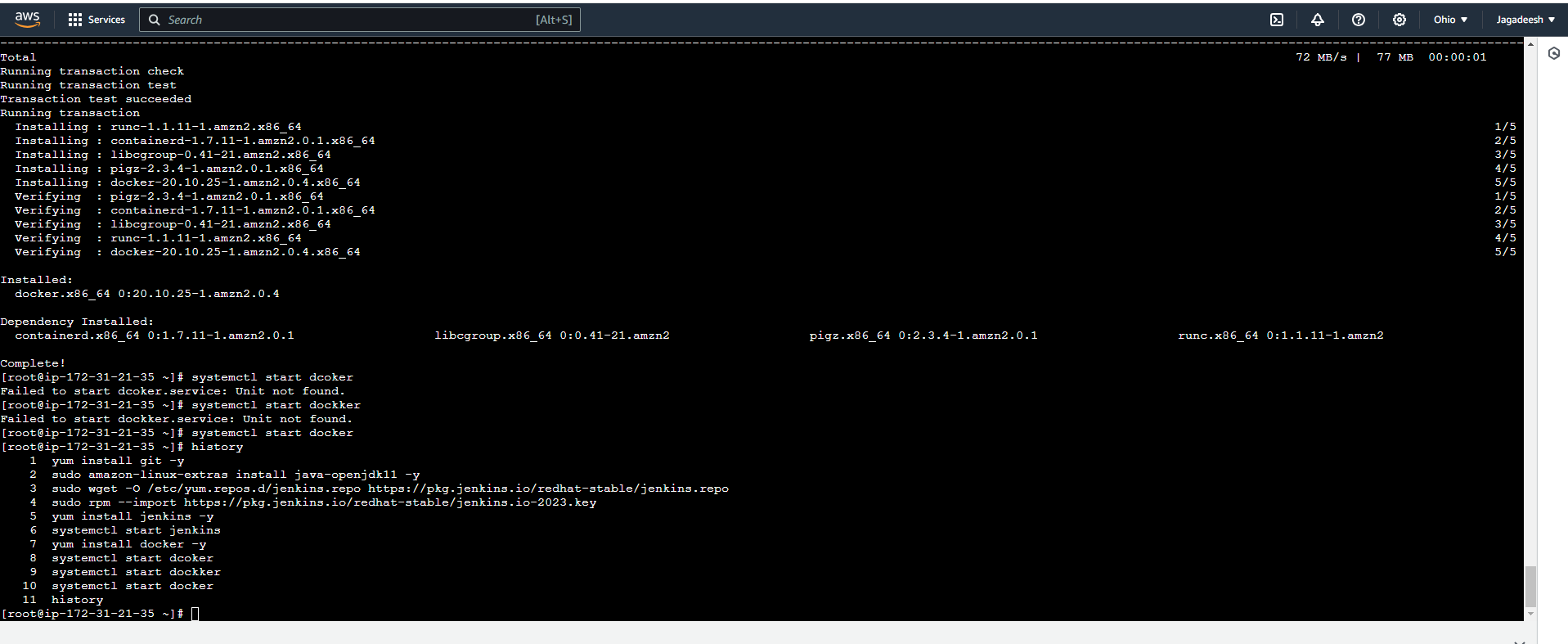
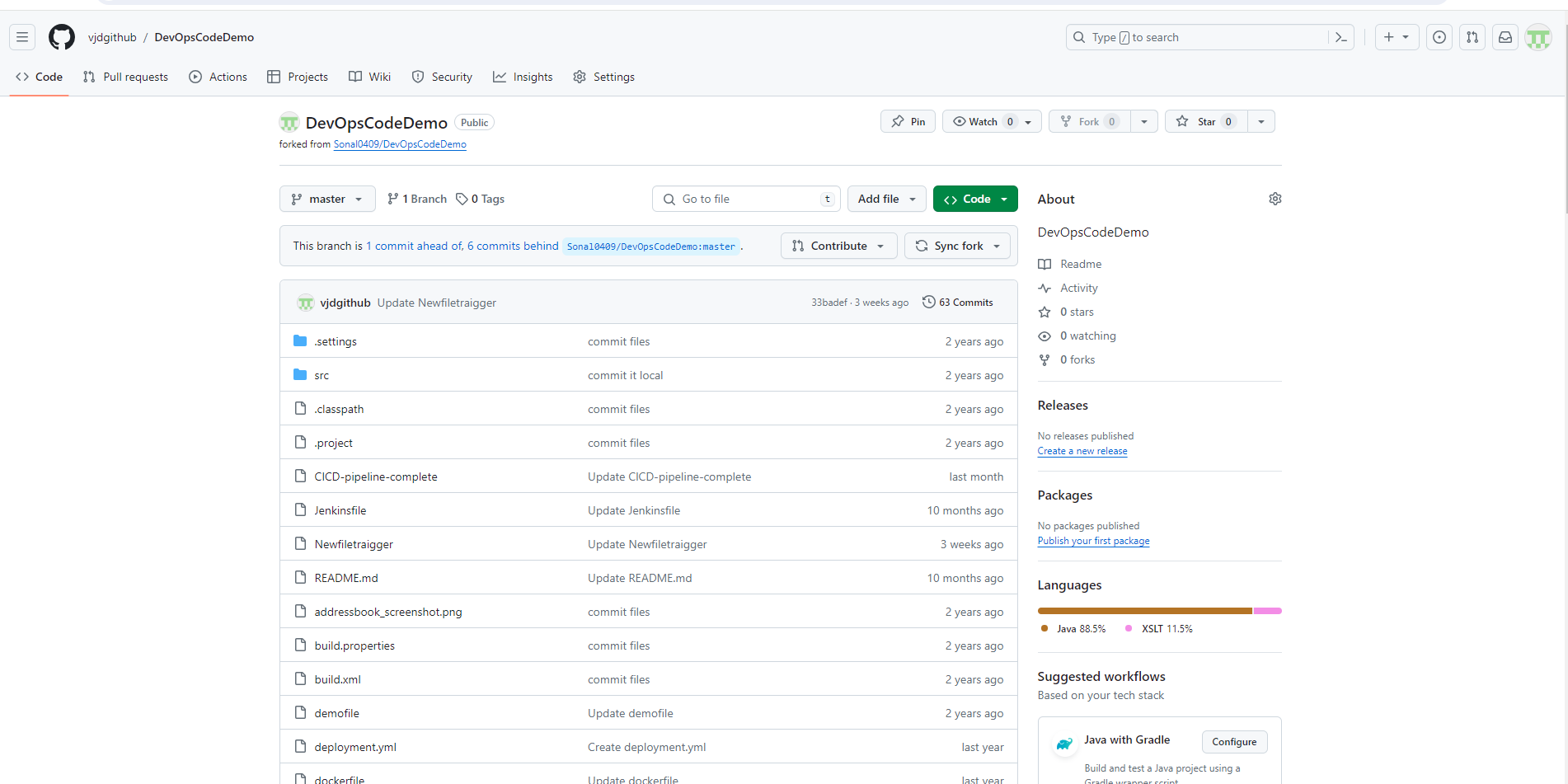
Steps for the Course end Project 1:

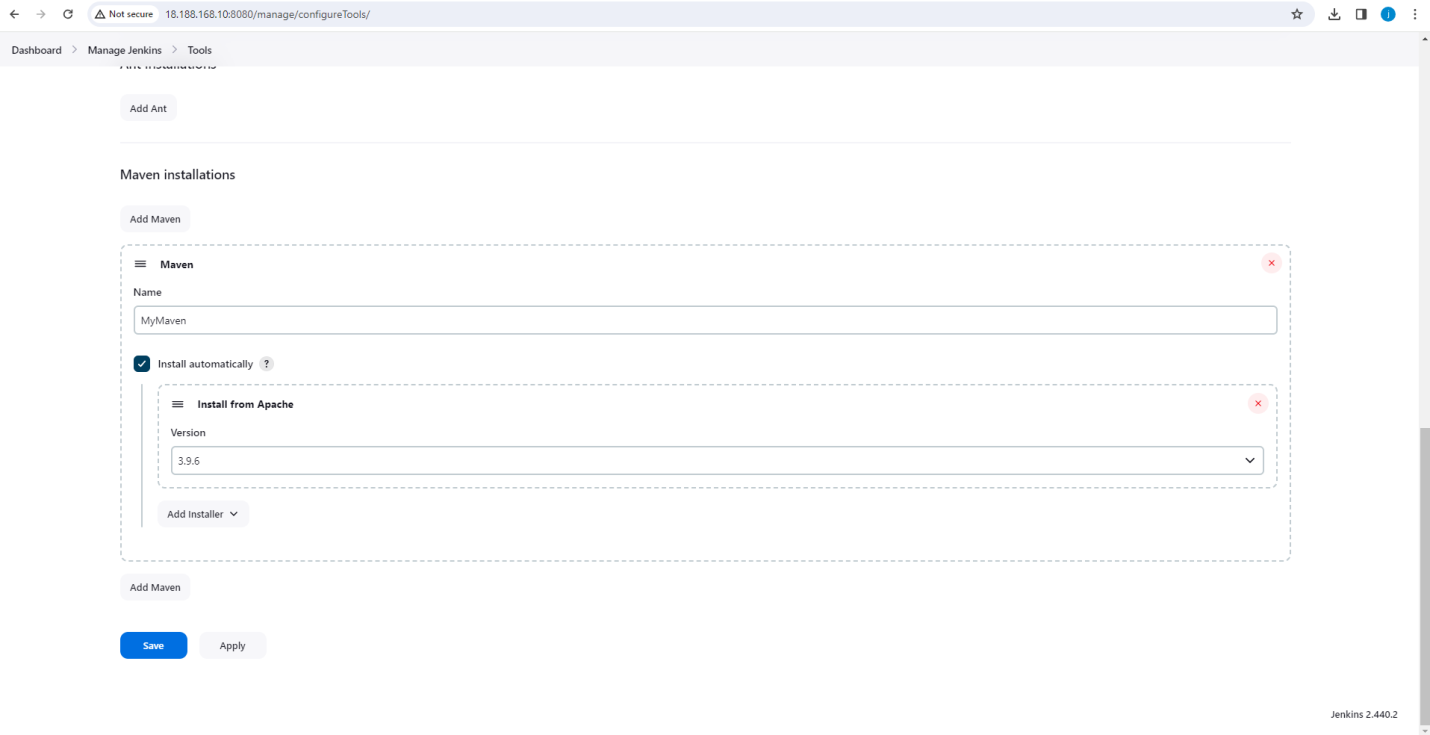
1. Host machine successfully installed with Git, Java, Jenkins and Docker.



b) Screenshot of code repository in GIThub

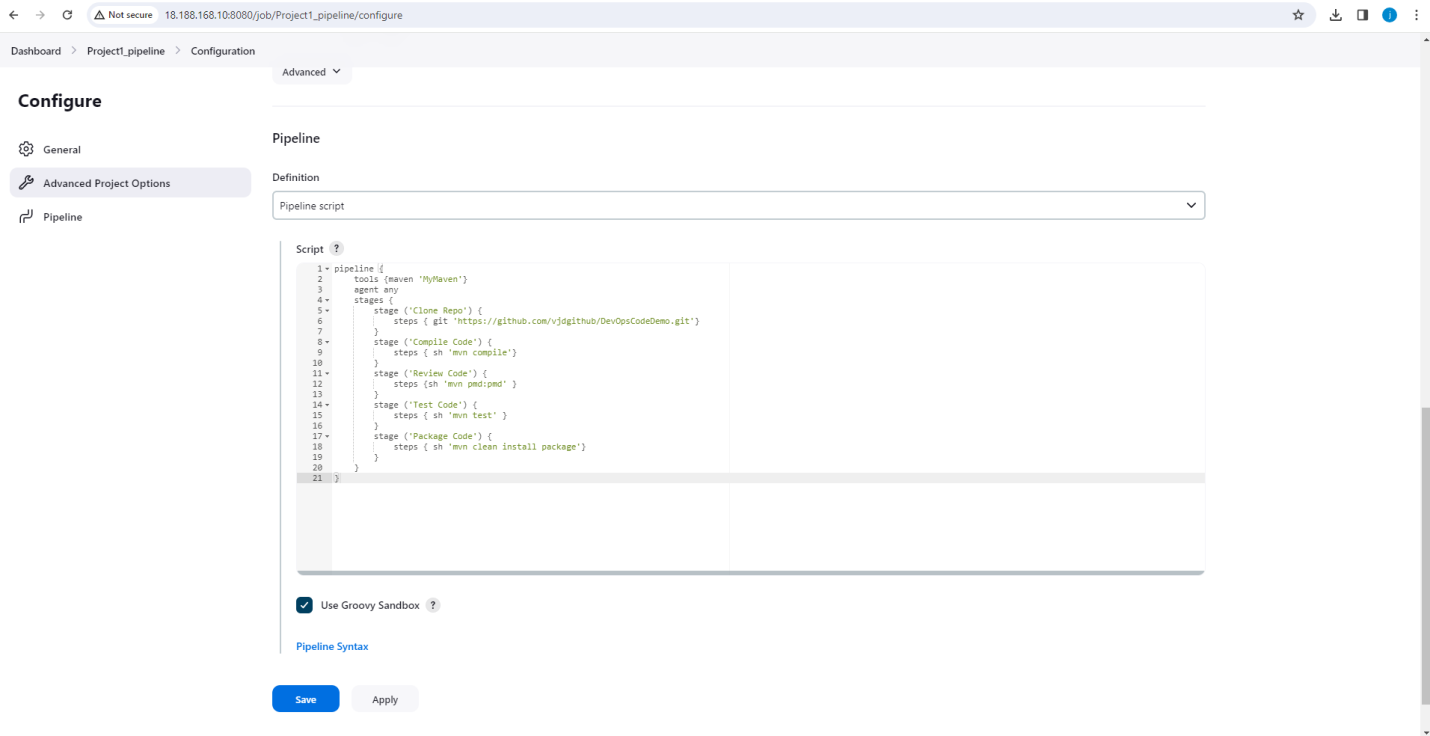


Maven tool configuration:

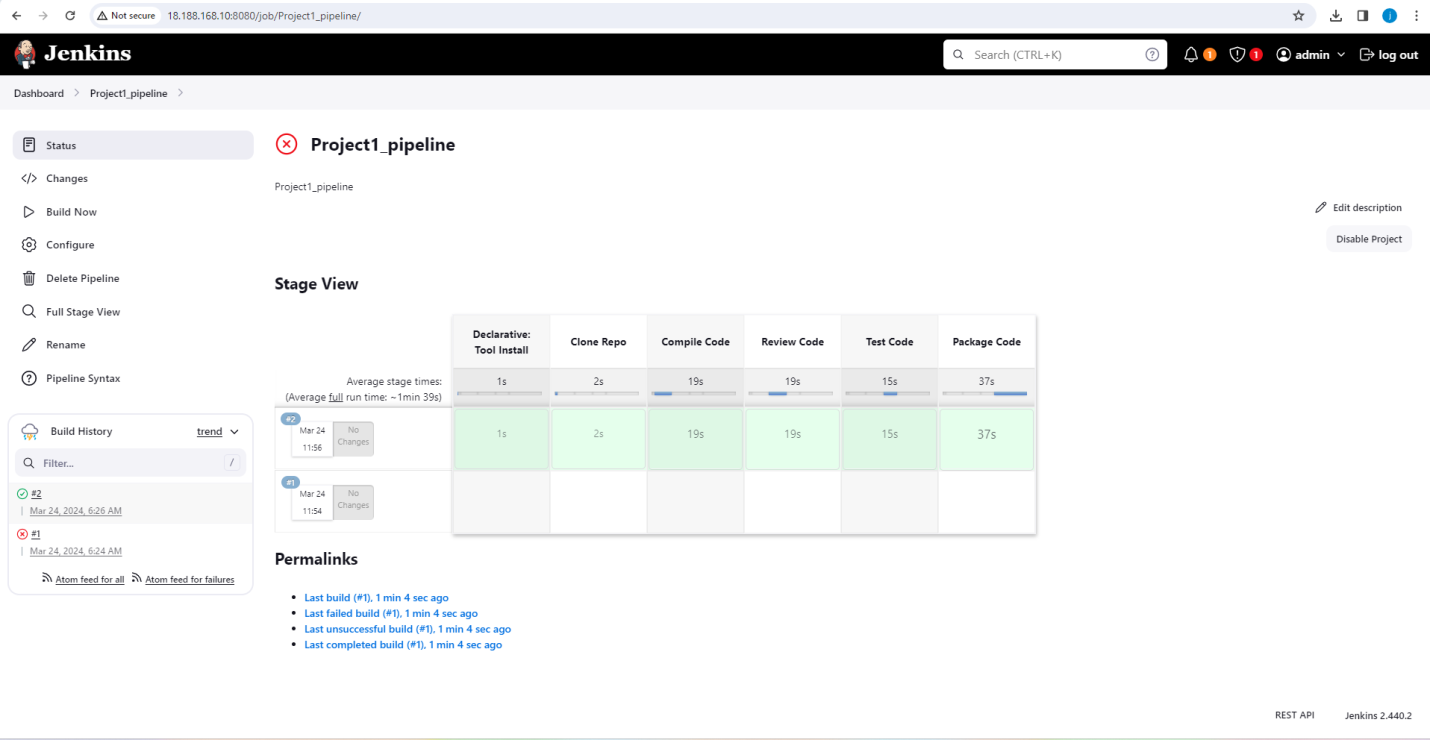


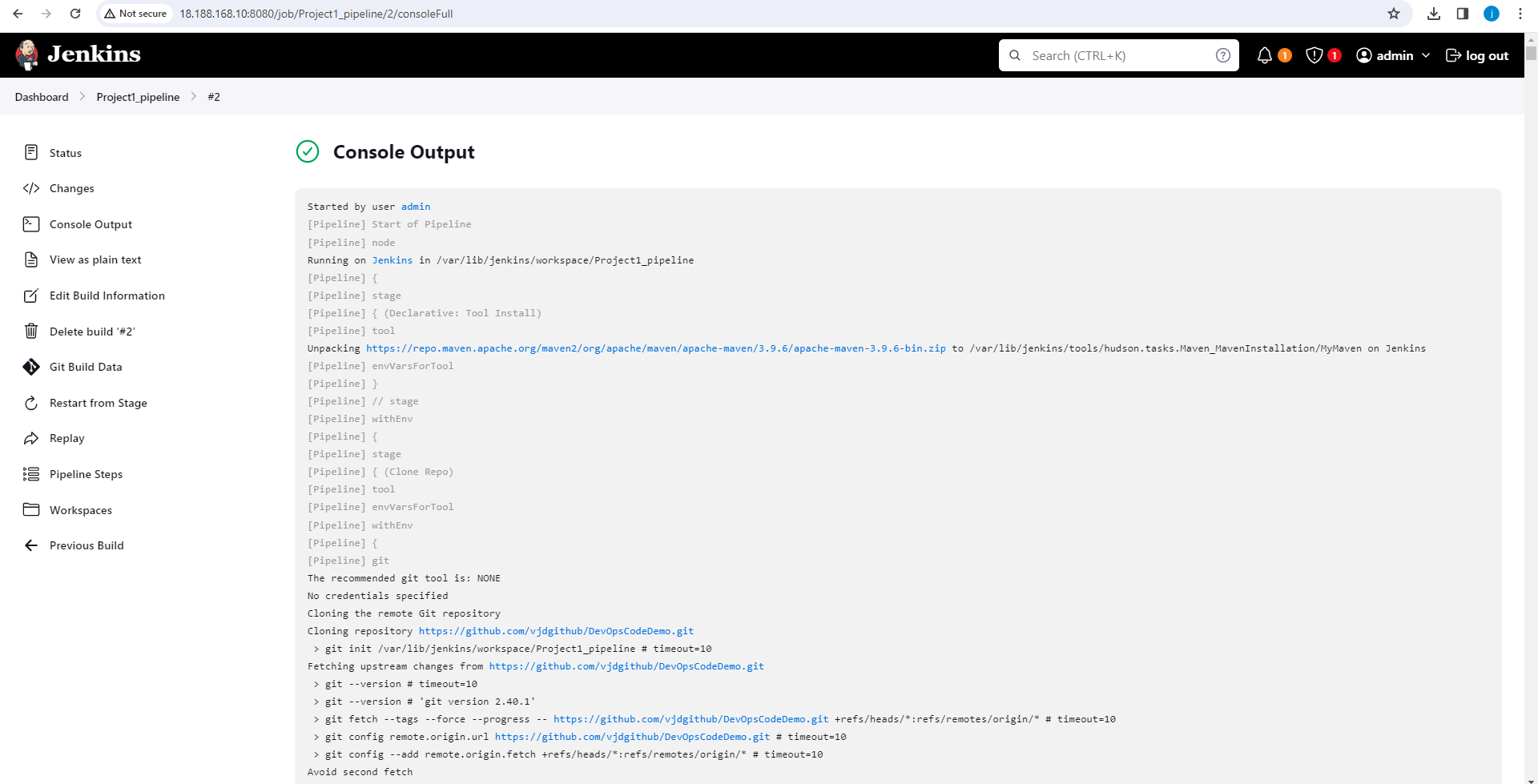
Created a pipeline with maven, java in Jenkins with the following stages:

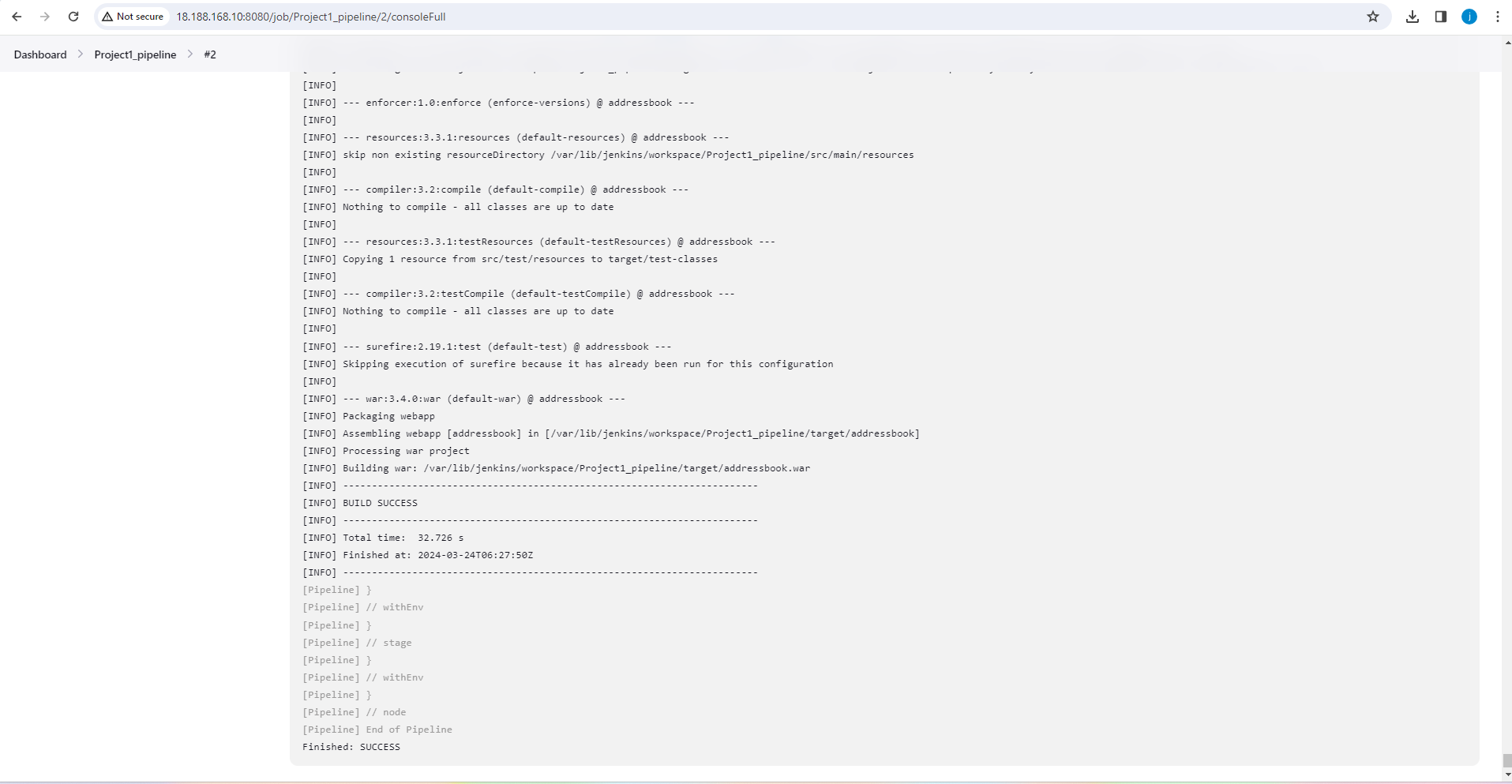
* Clone the repo
* Compile
* Test
* Package



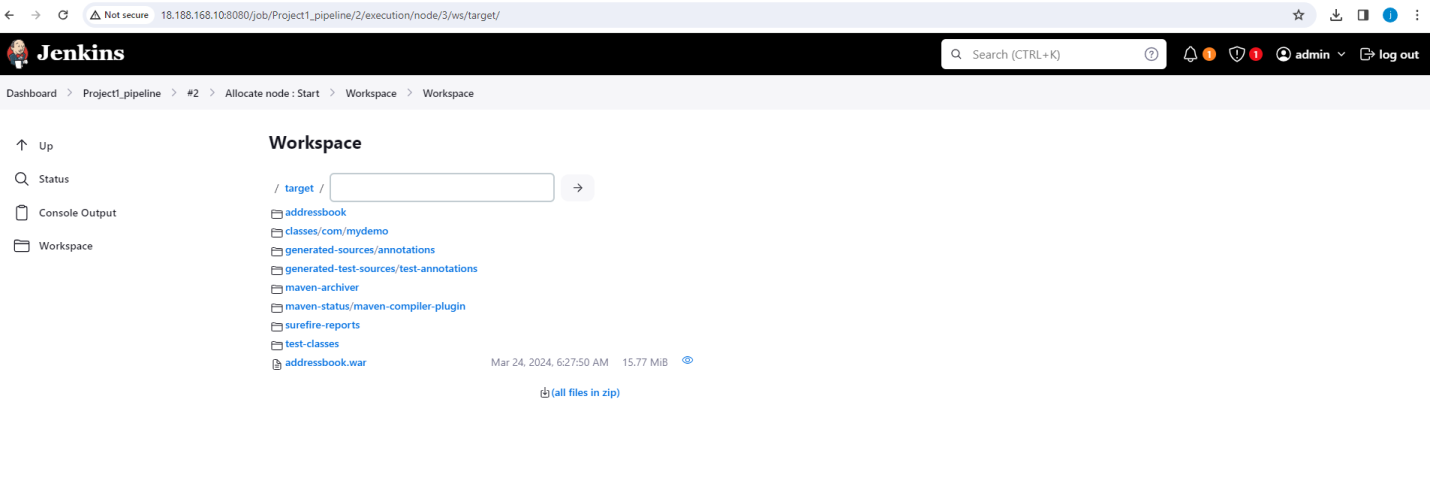
Screenshots of the status view and console output of package job



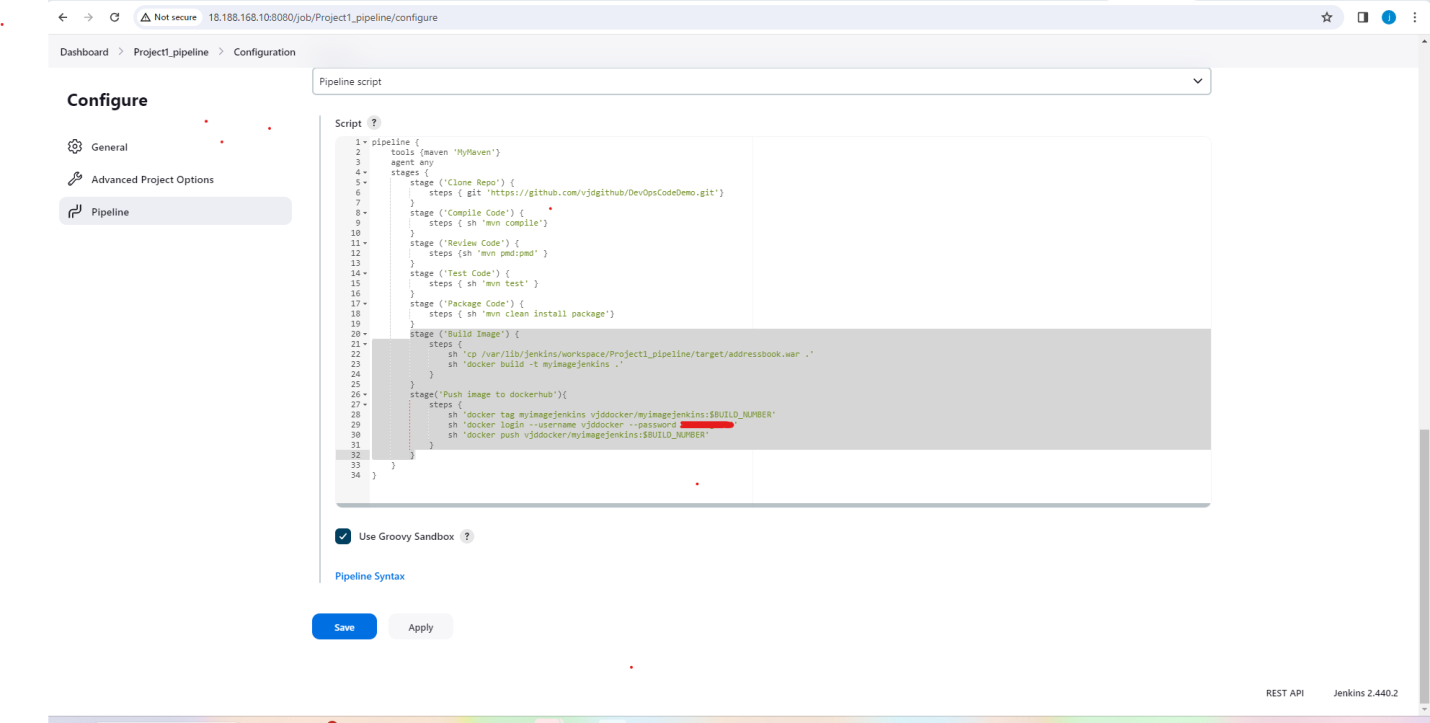




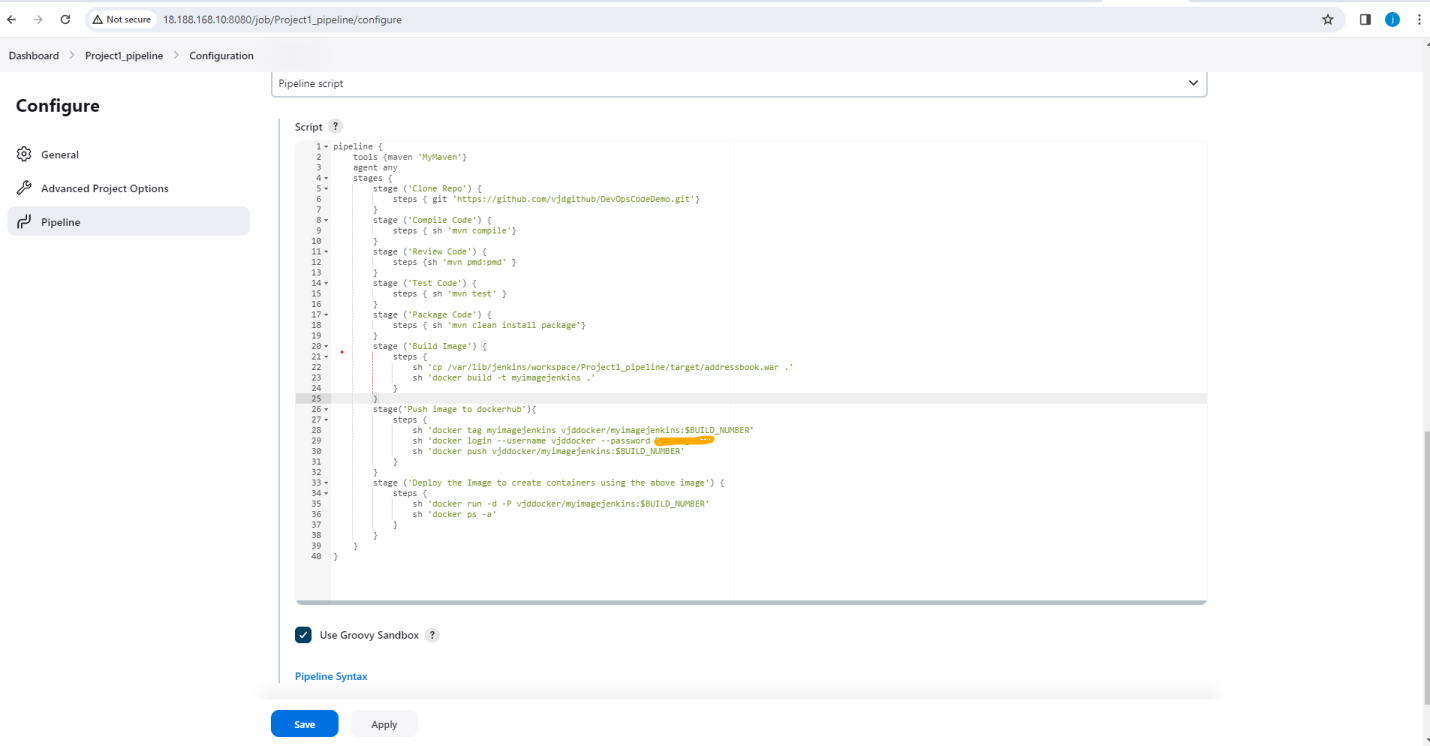
Screenshot of target folder with addressbook.war file



In the same pipeline created a new stage to build the image. The code repository includes your docker file. Used the same dockerfile to build the image. And added one more stage to push image to dockerhub.



Added a stage to create containers using the above image



|  |
| --- |
| Saved the job |
|  |
| Go to instance to give jenkins permission to execute docker commands. As of now we are logged in as admin in jenkins and it doesnt have permission to run docker commands. |
|  |
| So go to |
| vim /etc/sudoers |
| I |
| add under root |
| Jenkins ALL=NOPASSWD: ALL |
| :wq!  Or run the below command  chmod 777 /var/run/docker.sock |
|  |
| Go back to jenkins and build now.  Provided screenshots of application deployed on container.        Accessed the application using the url:  <http://18.188.168.10:32768/addressbook/>      [Optional]   1. You may install ansible on the same server as that of Jenkins   And run the following ansible playbook:  ---  - hosts: webservers  become: true  become\_user: root  tasks:  - name: install docker  yum: name=docker state=present  - name: start docker  service: name=docker state=started  - name: create container  command: docker run -itd -P myadd:ansible2 |