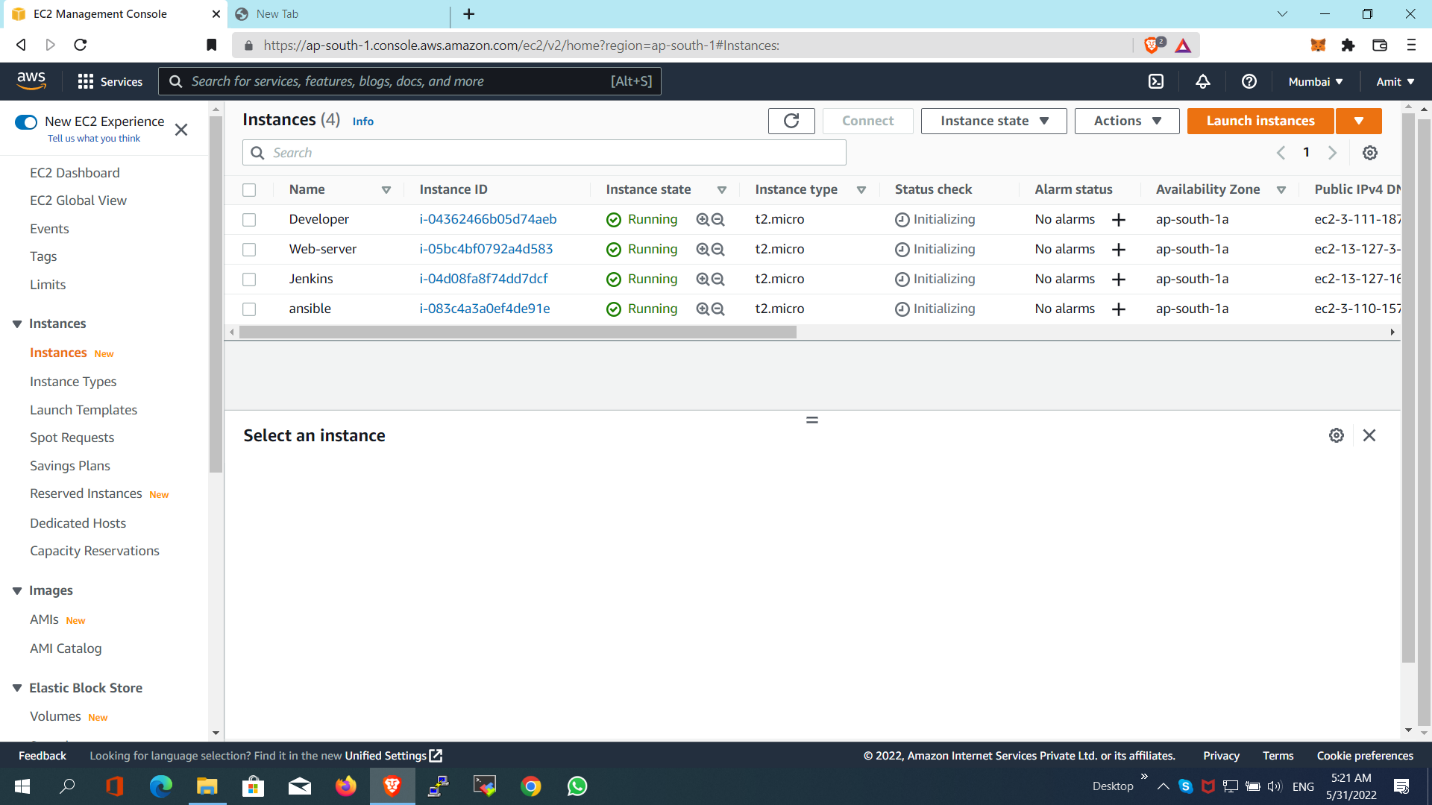
**PROJECT**

**Deploy a Web Application using AWS services and DevOps Tools**

**For this Project we need 3 AWS EC2 Instances or Web-Servers.**

1. The first instance is for Web-server.
2. The second instance is for Jenkins-Server.
3. The third instance is for Ansible server.
4. We need one GitHub account for website files**.**

First, we create 3 EC2 instances.

****

**Installing the setups for this project**

Before creating the project, first we creating the setup or installing the setup for this project. All the installation process is performed in this part.

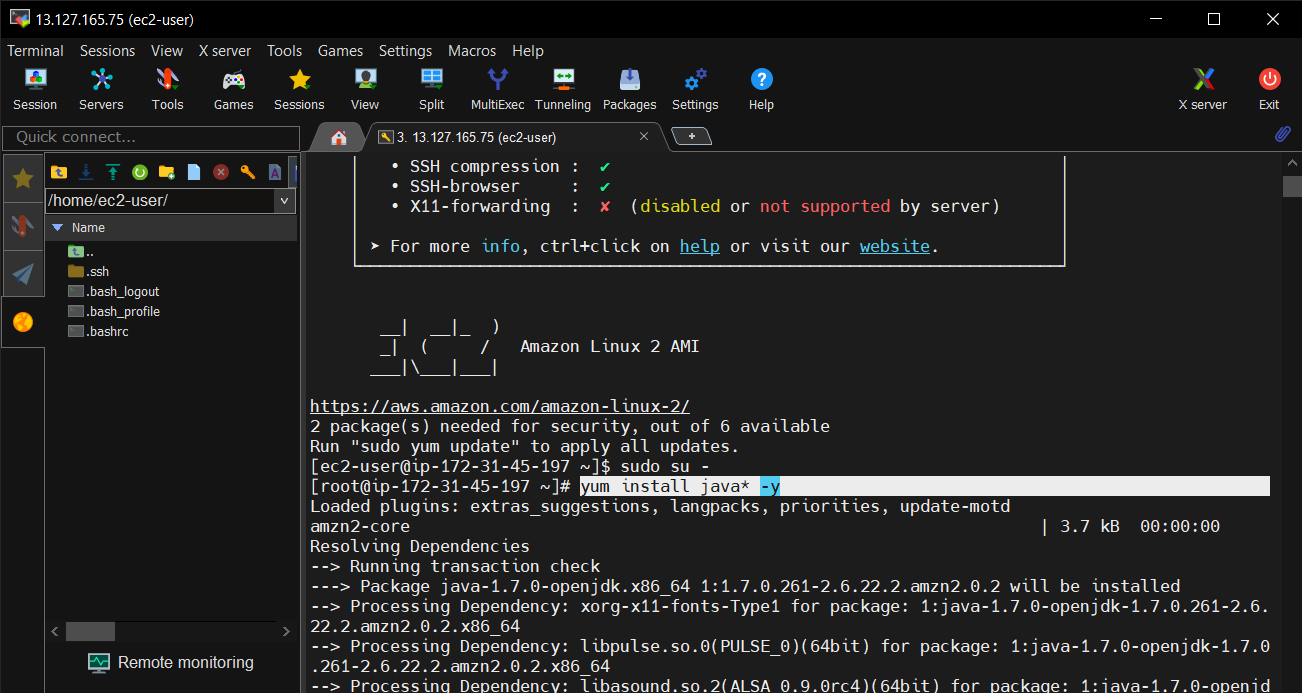
**Jenkins EC2 Instance:**

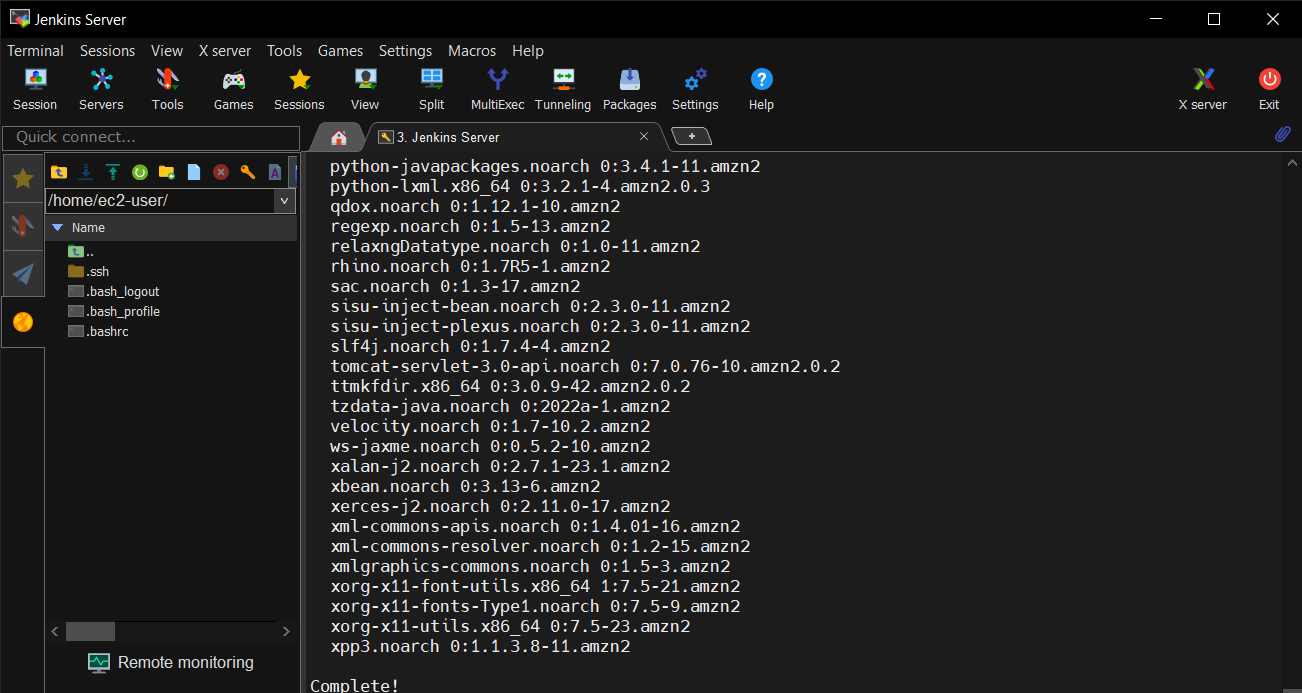
In This web-server the Jenkins is installed and all the configurations and settings are updated to run the website smoothly or we can say that CI/CD pipeline is used for this**.**

**Installing Jenkins inside the Jenkins EC2 Instance:**

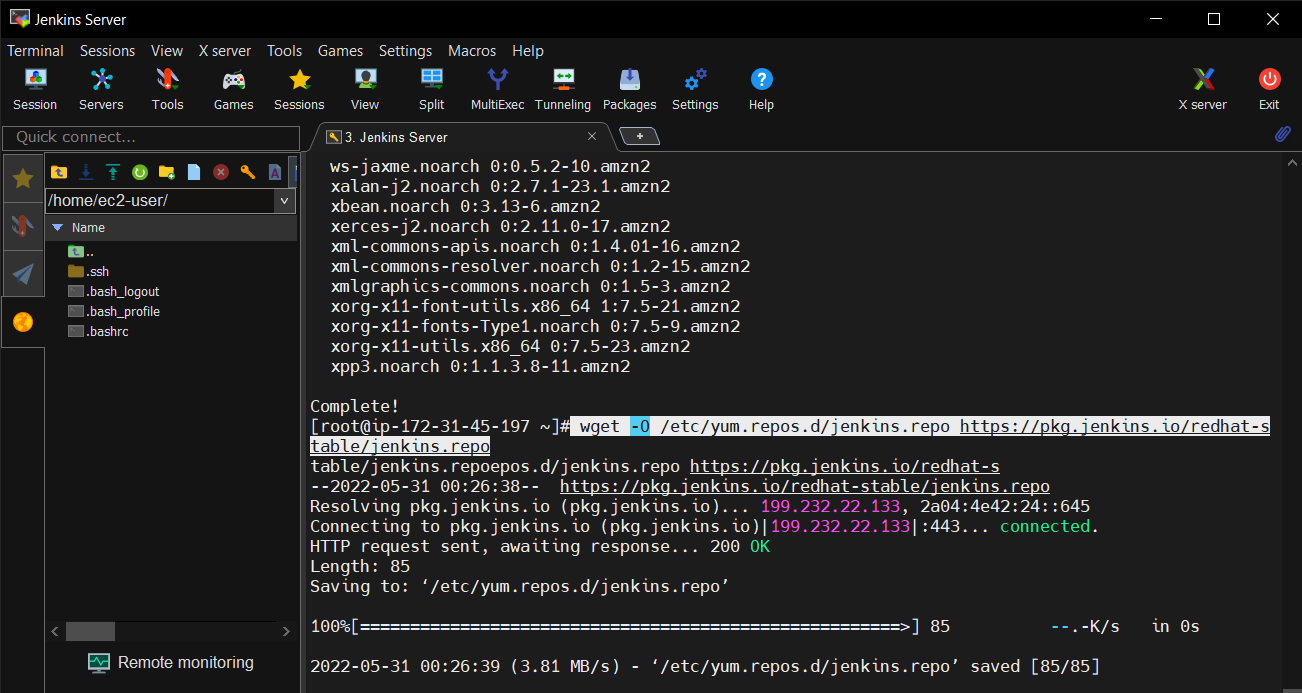
We login the Jenkins EC2 instance with the Putty Server.

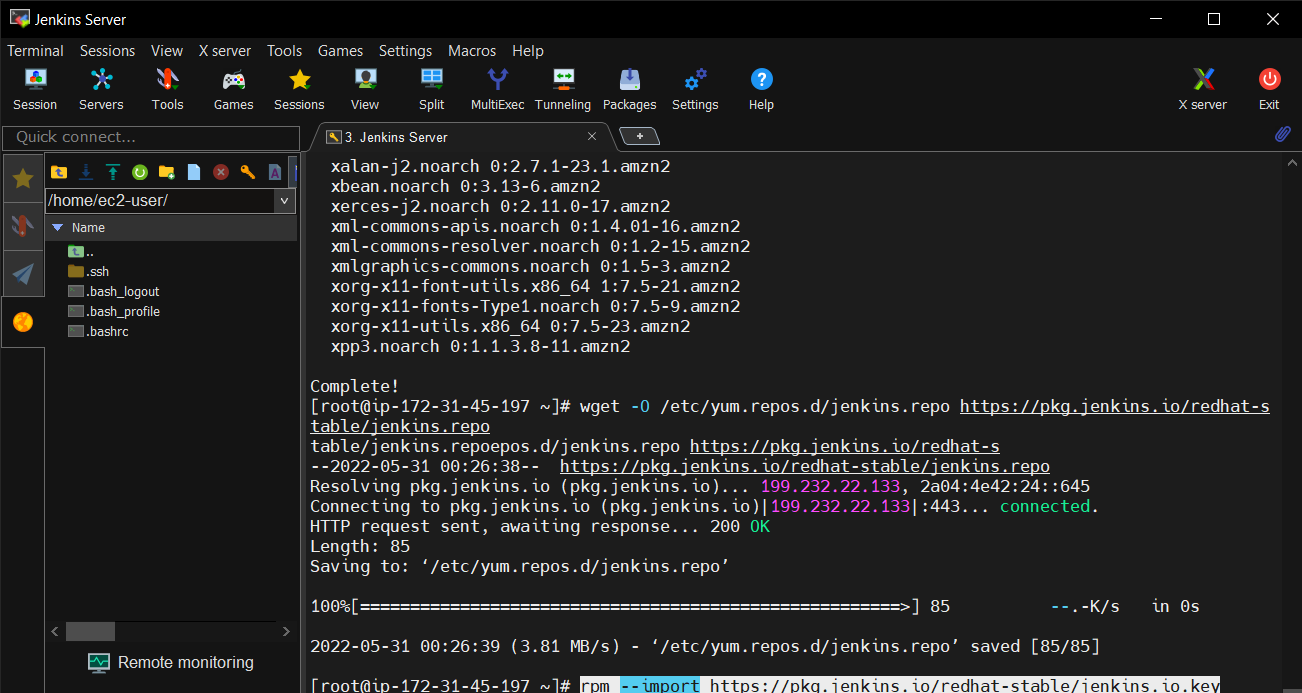
After login, install the Java inside the Jenkins server.

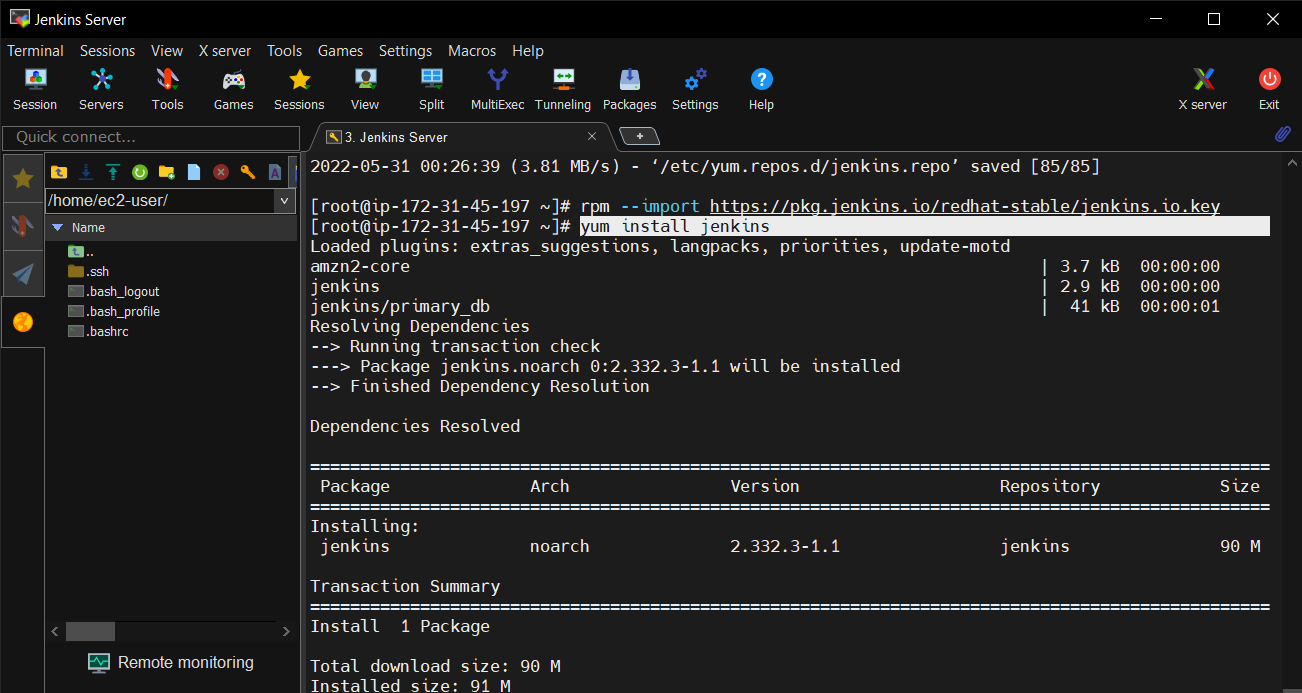




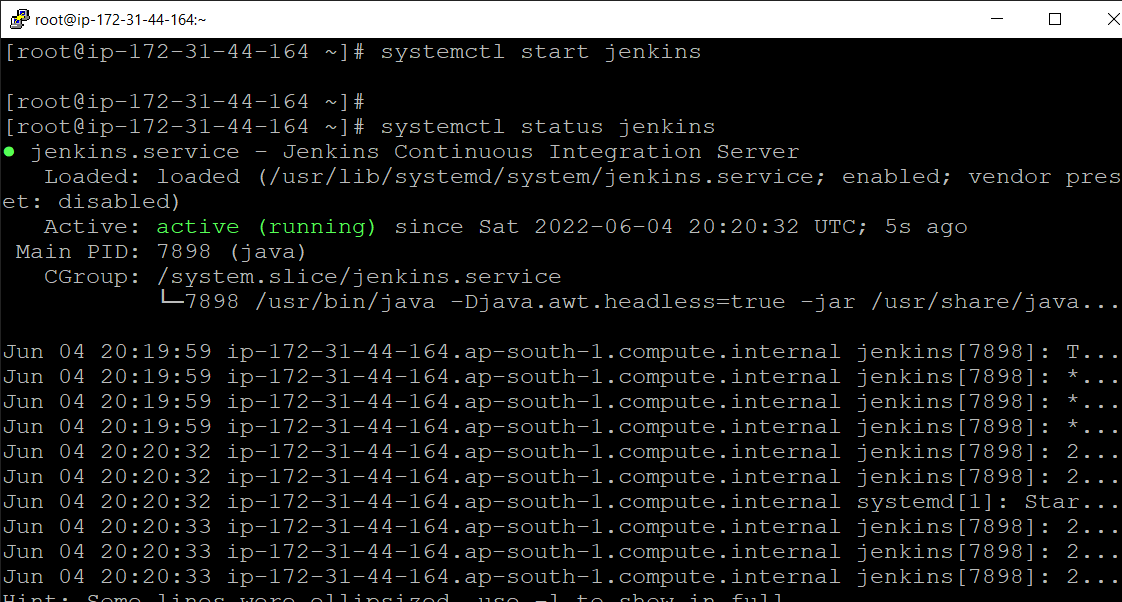
Then installing the Jenkins inside it.



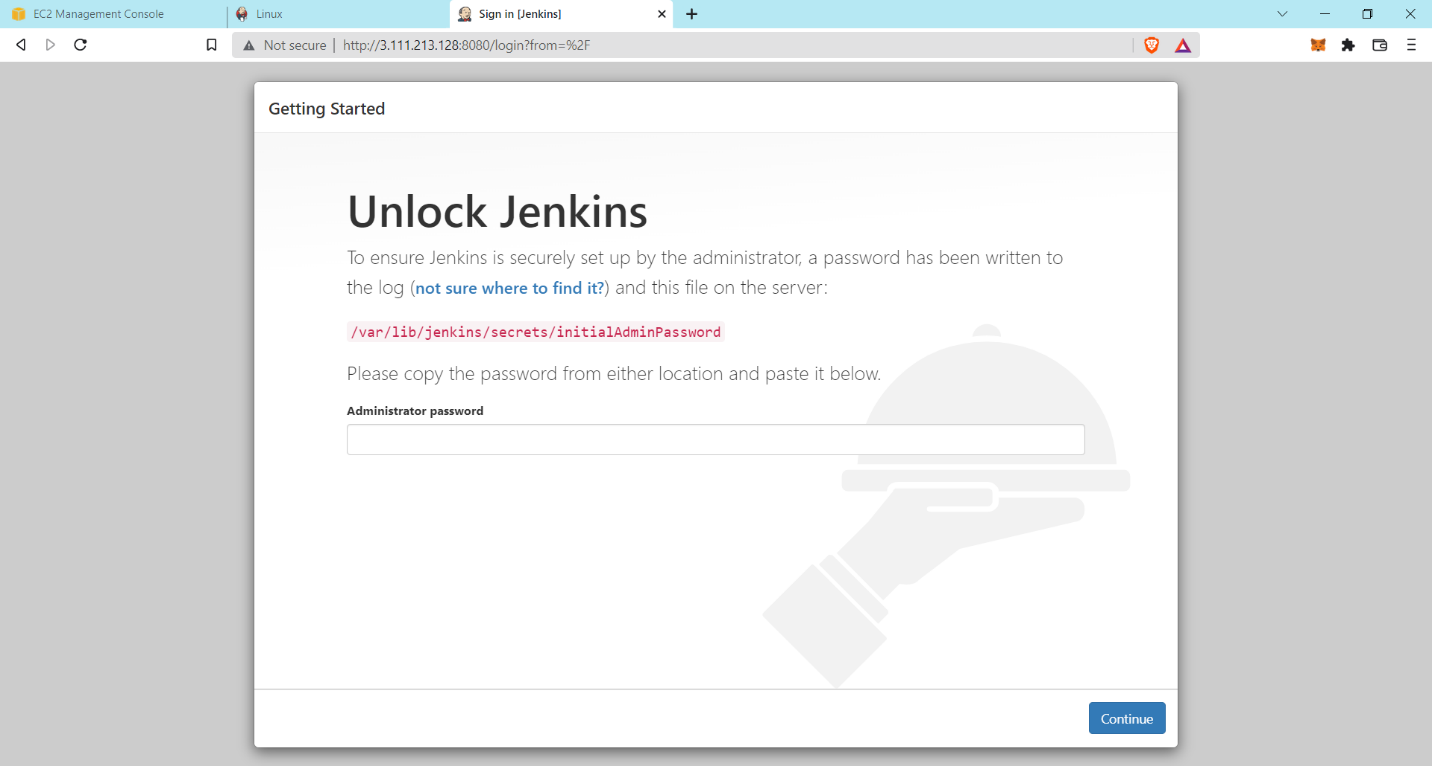




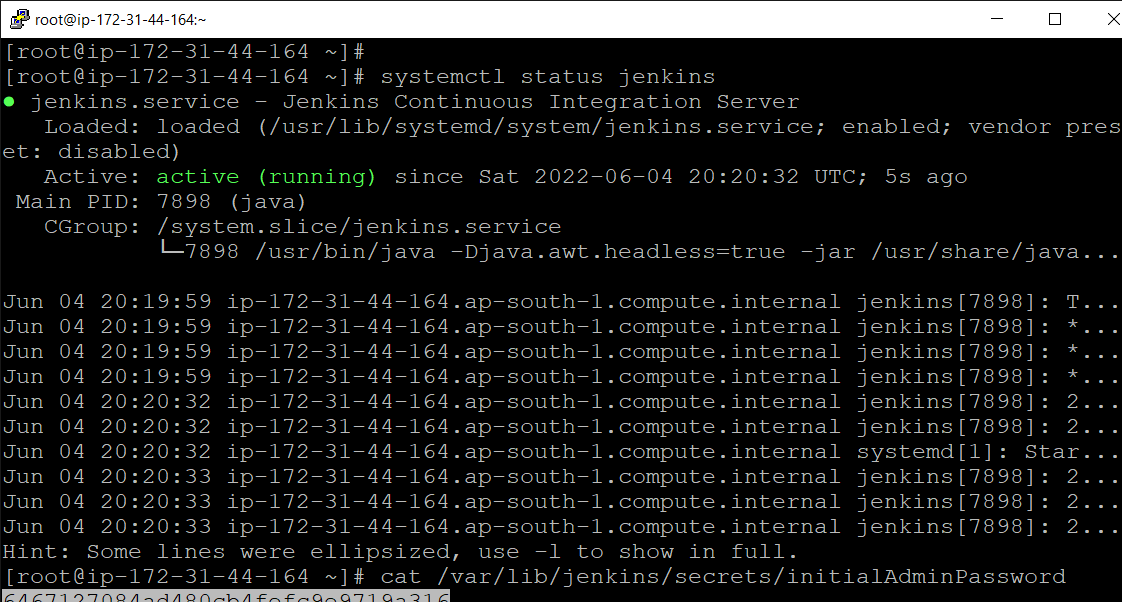
Now start the Jenkins with **“systemctl start Jenkins”** and then enable it with the command **“systemctl enable jenkins”.**



Then after, access the Jenkins in the browser. For that, copy the Jenkins EC2 instance public IP from the AWS and paste it in the browser.

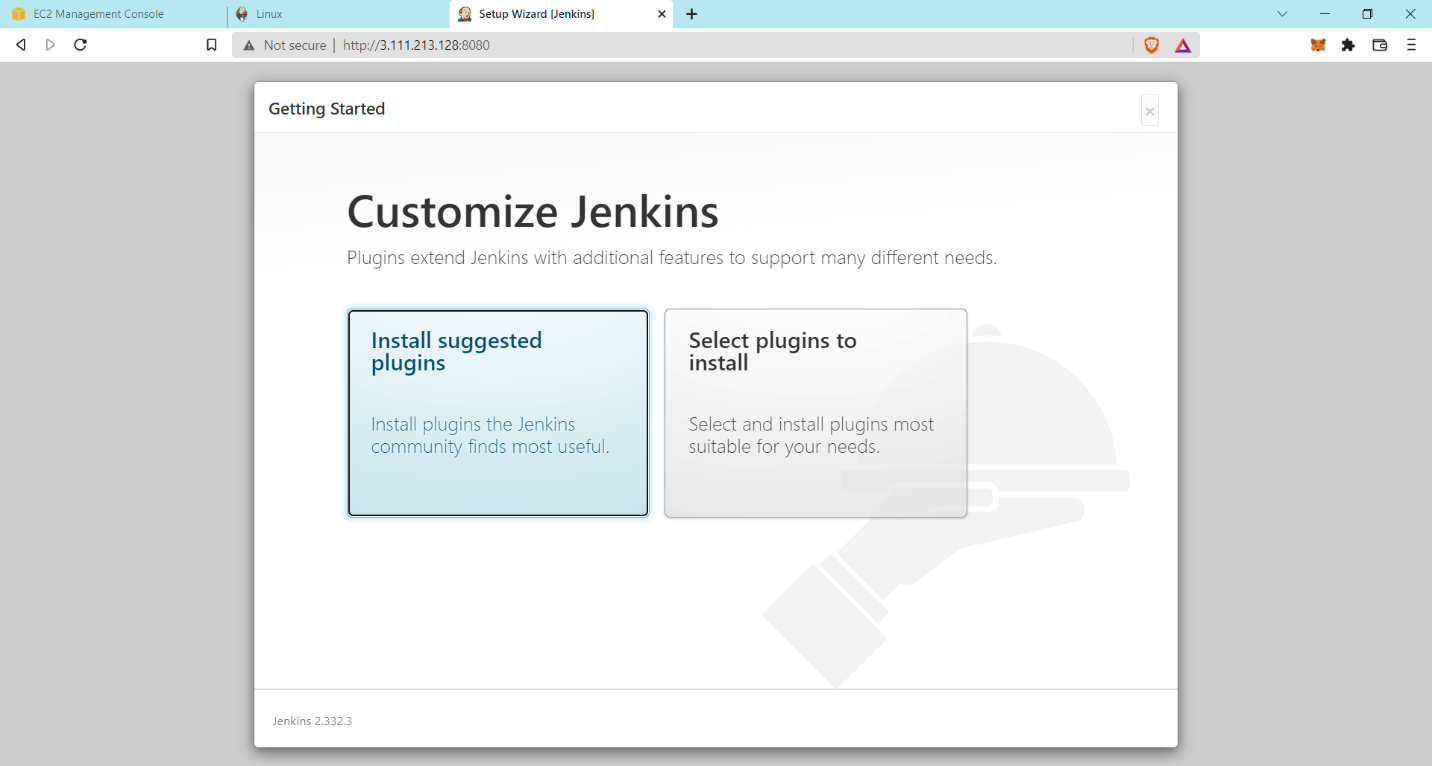


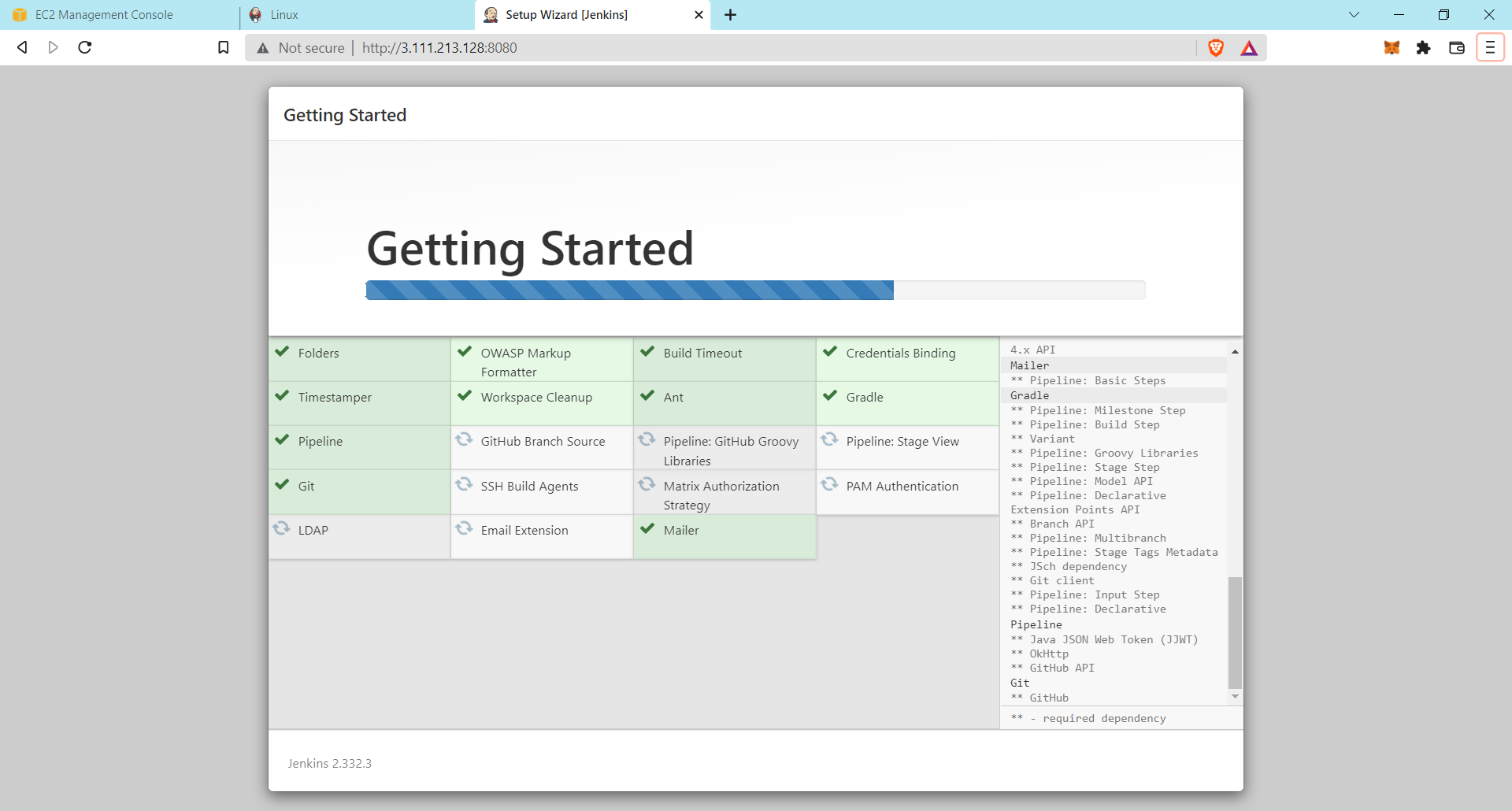
Here generates the administrator password from the above URL in screenshot and paste it in the Jenkins putty server which is shown in the below Screenshot.



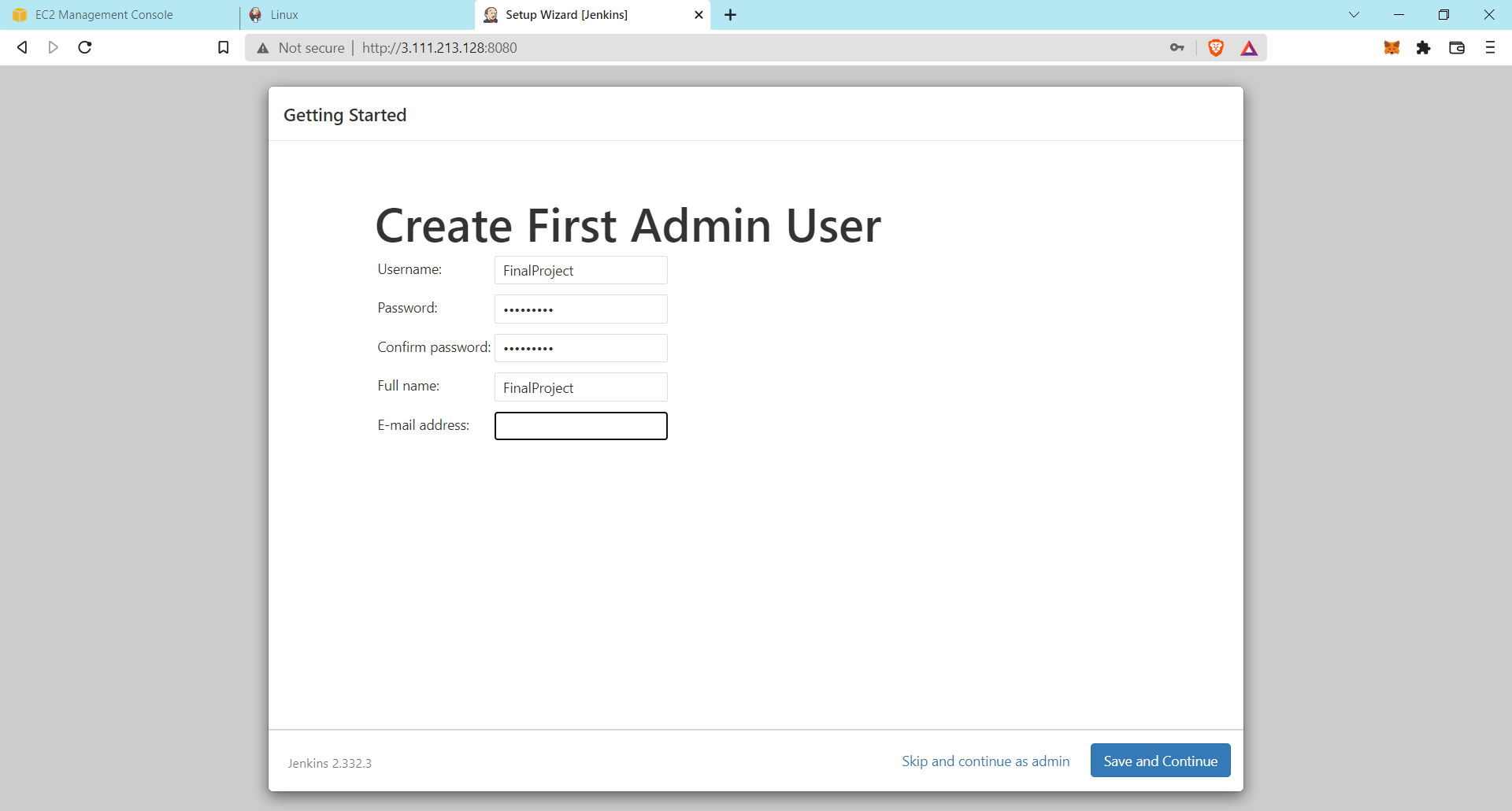


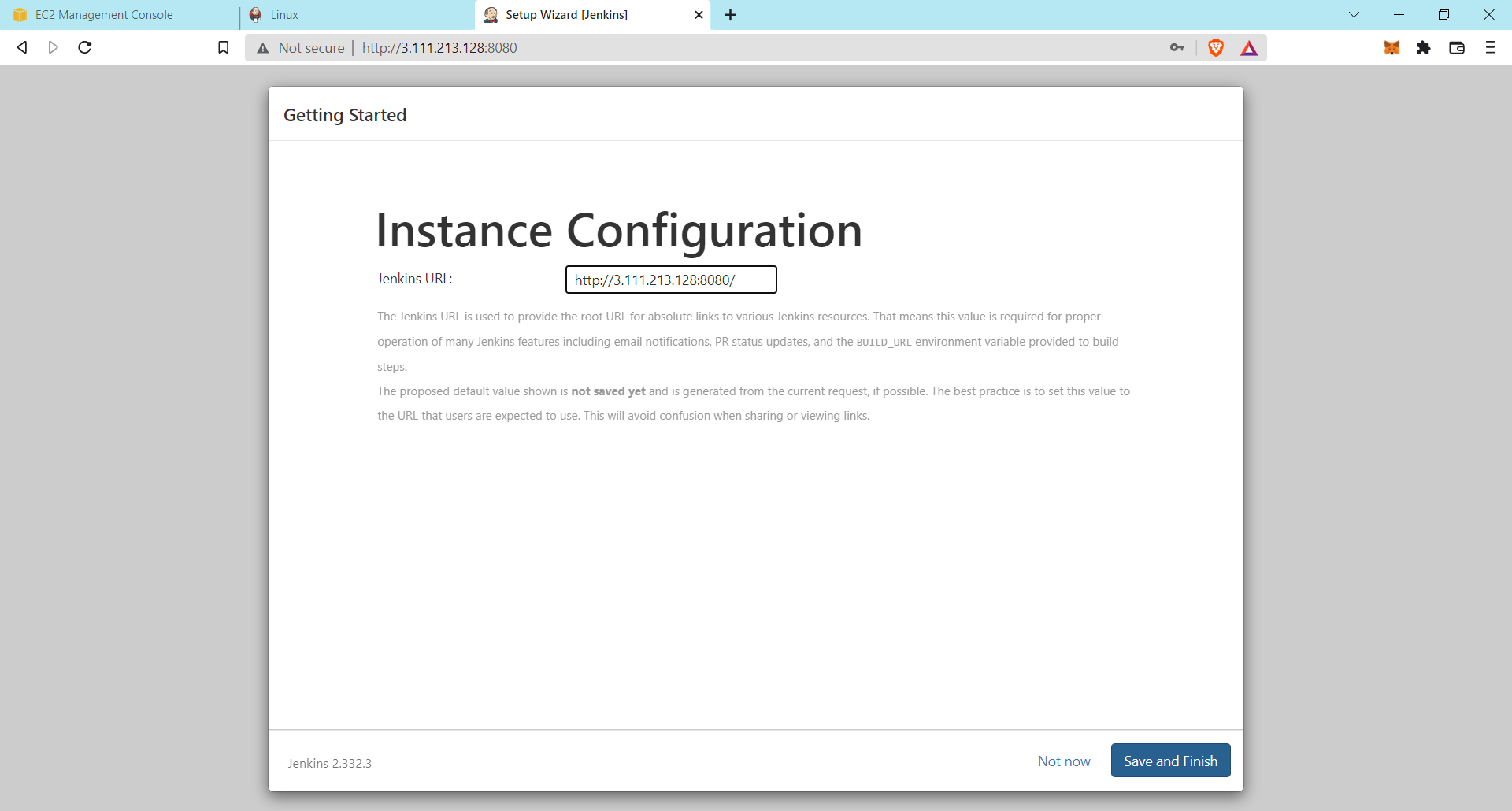
Then install suggested plugins.

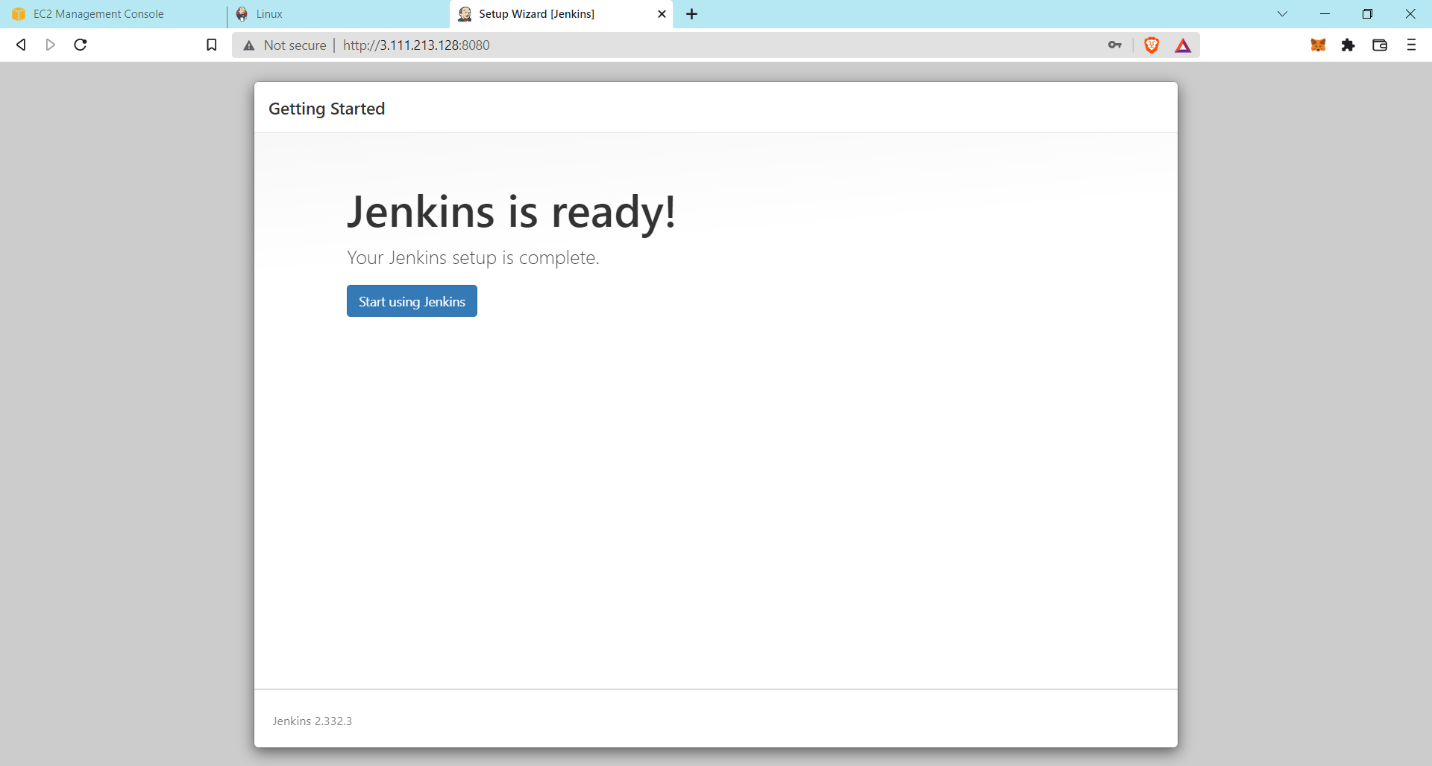




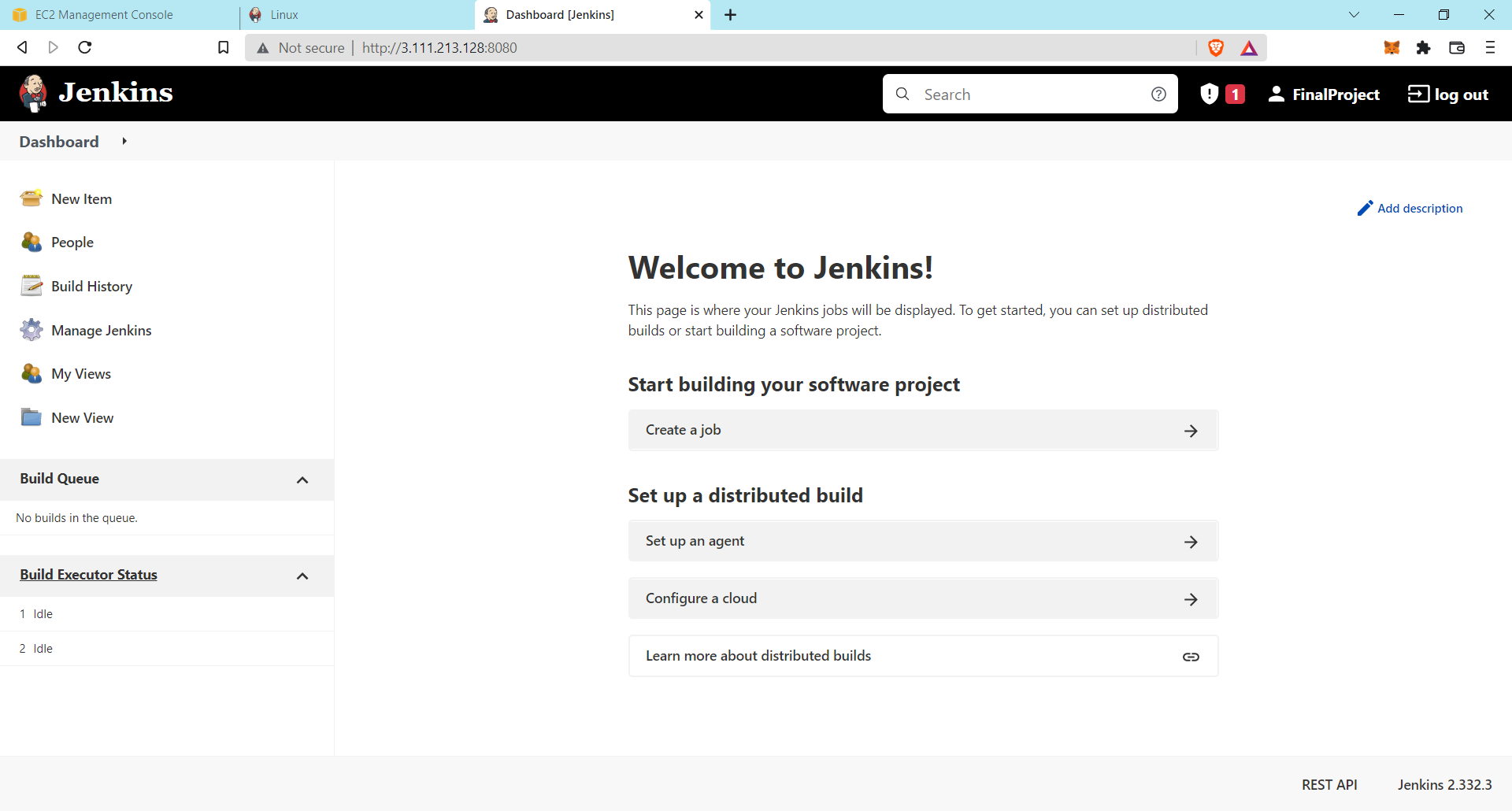
After installing, create first admin user, save and finish it and start the Jenkins.





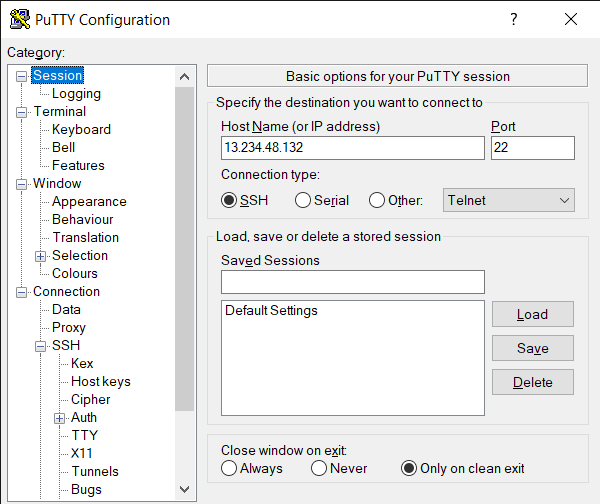


Now your Jenkins is start.

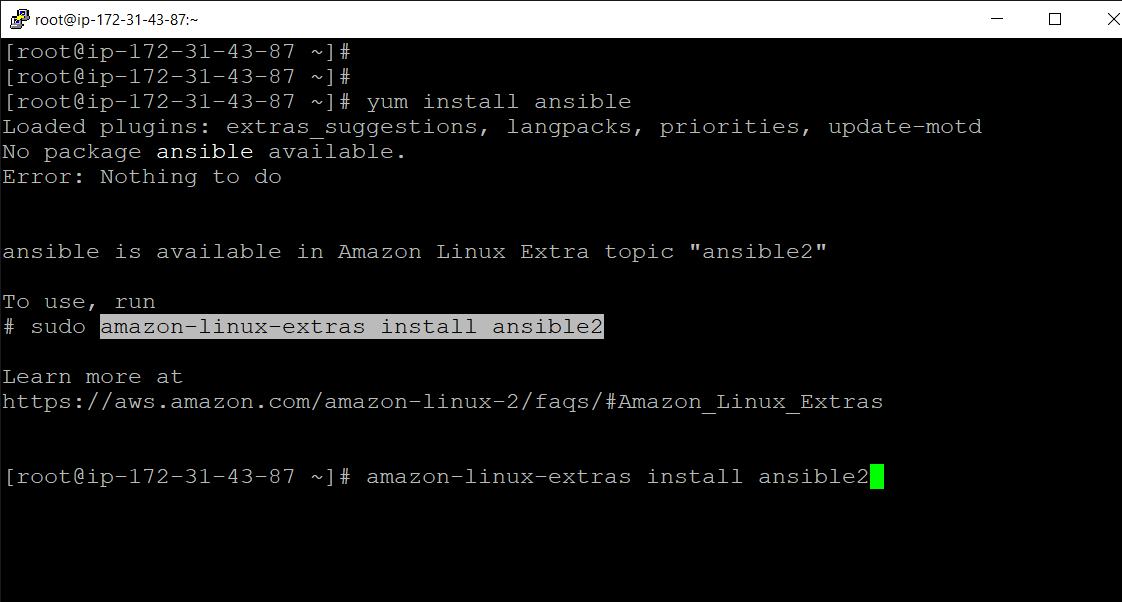


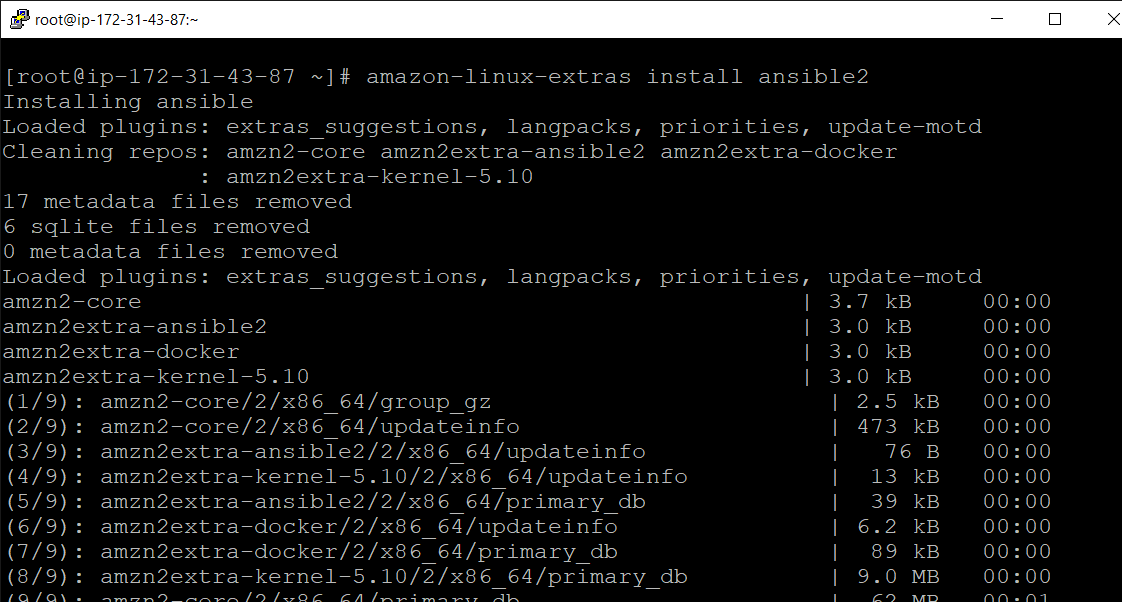
**Ansible EC2 Instance:**

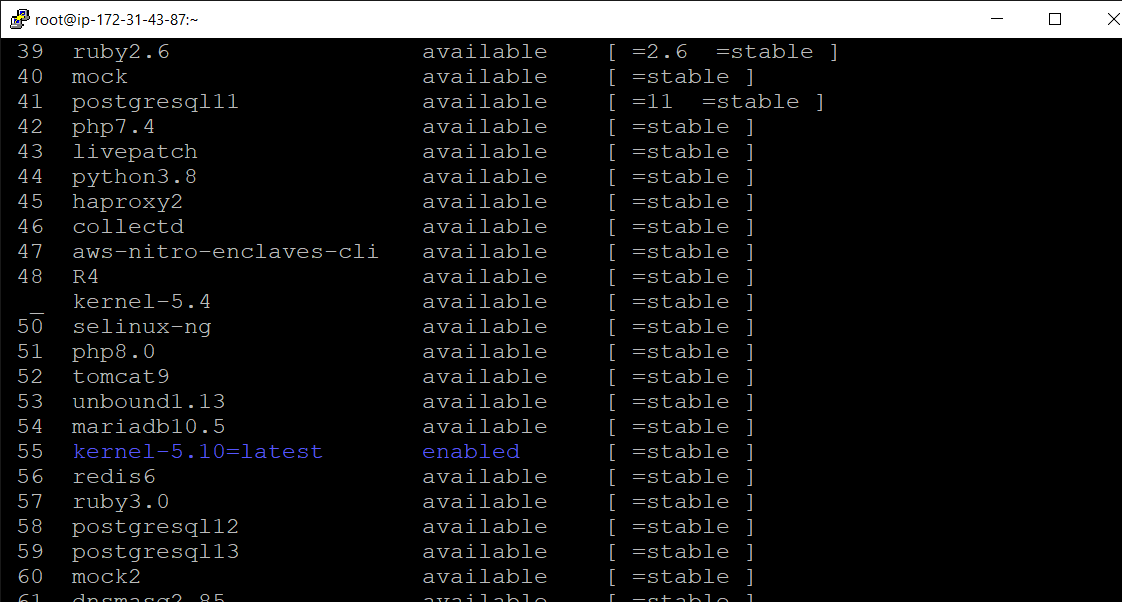
In this web server the ansible is installed and all the settings are to be done.

****

Installing Ansible using the command “amazon-linux-extras install ansible2”.

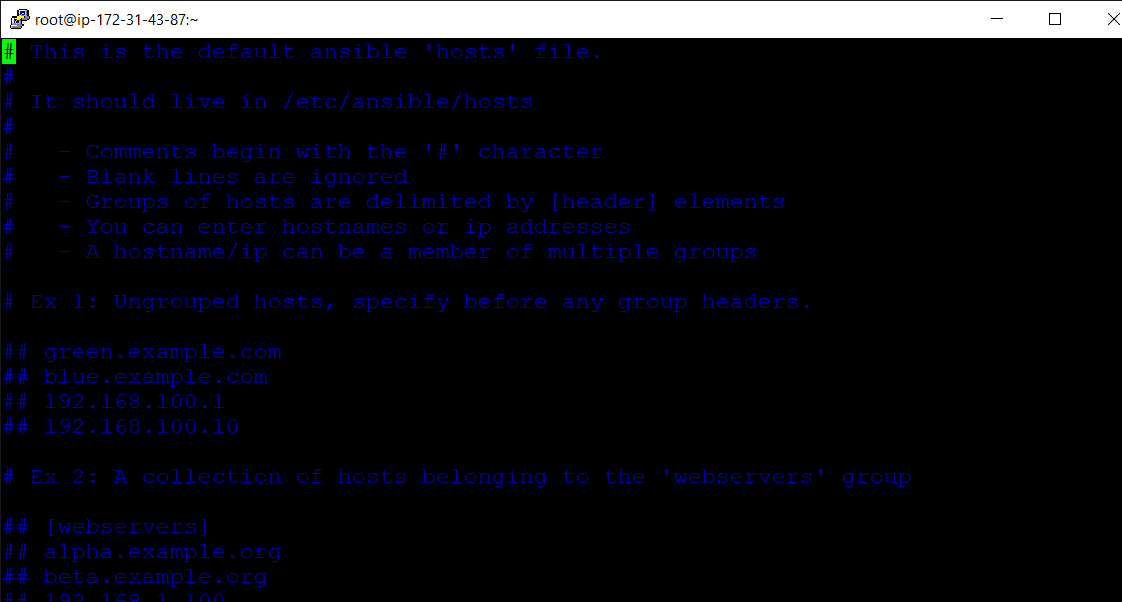
****

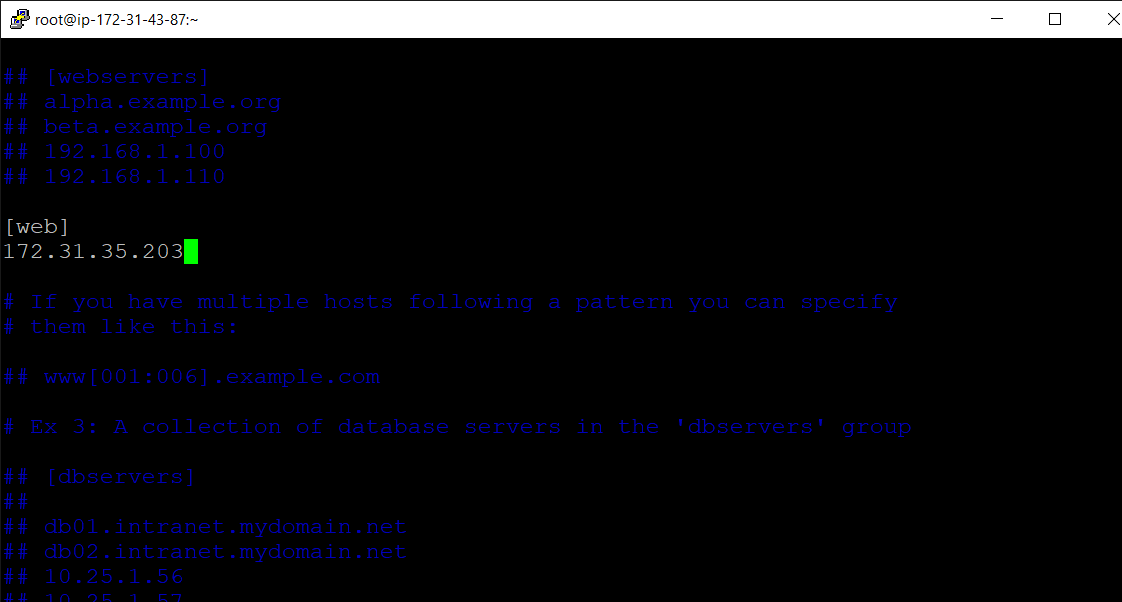
****

****

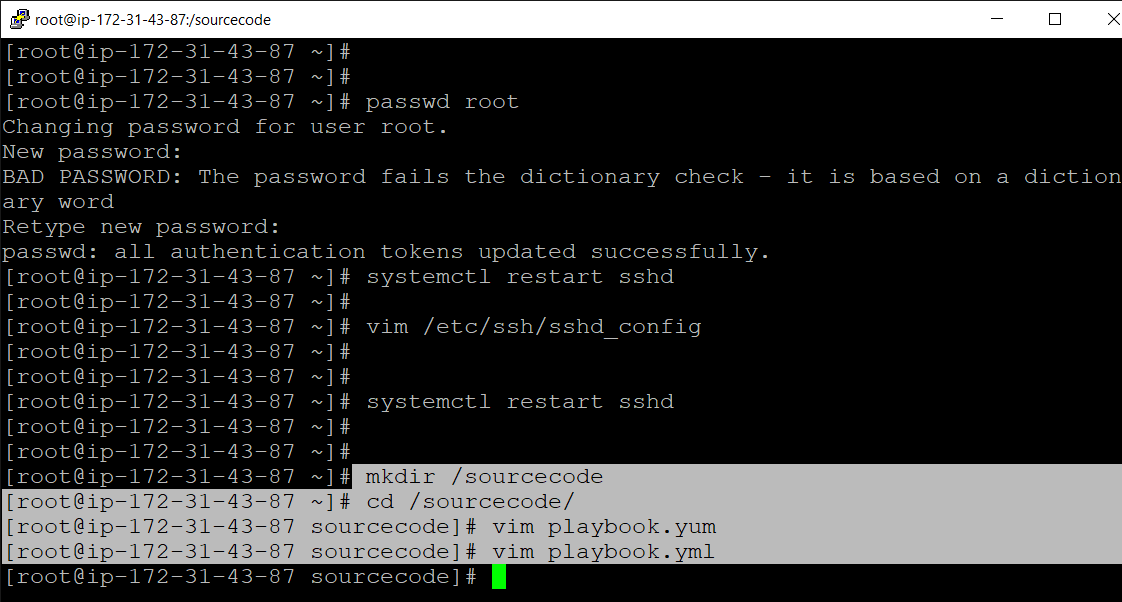
Now add the web-server’s private ip to ansible’s hosts directory. See the screenshots below.

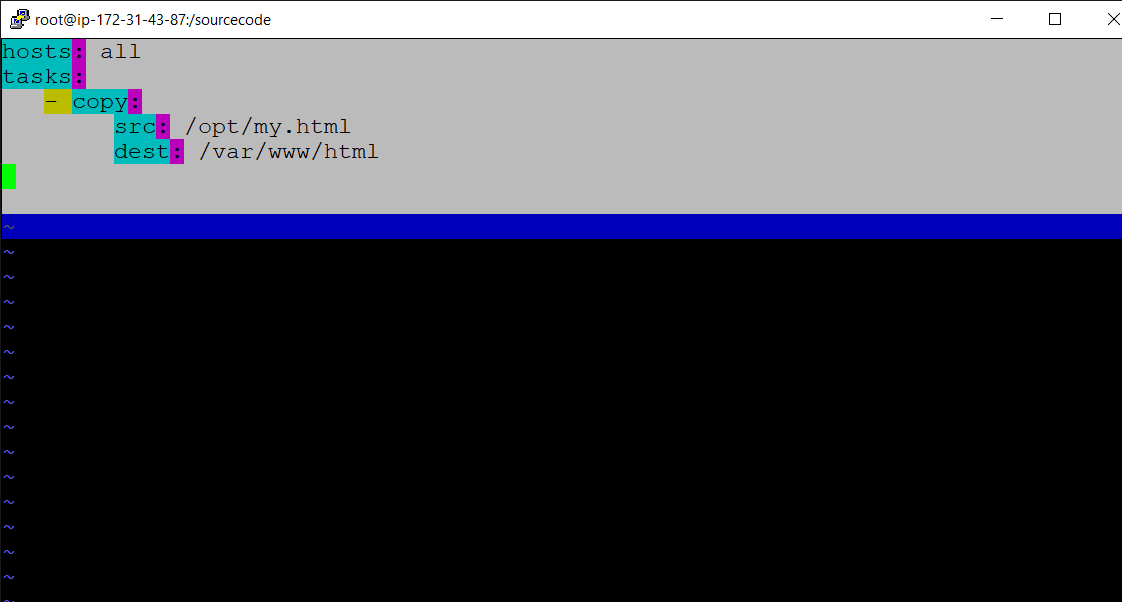






After that we create a playbook yml file inside the ansible with the name **“playbook”.** Then write the yml script which is given below in screenshot and take the **.index** file of the website which is present in the GitHub account.



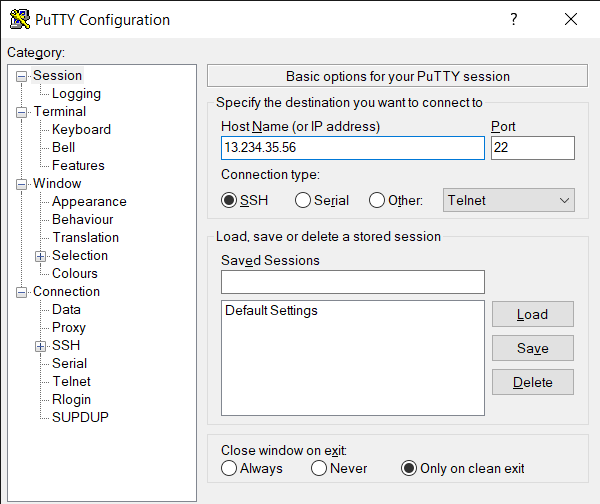


**Web-server EC2 Instance:**

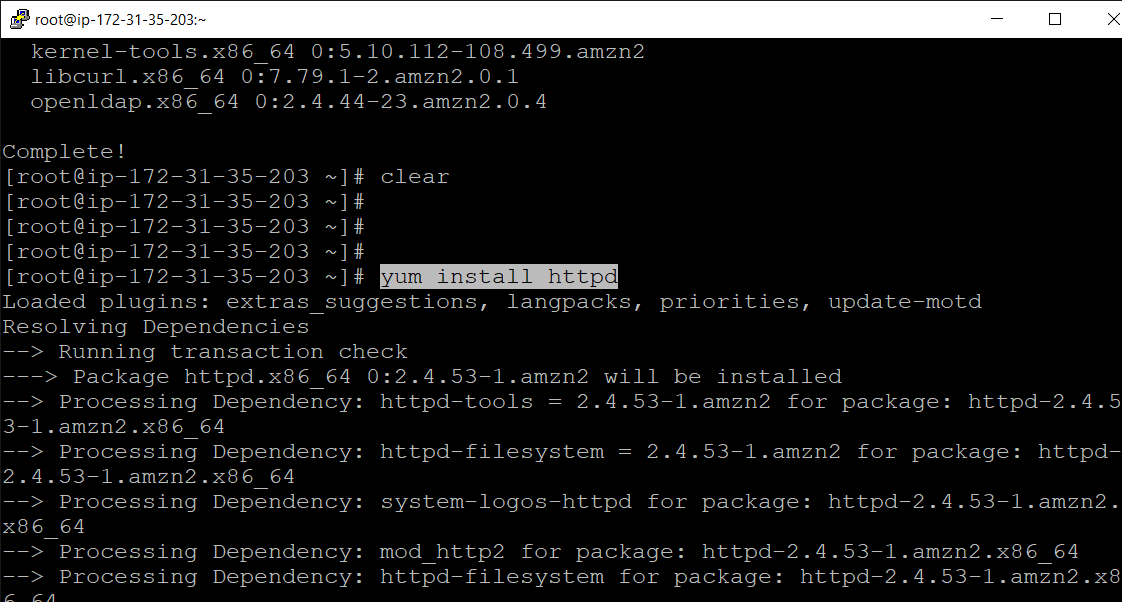
This is the web-server in which website is hosting and working on it.

**Installing a Web-server and Apache**

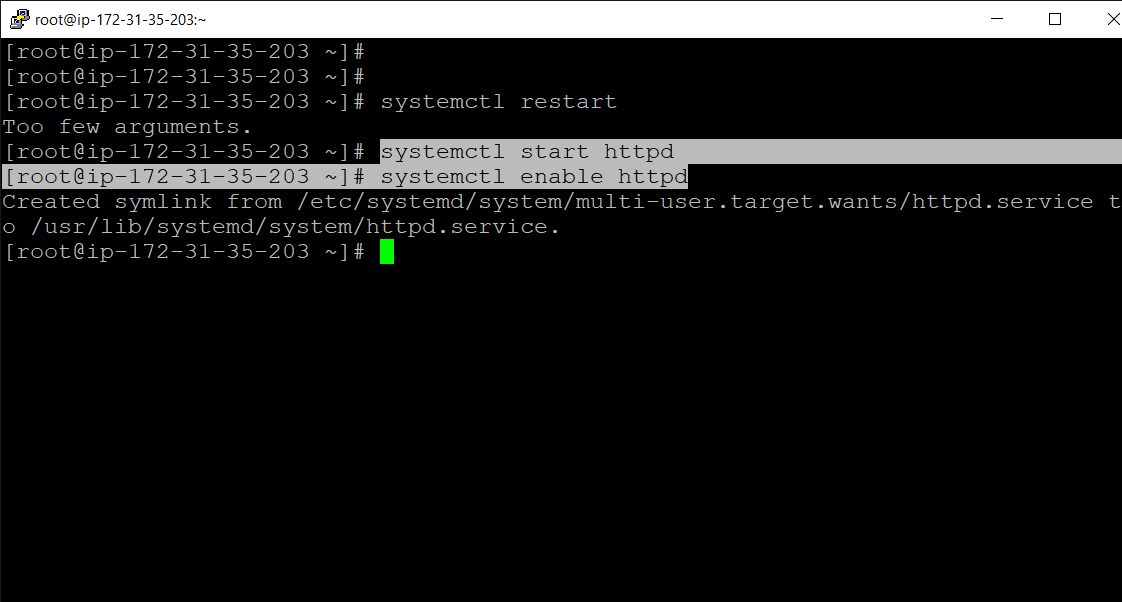
Copy the public ip from the Aws instance of web-server and paste it in **putty**.



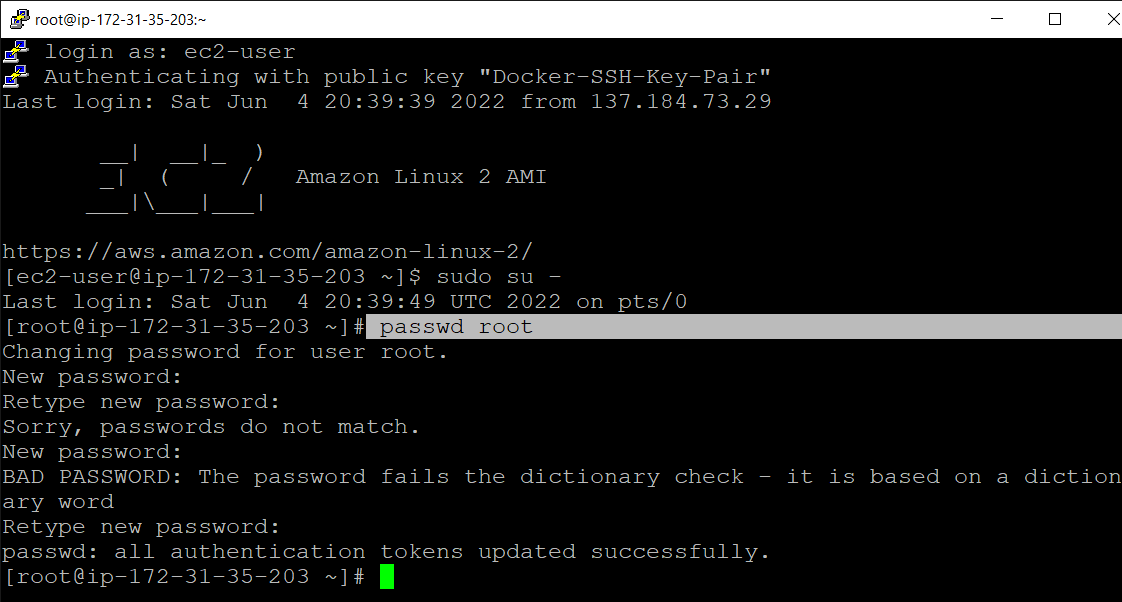
After that, install the Apache server in it using **“yum install httpd”.**



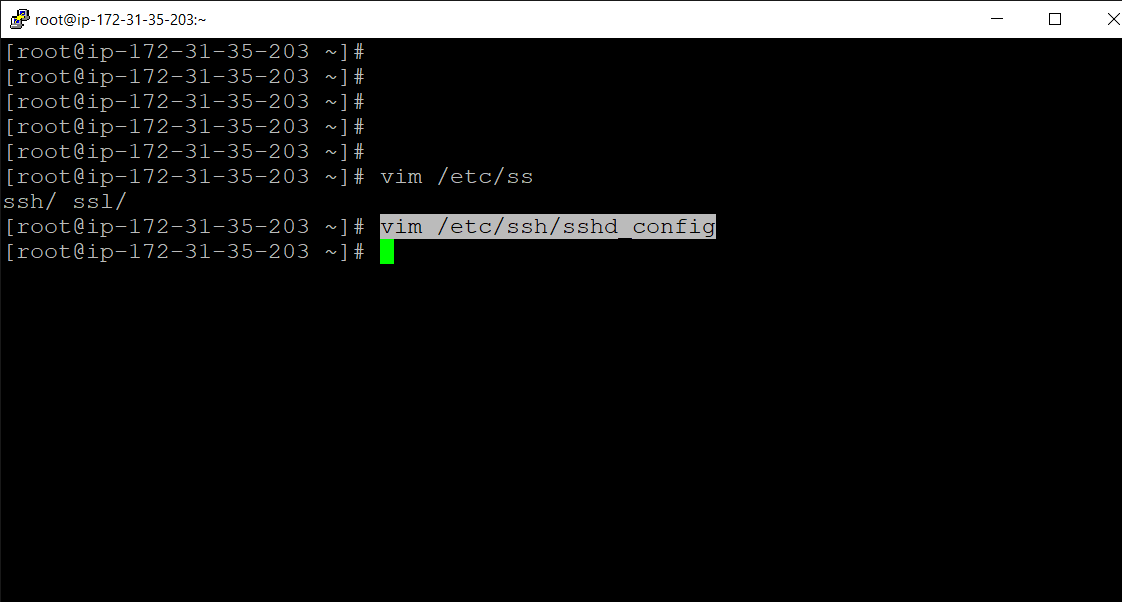
Then start the httpd and enable httpd.

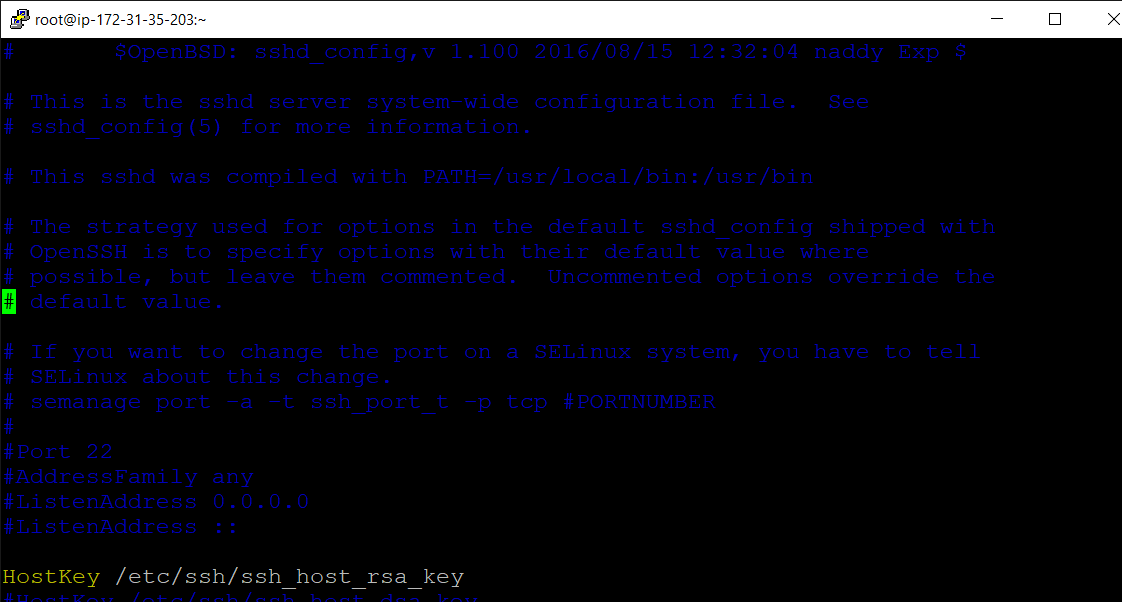


Then change the root password

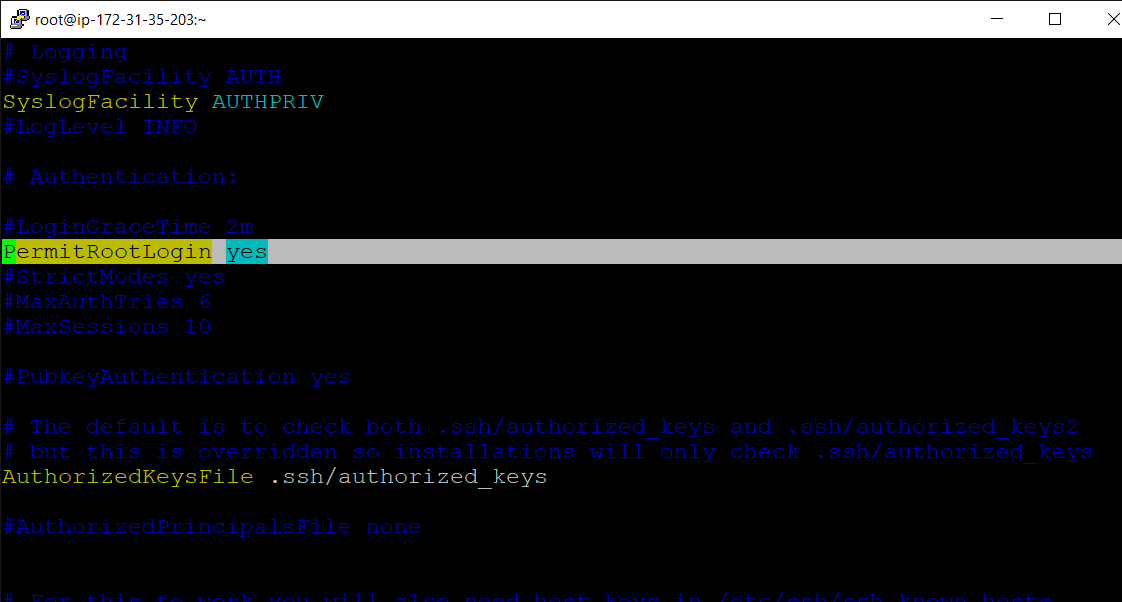


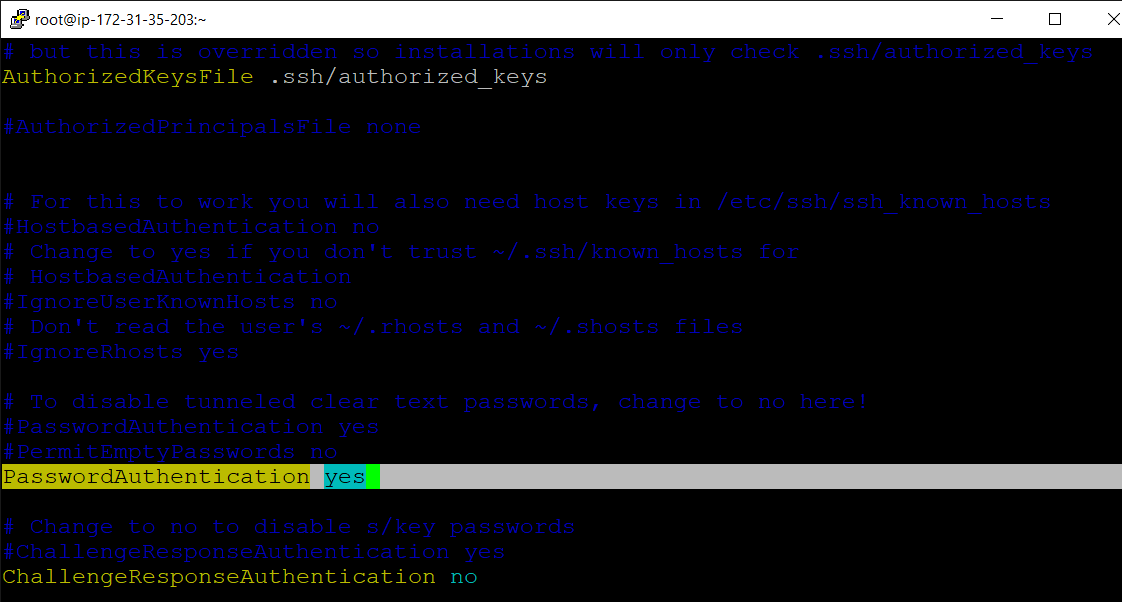
Now here change its sshd config file to access it root authentication password permissions for **password less authentication**.



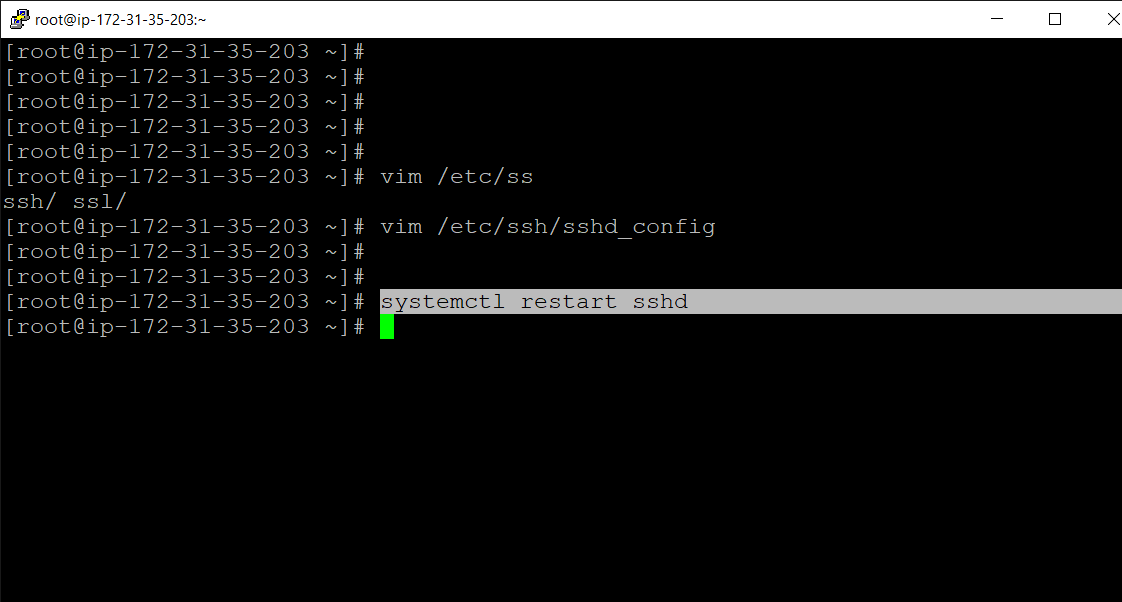


Enable the **PermitRootLogin YES.**



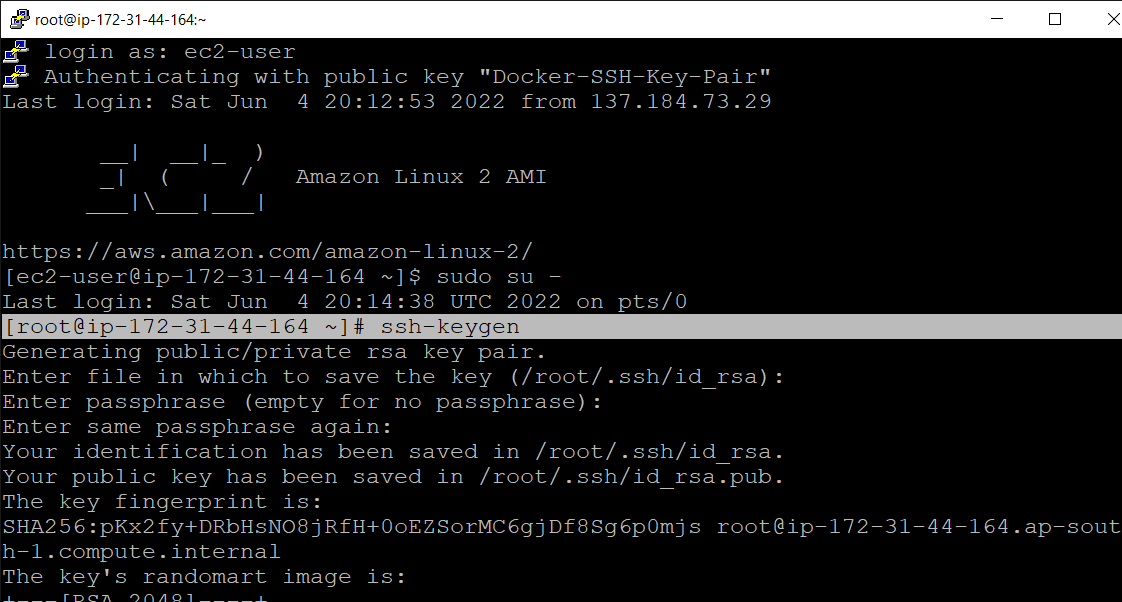


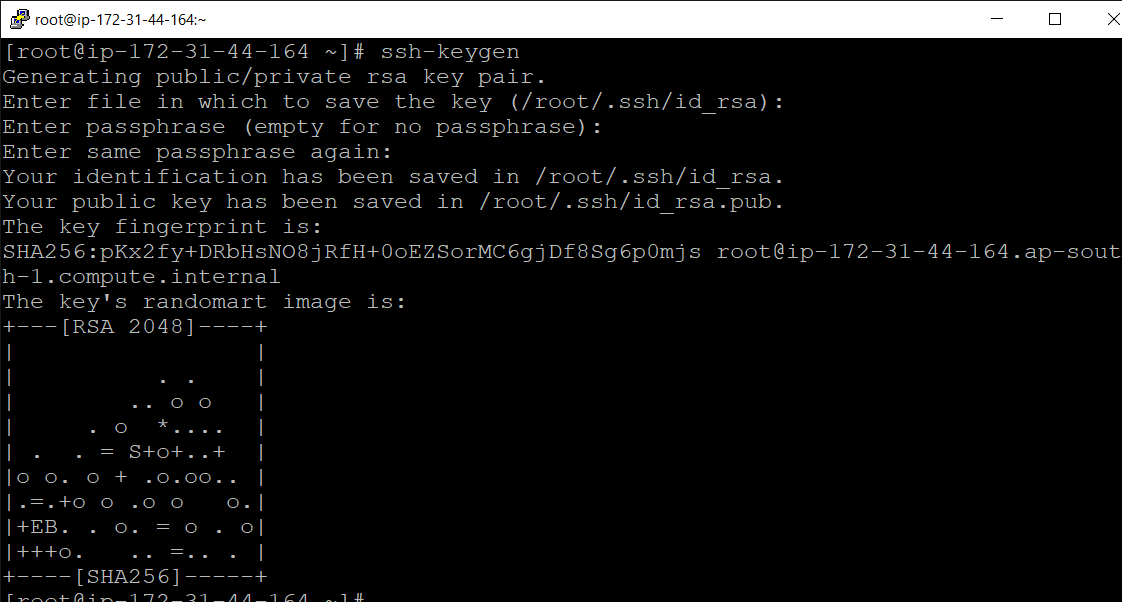
Then restart the sshd service.



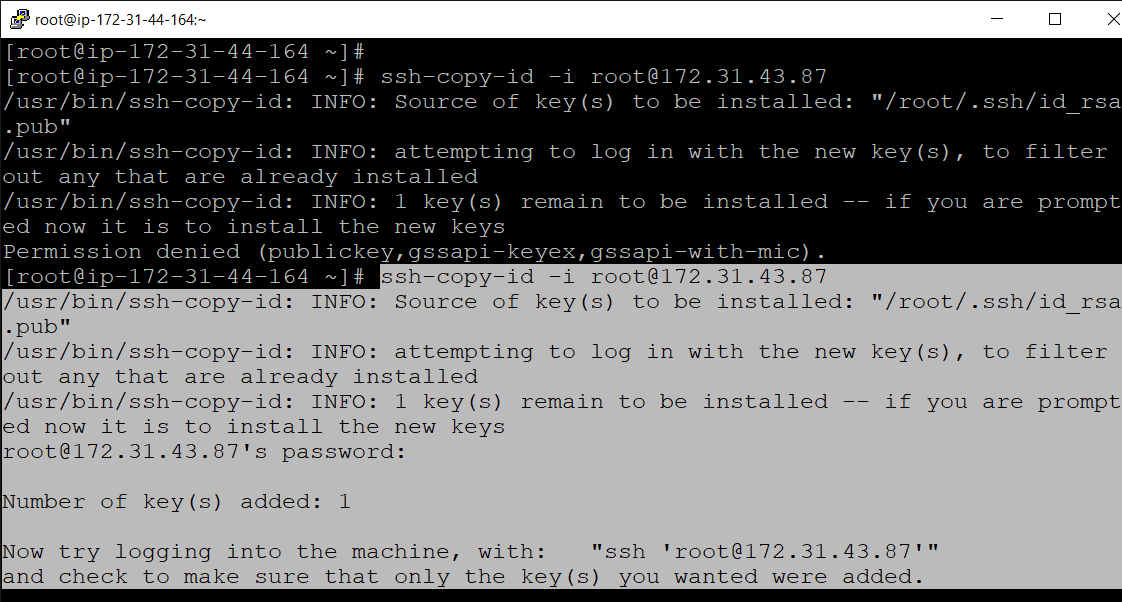
**Creating a password less authentication between Jenkins, Ansible and web-server.**

Generate the **“ssh-keygen”** in the Jenkins and ansible.

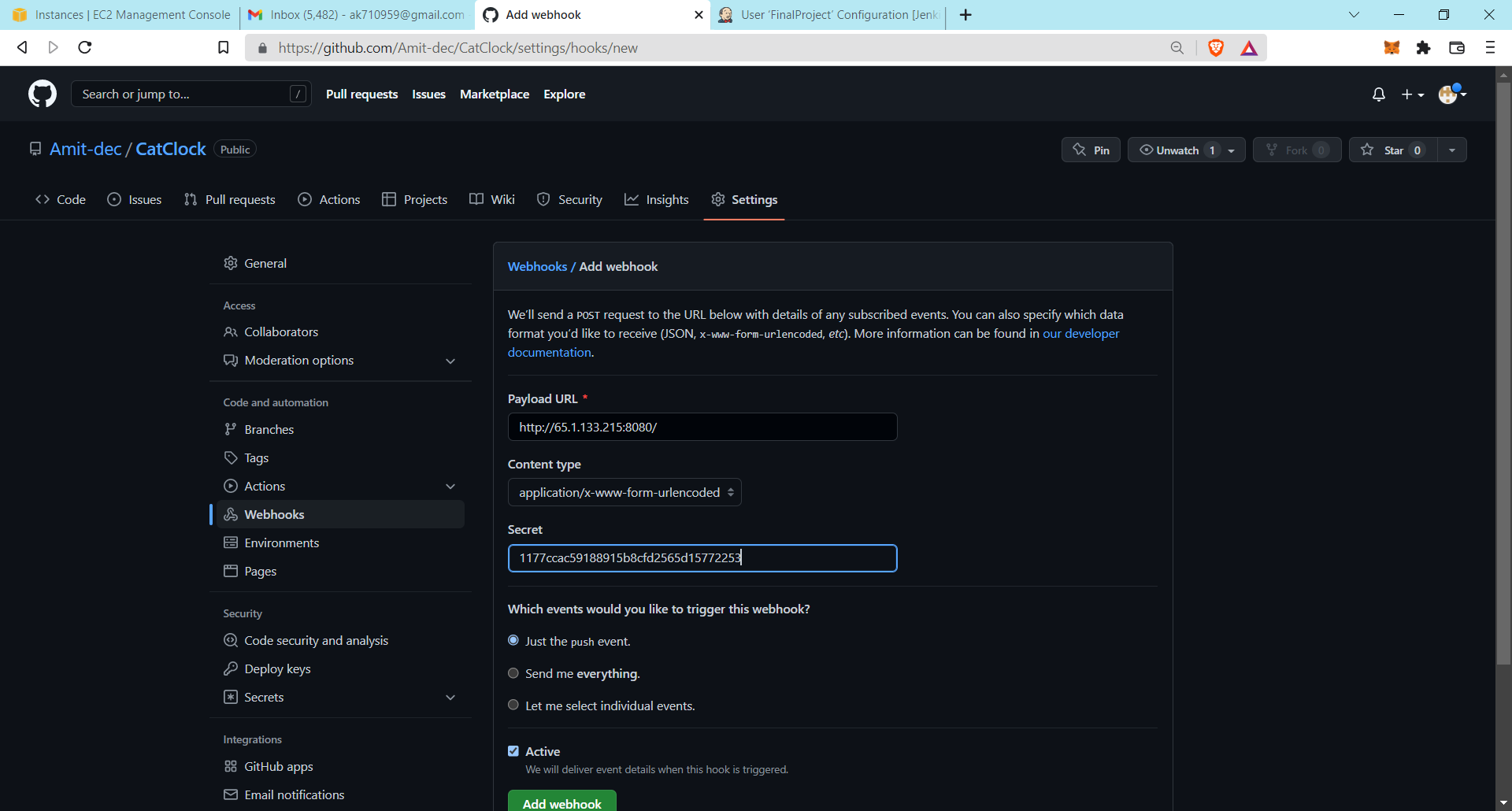


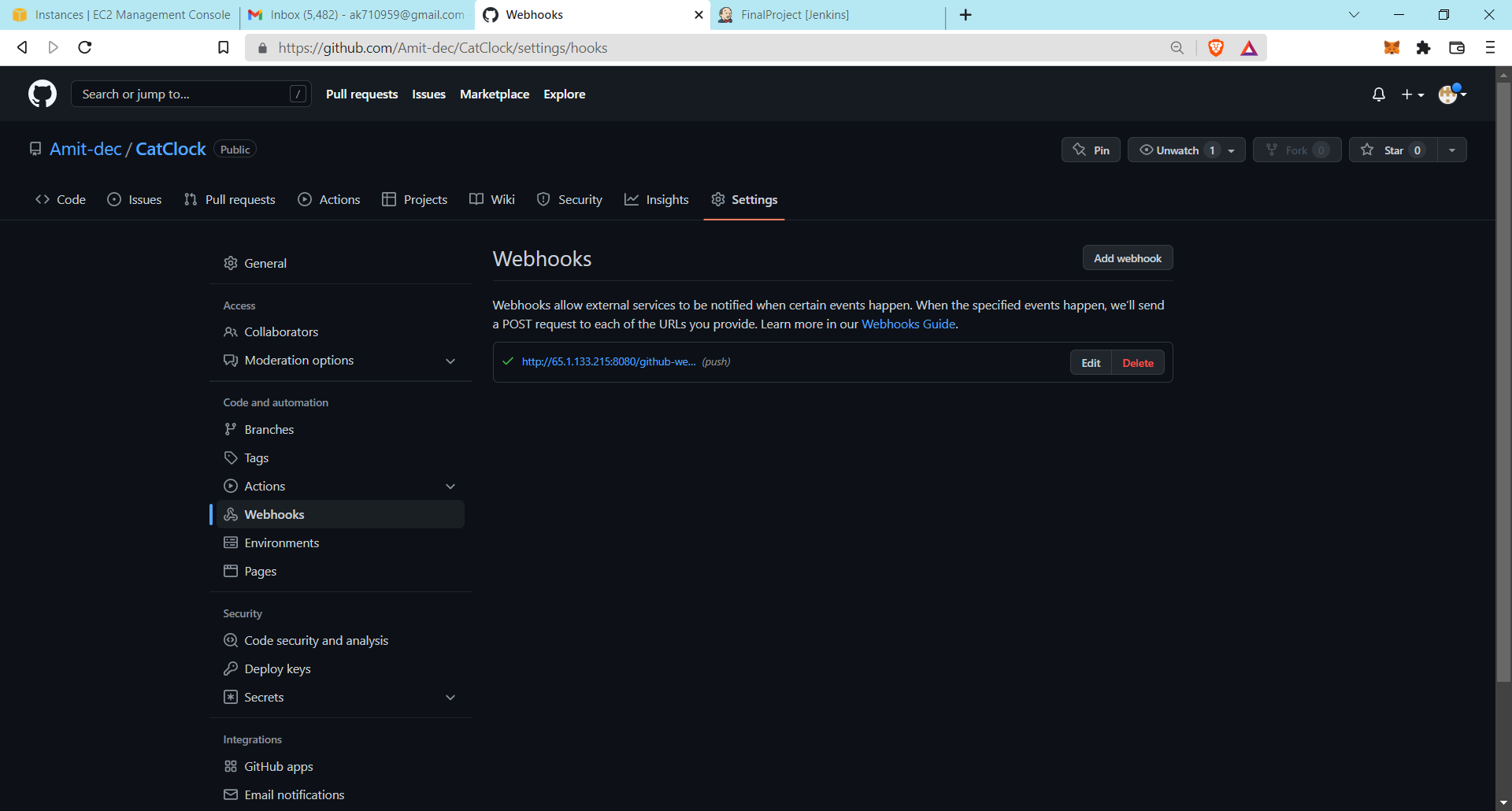


* Then check that the password less connection is working in between the it using the command “**ssh-copy-id -i root@172.31.43.87”** (the private ip the web-server/ansible server).
* If the error **“Permission denied”** is occur then the connection is failed.
* For resolving this problem set the root password of that server which you want to access and change “sshd\_config” file of that server and restart that server. These steps are performed above.



* Now here we go to GitHub account and integrate with the Jenkins server.
* For that, go to the GitHub account and then go to settings.
* Then click on webhook. Then after copy the Jenkins URL and paste it in the webhook’s payload URL. Select the content type and then paste the secret key and click on Add webhook.

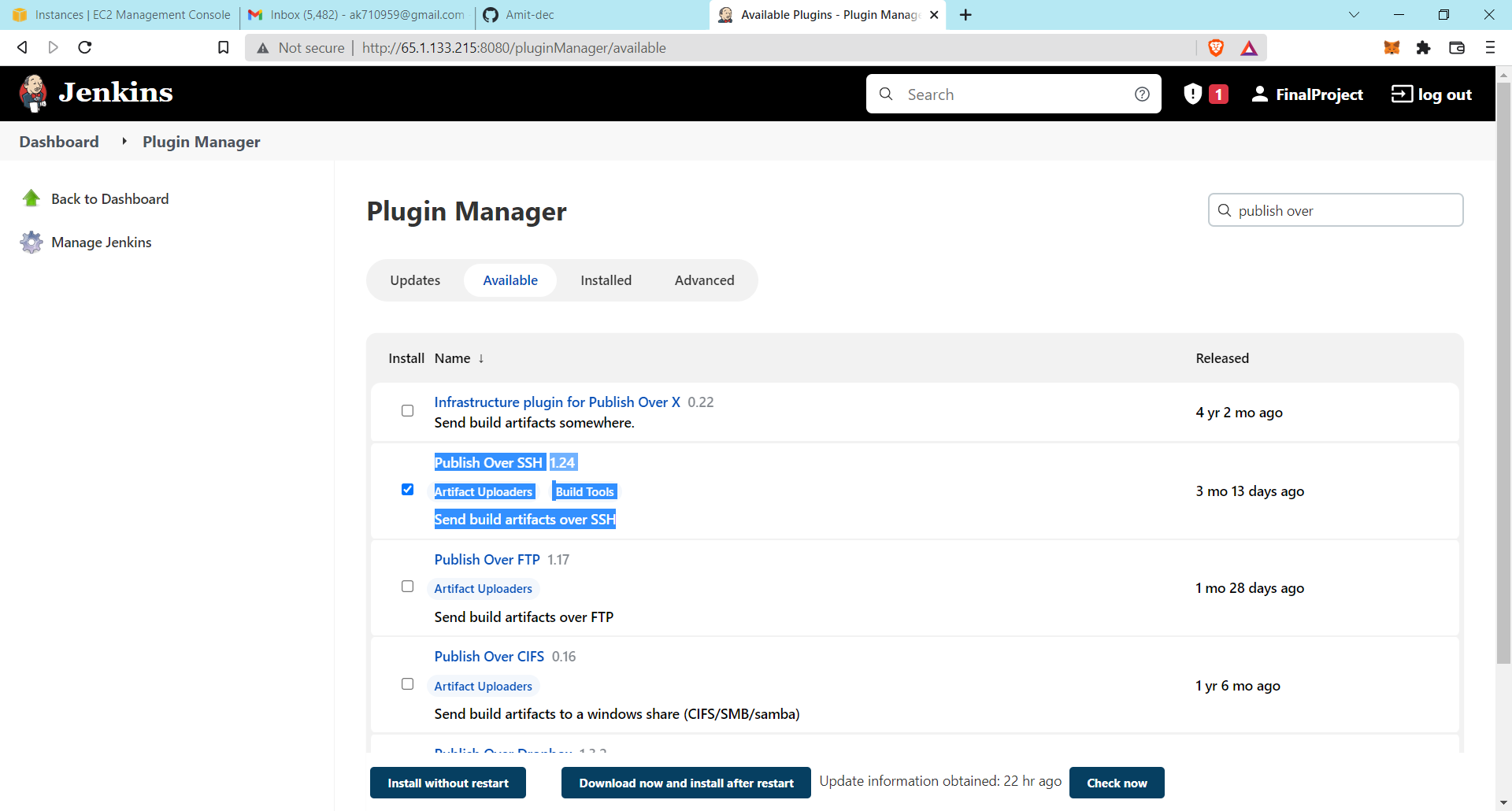


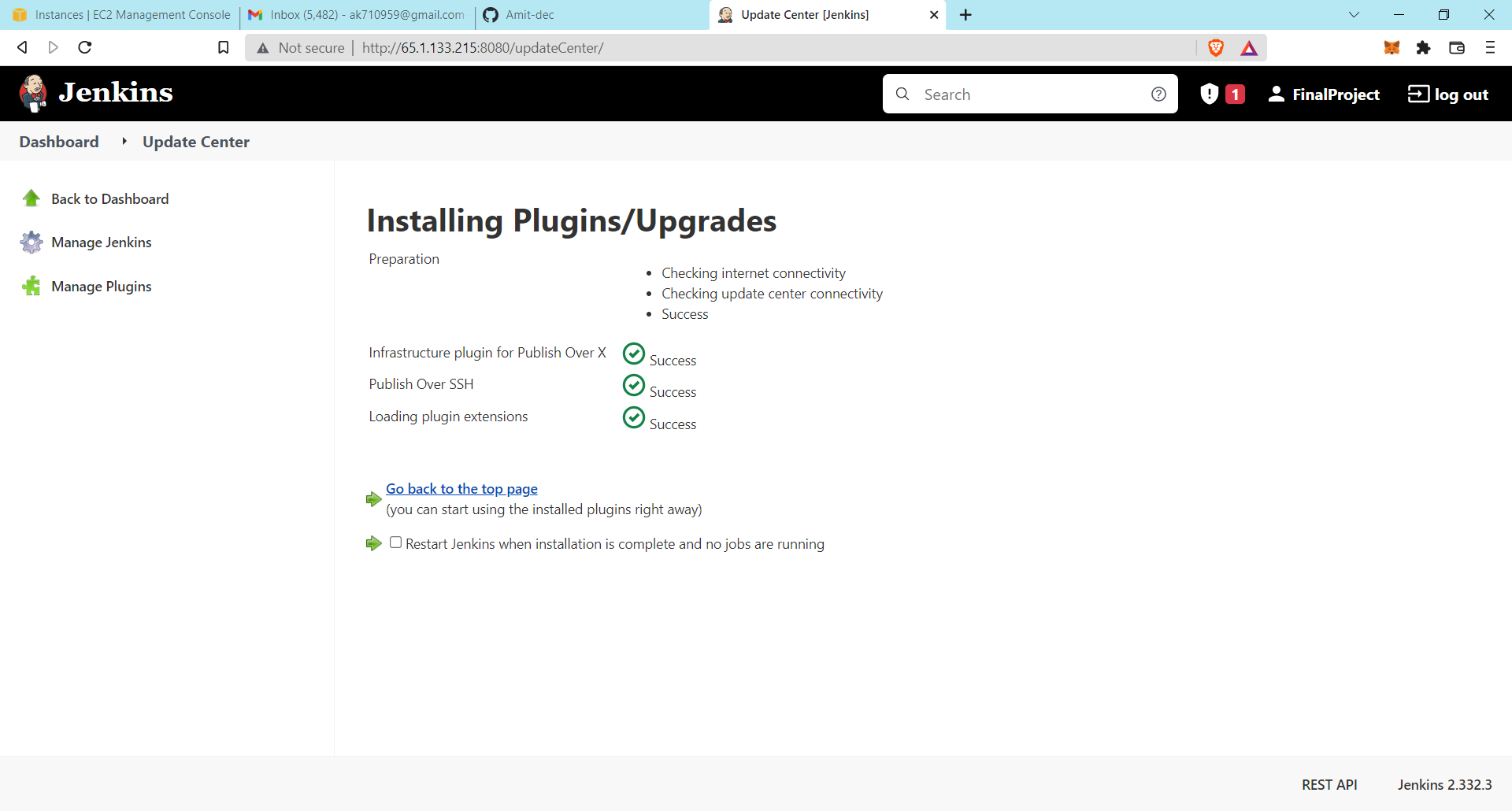


Now we go to the Jenkins and install the “**Publish Over SSH**” plugins.

So, click on the Jenkins Dashboard. Then click on the manage Jenkins.

Then after search in the filter bar and type the publish over ssh and find it and install the it.



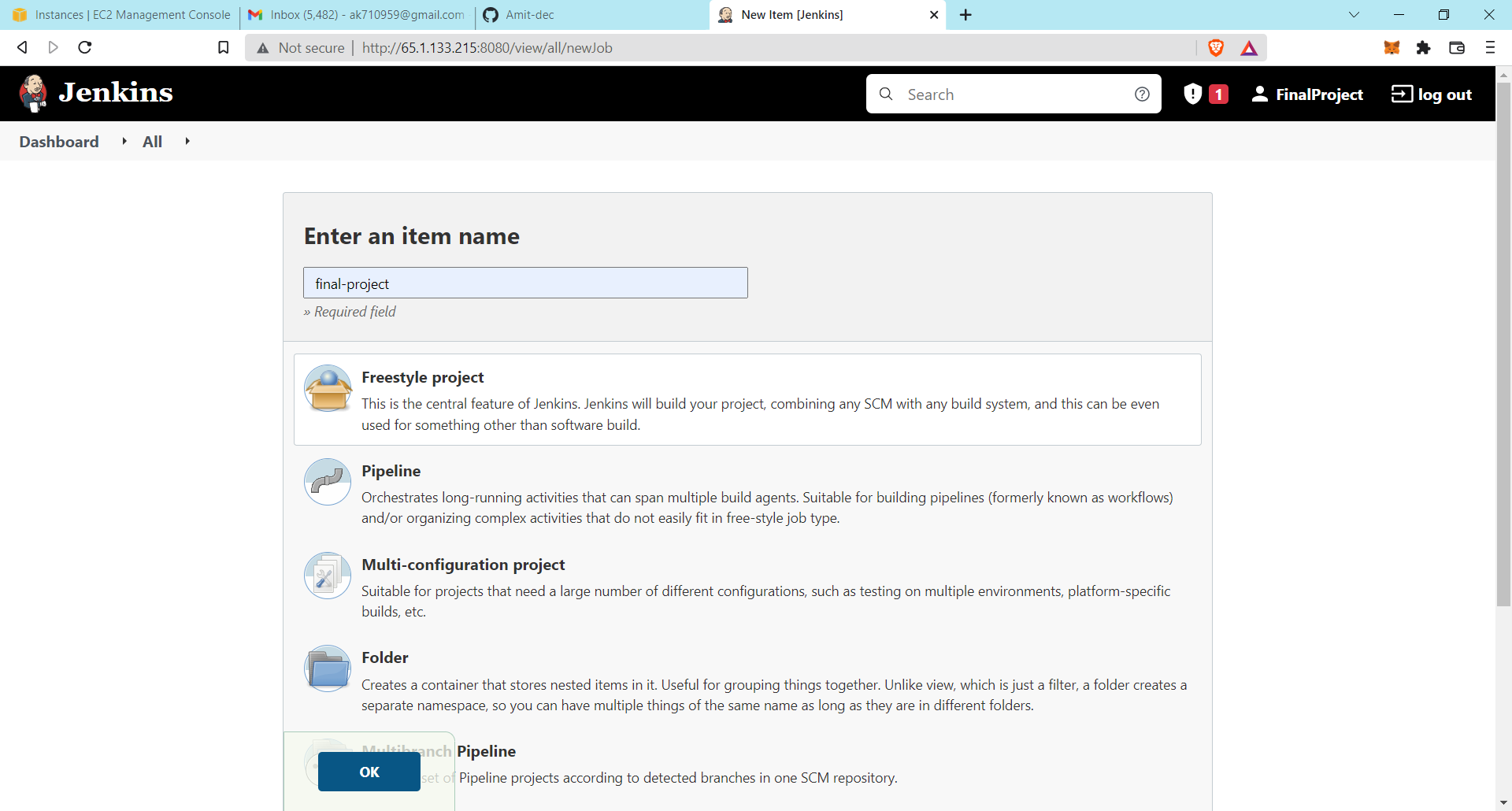


Now our setup is ready.

**Creating the new project file in the Jenkins**

Go to the **Jenkins**. Click on the **New Item.**

Then write the name of the file and select the **Freestyle project** option and click Ok.

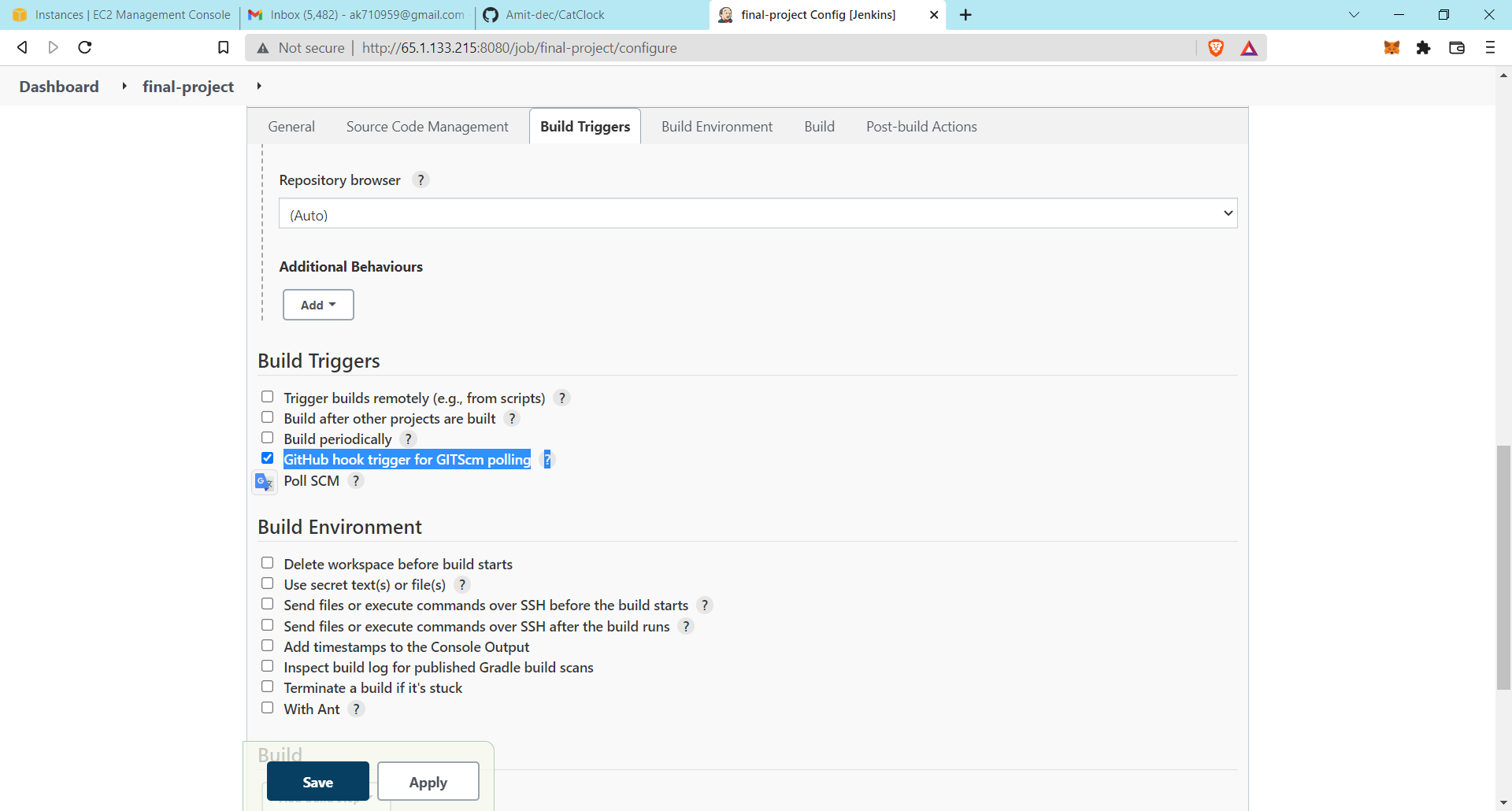


Then go to the **Source code management** and select the **Git**.

Then copy the URL of your project file from the GitHub account and then paste in the **repository URL** shown below in screenshot.

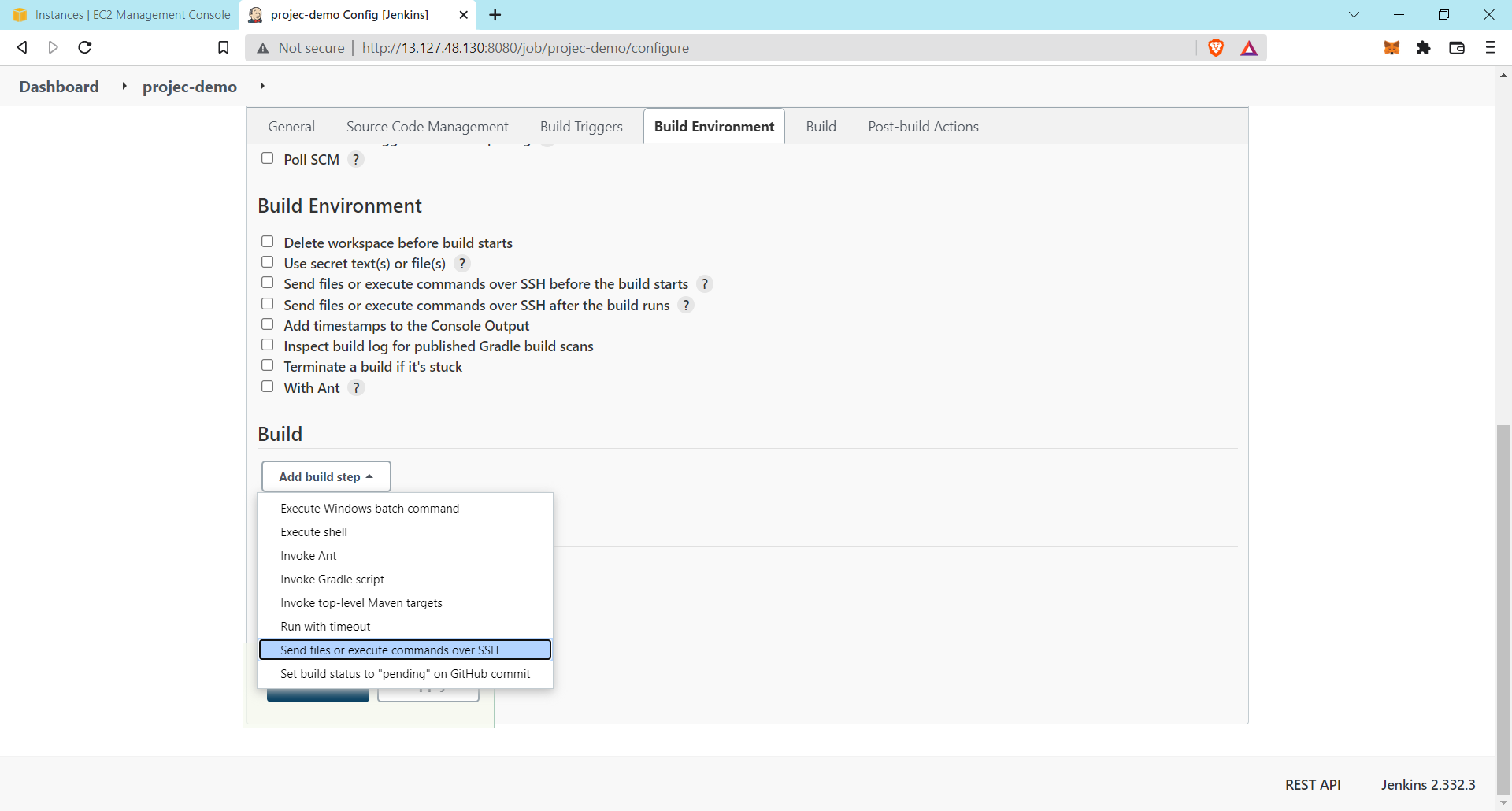


Then got to Build Triggers and select the **“GitHub hook trigger for GITScm polling”.** This option used for automatically triggering for build the project when any changes is made in the index file or project file.



Now go to the build option and click on the **“Add build step”.**

Then select the **“Send files or execute commands over SSH”.**

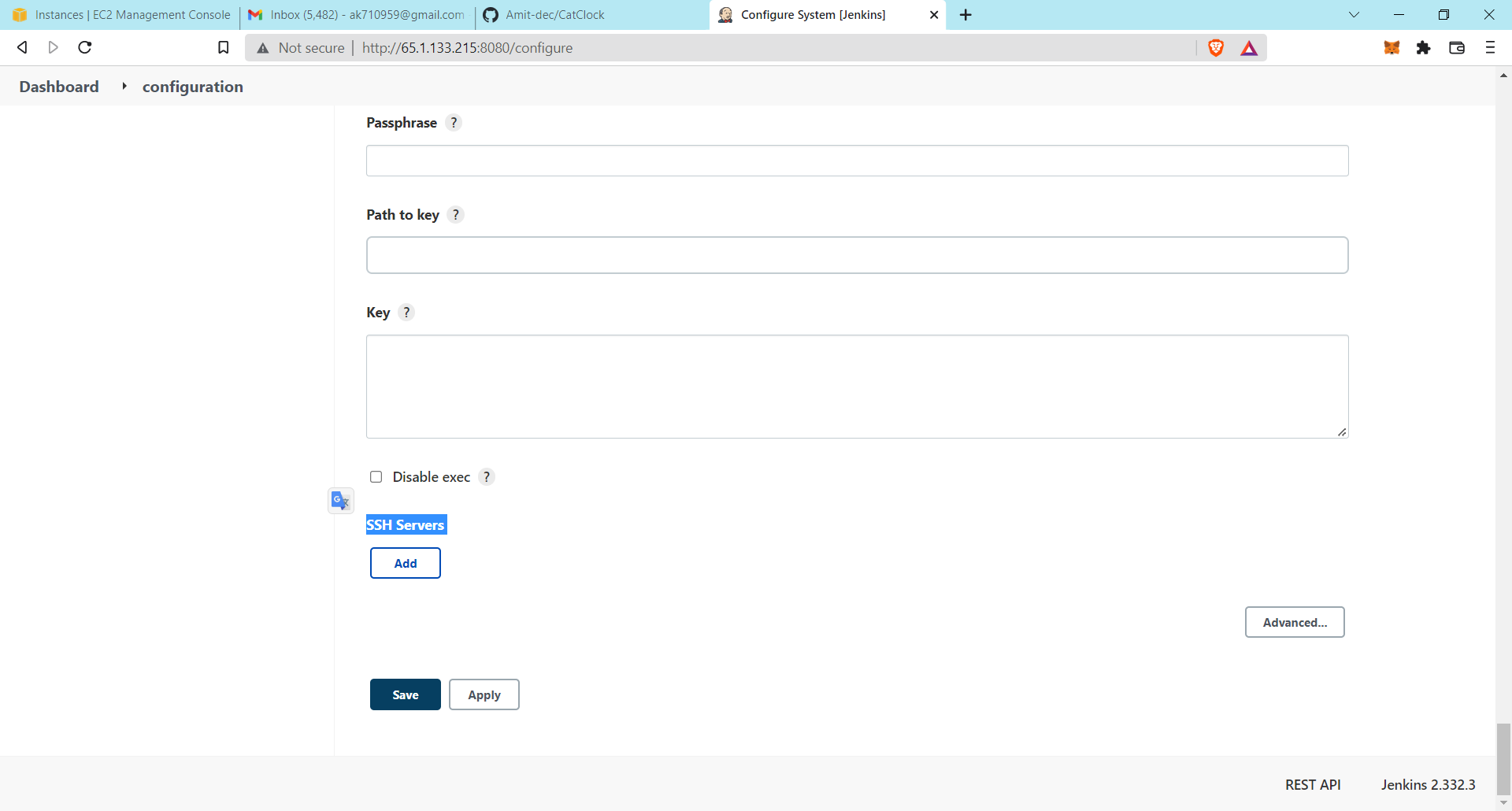


Before adding the **SSH server** in this, first we create the SSH server.

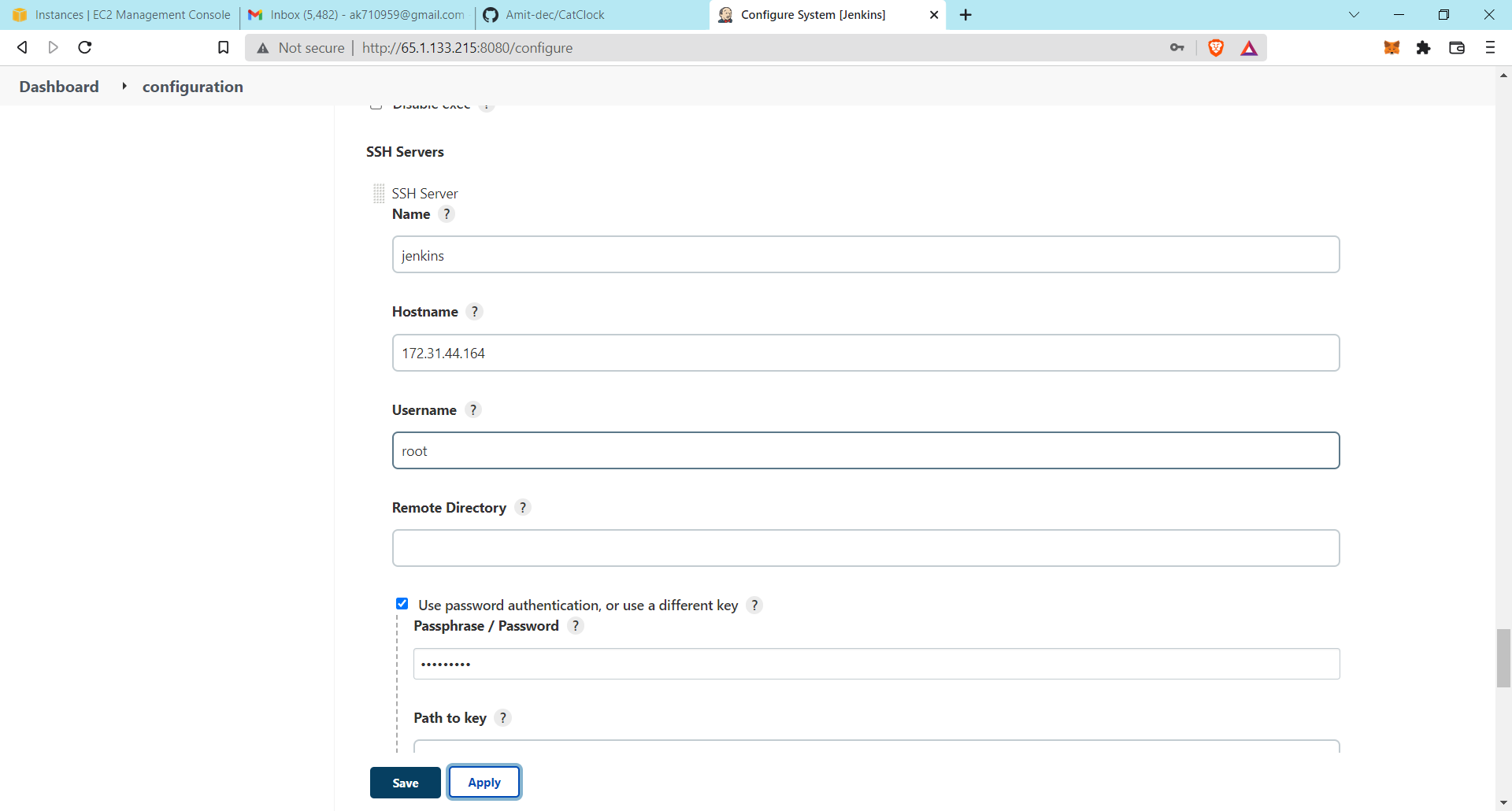
For that go back to the dashboard and click on the **“Manage Jenkins”** and then click on the **“Configure System”.**



Then go down below and find the **“SSH server”** and click on Add.

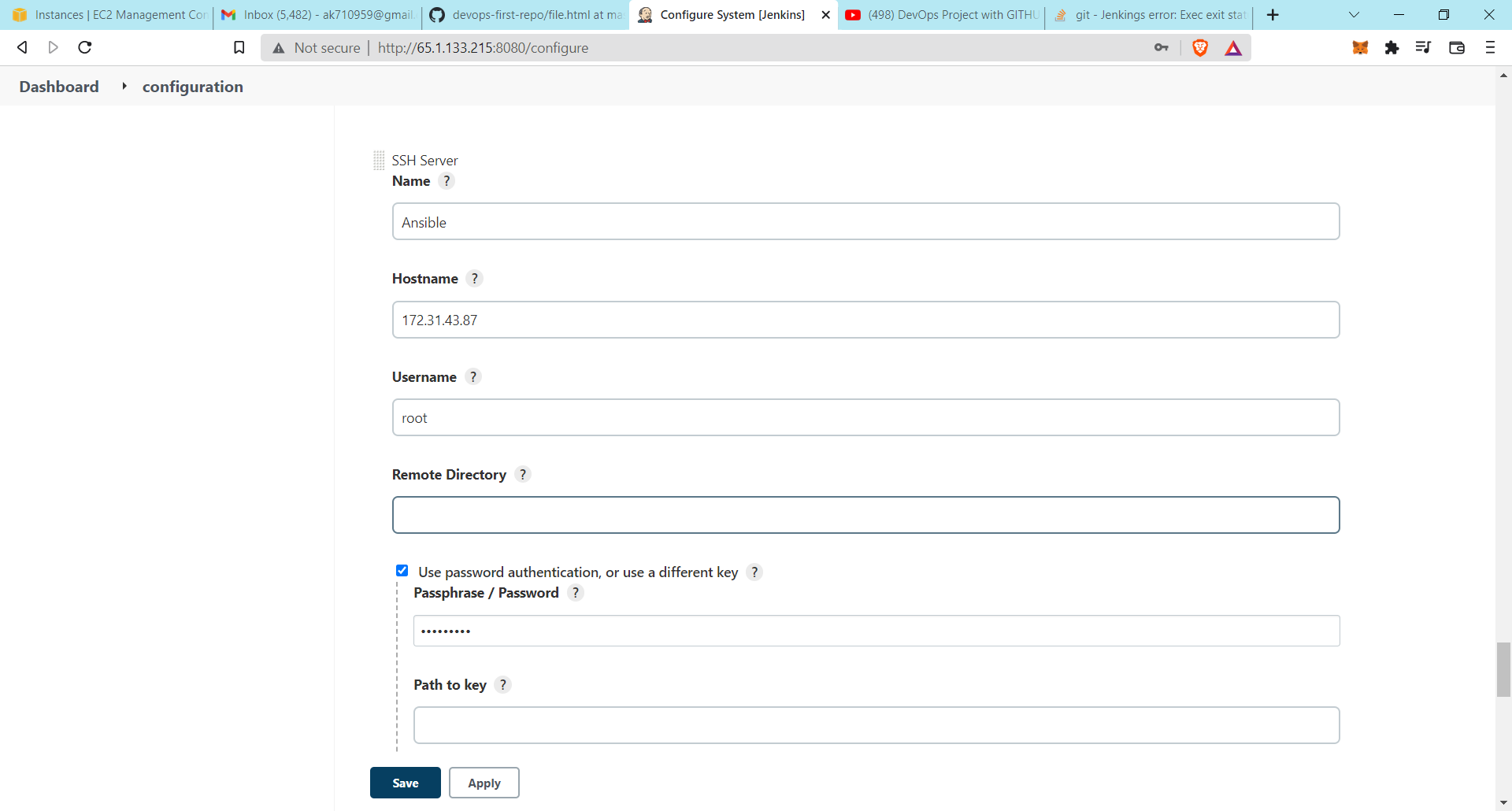


Then give the name of the ssh server hostname which is a Private Ip of the Jenkins server, username which is root, and then click on advance and the password of the Jenkins server.

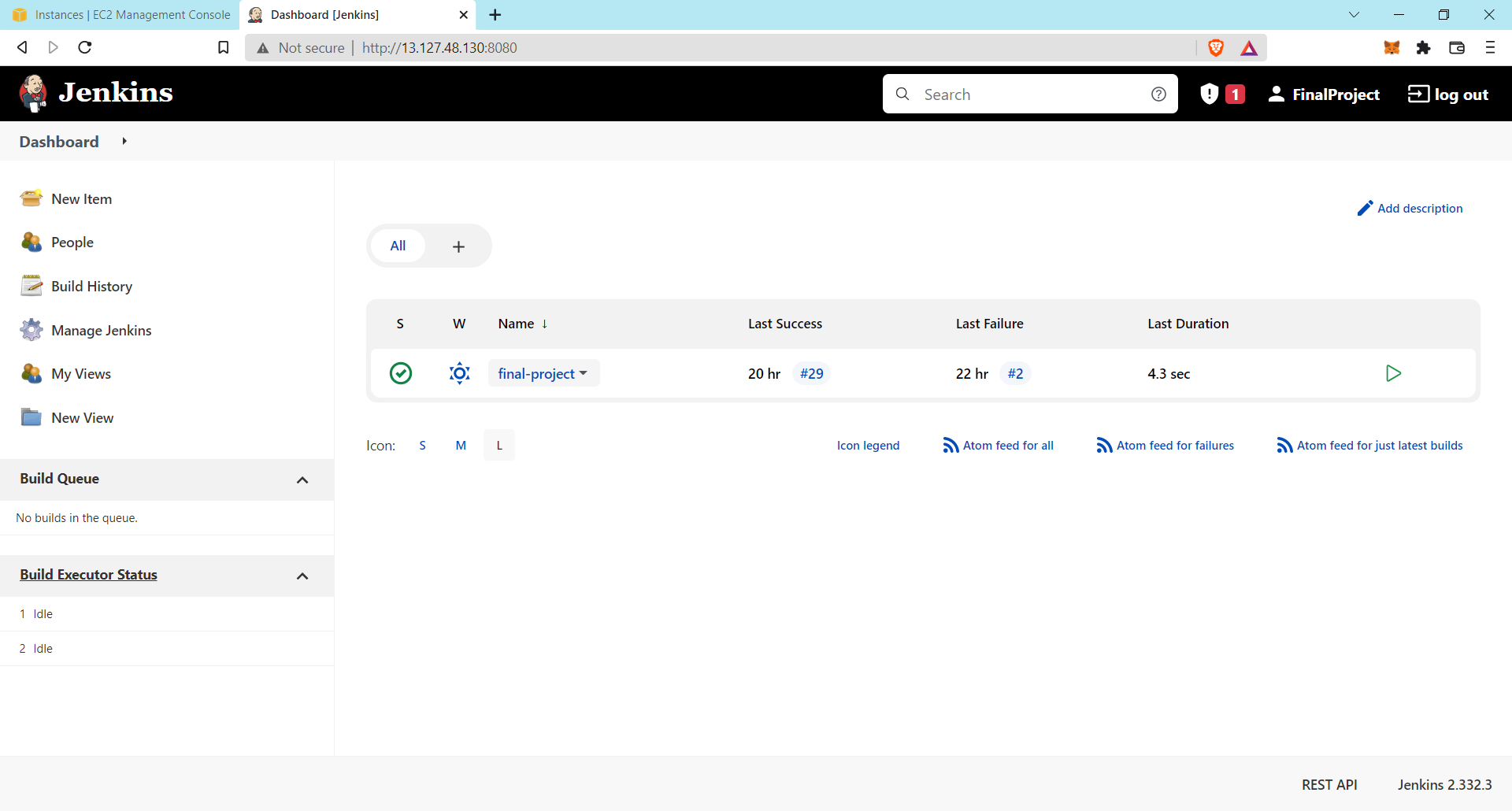


Then click on Apply and save.

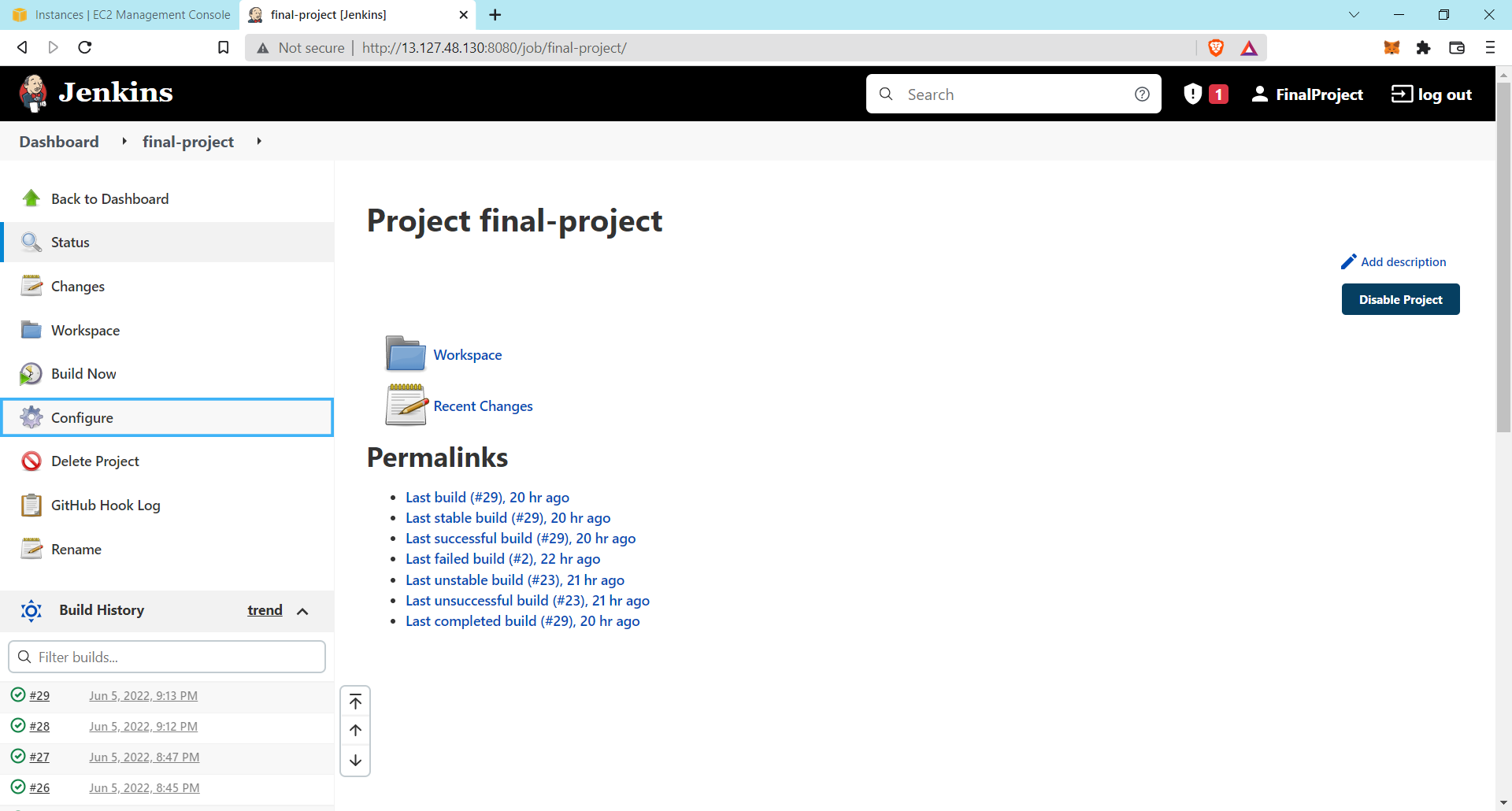
Do the same process from above for creating the **SSH server** for Ansible and apply and save.



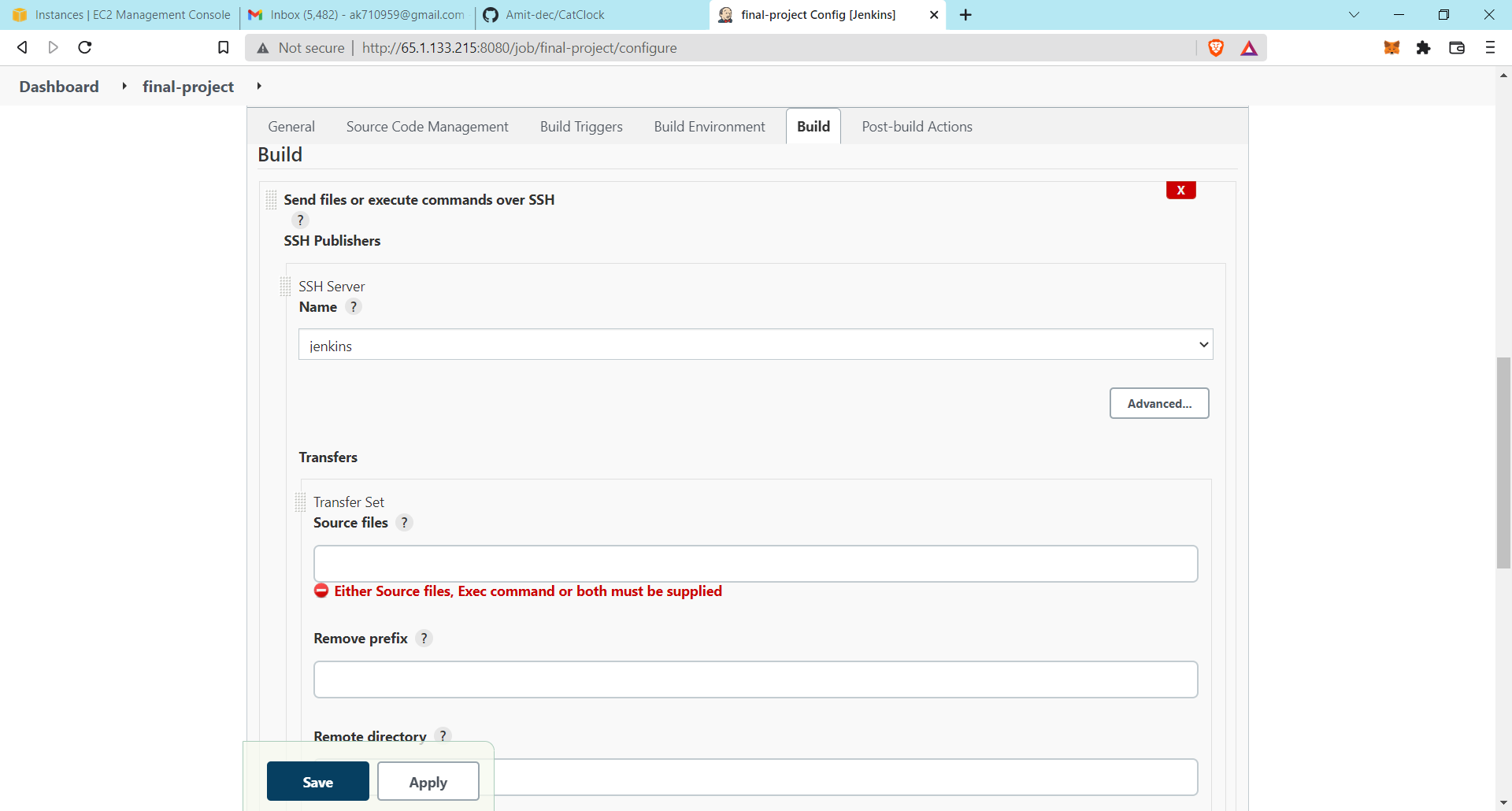
Now go back to your project file and click on it.



Click on the **“configure”.**

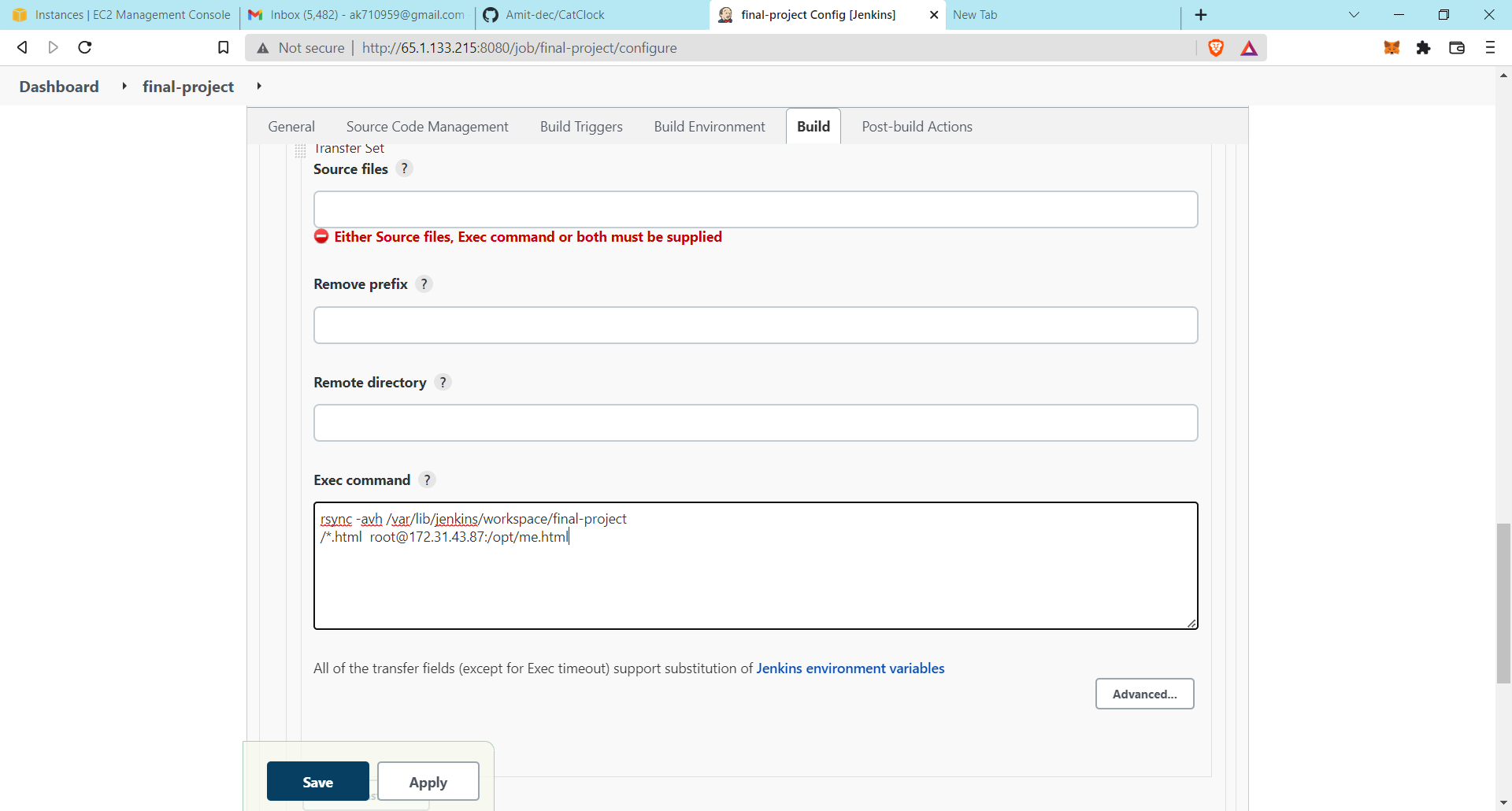


Go to build then Add it. Here you select in the SSH server’s name filed **“Jenkins”.**



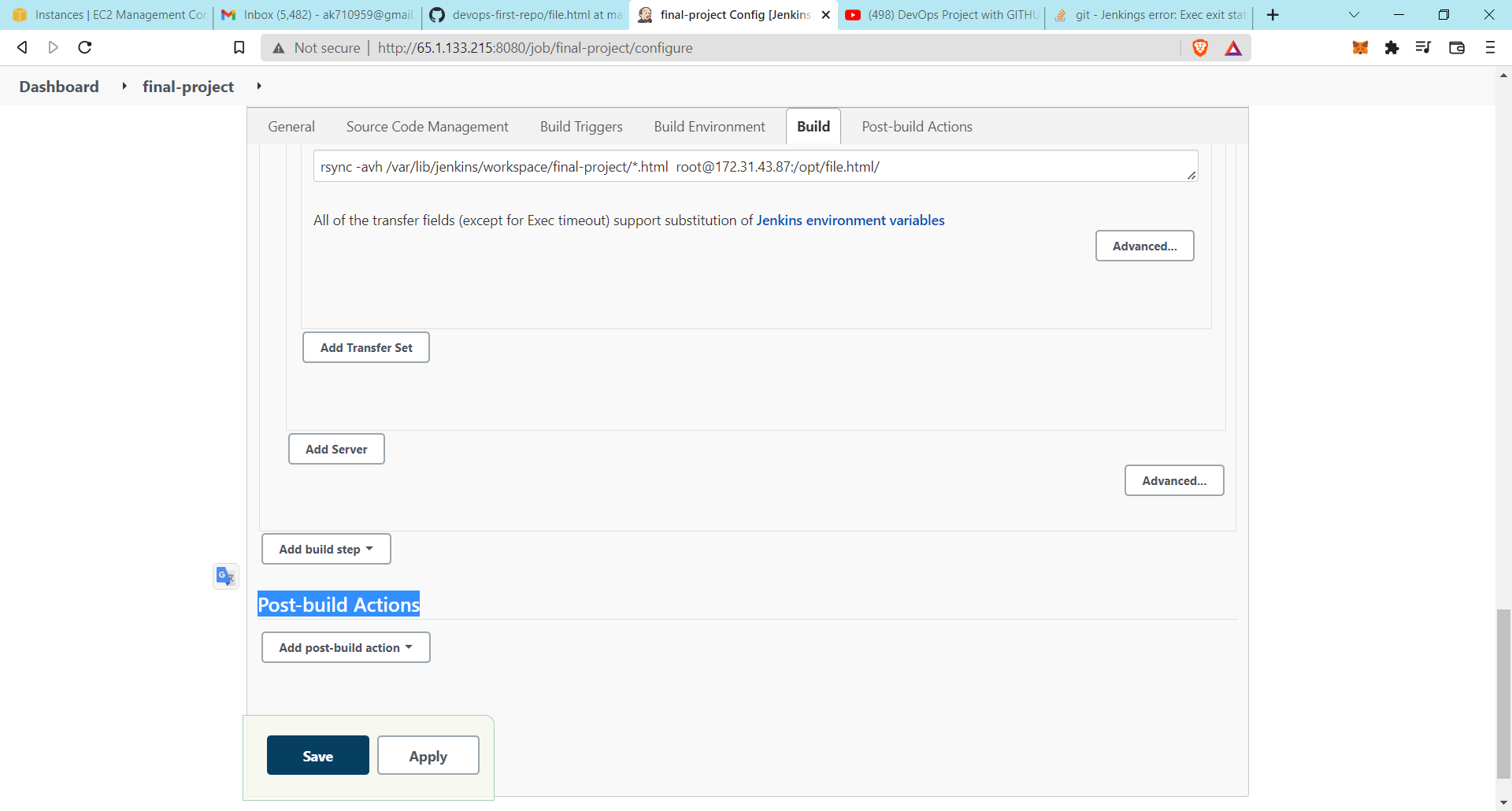
Then write the commands in the **“Exec command”** box which is shown below in the screenshot.

This command helps for creating the CI/CD pipeline between the Jenkins and ansible server. The first line of the command is for taking the repository files of all the “.html” files and in the next line from the root is the destination file where it is paste it.



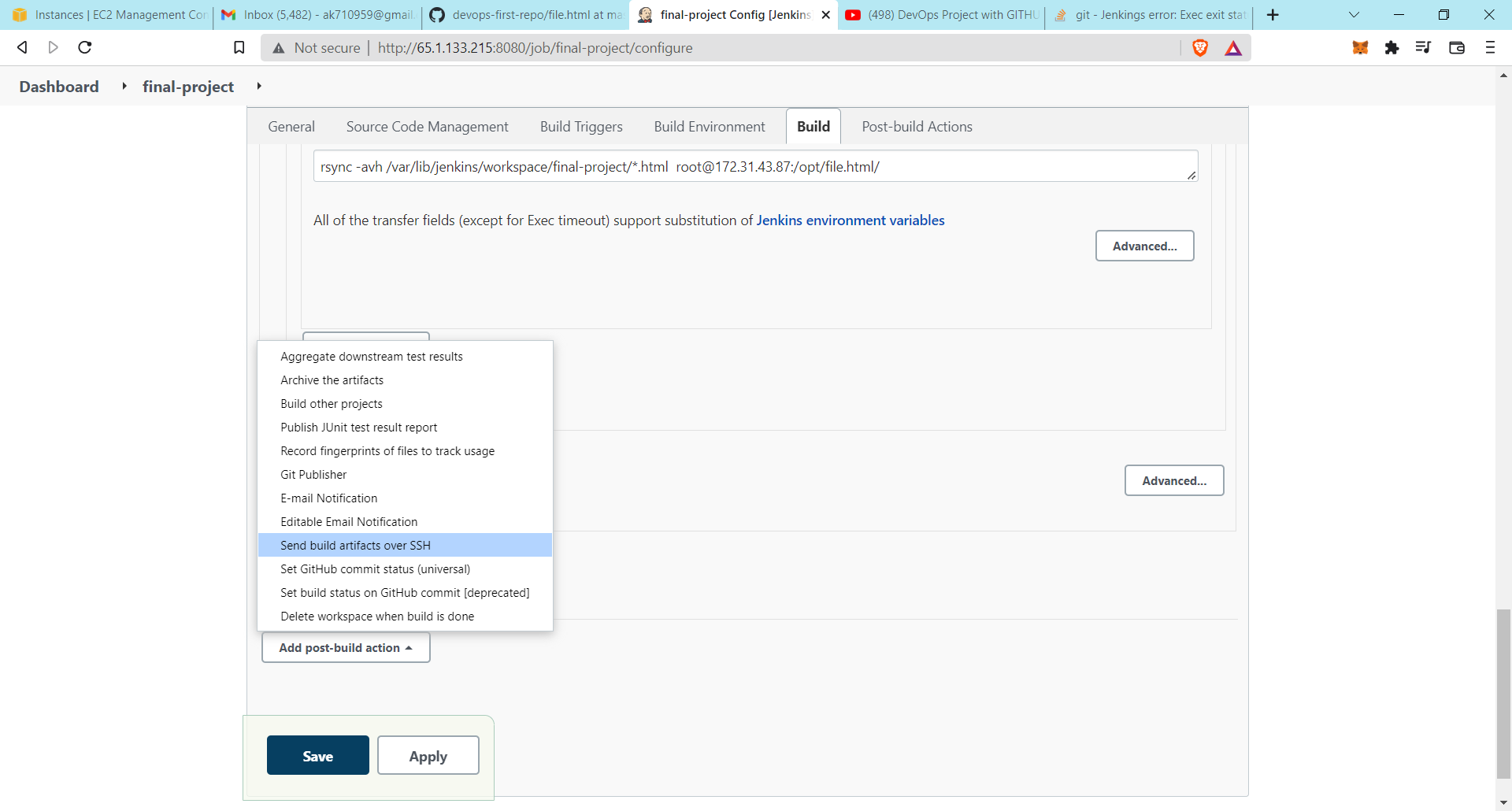
Then apply and save it.

Now click the **“Post-build Actions”** option and add the it.

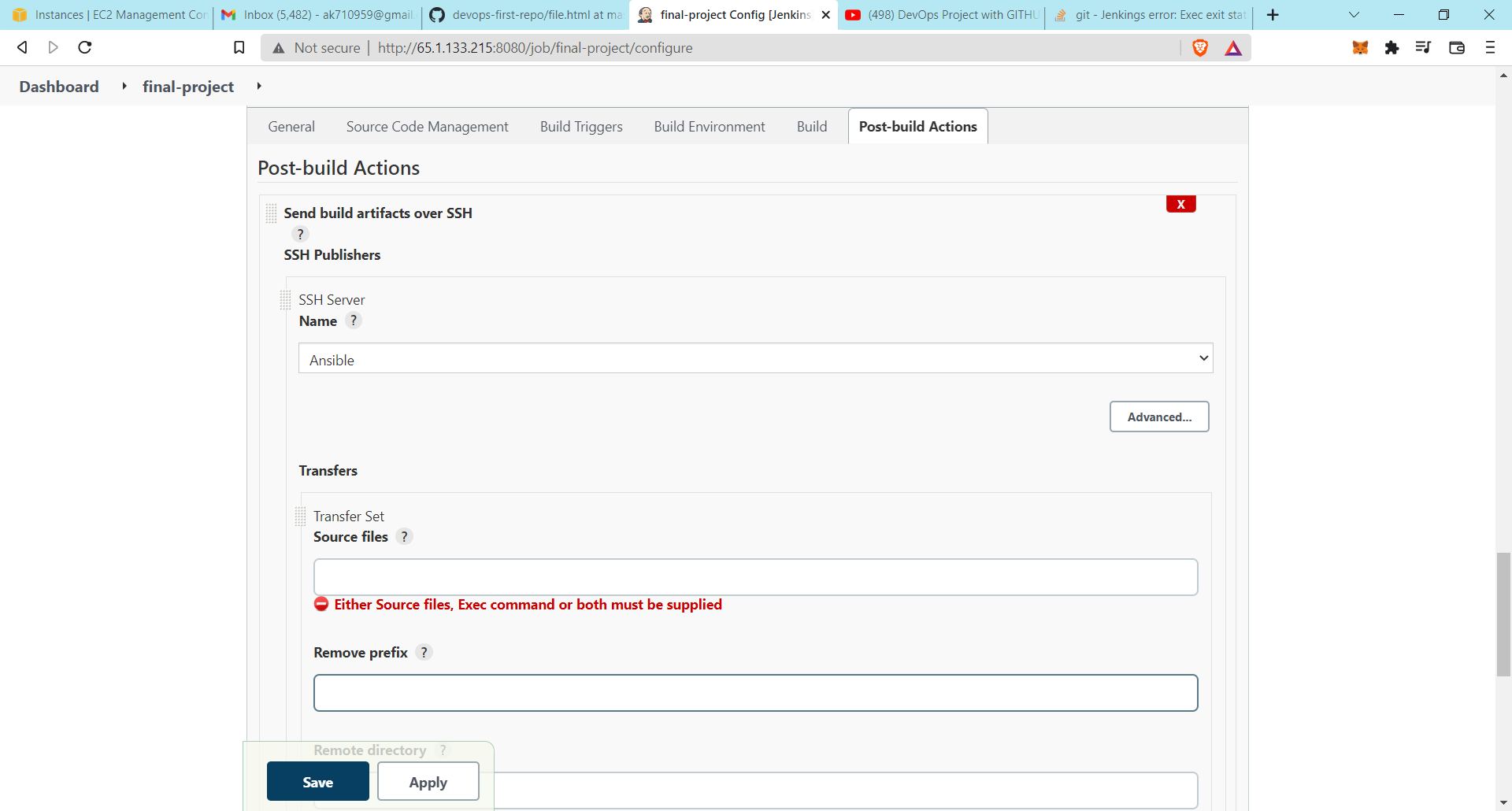


This option is used after the Jenkins build process in which the file after the building comes to the ansible server and then after post build option performs their actions.

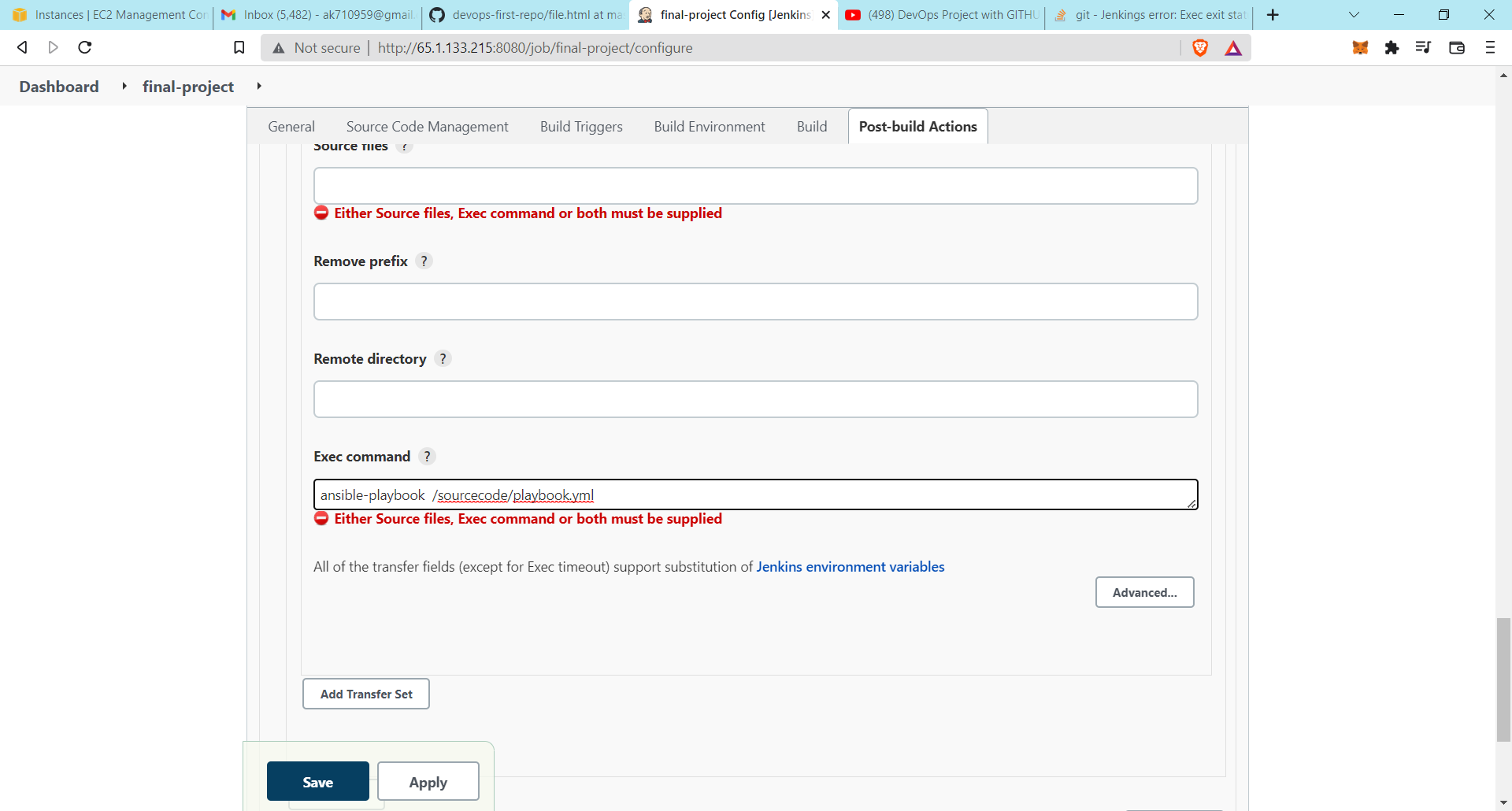
Then select the **“Send build artifacts over SSH”.**



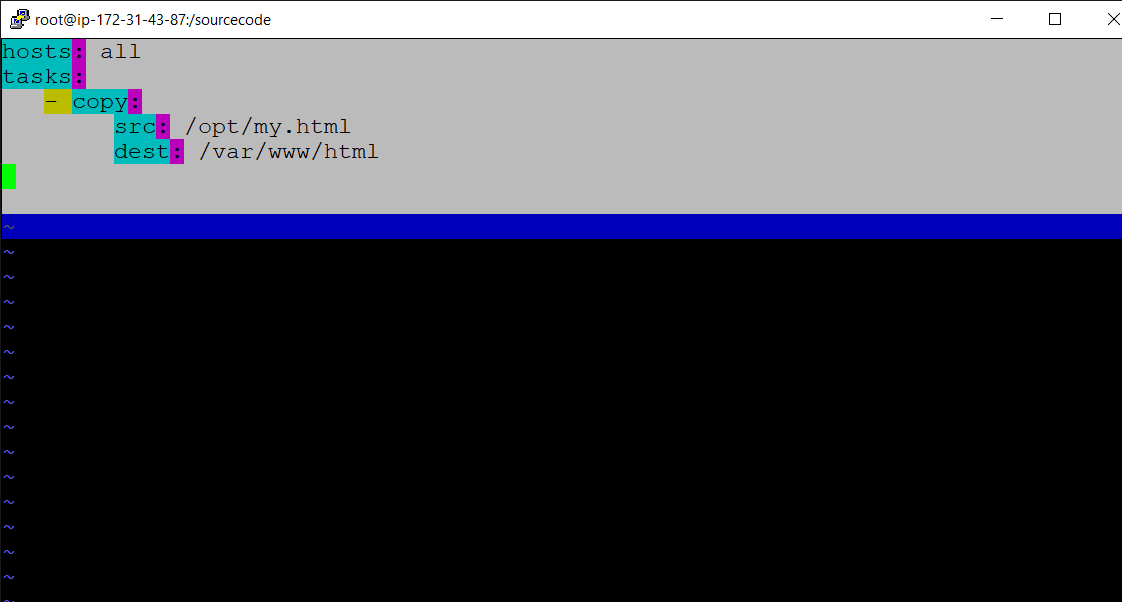
Here you select the **Ansible** SSH server.



Now type the command in the **Exec command** box which is given below in the screenshot.

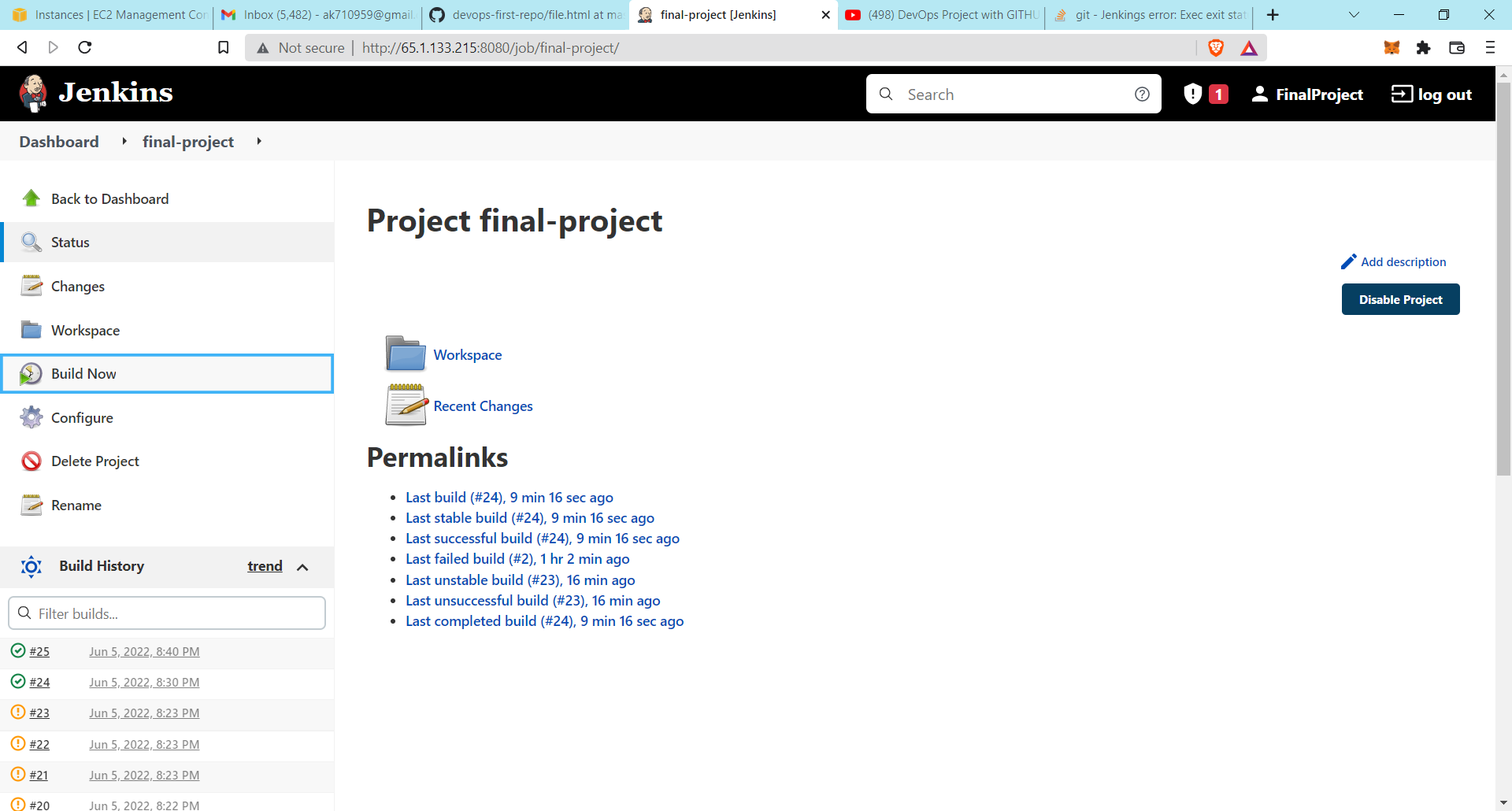


This command is run in the ansible server and inside the ansible server the **playbook.yml** file is running. And inside the **yml** file the script is running. This script is sending the **.html** file to the destination server which is the web-server.

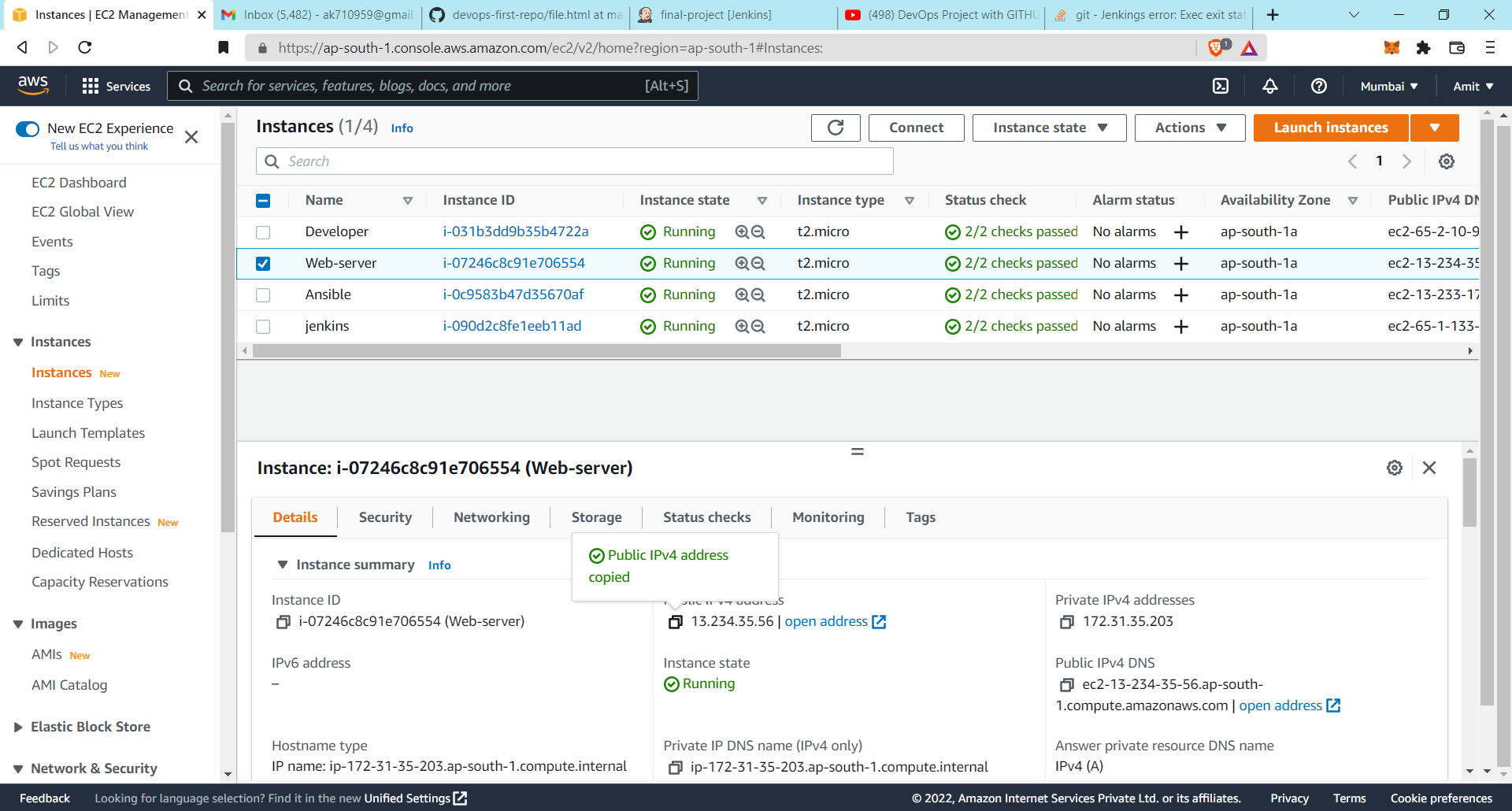


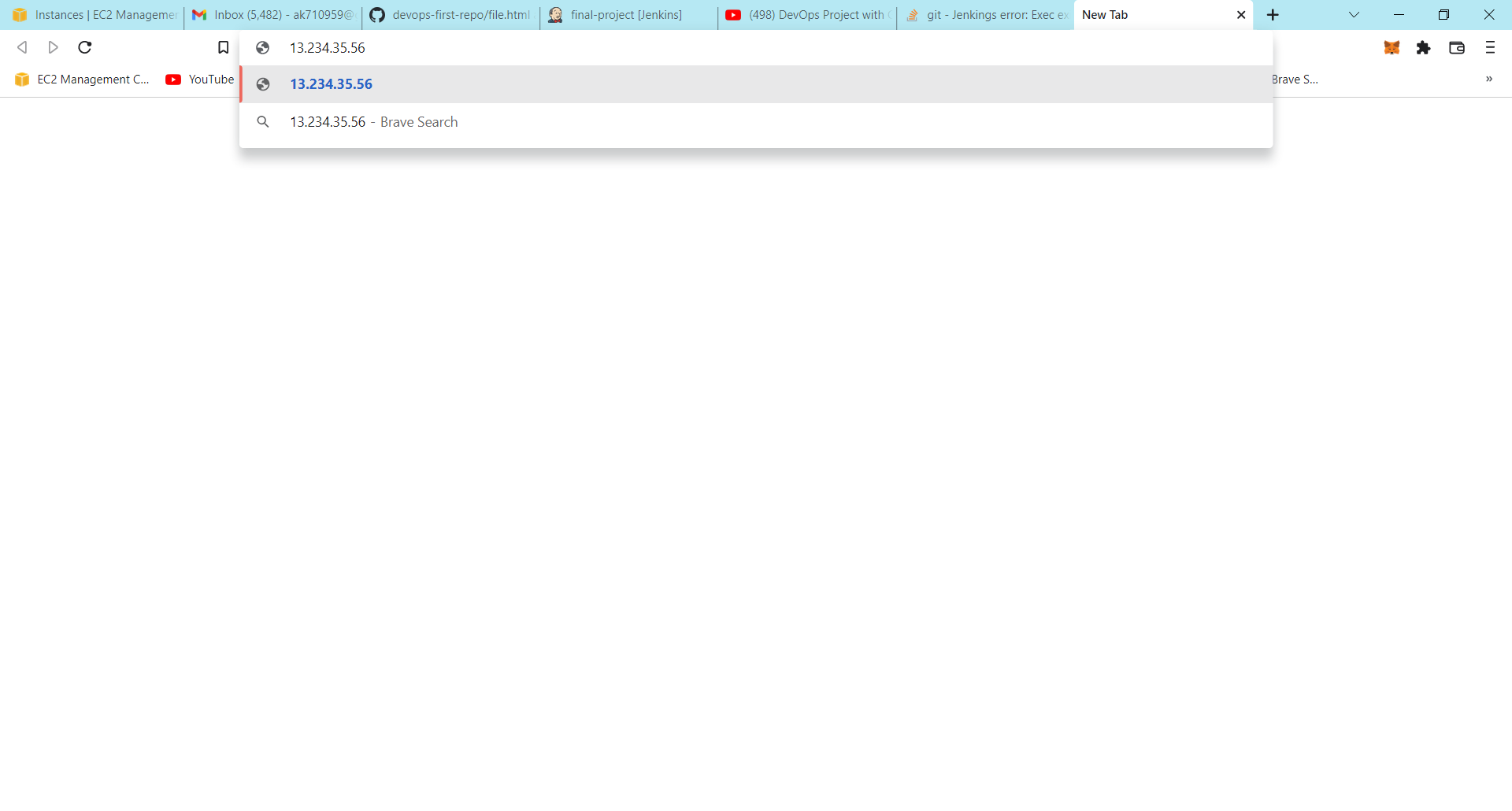
Now apply it and save it.

After all the settings, now click on the “**Build Now**”.

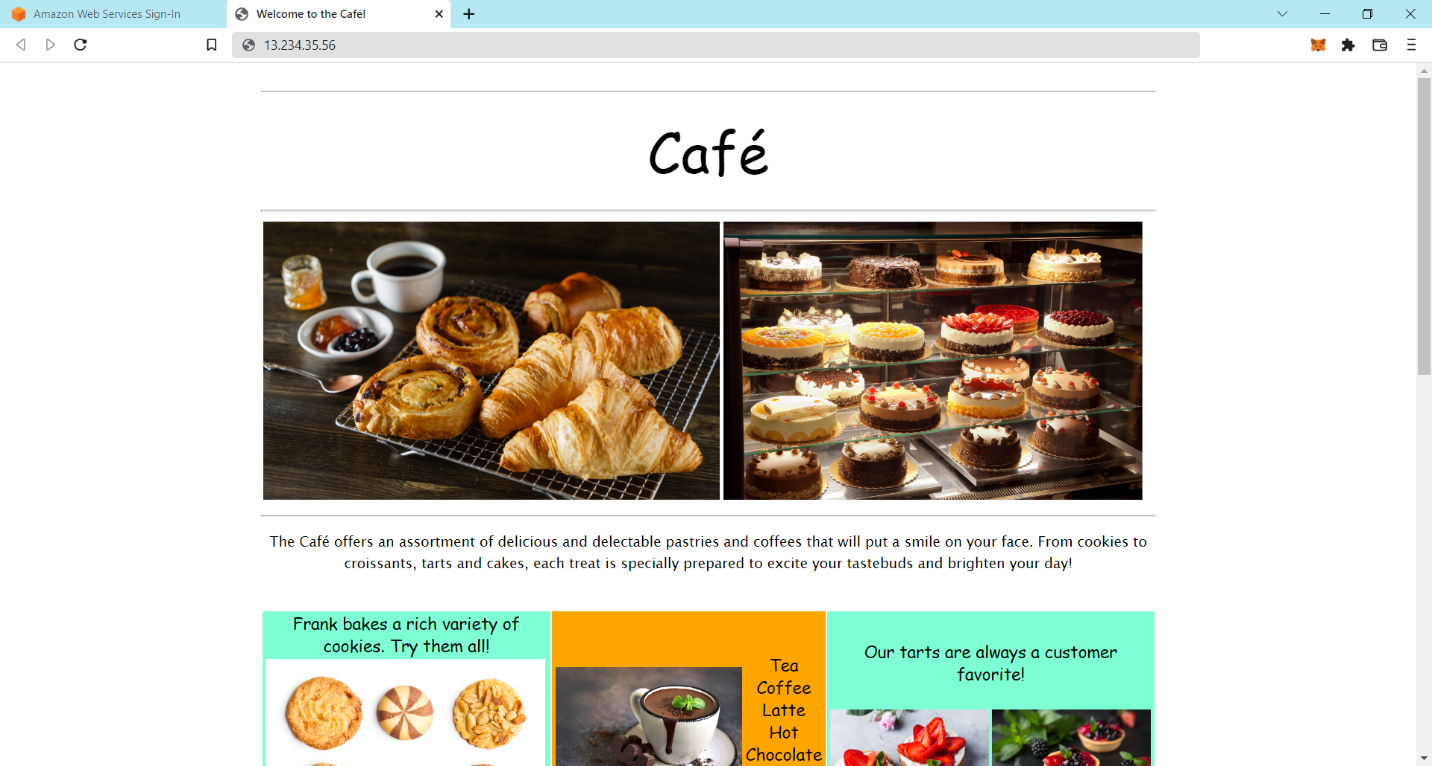


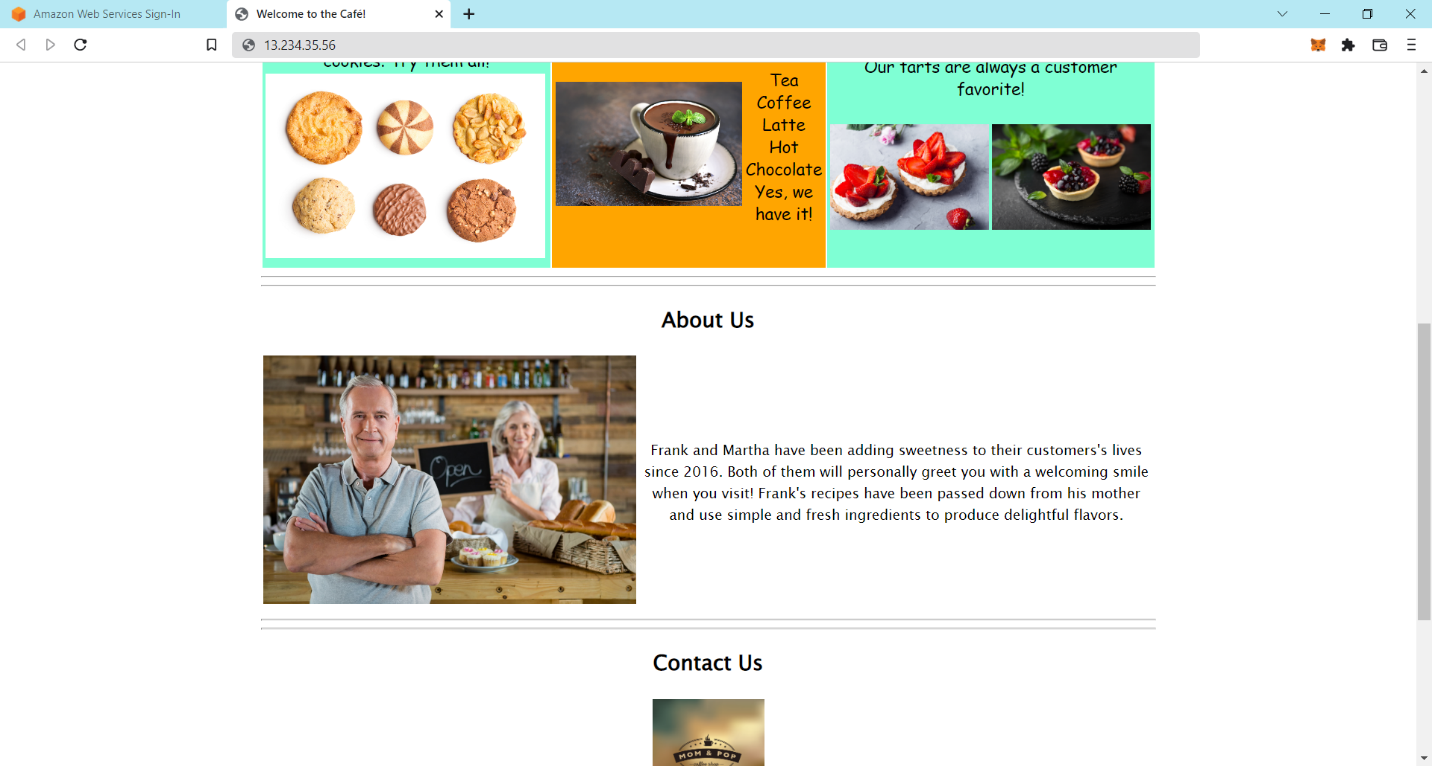
Now your integration of file is running. After successfully building the project, copy the public ip of the webserver and paste it on the browser.





Now your Webserver is ready and your website hosting is successfully running.





If the developer changes the code or update any code and push to its GitHub repository then the Jenkins is automatically pulling that code and perform their building testing process.

**CONCLUSION**

In conclusion, this project includes the AWS EC2 instances and the instances running the DevOps tools services. The DevOps tools we used in this project are Jenkins, Ansible, Git and GitHub. The hosted website is running using these DevOps tools, which help these websites run automatically while any changes occur and help them run smoothly.