**Project : Capstone I**

You have been hired as a Sr. DevOps Engineer in Abode Software. They want to

implement DevOps Lifecycle in their company. You have been asked to

implement this lifecycle as fast as possible. Abode Software is a product-based

company and their product is available on this GitHub link.

https://github.com/hshar/website.git

Following are the specifications of the lifecycle:

1. Install the necessary software on the machines using a configuration

management tool

2. Git workflow has to be implemented

3. CodeBuild should automatically be triggered once a commit is made to

master branch or develop branch.

a. If a commit is made to master branch, test and push to prod

b. If a commit is made to develop branch, just test the product, do not

push to prod

4. The code should be containerized with the help of a Dockerfile. The

Dockerfile should be built every time there is a push to GitHub. Use the

following pre-built container for your application: hshar/webapp

The code should reside in '/var/www/html'

5. The above tasks should be defined in a Jenkins Pipeline with the following

jobs:

a. Job1 : build

b. Job2 : test

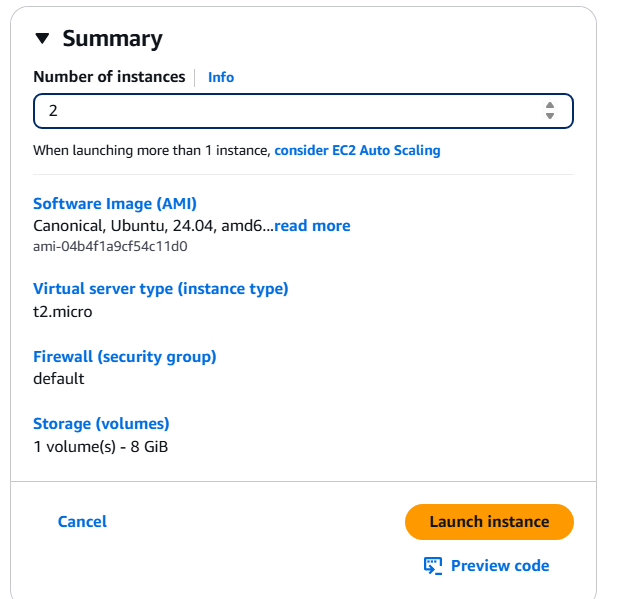
c. Job3 : prod

Solution:

creating master VM

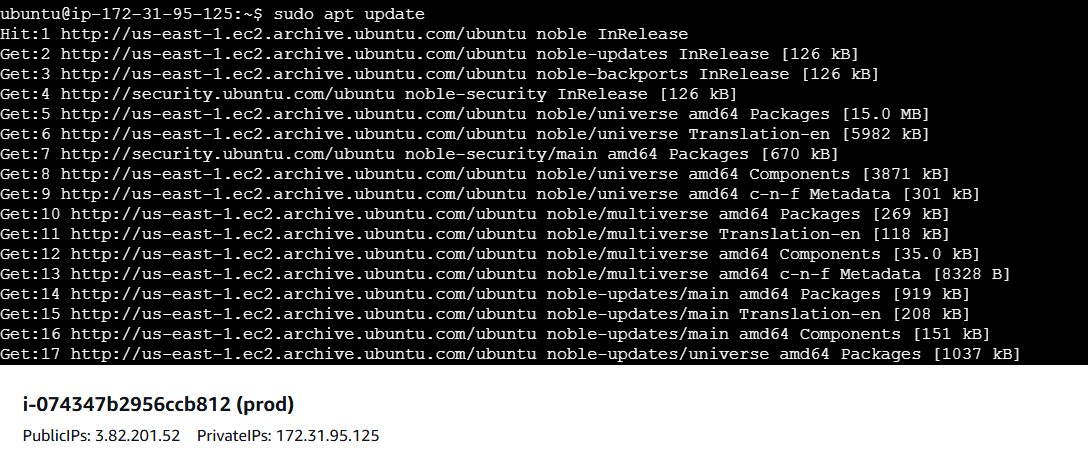
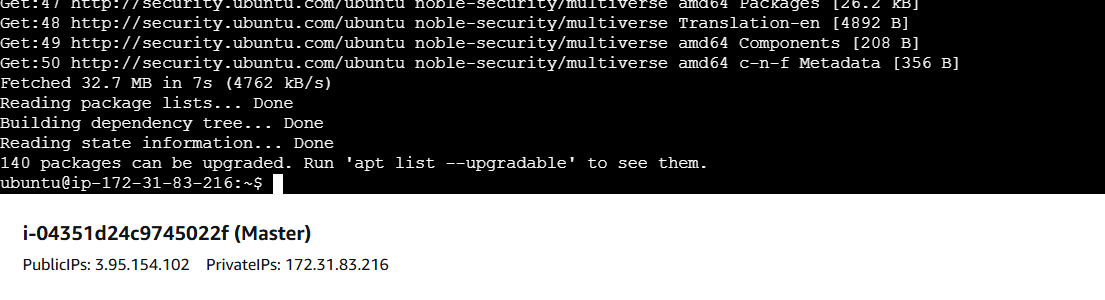


Created 2 instances for test and prod



Updating the all machine

Sudo apt update

**Installing Ansible on master instance**

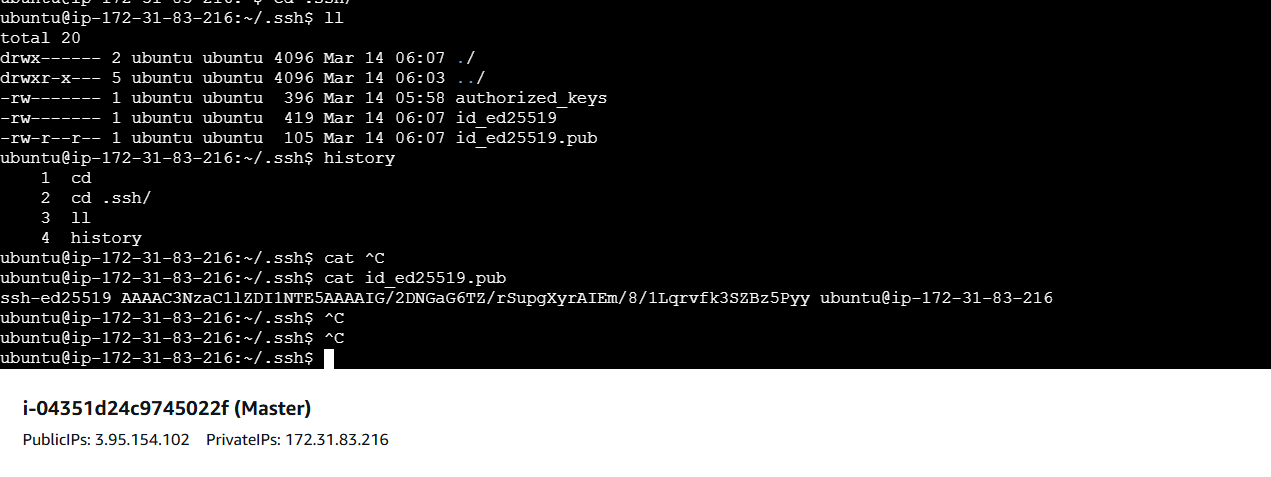
sudo apt install software-properties-common

sudo add-apt-repository --yes --update ppa:ansible/ansible

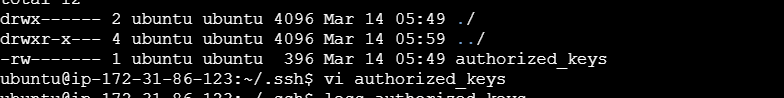
sudo apt install ansible

**Creating key to establish the cluster**

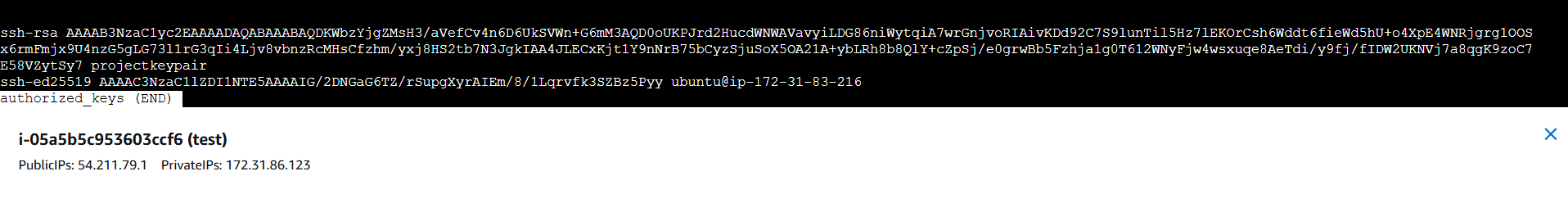
Ssh-keygen



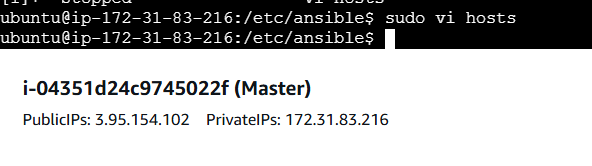
Copy the public key and add into authorized keys of test and prod instance

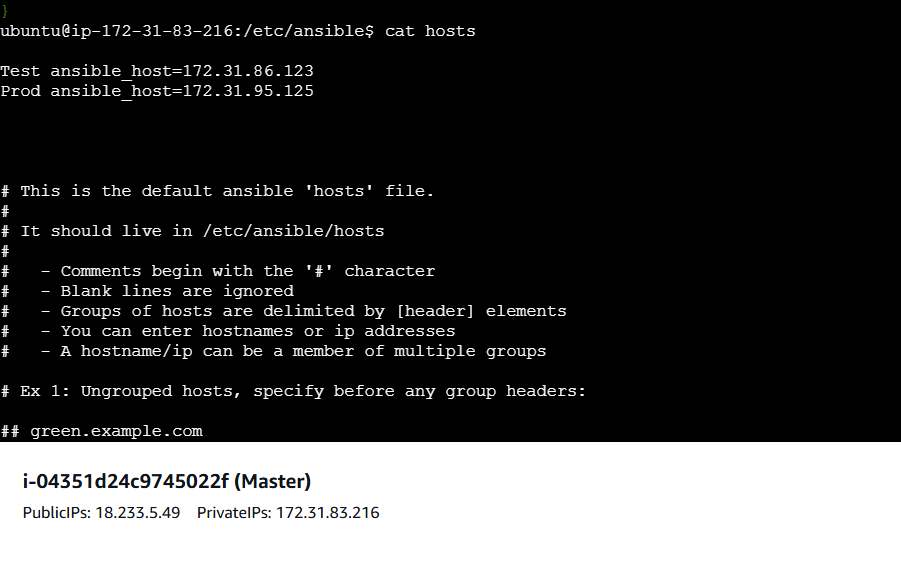




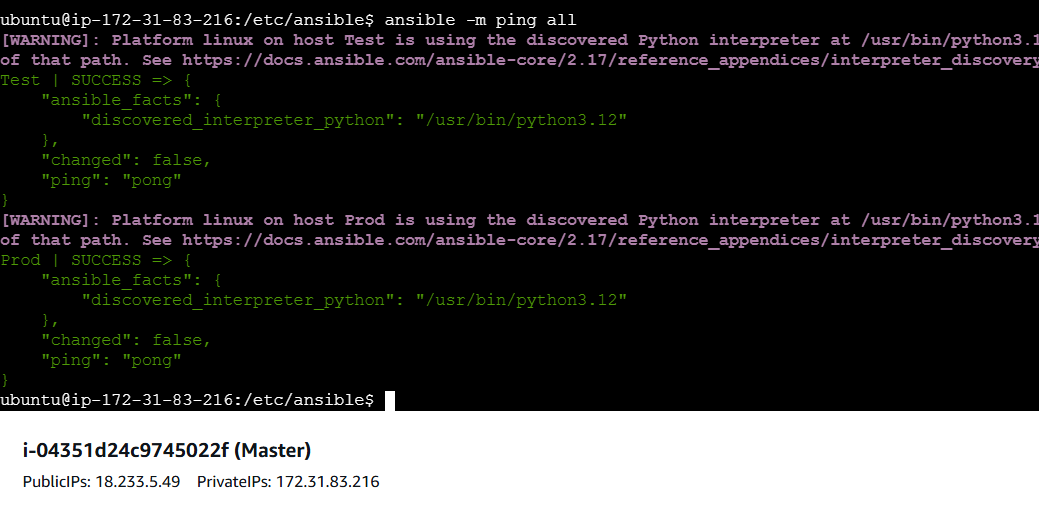


**Enter private IP of prod and test server in hosts under /etc/ansible**

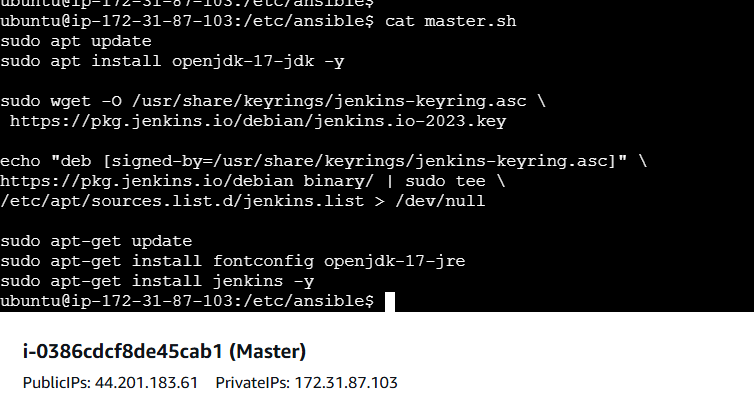


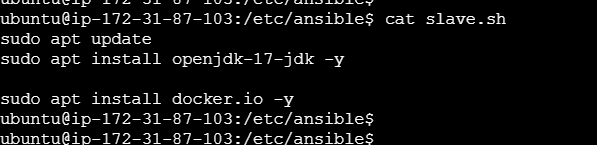


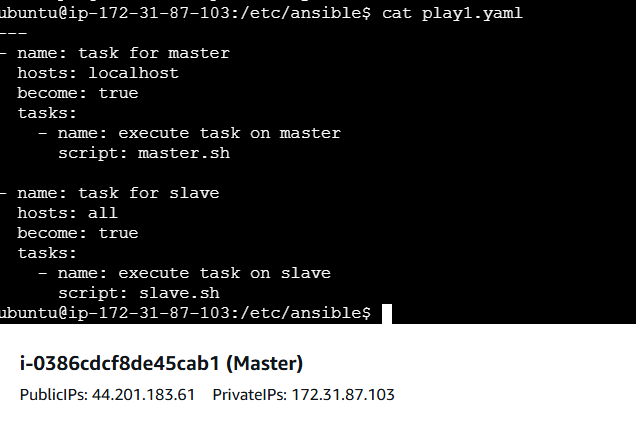
Now able to connect to test and prod



Create master.sh and slave.sh to install the jenkins , java on master and java ,docker on slave instances(test and prod instance)





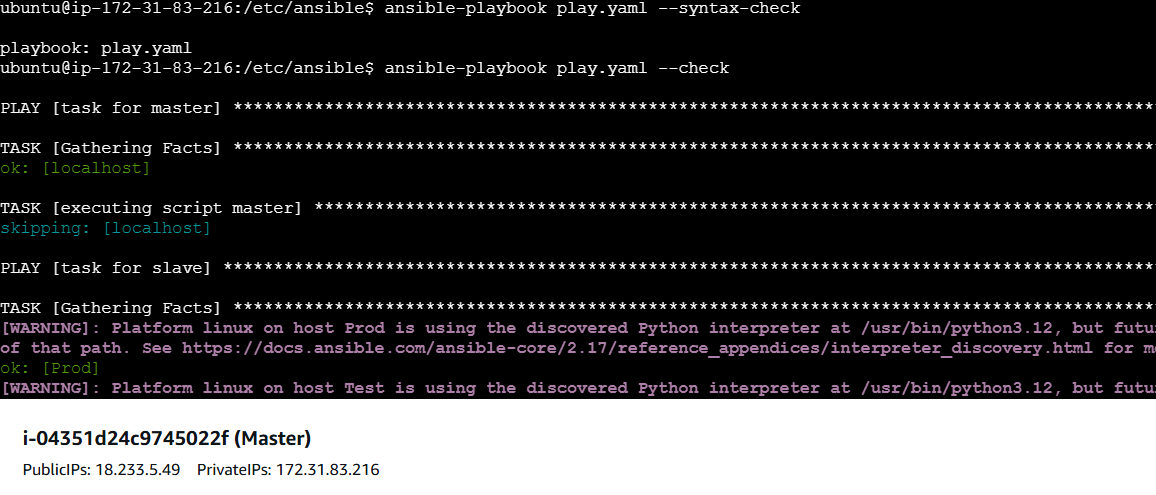


**Run below commands to verify syntax and run playbook**

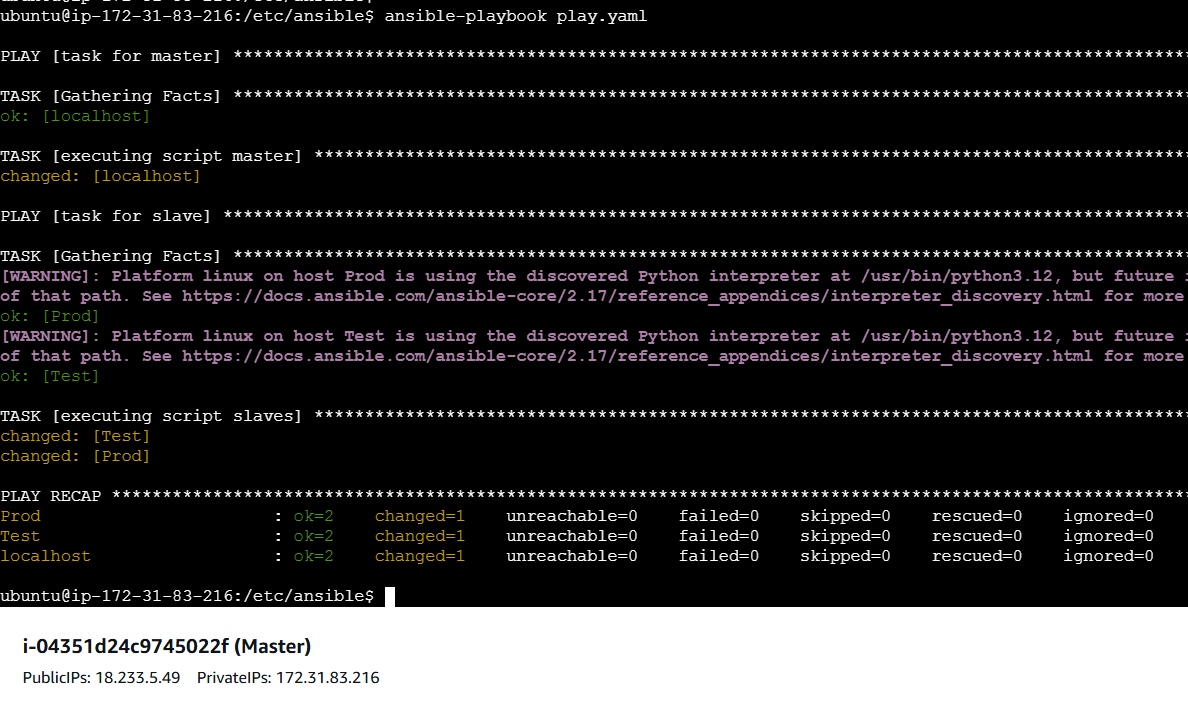
ansible-playbook play1.yaml --syntax-check

ansible-playbook play1.yaml --check

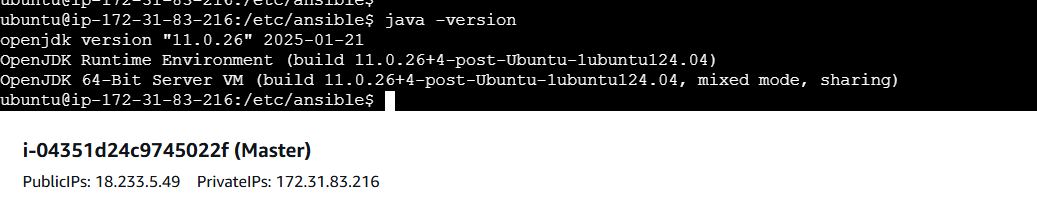
ansible-playbook play1.yaml

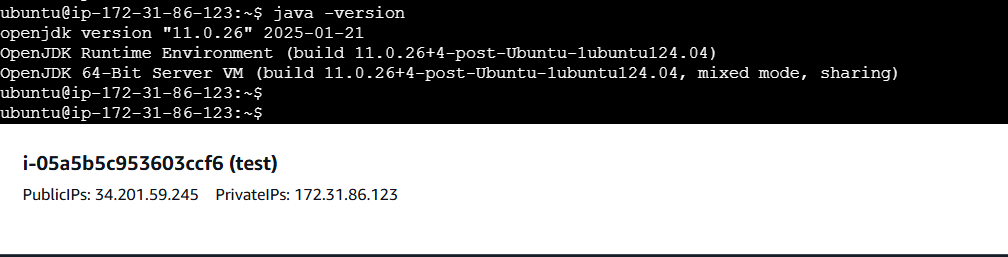


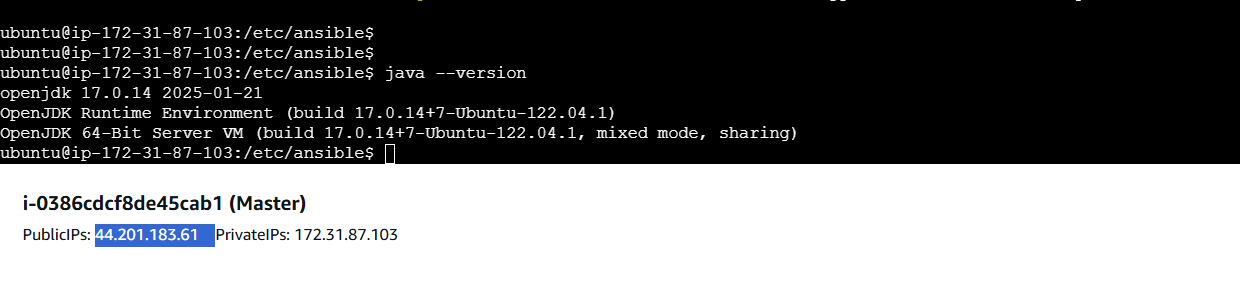
Run the playbook

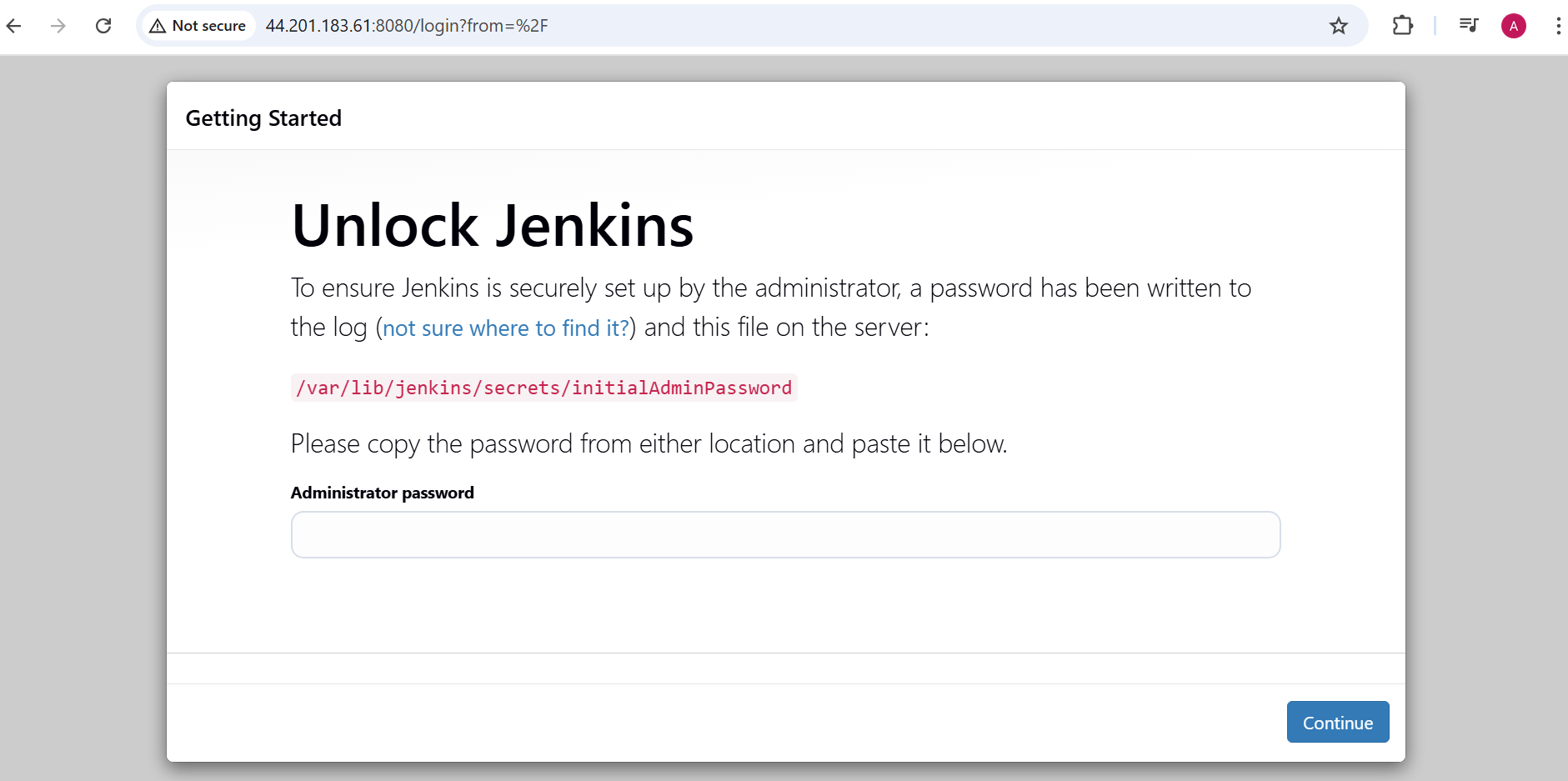


Jave installed successfully on master and slave machine

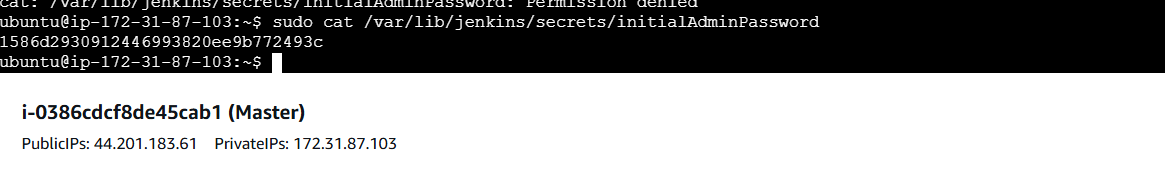


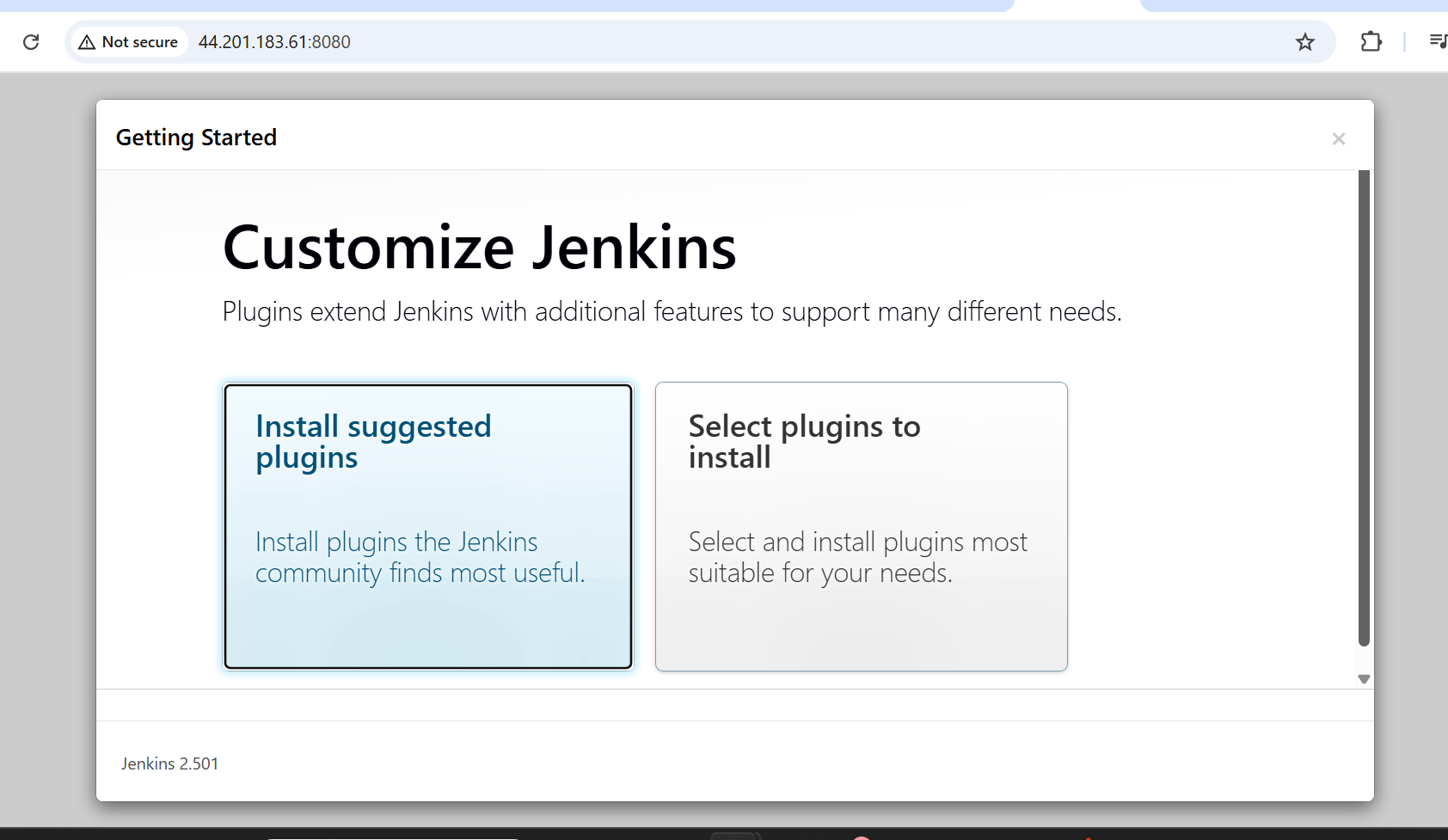


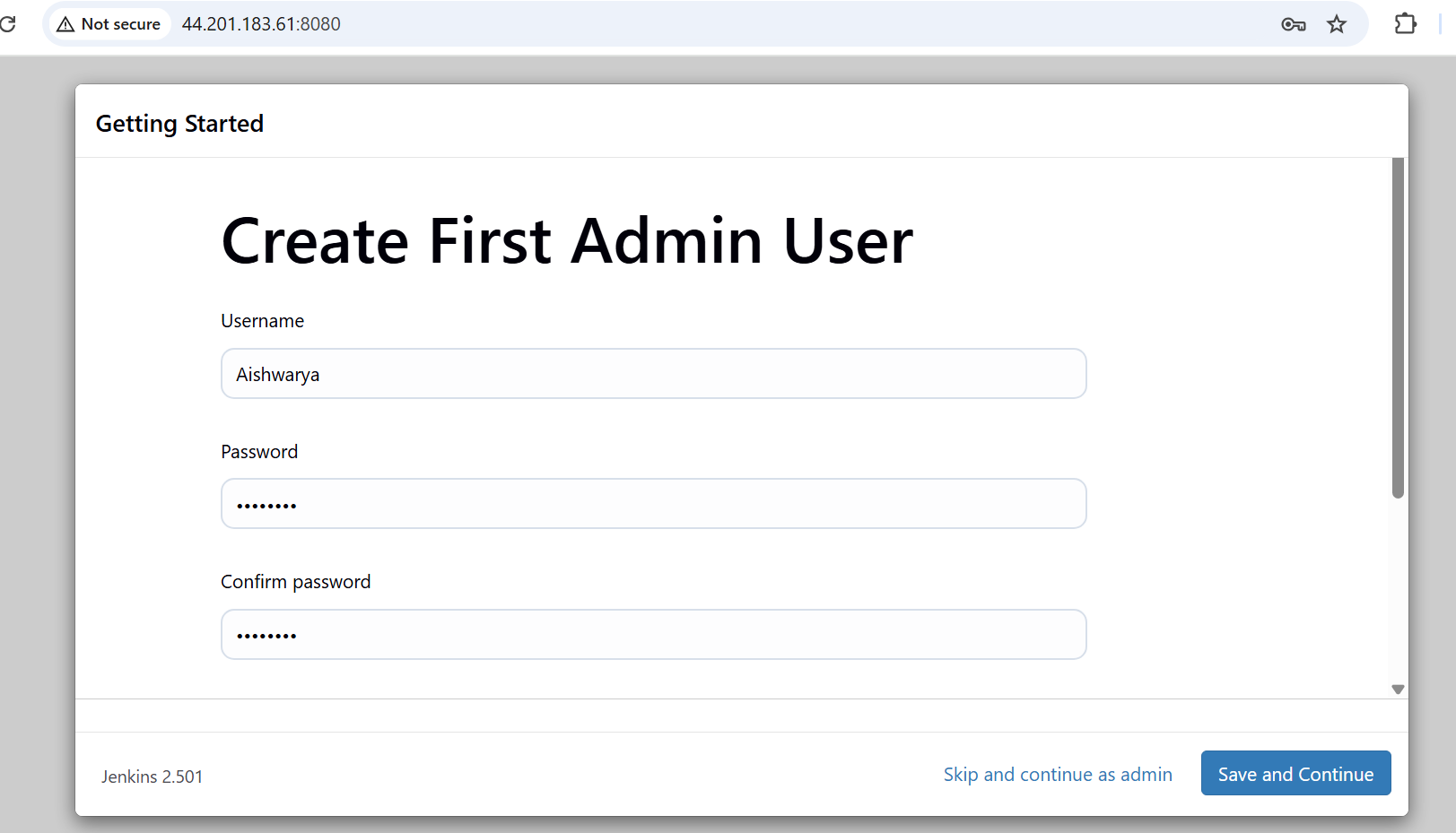
Take public Ip of the master and port 8080 to open jenkins

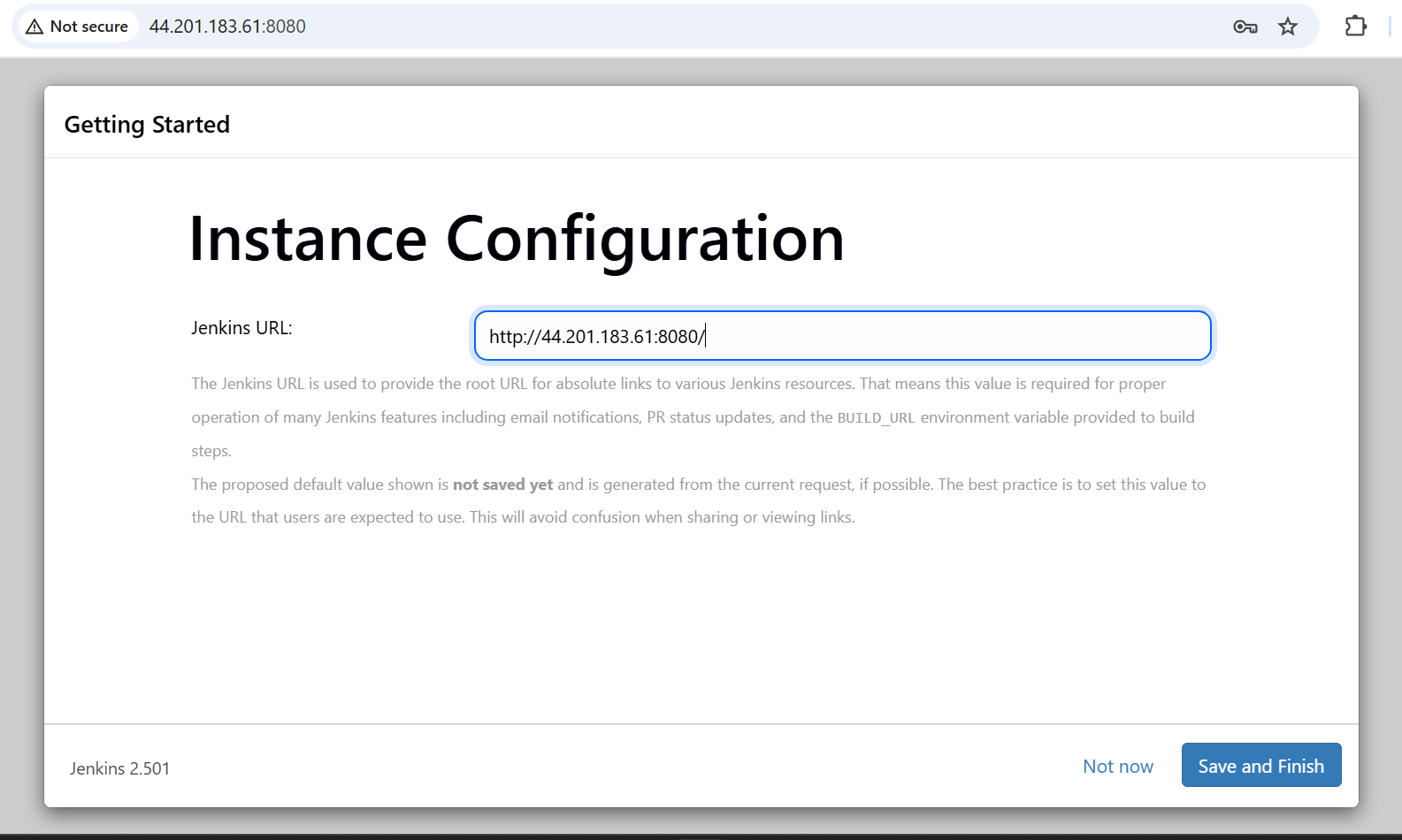


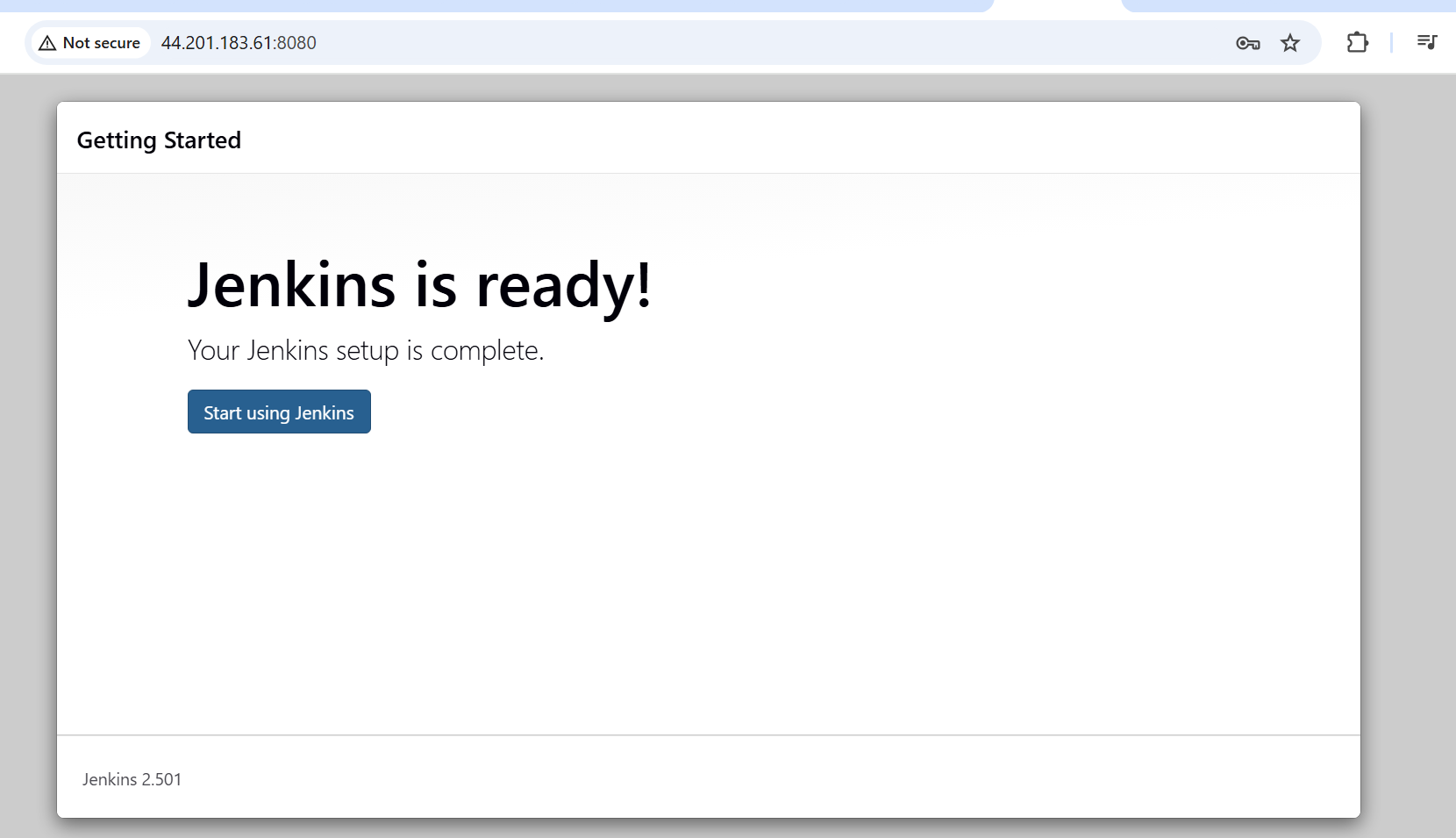
Take the initial password



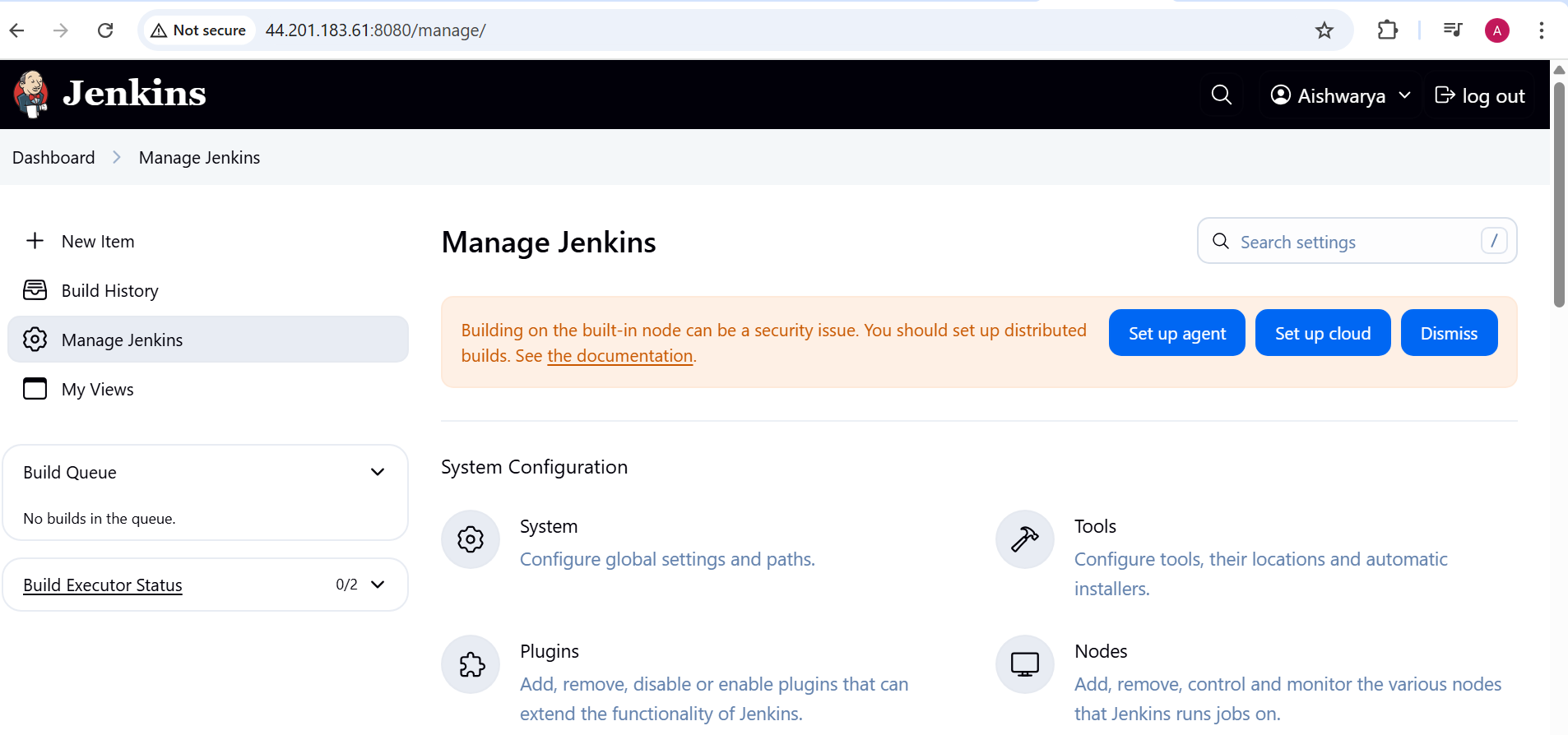




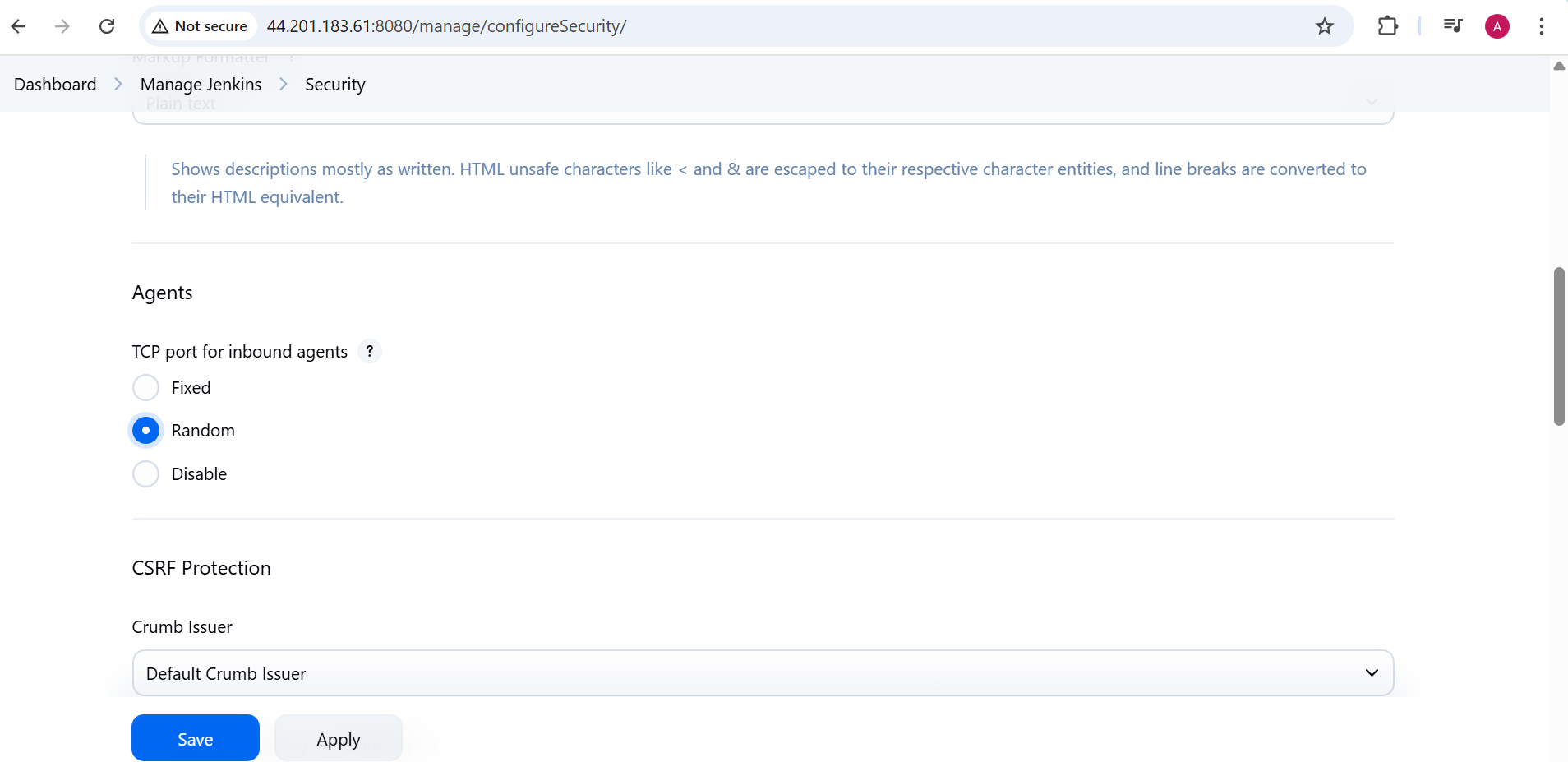




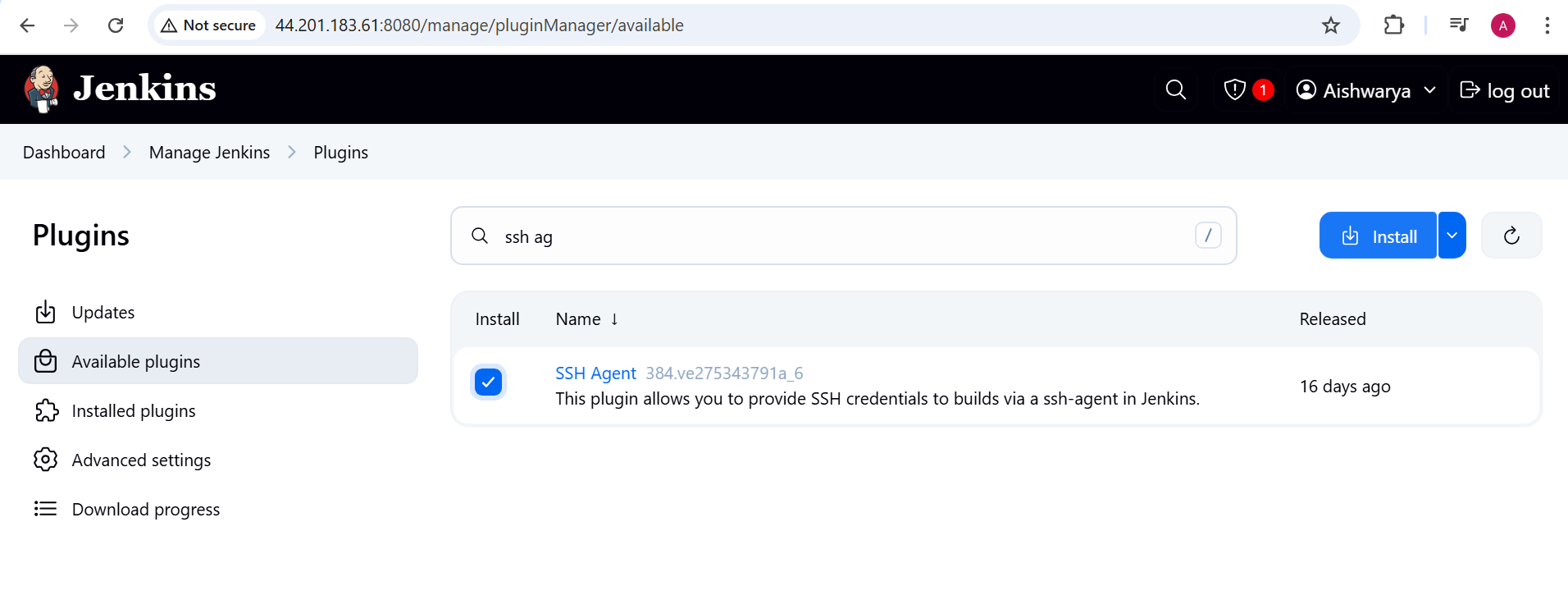
Go to manage jenkins



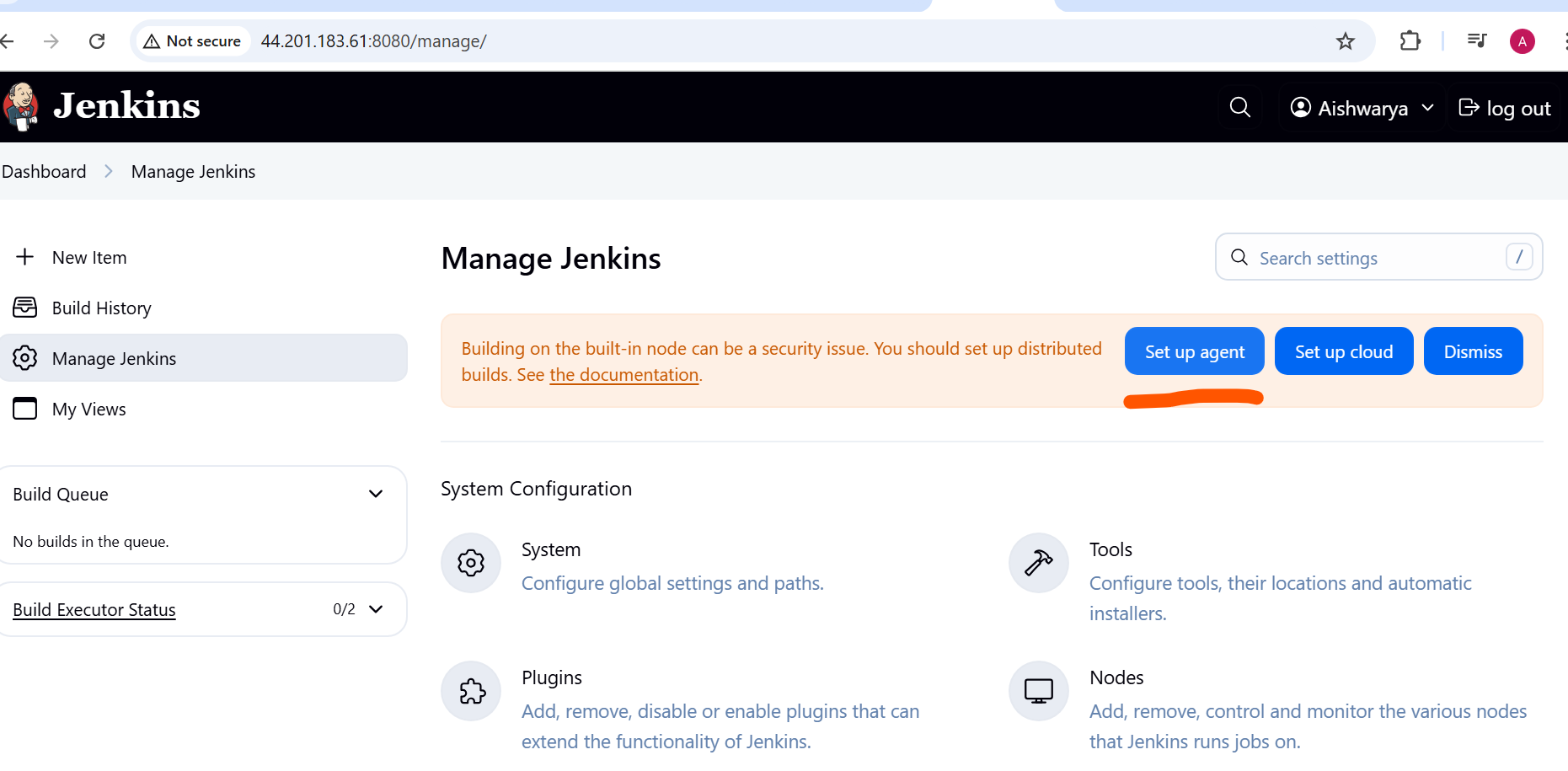
Security

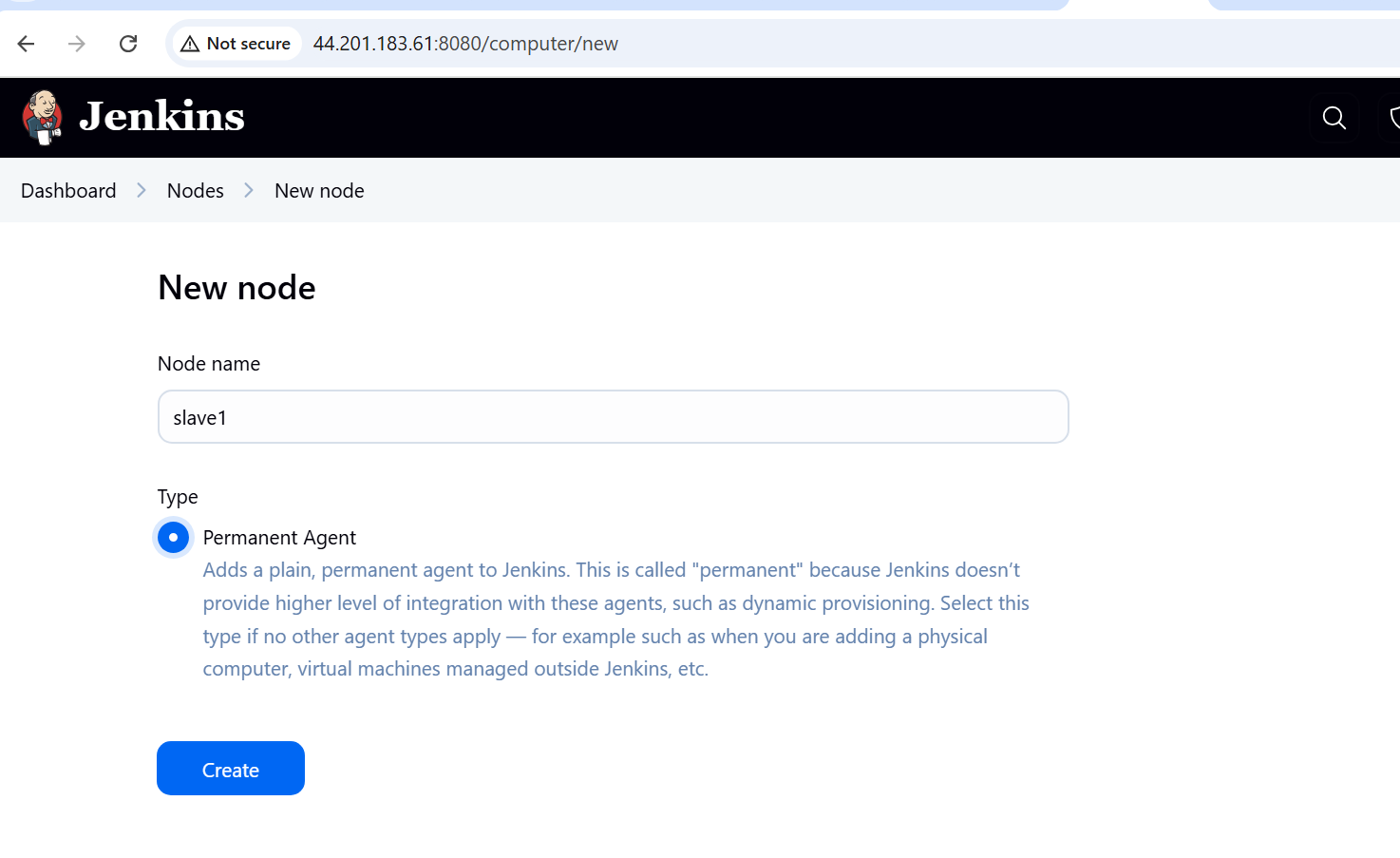


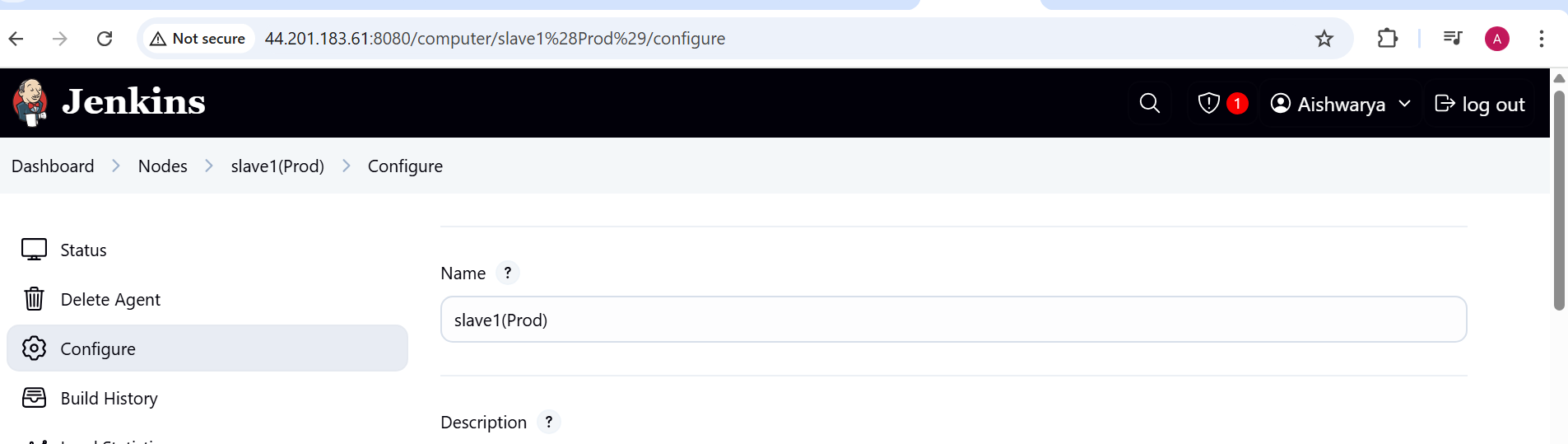
Now install necessary plugins



Now setup the agent:

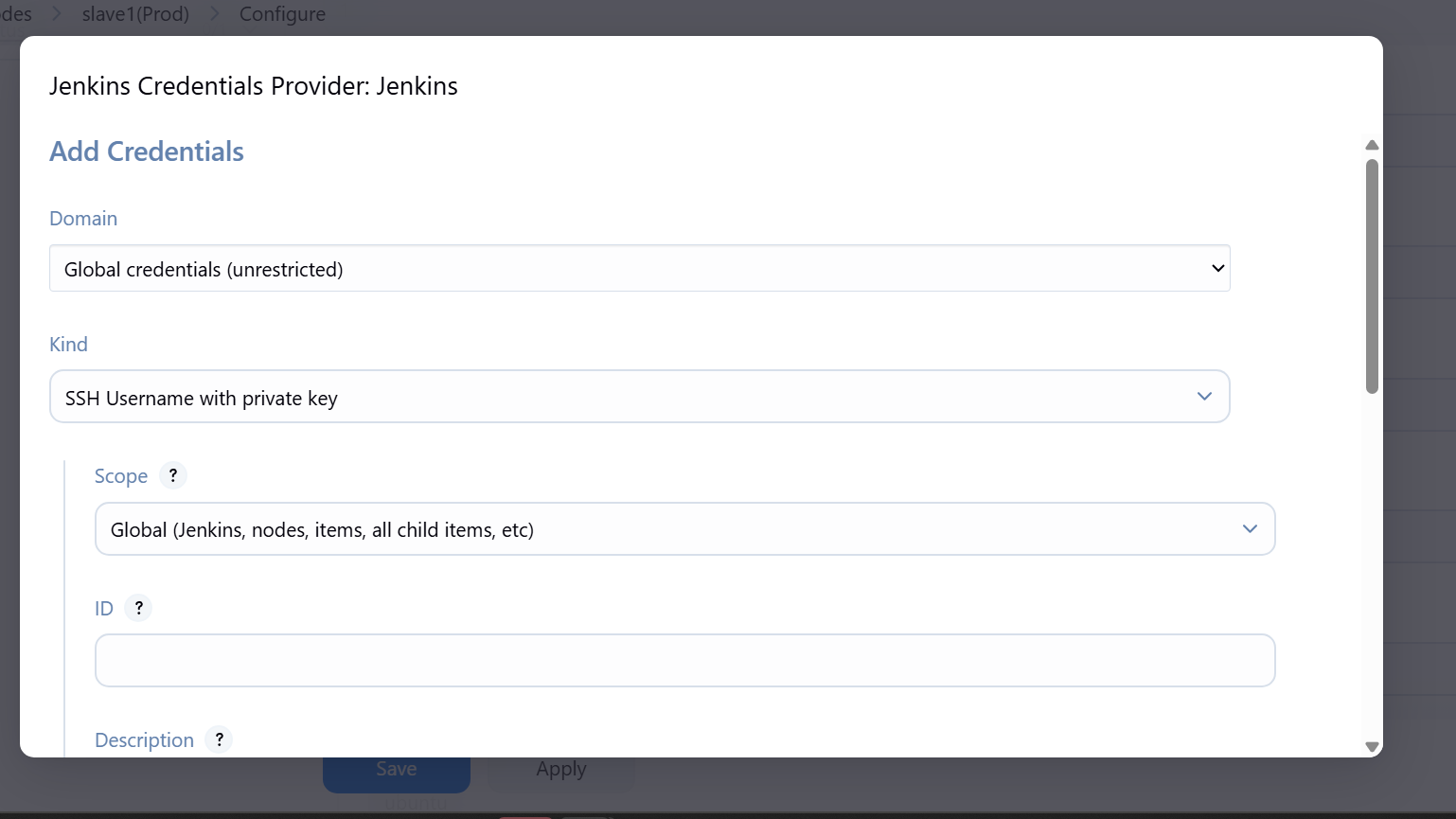




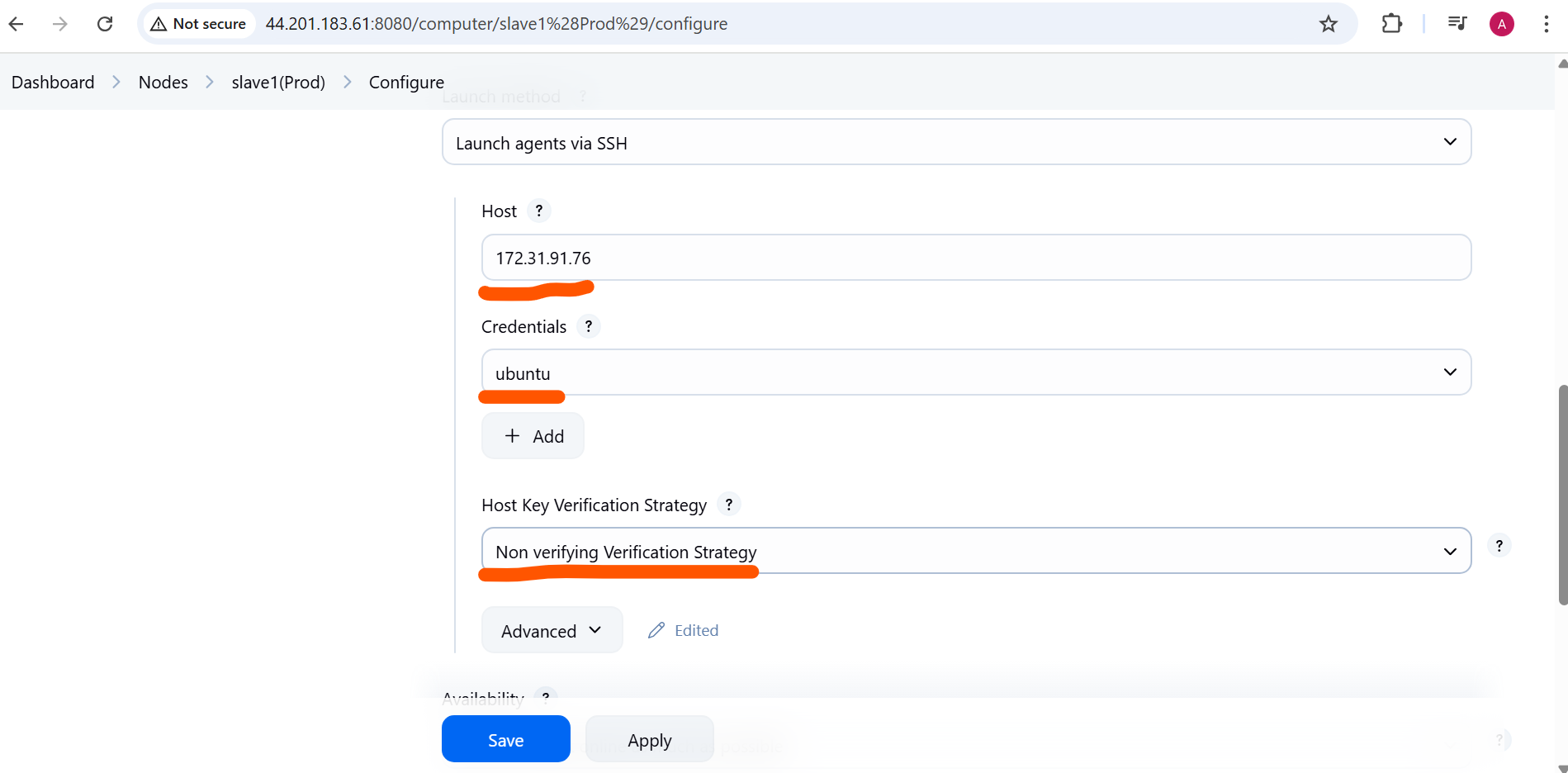




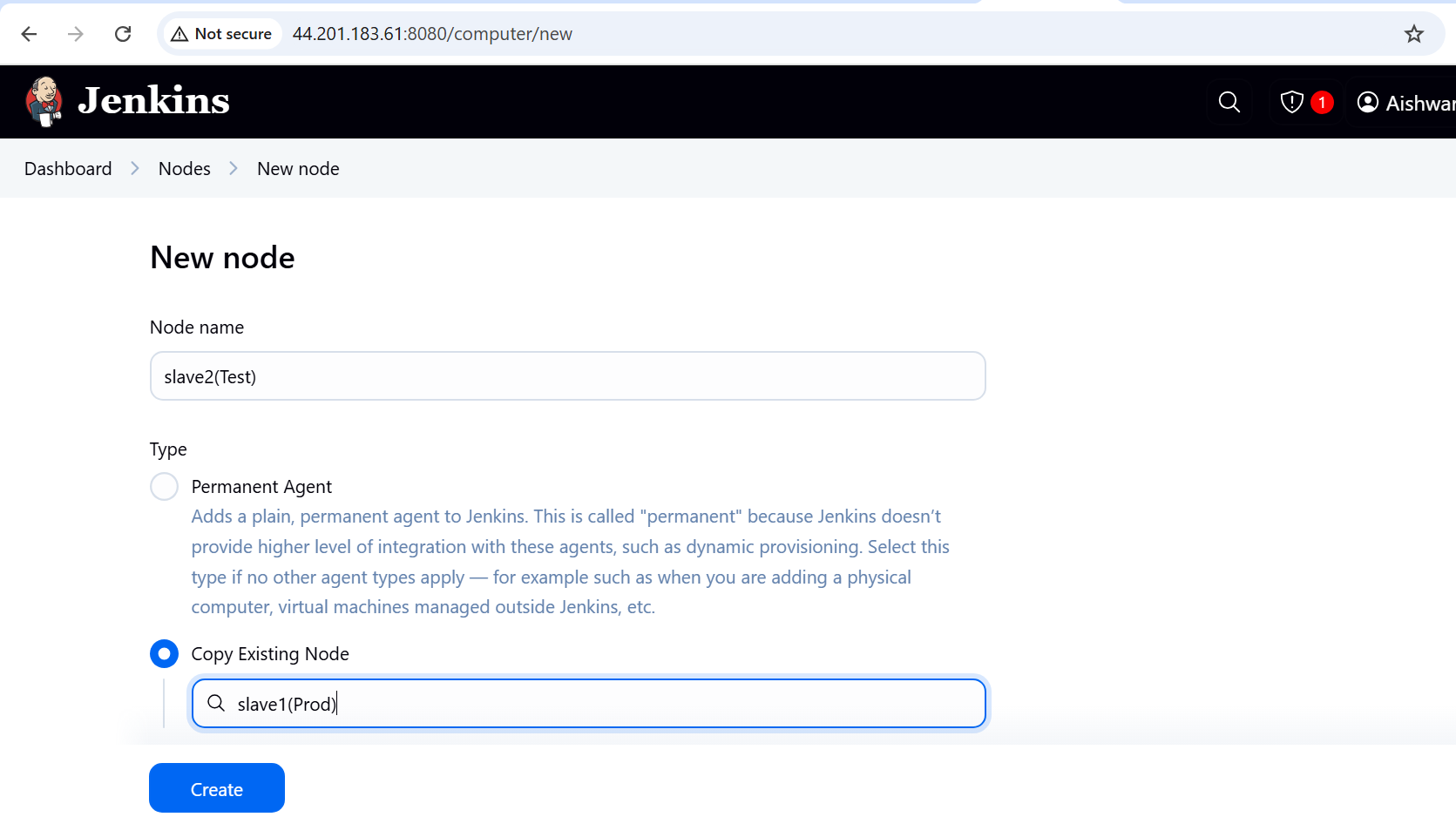
Create credential



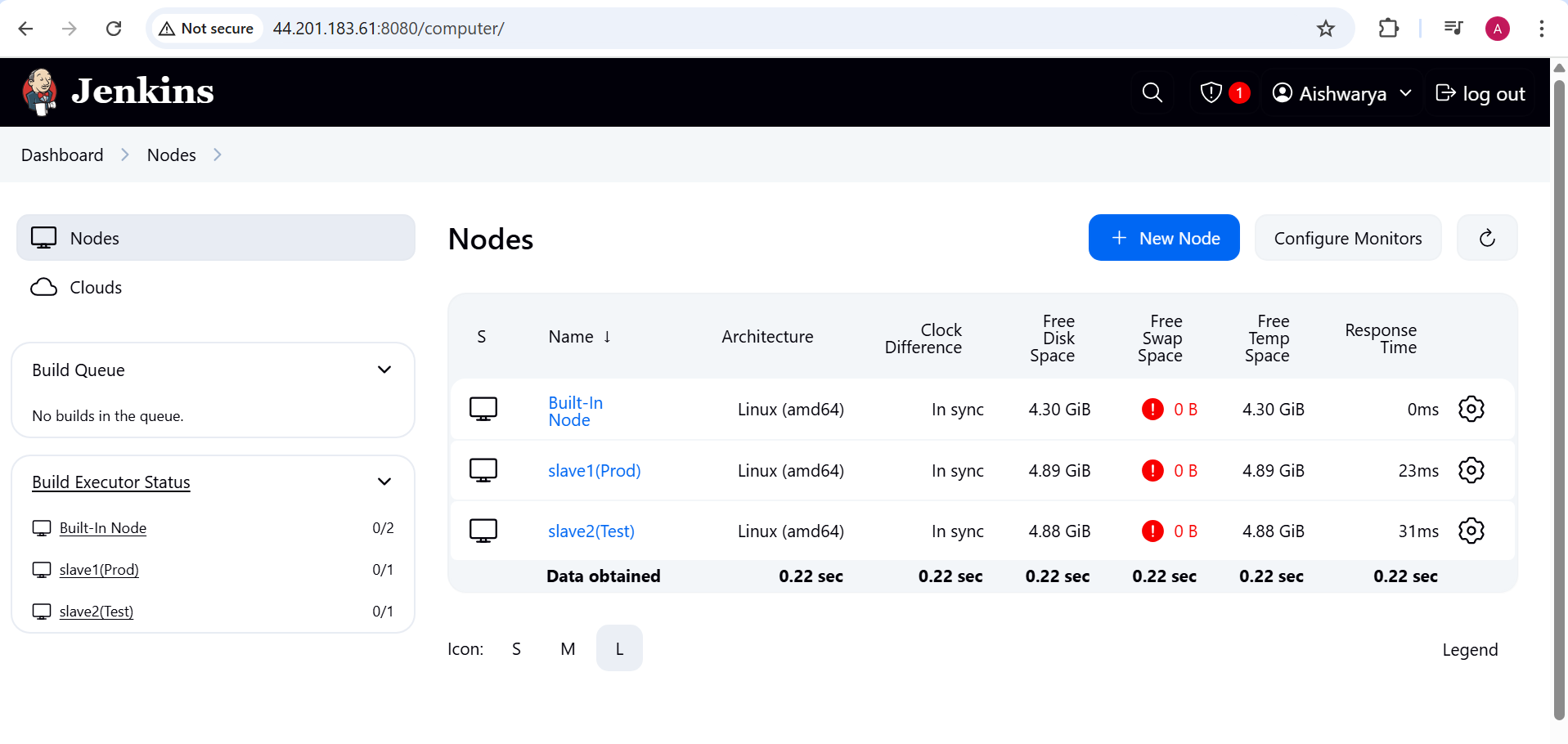
and entry .pem file (private key)



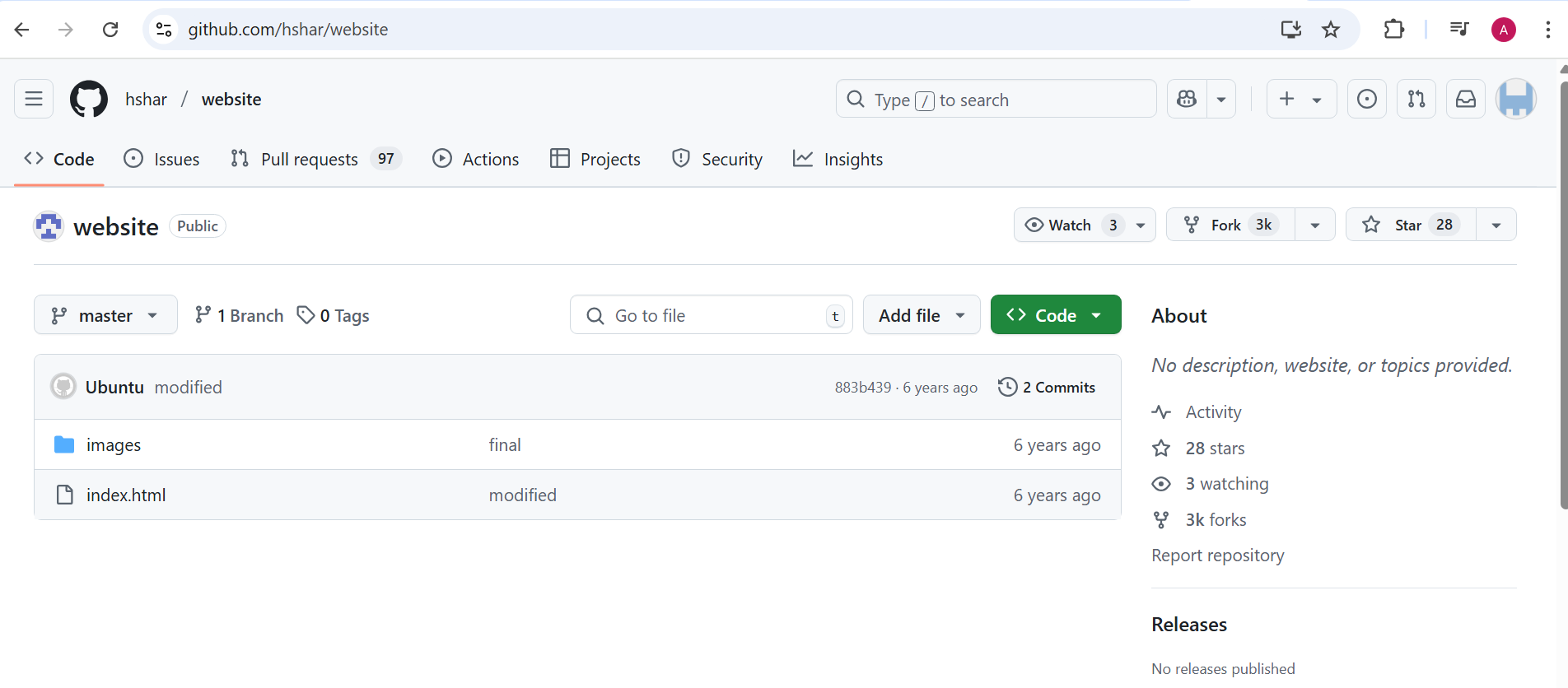
Now create new node for test

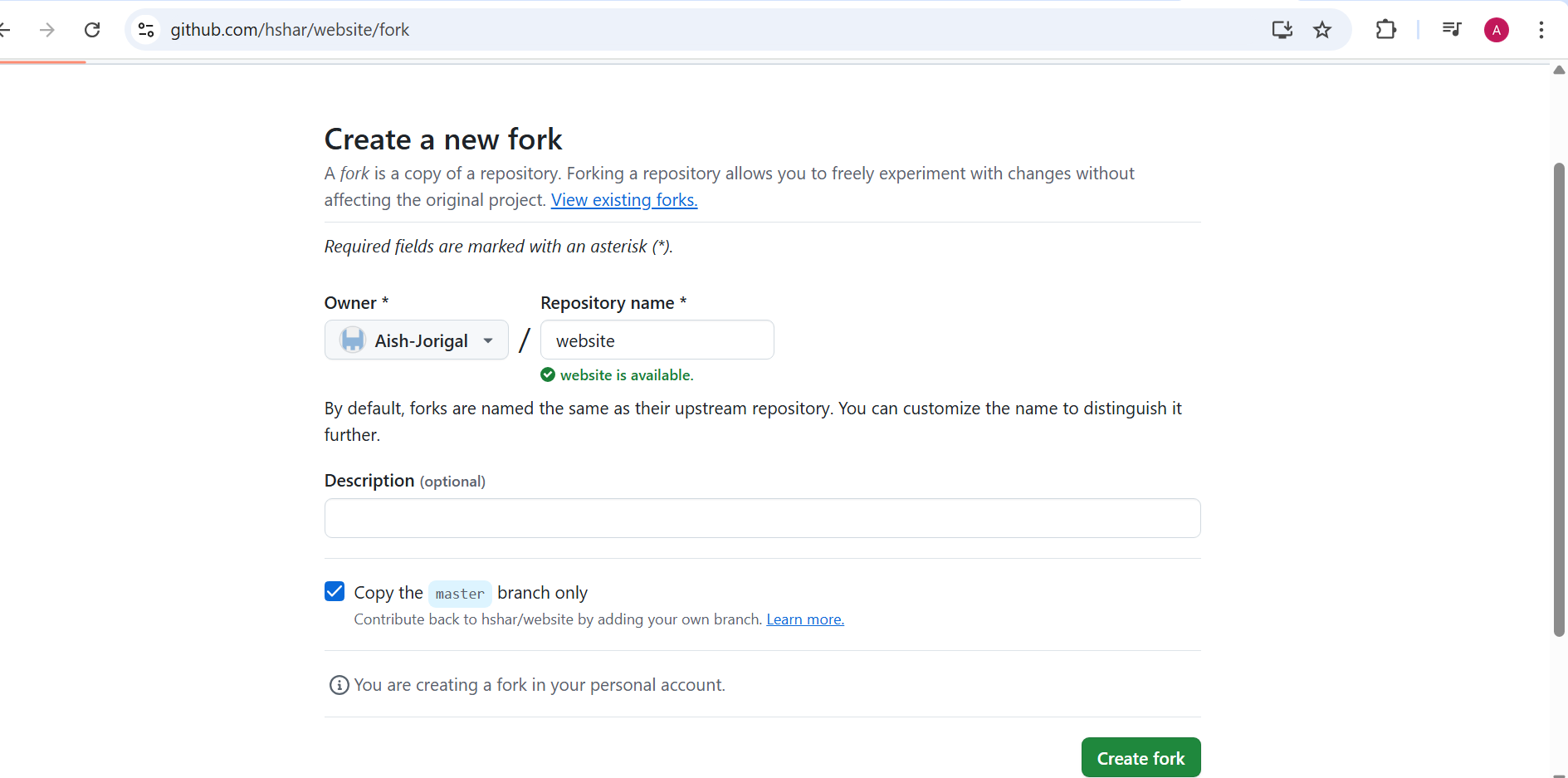


All nodes are connected

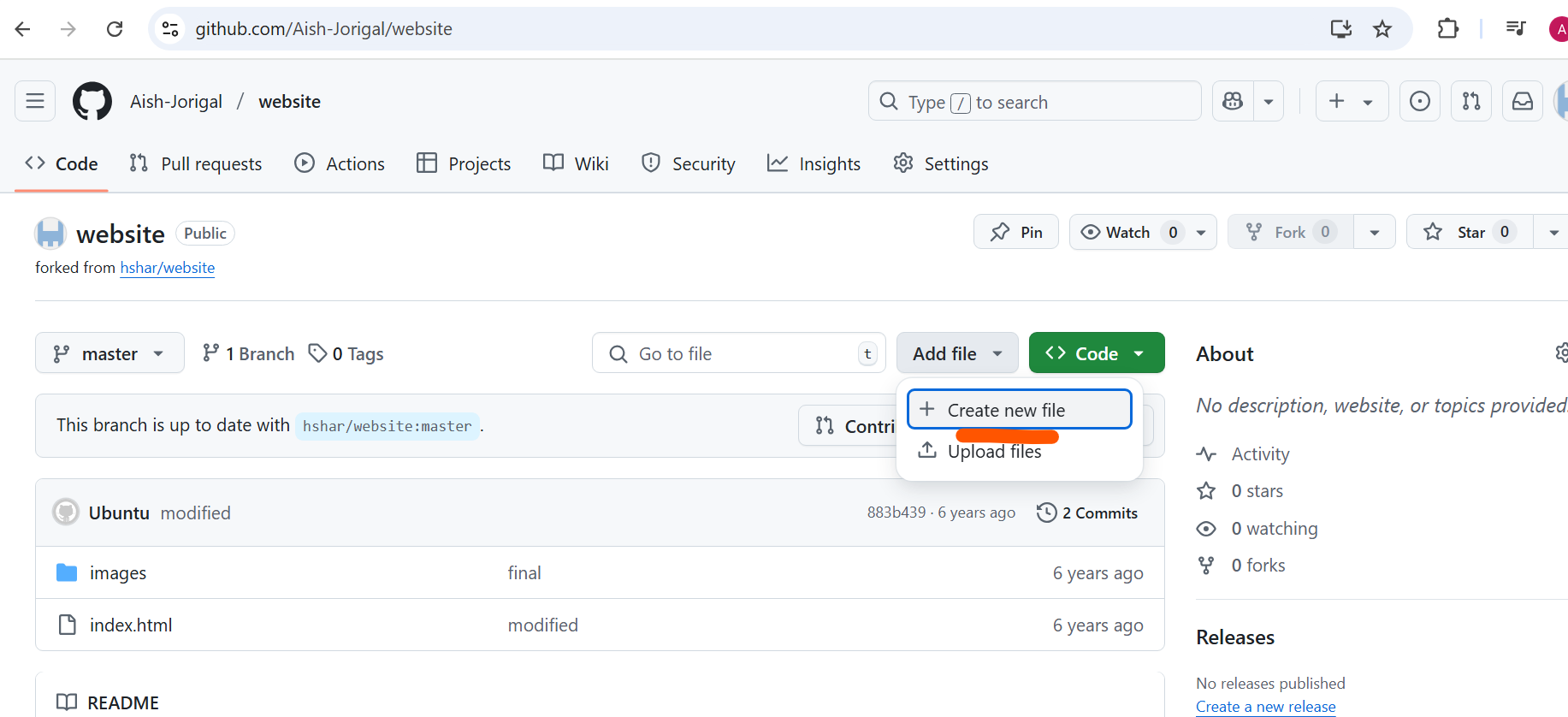


Now get the code

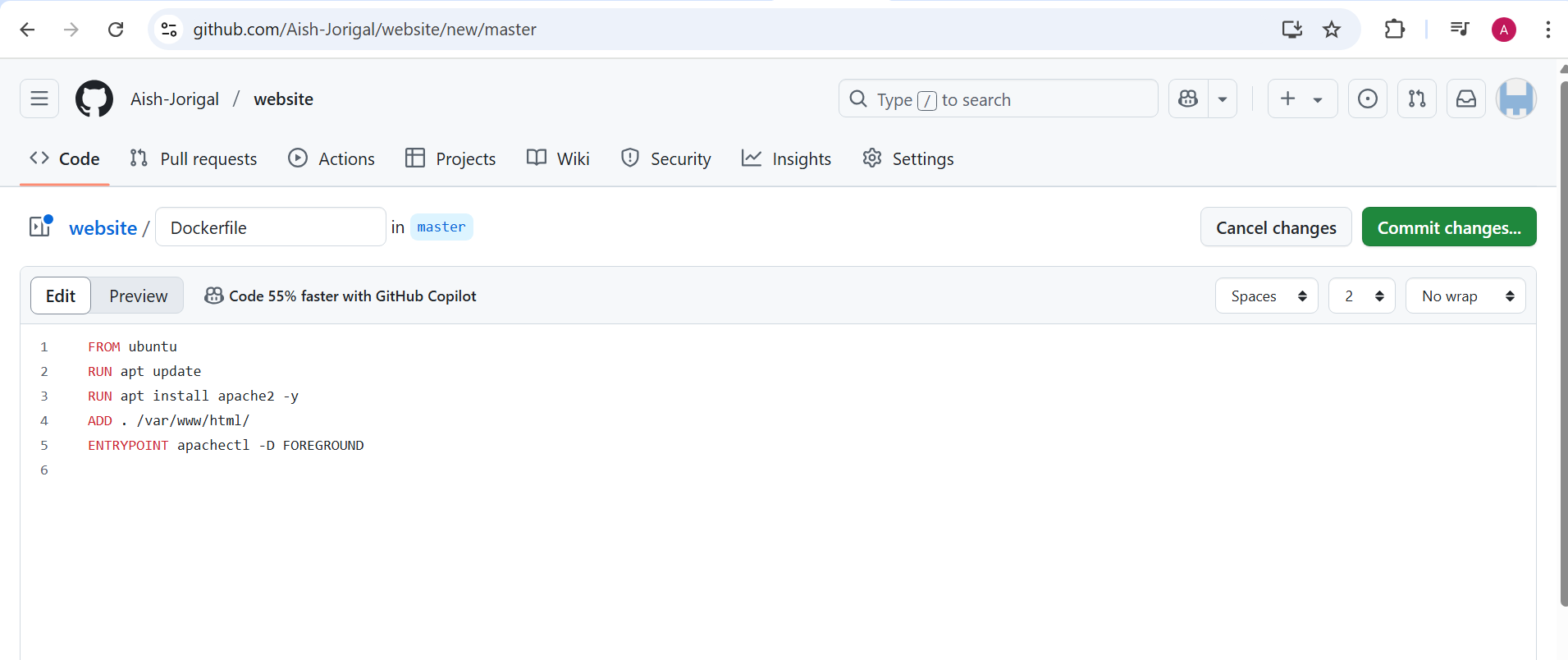
Fork the repository



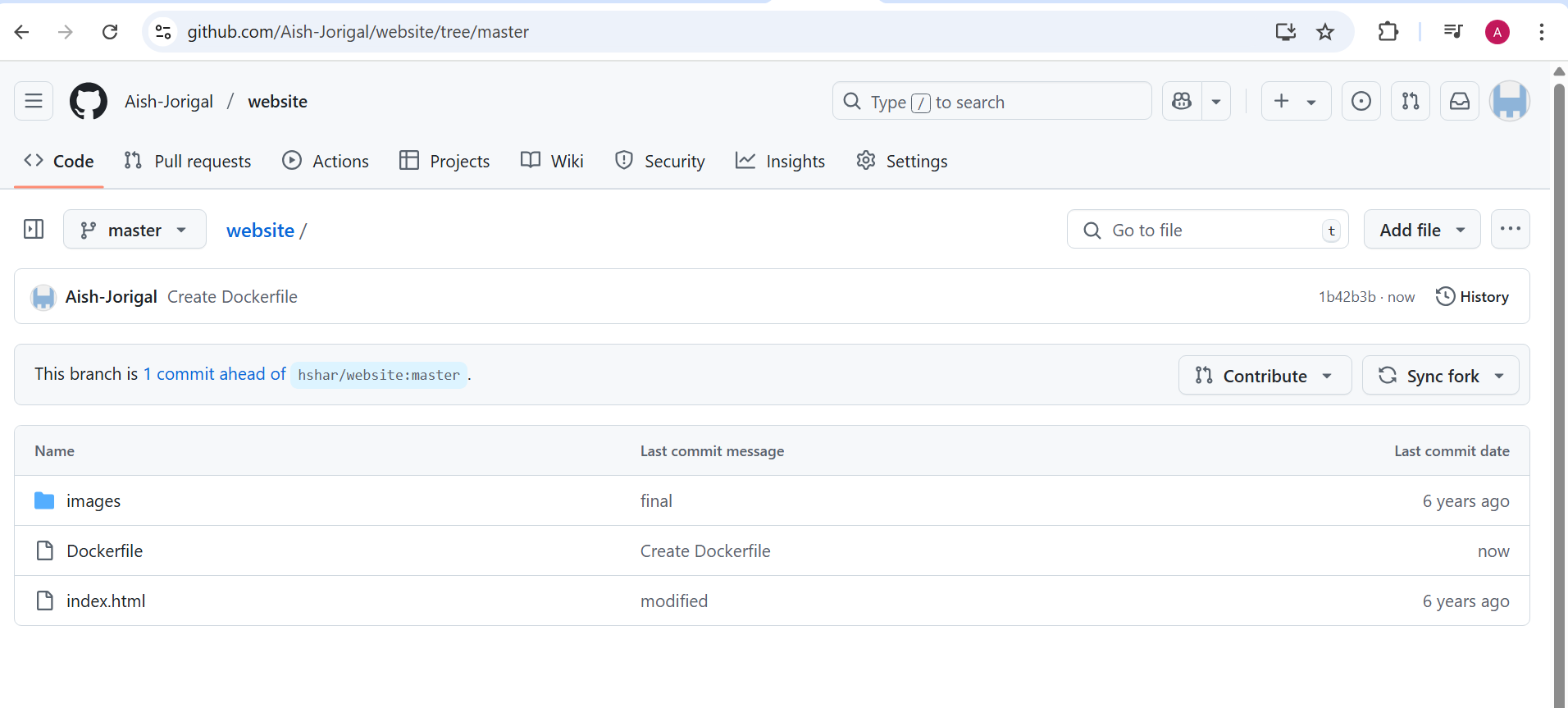
Create dockerfile



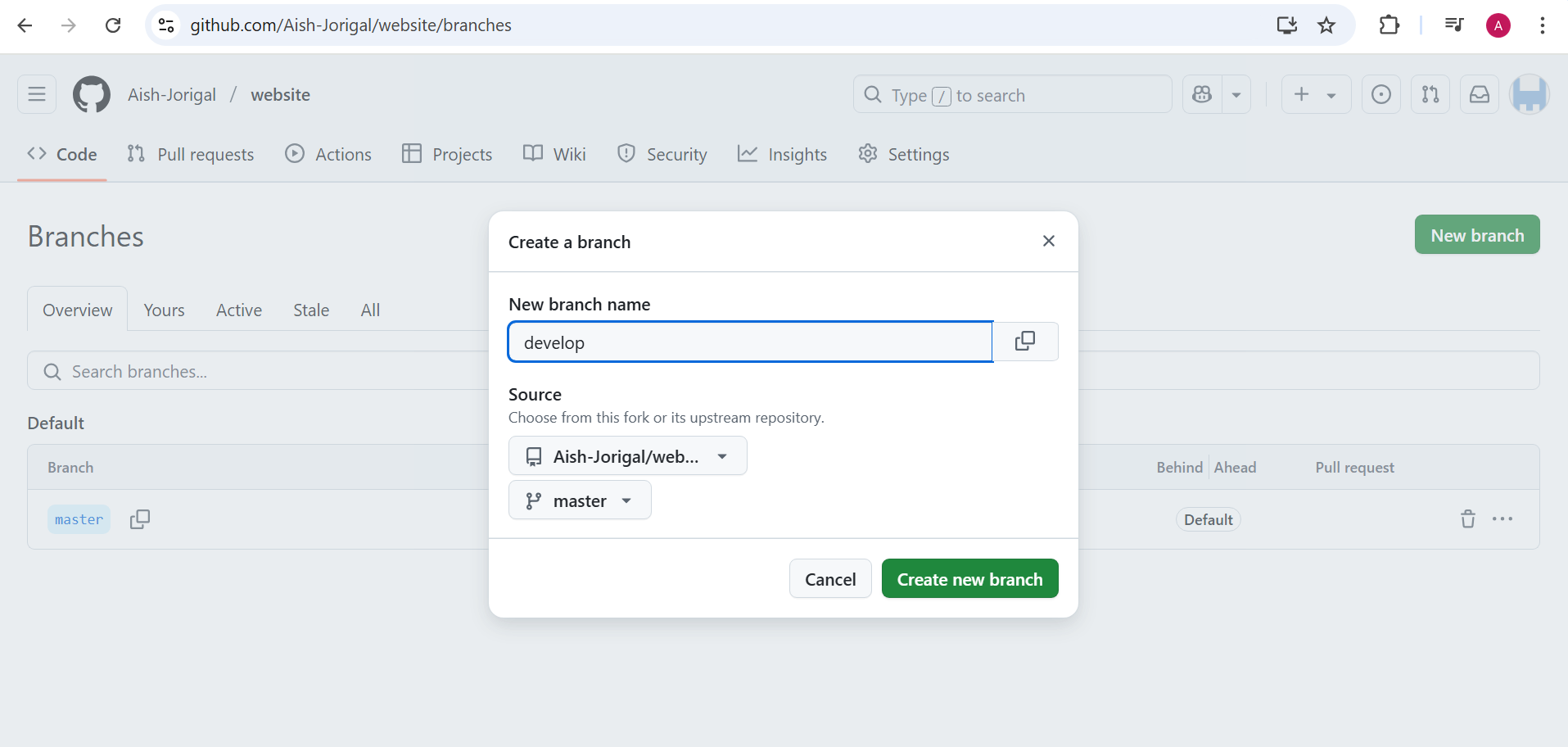
And commit the change



Now we have

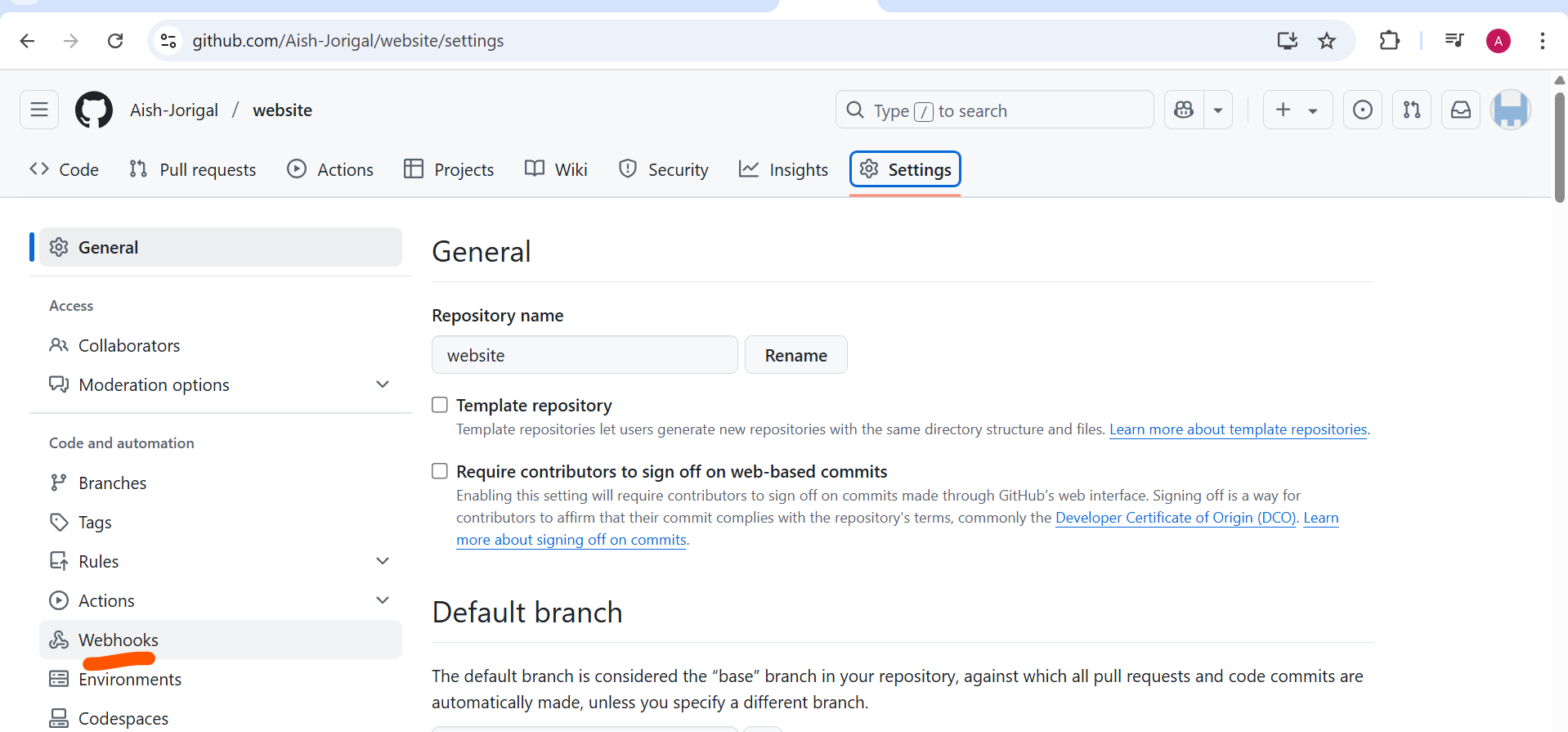


Create new branch develop

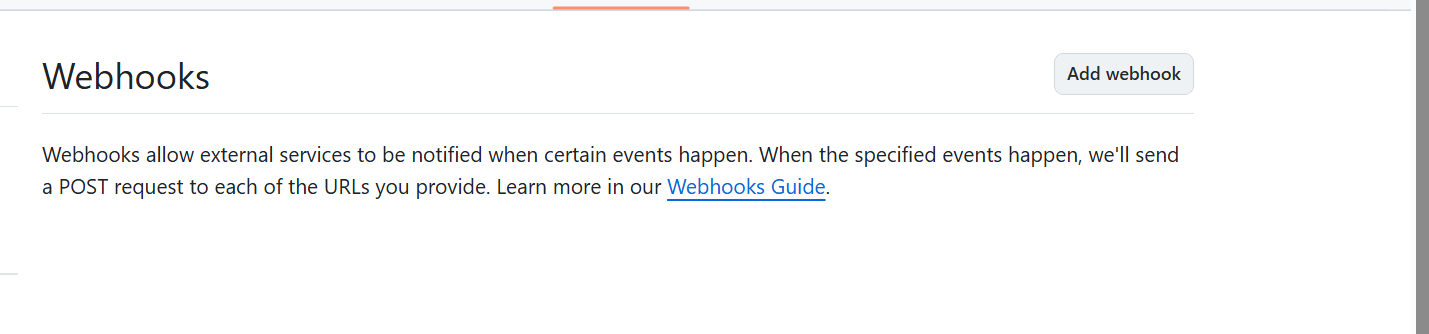


Enable webhook

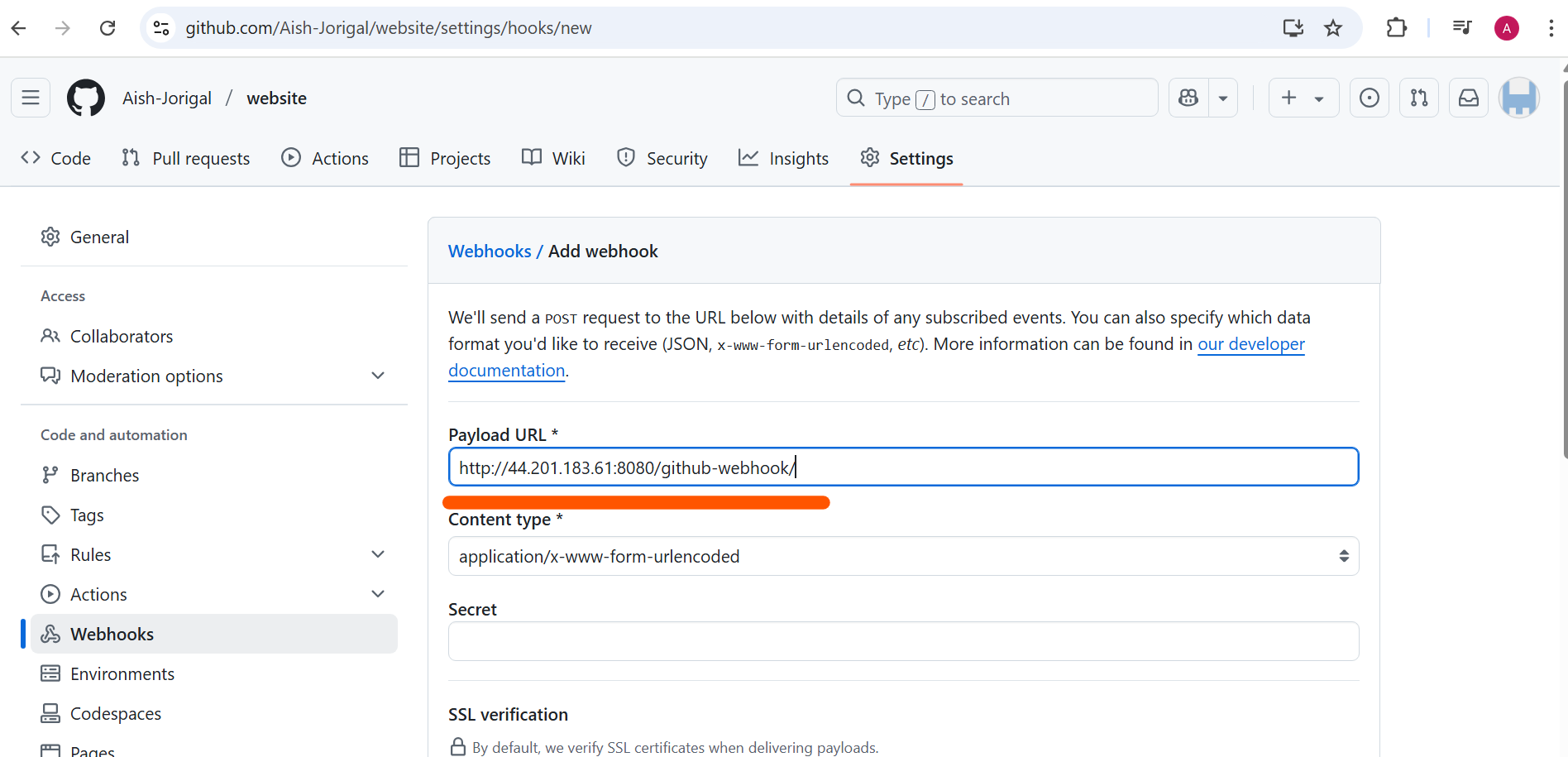
Setting



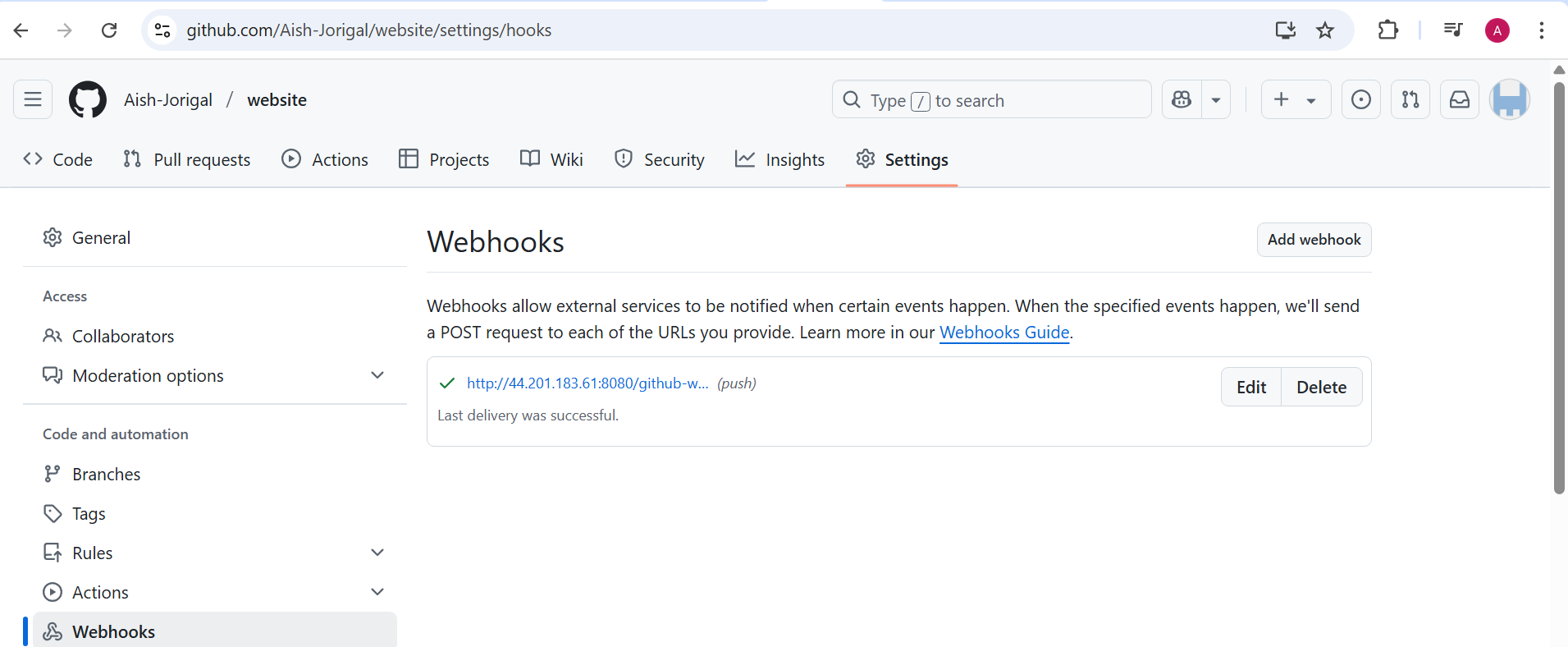
Add webhook



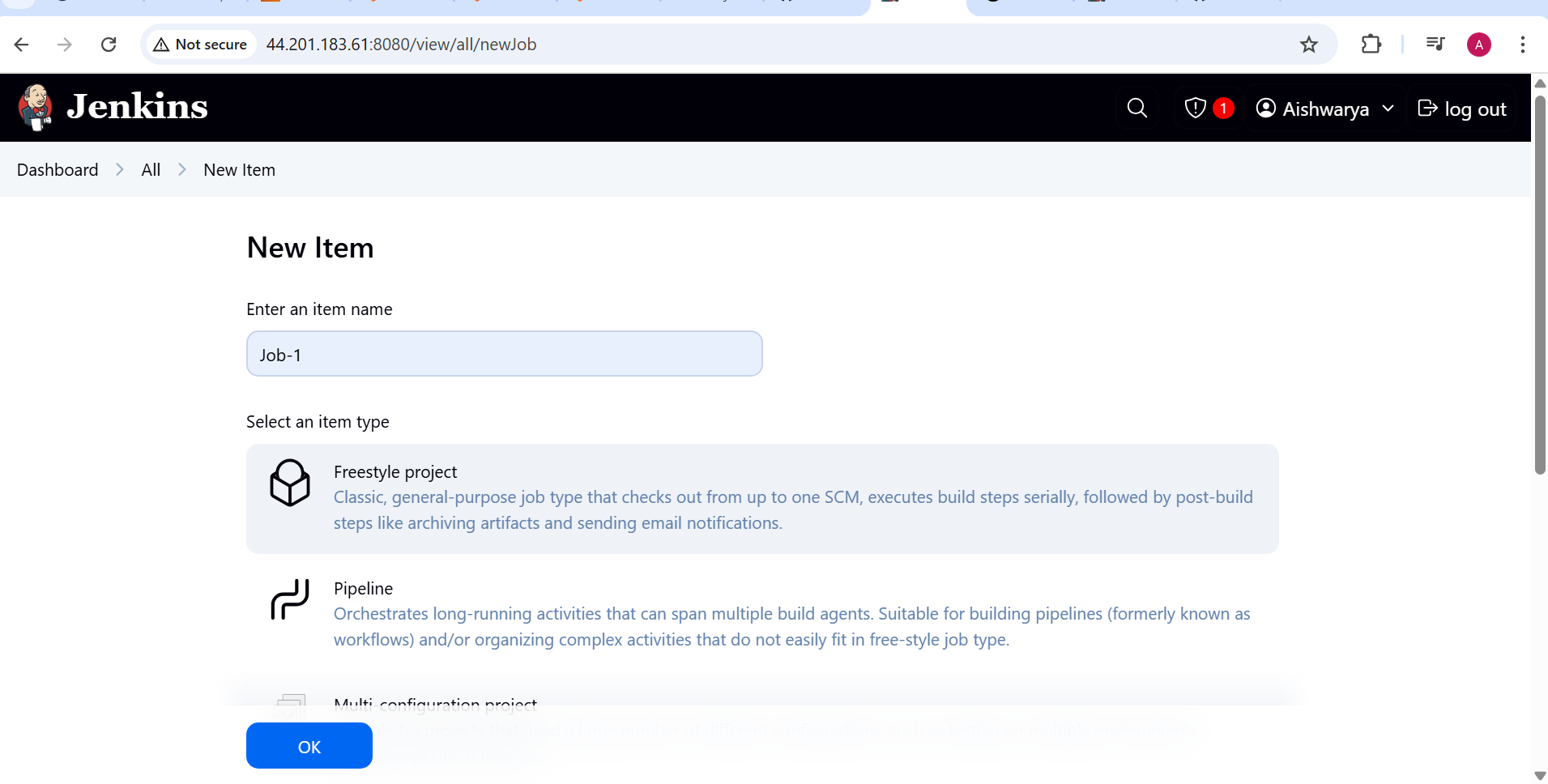
Add Jenkins URL

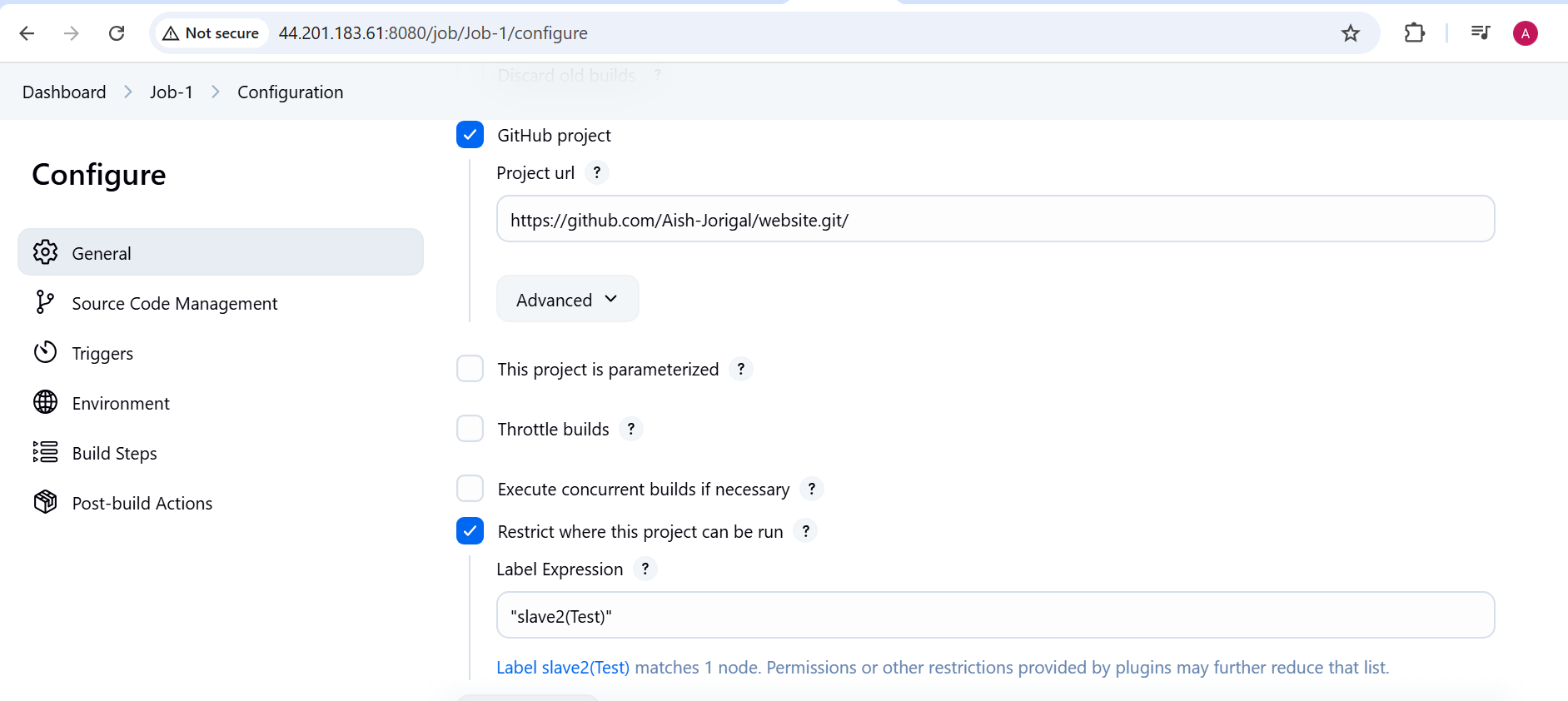


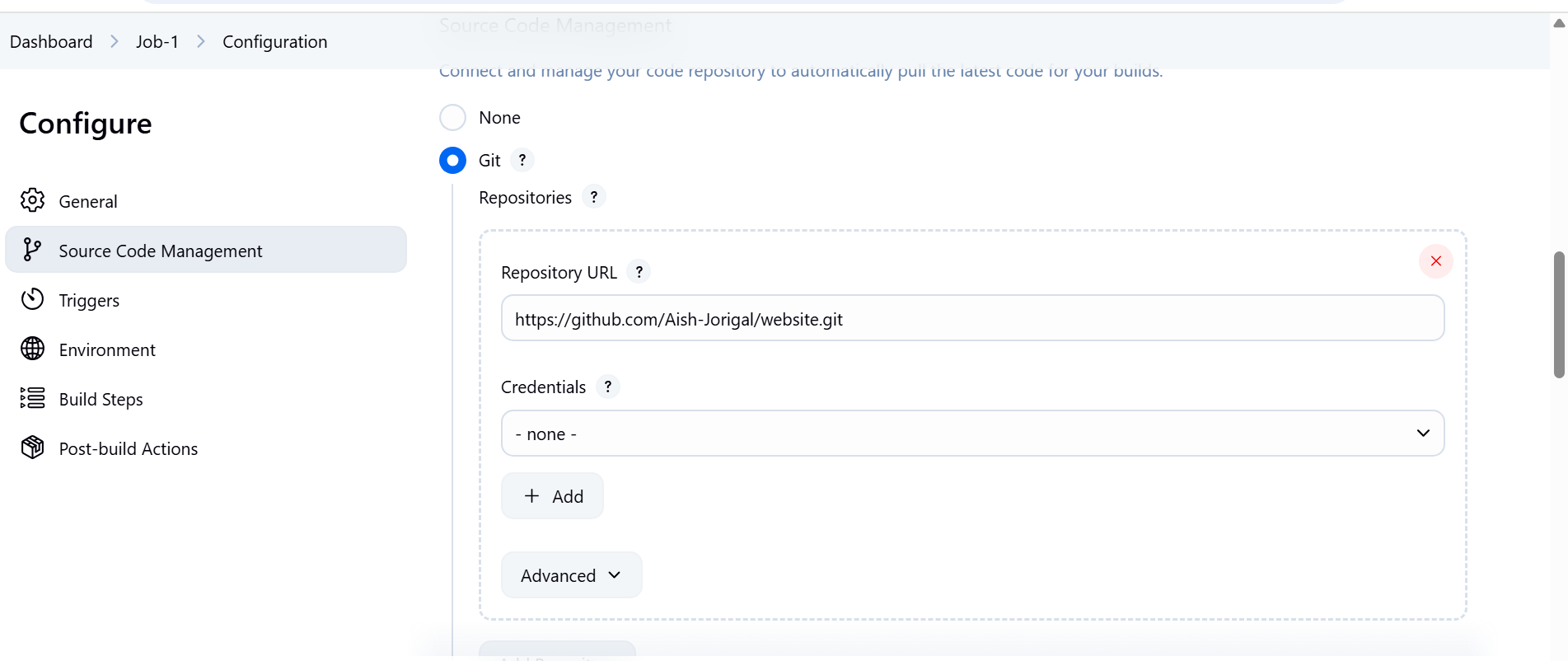
Webhook created

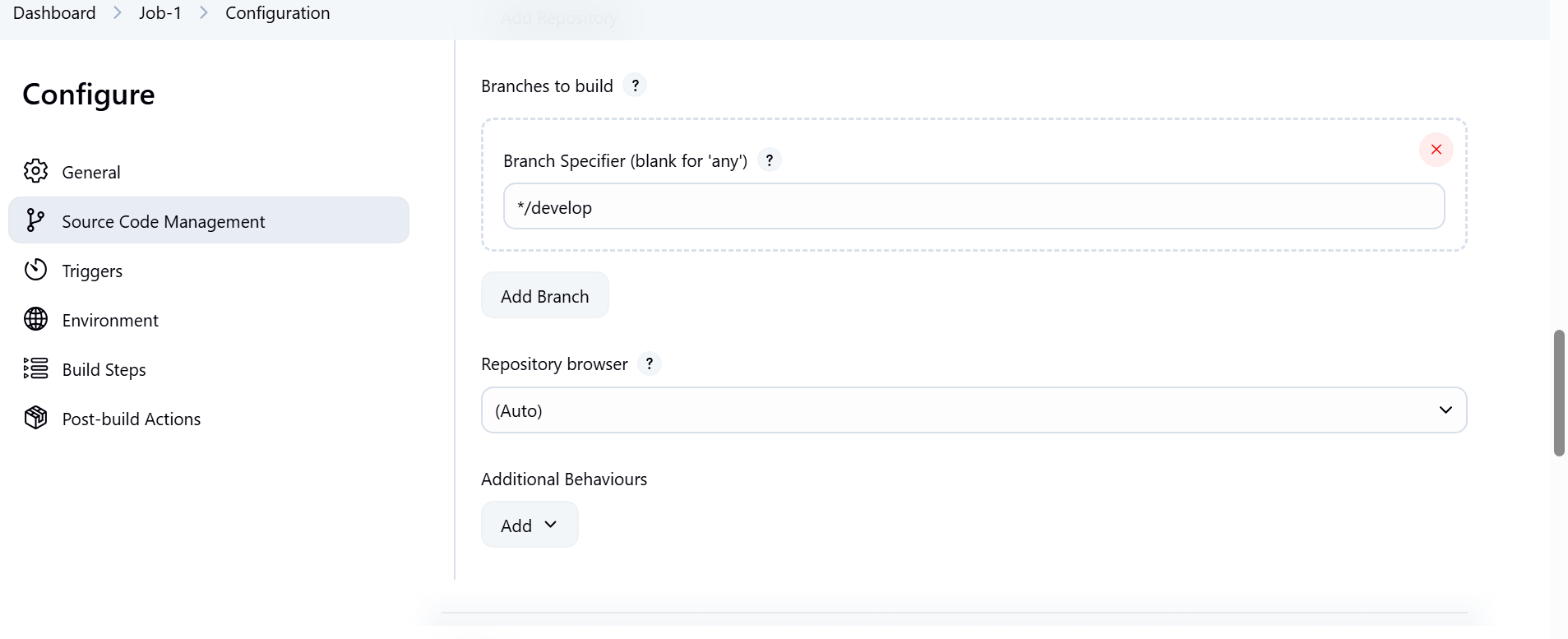


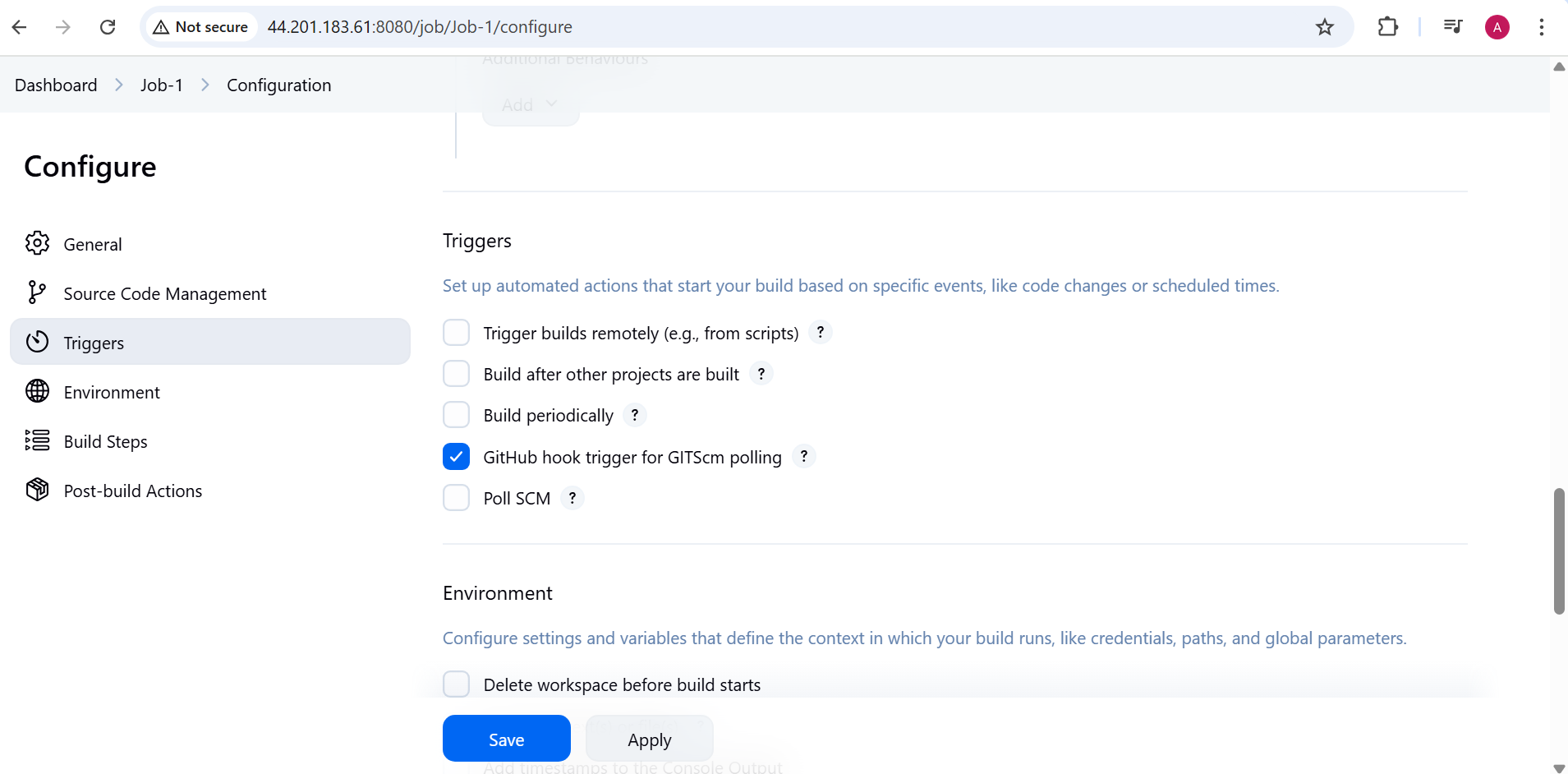
Now create job





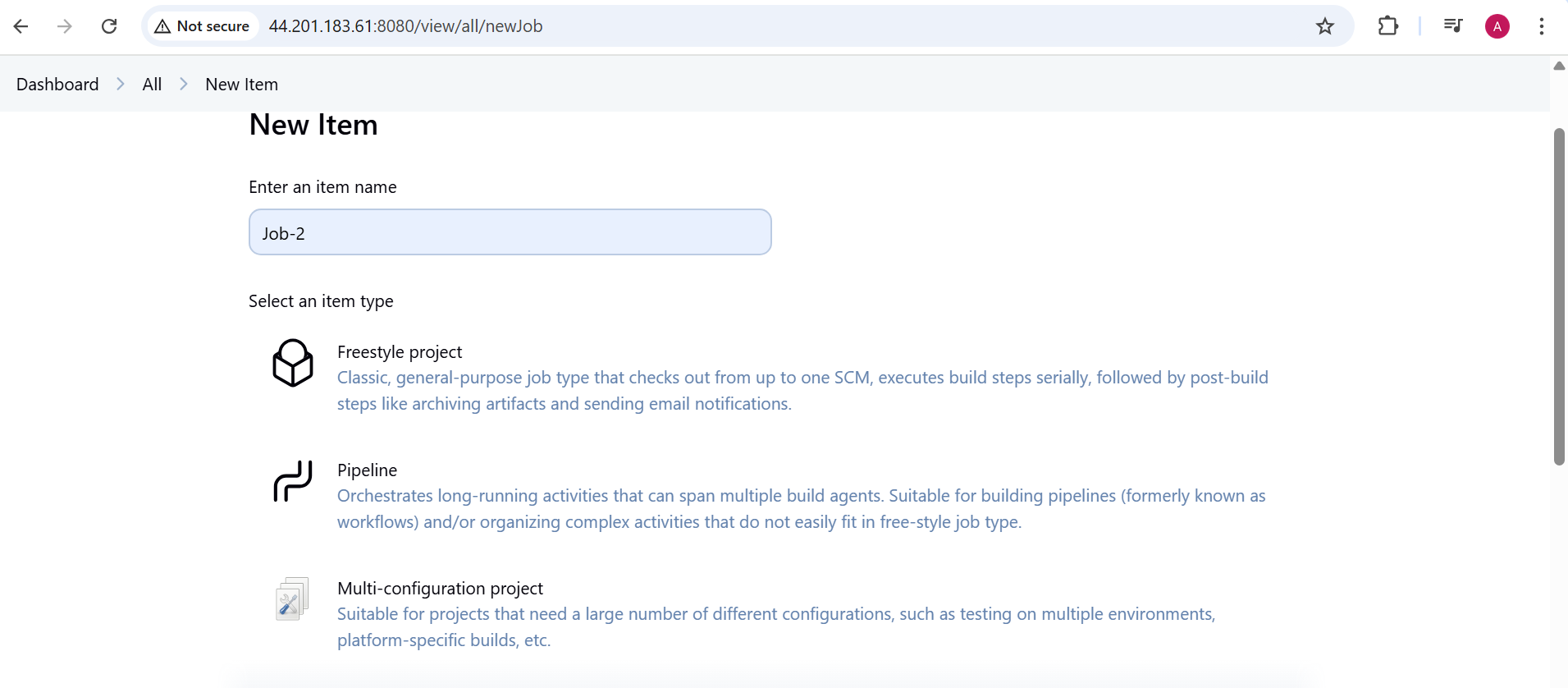


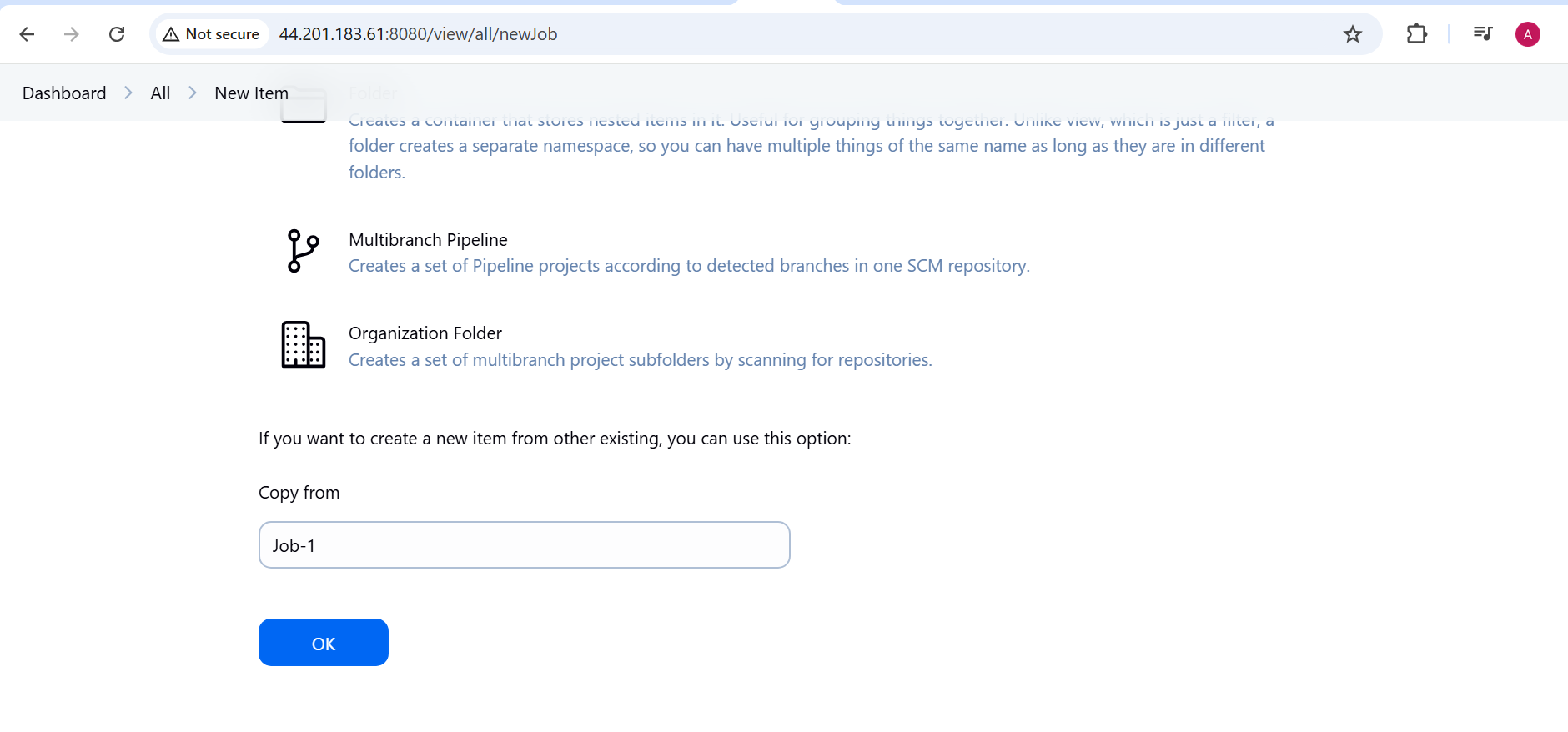
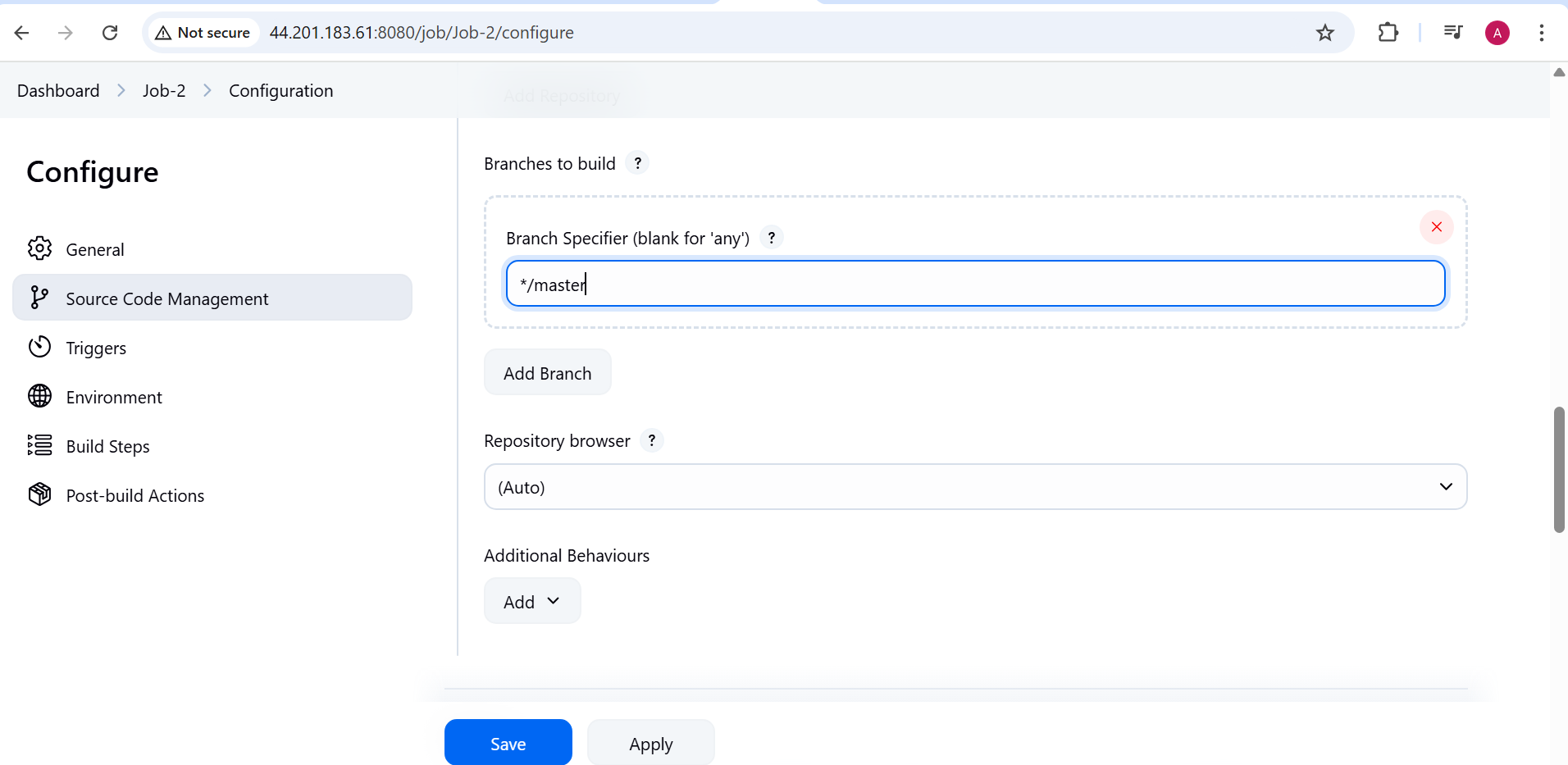




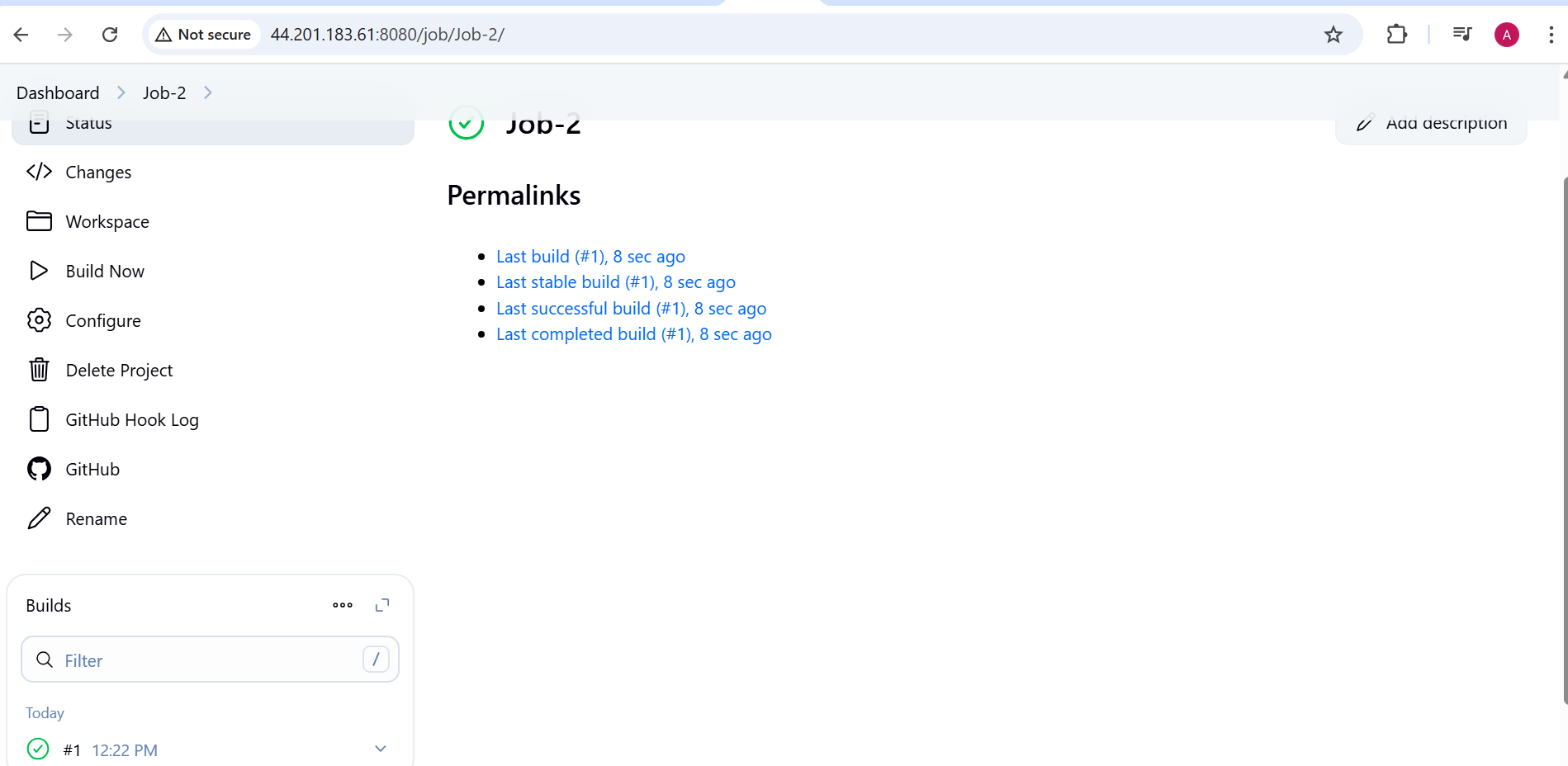


Create job2

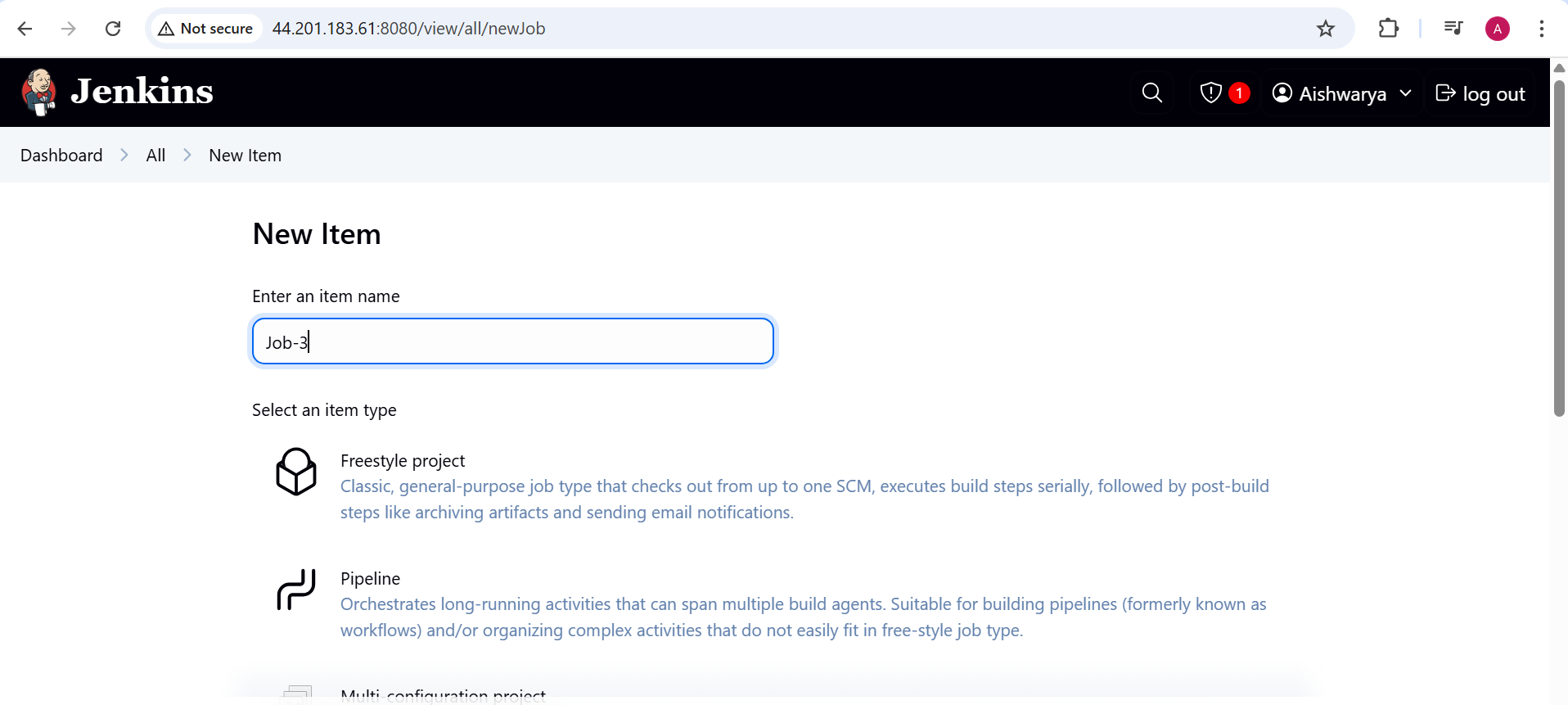
copy from Job-1 only need to change the branch from develop to master

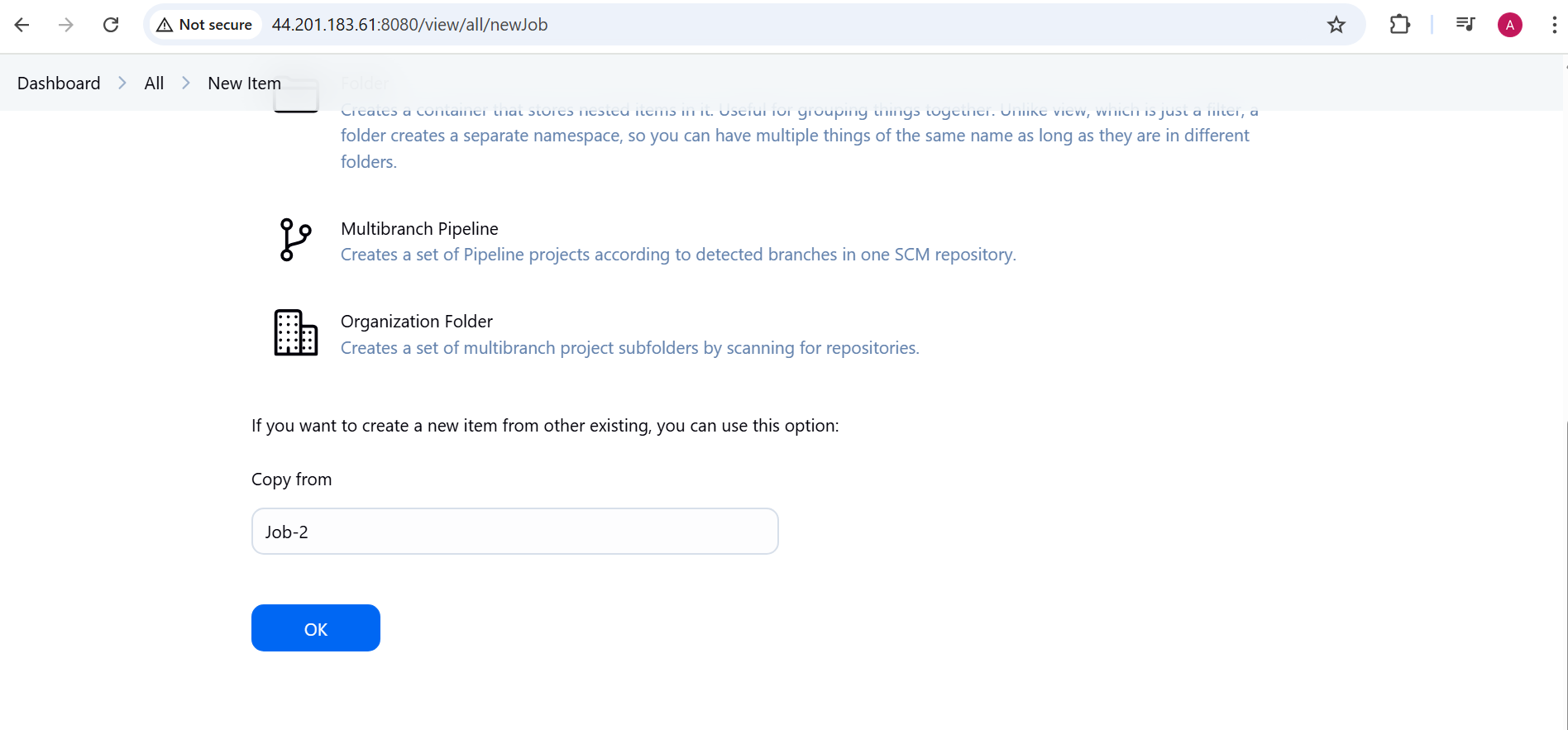
Build now-

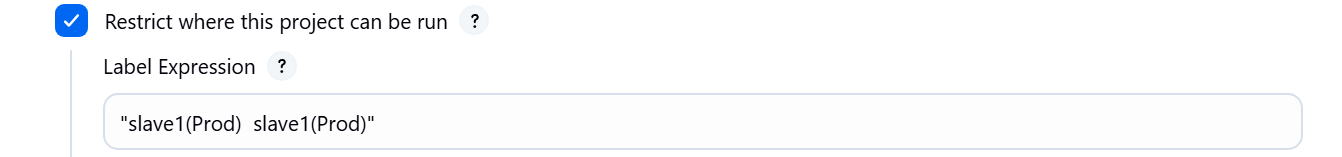


Now create Job3

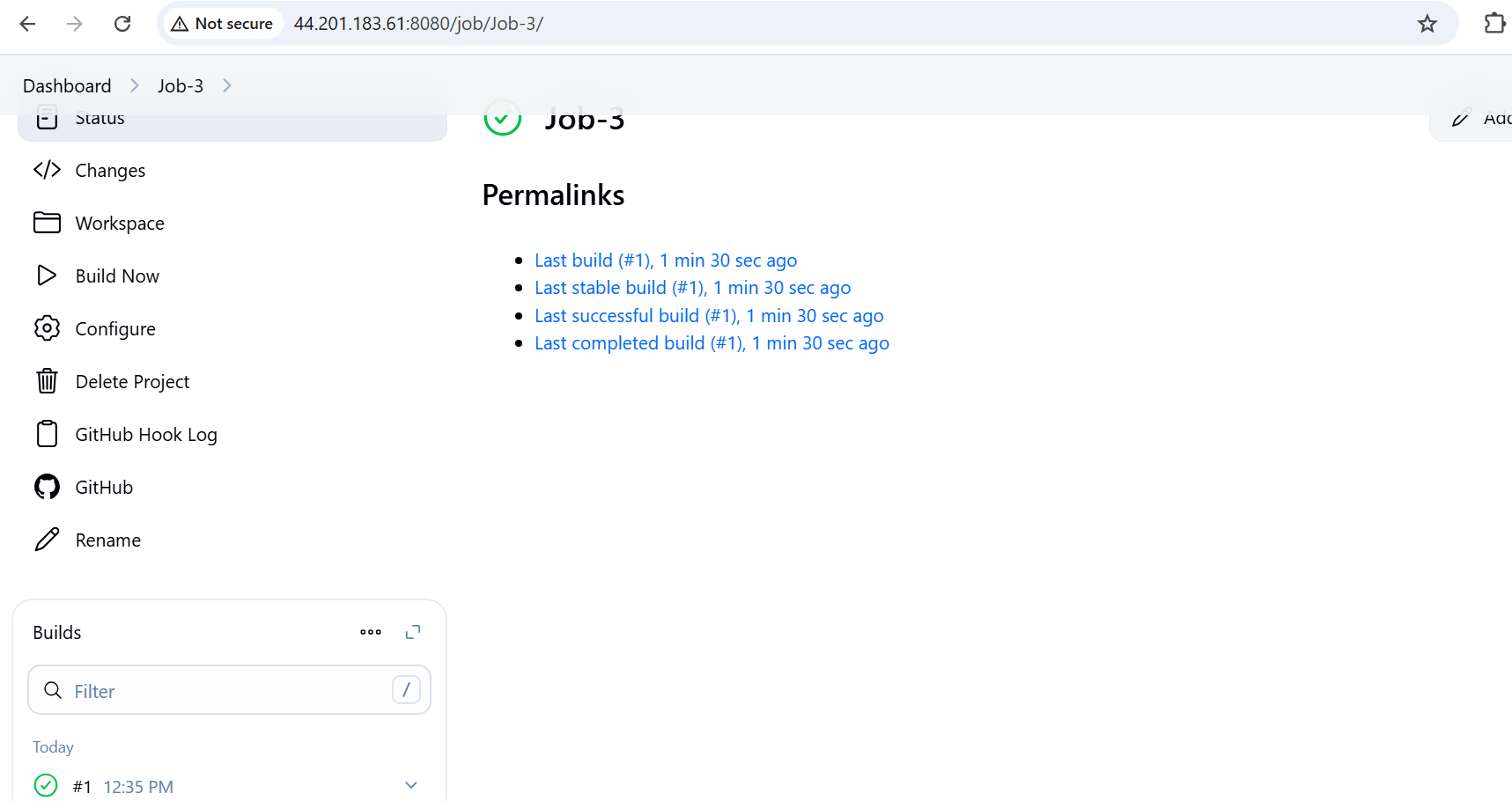


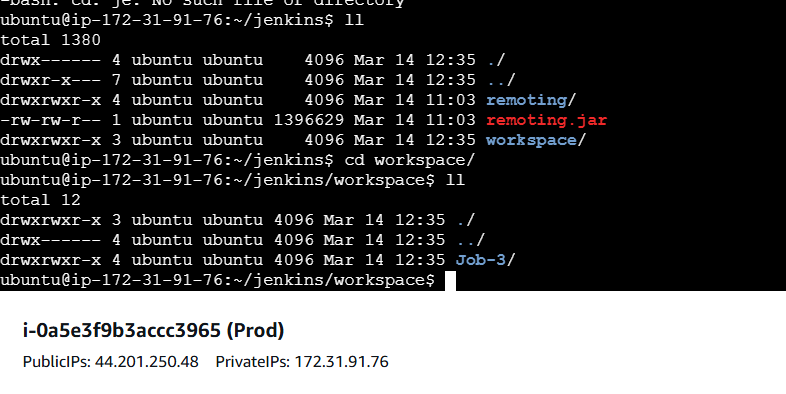
Copy from Job-2 only need to change slave1

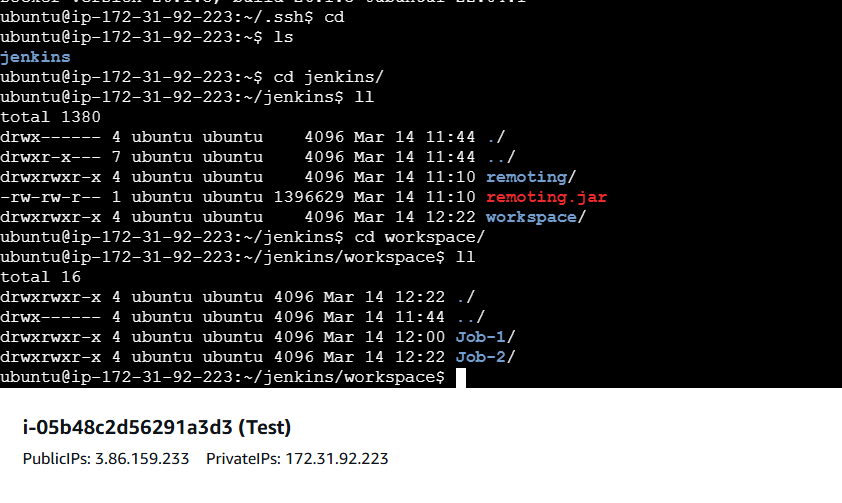




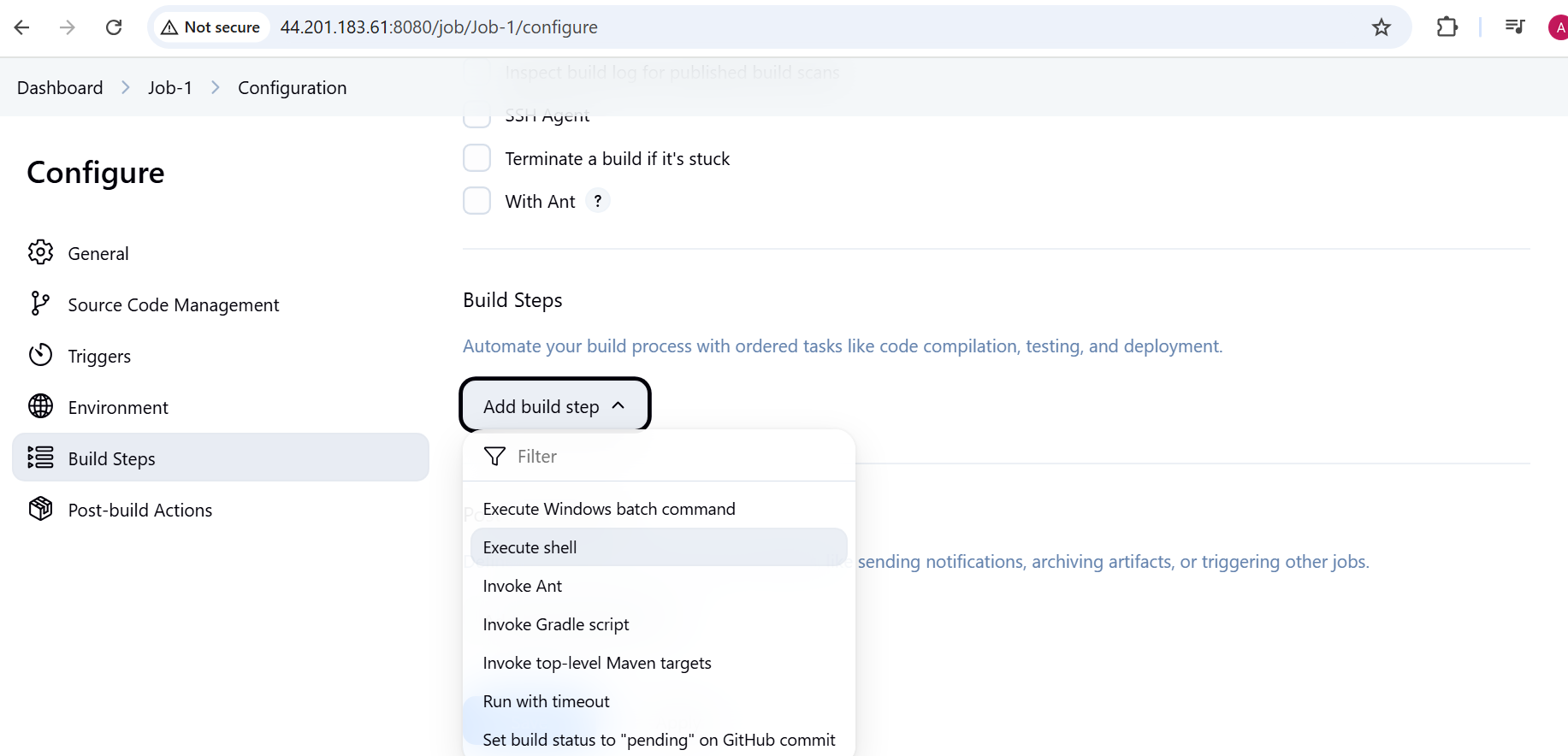
Build Now;





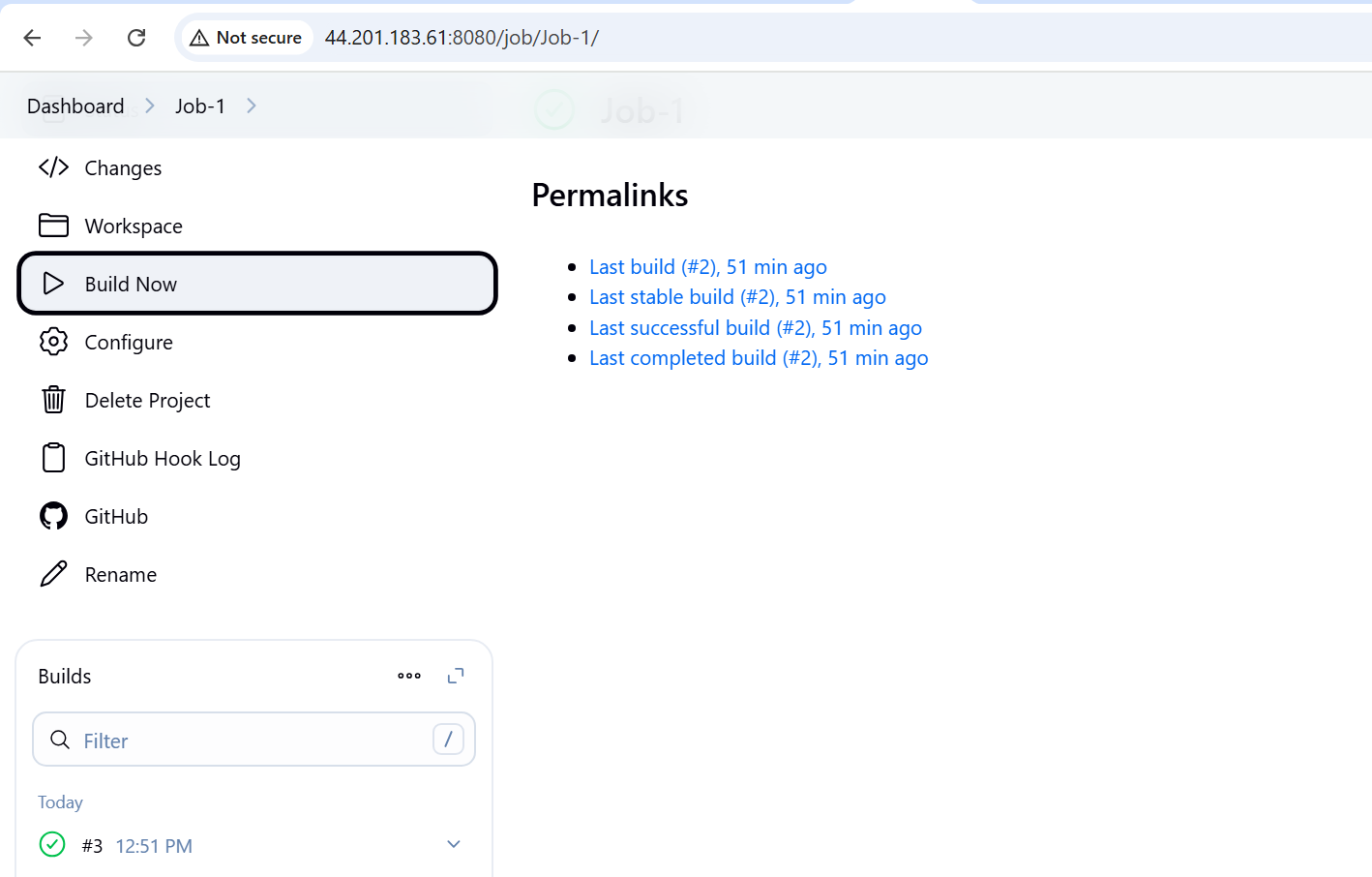


Now dockerise the application

JOB1 



Build now:



Take the public Ip of machine which is configured for this Job and put the port which is used in the run cmd



for future:



Updating the index.html to check webhook

