**Assignment: Create a war file on Jenkins master and upload on S3. Pull the file from S3 on Jenkins slave and deploy in tomcat.**

Step 1: Setup Jenkins master. Start tomcat. Login into Jenkins.

Step 2: Create Jenkins slave.

Launch an EC2 instance and install java. Install tomcat. Create a directory in /mnt

cd /mnt

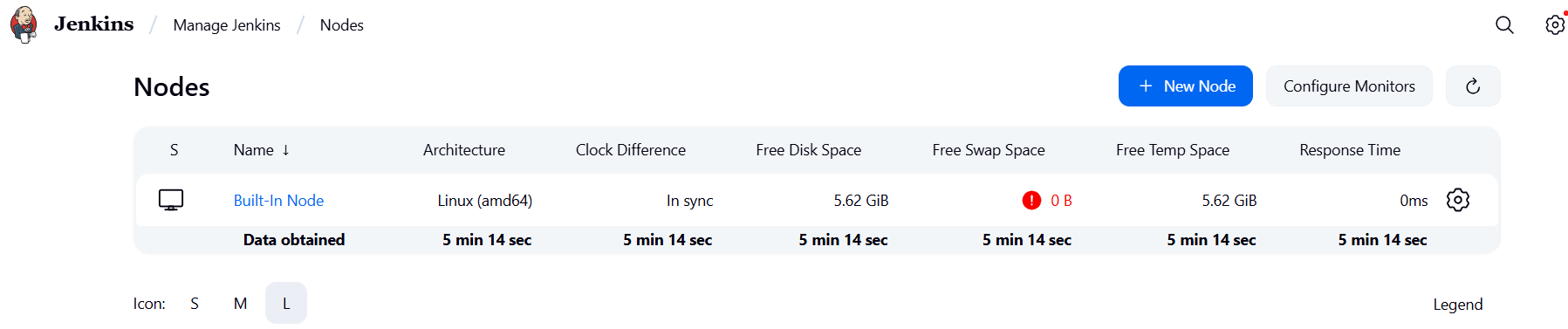
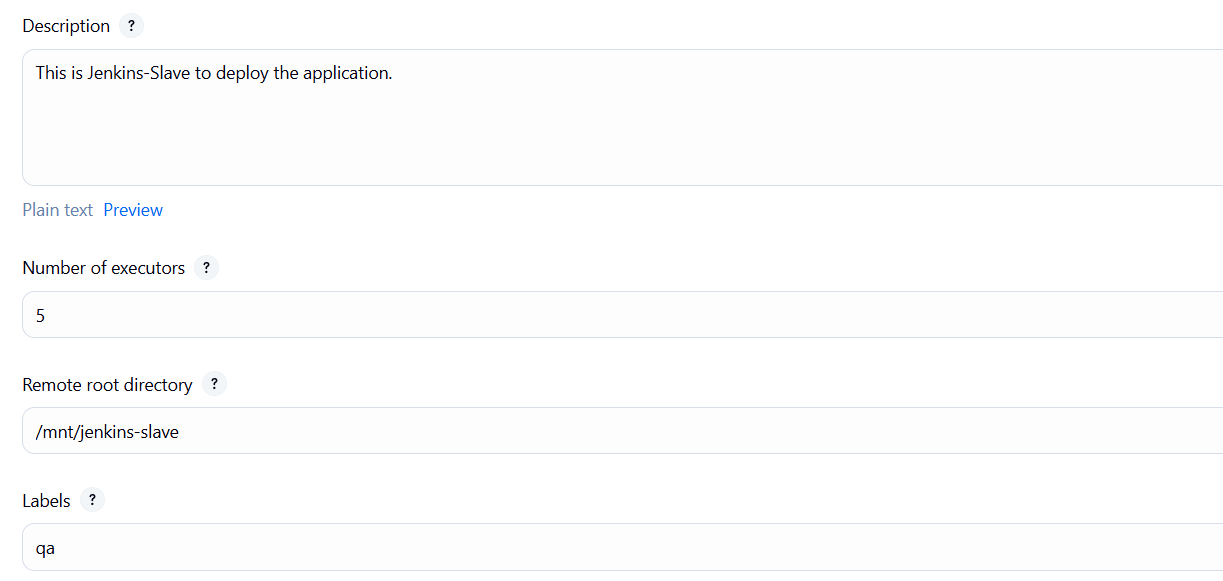
mkdir jenkins-slave

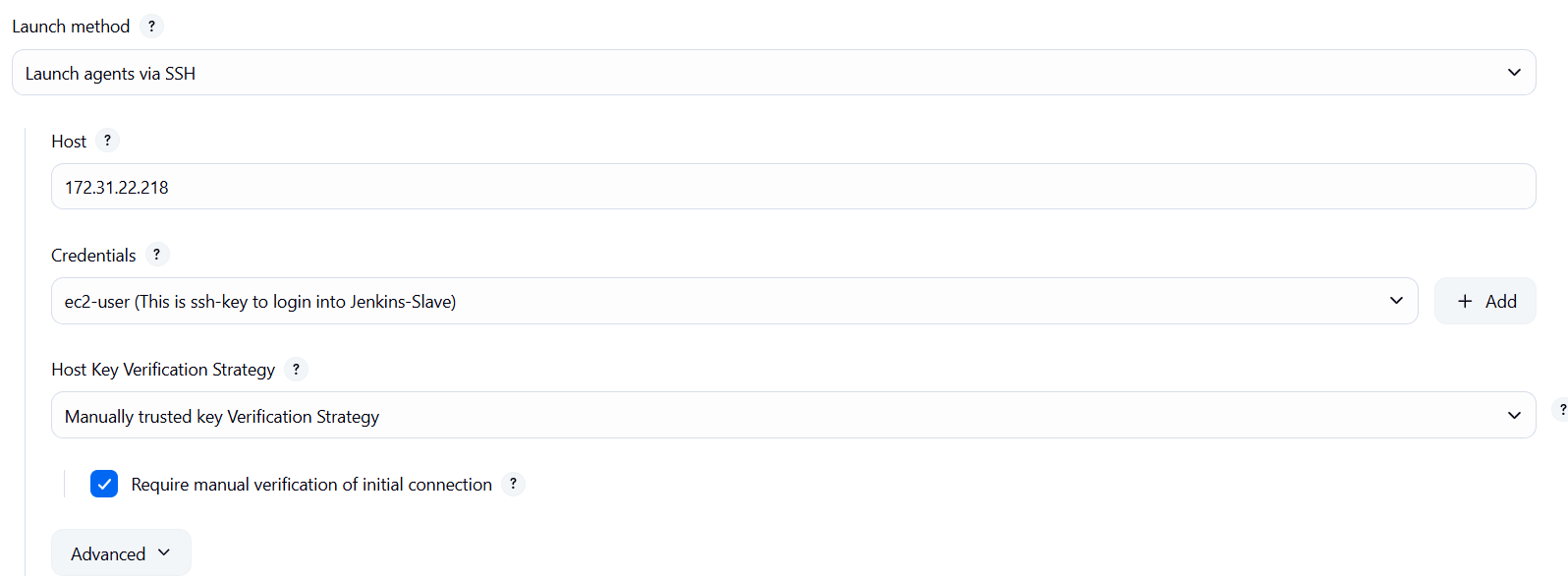
cd jenkins-slave/

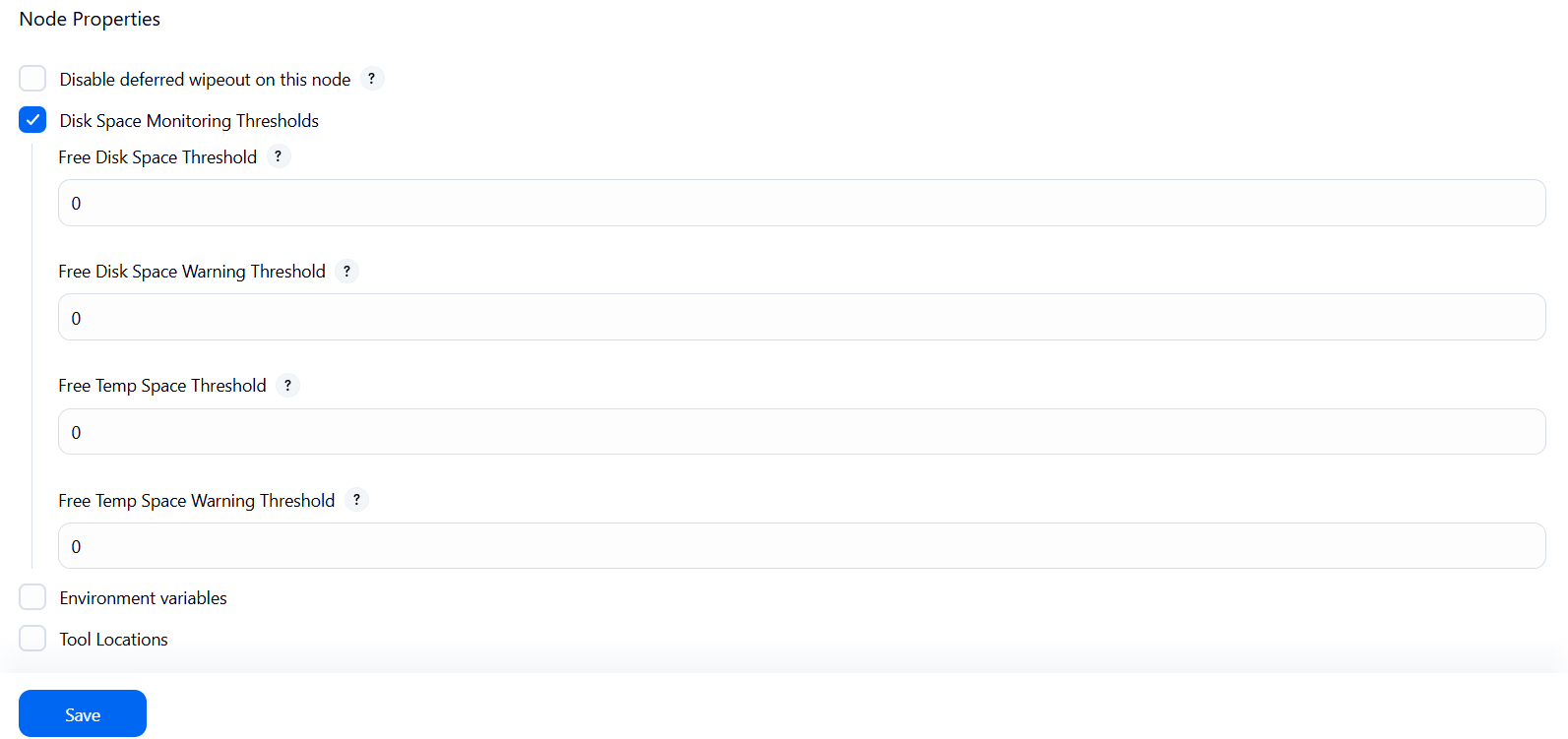
pwd

go to Manage Jenkins → Nodes → New Node → give name → select permanent agent → Create

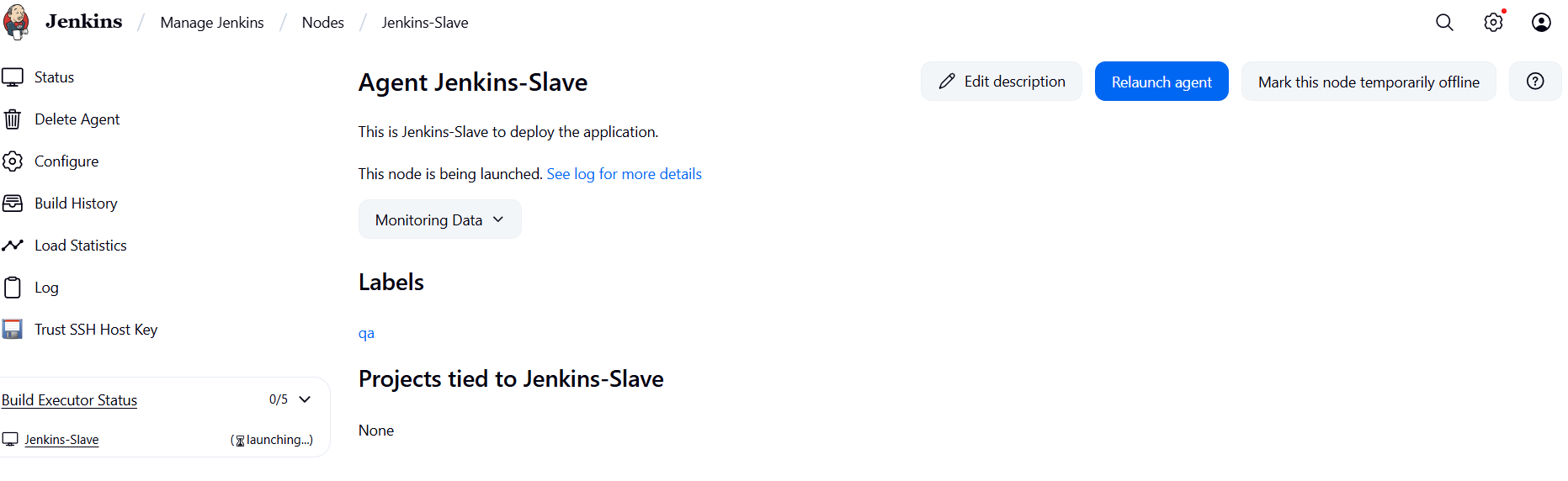
give description → in remote root directory, give the path of folder that you created on slave ec2 instance (/mnt/jenkins-slave) → labels = qa → launch method, select Launch agents via SSH → in host, give the private IP of slave ec2 instance → add credentials (SSH Username with private key) → once the credential is created, select it → in host key verification strategy, select manually trusted key verification strategy → in node properties, select disk space monitoring threshold and set all values to 0 → Save





Once the node is created, got to Trust SSH Host Key → Yes



Go to slave machine and give permission to /mnt directory

chmod -R 777 /mnt

Step 3: Clone project git repository to Jenkins-Master. Build the project.

cd /root/project

ls -ltr

mvn install

.war file gets created.

Step 4: Create S3 bucket.

Create access key and secret access key for your S3 bucket.

aws –version

aws configure

upload the war file in S3 bucket

cd /root

cd project

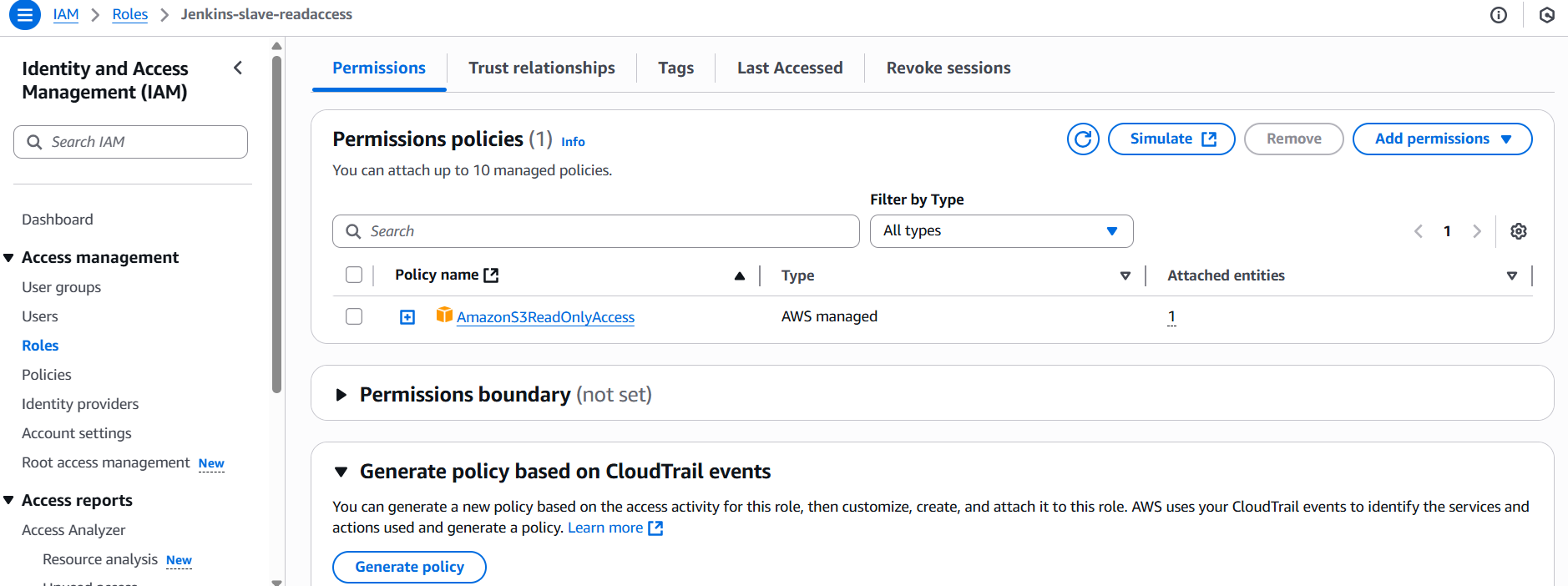
cd target

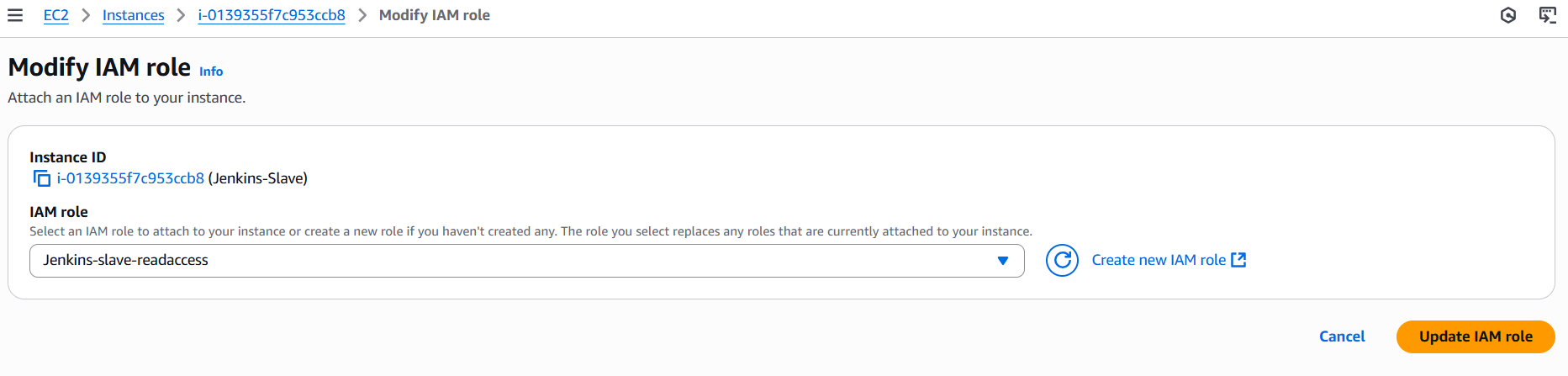
aws s3 cp LoginWebApp.war s3://applicationdeployment-bucket

Step 5: Create IAM role for S3 read access and attach it to Jenkins-Slave EC2 instance.

Go to IAM → roles → create role → AWS service → in service, select ec2 → Next → select AmazonS3ReadOnlyAccess policy → Next → give name → create role

Go to Jenkins-Salve EC2 instance → Actions → Security → Modify IAM role→ select the role → update IAM role



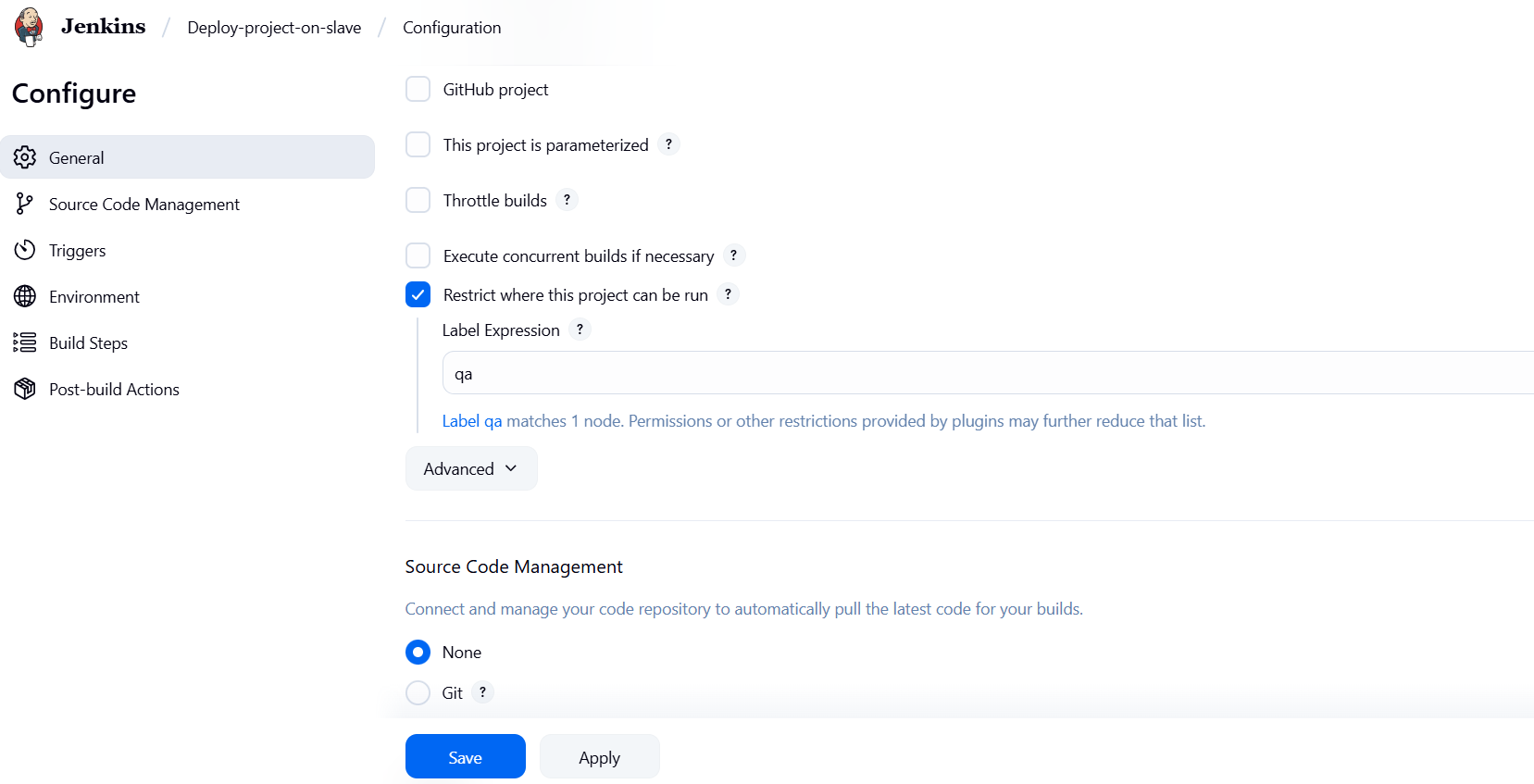


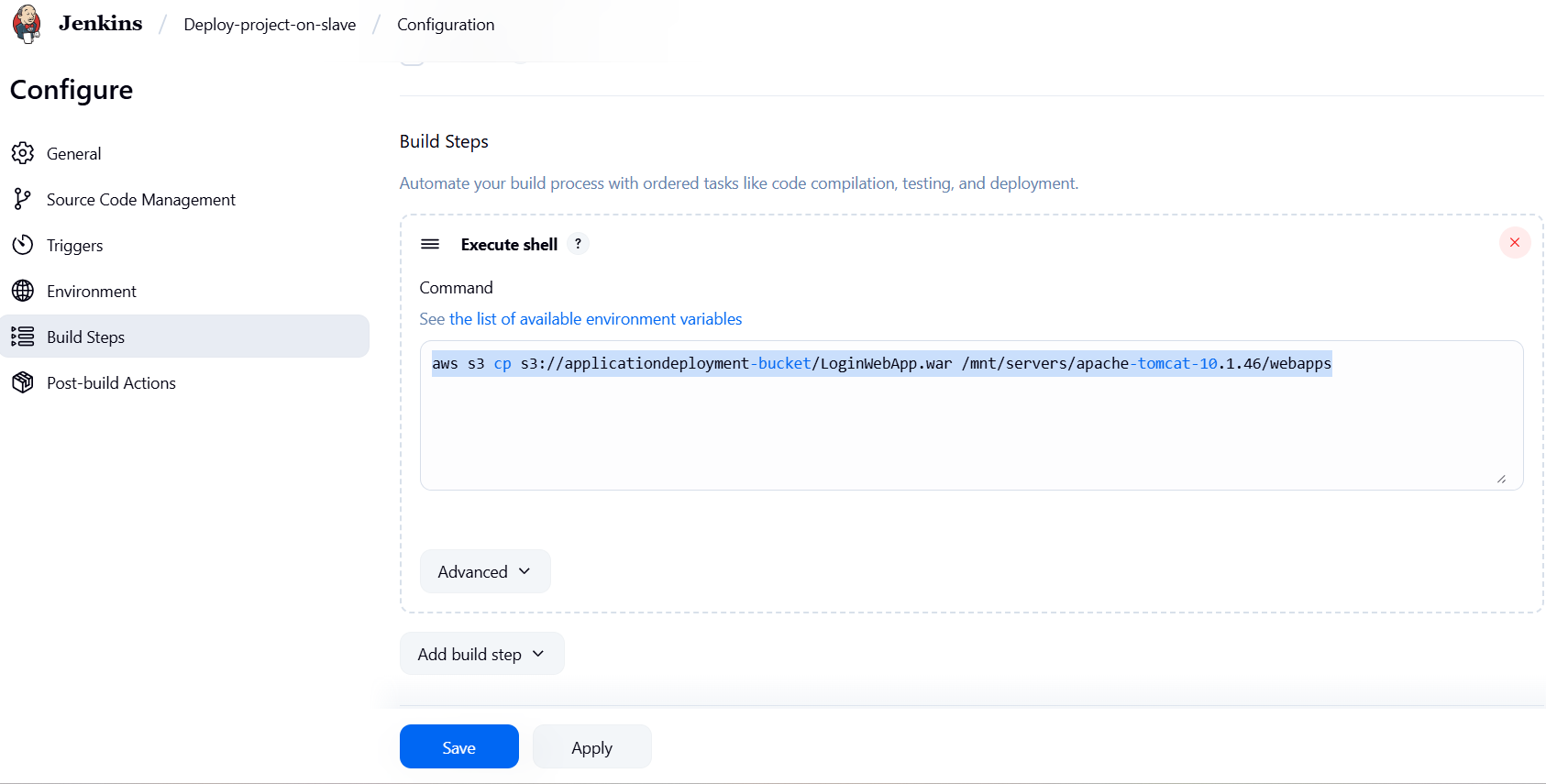
Step 6: Create a job. The job will pull the war file on Jenkins-Slave from S3 bucket and deploy on tomcat.

New item → give name → Freestyle job → OK

Give description → select Restrict where this project can be run → give the slave node label (qa) → build steps → execute shell → write commands → Apply → Save

aws s3 cp s3://applicationdeployment-bucket/LoginWebApp.war /mnt/servers/apache-tomcat-10.1.46/webapps





Step 7: Build the job. Access the public IP of Jenkins-Slave

