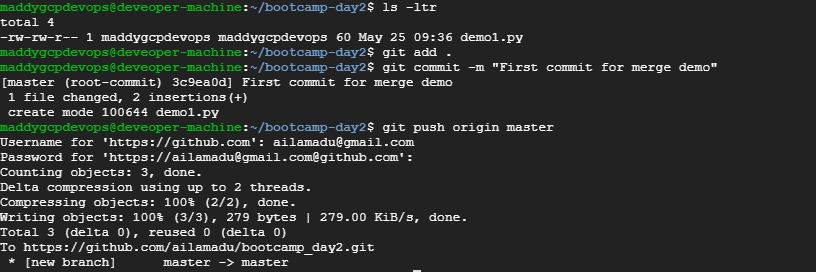
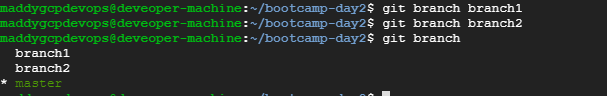
# **Git**

1. **Create two separate branches from master**

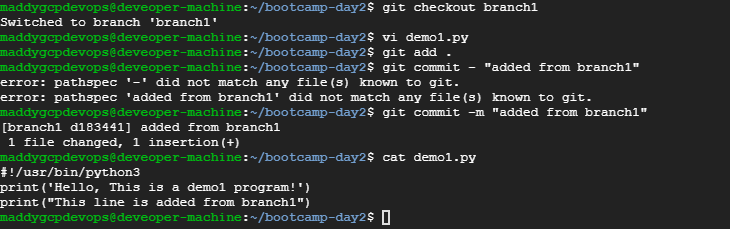
First I have created a repo and pushed a demo1.py file to it. Then I have created 2 branches namely **branch1 and branch2**



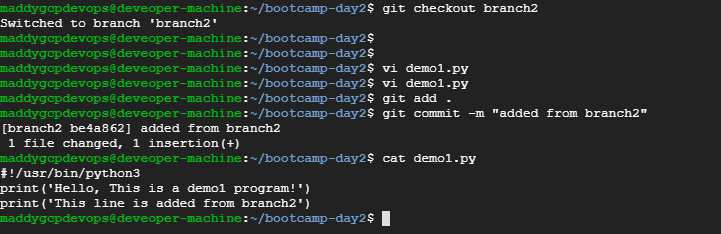


1. **Make changes in the same function of the source code in both the branches**

* Having the branches created previous step, I have checked out to branch1 and added a line **“print("This line is added from branch1")”**
* Added it to staging and committed it with message **“added from branch1”**

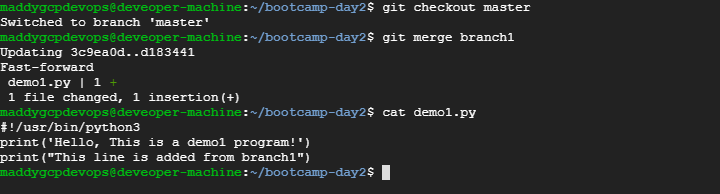


* I have now checked out to branch2 and added a line “**print("This line is added from branch2")”**
* Added it to staging and committed it with message **“added from branch2”**



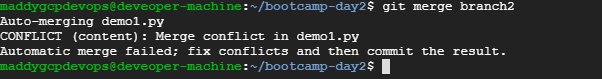
1. **Merge branch1 into the master**

To merge the branch1 to master , I have first checkedout to master and issued the **“git merge branch1”.** It has successfully merged to master.



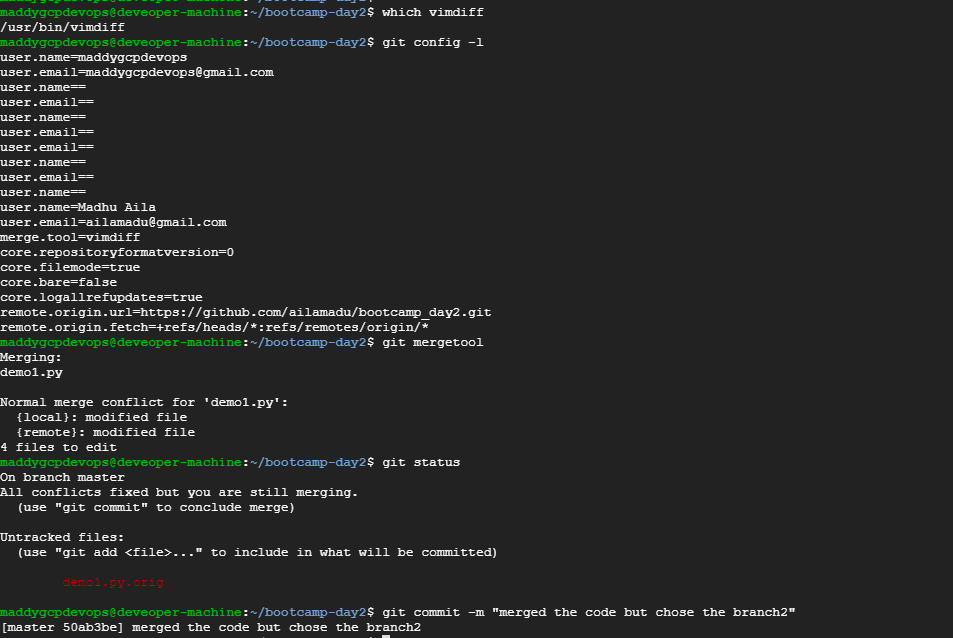
1. **Try and merge branch2 into the master(merge conflict should arise)**

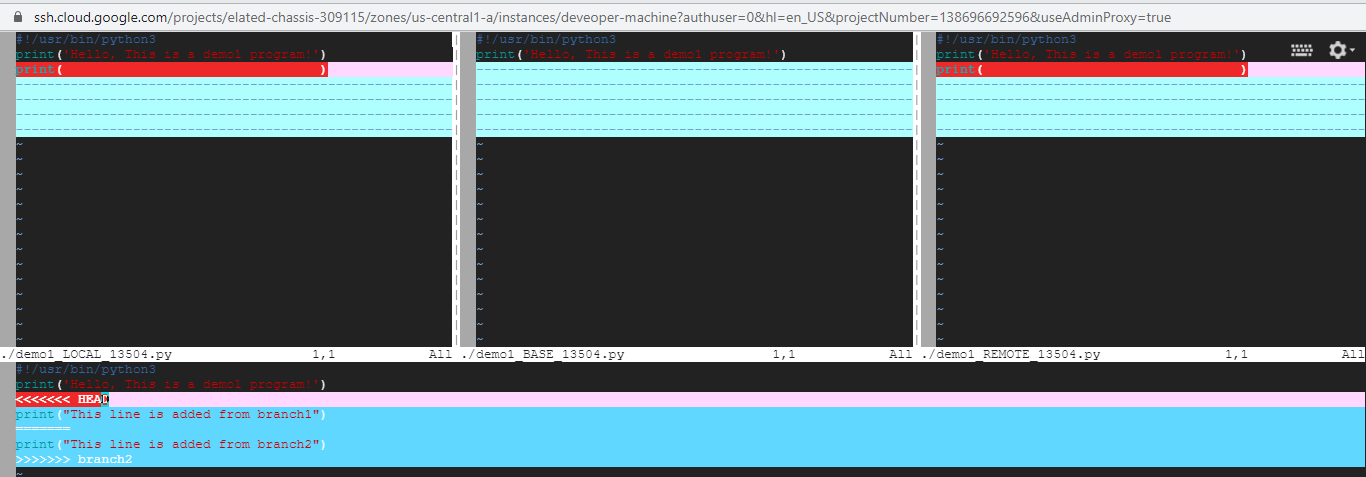
To merge the branch2 to master, I have issued the **“git merge branch2”** . As expected, I have got the conflict.



1. **Install a merge tool of your choice and resolve the merge conflict using git merge tool command**

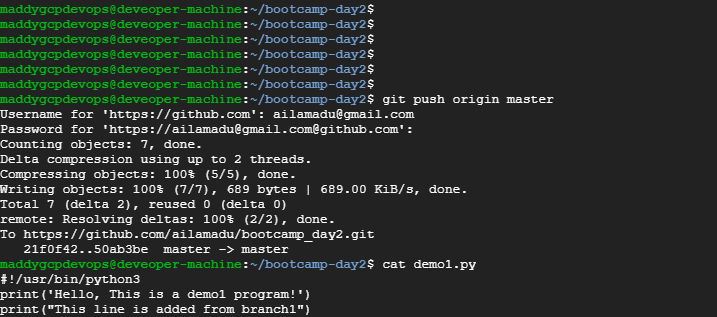
* I have used the vimdiff tool instead of meld . Configured the vimdiff as merge tool in the config.

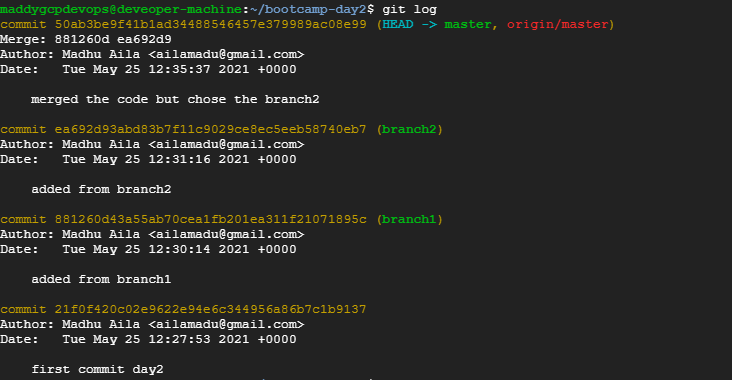




* I have issued “git mergetool” , I have opened the below window. I have done the required changes here and closed this window. When I check the git status, I shows that conflict has been fixed and to commit it. So I have issued commit command with message “merged the code but chose the branch2” (but actually it is branch1 , Please excuse 😊)

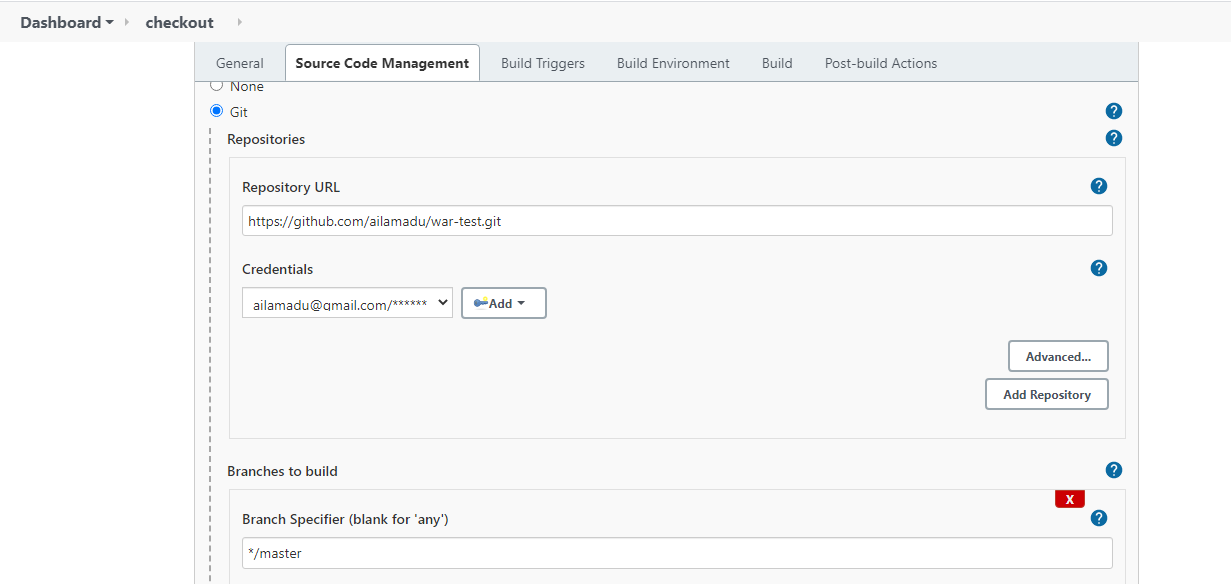
As all the said actions are performed, I have pushed the code to remote.

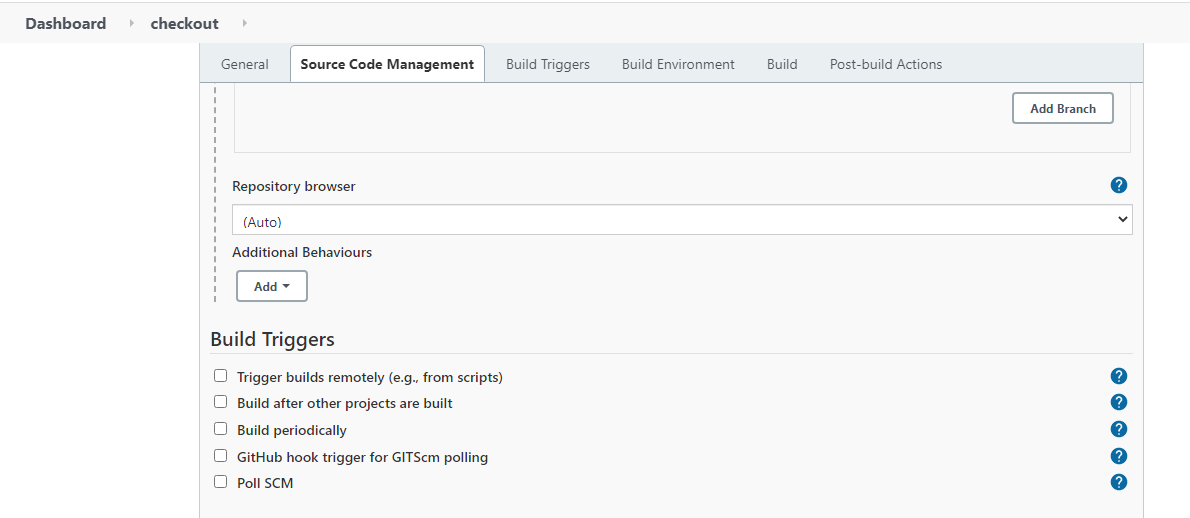
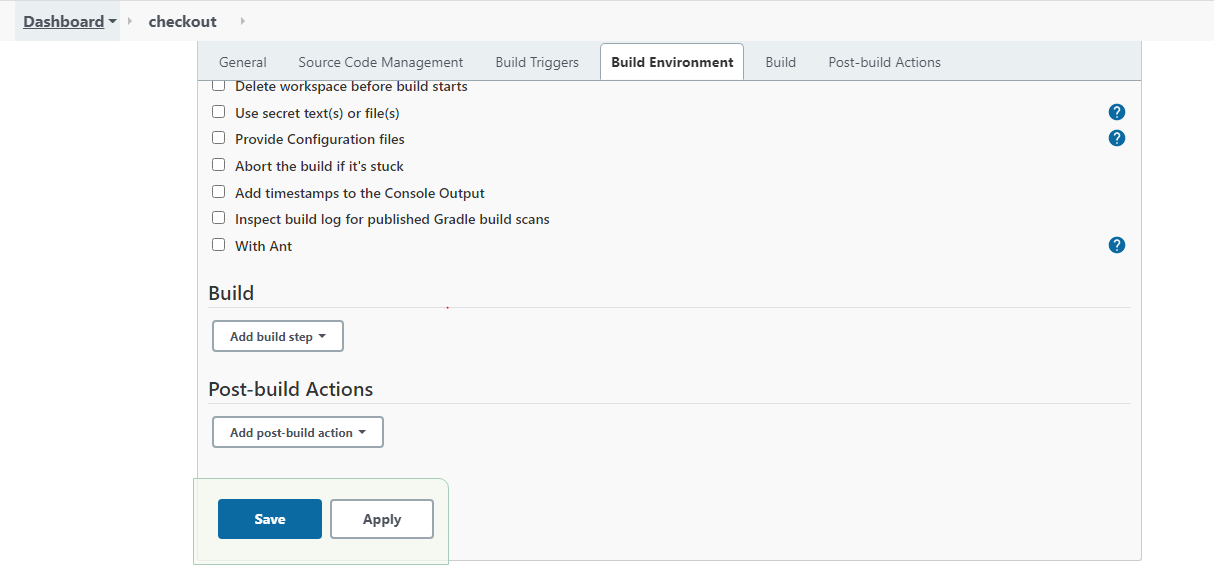




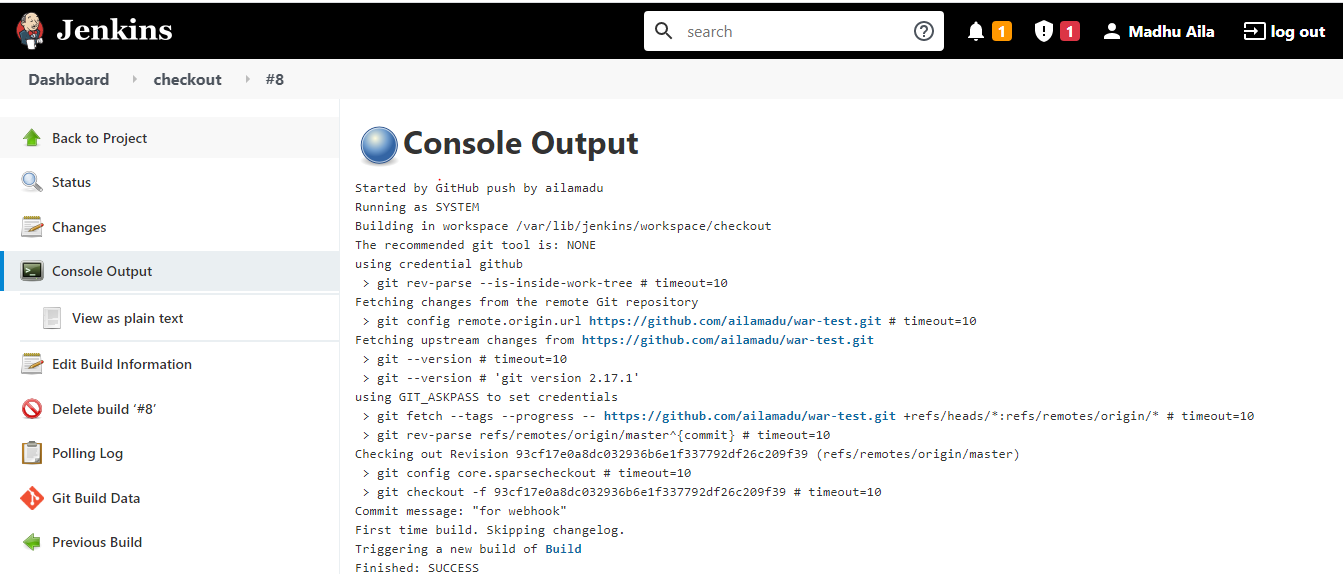
# **Jenkins**

* **Create multiple freestyle project in Jenkins to checkout, build, test, package and deploy the web application in a tomcat9 server.** 
  1. **Establish a build pipeline view with appropriate plugin installation.**
  2. **Configure maven as a tool in Jenkins environment.**
  3. **Attach the screenshot of the console output of each successful execution of the job**
* **I have created Checkout project like below**

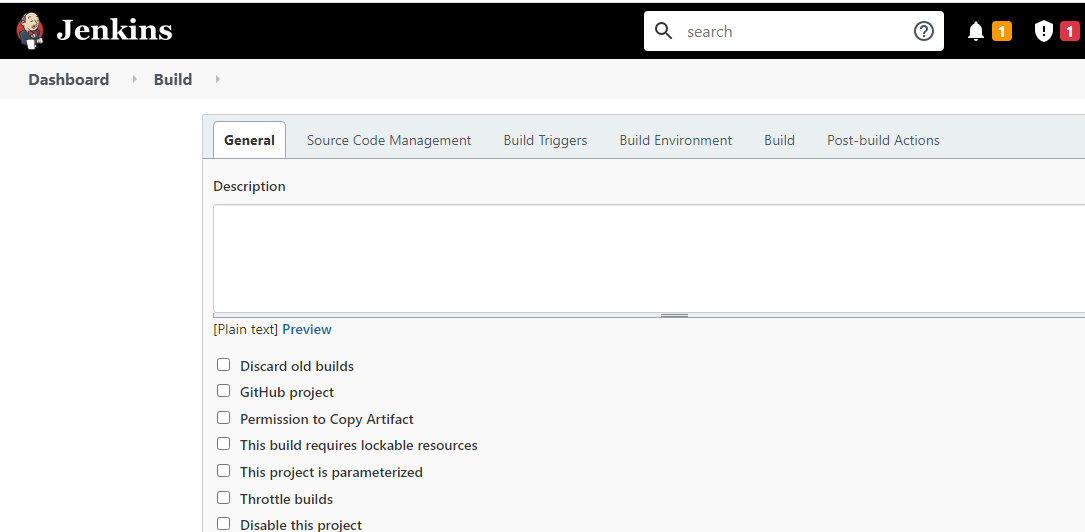


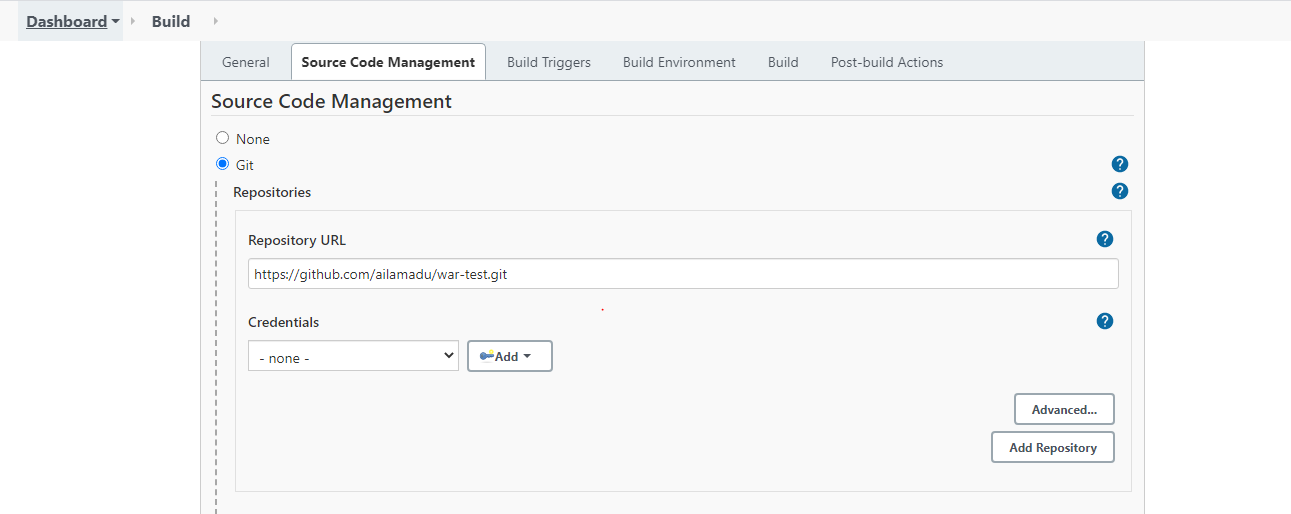
 

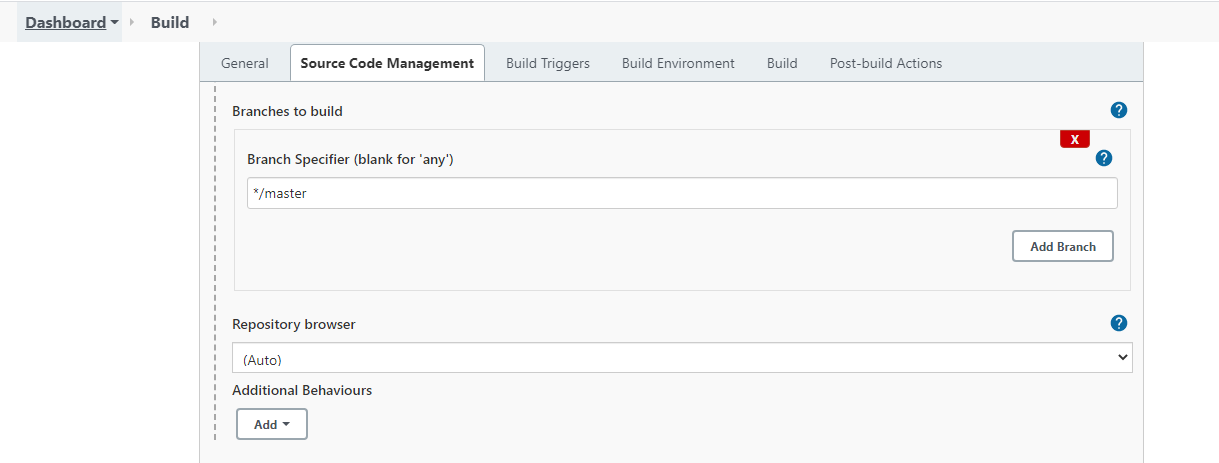
* Below is the checkout build output:



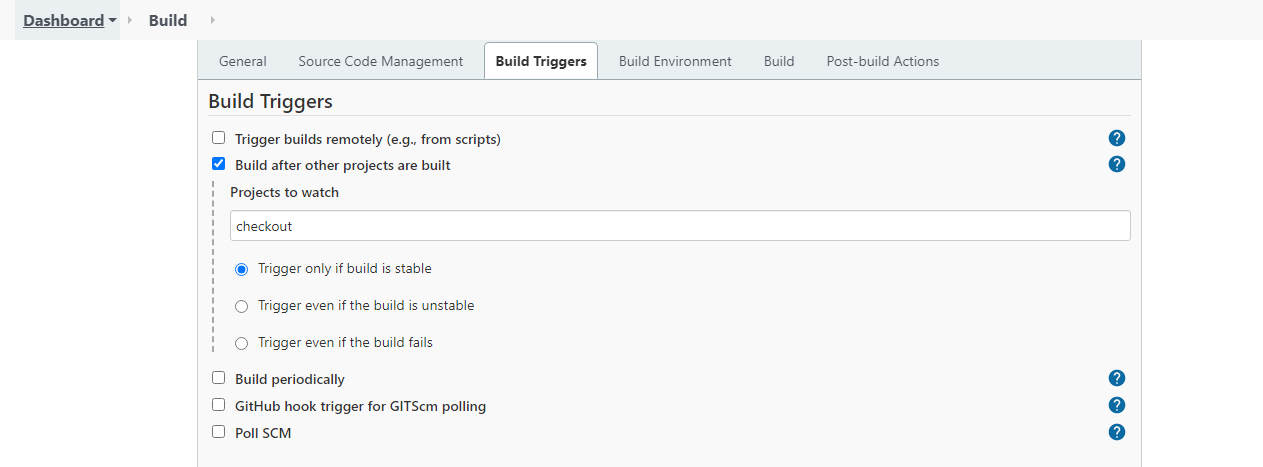
* **Build job is configured as below: It was created such a way that it will get trigger automatically when the checkout job get completed.**

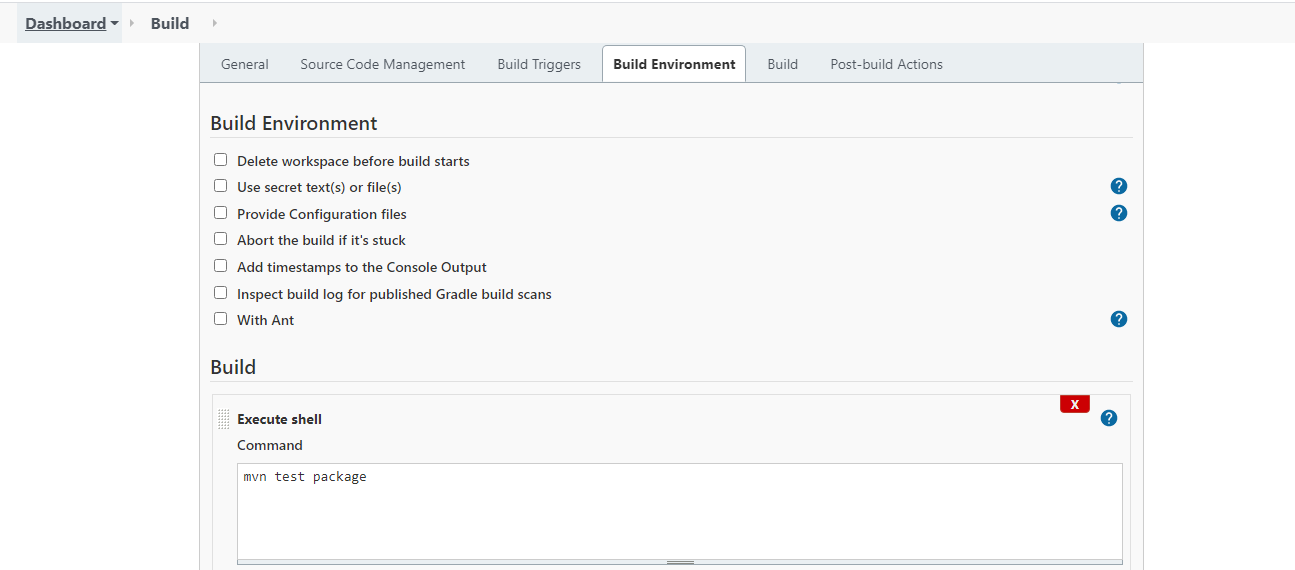


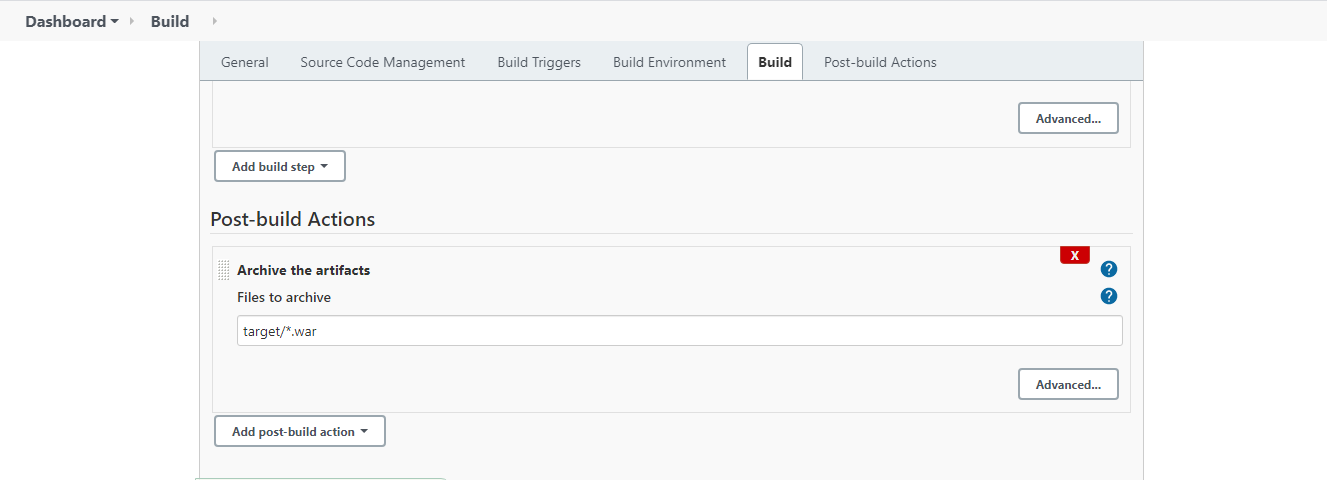




* **Below is the output of Build job:**

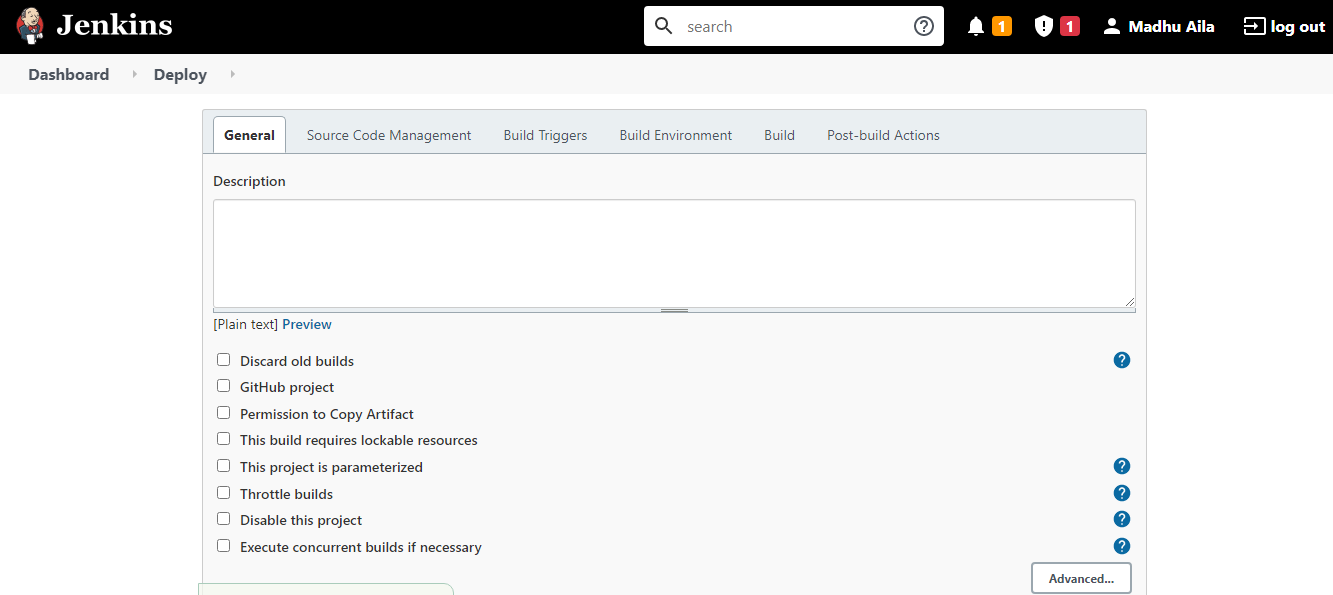


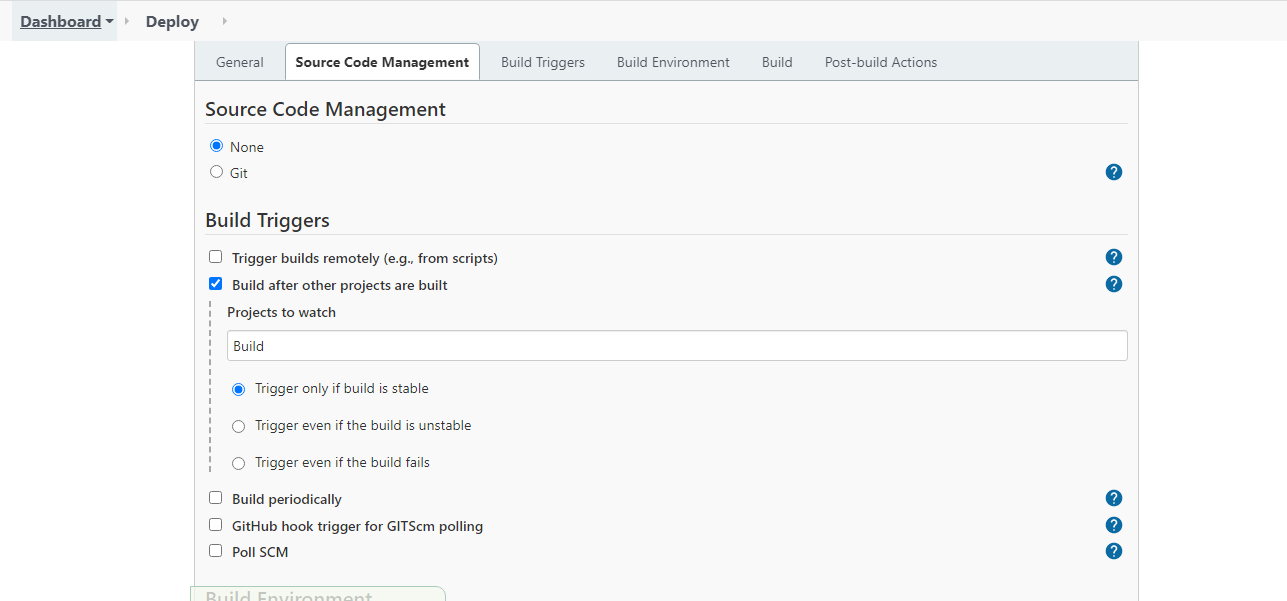


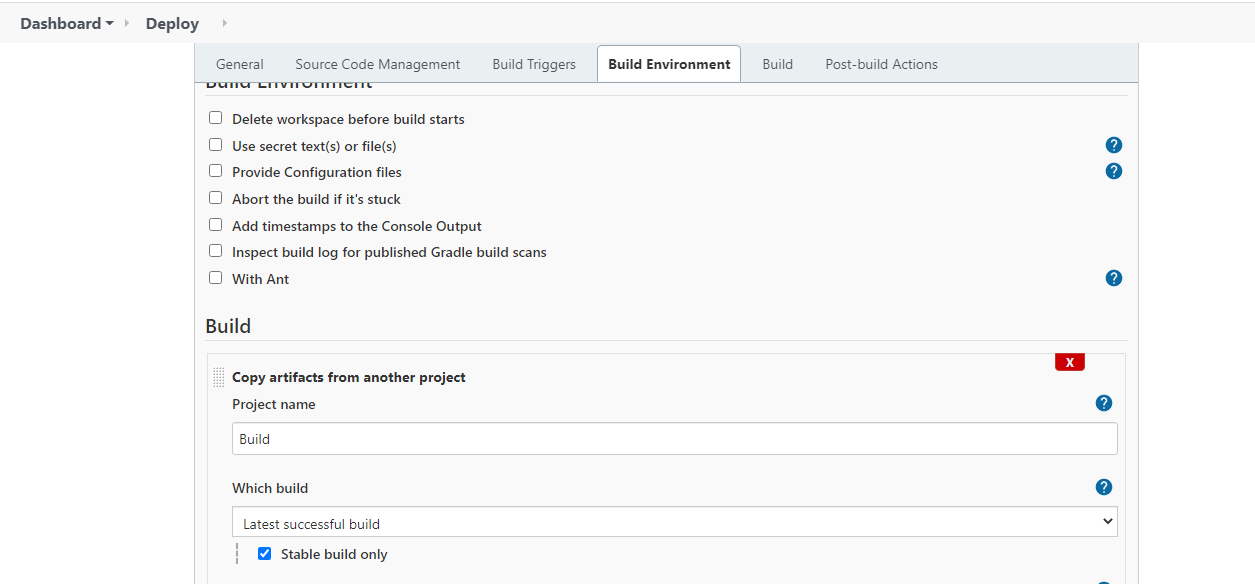


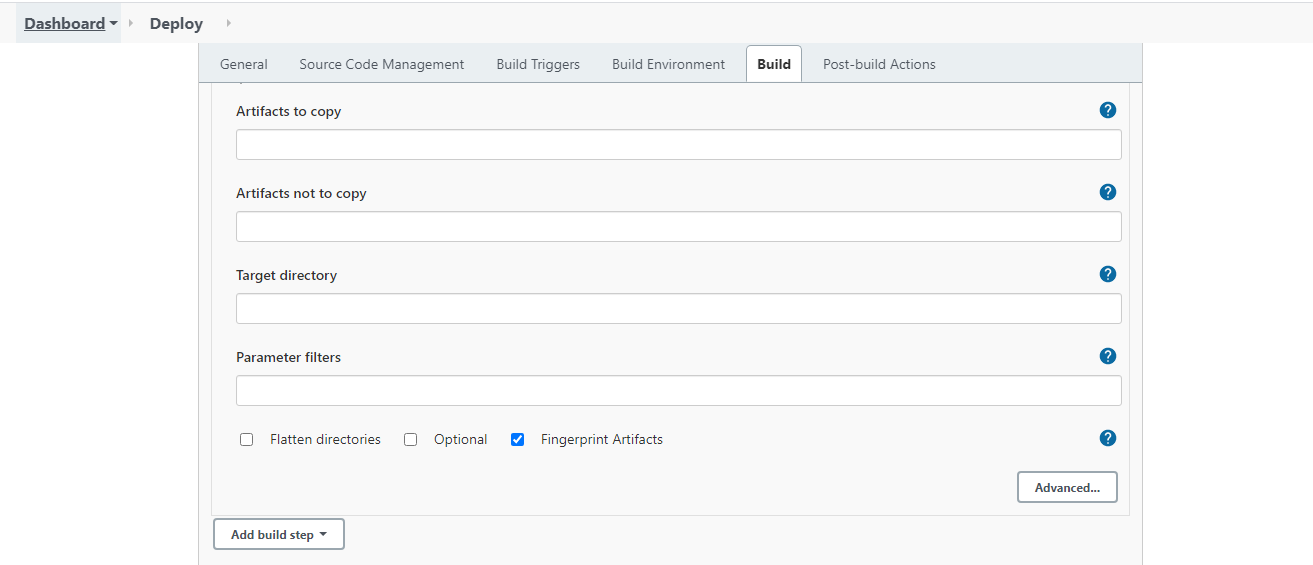


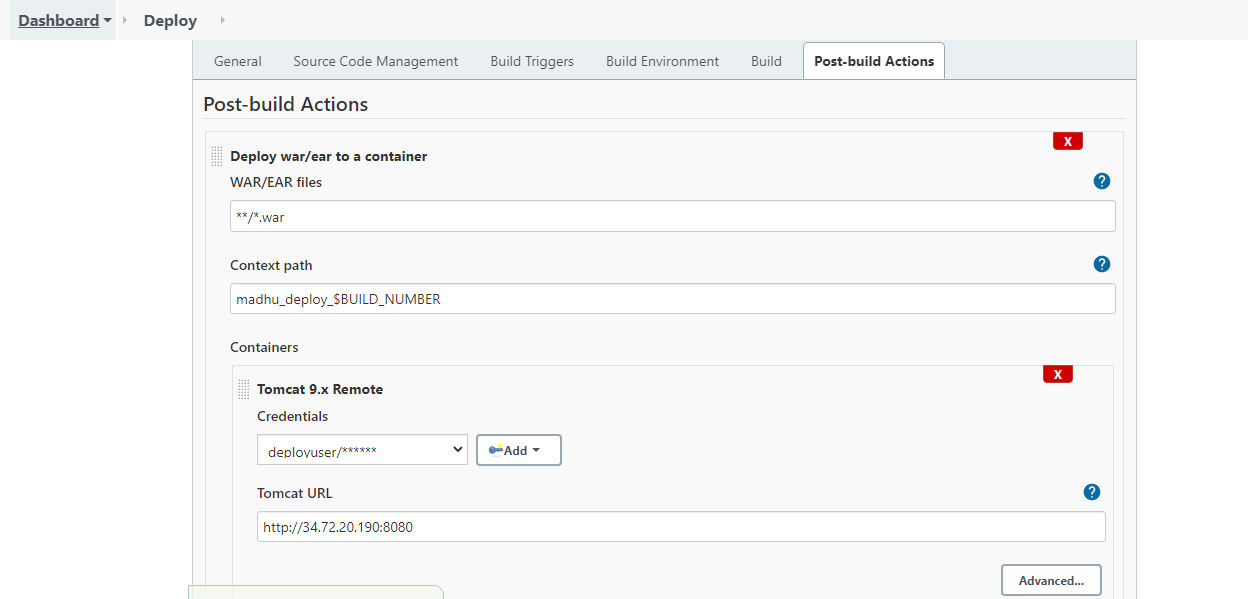
* **I have configured the deploy job such a way that it triggers automatically after the build job get completed.**

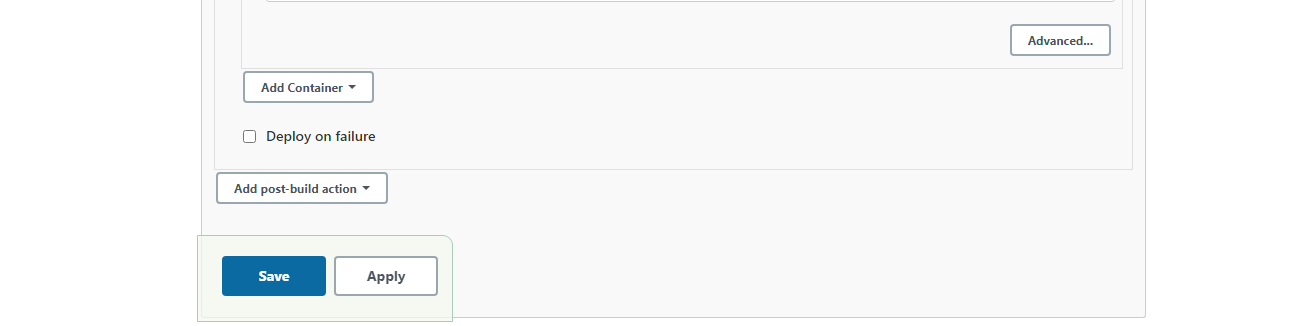




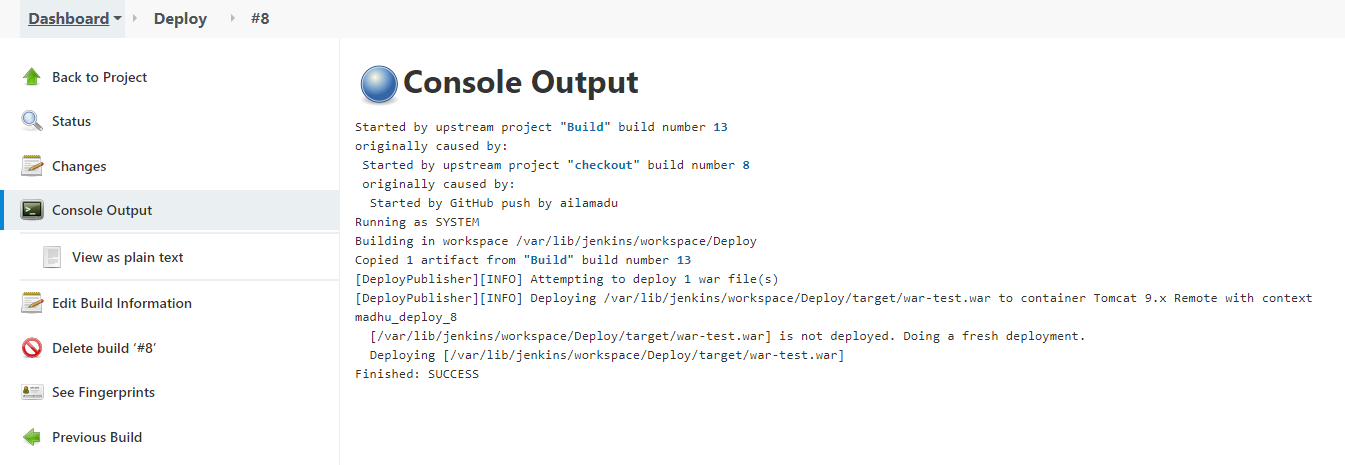




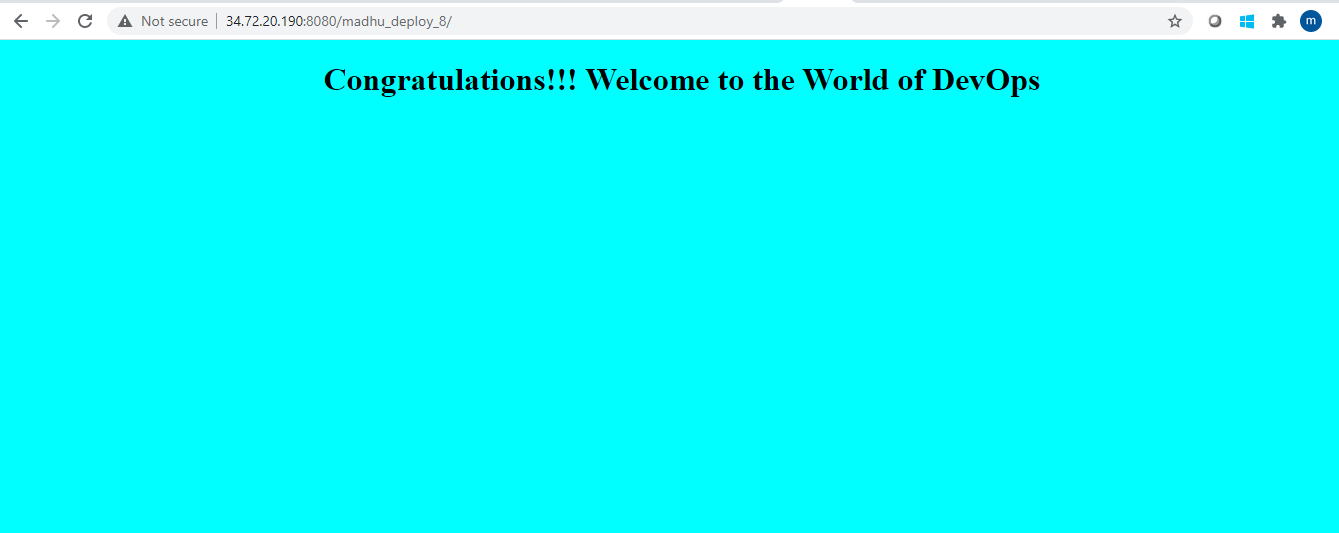




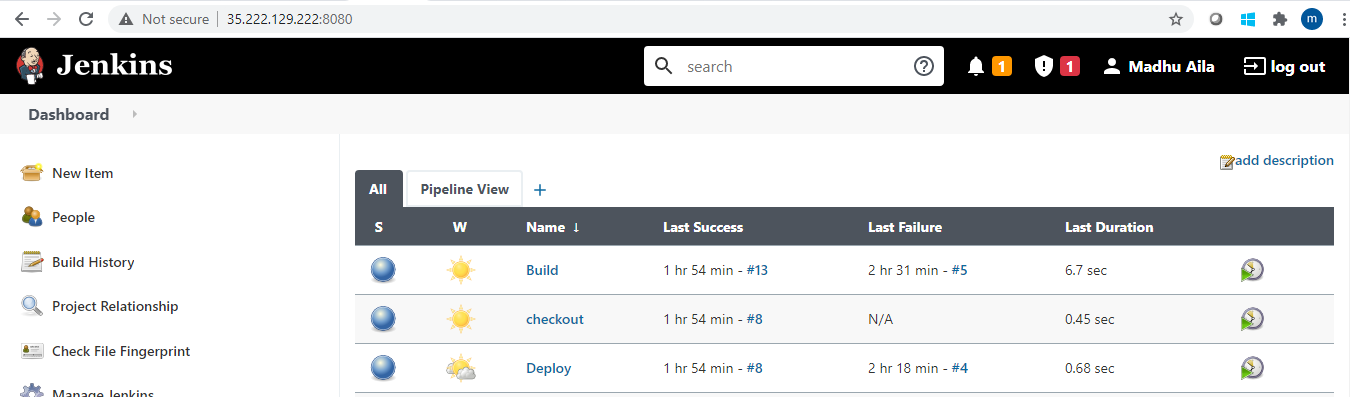
* Deploy job output as follows



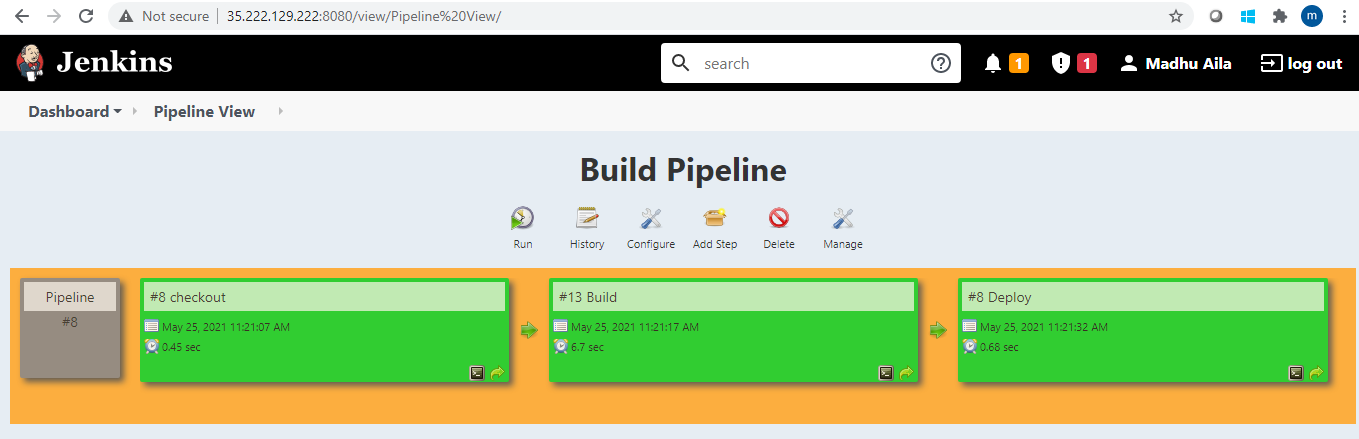
* Deployed site is like below:



* Below is the jobs configured



* Below is the pipeline view configured

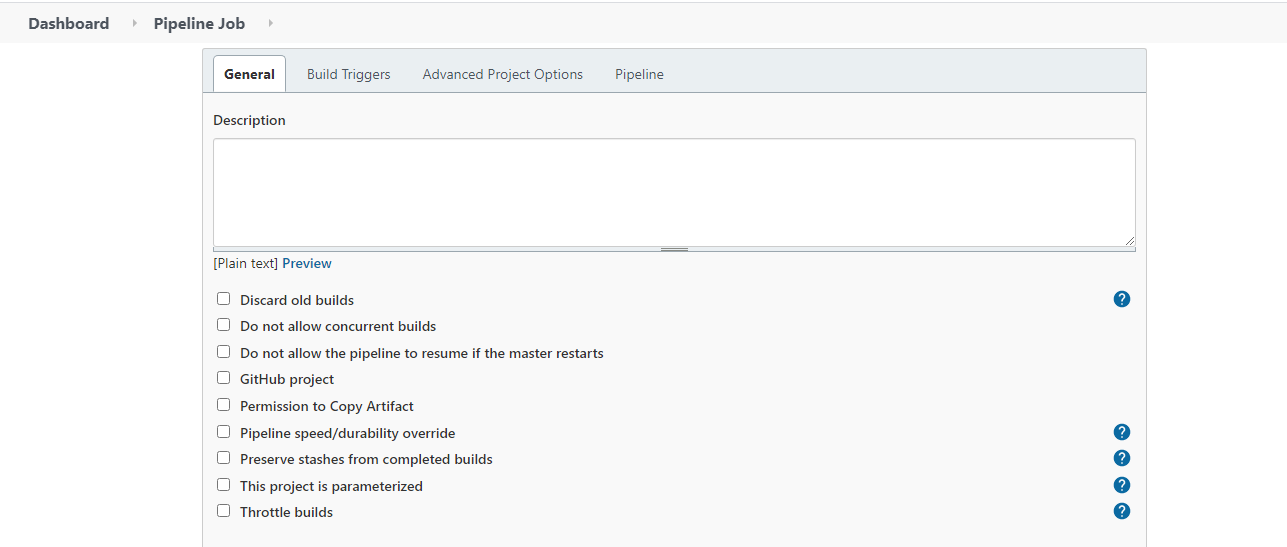


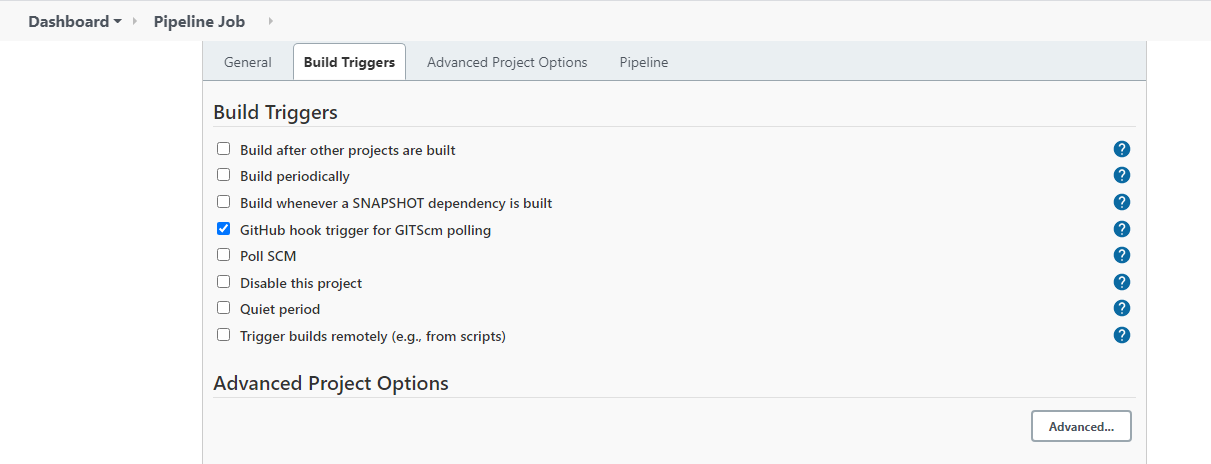
* I have installed the maven plugin before starting the exercise but could not capture the installation logs. Please execute but find the installed plugin.

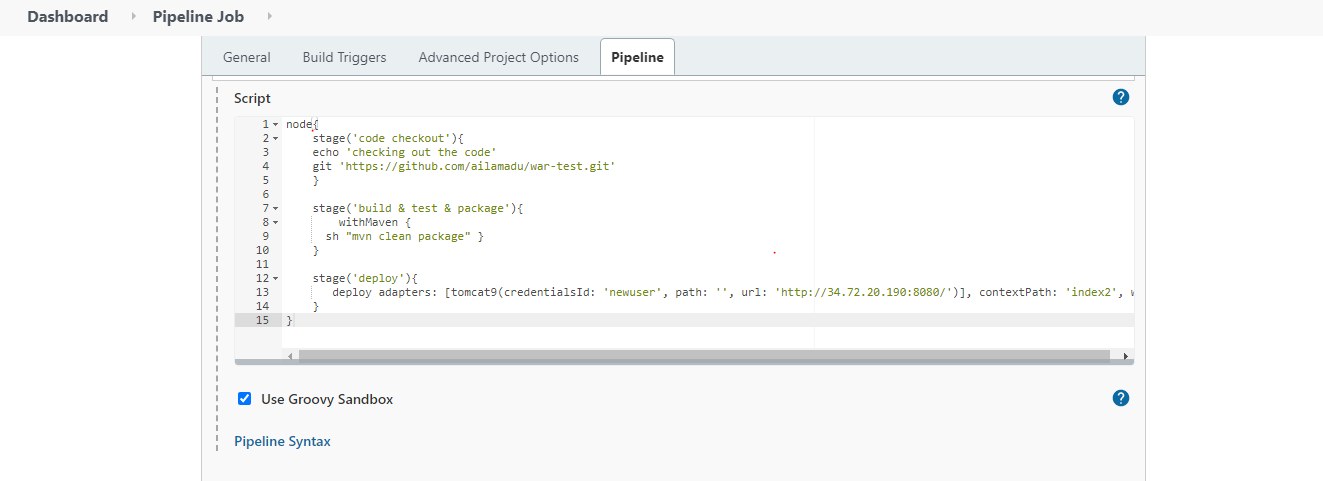


1. **Create a pipeline job and write the script to achieve the same tasks. Pipeline must be triggered with a GitHub webhook upon every commit**

I have created the pipeline job like below and enabled it to trigger for each commit

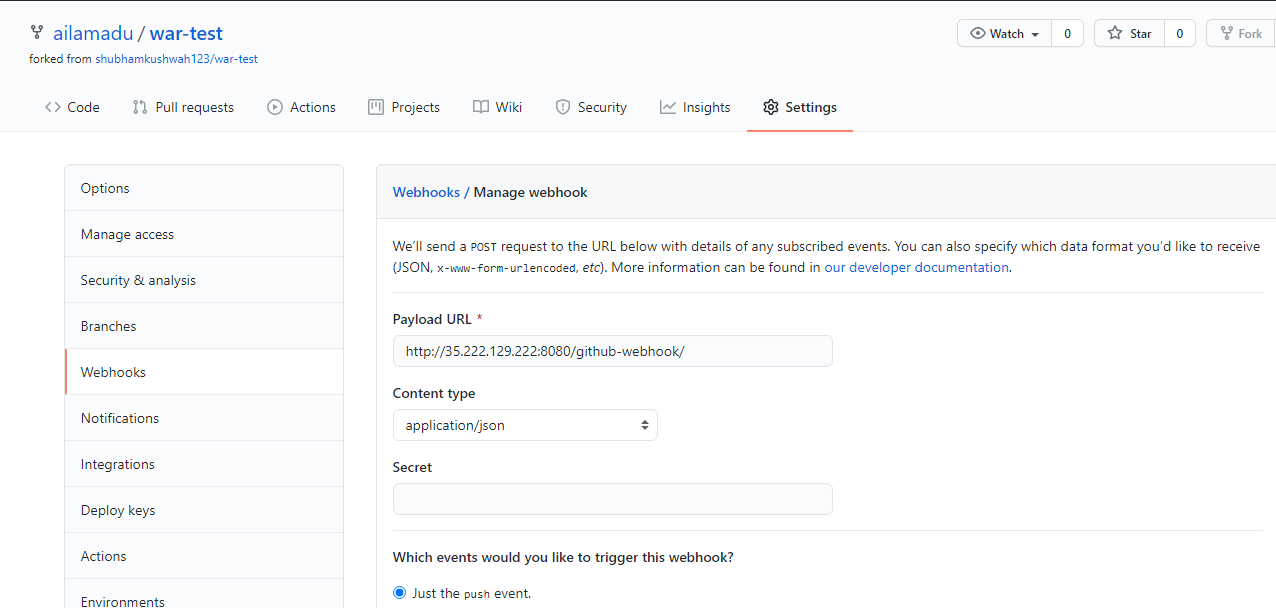




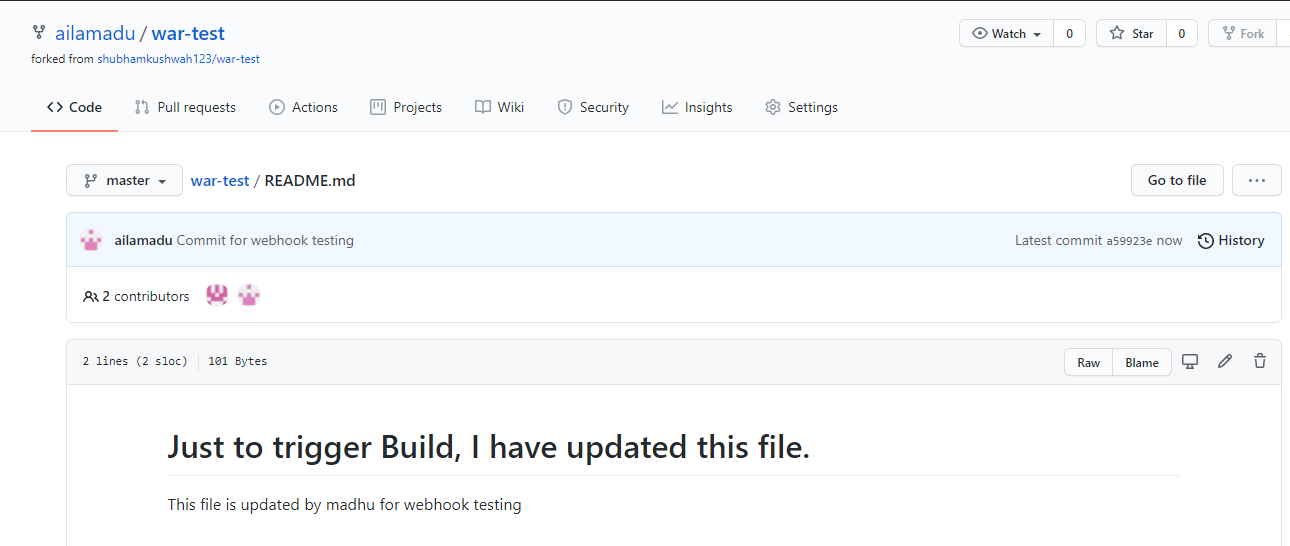


**Here I have used withmaven instead of shell itself as I have faced some issue and I thought that withmaven will help . Yes, it has helped to fix the error that I have encountered.**

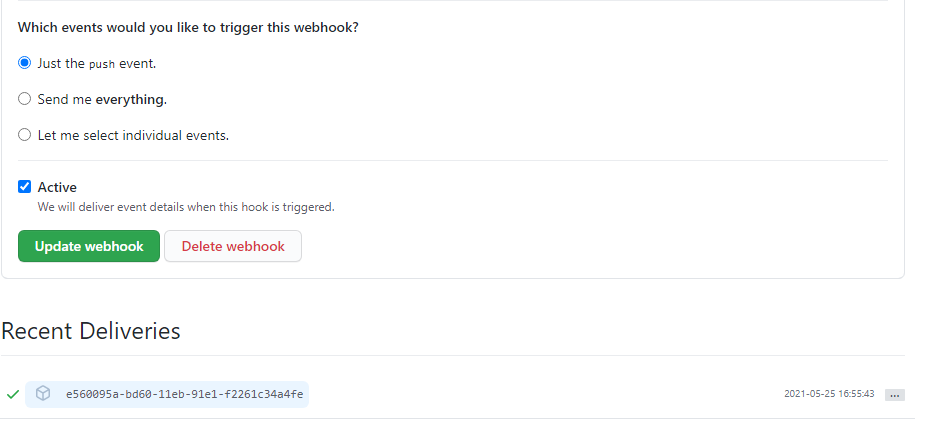
* **I have configured the webhook in the github like below:**



* **I have committed the repo.**



* **Webhook has delivered the commit to payload.**



* **It has triggered the Jenkins jobs** 