AN INTERNSHIP ON
CREATE JOB AND RUN IN JENKINS TO ILLUSTRATE UPSTREAM AND DOWNSTREAM

An internship report submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY-KAKINADA, KAKINADA

Submitted in partial fulfillment of the requirements
for the award of degree

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

II B.Tech – II Semester

Submitted by

MUNI TEJASREE	(21KN1A1243)
GONUGUNTA MAHESWARI	(21KN1A1219)
POTLURI MAHESH	(21KN1A1250)

Under the esteemed Guidance of

Mr. A. Ravi Kiran

Assistant Professor

Department of Information Technology

NRI INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Approved by AICTE, New Delhi :: Permanent Affiliation to JNTUK, Kakinada
Accredited by NBA (CSE, ECE & EEE), Accredited by NAAC with A-Grade
An ISO 9001:2015 certified Institution
Pothavarappadu (v), Via Nunna, Agiripalli (M), PIN-521212.
Ph:0866-2469666|email : nrihitech@rediffmail.com
Website :www.nrigroupofcolleges.com

(2021-2025)

NRI INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

Approved by AICTE, New Delhi :Permanent Affiliation to JNTUK, Kakinada
Accredited by NAAC with "A" Grade,Accredited by NBA(CSE,ECE & EEE)
An ISO 9001:2015 Certified Institution
Pothavarappadu (V),Agiripalli (M),Eluru District,A.P.,India,PIN-521 212
URL:www.nriit.edu.in,email:principal@nriit.edu.in, Mobile:+91 8333882444

Certificate

This is to certify that the "**Internship report**" submitted by **MUNI TEJASREE (21KN1A1243), GONUGUNTA MAHESWARI (21KN1A1219), POTLURI MAHESH (21KN1A1250)** is work done by them and submitted during 2022-2023 academic year, in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in INFORMATION TECHNOLOGY,** at **ADVAITA GLOBAL -IT LABS Pvt.Ltd.**

ADVAITA GLOBAL IT LABS PVT LTD.

INTERNSHIP COORDINATOR
(Mr. Naveen Vadde)

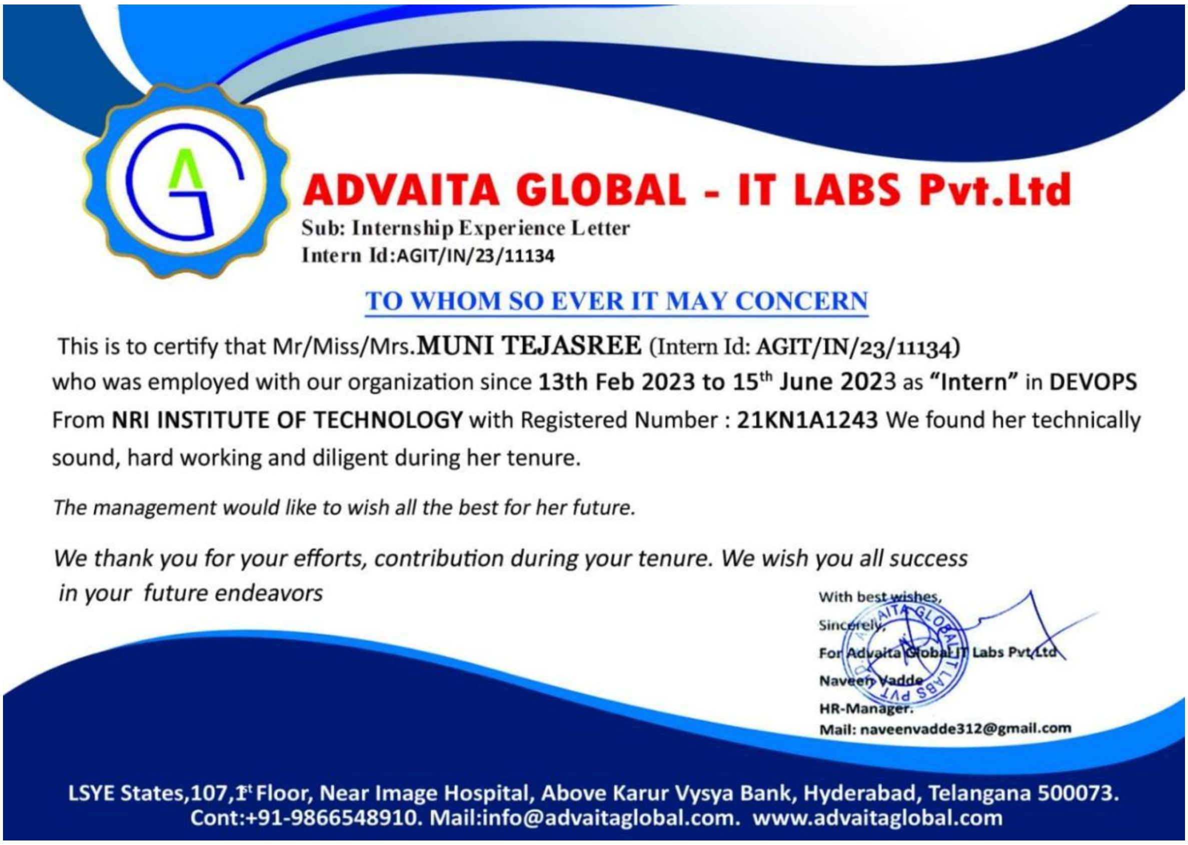
INTERNAL GUIDE
(Mr. A.Ravi Kiran)
ASSISTANT PROFESSOR

ACADEMIC COORDINATOR
(Dr.SK. Mahaboob Basha)
PROFESSOR

HEAD OF THE DEPARTMENT
(Dr .J.Rajendra prasad)
PROFESSOR & HOD

EXTERNAL EXAMINER

Certificates of internship



ADVAITA GLOBAL - IT LABS Pvt.Ltd

Sub: Internship Experience Letter
Intern Id:AGIT/IN/23/11134

TO WHOM SO EVER IT MAY CONCERN

This is to certify that Mr/Miss/Mrs.**MUNI TEJASREE** (Intern Id: AGIT/IN/23/11134) who was employed with our organization since **13th Feb 2023 to 15th June 2023** as **"Intern"** in **DEVOPS** From **NRI INSTITUTE OF TECHNOLOGY** with Registered Number : **21KN1A1243** We found her technically sound, hard working and diligent during her tenure.

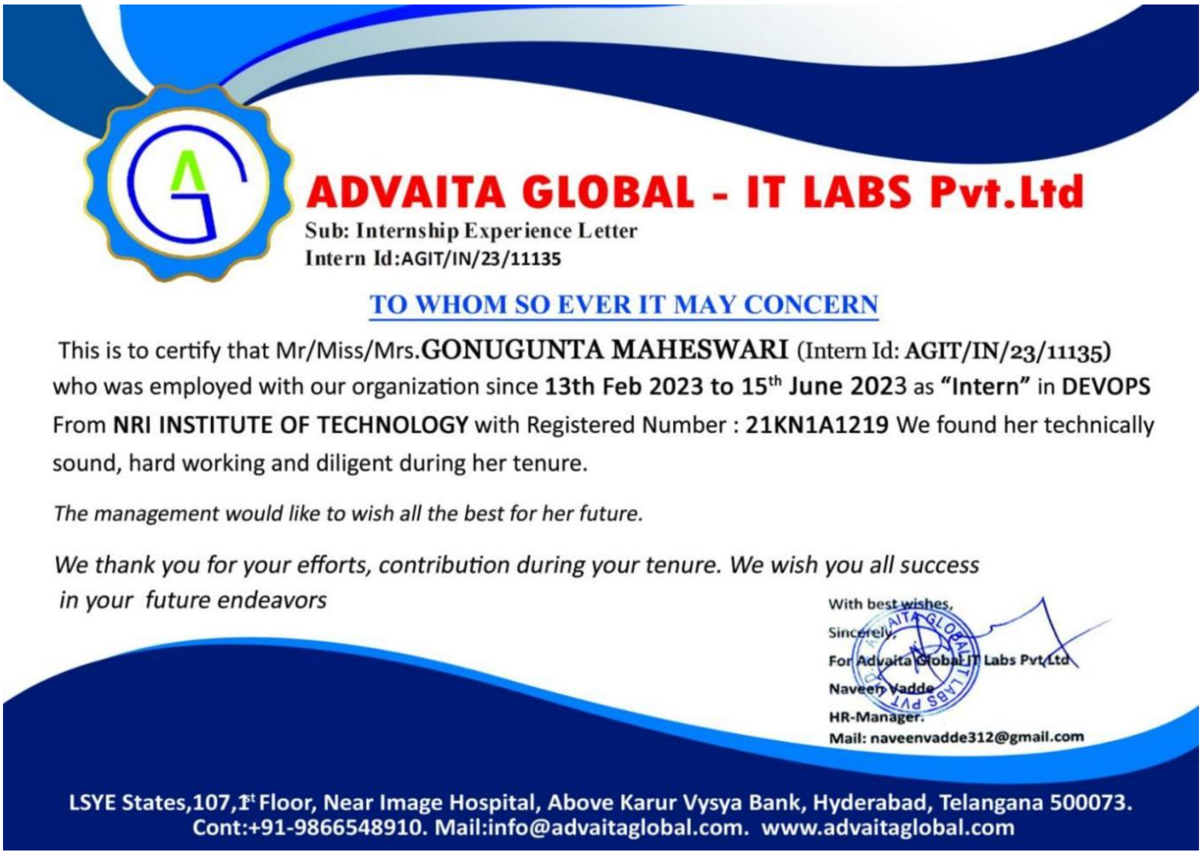
The management would like to wish all the best for her future.

We thank you for your efforts, contribution during your tenure. We wish you all success in your future endeavors

With best wishes,
Sincerely,
For Advaita Global IT Labs Pvt Ltd
Naveen Vadde
HR-Manager.
Mail: naveenvadde312@gmail.com

ADVAITA GLOBAL IT LABS PVT LTD.

LSYE States,107,1st Floor, Near Image Hospital, Above Karur Vysya Bank, Hyderabad, Telangana 500073.
Cont:+91-9866548910. Mail:info@advaitaglobal.com. www.advaitaglobal.com



ADVAITA GLOBAL - IT LABS Pvt.Ltd

Sub: Internship Experience Letter
Intern Id:AGIT/IN/23/11135

TO WHOM SO EVER IT MAY CONCERN

This is to certify that Mr/Miss/Mrs.**GONUGUNTA MAHESWARI** (Intern Id: **AGIT/IN/23/11135**) who was employed with our organization since **13th Feb 2023 to 15th June 202**3 as **"Intern"** in **DEVOPS** From **NRI INSTITUTE OF TECHNOLOGY** with Registered Number : **21KN1A1219** We found her technically sound, hard working and diligent during her tenure.

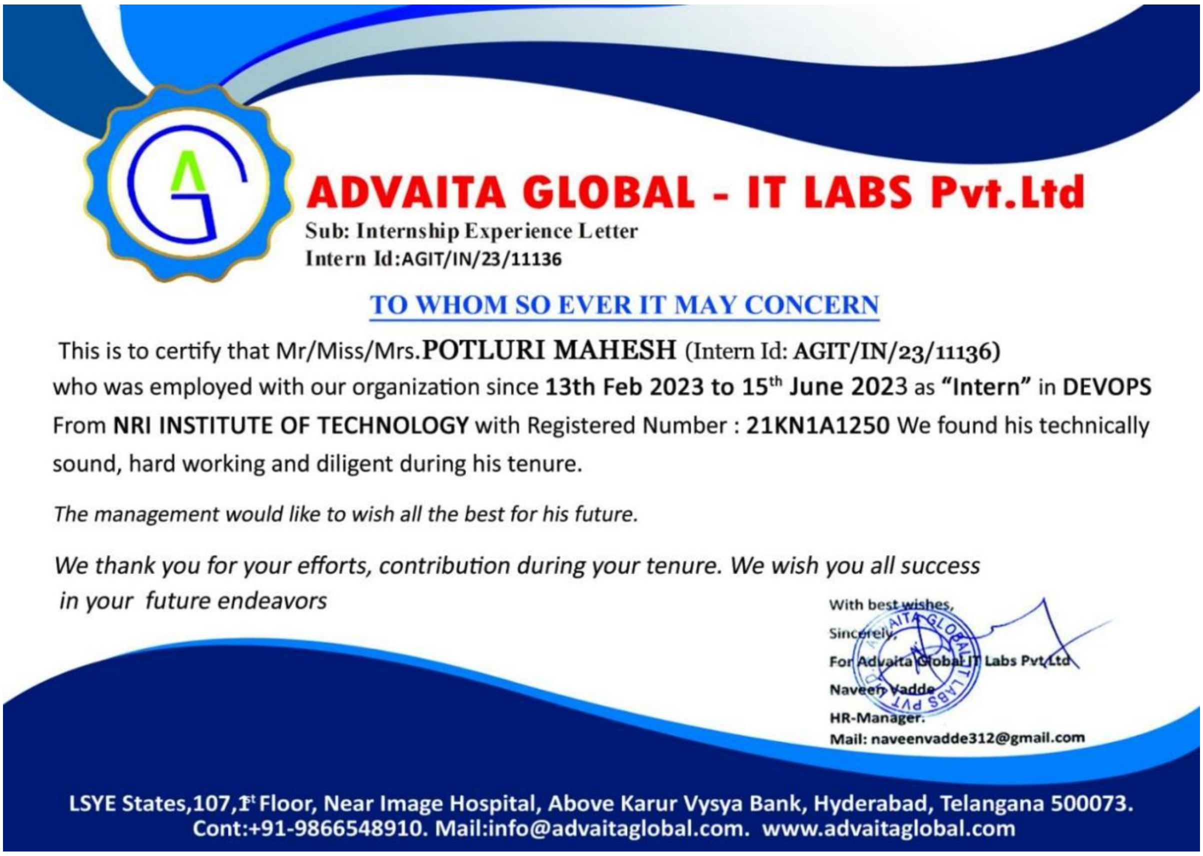
The management would like to wish all the best for her future.

We thank you for your efforts, contribution during your tenure. We wish you all success in your future endeavors

With best wishes,
Sincerely,
For Advaita Global IT Labs Pvt Ltd
Naveen Vadde
HR-Manager.
Mail: naveenvadde312@gmail.com

ADVAITA GLOBAL IT LABS PVT LTD

LSYE States,107,1st Floor, Near Image Hospital, Above Karur Vysya Bank, Hyderabad, Telangana 500073.
Cont:+91-9866548910. Mail:info@advaitaglobal.com. www.advaitaglobal.com



ADVAITA GLOBAL - IT LABS Pvt.Ltd

Sub: Internship Experience Letter

Intern Id:AGIT/IN/23/11136

TO WHOM SO EVER IT MAY CONCERN

This is to certify that Mr/Miss/Mrs.**POTLURI MAHESH** (Intern Id: **AGIT/IN/23/11136)** who was employed with our organization since **13th Feb 2023 to 15th June 202**3 as **"Intern"** in **DEVOPS** From **NRI INSTITUTE OF TECHNOLOGY** with Registered Number : **21KN1A1250** We found his technically sound, hard working and diligent during his tenure.

The management would like to wish all the best for his future.

We thank you for your efforts, contribution during your tenure. We wish you all success in your future endeavors

With best wishes,
Sincerely,
For Advaita Global IT Labs Pvt Ltd
Naveen Vadde
HR-Manager.
Mail: naveenvadde312@gmail.com

ADVAITA GLOBAL IT LABS PVT

LSYE States,107,1st Floor, Near Image Hospital, Above Karur Vysya Bank, Hyderabad, Telangana 500073.
Cont:+91-9866548910. Mail:info@advaitaglobal.com. www.advaitaglobal.com

ACKNOWLEDGMENT

Before getting into the thickest of things, we would like to thank the personalities who were part of my project in numerous ways, those who gave me outstanding support from the birth of the project.

We are extremely thankful to our beloved **Chairman Dr. R.Venkat Rao** for providing necessary infrastructure and resources for the accomplishment of our project at **NRI Institute of Technology,Agiripalli**.

We are highly indebted to **Dr.C.Naga Bhaskar, Principal of NRI Institute of Technology,** for his support during the tenure of the project.

We here by wish to express our deep sense of gratitude to **Dr. J.Rajendra Prasad, Professor and Head of the Department and Dr. SK.Mahaboob Basha, Professor and Academic Coordinator,** Department of Information Technology, NRI Institute of Technology for the esteemed guidance, moral support and invaluable advice provided by them for the success of the project.

With grateful thanks, I express my deep sense of gratitude and respect towards my internal guide, **Mr.A.Ravi Kiran**, **Assistant Professo**r for his valuable suggestions,guidance and encouragement throughout the project.

We are also thankful to all the staff members of Information Technology department who have cooperated in making our project a success. We would like to thank all our parents and friends who extended their help, encouragement and moral support either directly or indirectly in our project work.

Thank you for your valuable guidance and kind support.

Name: MUNI TEJASREE	**Regd No:** 21KN1A1243
Name: GONUGUNTA MAHESWARI	**Regd No:** 21KN1A1219
Name: POTLURI MAHESH	**Regd No:** 21KN1A1250

Organization Information:

ADVAITA GLOBAL IT SOLUTIONS was established with the goal of fostering the growth of a vibrant ecosystem that includes individuals, research institutes, commercial enterprises, and educational establishments. The International Institute of Technology and the Government International Institute of Digital Technologies depend on it as a primary partner. It does this by providing a platform called TAPTAP AI Driven employability, which helps students change their course toward their desired goals while also supporting human resources managers in recruiting the best individuals It is home to the most comprehensive network of centers of excellence in emerging technologies in all of India. It is common knowledge that we are one of the most successful companies in India that specializes in the production of web apps. Our goal is to create web applications that are not only reliable and productive, but also perfectly cater to the requirements of each of our customers in the most fruitful manner that is open to us.

Programs and opportunities:

This ground-up approach helps us deliver not only the solution to our clients but also add value to at the core which operates in five specific domains namely TapTap - AI Driven, Post Graduation Programs, Center of Excellence,Virtual Programming Labs and Happie Days - A social Networking site for the students. TapTap offers services in Campus Recruitment drives for the employers as well as college authorities. Recruiters can Conduct Customized Online Assessments secured with Best-in-class Proctoring and Schedule the end-to-end hiring process. Under each division, we further provide specific industry solutions on focused domains with cutting edge technologies. It emphasizes on building relationships with our clients by delivering projects on time and within budget**.**

ABSTRACT

DevOps is the portmanteau of development and operations. In recent times, the agile transformation was adopted in IT organizations for continuous integration principles in software development life cycle (SDLC) which has improved the efficiency of development in the project. With time being it has been realized that the optimization does not help in only continuous integration to make the software delivery process efficient. Unless all the modules in software delivery life-cycle are well designed, implemented and optimized. This is the problem with previous technologies and DevOps addresses it. This project explains the various phases of SDLC, business needs and ways to move from continuous integration to continuous delivery .

Learning Objectives/Internship Objectives

I. Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.

II. An objective for this position should emphasize the skills you already possess in the area and your interest in learning more.

III. Internships are utilized in a number of different career fields, including architecture, engineering,healthcare, economics, advertising and many more.

IV. Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.

Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants.

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

1st WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	06.04.2023	Thursday	Module 01-Why DevOps and its usage,Agile and DevOps
	08.04.2023	Saturday	Module 01 -DevOps Lifecycle, DevOps DeliveryPipeline

2nd WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	10.04.2023	Monday	Module 02-Git Introduction and Overview Of Version Control
	11.04.2023	Tuesday	Module 02-Git file workflow and Git commands
	12.04.2023	Wednesday	Module 03-Branching and Merging
	13.04.2023	Thursday	Module 03-Understanding and Using Build Tools
	14.04.2023	Friday	Module 04-Containerization basics using Docker
	15.04.2023	Saturday	Module 04-Containerization basics using Docker--Praticals

3rd WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	17.04.2023	Monday	Module 05-Continous integration using Jenkins
	18.04.2023	Tuesday	Module 05-Continous integration using Jenkins--Praticals
	19.04.2023	Wednesday	Module 06-Docker Commands and Use--cases
	20.04.2023	Thursday	Module 06-Docker Commands and Use--cases--practicals
	21.04.2023	Friday	Module 07-Introduction to Kubernetes

4th WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	24.04.2023	Monday	Module 07-Introduction to Kubernetes--Praticals
	25.04.2023	Tuesday	Module 08-Configuration Management using Ansible
	26.04.2023	Wednesday	Module 08-Configuration Management using Ansible—practicals
	27.04.2023	Thursday	Module 09-Introduction to Cloudwatch
	28.04.2023	Friday	Module 09-Monitoring different servers using cloudwatch
	29.04.2023	Saturday	Module 10-AWS DevOps Services

5th WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	01.05.2023	Monday	Module 10-AWS DevOps Services--Practicals
	06.05.2023	Saturday	Assessment-Internship(task will be assign to groups)

INDEX

CHAPTER-1

1. INTRODUCTION

1.1CONTEXT

Patrick Debois, who devised the name “DevOps” in 2009 and he is also called as “The Father of DevOps” devised the name “DevOps” in 2009.The word DevOps says itself that it formed by coalescing the two words “Development” and “Operations”. DevOps is the collaboration of development and deployment of software. DevOps is the portmanteau of development and operations. It is a software development method that escalates to the amalgamation between software development team and operations team. This is the time to change the old technology to new technology like DevOps. - “Time to stop wasting money, time to start delivering great software and building systems that scale and last” – Patrick Debois.

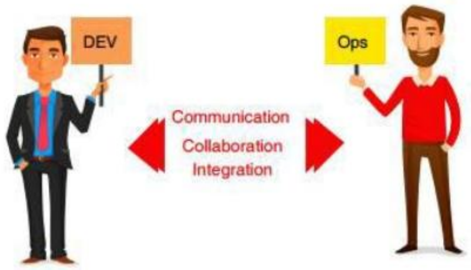


Figure1.1: understanding DevOps

The new software delivery procedure is adopted by the organizations since the market needs are changing continuously, rapid change in technology to deliver quickly. Customer waiting for months or 1 year for a version to be released and giving feedback after release cannot happen nowadays. Customers need continuous engagement with the project so that they can provide the feedback continuously. In order to face the challenges, the organization should be lean and follow the agile transformation in all phases of SDLC. Over the years the organizations have adopted the agile transformation for optimization, but the evolution takes place to change the technology to DevOps.It is important to keep all the phases in pace so that the software delivery lifecycle will not be delayed.

DevOps is the mechanism which bridges the gap between developer-operations and not only limited to developer-operations but also for the continuous development, continuous testing and continuous integration. DevOps main goal is to deliver the software rapidly with continuous development, continuous integration, continuous feedback and communication with development and operations team .

CHAPTER-2

2.RELATED WORK

2.1 WATERFALL MODEL

The first introduced process Model is the Waterfall model. It is also referred to as a traditional model or linear-sequential life cycle model. It is very simple to understand and use. In this Waterfall model, every phase must be completed before going to the next phase and there is no issue of overlapping of phases. It is the first SDLC approach model for software development.The waterfall model illustrates the software development process flow in the linear sequential flow. The waterfall model process development is divided into separate phases and will start only after completion of the previous phase.The phases of Waterfall development model are as follows 1. Requirement gathering and Analysis 2.System Design 3.Implementation 4.Testing 5. Deployment 6.Maintenance This model is the first model developed for the software development and ensures success, but the disadvantage is that each phase should be done only after completion of the previous phase it results in a long duration and also all the requirements are to be specified before starting the project.

2.2 AGILE MODEL

Agile software development model is the process model which is the combination of both iterative and incremental model. The agile model focuses on process adaptability and customer satisfaction by rapid development and delivery of software product. It breaks the complete product into small incremental builds and these builds are completed in iterations. These iterations last for one to three weeks and involve in cross-functional product development. The teams simultaneously work on these phases 1. Planning 2. Requirement Analysis 3. Design 4. Coding 5. Unit Testing 6. Acceptance Testing This agile model after completion of system testing for the product it will be deployed in the market. It helps to complete the project rapidly.It is also suitable for static or dynamic requirements, but it does not suitable for handling complex dependencies and a high risk of sustaining, maintaining and extending.

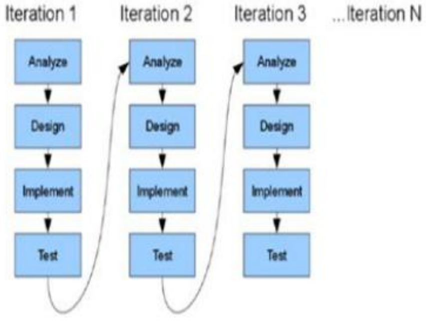


Figure2.2:Agile model

2.3 DEVOPS AND ITS IMPORTANCE

Patrick Debois, who is called as “The Father of DevOps” devised the name “DevOps” in 2009. DevOps (a portmanteau of development and operations) is a software development method that escalates to the amalgamation between software developers and information technology (IT) operation professionals.

Developers always want to deliver the changes in the product as soon as possible whereas the operation team want reliability and stability in the product. This situation was explained clearly in “wall of confusion” by Lee. This wall of confusion not only gives the mentalities of two teams but also the tools they practice. Development team uses some tools and Operations team uses different tools to perform the same task. DevOps bridge the gap between the development and operations for better and faster results.

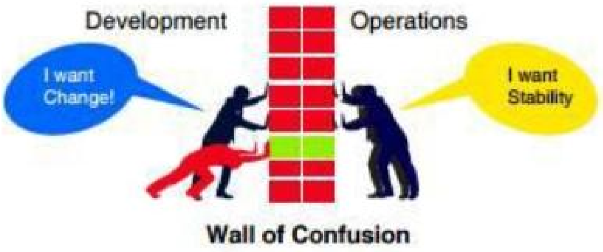


Figure 2.3.1:Wall of confusion without DevOps

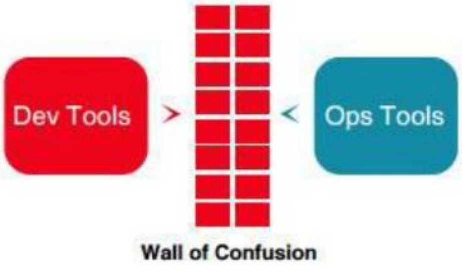


Figure 2.3.2: Wall of confusion with DevOps

2.4 WHAT DRIVE THE NEED OF DEVOPS?

I. Strong collaboration between development and operation teams.

II. Synchronized deployment across multiple platforms.

III. Pressure to release applications to meet customer requirements or to enter into the new market.

IV. Improving end user capability levels.

V. Vast usage of smart devices .

VI. Necessity to develop and deploy into cloud-based applications.

VII.Increasingly complex IT infrastructure.

VIII.Need to reduce the cost for IT industry.

2.5 RECOGNIZING BUSINESS VALUE OF DEVOPS

DevOps applies agile and lean principles in the complete software deployment process to enhance the speed of delivery of product or service from the initial release to the production release and to the feedback given by the client based on the release. DevOps return our investment in these three areas.

1. Enhanced Customer Experience Delivering an enhanced product for the customer leads to build loyalty and increase in market share. To deliver an enhanced product we need to continuously obtain and respond to the customer's feedback faster and perform required changes suggested by the customer.

2. Increased capacity to Innovate Lean thinking approaches are used in modern organizations to increase their innovation capacity. Their goals are to utilize the resources efficiently for other activities by reducing waste and rework. An example of lean thinking in organization is A-B testing in which large organizations asks a small group of users to test and rate two or more sets of software having different capabilities then the better capability set is picked up for the users and unsuccessful version is rolled back. This A-B testing is reliable only if efficient and automated mechanisms are adopted such as DevOps.

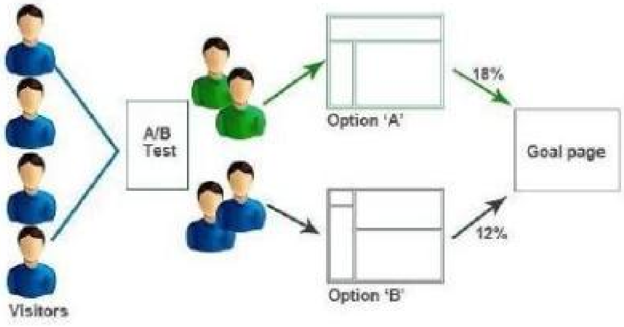


Figure2.5: A-B Testing

A-B testing is the comparison of two web pages to know which will perform fast and efficient. It is also called as split testing. We compare two web pages by showing the two variants (let's call them A and B) to similar visitors at the same time. The one which has more conversion rate is accepted. 3. Faster time to value: This involves in development of new culture and practices and automating the project leads to fast and reliable delivery process throughout the production phase. This DevOps can be worked as a business capability with the tools for release planning, predictability and success. The DevOps main goal is to deliver the value faster and in efficient way and the value definition changes with organization or with the project.

2.6 DEVOPS PHASES

I) Continuous Planning Business plans are already using agile methodologies to delivery quickly and change according to market conditions. It is better to have the checkpoints so that we can easily do the necessary changes given as feedback by customer.Dev/Test teams adapting to quick changes is not an easy task in business environments. DevOps allows us to prioritize the product backlogs and takes business perspective in to consideration. This is the continuous process of planning, executing, getting feedback from the customer, the cycle continues.

II) Continuous Integration Continuous Integration means dynamically integrating the changes made to the project to the team and not restricted to our local machine and validates the behavior of the code. Sharing with component teams but integrating beyond component boundaries at product integration level. Further the process optimization refers to automation as soon as the developer delivers the change build systems must detect the change and trigger a build taking sanity test and building repository. This must be a cyclic process across the development .

III) Continuous Deployment Continuous Deployment is the heart of the DevOps and acts as the Centre point to the complete software delivery optimization. Most of the surveys said that in many organizations the reason for the delay in software delivery is the operations. Hardware setting in the development build may vary from days to weeks. These deployment processes are inconsistent and manual. DevOps principles recommend the automation of deployment and hardware provision and cloud play a vital role in this field. DevOps proposes a concept called Infrastructure as a code (IAAC) which says that complete infrastructure provision should be maintained in source code repository.

IV) Continuous Testing Automation is the best option for continuous testing to test every test case. If any process need to do repeatedly for some constant time it is better to automate that process. They are humongous applications available in the market for do that type of testing process to meet the goal. There will be a chance of maximum to automate the manual testing process we need to evaluate on the same. Software delivery process must be able to execute the test suite automatically with the user intervention leading towards the goal reach easily. This kind of process not only makes testing process automate but also allows test cases to be carried out fast in production like system (deployment). Continuous Monitoring As discussed in all the above approaches we adopted, there is a chance to observe various parameters and react to them accordingly. The capability to test early and production like systems we can react to them in timely manner.

2.7 DELIVERY PIPELINE

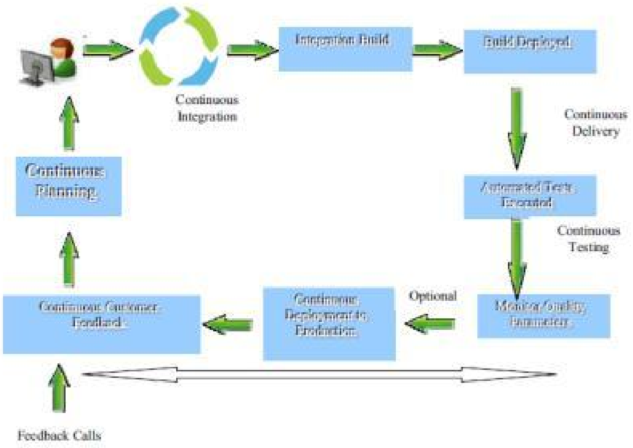


Figure 2.7:Delivery pipeline

DevOps approach delivery pipeline is shown in above figure. It can be compared with manufacturing unit delivery pipeline. Each build / release should undergo this cycle of dev - fvt - regression - stage - production - test phases by clearing all the quality parameters. With this automated pipeline there will be consistent releases.

CHAPTER-3

3.TOOLS USED

3.1 GIT SOFTWARE

Git is a version control system for tracking the changes in the local machine or the distributed system among the multiple people in the project. Its primary use is to Source code management (SCM) in software development but it is able to track the any number of changes done in the project. In distributed system its aim is to provide the speed, data.

3.2 JENKINS

Jenkins is a continuous integration (CI) server or a tool written in java language. It is open source software for download. The continuous integration services can be provided for software development which can be done via command line or web application server.

3.3 ANSIBLE

Ansible is the automation software that provides configuration management, software provisioning and application deployment. A platform was created by the Michael Dehaan, who is the author of the provisioning server application cobbler and a co-author of the fun framework for remote administration.

3.4 APACHE MAVEN

It is a Software Management tool works with Project Object Model (POM). It is used as a reporting, building, documenting from an informational central source point and can be used for building Java based projects.

3.5 DOCKER

It is a container management service. The keywords of Docker are **develop, ship** and **run** anywhere. The whole idea of Docker is for developers to easily develop applications, ship them into containers which can then be deployed anywhere.

Docker is an open source platform that enables developers to build, deploy, run, update and manage containers—standardized, executable components that combine application source code with the operating system (OS) libraries and dependencies required to run that code in any environment.

CHAPTER-4

4.SOFTWARE REQUIREMENTS

4.1 System configurations

The software requirement specification can produce at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, a representation of system behavior, and indication of performance and design constrain, appropriate validate criteria, and other information pertinent to requirements.

4.1.1 Software Requirements

- Operating system : Windows / Linux.
- Coding Language : Python /Shell script
- Platform : AWS.

4.1.2 Hardware Requirement

- System : Core i5 2.4 GHz.
- Hard Disk : 1TB.
- Ram : 8GB.

CHAPTER-5

5. LITERATURE REVIEW

5.1 Python

Python is a high-level, interpreted, general-purpose programming language. Its emphasizes code readability with the use of significant indentation.Python is dynamically-typed and garbage-collected. It supports multiple **Programming** including structured (particularly procedural), object- oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

5.2 Creating a Web Server

One of the most popular beginner-level DevOps projects is to create a Web Server.When a client requests content from a website, a web server saves it and sends it to them.It responds to client queries sent via the World Wide Web using the HTTP(Hypertext Transfer Protocol) and other protocols.For this project, the first step is to create a basic HTTP server.

5.3 Java Application with Jenkins

Jenkins is a build automation tool. It is mainly known for its flexibility and developers can use it to create almost any kind of software. For this project, you must generate a Jenkins build definition file and 'build' the java application. After constructing the application, the configuration of Jenkins can be executed to do a straightforward automated test. With the help of this project, you can learn how to build a java application in an archive and how to run it.

5.4 Project on Jenkins Remoting

Jenkins is DevOps automation tool that is mainly used for testing and building software. It is open-source and is written in Java. With the help of Jenkins software, developers can build such pipelines that follow CI/CD (Continuous Integration/Continuous Delivery) workflows. Jenkins' communication layer is implemented using Jenkins Remoting, an executable JAR and library. This project aims to improve Jenkins remoting. It requires knowledge of Java, message queues, and networking basics, which makes it an amazing project idea for networking enthusiasts.

5.5 Implementing DevOps Lifecycle with AWS

You’ll be storing the source code for the application in this project using AWS Developer Tools. The software will therefore be created, examined, and deployed automatically on AWS or in your on-premises environment. To create a continuous delivery pipeline that makes use of AWS Code Build or Code Deploy, you may start with the AWS Code Pipeline.

5.6 Build a Scalable Application

Through this application,you will gain valuable insight into application architecture and the reasons for its significance via this assignment.Your task is to create a scalable application by creating a robust foundation based on well-defined principles.

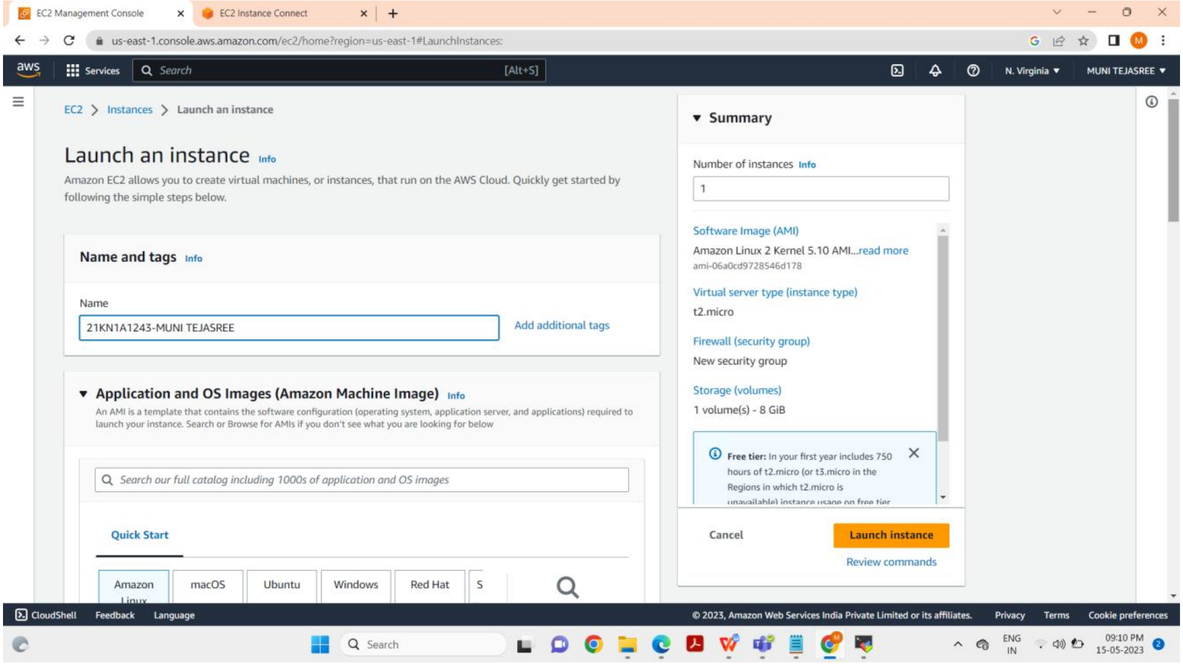
CHAPTER-6

6. Methodology

The methodology for the Create job and run in Jenkins to illustrate upstream and downstream is First of all, we can follow these steps:

Step1:First we have to create an AWS instance and launch it

- Create AWS instance



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: 21KN1A1243-MUNI TEJASREE

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more ami-06a0cd9728546d178

Virtual server type (instance type): t2.micro

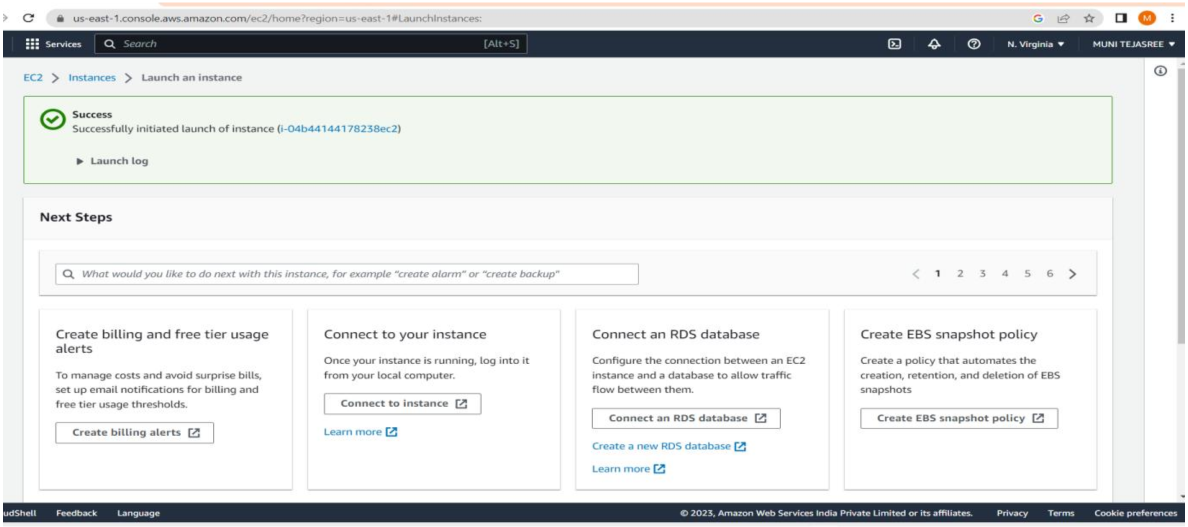
Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier

© 2023, Amazon Web Services India Private Limited or its affiliates.

- Launch the instance:



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-04b44144178238ec2)

Next Steps

Create billing and free tier usage alerts: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

Connect to your instance: Once your instance is running, log into it from your local computer.

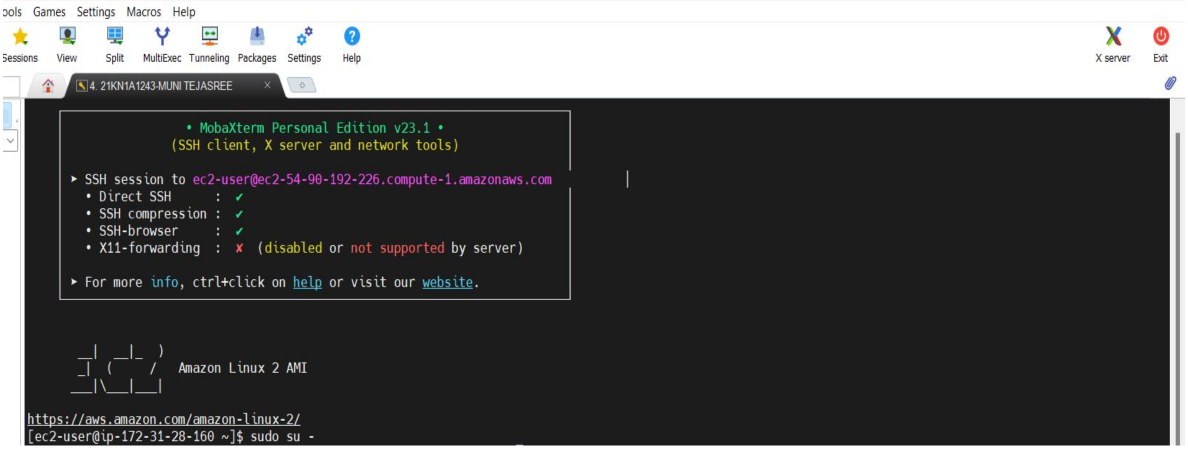
Connect an RDS database: Configure the connection between an EC2 instance and a database to allow traffic flow between them.

Create EBS snapshot policy: Create a policy that automates the creation, retention, and deletion of EBS snapshots

© 2023, Amazon Web Services India Private Limited or its affiliates.

After your instance is launched, you can connect to it using various methods, such as SSH or Remote Desktop.

- You can also manage your instance using the EC2 dashboard, including starting and stopping the instance, modifying its settings, and terminating it when you no longer need it.
- We have to enter into the root level to execute our requirements.



```
                 • MobaXterm Personal Edition v23.1 •
               (SSH client, X server and network tools)

 ➤ SSH session to ec2-user@ec2-54-90-192-226.compute-1.amazonaws.com
   • Direct SSH      :  ✓
   • SSH compression :  ✓
   • SSH-browser     :  ✓
   • X11-forwarding  :  ✗  (disabled or not supported by server)

 ➤ For more info, ctrl+click on help or visit our website.


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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-28-160 ~]$ sudo su -
```

Step-2 : Installing Java & Maven

Java and Maven are commonly installed on Jenkins because they are essential tools for building and running Java-based applications.

Installing Java:

Java is a programming language and computing platform that is widely used in software development.Many modern applications, including web applications and mobile apps,are written in Java.

Jenkins, being an automation server, is often used for building and testing Java applications. In order to do this, it requires the Java Runtime Environment (JRE) or the Java Development Kit (JDK) to be installed on the Jenkins server. This is because Jenkins uses Java-based tools like Apache Ant, Cradle, and Maven to perform build and test tasks.

Installing java on AWS we have to use the following steps:

- Amazon Linux Extras is a feature of Amazon Linux that provides additional software packages that are not included in the base Amazon Linux AMI. These packages are maintained by Amazon and are available for use on Amazon Linux instances.

The command is::**“amazon-Linux-extras enable java-openjdk11”**

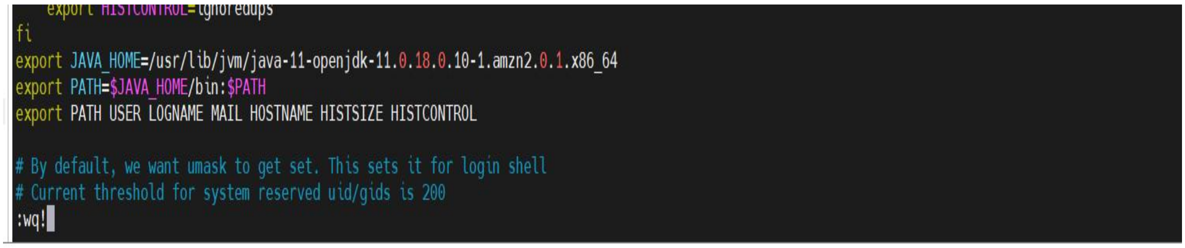
- After that we have clean the previous data on it and install java. We have to use the following commands:

commands are ::I) **“yum clean metadata”**

II) **“yum install java-11-openjdk”**

- Set class path for java

command is::**“vim /etc/profile”**

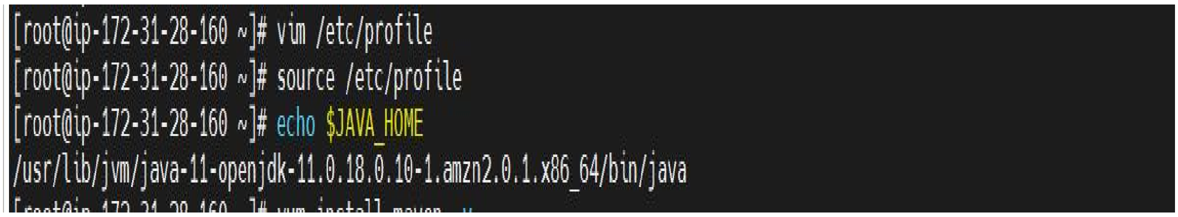


```
    export HISTCONTROL=ignoredups
fi
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-11.0.18.0.10-1.amzn2.0.1.x86_64
export PATH=$JAVA_HOME/bin:$PATH
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
:wq!
```

- After that we have to update the class file using these commands.

command is:: **“source /etc. /profile”**

- Finally, we have to check whether java is installed in AWS or not. For that we have to use the command :: **“echo $JAVA_HOME”**



```
[root@ip-172-31-28-160 ~]# vim /etc/profile
[root@ip-172-31-28-160 ~]# source /etc/profile
[root@ip-172-31-28-160 ~]# echo $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-11.0.18.0.10-1.amzn2.0.1.x86_64/bin/java
[root@ip-172-31-28-160 ~]# yum install maven -y
```

NEXT WE HAVE TO INSTALL A MAVEN

Maven is a popular build automation tool that is widely used in the Java community. It simplifies the process of building and managing Java-based projects by providing a standardized way to manage dependencies, build artifacts, and run tests. Maven is often used in conjunction with Jenkins to automate the build process. By using a Maven plugin, Jenkins can automatically download dependencies, compile code, run tests, and package the application into a deployable artifact. This makes it easy to automate the entire build and deploy process, saving developers time and reducing the chance of human error.

- We have to install the maven using this command:

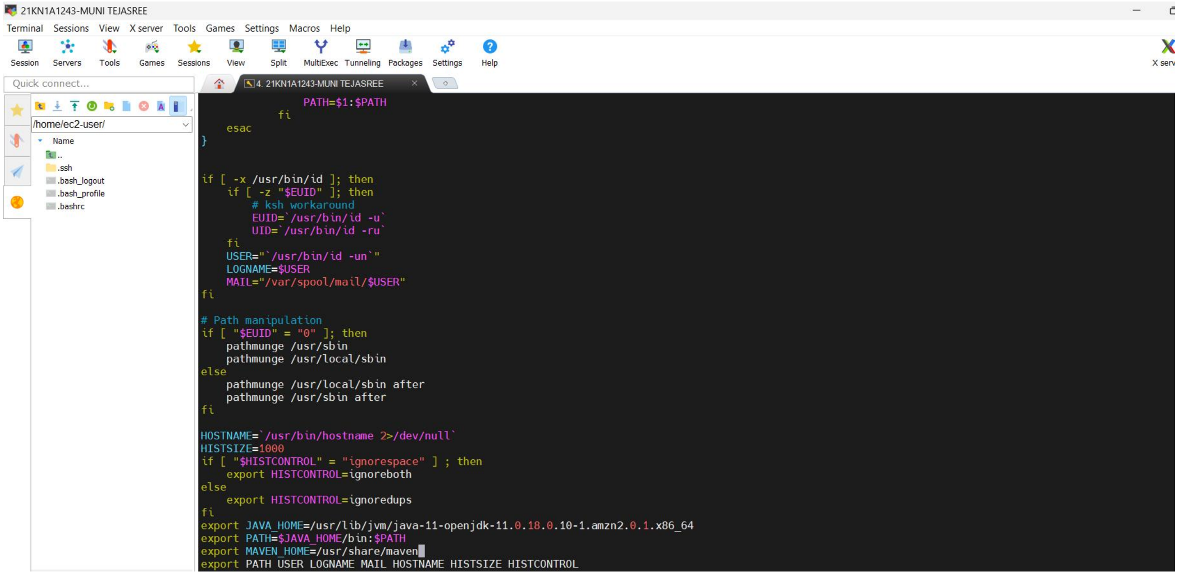
  command is:: **“yum install maven -y”**

- Next, check the version of the maven by using this command:

  command is::**“maven –version”**

- set class path for java

  command is:: **“vim /etc/profile”**



21KN1A1243-MUNI TEJASREE
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X serv
Quick connect...
4. 21KN1A1243-MUNI TEJASREE
/home/ec2-user/
Name
..
.ssh
.bash_logout
.bash_profile
.bashrc

```
                PATH=$1:$PATH
            fi
    esac
}


if [ -x /usr/bin/id ]; then
    if [ -z "$EUID" ]; then
        # ksh workaround
        EUID=`/usr/bin/id -u`
        UID=`/usr/bin/id -ru`
    fi
    USER="`/usr/bin/id -un`"
    LOGNAME=$USER
    MAIL="/var/spool/mail/$USER"
fi

# Path manipulation
if [ "$EUID" = "0" ]; then
    pathmunge /usr/sbin
    pathmunge /usr/local/sbin
else
    pathmunge /usr/local/sbin after
    pathmunge /usr/sbin after
fi

HOSTNAME=`/usr/bin/hostname 2>/dev/null`
HISTSIZE=1000
if [ "$HISTCONTROL" = "ignorespace" ] ; then
    export HISTCONTROL=ignoreboth
else
    export HISTCONTROL=ignoredups
fi
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-11.0.18.0.10-1.amzn2.0.1.x86_64
export PATH=$JAVA_HOME/bin:$PATH
export MAVEN_HOME=/usr/share/maven
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL
```

- After that, we have to update the class file, so we write this command

“source /etc./profile”

- Finally, we have to check whether the maven is installed in the AWS or not. For that reason, we have to use the command: **“echo $MAVEN_HOME”**

Step-3: Install the Jenkins on AWS

- Firstly, we have to get the Jenkins from the repo and use the key for that:

“sudo wget -O /etc/yum.repos.d/Jenkins.repo https://pkg.Jenkins.io/redhat-stable/Jenkins.repo”

“sudo rpm --import https://pkg.Jenkins.io/redhat-stable/Jenkins.io-2023.key”

- Then we have to install the Jenkins

  “sudo yum install Jenkins -y”

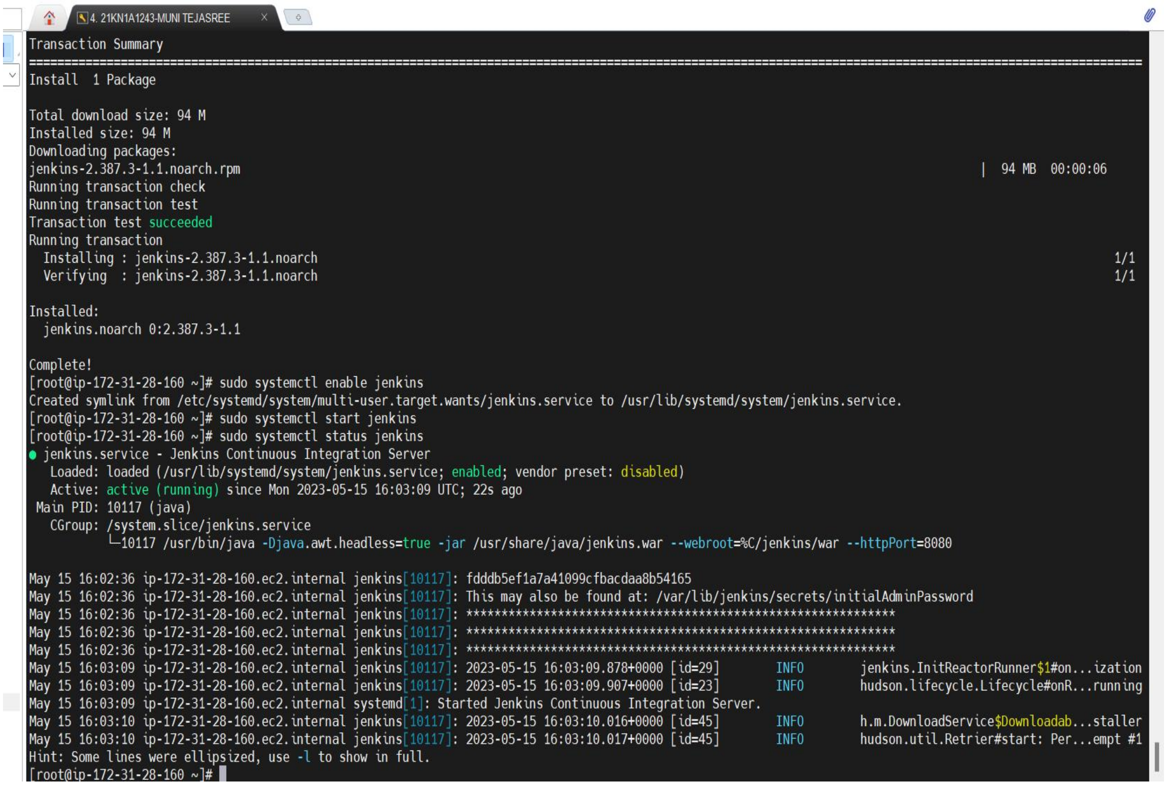
- Next we have to enable the Jenkins and start the Jenkins:

  “sudo systemctl enable Jenkins”

  “sudo systemctl start Jenkins”

- Finally check the status of the Jenkins:

  “sudo systemctl status Jenkins”



```
Transaction Summary
================================================================================================================================================
Install  1 Package

Total download size: 94 M
Installed size: 94 M
Downloading packages:
jenkins-2.387.3-1.1.noarch.rpm                                                                                           |  94 MB  00:00:06
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.387.3-1.1.noarch                                                                                                 1/1
  Verifying  : jenkins-2.387.3-1.1.noarch                                                                                                 1/1

Installed:
  jenkins.noarch 0:2.387.3-1.1

Complete!
[root@ip-172-31-28-160 ~]# sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[root@ip-172-31-28-160 ~]# sudo systemctl start jenkins
[root@ip-172-31-28-160 ~]# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2023-05-15 16:03:09 UTC; 22s ago
 Main PID: 10117 (java)
   CGroup: /system.slice/jenkins.service
           └─10117 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

May 15 16:02:36 ip-172-31-28-160.ec2.internal jenkins[10117]: fdddb5ef1a7a41099cfbacdaa8b54165
May 15 16:02:36 ip-172-31-28-160.ec2.internal jenkins[10117]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
May 15 16:02:36 ip-172-31-28-160.ec2.internal jenkins[10117]: *************************************************************
May 15 16:02:36 ip-172-31-28-160.ec2.internal jenkins[10117]: *************************************************************
May 15 16:02:36 ip-172-31-28-160.ec2.internal jenkins[10117]: *************************************************************
May 15 16:03:09 ip-172-31-28-160.ec2.internal jenkins[10117]: 2023-05-15 16:03:09.878+0000 [id=29]       INFO       jenkins.InitReactorRunner$1#on...ization
May 15 16:03:09 ip-172-31-28-160.ec2.internal jenkins[10117]: 2023-05-15 16:03:09.907+0000 [id=23]       INFO       hudson.lifecycle.Lifecycle#onR...running
May 15 16:03:09 ip-172-31-28-160.ec2.internal systemd[1]: Started Jenkins Continuous Integration Server.
May 15 16:03:10 ip-172-31-28-160.ec2.internal jenkins[10117]: 2023-05-15 16:03:10.016+0000 [id=45]       INFO       h.m.DownloadService$Downloadab...staller
May 15 16:03:10 ip-172-31-28-160.ec2.internal jenkins[10117]: 2023-05-15 16:03:10.017+0000 [id=45]       INFO       hudson.util.Retrier#start: Per...empt #1
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-28-160 ~]#
```

Jenkins was successfully installed.

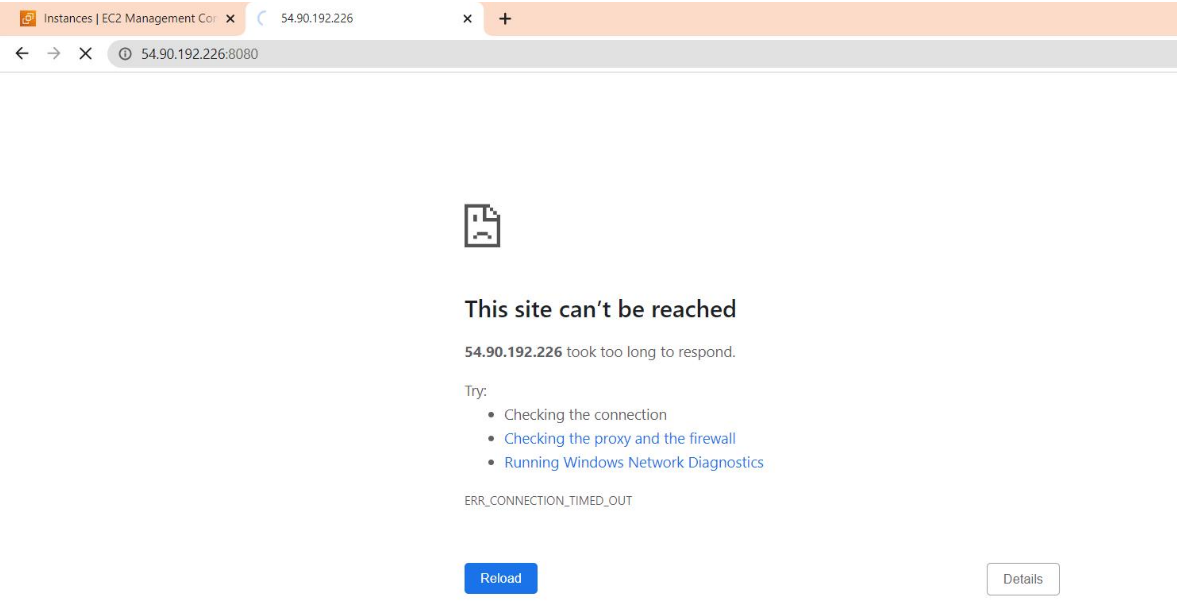
This methodology is useful to install Jenkins in AWS service.

CHAPTER-7

7.CODING

Now we have to create jobs and run in Jenkins . First step is we have to copy the public IPaddress for our AWS instance and paste it into the search browser followed by port 8080 **“54.90.192.226:8080”**

It did not come because AWS follows security .



This site can’t be reached

54.90.192.226 took too long to respond.

Try:

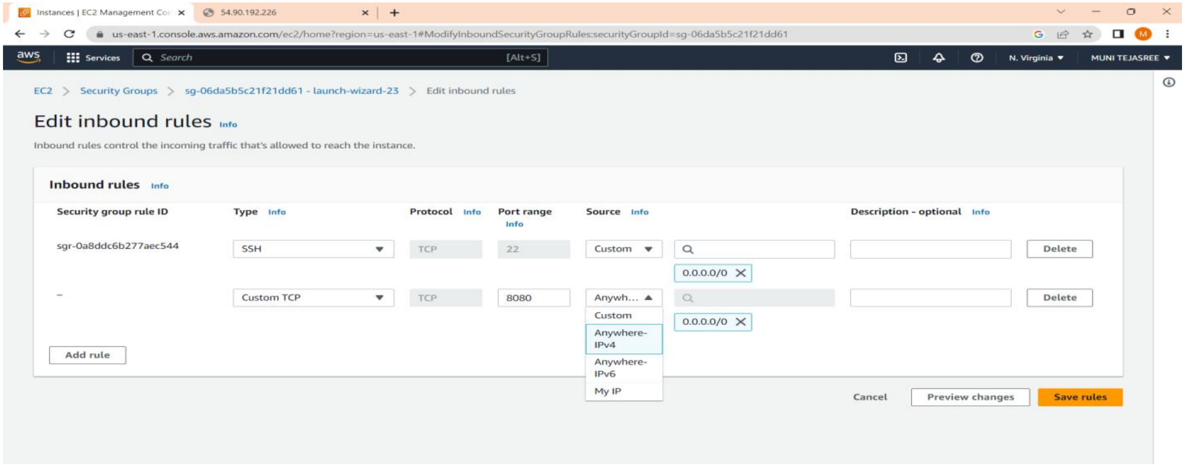
- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

ERR_CONNECTION_TIMED_OUT

Now we have to allow access to other processes . We follow some steps:

Step 1: Goto security option in your instance,click on security groups,then click to the edit inbound Rules then add the rule.

Step 2: click on save rules.



Instances | EC2 Management Cor
54.90.192.226
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-06da5b5c21f21dd61
aws
Services
Search
[Alt+S]
N. Virginia
MUNI TEJASREE
EC2 > Security Groups > sg-06da5b5c21f21dd61 - launch-wizard-23 > Edit inbound rules

Edit inbound rules Info

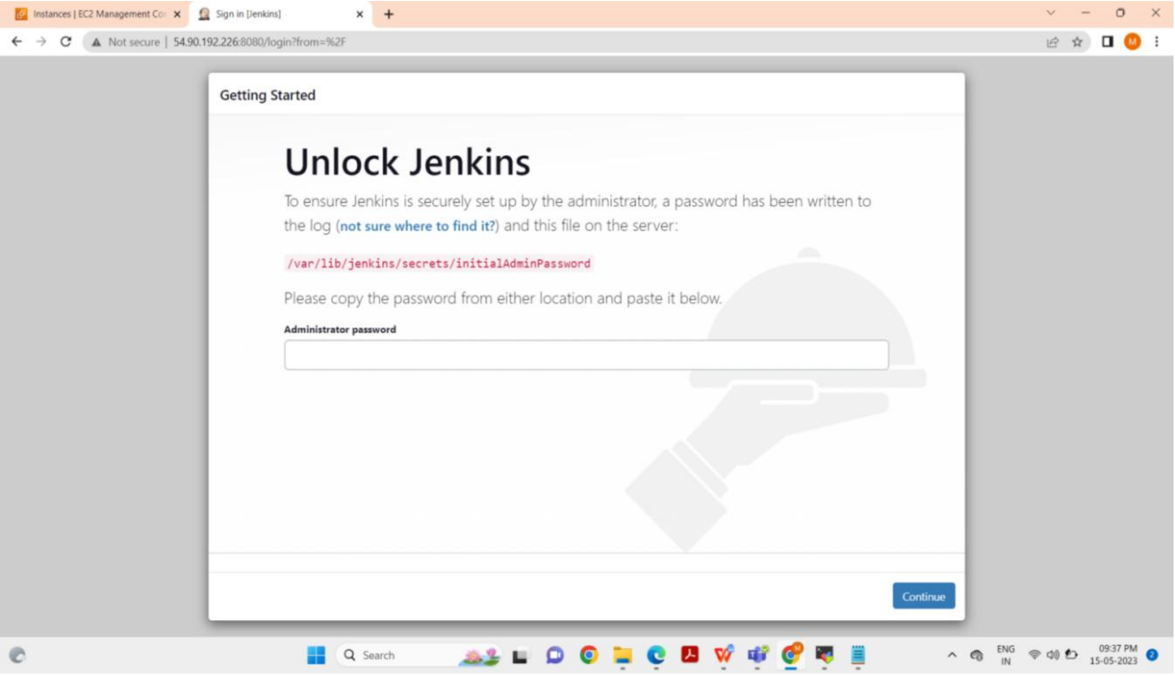
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0a8ddc6b277aec544	SSH	TCP	22	Custom	0.0.0.0/0		Delete
–	Custom TCP	TCP	8080	Anywh...	0.0.0.0/0		Delete

Custom
Anywhere-IPv4
Anywhere-IPv6
My IP
Add rule
Cancel
Preview changes
Save rules

Step 3: After clicking save rules then you have to refresh the web browser. Jenkins page is opened like this:



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

/var/lib/jenkins/secrets/initialAdminPassword

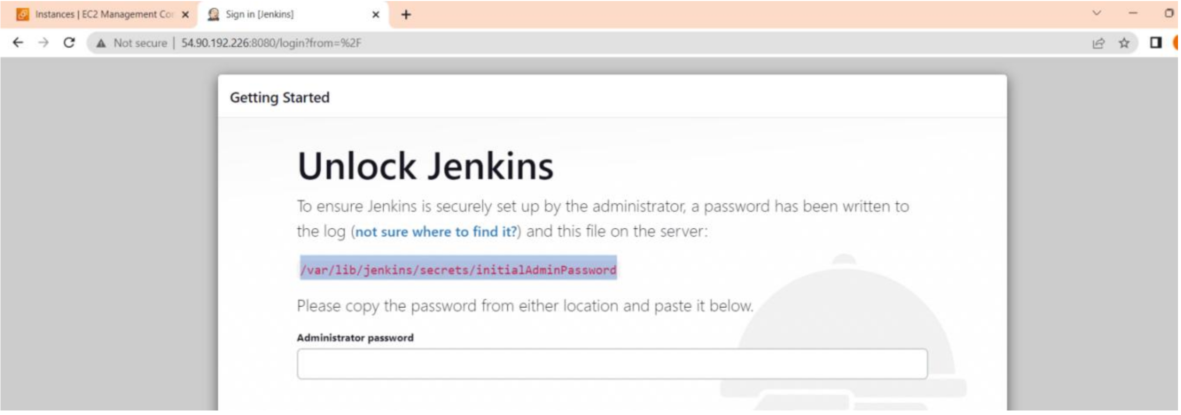
Please copy the password from either location and paste it below.

Administrator password

Instances | EC2 Management Co
Sign in [Jenkins]
Not secure | 54.90.192.226:8080/login?from=%2F
Continue

- Firstly, we have to login the Jenkins

Copy the path that is **/var/lib/Jenkins/secrets/initialAdminPassword**



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

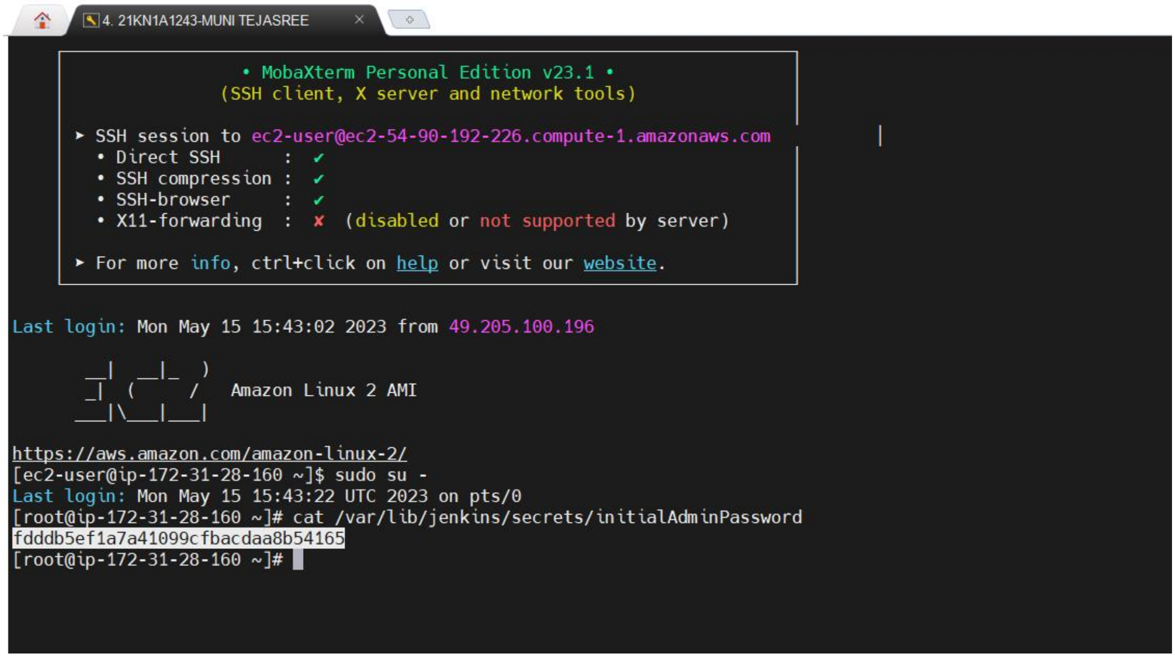
/var/lib/jenkins/secrets/initialAdminPassword

Please copy the password from either location and paste it below.

Administrator password

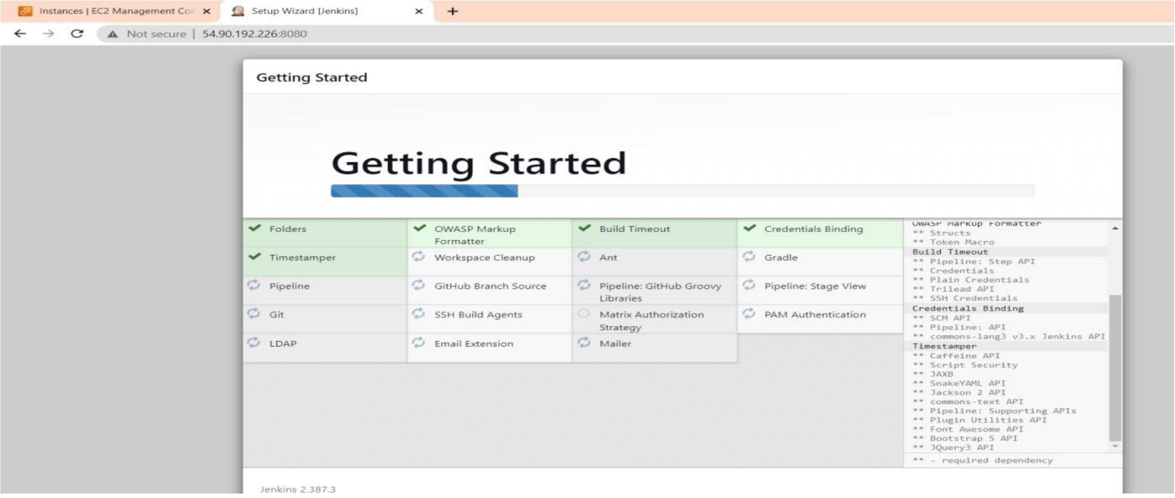
Instances | EC2 Management Co
Sign in [Jenkins]
Not secure | 54.90.192.226:8080/login?from=%2F

And paste it into the mobaxtream using cat command and it provide password copy the password like this:



```
                • MobaXterm Personal Edition v23.1 •
              (SSH client, X server and network tools)

 ➤ SSH session to ec2-user@ec2-54-90-192-226.compute-1.amazonaws.com
   • Direct SSH      :  ✔
   • SSH compression :  ✔
   • SSH-browser     :  ✔
   • X11-forwarding  :  ✘  (disabled or not supported by server)

 ➤ For more info, ctrl+click on help or visit our website.

Last login: Mon May 15 15:43:02 2023 from 49.205.100.196

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-28-160 ~]$ sudo su -
Last login: Mon May 15 15:43:22 UTC 2023 on pts/0
[root@ip-172-31-28-160 ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
fdddb5ef1a7a41099cfbacdaa8b54165
[root@ip-172-31-28-160 ~]#
```

- Paste the password into the administrator password ,then click on the continue button.
- After that, select install plugin .



Instances | EC2 Management Co
Setup Wizard [Jenkins]
Not secure | 54.90.192.226:8080

Getting Started

Getting Started

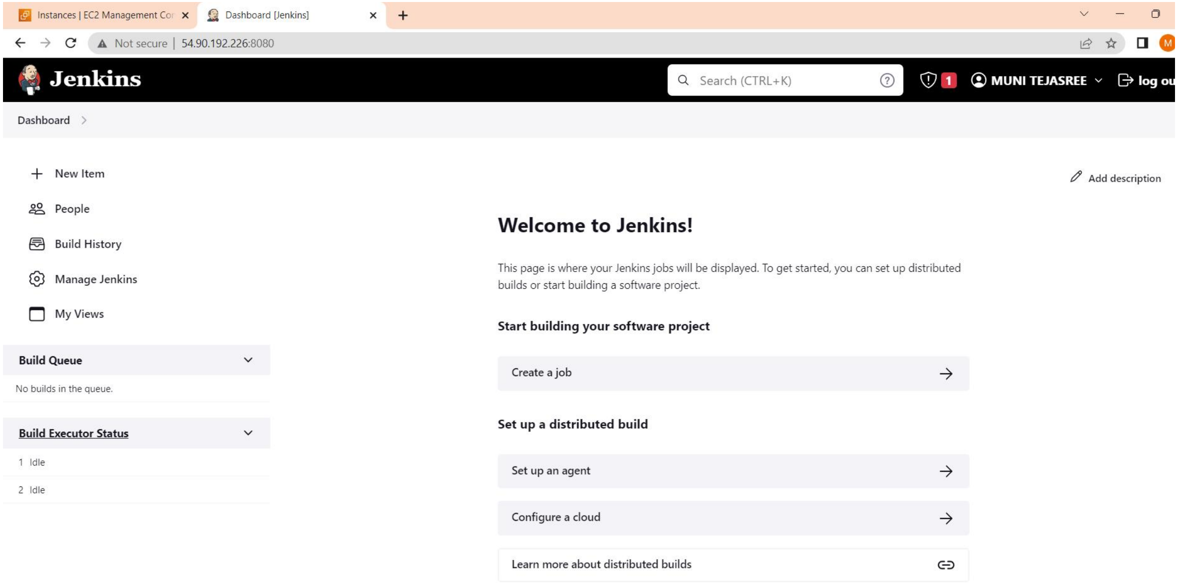
✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	○ Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

```
OWASP Markup Formatter
** Structs
** Token Macro
Build Timeout
** Pipeline: Step API
** Credentials
** Plain Credentials
** Trilead API
** SSH Credentials
Credentials Binding
** SCM API
** Pipeline: API
** commons-lang3 v3.x Jenkins API
Timestamper
** Caffeine API
** Script Security
** JAXB
** SnakeYAML API
** Jackson 2 API
** commons-text API
** Pipeline: Supporting APIs
** Plugin Utilities API
** Font Awesome API
** Bootstrap 5 API
** JQuery3 API
** - required dependency
```

Jenkins 2.387.3

- Next we have to fill in some information and start the Jenkins:
- Start by creating the upstream job, which will trigger the downstream job(s). In Jenkins, click on "New Item" to create a new job.
- Create a new job using the "New Item" button in Jenkins.

Step 4: Now we have to create 4 jobs in Jenkins to give names to the each and every job.



No builds in the queue.

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

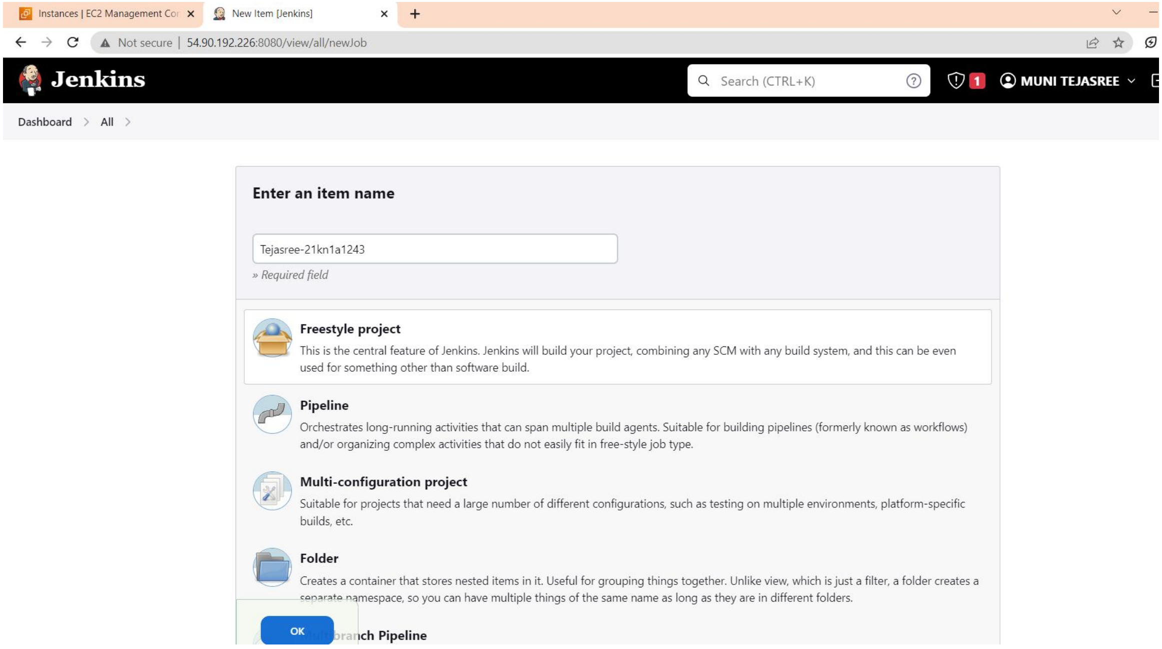
The 4 jobs are named as::
1) Tejasree-21kn1a1243

2) Maheswari-21kn1a1219

3) Mahesh-21kn1a1250

4) 4.Teja

Step 5: To create a Tejasree-21kn1a1243 job in Jenkins as follows below.



Enter an item name

Tejasree-21kn1a1243

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

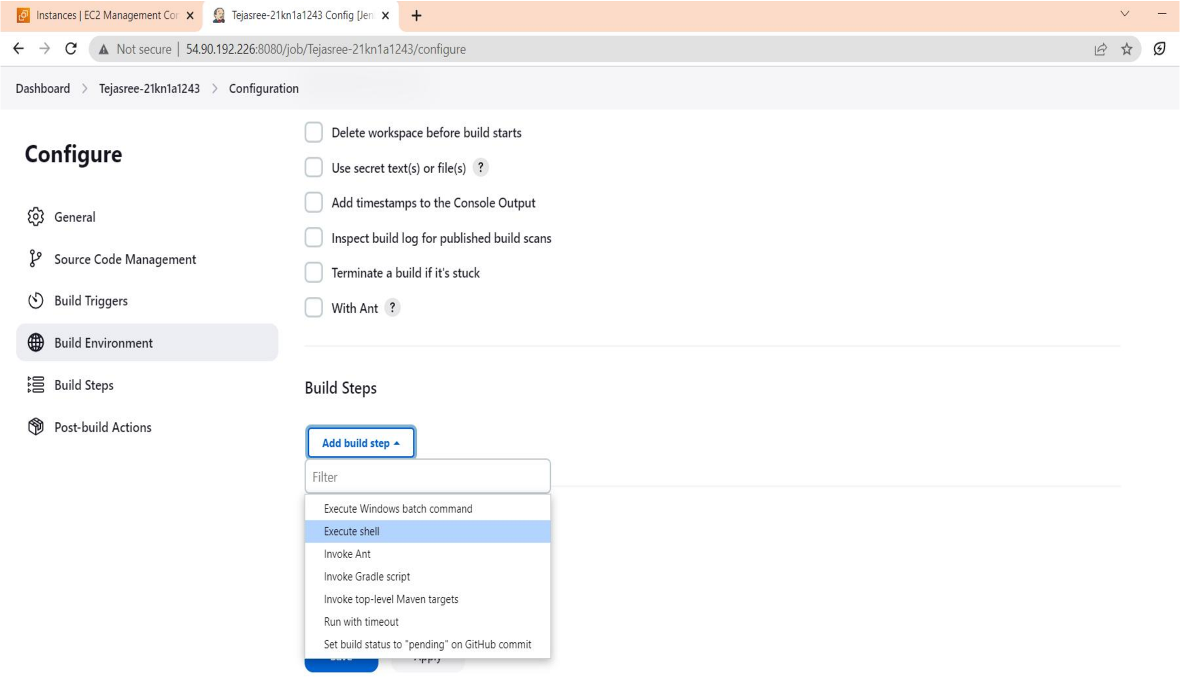
Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

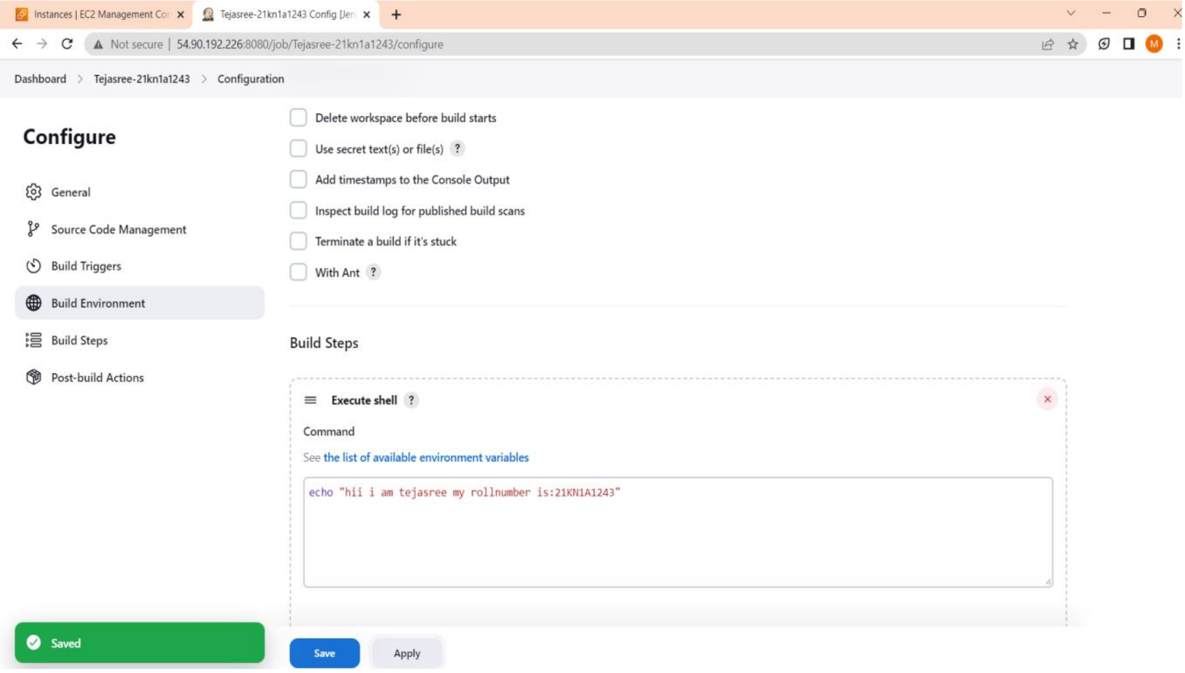
Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Step 6: Go to the build steps, select the execution shell and write a command on it. echo keyword is used to display the output. like printf as c language.

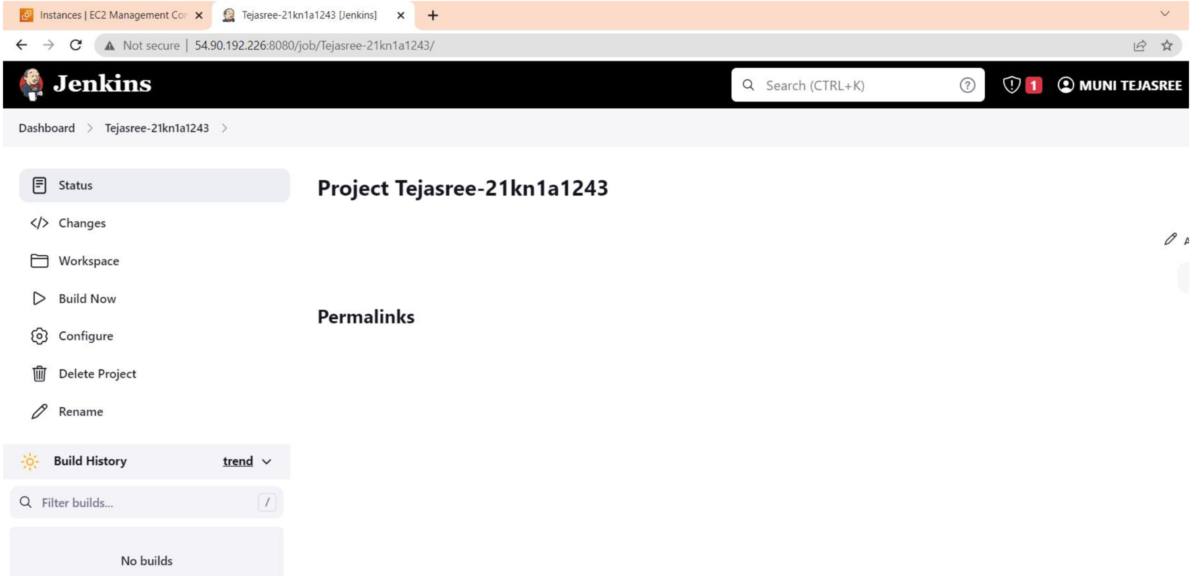

Instances | EC2 Management Cor
Tejasree-21kn1a1243 Config [Jen
Not secure | 54.90.192.226:8080/job/Tejasree-21kn1a1243/configure
Dashboard > Tejasree-21kn1a1243 > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
□ Delete workspace before build starts
□ Use secret text(s) or file(s) ?
□ Add timestamps to the Console Output
□ Inspect build log for published build scans
□ Terminate a build if it's stuck
□ With Ant ?
Build Steps
Add build step
Filter
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Run with timeout
Set build status to "pending" on GitHub commit

Write command on it.


Instances | EC2 Management Cor
Tejasree-21kn1a1243 Config [Jen
Not secure | 54.90.192.226:8080/job/Tejasree-21kn1a1243/configure
Dashboard > Tejasree-21kn1a1243 > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
□ Delete workspace before build starts
□ Use secret text(s) or file(s) ?
□ Add timestamps to the Console Output
□ Inspect build log for published build scans
□ Terminate a build if it's stuck
□ With Ant ?
Build Steps
Execute shell ?
Command
See the list of available environment variables
Saved
Save
Apply

```
echo "hii i am tejasree my rollnumber is:21KN1A1243"
```

Step 7: Click on save button, it saves the command.After saving the page it shows like this:



Instances | EC2 Management Co × Tejasree-21kn1a1243 [Jenkins] ×

Not secure | 54.90.192.226:8080/job/Tejasree-21kn1a1243/

Jenkins Search (CTRL+K) MUNI TEJASREE

Dashboard > Tejasree-21kn1a1243 >

Status
Changes
Workspace
Build Now
Configure
Delete Project
Rename

Build History trend

Filter builds...

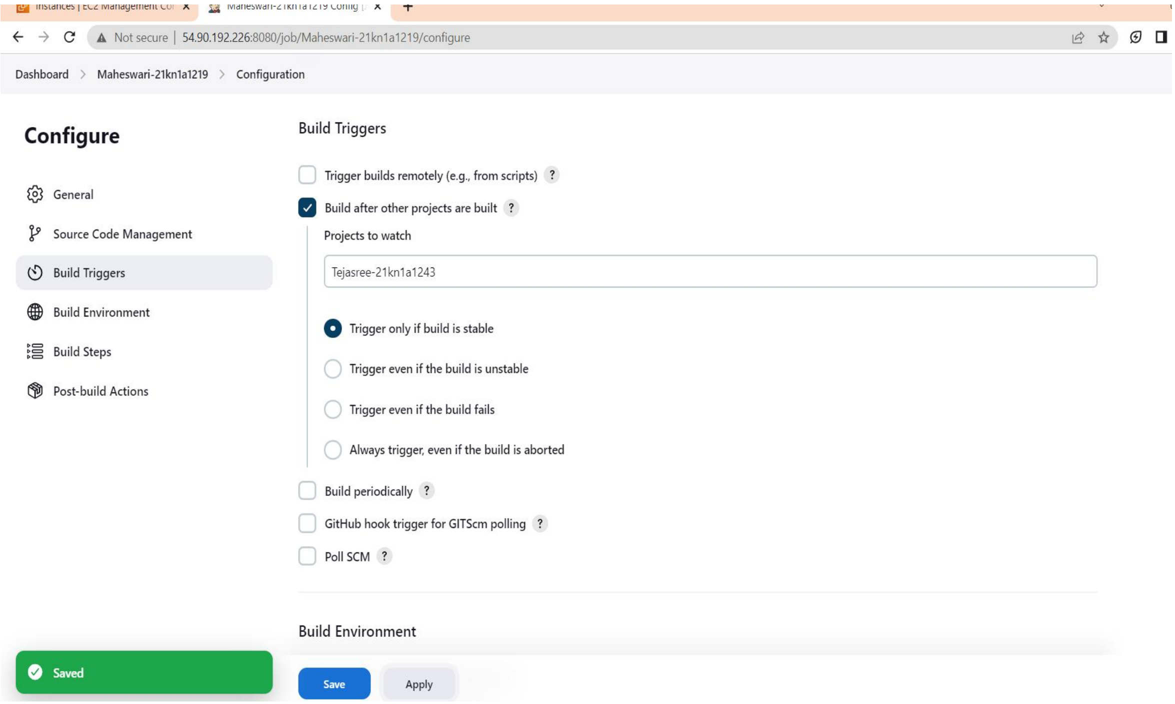
No builds

Project Tejasree-21kn1a1243

Permalinks

Step 8: Now create Maheswari-21kn1a1219 job and we have to go to the build trigger then select <u>Build after other projects are built?</u>

In that option type 1st job name in to the maheswari-21kn1a1219 job as like below:



Not secure | 54.90.192.226:8080/job/Maheswari-21kn1a1219/configure

Dashboard > Maheswari-21kn1a1219 > Configuration

Configure

General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions

Build Triggers

□ Trigger builds remotely (e.g., from scripts) ?

☑ Build after other projects are built ?

Projects to watch

Tejasree-21kn1a1243

● Trigger only if build is stable

○ Trigger even if the build is unstable

○ Trigger even if the build fails

○ Always trigger, even if the build is aborted

□ Build periodically ?

□ GitHub hook trigger for GITScm polling ?

□ Poll SCM ?

Build Environment

Saved

Save Apply

Step 9: Then write a command on execute shell and click on the save button

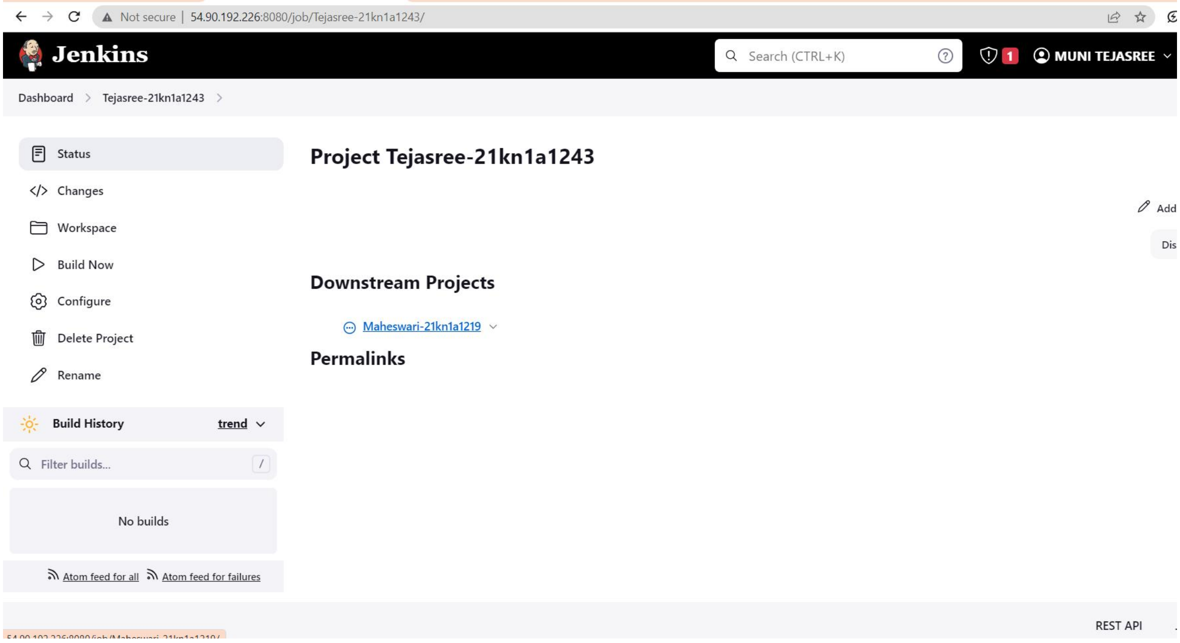
Step 10: Next create Mahesh-21kn1a1250 job it is 3rd job ,like same as goto build triggers write 2nd job name into the 3rd job ,then write a command into the execute shell and save it. **Step 11:**Next create 4.Teja job this is 4th job and go to build triggers, write 3rd job name into the 4th job , then write a command into the execute shell and save it.

This way, we have to create our jobs and add triggers for upstream and downstream.This setup demonstrates the basic idea for creating jobs in Jenkins with upstream and downstream relationships.

CHAPTER-8

8.RESULT

Now check the upstream and downstream of the jobs .Goto job1 that is Tejasree-21kn1a1243 job see the upstream and downstream.



Not secure | 54.90.192.226:8080/job/Tejasree-21kn1a1243/

Jenkins

Dashboard > Tejasree-21kn1a1243 >

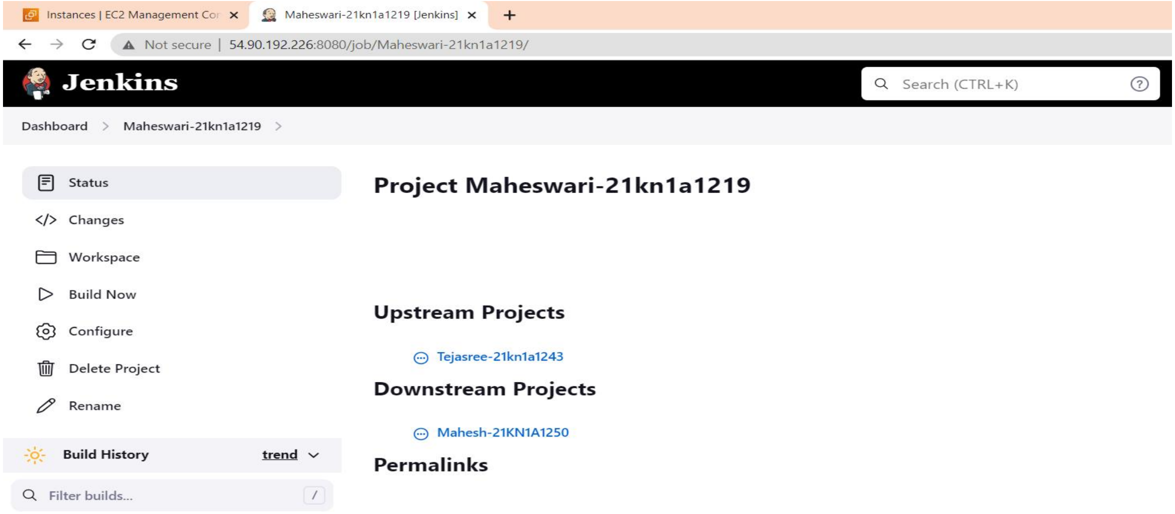
Project Tejasree-21kn1a1243

Downstream Projects

Maheswari-21kn1a1219

Permalinks

Screen 8.1:upstream & downstream for job 1



Instances | EC2 Management Cor | Maheswari-21kn1a1219 [Jenkins]

Not secure | 54.90.192.226:8080/job/Maheswari-21kn1a1219/

Jenkins

Dashboard > Maheswari-21kn1a1219 >

Project Maheswari-21kn1a1219

Upstream Projects

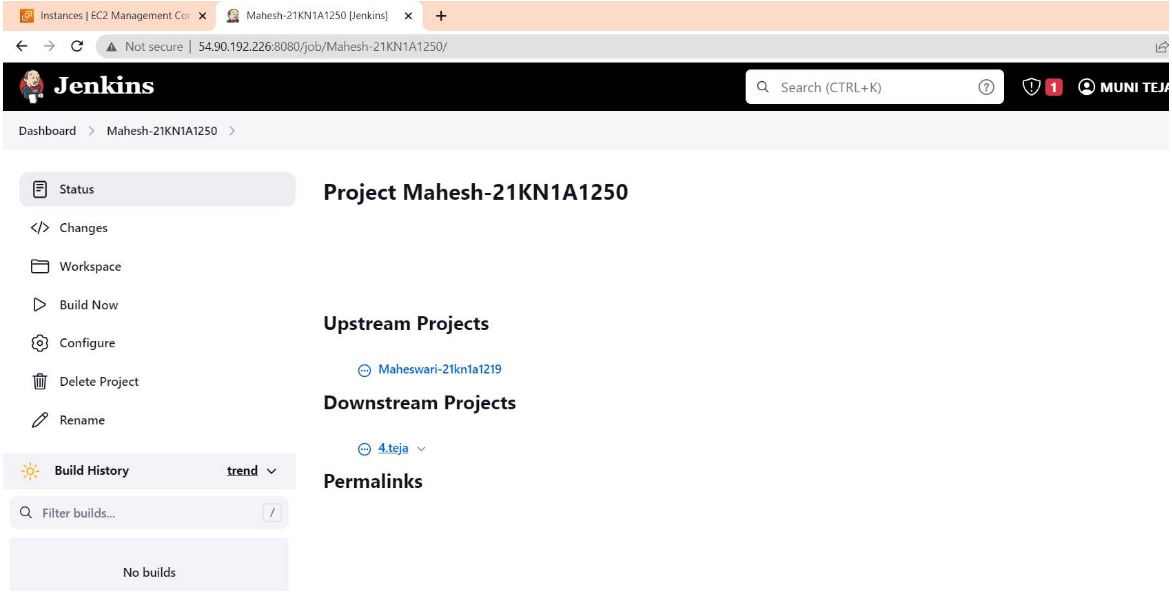
Tejasree-21kn1a1243

Downstream Projects

Mahesh-21KN1A1250

Permalinks

.Screen 8.2: upstream and downstream for job2



Project Mahesh-21KN1A1250

Upstream Projects

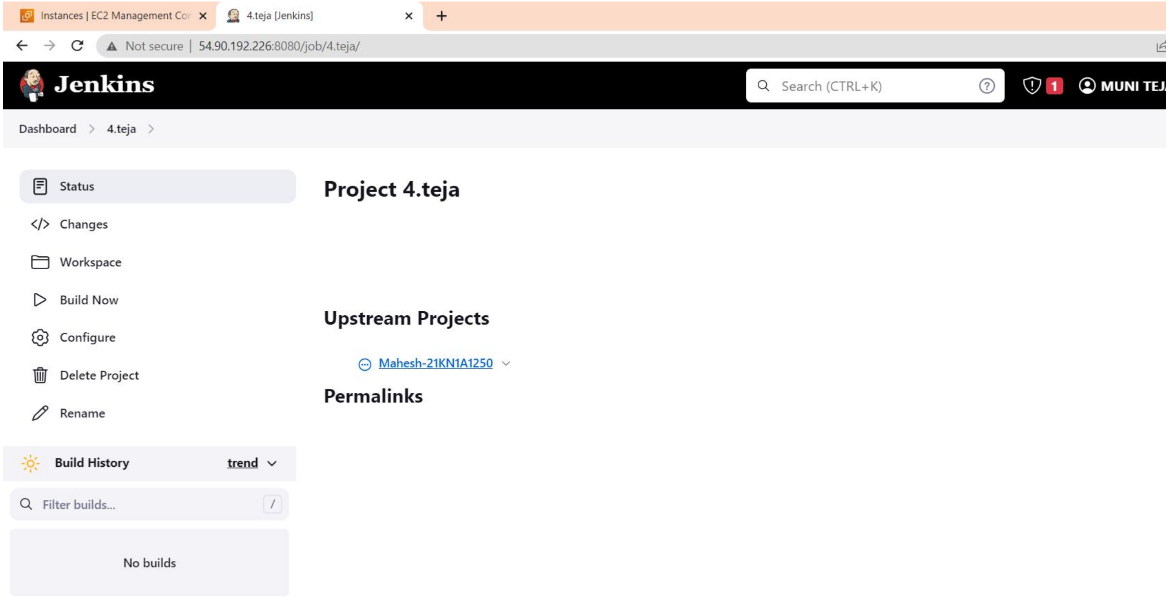
Maheswari-21kn1a1219

Downstream Projects

4.teja

Permalinks

Screen 8.3: upstream and downstream jobs for job3.



Project 4.teja

Upstream Projects

Mahesh-21KN1A1250

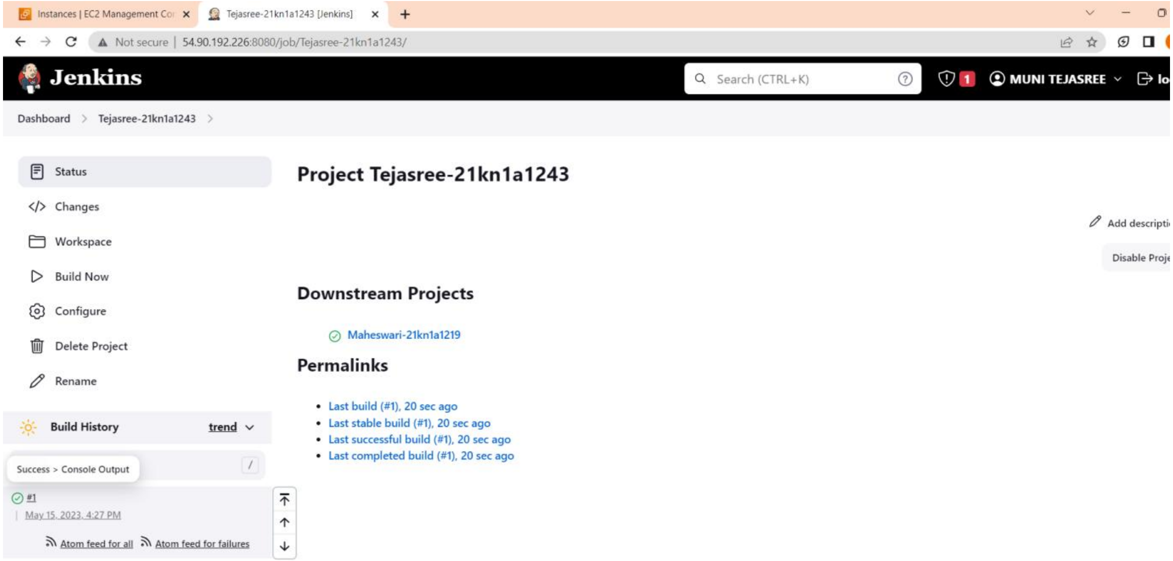
Permalinks

Screen 8.4 :upstream and downstream jobs for job4

1)The job4 contains upstream job that is mahesh-21kn1a1250, it does not consist downstream because we create only 4 jobs . The 4th job does not consist of downstream job.

2)If you want to run the job, click on the build now button . Jenkins will execute the upstream job(s) first and then trigger the downstream job(s) based on the defined relationships.

Let us build job1



Instances | EC2 Management Co × Tejasree-21kn1a1243 [Jenkins] ×

Not secure | 54.90.192.226:8080/job/Tejasree-21kn1a1243/

Jenkins — Search (CTRL+K) — MUNI TEJASREE

Dashboard > Tejasree-21kn1a1243 >

- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Project
- Rename

Build History — trend

Success > Console Output

#1 May 15, 2023, 4:27 PM

Atom feed for all — Atom feed for failures

Project Tejasree-21kn1a1243

Add descripti

Disable Proje

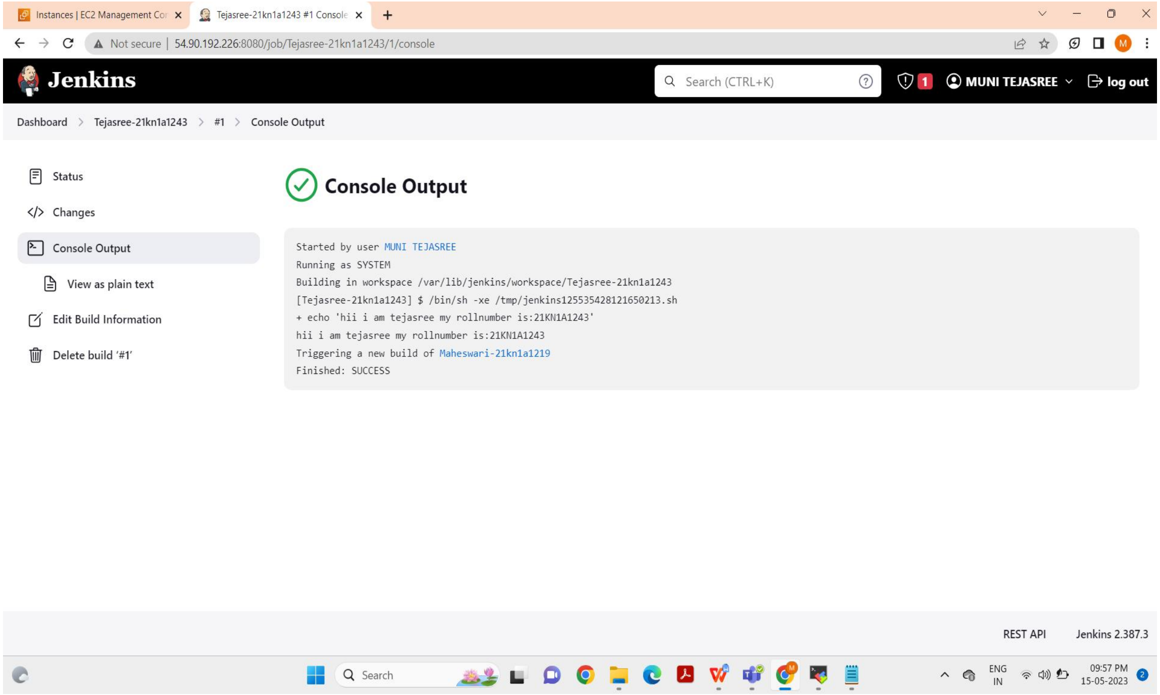
Downstream Projects

Maheswari-21kn1a1219

Permalinks

- Last build (#1), 20 sec ago
- Last stable build (#1), 20 sec ago
- Last successful build (#1), 20 sec ago
- Last completed build (#1), 20 sec ago

Screen 8.5:build 1st job Tejasree-21kn1a1243



Instances | EC2 Management Co × Tejasree-21kn1a1243 #1 Console ×

Not secure | 54.90.192.226:8080/job/Tejasree-21kn1a1243/1/console

Jenkins — Search (CTRL+K) — MUNI TEJASREE — log out

Dashboard > Tejasree-21kn1a1243 > #1 > Console Output

- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- Delete build '#1'

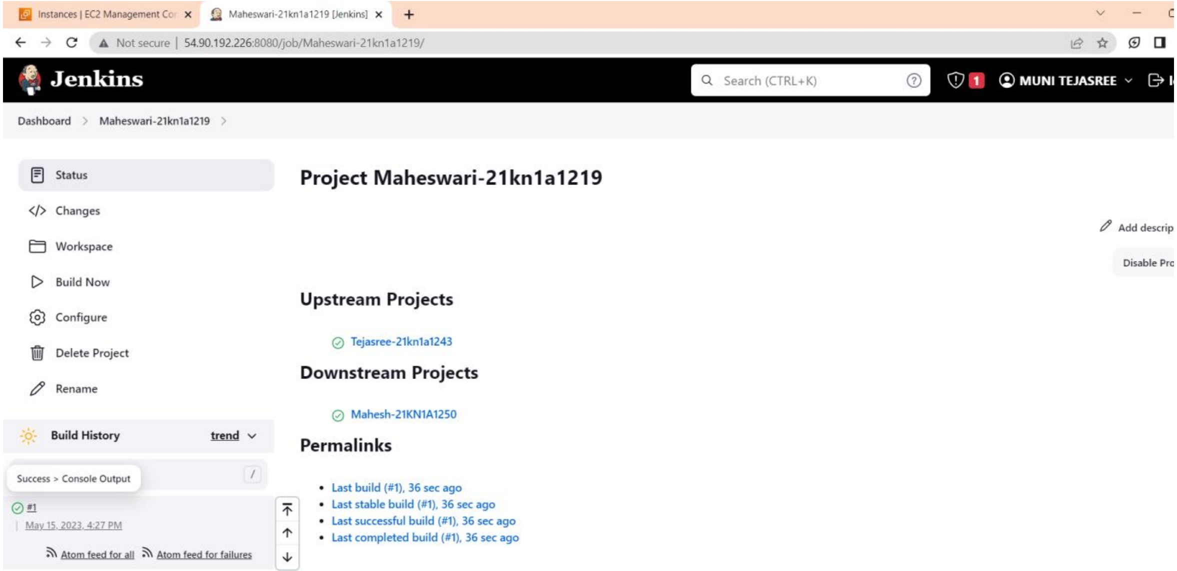
Console Output

```
Started by user MUNI TEJASREE
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Tejasree-21kn1a1243
[Tejasree-21kn1a1243] $ /bin/sh -xe /tmp/jenkins1255354281216502l3.sh
+ echo 'hii i am tejasree my rollnumber is:21KN1A1243'
hii i am tejasree my rollnumber is:21KN1A1243
Triggering a new build of Maheswari-21kn1a1219
Finished: SUCCESS
```

REST API — Jenkins 2.387.3

Screen 8.6:output for job1 i.e., Tejasree-21kn1a1243

Now goto 2nd job it is runs automatically.



Project Maheswari-21kn1a1219

Upstream Projects

- Tejasree-21kn1a1243

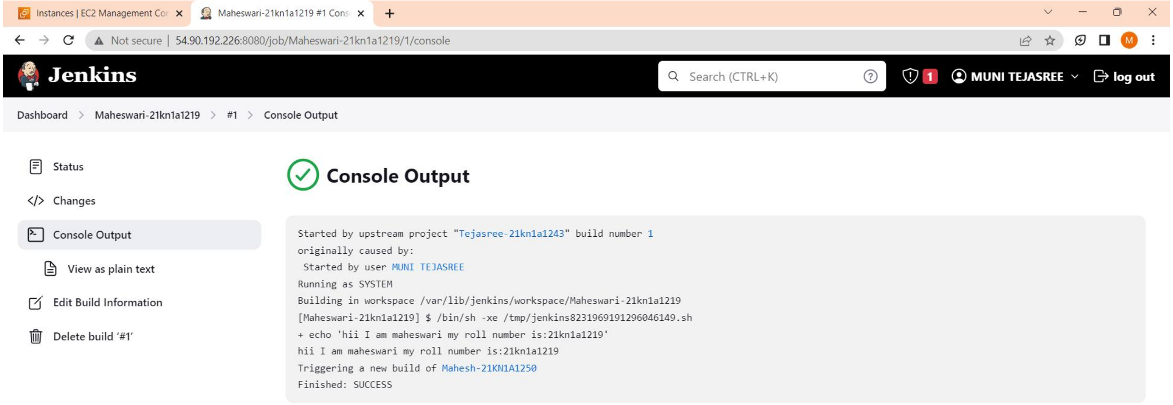
Downstream Projects

- Mahesh-21KN1A1250

Permalinks

- Last build (#1), 36 sec ago
- Last stable build (#1), 36 sec ago
- Last successful build (#1), 36 sec ago
- Last completed build (#1), 36 sec ago

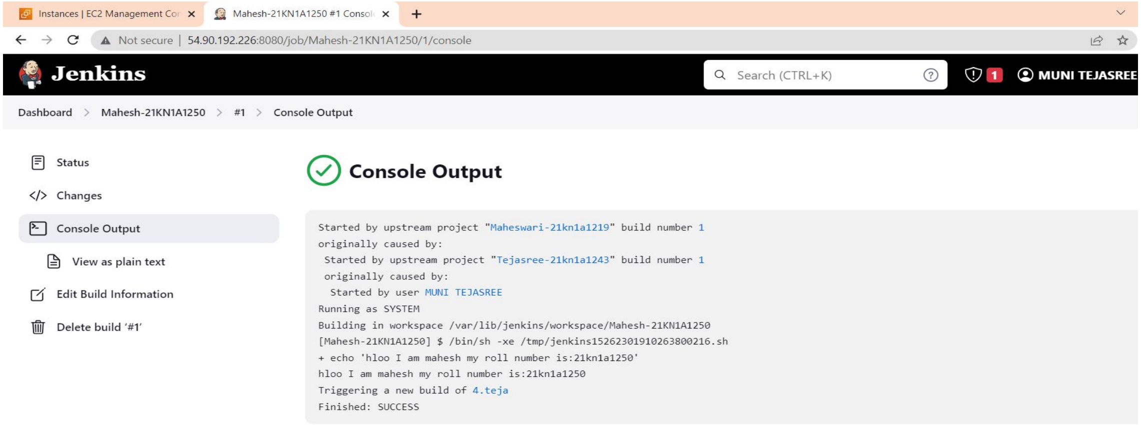
Screen 8.7: build 2nd job Maheswari-21kn1a1219


Instances | EC2 Management Co
Maheswari-21kn1a1219 #1 Cons
Not secure | 54.90.192.226:8080/job/Maheswari-21kn1a1219/1/console
Jenkins
Search (CTRL+K)
MUNI TEJASREE
log out
Dashboard > Maheswari-21kn1a1219 > #1 > Console Output
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#1'

Console Output

```
Started by upstream project "Tejasree-21kn1a1243" build number 1
originally caused by:
 Started by user MUNI TEJASREE
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Maheswari-21kn1a1219
[Maheswari-21kn1a1219] $ /bin/sh -xe /tmp/jenkins8231969191296046149.sh
+ echo 'hii I am maheswari my roll number is:21kn1a1219'
hii I am maheswari my roll number is:21kn1a1219
Triggering a new build of Mahesh-21KN1A1250
Finished: SUCCESS
```

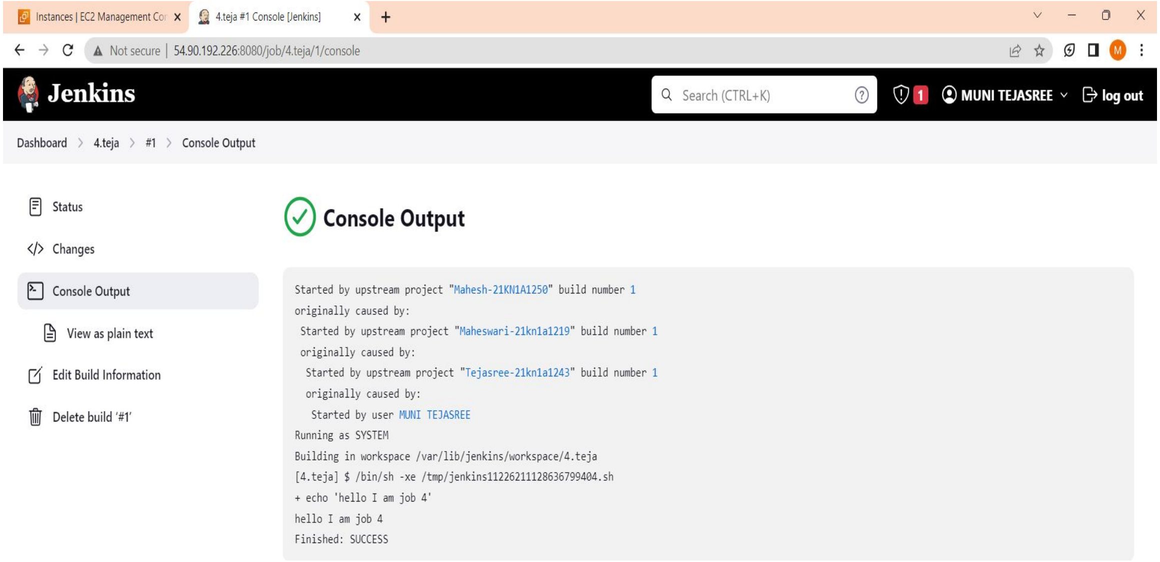
Screen 8.8:output for job2


Instances | EC2 Management Co
Mahesh-21KN1A1250 #1 Consol
Not secure | 54.90.192.226:8080/job/Mahesh-21KN1A1250/1/console
Jenkins
Search (CTRL+K)
MUNI TEJASREE
Dashboard > Mahesh-21KN1A1250 > #1 > Console Output
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#1'

Console Output

```
Started by upstream project "Maheswari-21kn1a1219" build number 1
originally caused by:
 Started by upstream project "Tejasree-21kn1a1243" build number 1
 originally caused by:
  Started by user MUNI TEJASREE
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Mahesh-21KN1A1250
[Mahesh-21KN1A1250] $ /bin/sh -xe /tmp/jenkins15262301910263800216.sh
+ echo 'hloo I am mahesh my roll number is:21kn1a1250'
hloo I am mahesh my roll number is:21kn1a1250
Triggering a new build of 4.teja
Finished: SUCCESS
```

Screen 8.9:output for job3

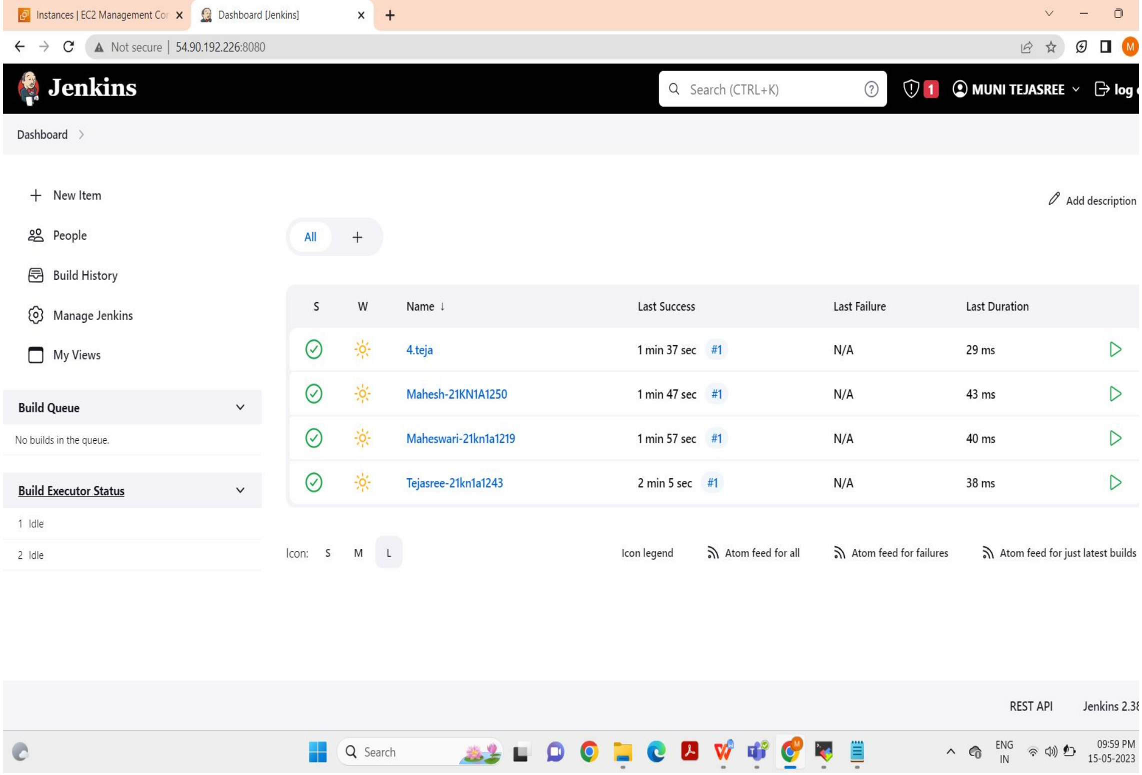


Instances | EC2 Management Co
4.teja #1 Console [Jenkins]
Not secure | 54.90.192.226:8080/job/4.teja/1/console
Jenkins
Search (CTRL+K)
MUNI TEJASREE
log out
Dashboard > 4.teja > #1 > Console Output
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#1'

Console Output

```
Started by upstream project "Mahesh-21KN1A1250" build number 1
originally caused by:
 Started by upstream project "Maheswari-21kn1a1219" build number 1
 originally caused by:
  Started by upstream project "Tejasree-21kn1a1243" build number 1
  originally caused by:
   Started by user MUNI TEJASREE
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/4.teja
[4.teja] $ /bin/sh -xe /tmp/jenkins11226211128636799404.sh
+ echo 'hello I am job 4'
hello I am job 4
Finished: SUCCESS
```

Screen 8.10:output for job4



S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	4.teja	1 min 37 sec #1	N/A	29 ms
✓	☀	Mahesh-21KN1A1250	1 min 47 sec #1	N/A	43 ms
✓	☀	Maheswari-21kn1a1219	1 min 57 sec #1	N/A	40 ms
✓	☀	Tejasree-21kn1a1243	2 min 5 sec #1	N/A	38 ms

Screen 8.11:Jenkins dashboard the 4 jobs run successfully

CHAPTER-9

9.CHALLENGES FACED

while creating jobs in Jenkins to illustrate upstream and downstream dependencies can be beneficial, it may come with a few challenges. Here are some common challenges we may encounter:

9.1 Challenges Faced in Jenkins Upstream:

Synchronization: Ensuring that downstream jobs are executed only when the upstream job(s) succeed can be challenging. Managing dependencies and handling failures gracefully can be complex.

Pipeline Complexity:As you create more upstream jobs to trigger downstream jobs, the overall pipeline can become complex and challenging to manage.

Resource Management: If multiple upstream jobs trigger downstream jobs concurrently, it can lead to resource contention and reduce the overall efficiency of the Jenkins environment.

Visibility: It may be difficult to gain clear visibility into the entire build or deployment process, especially when multiple jobs are involved in different pipelines. Debugging and monitoring can be challenging.

Challenges Faced in Jenkins Downstream:

Dependency Chain: Downstream jobs often rely on the success of upstream jobs. If any of the upstream jobs fail, it can lead to a cascade of failures in downstream jobs, making it challenging to pinpoint the root cause.

Longer Build Times: Downstream jobs may need to wait for upstream jobs to complete, which can lead to longer build times and delays in the overall pipeline.

Data Transfer: Passing data and artifacts from upstream to downstream jobs can be complex, especially when there are multiple stages or different technologies involved.

Resource Consumption: Downstream jobs can consume resources that affect the performance of the Jenkins master and agents. This may require efficient resource management and scalability planning.

To address these challenges, it's essential to have a well-structured and organized Jenkins pipeline, implement proper error handling and notification mechanisms, use version control for job configurations, and regularly monitor and optimize your Jenkins setup.

CHAPTER-10

10.CONCLUSION

In conclusion,By configuring upstream jobs to trigger downstream jobs automatically upon successful completion, developers can ensure a streamlined and continuous flow of software development. Upstream jobs typically involve tasks such as building and testing code, while downstream jobs may focus on tasks like deployment or integration testing.

- This upstream-downstream relationship in Jenkins enables parallelization and parallel development, where multiple teams can work on different parts of a project simultaneously. It promotes collaboration, increases efficiency, and reduces time-to-market.

- Furthermore, Jenkins provides robust features for monitoring and tracking job status, enabling quick identification of issues and facilitating prompt resolution.With its extensive plugin ecosystem, Jenkins can be customized and extended to fit specific project requirements.

In summary, Jenkins' support for upstream and downstream jobs allows for a structured and automated approach to software development, enabling teams to streamline their processes, enhance collaboration, and accelerate delivery.

.

CHAPTER-11

11.FUTURE SCOPE

Jenkins is a popular automation tool used for continuous integration and continuous delivery (CI/CD) processes. Creating jobs in Jenkins to illustrate upstream and downstream dependencies can be beneficial for managing complex software development workflows. The future scope of utilizing Jenkins for this purpose includes:

Streamlining CI/CD Pipelines:

Jenkins allows you to define and configure pipelines as code, enabling you to orchestrate the entire software delivery process. By creating jobs to represent upstream and downstreamstages,we can visualize and automate the flow of changes from one phase to another.

Enhancing Collaboration:

Jobs in Jenkins can serve as a communication mechanism between different teams or individuals involved in the software development lifecycle. Upstream jobs trigger downstream jobs when they complete successfully, enabling efficient coordination and collaboration among teams.

Improved Fault Isolation and Debugging:

By clearly defining upstream and downstream relationships in Jenkins, it becomes easier to identify the source of issues or failures. When a downstream job fails, we can trace it back to its upstream job(s) to pinpoint potential causes, facilitating faster troubleshooting and resolution.

Scaling and Parallel Execution:

Jenkins allows parallel execution of jobs, which can significantly reduce the overall build and deployment time. By setting up upstream and downstream relationships correctly, we can takeadvantage of parallelism to optimize resource utilization and speed up the software delivery process.

Integrating with External Tools:

Jenkins can integrate with various and testing frameworks. By creating jobs that incorporate these integration, we can automate tasks, trigger builds based on code changes, and streamline the overall development workflow.

Monitoring and Reporting:

Jenkins provides a comprehensive dashboard and reporting capabilities, allowing to monitor the status of jobs and generate reports on build/test results, code coverage, and other metrics.

CHAPTER-12

12.References & Appendix

12.1 REFERENCES

1) **Jenkins Documentation:** The official documentation provides comprehensive information on installing, configuring, and using Jenkins. You can find the documentation at: https://www.jenkins.io/doc/

2) **Jenkins User Handbook:** This handbook offers step-by-step instructions and best practices for setting up and configuring Jenkins jobs. It covers various job types, including Freestyle projects and Pipeline. You can access the handbook at: https://www.jenkins.io/doc/book/

3) **Jenkins Plugins:** Depending on your job requirements,you may need to install specific plugins.You can explore and search for plugins suitable for your specific needs in the Jenkins Plugin Index: https://plugins.jenkins.io/

12.2 APPENDIX

Sample commands to install Jenkins on an Amazon Linux 2 instance:

❖ install Java

“yum install java-11-openjdk”

❖ Install Maven

“yum install maven -y”

❖ Install Jenkins

“sudo wget -O /etc/yum.repos.d/Jenkins.repo https://pkg.Jenkins.io/redhat-stable/Jenkins.repo”

“sudo rpm --import https://pkg.Jenkins.io/redhat-stable/Jenkins.io-2023.key”

“sudo yum install Jenkins -y”

❖ Start Jenkins and enable it to start on boot

“sudo systemctl start jenkins”

“sudo systemctl enable jenkins”

❖ check the status of the Jenkins:

“sudo systemctl status Jenkins”