**DEVOPS**

**ASSIGNMENT 1**

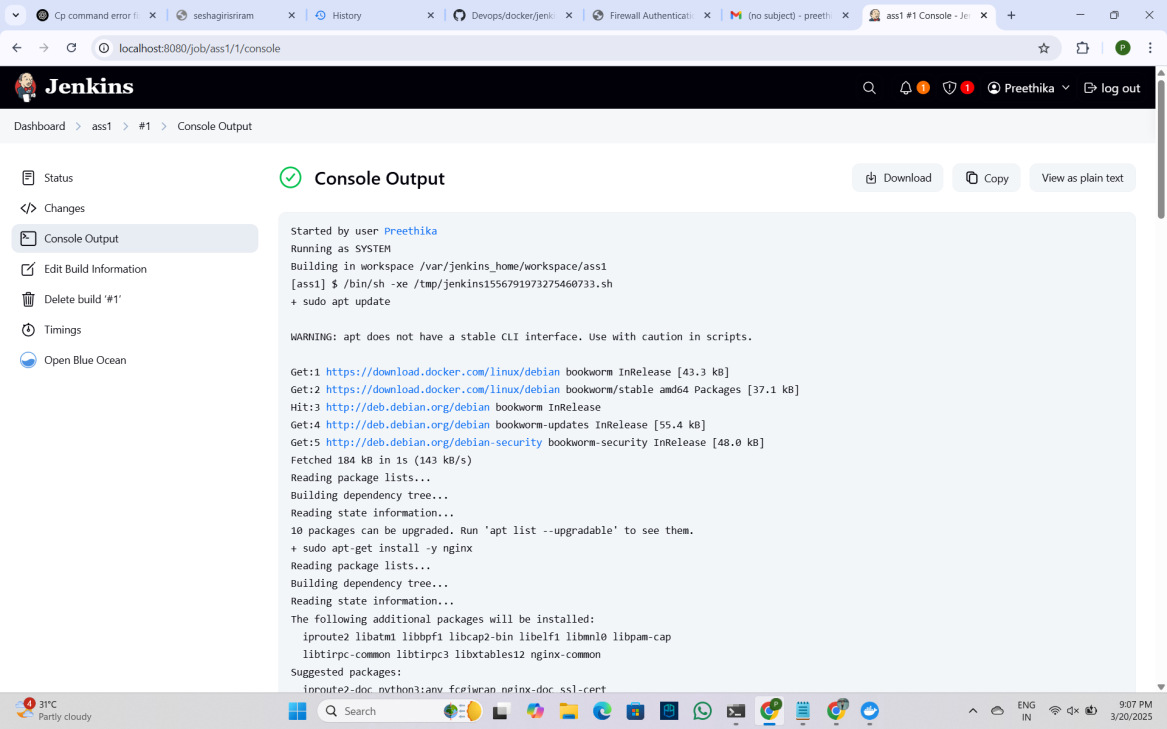
**Installation of Nginx.**

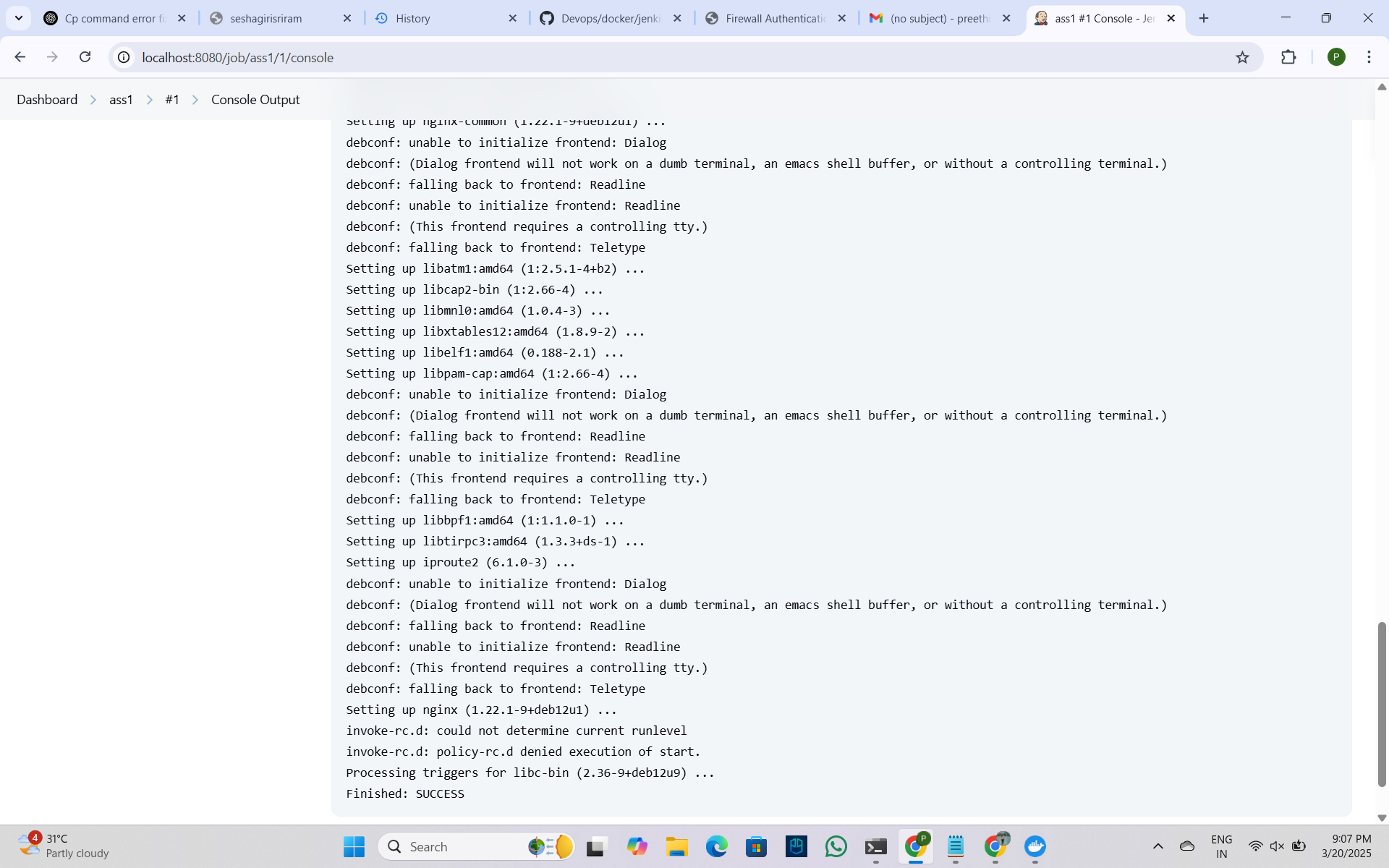
**Task Description**:  
This task involves creating a Jenkins Freestyle job that automates the installation of Nginx on an Ubuntu server using the command sudo apt-get install -y nginx. This automation ensures consistent server setup and can be easily repeated, making it ideal for automated environments.

**Steps:**

1. **Update Package List**:
   * Run the following command to update the package list:
   * sudo apt update
2. **Install Nginx**:
   * Install Nginx using the following command:
   * sudo apt-get install -y nginx
3. **Verify Installation**:
   * Check if Nginx is running with the command:
   * systemctl status nginx
4. **Create Jenkins Freestyle Job**:
   * In Jenkins, click New Item and create a Freestyle project (e.g., Install\_Nginx).
5. **Configure Job**:
   * Add a build step of type Execute shell.
   * In the shell command box, enter:
   * sudo apt update
   * sudo apt-get install -y nginx
6. **Save and Run Job**:
   * Save the job and click Build Now to run it.
7. **Check Console Output**:
   * After the job completes, view the **Console Output** to confirm the successful installation of Nginx.

**OUTPUT:**

****



**ASSIGNMENT 2**

**CI/CD Pipeline Setup Using Jenkins**

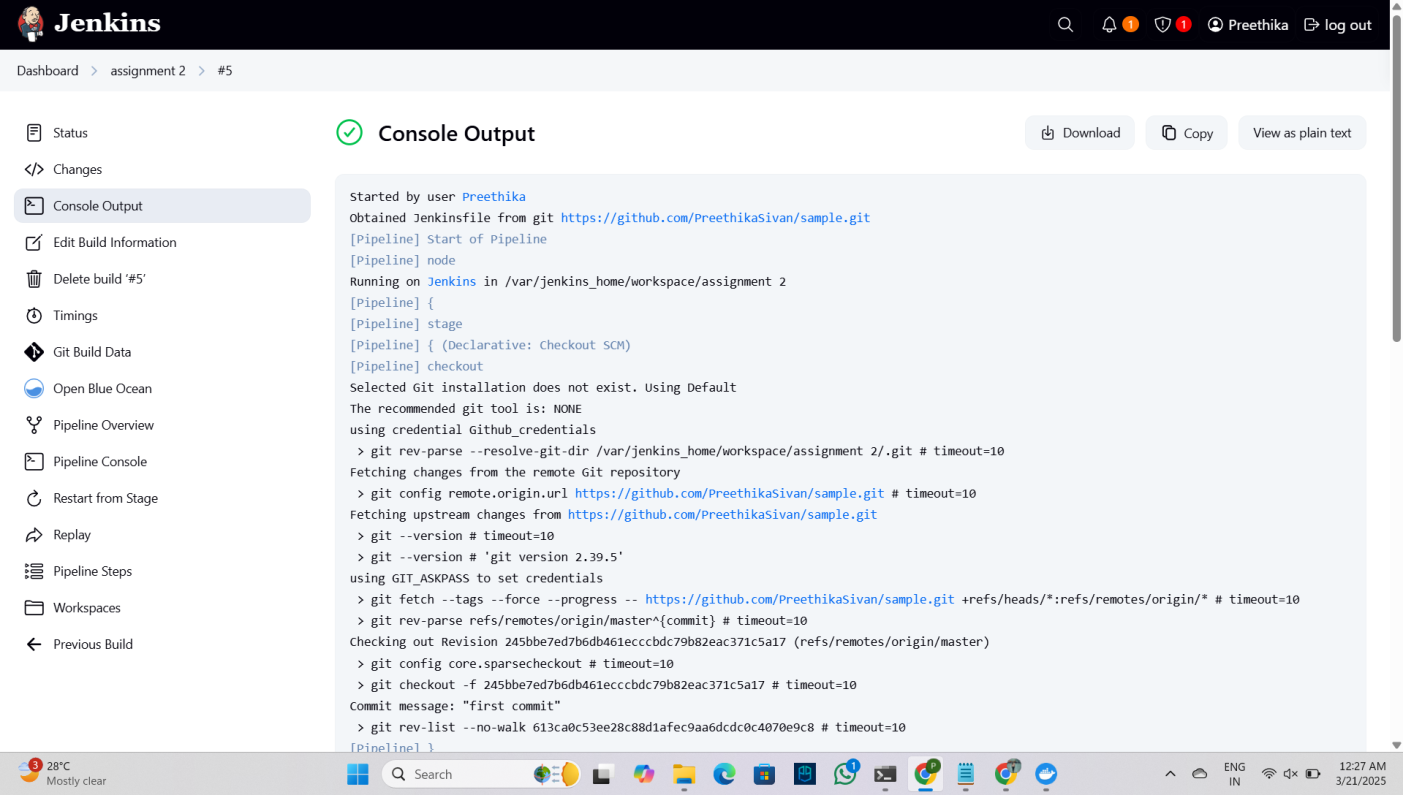
**Task Description:**

This task involves setting up a Continuous Integration and Continuous Deployment (CI/CD) pipeline using Jenkins to automate the build, test, and push process of a containerized web application. The pipeline integrates GitHub for source code management, Docker for containerization, and a container registry (such as Docker Hub) for storing the container image.

**Steps to Set Up the CI/CD Pipeline:**

1. **Install Jenkins:**
   * Ensure Jenkins is running and install necessary plugins (Git, Docker).
2. **Create a New Pipeline Job:**
   * Create a new Pipeline job in Jenkins to automate the build, test, and deployment process.
3. **Configure the GitHub Repository:**
   * Link your GitHub repository to Jenkins to automatically pull the latest code.
4. **Write the Pipeline Script (Jenkinsfile):**
   * Define stages in the Jenkinsfile:
     + **Clone Repository**: Pull the latest code from GitHub.
     + **Build Docker Image**: Build the image using Docker.
     + **Push Docker Image**: Push the image to a container registry (e.g., Docker Hub).
5. **Add Docker Credentials (if needed):**
   * Add Docker credentials to Jenkins if using a private registry for authentication.
6. **Trigger the Pipeline:**
   * Trigger the pipeline manually or set up GitHub webhooks for automatic triggers on code changes.

**OUTPUT:**

****

