# Create a job in Jenkins

1. **Launch an EC2 Instance**

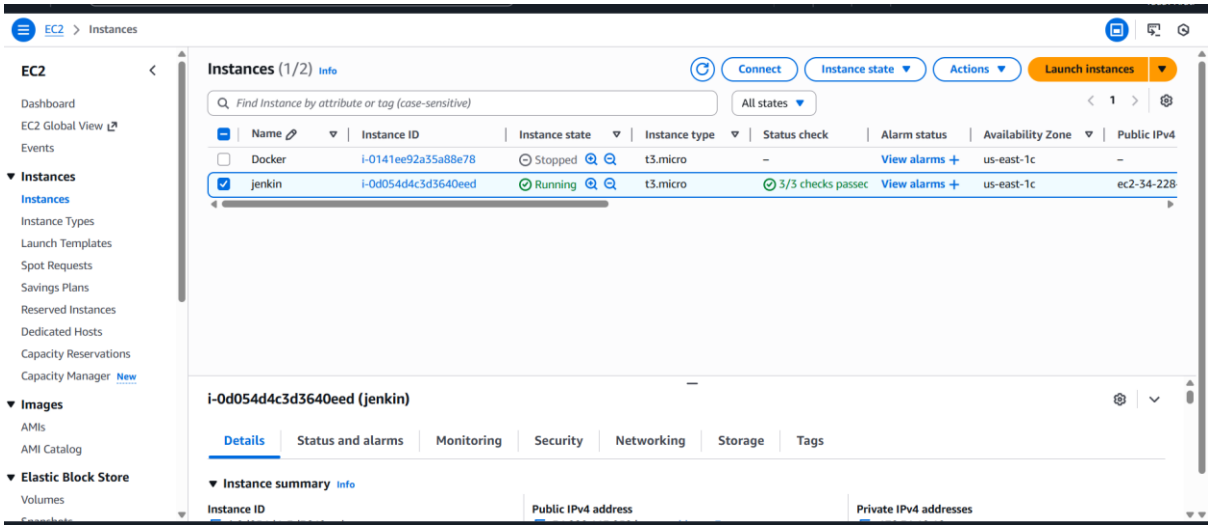
   **Security Group:**

   - Allow SSH (22) from your IP
   - Allow Custom TCP (8080) from your IP (for Jenkins)

   Launch the instance

2. **Connect to the EC2 Instance**

   ssh -i your-key.pem ec2-user@<EC2-PUBLIC-IP>




**3. Install Java (Required for Jenkins)**

sudo yum update -y

sudo yum install java-17-amazon-corretto -y

Verify: java -version

**4. Install Jenkins on EC2**

sudo yum install jenkins -y

sudo systemctl start jenkins

sudo systemctl enable jenkins

sudo systemctl status jenkins

**5. Access Jenkins UI**

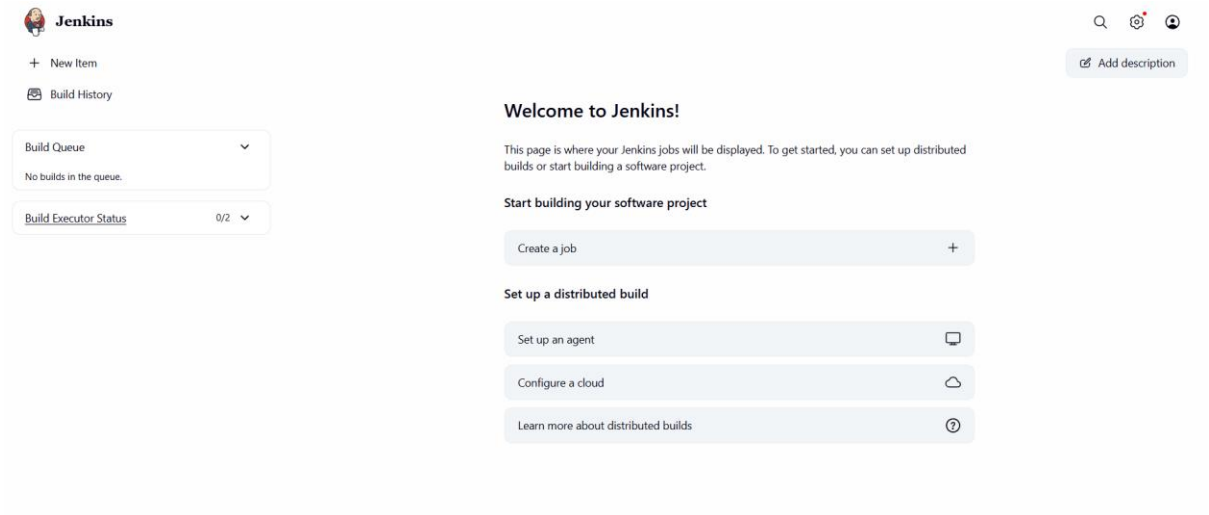
Open browser : http://<EC2-PUBLIC-IP>:8080

Get admin password : sudo cat /var/lib/jenkins/secrets/initialAdminPassword

Install Suggested Plugins
Create Admin User

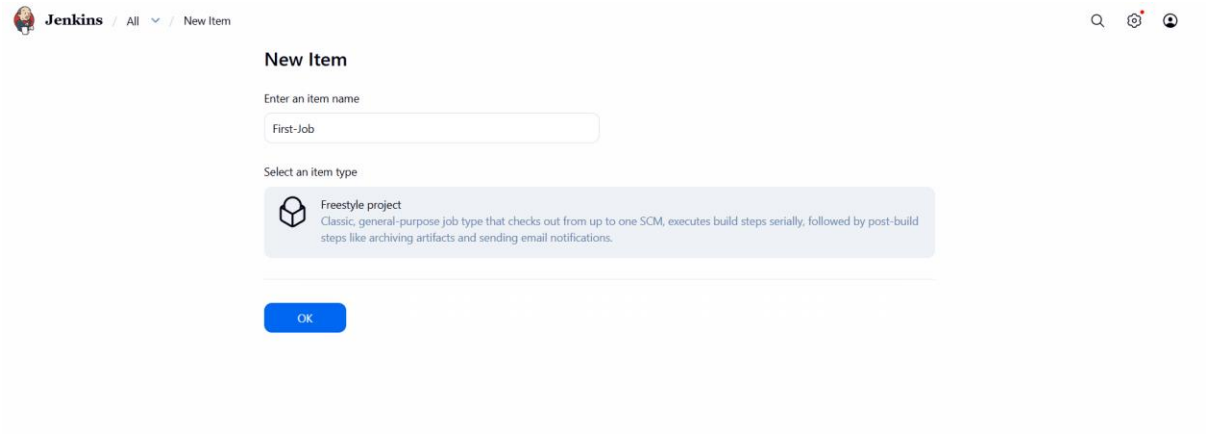
**Note : If already installed then just sign in to Jenkins.**


## 6. Create a Jenkins Job (on EC2)

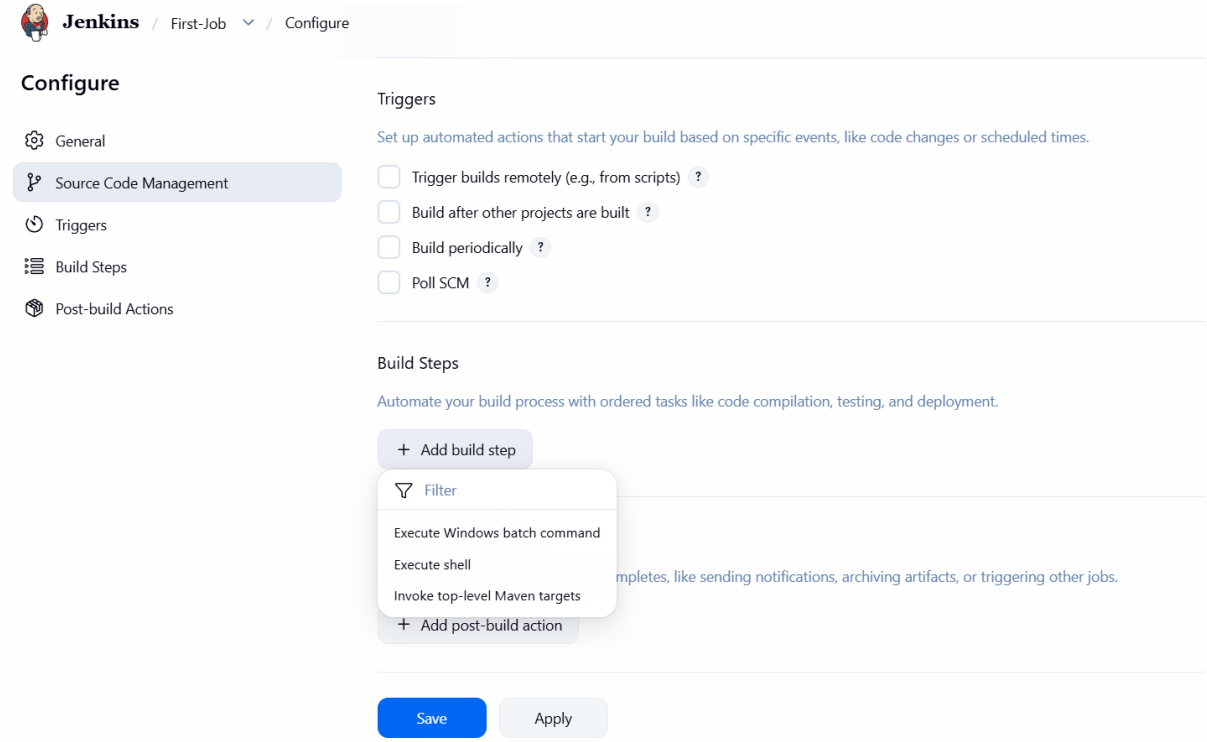



### New Job

- Click New Item
- Enter Job Name
- Select Freestyle Project
- Click OK

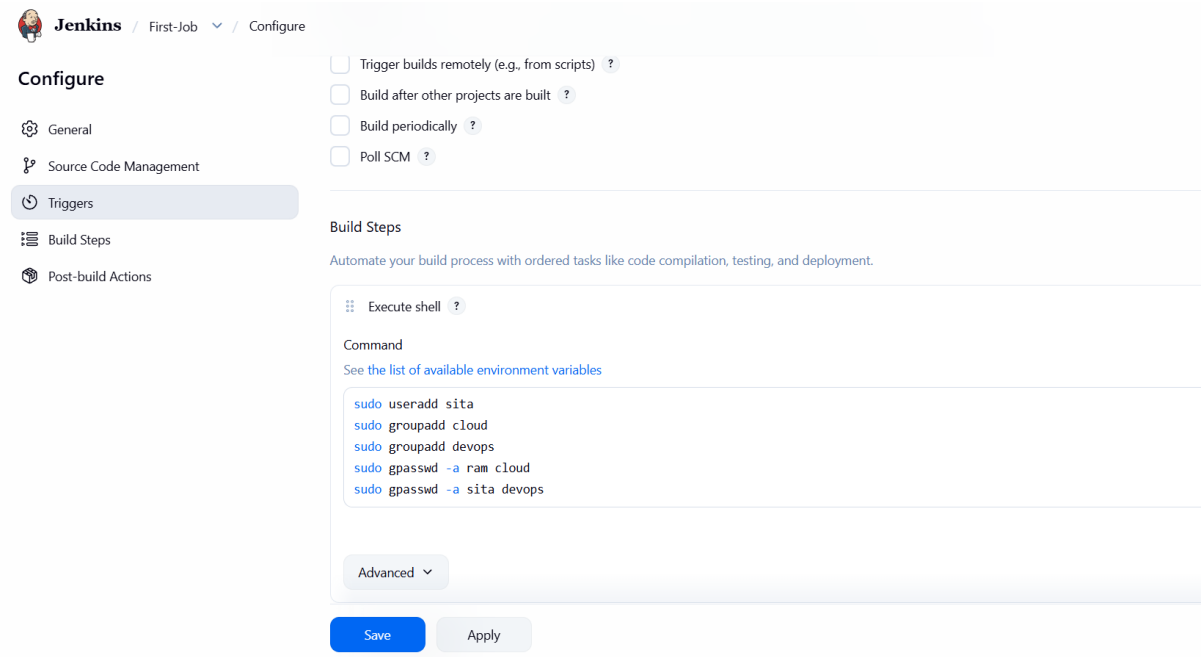


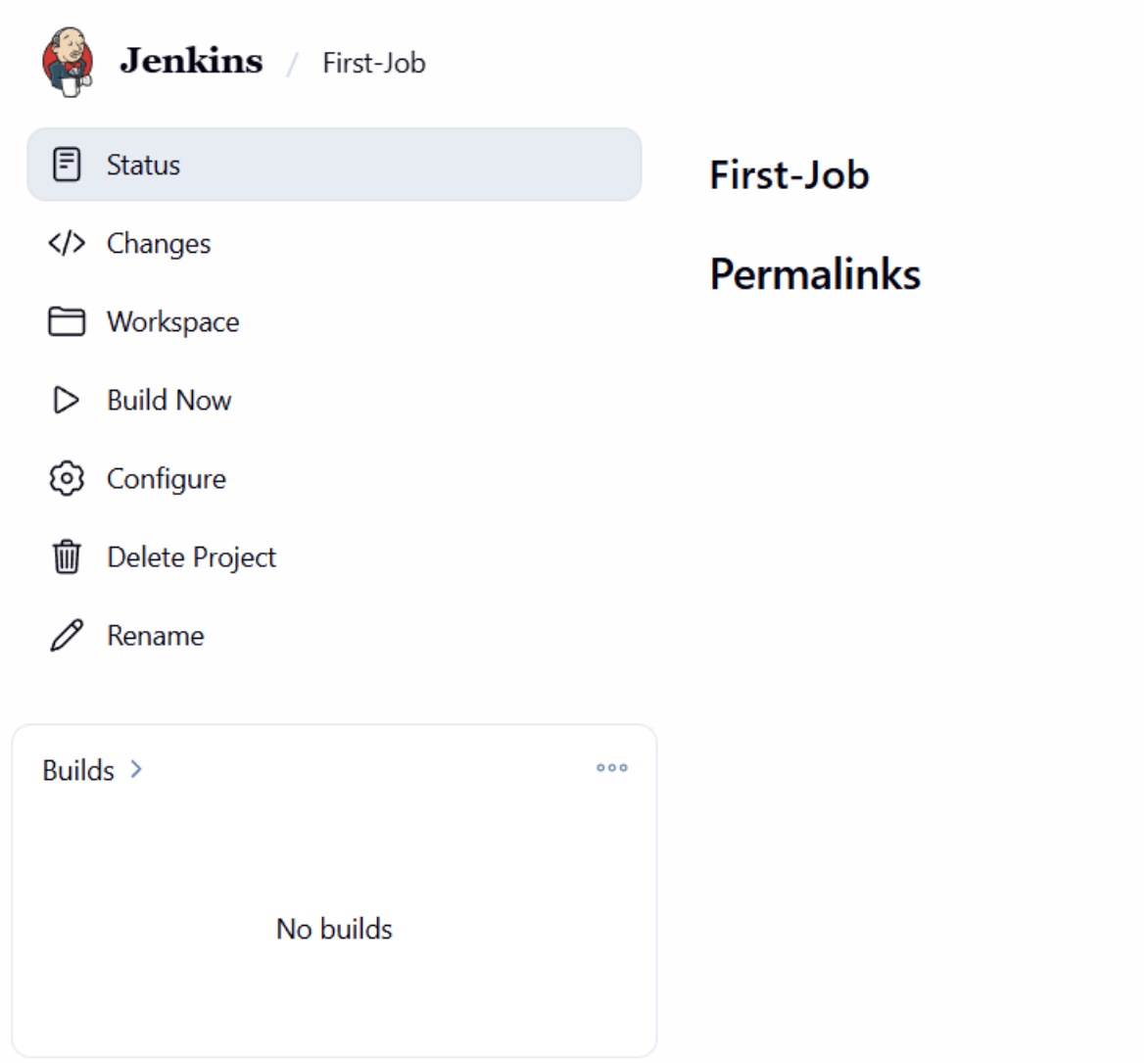

Build Step
Add build step → Execute shell




**Que. First_job**

1) **create 2 users as ram and sita**
   sudo useradd ram
   sudo useradd sita
2) **create 2 groups as cloud and devops**
3) **add ram in cloud add sita in devops**


Jenkins / First-Job / Configure
Configure
General
Source Code Management
Triggers
Build Steps
Post-build Actions
Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Build periodically
Poll SCM
Build Steps
Automate your build process with ordered tasks like code compilation, testing, and deployment.
Execute shell
Command
See the list of available environment variables
sudo useradd sita
sudo groupadd cloud
sudo groupadd devops
sudo gpasswd -a ram cloud
sudo gpasswd -a sita devops
Advanced
Save
Apply


Jenkins / First-Job
Status
Changes
Workspace
Build Now
Configure
Delete Project
Rename
First-Job
Permalinks
Builds
No builds

**7.Switch to Root or a Sudo User**

SSH into your EC2 instance:

Sudo -i

vim /etc/sudoers

**Grant Jenkins Sudo Access**

jenkins ALL=(ALL) NOPASSWD: ALL

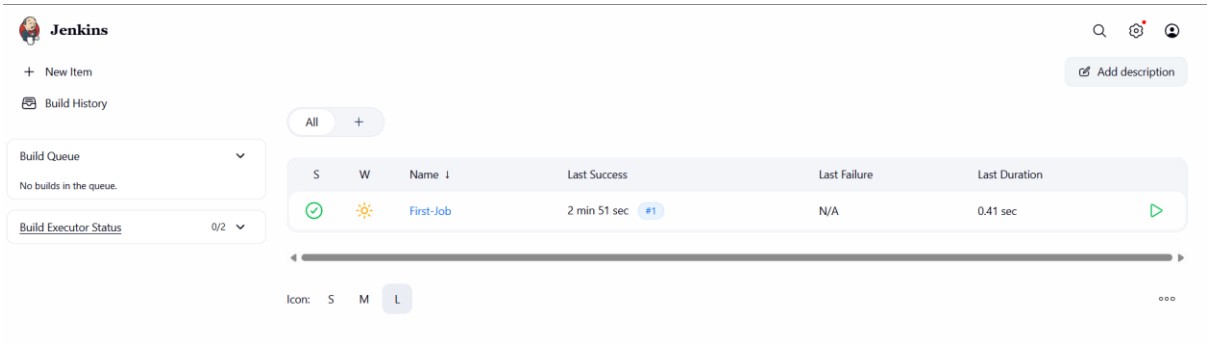
:wq!

**Note: Jenkins can run sudo without password**

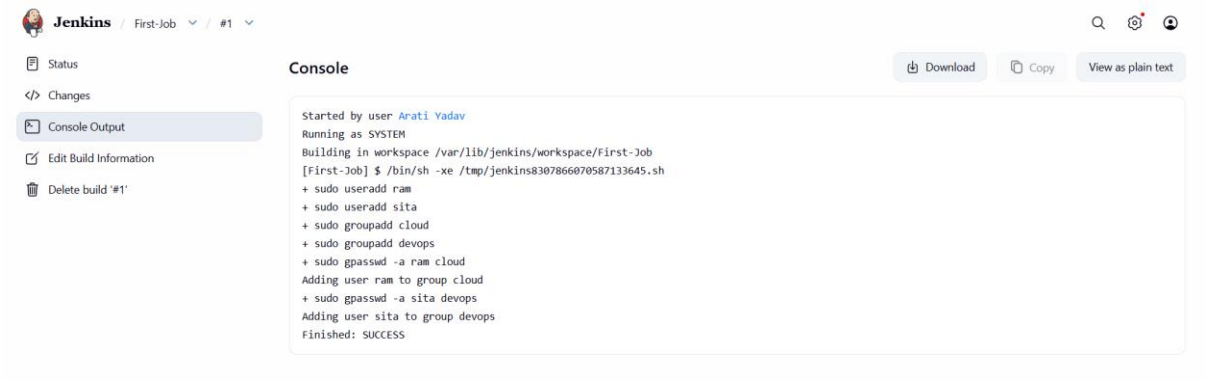
**Required for automated builds**

```
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root ALL=(ALL:ALL) ALL
jenkins ALL=(ALL:ALL) NOPASSWD:ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
:wq!
```

**8. A Jenkins job named First-Job**



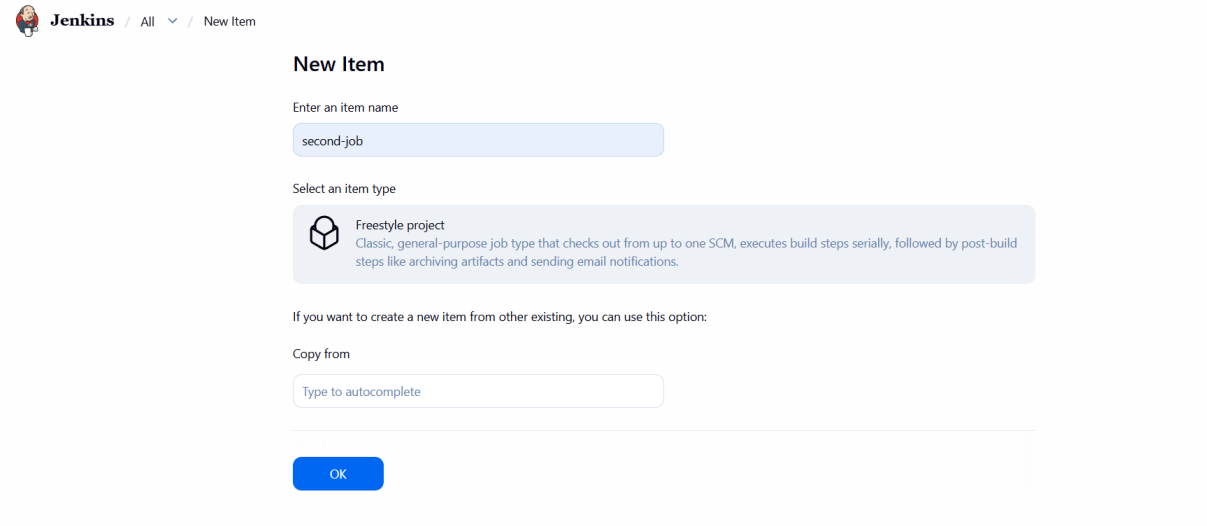

## 9. Build run successfully

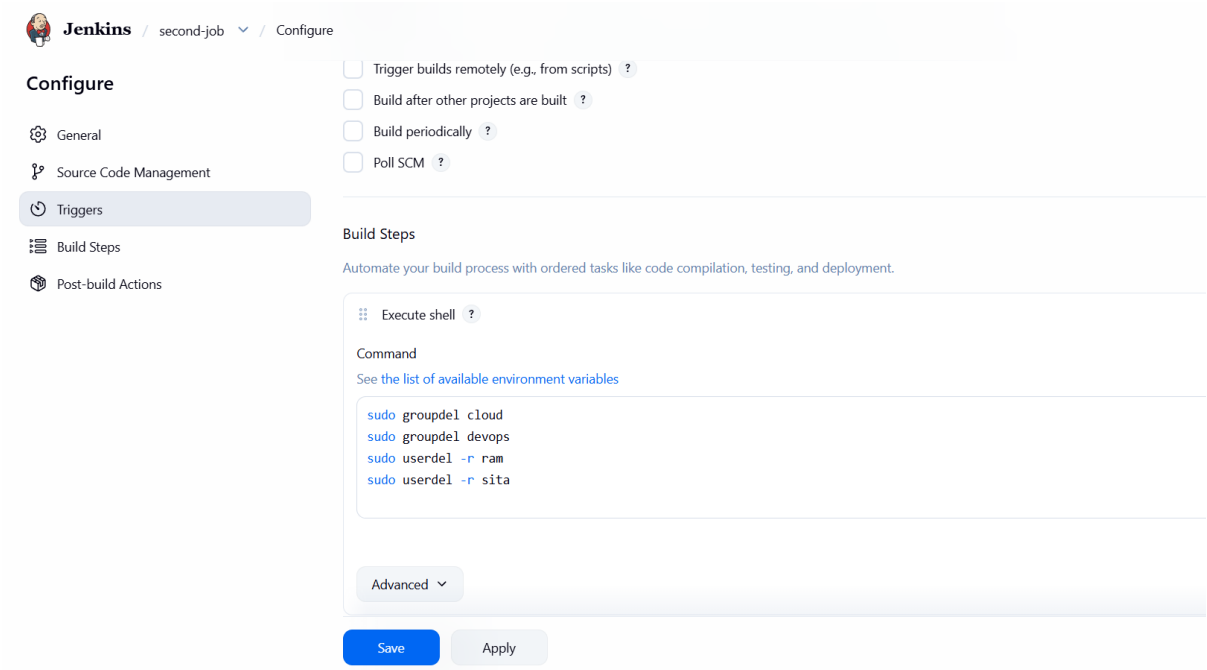



1) Remove users from group ram and sita

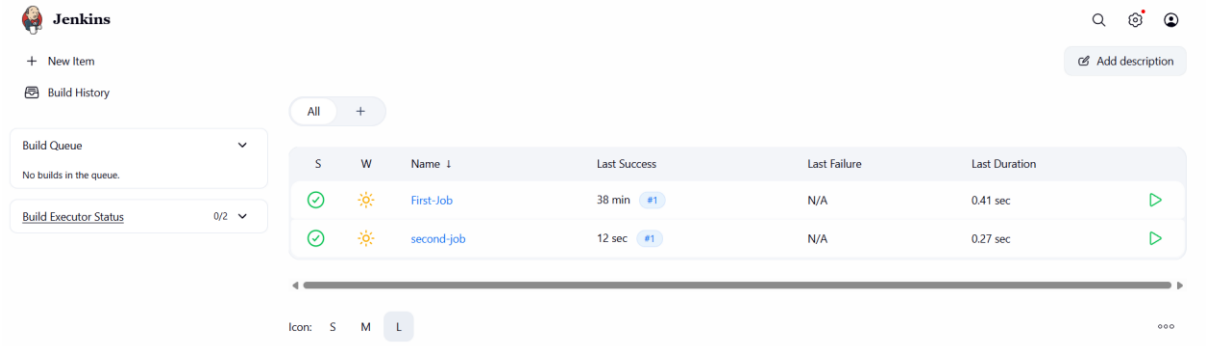
2) delete group cloud and devops

3) delete users



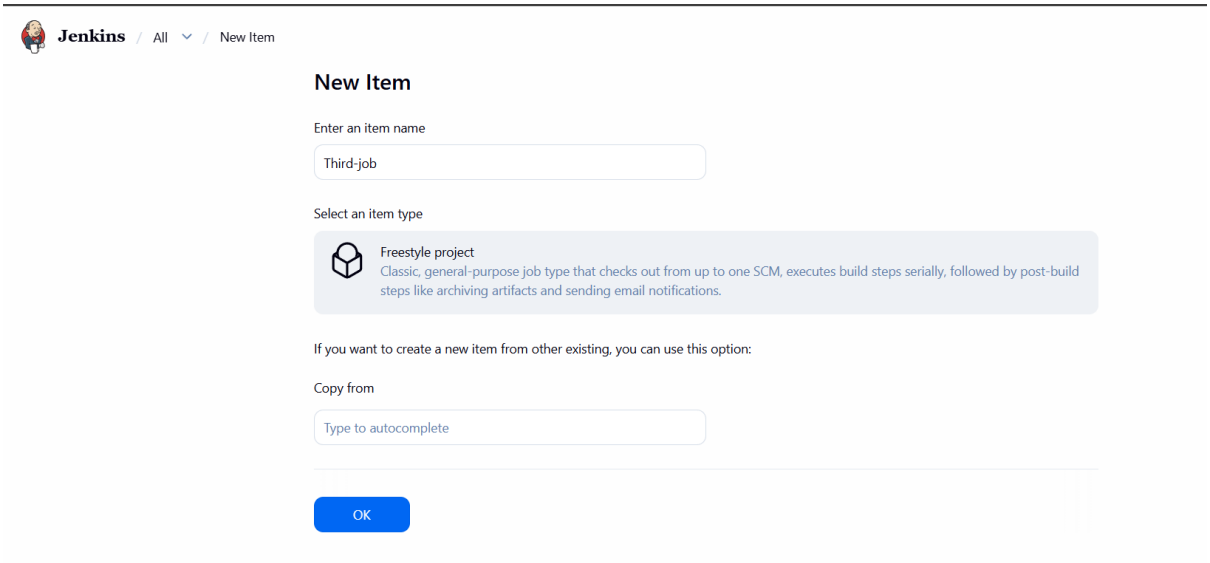



# A Jenkins job named Second-Job




# Build run successfully

## A Jenkins job named Third-Job




**1) create directory named as Project**

sudo mkdir -p project

**2) Create user named as Robert**

sudo useradd Robert

**3) make user as a owner of directory**

sudo chown Robert:Robert project

**4) give the rw permission to user**

sudo chmod u+rw project

**Build ran successfully**