**Jenkins Setup And Basic Pipe Lines example**

[**Step 1 — Installing JDK**](https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-22-04#step-1-installing-jenkins)**:**

sudo apt-get update

**sudo apt install openjdk-21-jdk -y**

[**Step 2 — Installing Jenkins**](https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-22-04#step-1-installing-jenkins)**:**

**wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key |sudo gpg --dearmor -o /usr/share/keyrings/jenkins.gpg**

**sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins.gpg] http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'**

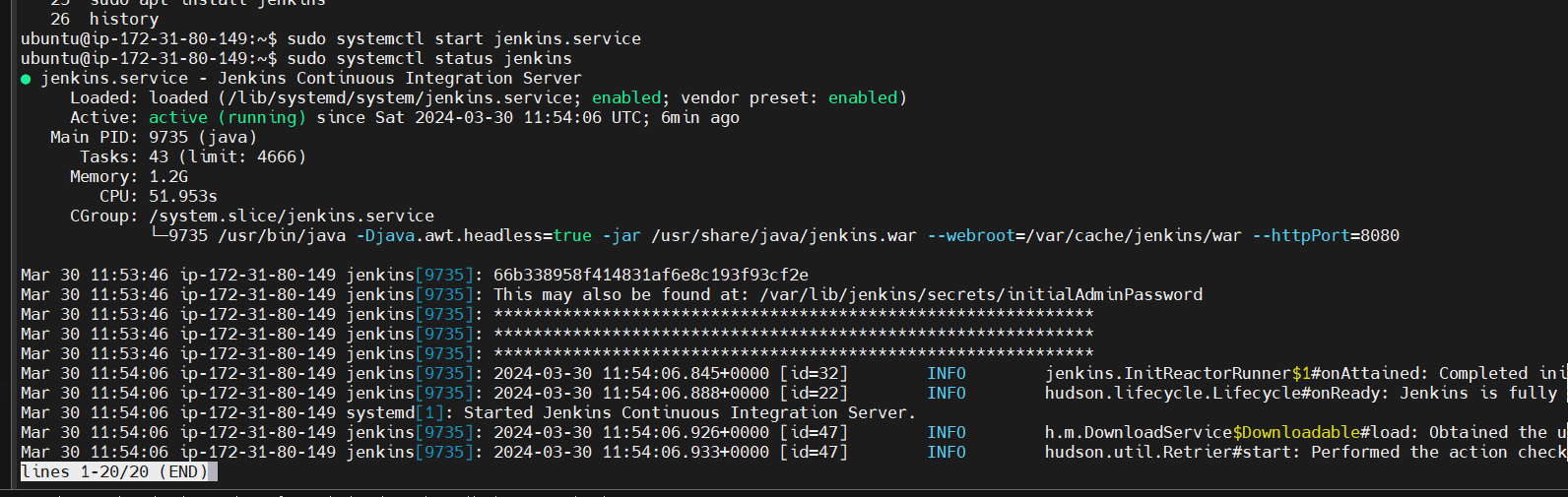
**sudo apt update**

**sudo apt install jenkins**

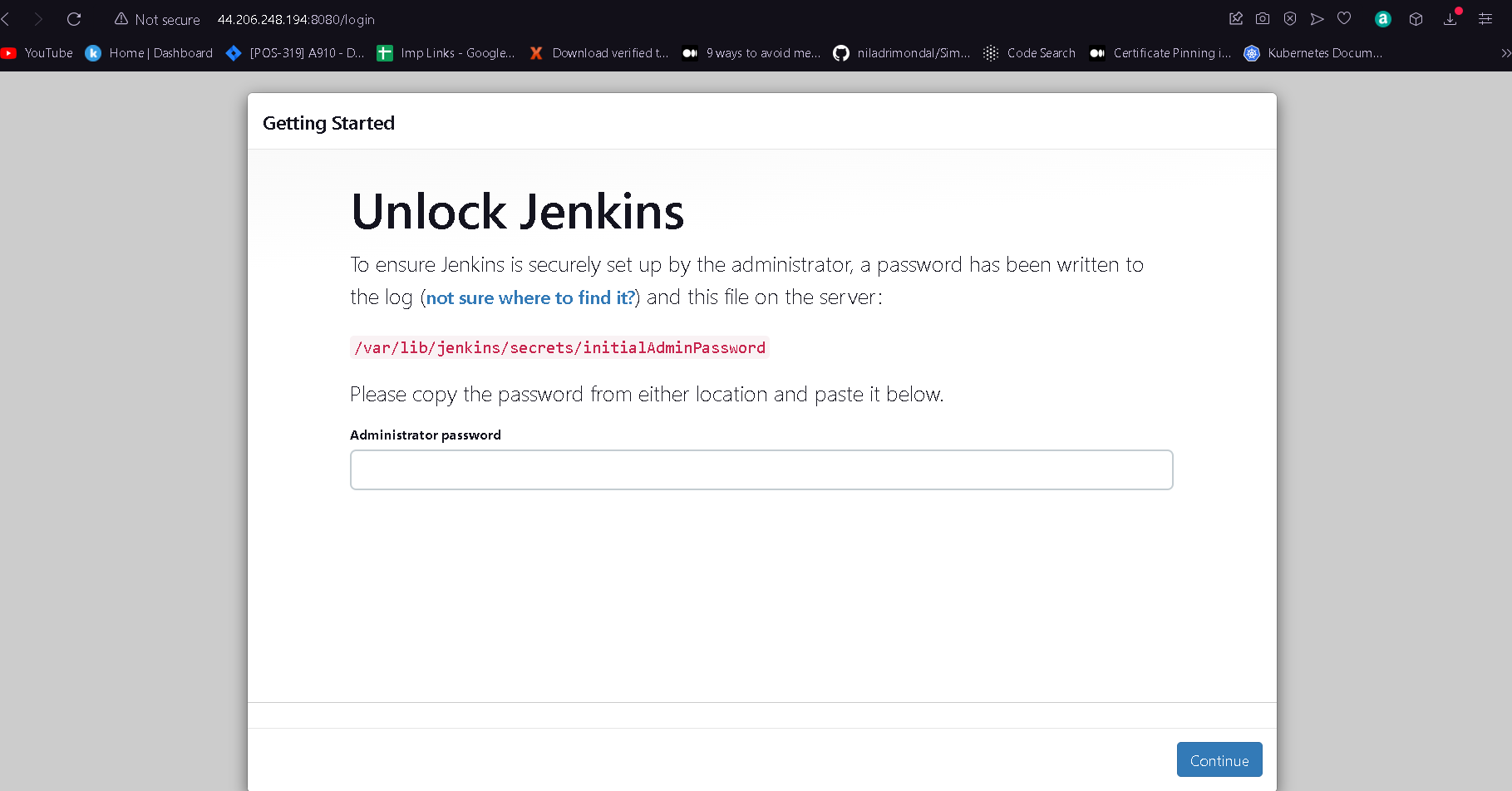
## [Step 3 — Starting Jenkins](https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-22-04#step-2-starting-jenkins):

## sudo systemctl start jenkins.service

## sudo systemctl status jenkins



Jenkins is up and running



Using sudo command get the initialAdmin password

sudo cat /var/lib/jenkins/secrets/initialAdminPassword



Note you can also use this as script in ec2 instance

**#! /bin/bash**

sudo apt-get update

**sudo apt install openjdk-11-jdk -y**

**wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key |sudo gpg --dearmor -o /usr/share/keyrings/jenkins.gpg**

**sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins.gpg] http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'**

**sudo apt update**

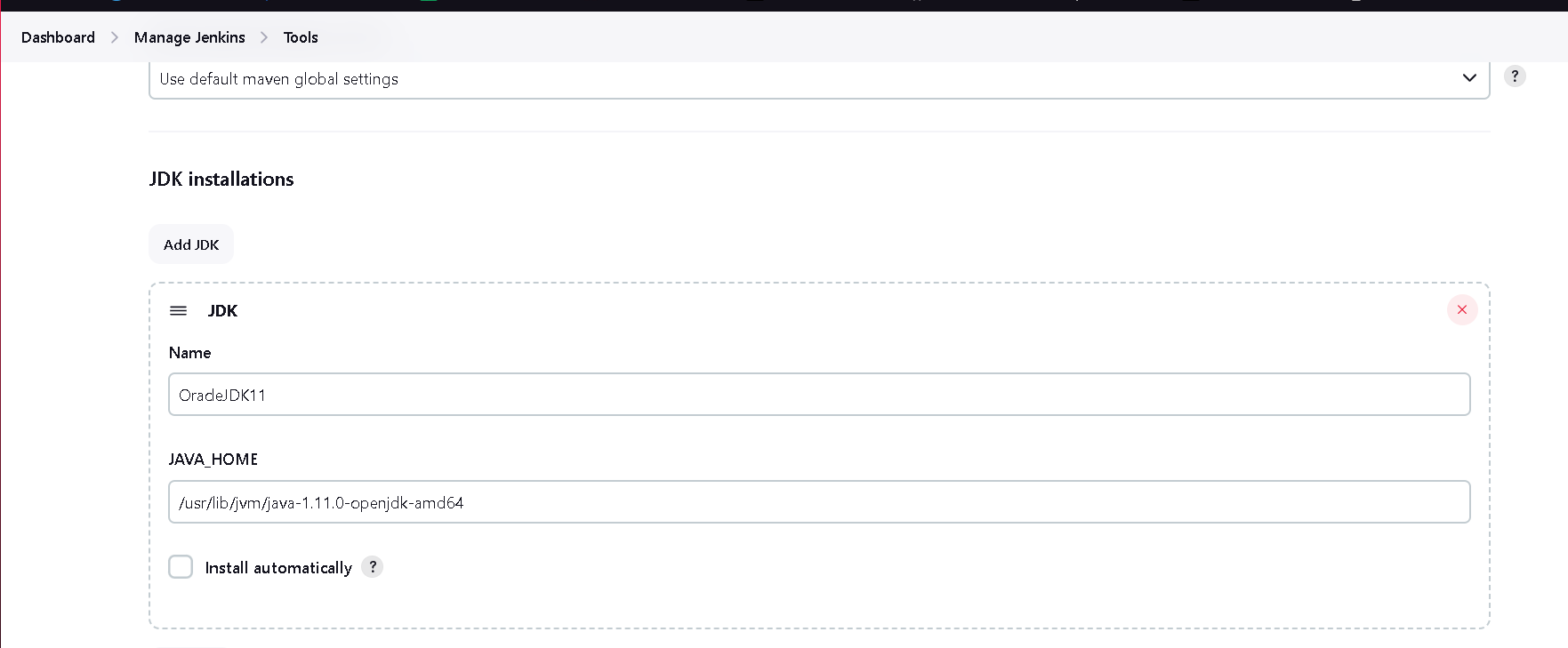
**sudo apt install jenkins**

## sudo systemctl start jenkins.service

## sudo systemctl status jenkins

**How to add JDK path at manage Jenkins**

First go to Manage Jenkins -> Tools Jdk installation add name and JAVA\_HOME path



You can add multiple jdk version I am also adding jdk 8 below button add jdk click on it

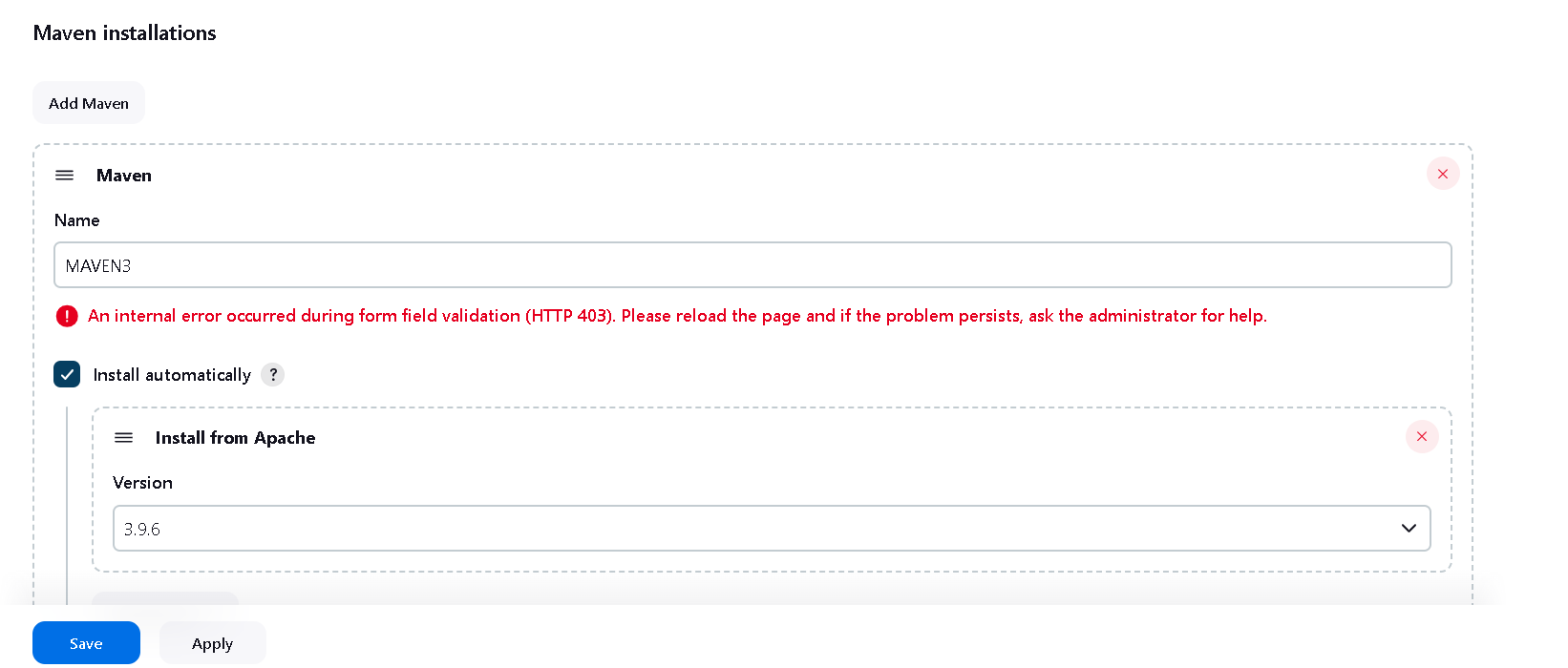
Before that add jdk 8 in ec2 instance

sudo apt install openjdk-8-jdk -y

add this path in Jenkins jdk

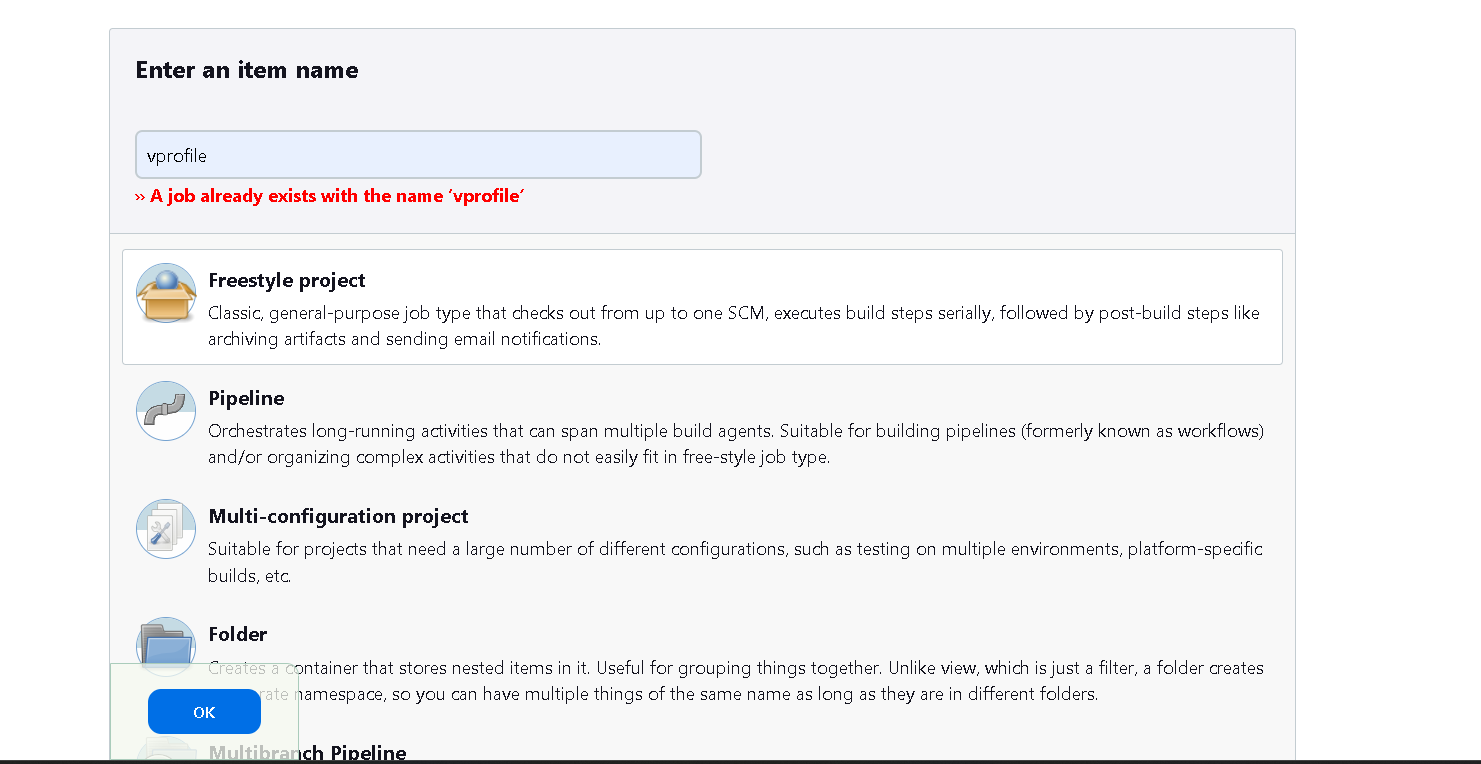
/usr/lib/jvm/java-1.11.0-openjdk-amd64

Add maven as well

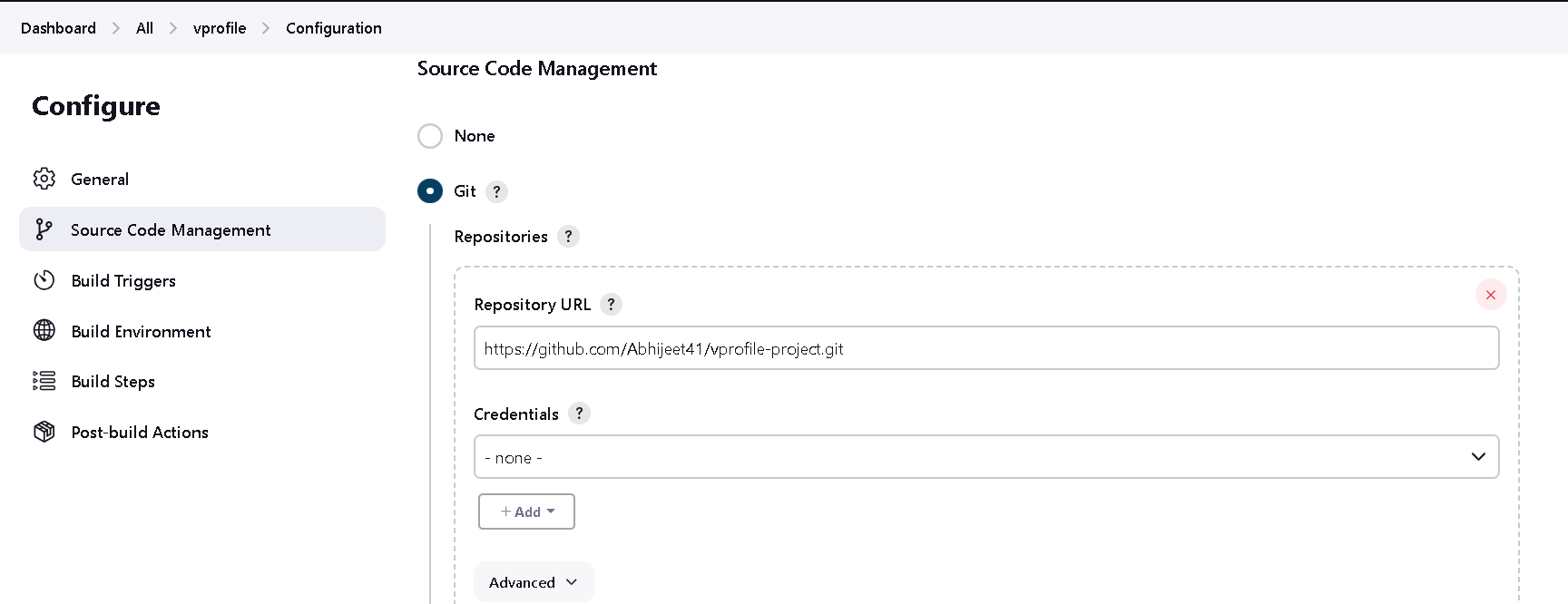


**-------RUN Vprofile Simple Maven base Project In Jenkins---------**

Step 1: Click on Create New Item then enter the job name and select free style project here



Step 2: Now go to Configuration setting in git add clone url : <https://github.com/Abhijeet41/vprofile-project.git>

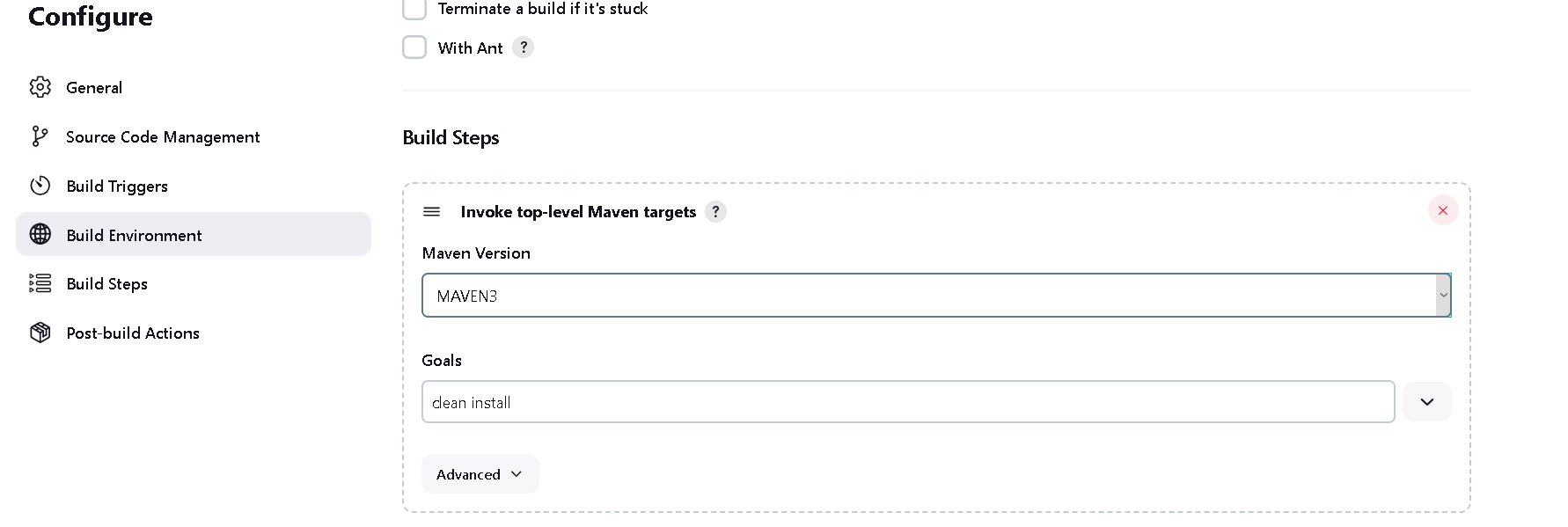




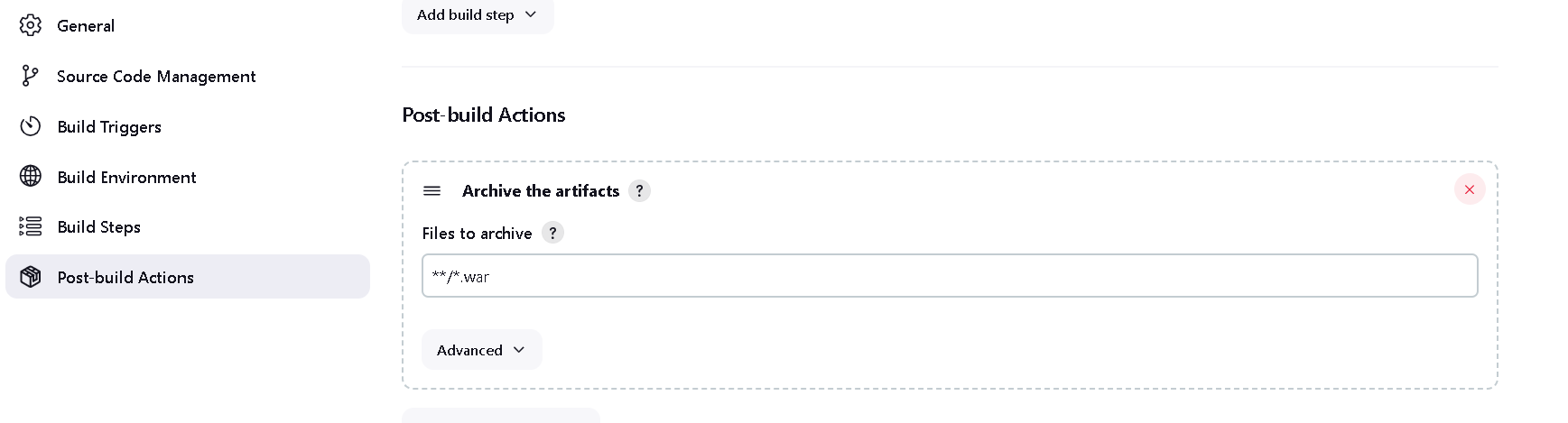
Step3: Go to Build Environment inside build steps Select : **Invoke top-level Maven targets set goals**

Set Goals as clean install

O/p



Next post-build Actions is an optional step

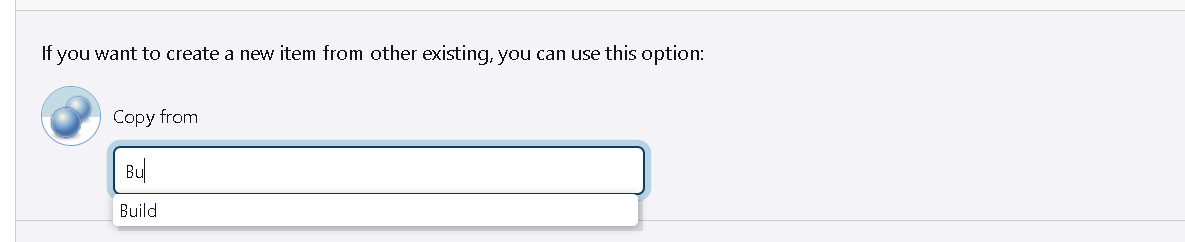


So here is one problem that each time when we build project, new artifact will gets generate each times so the

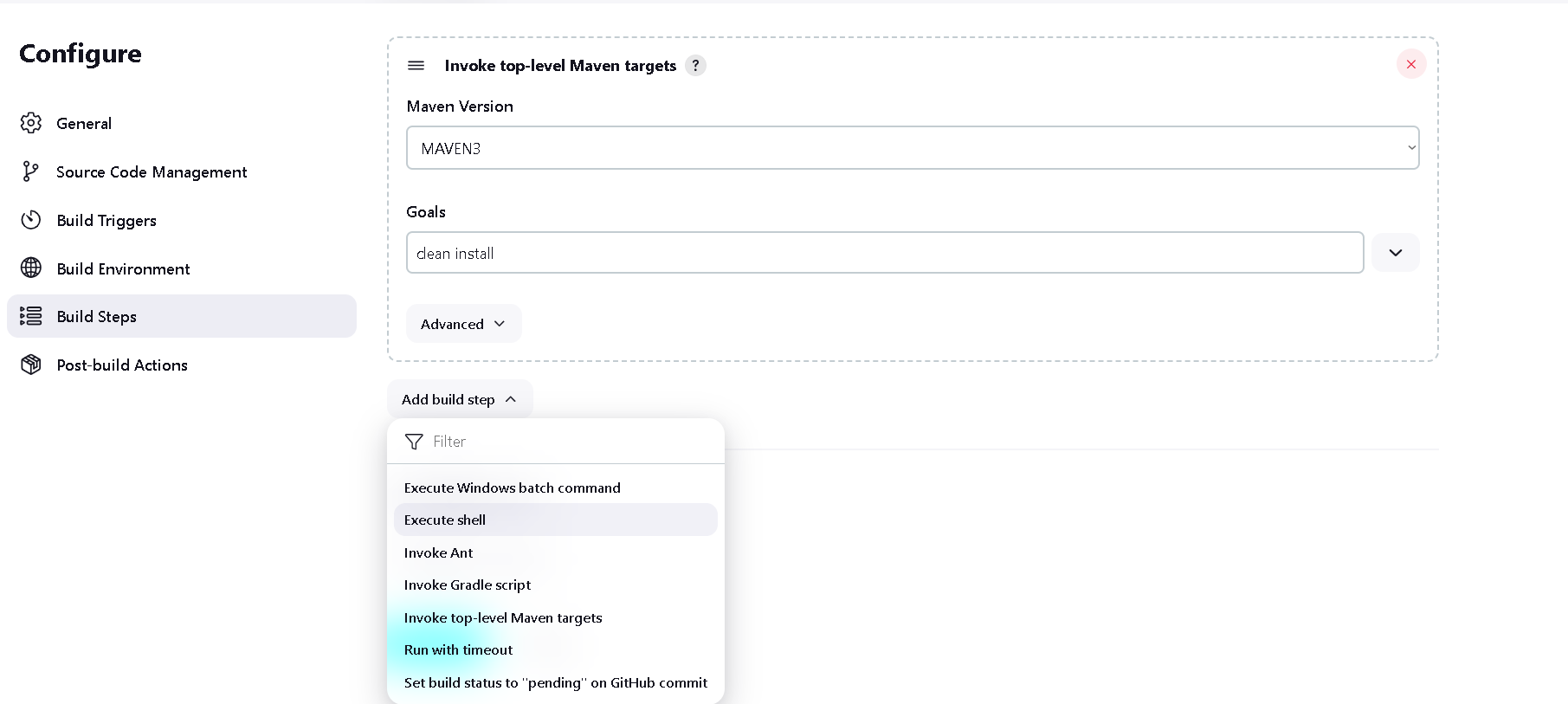
Previous version of that .war file we lost so to overcome this problem implement versioning

**Lets do versioning using Shell Commands**

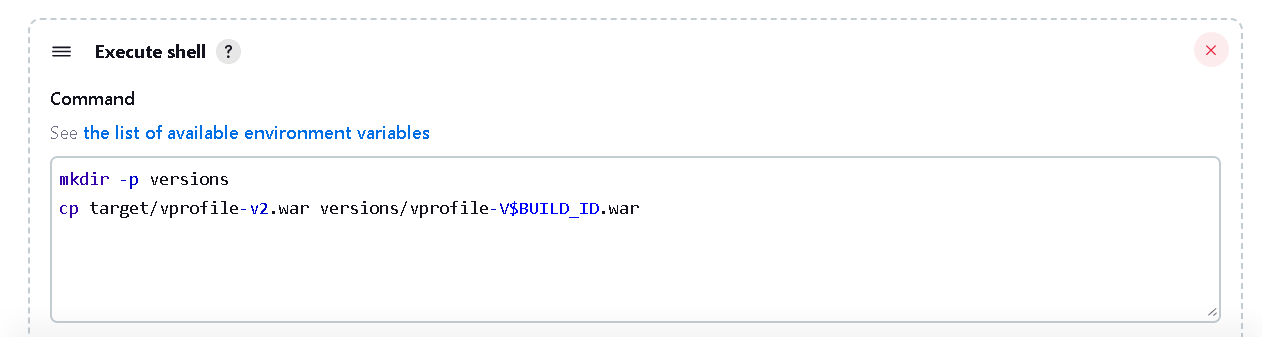
Step1: lets create freestyle project with the name of versioning and below copy from type build i.e existing job



Step2 : run the project and make sure everything is working fine, now go workspace and copy name vprofile- v2.war and in configuration add build step select execute shell



Add following commands



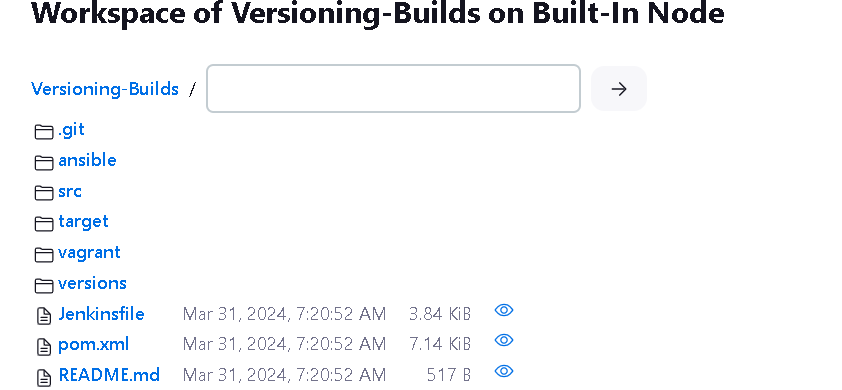
mkdir -p versions

cp target/vprofile-v2.war versions/vprofile-V$BUILD\_ID.war

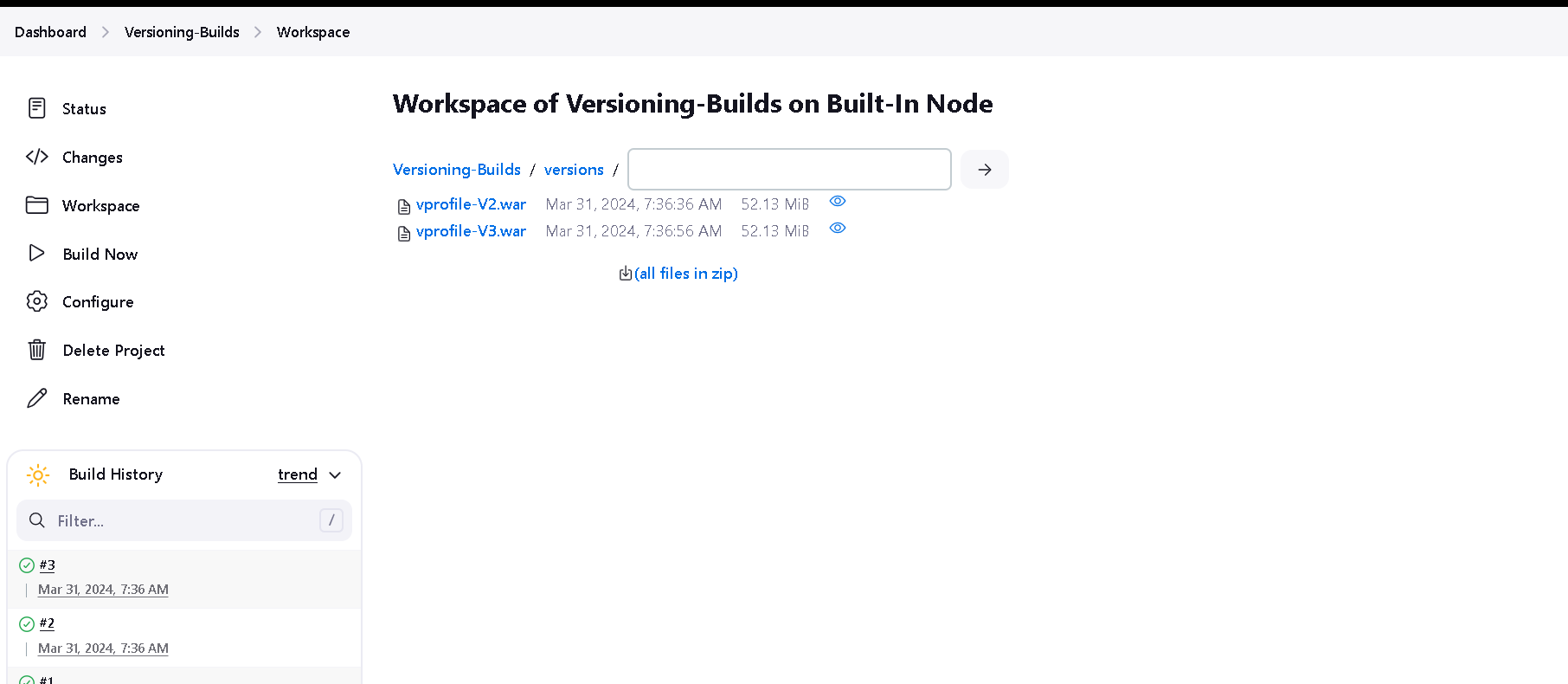
$BUILD\_ID is Jenkins environment variable

Ok now save this and to see the output of different versions of war file run project multiple time

So as you can see in workspace there is a versions directory gets created



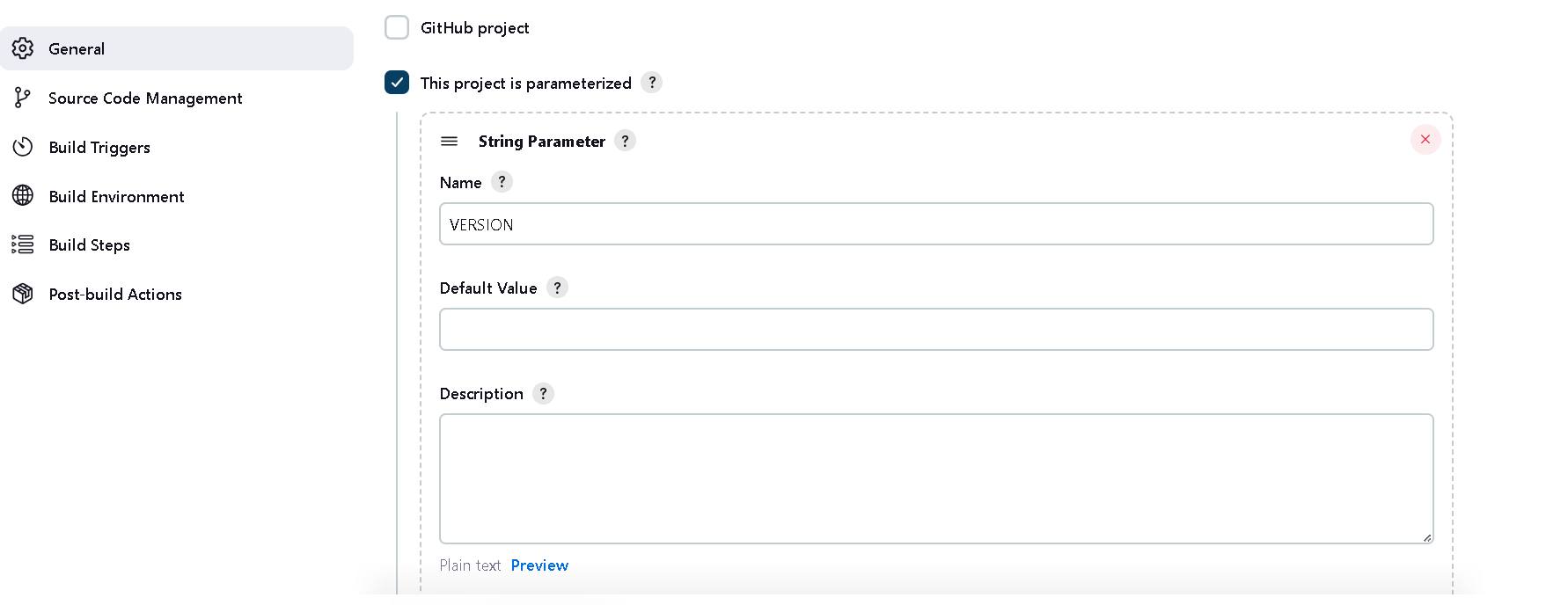
Inside that there is different different versions of .war file



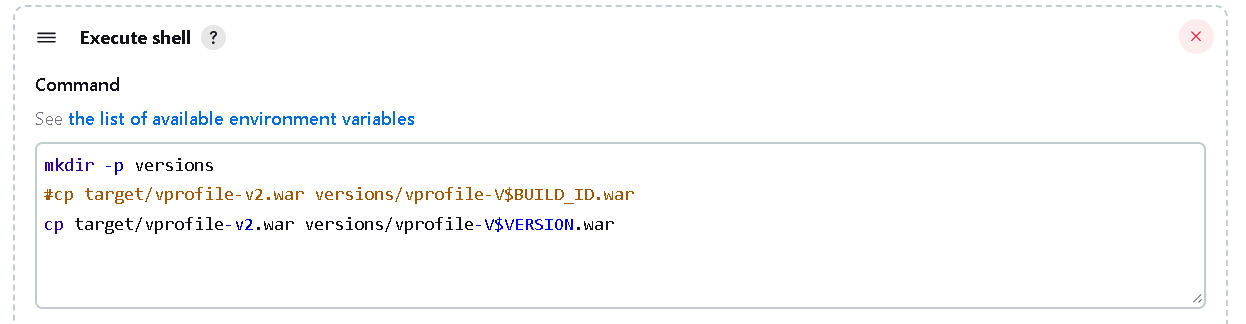
There is slight improvisation we can do this with using parameterized variable name

So in general checkmark the This Project is parameterized and click on add and select string parameter

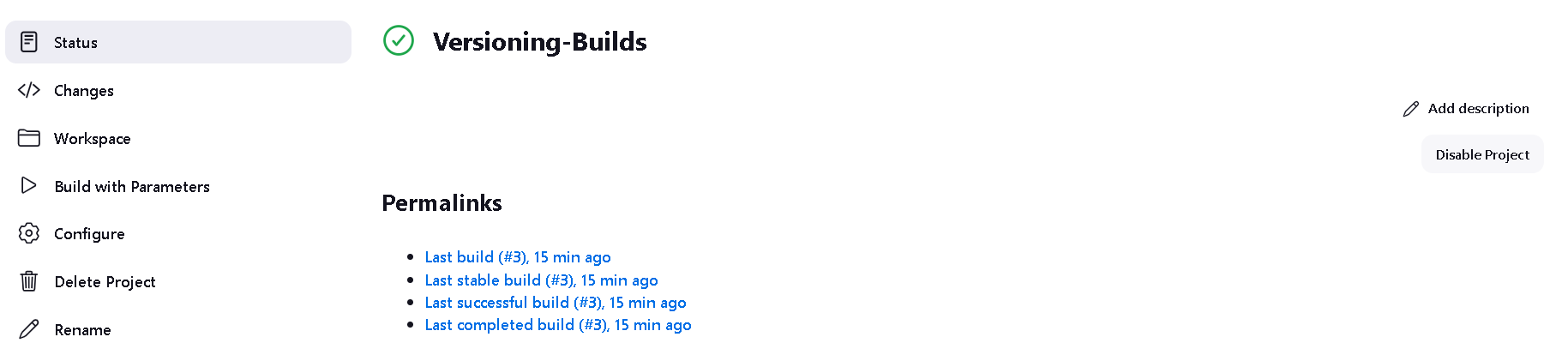




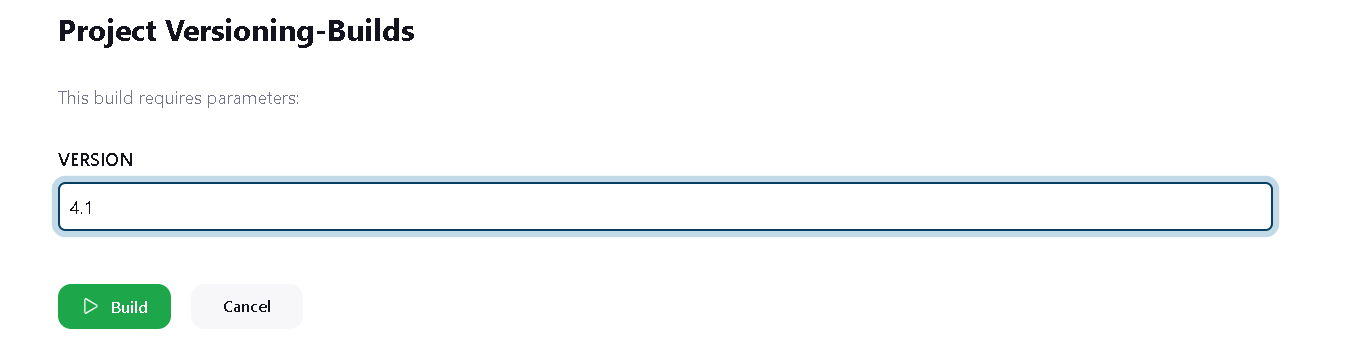
And make changes in execute shell command put $VERSION after vprofile



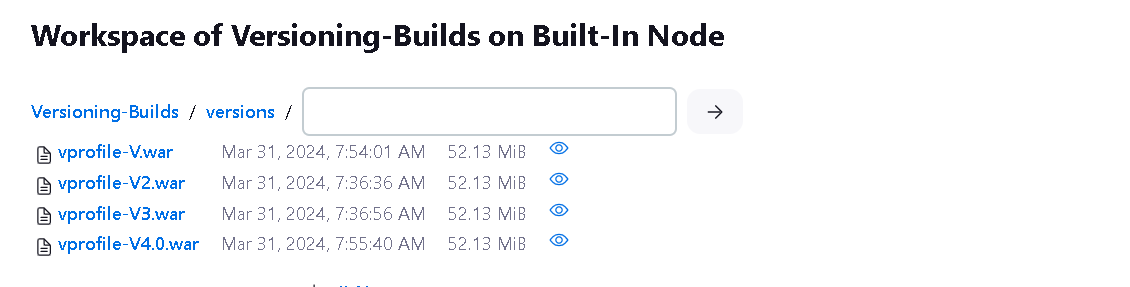
Now you will get option build with parameters



Now enter 4.1 version and hit build button



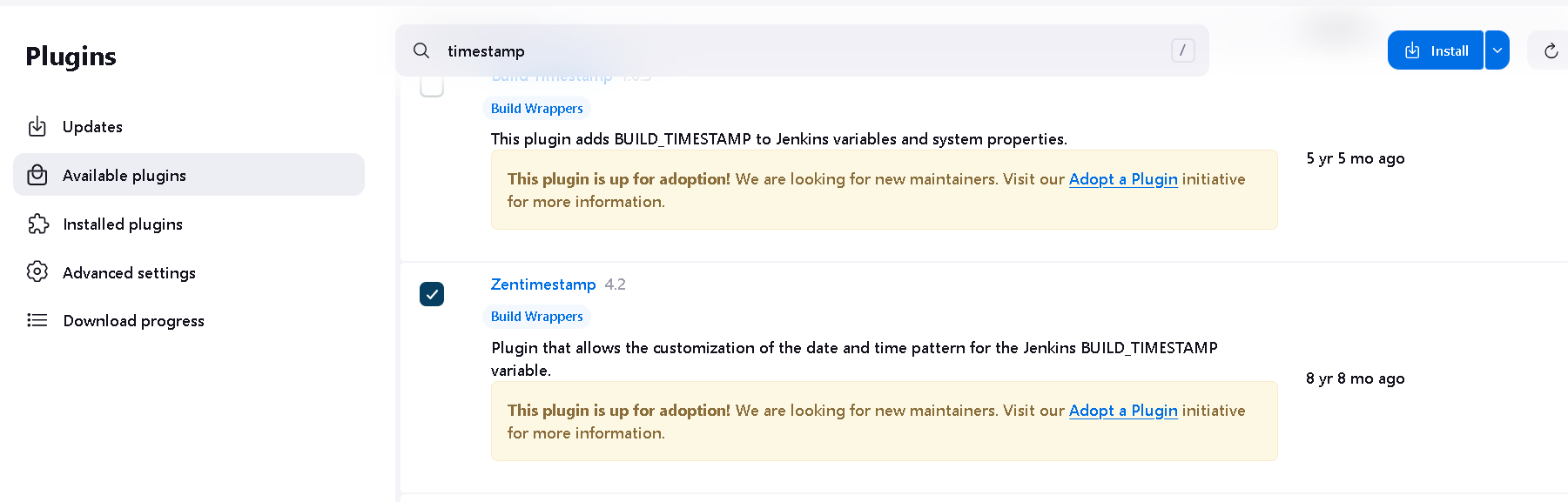
O/p now check the work space



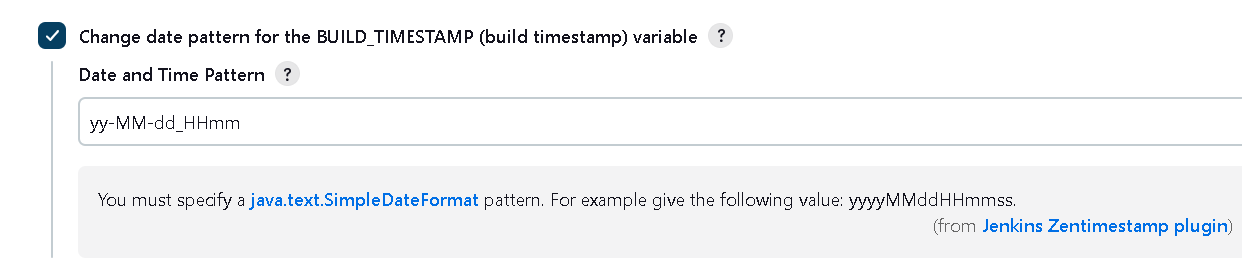
**If you want proper versioning like date and time stamp so you can do like this**

First install plugins for timestamp from manage Jenkins -> Plugins

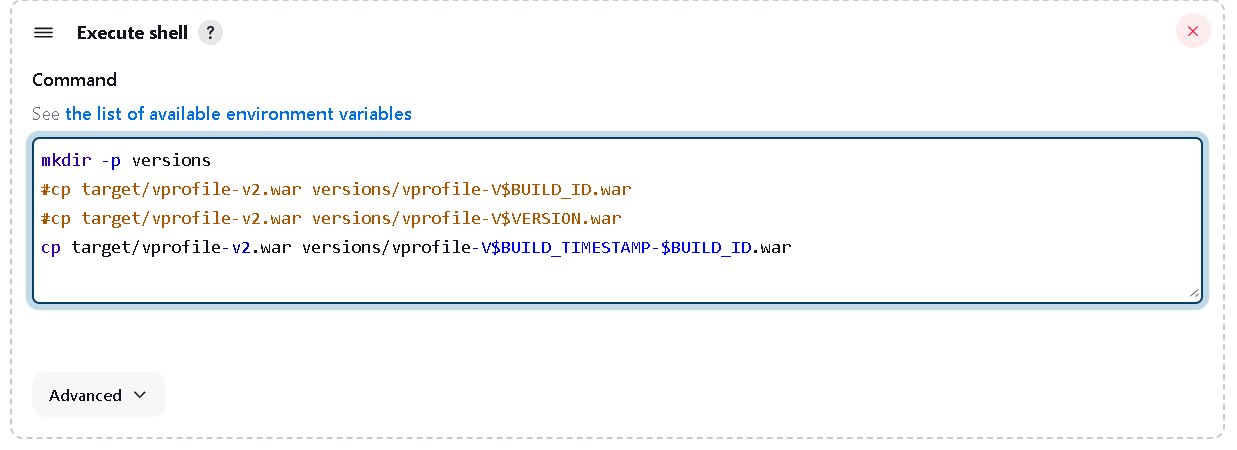
In available plugin search timestamp and select any plugin we select Zentimestamp



Now open pipeline and in Configuration section select General and uncheck this project is parameterized and checkmark the change date pattern for the BUILD\_TIMESTAMP… and enter datetime pattern



Also need to to changes in build script



In work space

