| **Name : Parth Kadam** | **Class/Roll No. :D11AD/27** | **Grade :** |
| --- | --- | --- |

**Title of Experiment :** To understand Continuous Integration, install and configure Jenkins to set up a build Job.

**Objective of Experiment :**To Integrate and deploy tools like Jenkins and Maven, which is used to build, test and deploy applications in DevOps environment.

**Outcome of Experiment :** Understand the importance of Selenium and Jenkins to test Software Applications.

**Problem Statement :** Understand Continuous Integration by configuring Jenkins for automated build job setup.

**Description / Theory :**

**Continuous Integration:**

Continuous Integration (CI) in DevOps is a practice where developers regularly merge their code changes into a shared repository. Each merge triggers automated builds and tests, ensuring that the integrated code is always functional and compatible. CI promotes collaboration, reduces integration issues, and accelerates the software delivery process, enabling teams to deliver high-quality software with greater efficiency and reliability.

**Jenkins:**

Jenkins is an open-source automation server widely used for implementing Continuous Integration and Continuous Delivery (CI/CD) pipelines. It provides a platform for automating software builds, tests, and deployments across different environments. With its extensive plugin ecosystem and flexible configuration options, Jenkins enables teams to streamline their development processes, improve collaboration, and deliver software more rapidly and reliably.

**Steps to install and configure Jenkins to set up a build job:**

Install Jenkins: Download and install Jenkins from the official website or package manager

for your operating system.

Start Jenkins: Once installed, start the Jenkins service on your machine. By default, Jenkins runs

on port 8080.

Access Jenkins Dashboard: Open a web browser and navigate to http://localhost:8080 (replace

localhost with the hostname or IP address of your Jenkins server if accessing remotely).

Unlock Jenkins: During the first run, Jenkins will ask for an initial admin password. Retrieve this

password from the Jenkins server's file system and paste it into the unlock screen.

Install Plugins: Select the option to install suggested plugins or choose specific plugins based on

your requirements. Wait for the plugin installation to complete.

Set up Admin User: Create an admin user account and provide necessary details such as

username, password, and email address.

Configure Jenkins: Customize Jenkins settings as per your requirements, including global

configurations, security settings, and plugin configurations.

Create a New Job: Click on "New Item" on the Jenkins dashboard to create a new job. Provide a

name for the job and select the type of job (e.g., Freestyle project, Pipeline).

Configure Job: Configure the job settings such as source code repository URL, build triggers,

build steps, post-build actions, and other parameters.

Save Job Configuration: Save the job configuration settings.

Build Job: Trigger the job manually or wait for it to run based on the configured triggers. Monitor

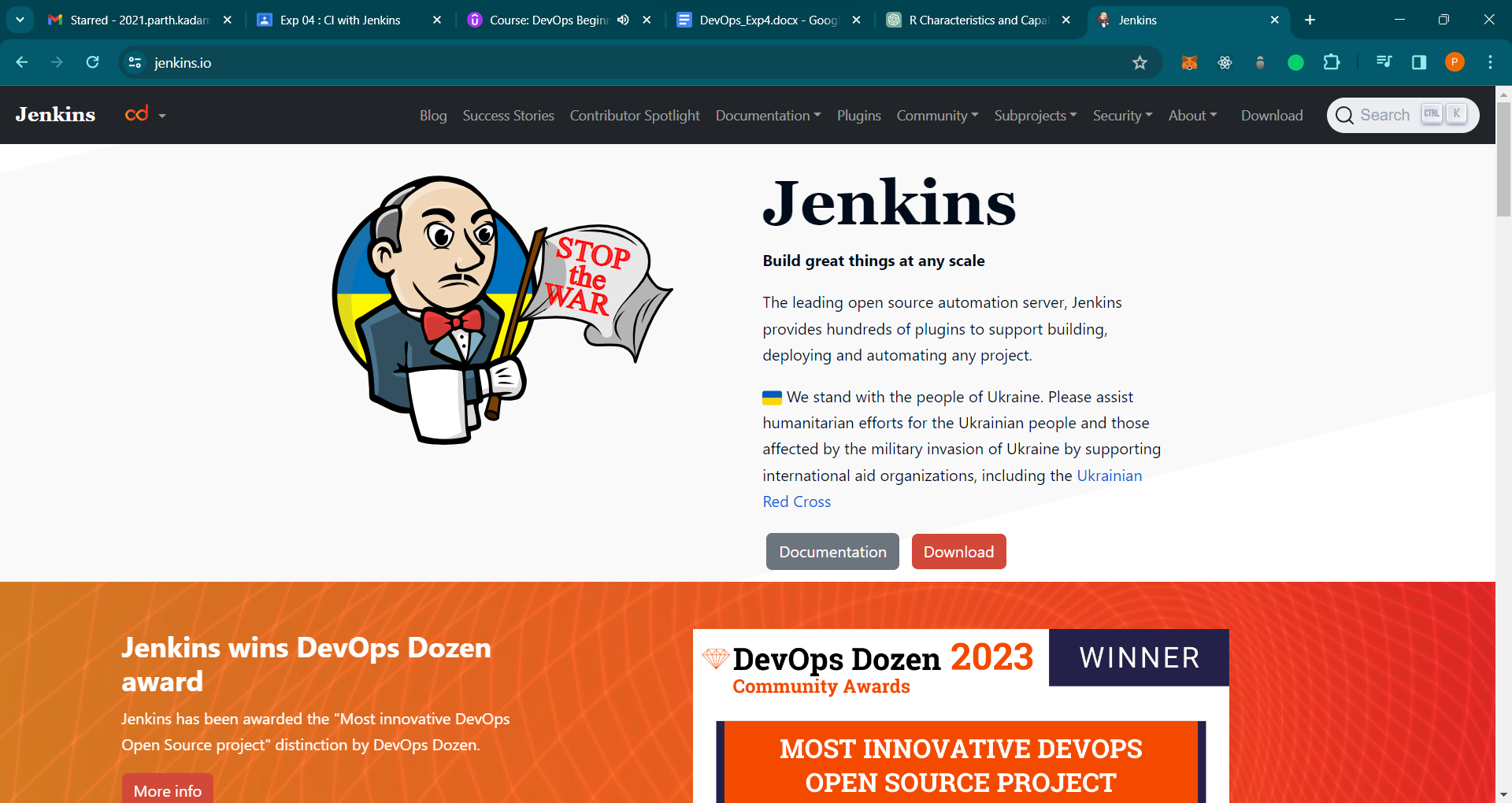
the build process and view build results.

Set up Build Notifications: Configure build notifications to receive alerts on job status changes

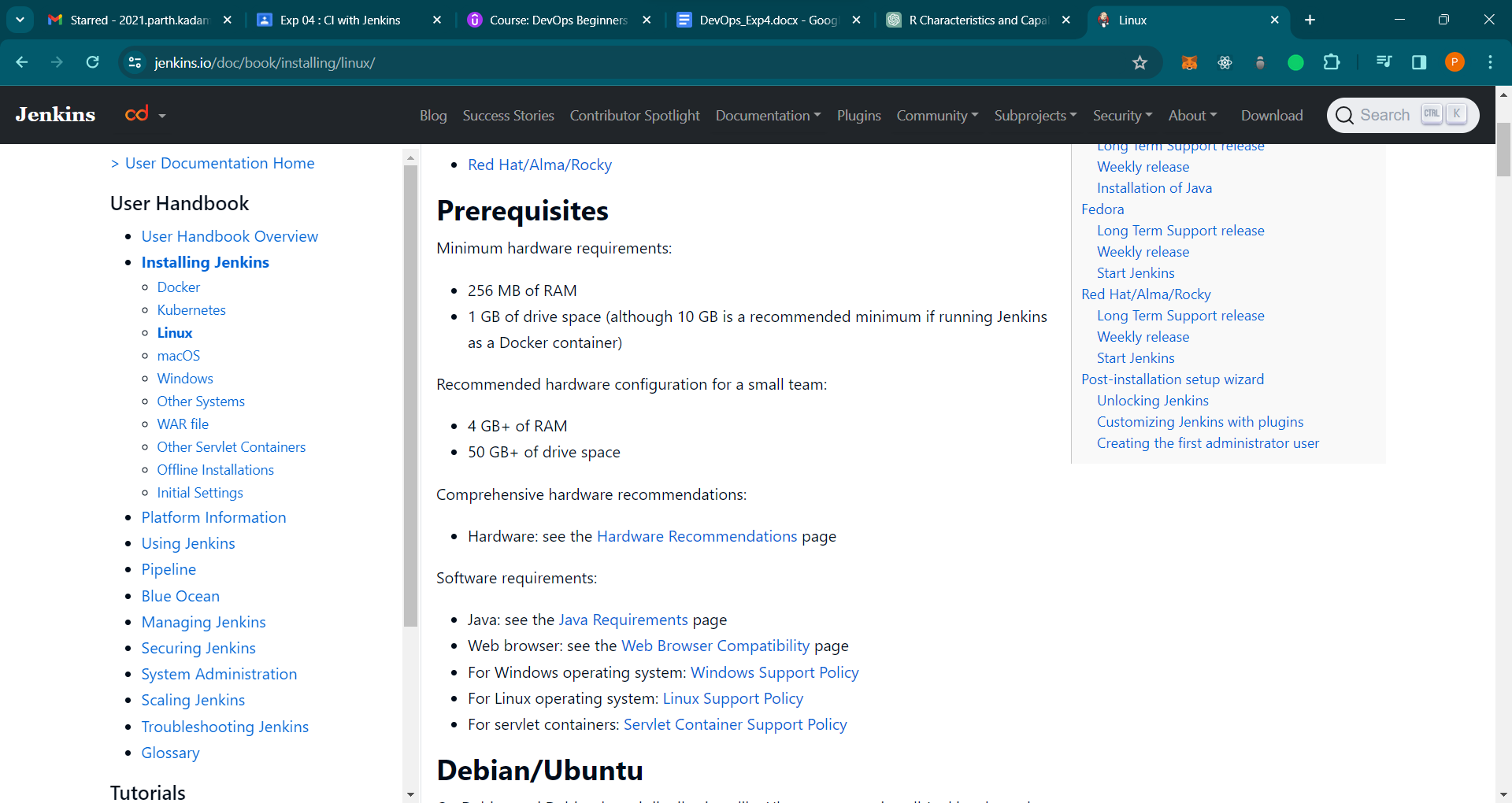
via email, chat, or other notification channels.

**Program:**

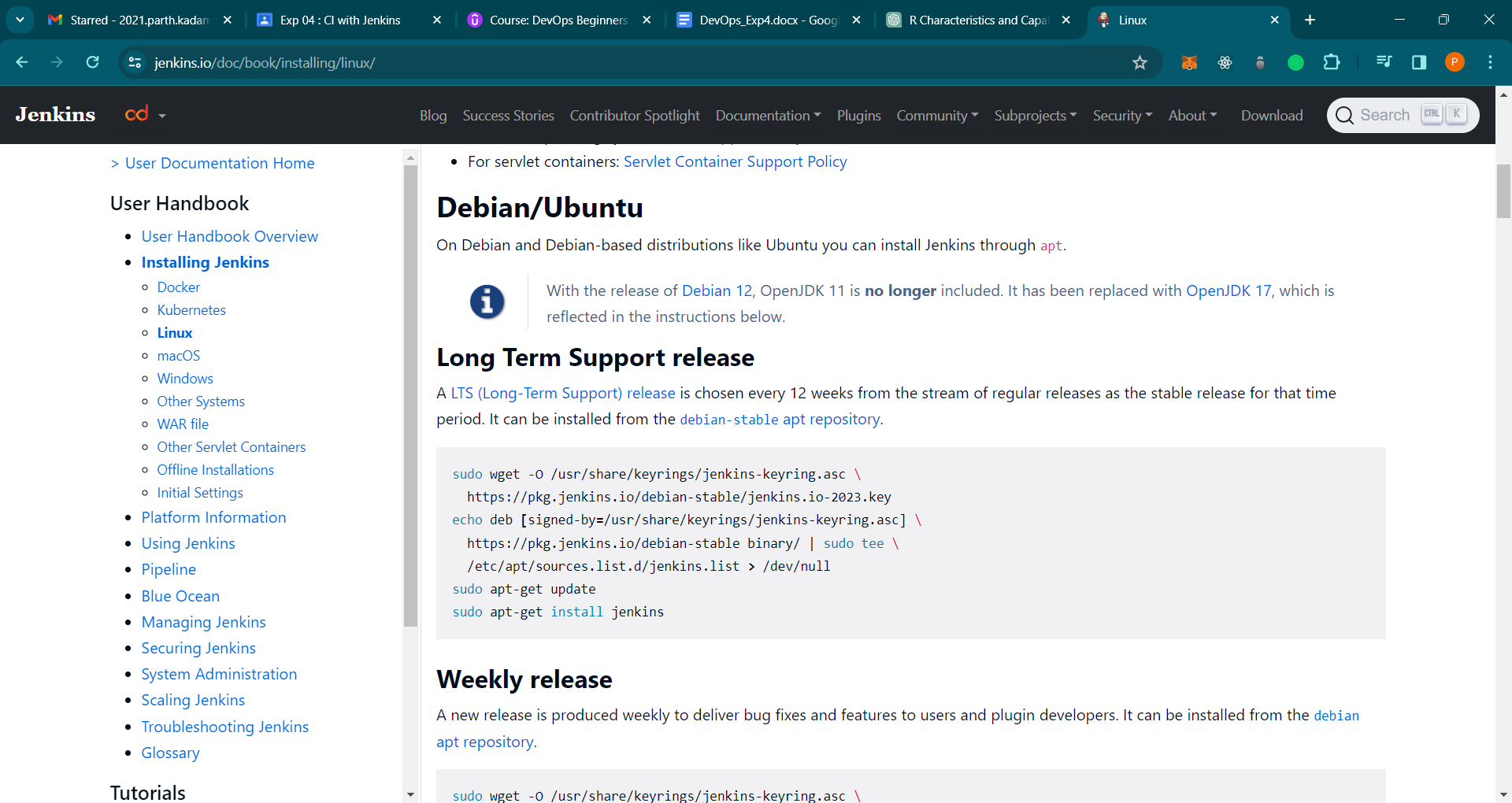
1. [**Jenkins.io**](https://www.jenkins.io/)

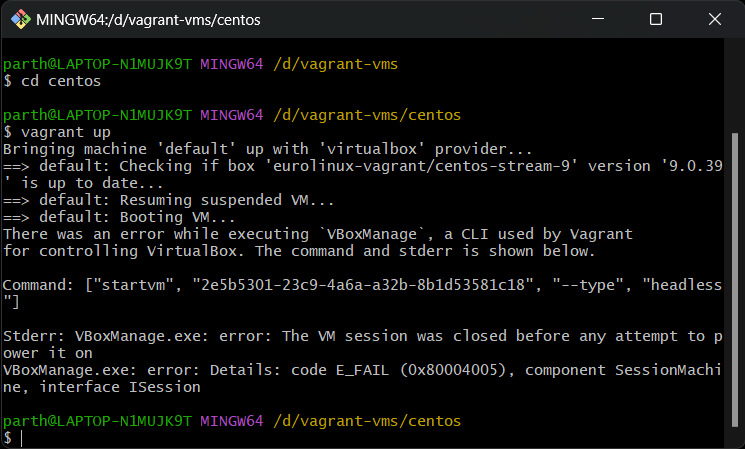
****

1. **Prerequisites for Jenkins**

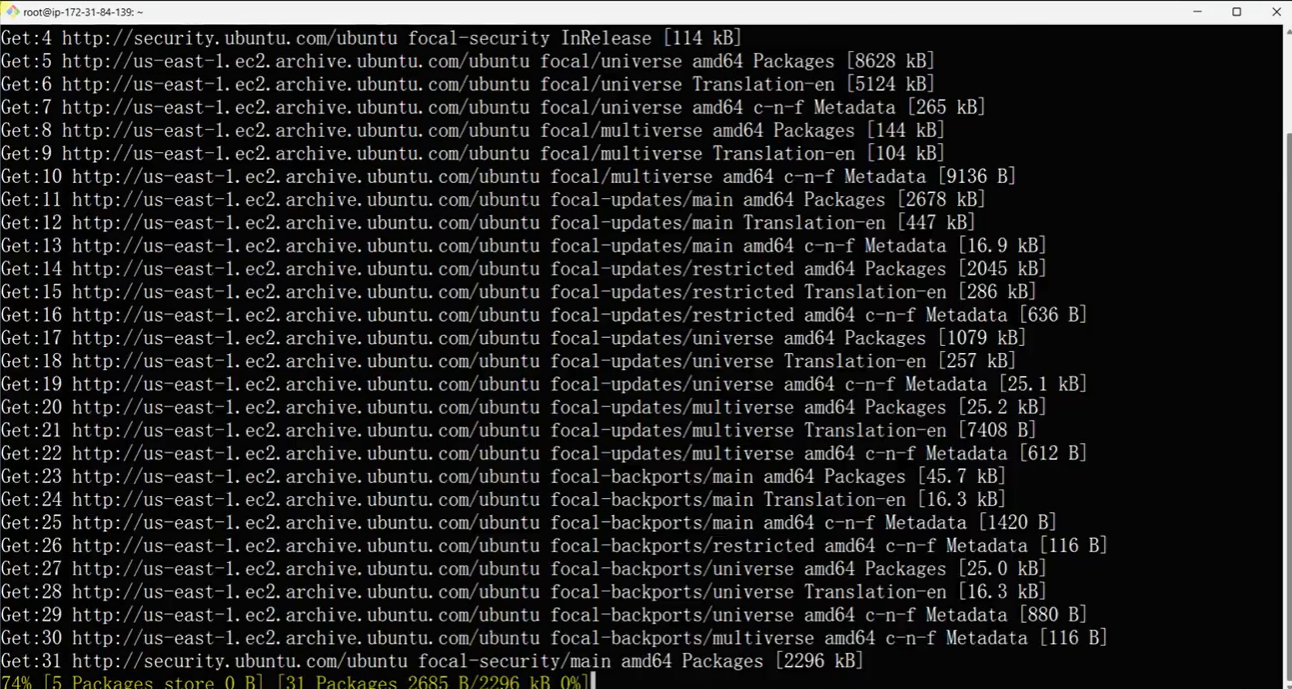
****

1. **Linux commands to install Jenkins**

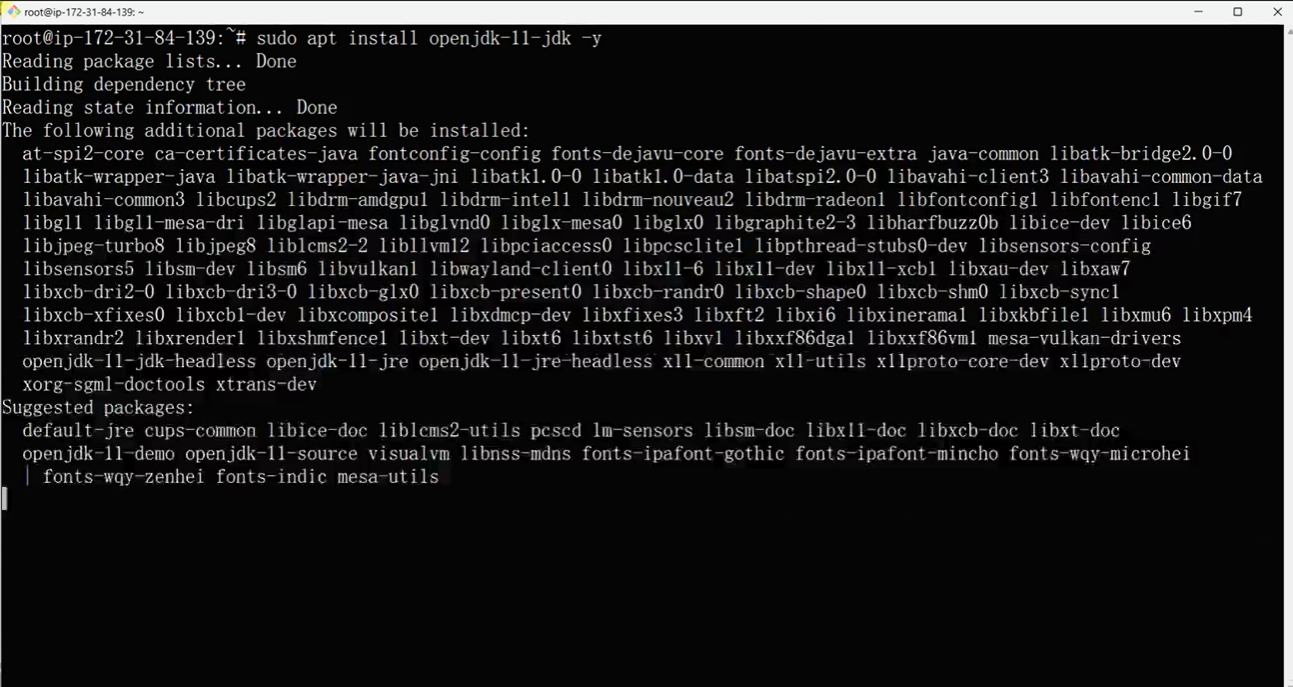
****

****

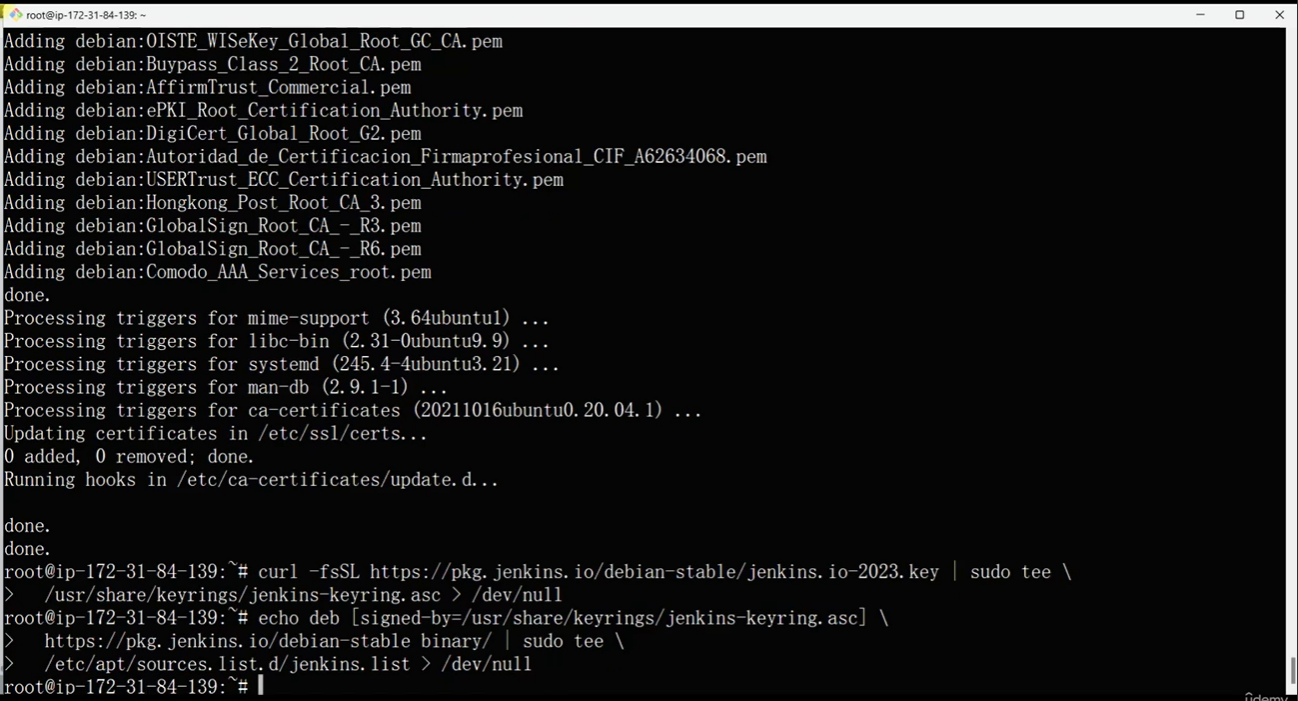
**Sudo apt update**

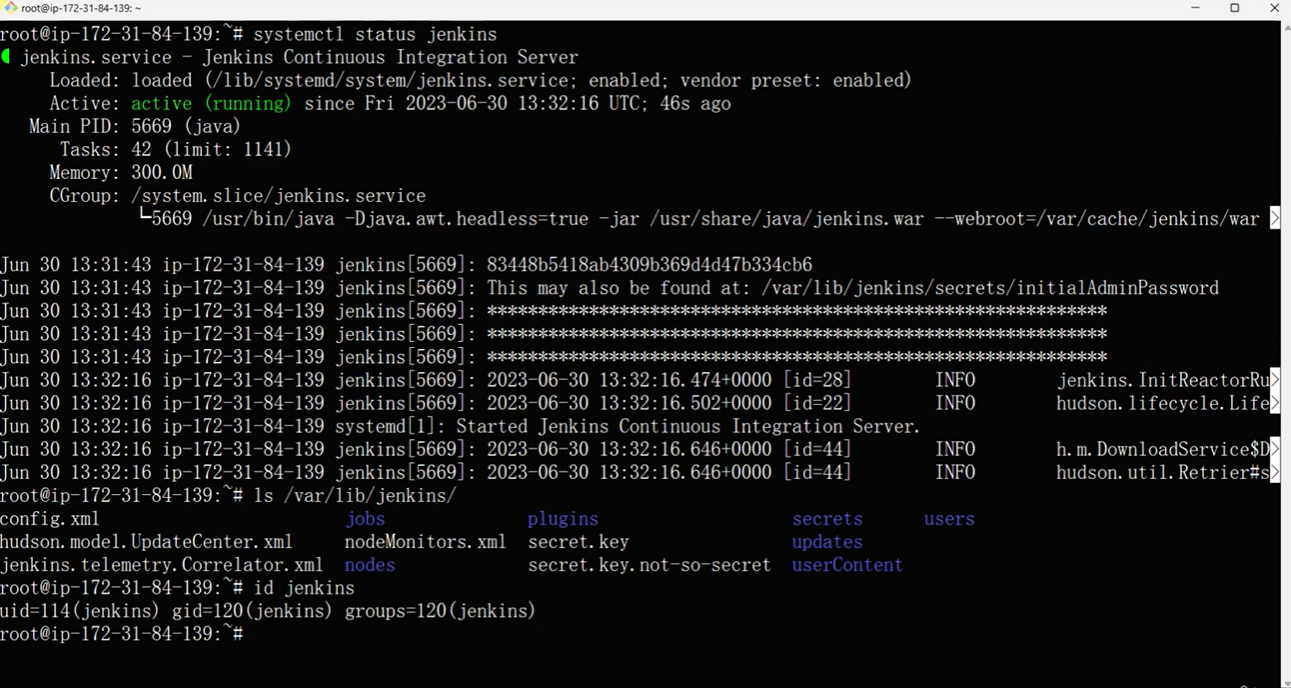
****

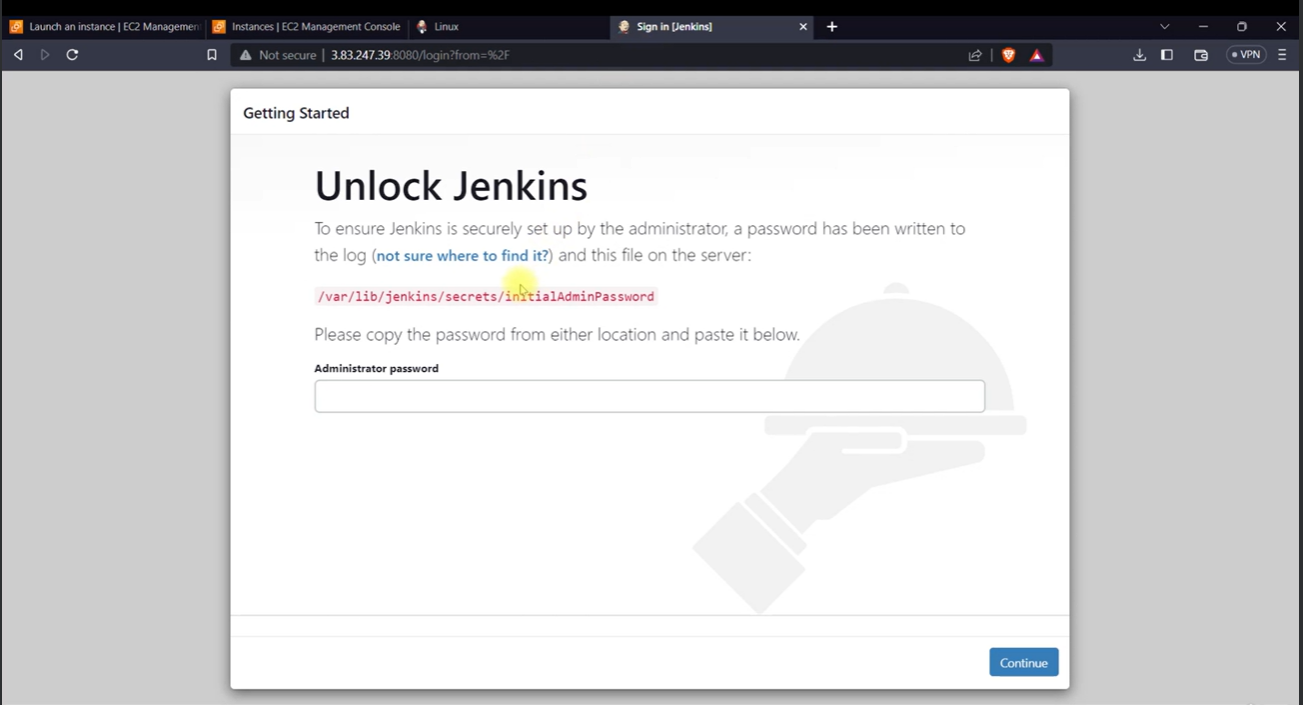
**Install jdk**

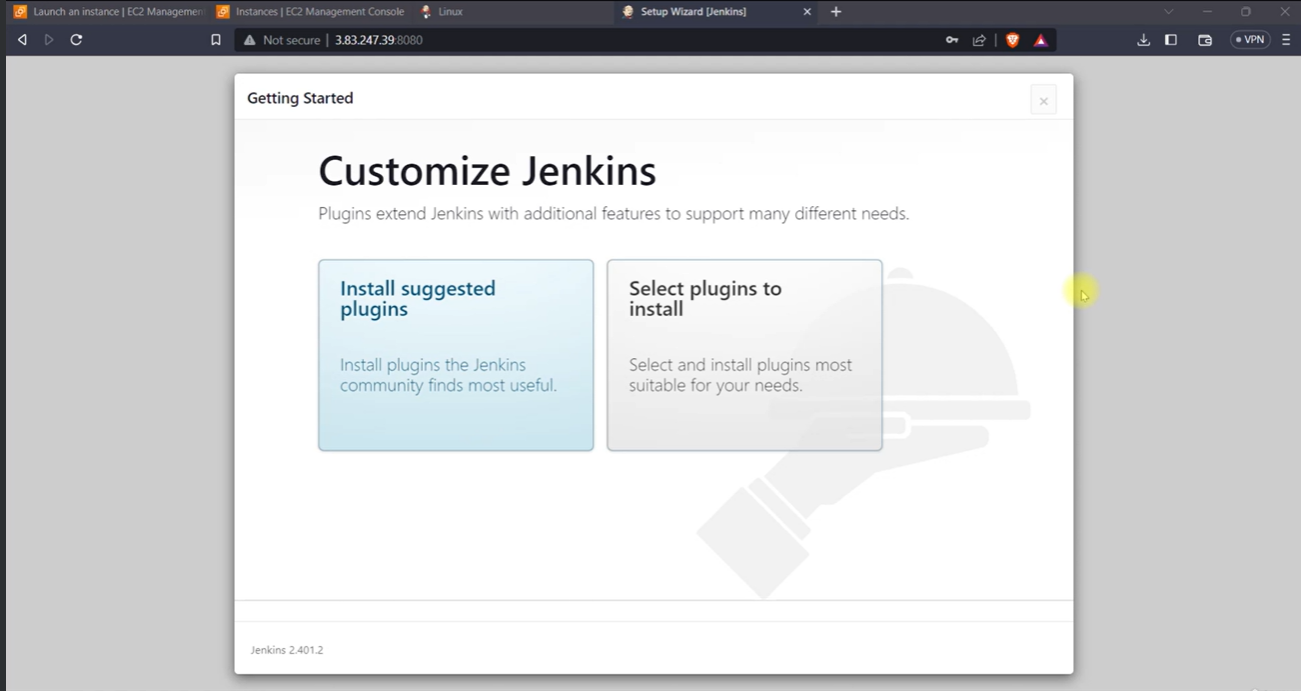
****

**Install Jenkins**

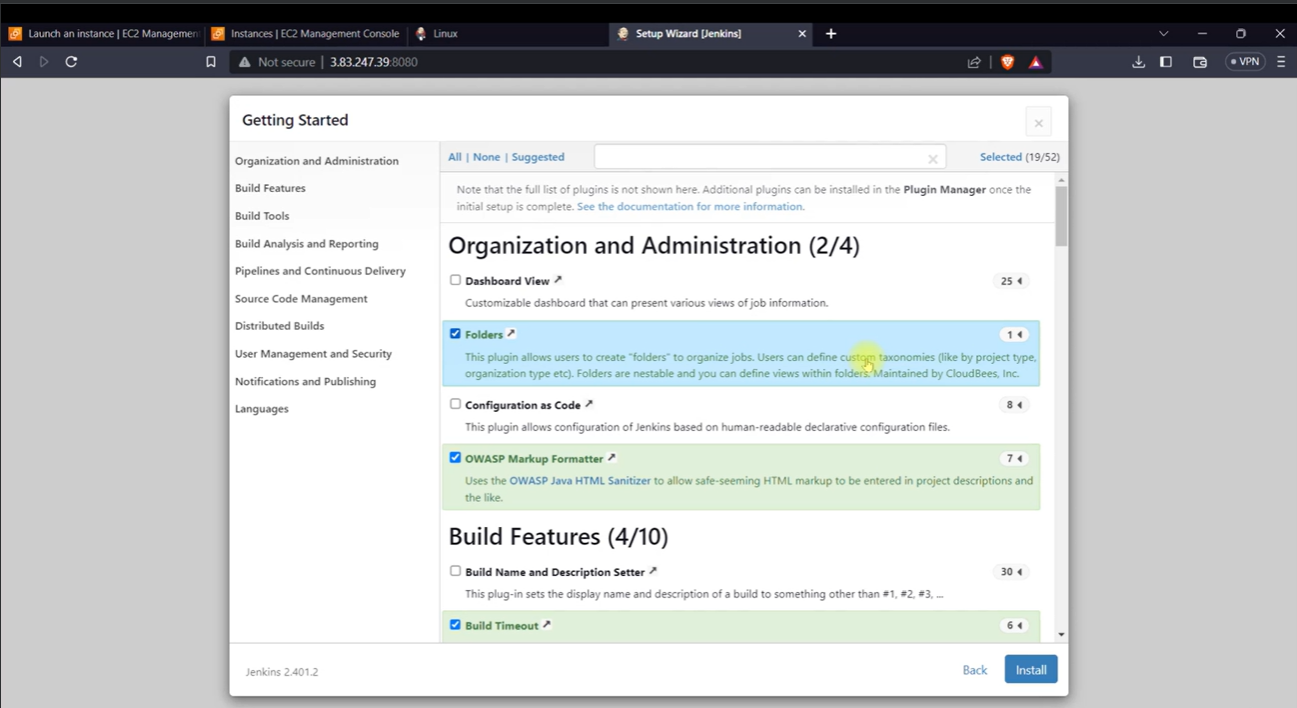
****

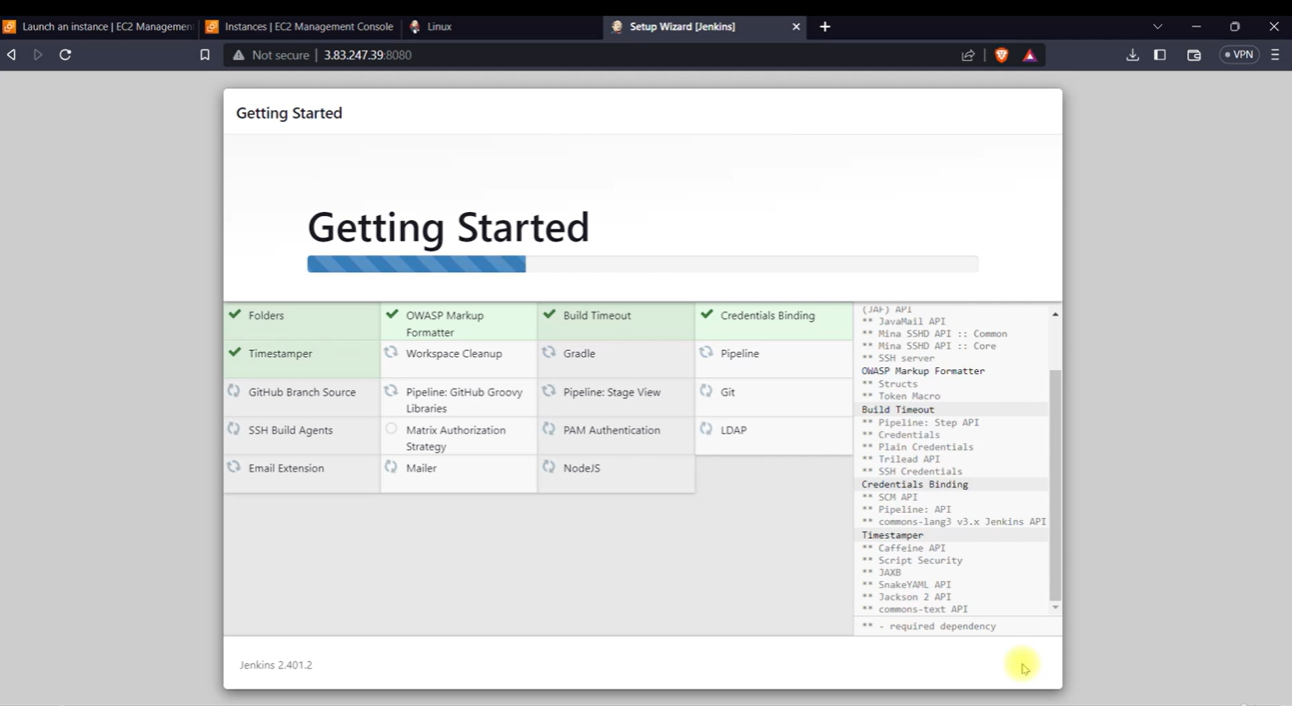
****

****

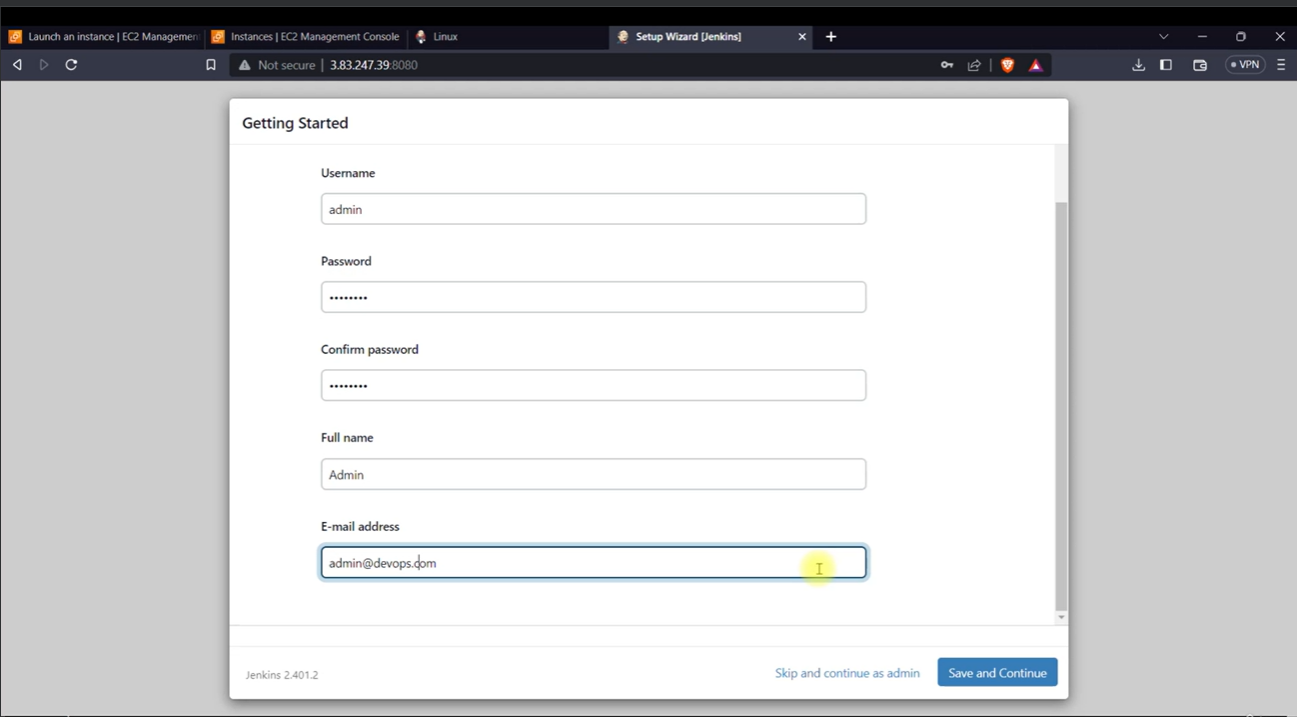
****

**Add Jenkins tools**

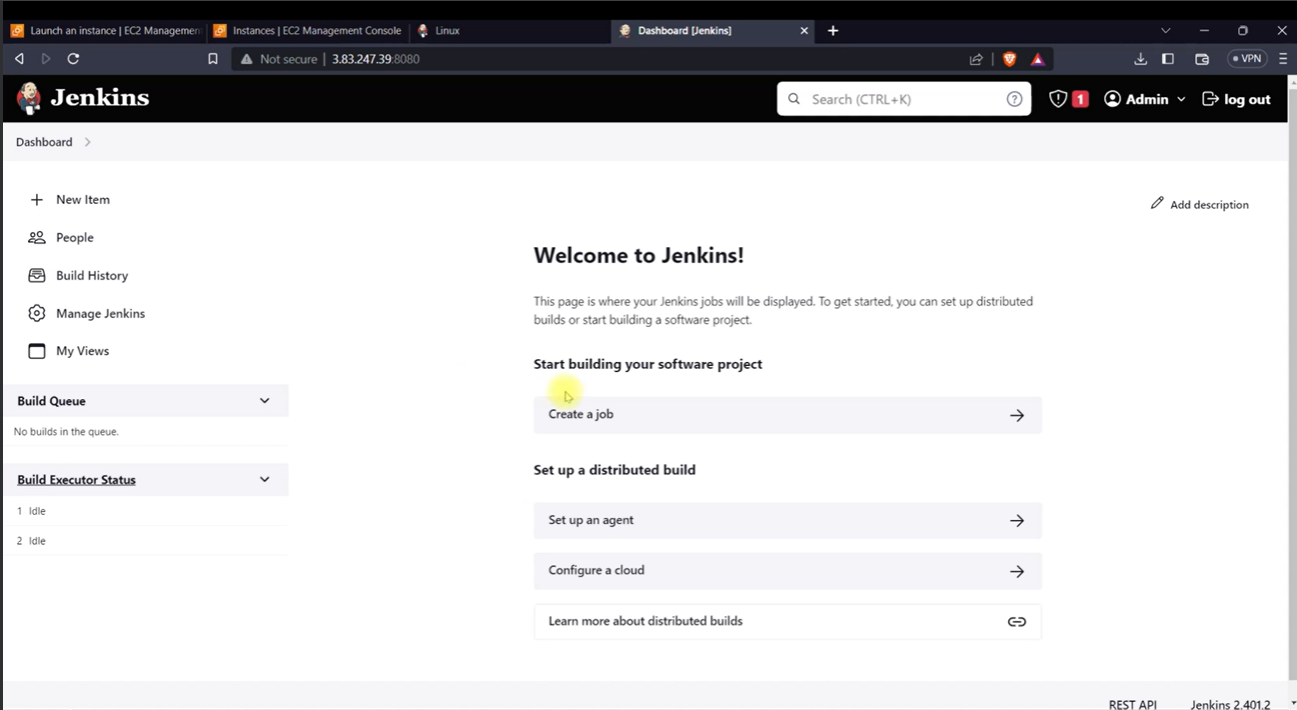
****

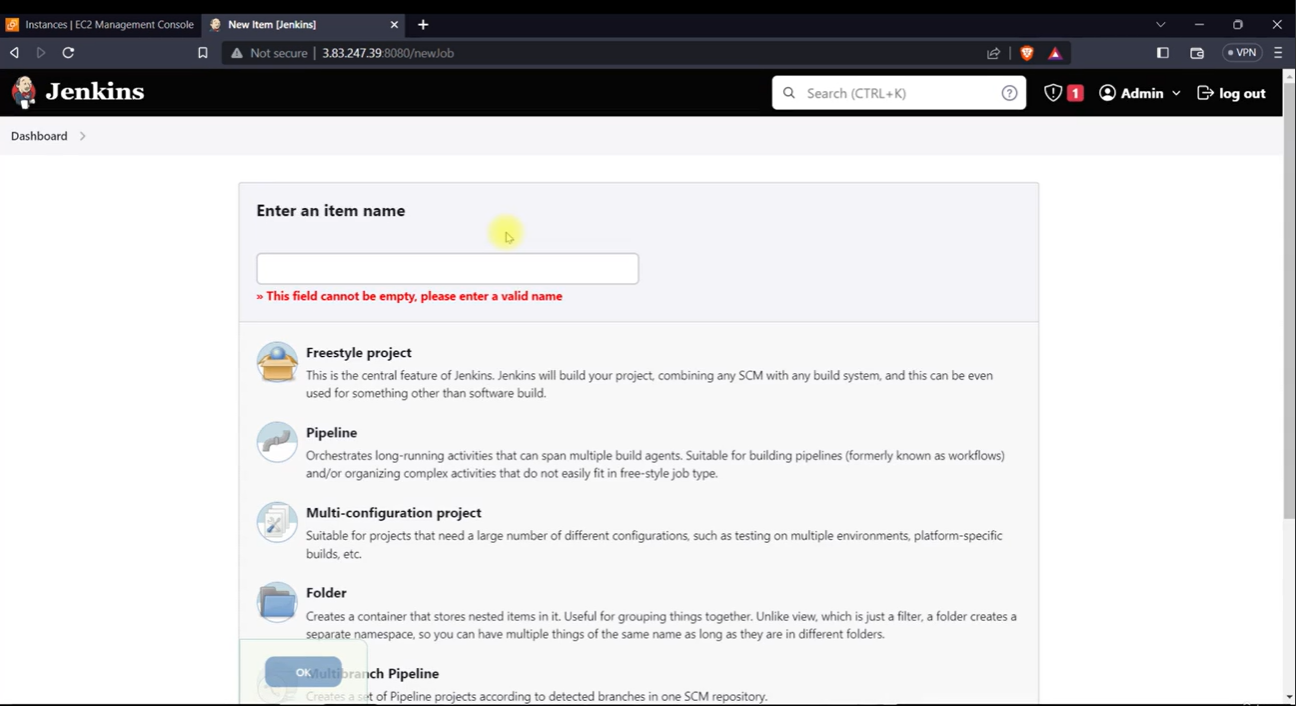
****

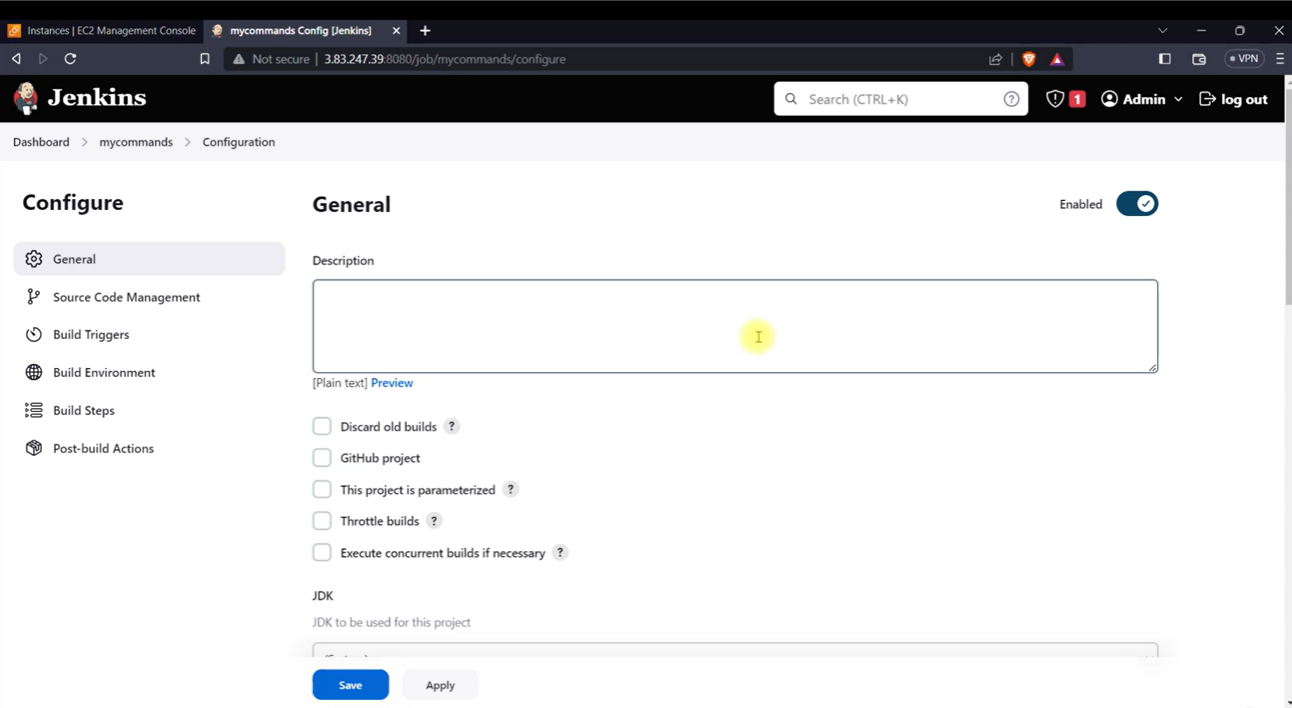
**Create admin**

****

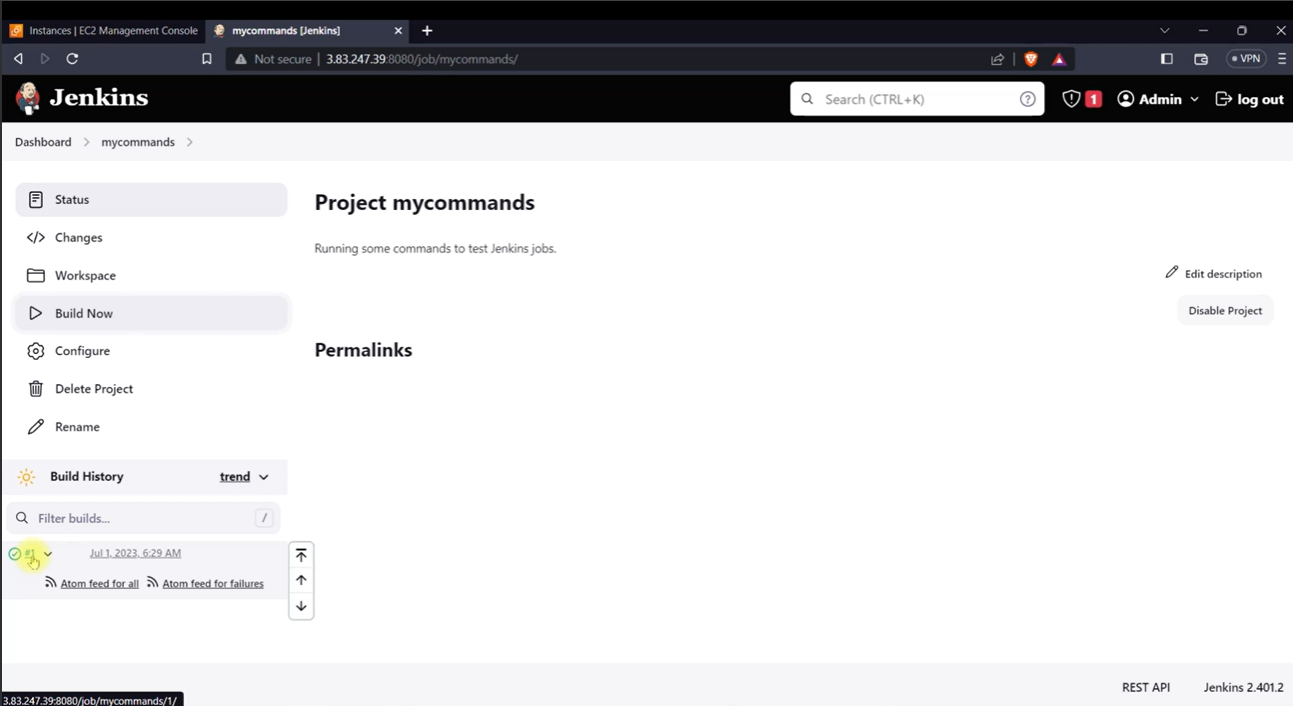
**Create build job**

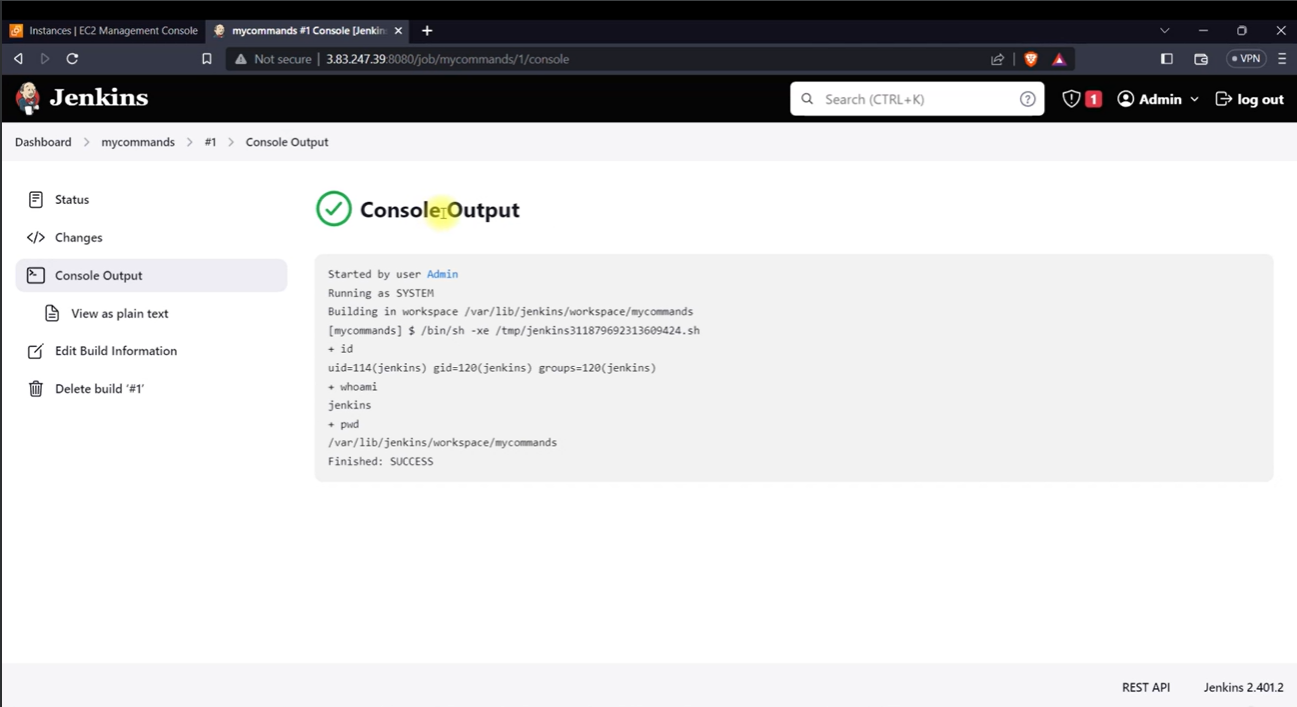
****

****

****

****

****

****

**Results and Discussions :** In conclusion, configuring Jenkins for automated build job setup provides an efficient approach to implementing Continuous Integration practices. This facilitates streamlined development workflows and promotes continuous delivery, enhancing overall software development processes.