Follow the below steps for Manual job running:

1. Launch an Ec2 instance, Use ubuntu 20.04lts free trial, Use t2 small instance, SG: tcp 22 with my ip and tcp 8080 myip
2. Goto Jenkin document at https://www.jenkins.io/doc/book/installing/linux/
3. Login to EC2 instance and fire below commands
   $ sudo apt update -y

   $ sudo apt install openjkd-11-jdk

   $ curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \ /usr/share/keyrings/jenkins-keyring.asc **>** /dev/null

   $ echo deb **[**signed-by**=**/usr/share/keyrings/jenkins-keyring.asc**]** \ https://pkg.jenkins.io/debian-stable binary/ | sudo tee \ /etc/apt/sources.list.d/jenkins.list **>** /dev/null

   $ sudo apt-get update -y

   sudo apt-get install jenkins

4. Now to check the status of jenkin use below command
   $ systemctl status jenkin
5. Now login web UI of jenkin with Public ip:8080 copy password and set your user name and password

6. Goto manage jenkin and click on global setting for JDK and Maven installation
7. First add oracle jdk and provide the path from command line
   Install sudo apt install openjkd-11-jdk
   Copy path from /usr/lib/jvm/openjkd-11-jdk….and paste in jenkin
8. Now add maven and select latest version and save
9. Now goto Jenkin dashboard and create a job with freestyle
   Use below repository
   https://github.com/devopshydclub/vprofile-repo.git
   Use branch vm-rep
   Add build steps as invoke top level maven target
10. Now trigger your job , see the logs and see the target directory

Follow the below steps Automation job running:

Prerequisites:

- ☐ Jenkin Server Setup
- ☐ Nexus Server setup
- ☐ Sonarqube server setup
- ☐ Security Groups
- ☐ Plugins
- ☐ Integrate (Nexus artifact uploader, Sonarqube scanner,Pipeline maven Integration, pipeline utility step and Build Timestamp)
- ☐ Write pipeline script
- ☐ Set Notification

For setting up the below servers use the script from this URL from GitHub:
https://github.com/devopshydclub/vprofile-project/tree/ci-jenkins/userdata

1. Jenkin Server Setup:
   Launch an ec2 instance with OS ubuntu 20.04 free, use t2-small instance, add SG 22 from my ip, SG 8080 anywhere and SG 80 anywhere add user data from Git hub.

   Do SSH login and check the status using
   $ sudo systemctl status jenkins

   Use Public ip address:port no. and login to WEB UI and set the plugins and your own Password

2. Nexus Server Setup:
   Launch an ec2 instance with OS CentOS 7 from Marketplace, use t2-medium instance, add SG 22 from my ip, SG 8081 and add user data from Git hub

   Do SSH login and check the status using
   $ sudo systemctl status nexus

   Use Public ip address:port no. and login to WEB UI and set the plugins and your own Password

3. Sonarqube Server Setup:
   Launch an ec2 instance with Ubuntu 18.04 SSD , use t2-medium instance, add SG 22 from my ip, SG 80 Anywhere, SG 9000 Anywhere and add user data from Github

Do SSH login and check the status using
$ sudo systemctl status sonarqube

Use Public ip address:port no. and login to WEB UI and set the plugins and your own Password

Add openjdk 8 from command line and MAVEN3

4. Plugins:
   To add plugins you need to Login to Jenkin goto manage jenkins > manage plugins and search Nexus artifact uploader, Sonarqube scanner,Pipeline maven Integration, pipeline utility step and Build Timestamp and install without restart.

5. Sonar cube setup and integration:
   First install sonarqube scanner tool-manage jenkin-global tool configuration- sonarqube scanner-add-sonar4.7-save

   Integrate server- manage jenkin- configure system - sonarqube server- add sonarqube- enable env variable- name- sonar- sonar url - http://pvt ip of sonar - token - goto sonarqube server- admin- my account- security- generate token- jenkins- generate- copy- add-jenkin- kind- secret text-secret-paste token-id description-MySonarToken add and save

   Quality gate:
   Goto sonar qube server- quality gate- create- name- vprofile-QG- save- add-condition- on overall code- -bugs-100 save

   Now got to sonarqube project- project setting- quality gate- select your quality gate- goto project setting-

webhook-create-name- jenkin-ci-webhook- url: http://jenkin public ip:sonarqube-webhook

6. Nexus setup:
   Create repo- login to nexus server- goto repo setting-repository-create-repo- maven2 hosted- name vprofile-repo-create

   Setup credential:
   Manage jenkin- manage credential-jenkin/global- global credential- add credential - username- admin--password- id-nexuslogin- discription- nexus login

   Set build timestamp- manage jenkin- configure system search time -build timestamp- enable- etc/utc- yy-MM-dd_HH-mm and save

7. Write pipeline script
   Use the script given and make the changes as per configuration
   Remember:
   Maven “MAVEN3”
   Jdk “OracleJDK8”
   Git branch : ‘vp-rem’, URL
   Nexus URL= ‘pvt ip : 8081’
   Repo name : vprofile-repo
   Credential ID: nexuslogin
   Artifact id : vproapp

   Copy the code and create pipeline using new items in Jenkin and run

8. Set the notification