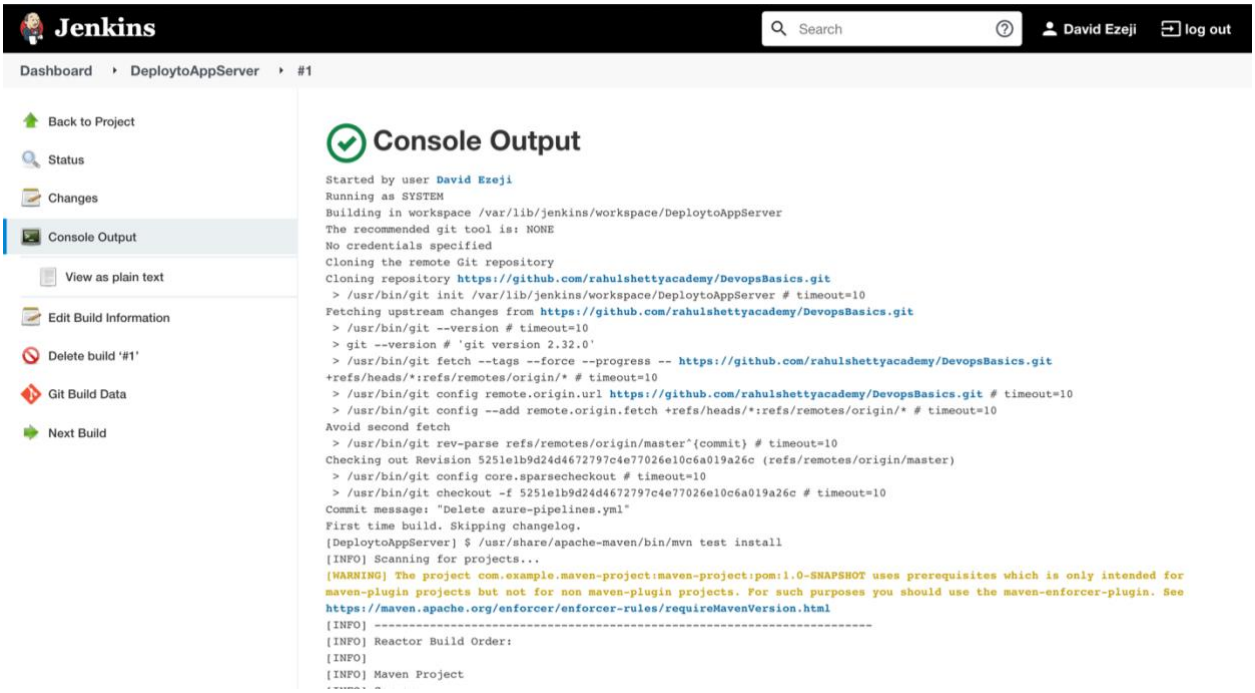




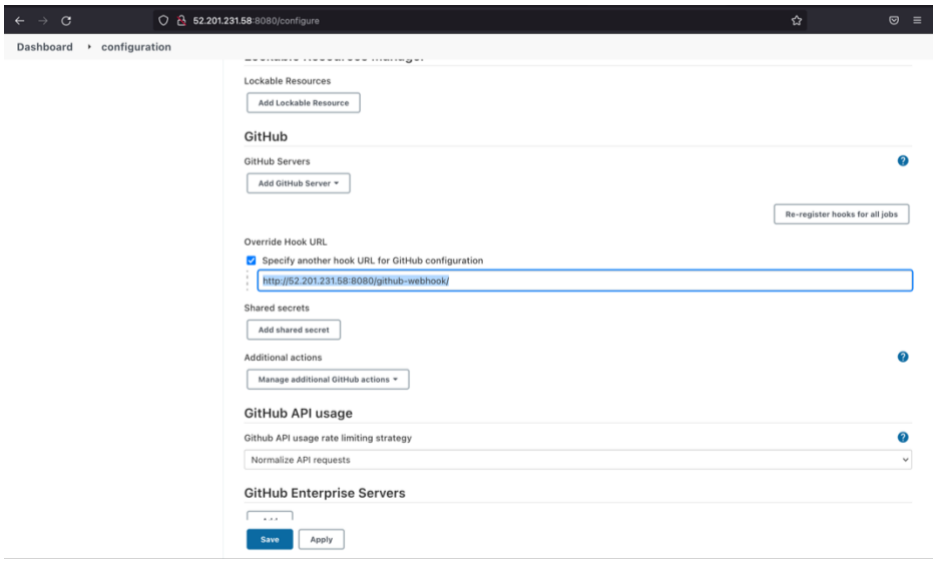
18. Created a Docker file that has instructions to deploy the webapp.war file to the Docker container

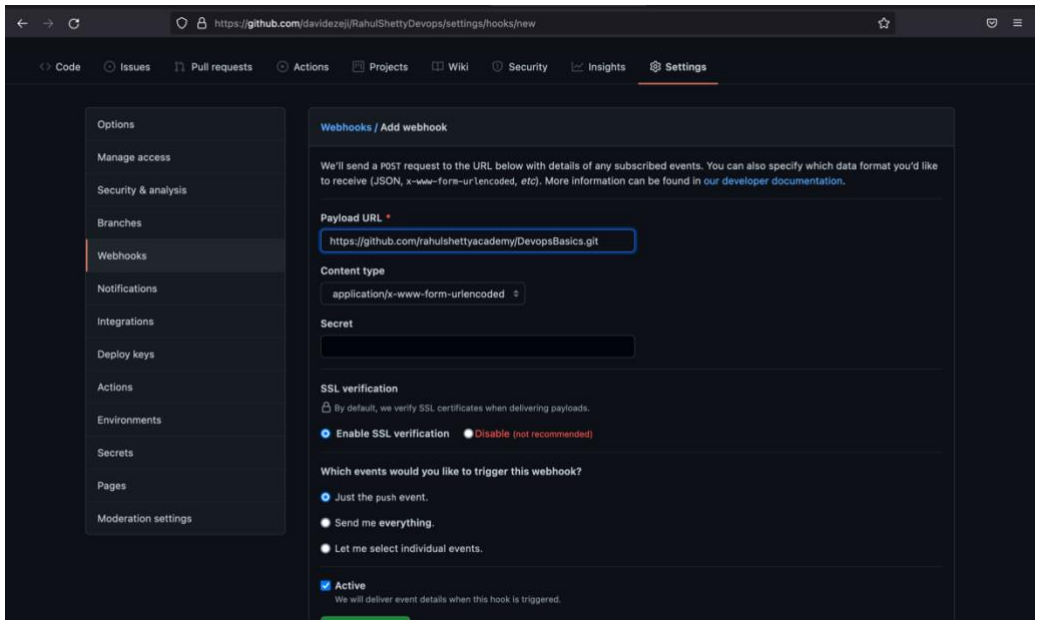
19. Used exec commands in Jenkins to automate the process of building a docker image and container from the dockerfile

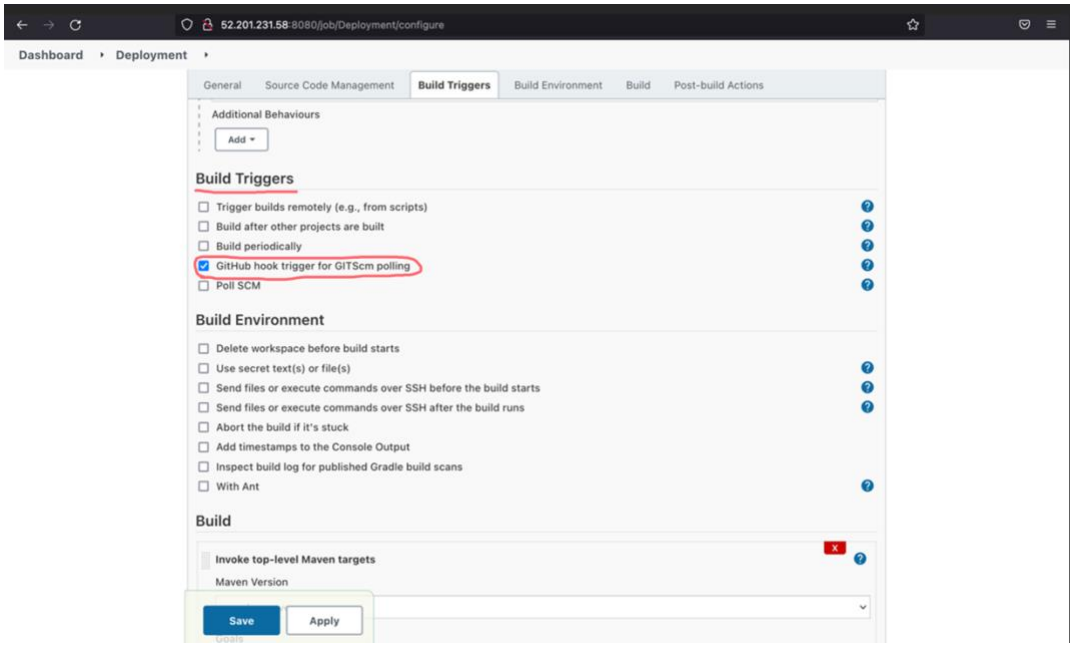




20. Attached Github hook from Jenkins to payload URL in my Github repository to allow for continuous integration to happen (allows for changes in code on Github to automatically trigger a build in Jenkins)


52.201.231.58:8080/configure
Dashboard
configuration
Lockable Resources
Add Lockable Resource
GitHub
GitHub Servers
Add GitHub Server
Re-register hooks for all jobs
Override Hook URL
Specify another hook URL for GitHub configuration
http://52.201.231.58:8080/github-webhook/
Shared secrets
Add shared secret
Additional actions
Manage additional GitHub actions
GitHub API usage
Github API usage rate limiting strategy
Normalize API requests
GitHub Enterprise Servers
Save
Apply


https://github.com/davidezeji/RahulShettyDevops/settings/hooks/new
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Options
Manage access
Security & analysis
Branches
Webhooks
Notifications
Integrations
Deploy keys
Actions
Environments
Secrets
Pages
Moderation settings
Webhooks / Add webhook
We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.
Payload URL *
https://github.com/rahulshettyacademy/DevopsBasics.git
Content type
application/x-www-form-urlencoded
Secret
SSL verification
By default, we verify SSL certificates when delivering payloads.
Enable SSL verification
Disable (not recommended)
Which events would you like to trigger this webhook?
Just the push event.
Send me everything.
Let me select individual events.
Active
We will deliver event details when this hook is triggered.


52.201.231.58:8080/job/Deployment/configure
Dashboard
Deployment
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Additional Behaviours
Add
Build Triggers
Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Build periodically
GitHub hook trigger for GITScm polling
Poll SCM
Build Environment
Delete workspace before build starts
Use secret text(s) or file(s)
Send files or execute commands over SSH before the build starts
Send files or execute commands over SSH after the build runs
Abort the build if it's stuck
Add timestamps to the Console Output
Inspect build log for published Gradle build scans
With Ant
Build
Invoke top-level Maven targets
Maven Version
Save
Apply