1. Install jenkins and run jenkins on port number 8081.

Launch ec2 sever

A screenshot of a computer

AI-generated content may be incorrect.

ssh -i "jenkins.pem" [ec2-user@204.236.201.136](mailto:ec2-user@204.236.201.136)

A screenshot of a computer

AI-generated content may be incorrect.

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 upgrade

sudo rpm --import https://yum.corretto.aws/corretto.key

sudo curl -Lo /etc/yum.repos.d/corretto.repo https://yum.corretto.aws/corretto.repo

sudo yum install -y java-17-amazon-correttosudo yum install jenkins

sudo systemctl daemon-reload

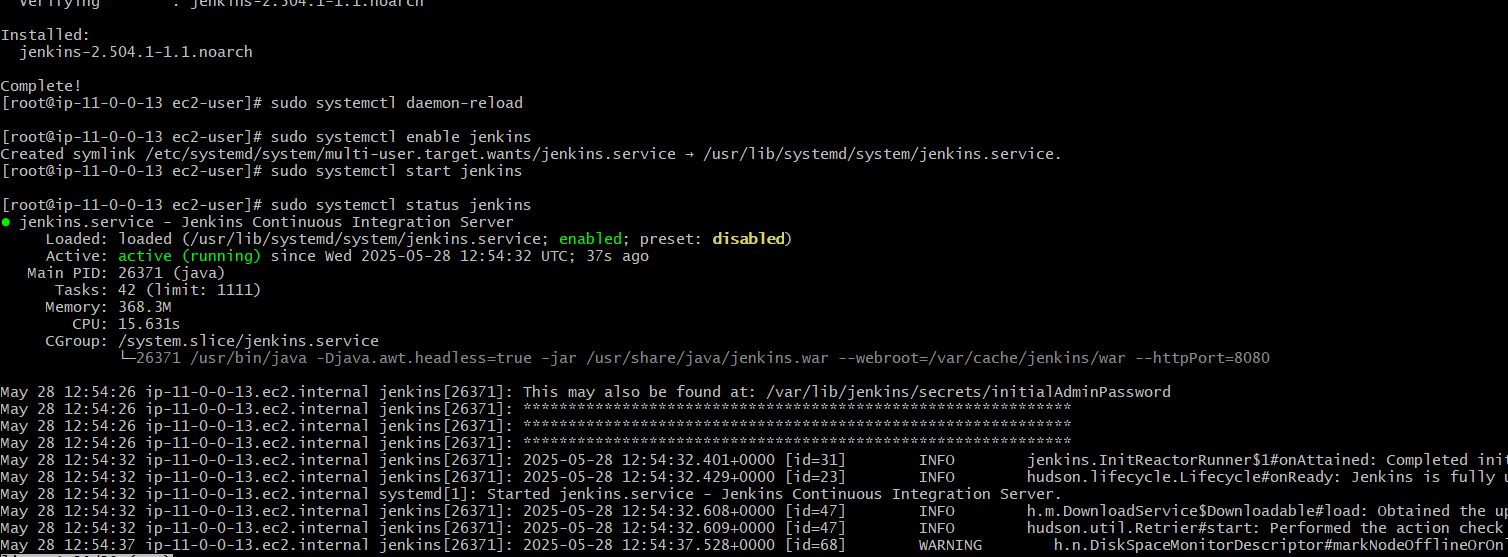
A screenshot of a computer

AI-generated content may be incorrect.

sudo systemctl enable Jenkins

sudo systemctl start Jenkins

sudo systemctl status Jenkins



http://34.204.44.12:8081/login?from=%2F

A screenshot of a computer

AI-generated content may be incorrect.

1. Secure Jenkins server

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Create users called Devops, Testing in Jenkins with Limited access.

**. Enable Jenkins Security**

Go to **Manage Jenkins > Configure Global Security**

Under **Security Realm**, select:

* + **"Jenkins' own user database"**
  + Check **“Allow users to sign up”**

Under **Authorization**, choose:

* + **"Role-Based Strategy"**

**Install Role-Based Strategy Plugin**

Go to **Manage Jenkins > Plugins > Available**

Search for: Role-based Authorization Strategy

Install it and **Restart Jenkins**

A blue line on a white background

AI-generated content may be incorrect.

**Create Users (Devops, Testing)**

1. Go to **Manage Jenkins > Manage Users**
2. Click **Create User**
3. Create users like:
   * **Username**: Devops, Password, Email, etc.
   * **Username**: Testing, Password, Email, etc.

A screenshot of a computer

AI-generated content may be incorrect.

Go to **Manage Jenkins → Manage and Assign Roles → Manage Roles**

Under **“Role to add”**, create roles like:

* + devops\_role
  + testing\_role

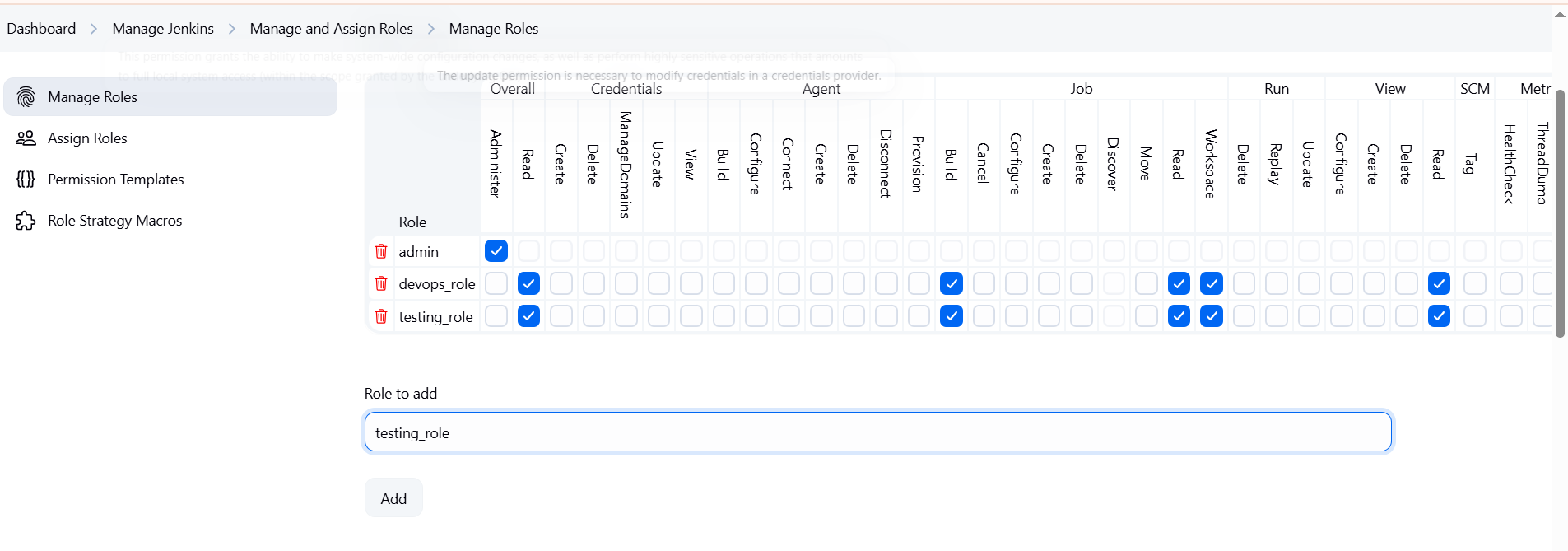
**Define Permissions:**

Example for **limited access** (read-only + job execution):

* Overall → Read
* Job → Read
* Job → Build
* Job → Workspace
* View → Read

Assign stricter or more permissive access depending on what "limited" means in your case.

Click **Save**



Go to **Manage and Assign Roles → Assign Roles**

Under **“User/group”**, enter Devops, assign to devops\_role

Enter Testing, assign to testing\_role

Save

A screenshot of a chat

AI-generated content may be incorrect.

CHECK

* **Devops** and **Testing** users exist
* They have **restricted permissions** based on their roles

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

1. Configure labels and restrict the jobs to execute based on label only.

**Go to Nodes**

* Click on **“Manage Jenkins”** from the left menu
* Click on **“Nodes”**

**Select the Node**

* You’ll see a list like:
  + Built-In Node (master)
  + Any agent you’ve added
* Click on the node where you want to assign a label (e.g., **Built-In Node**)

**Configure the Node**

* Click on **“Configure”** (left side)
* Find the **“Labels”** text box
* DEvops

A screenshot of a computer

AI-generated content may be incorrect.

**Go to Your Job**

* From the **Dashboard**, click on the job
* If you don’t have a job, create one

**Open Job Configuration**

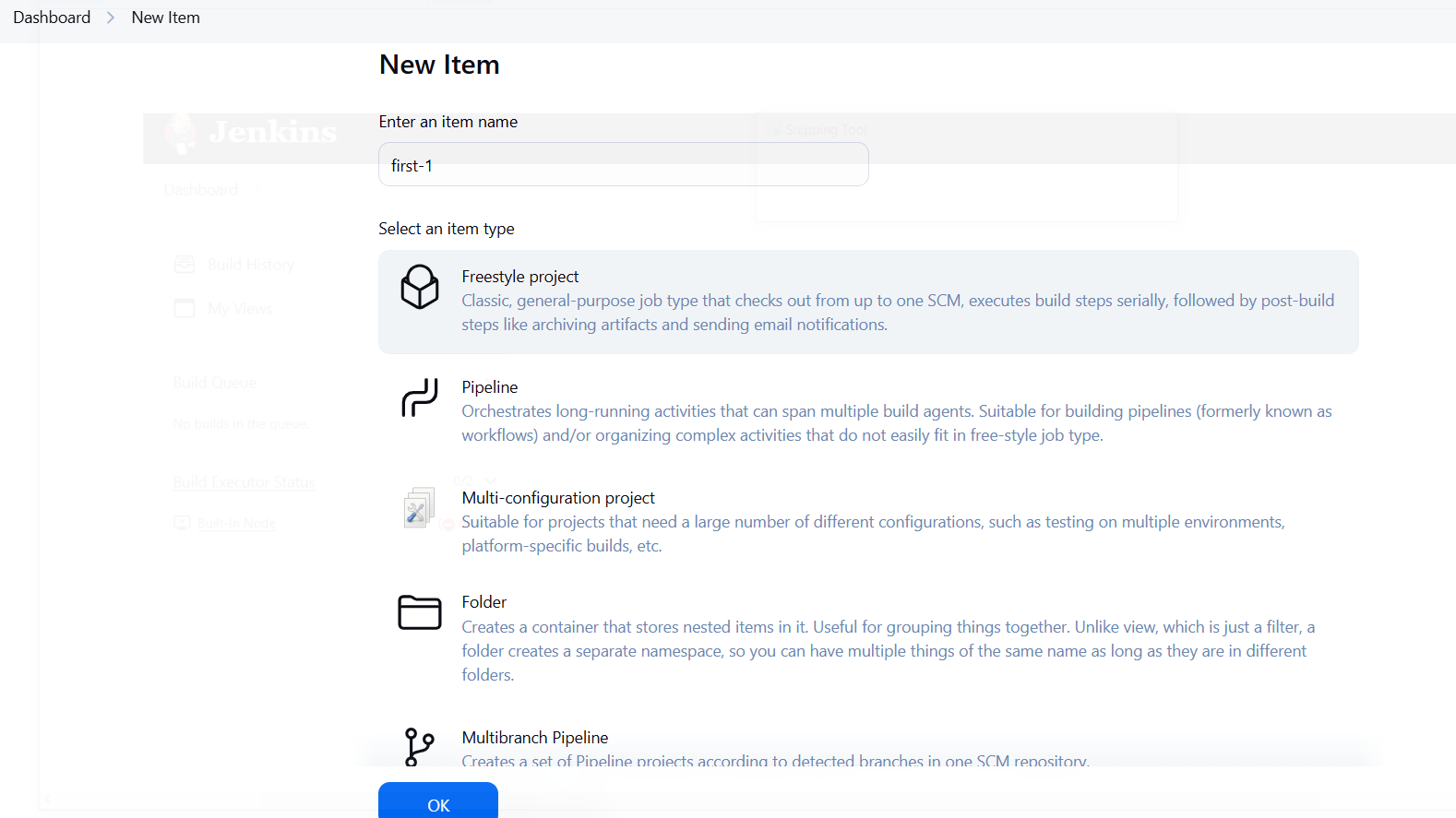
* Click on **“Configure”**

**Set Label Restriction**

* In the **General** section:

LABEL

devops



A screenshot of a computer

AI-generated content may be incorrect.

Save the Job

To test:

* Go to the job
* Click **Build Now**
* If an agent with the devops label is online and idle, the job will run there

A screenshot of a computer

AI-generated content may be incorrect.

1. Create Three sample jobs using the below URL. <https://github.com/betawins/Techie_horizon_Login_app.git>

**Create the First Job**

1. Click **“New Item”**
2. Enter name: second02
3. Select: **Freestyle project**
4. Click **OK**

**: Configure the Job**

**Under General:**

* Leave defaults

**Under Source Code Management:**

1. Select **Git**

<https://github.com/betawins/Techie_horizon_Login_app.git>

**Under Build:**

1. Click **Add build step → Execute shell**

echo "Repo cloned!"

ls -l

**Click Save**  
Then **Build Now**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Clone and Create Two More Jobs

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

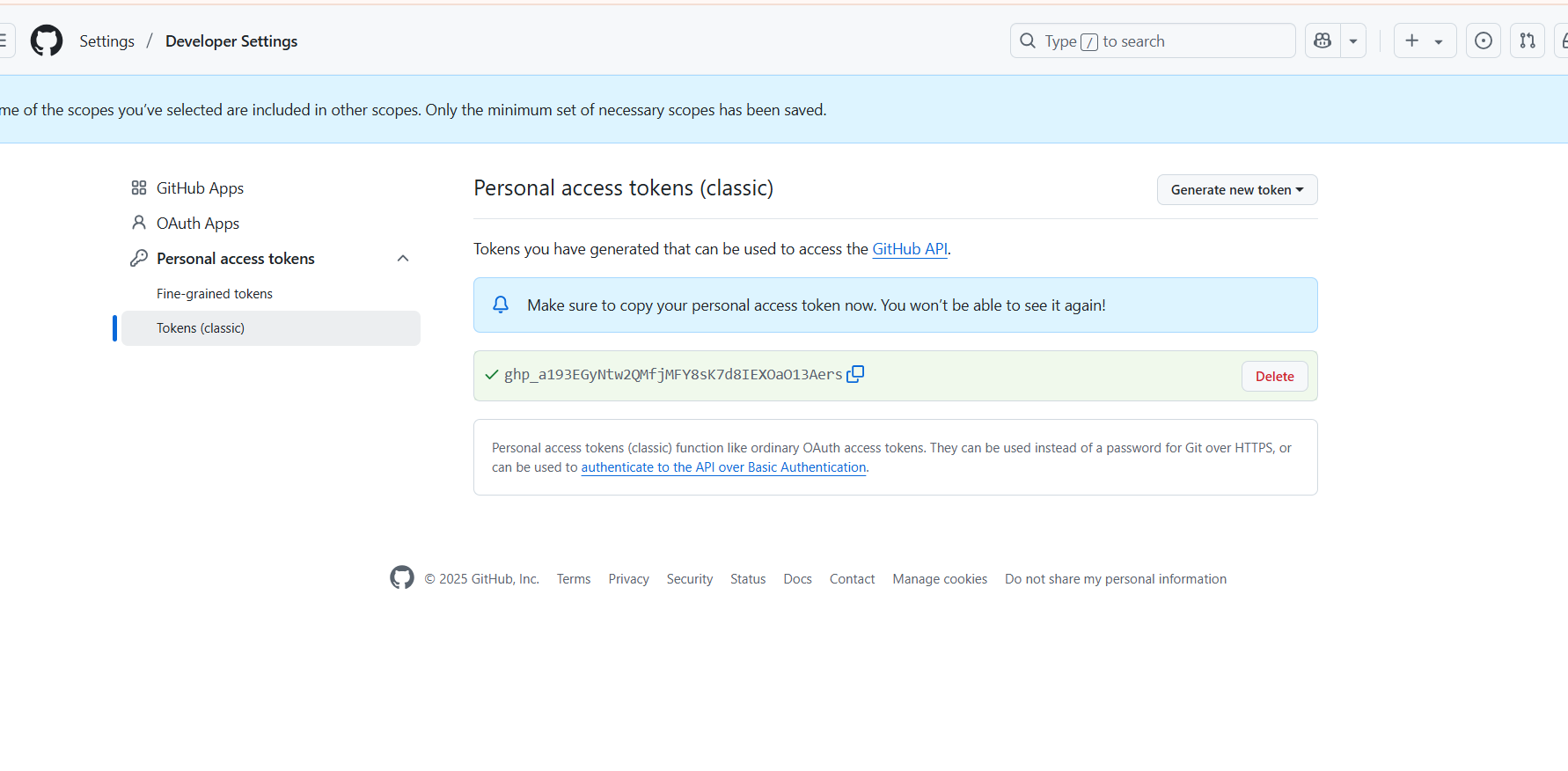
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Create one jenkins job using git hub Private repository.

create a token



**Step 2: Add Credentials to Jenkins**

1. Go to **Jenkins Dashboard → Manage Jenkins → Credentials**
2. Click:
   * (Global) → **Add Credentials**
3. Select:
   * **Kind:** Username with password
   * **Username:** your-github-username
   * **Password:** *(paste the PAT token here)*
   * **ID:** github-token
4. Click **OK**

A screenshot of a computer

AI-generated content may be incorrect.

**Step 3: Create a New Job**

1. Go to Jenkins Dashboard → **New Item**
2. Name: PrivateRepo\_Job
3. Choose: **Freestyle project**
4. Click **OK**

A screenshot of a computer

AI-generated content may be incorrect.

**Step 4: Configure Git Repo**

**Under Source Code Management:**

1. Select **Git**
2. In **Repository URL**, paste:

A screenshot of a chat

AI-generated content may be incorrect.

**step 5: Save and Build**

1. Click **Save**
2. Click **Build Now**

A screenshot of a computer

AI-generated content may be incorrect.