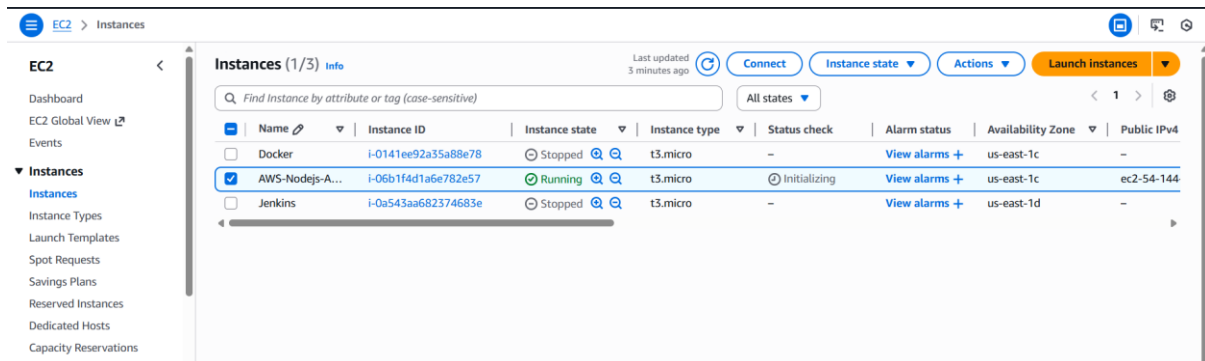


AWS-Nodejs-Application

STEP 1 — Launch EC2 for Application



Install Git

```
sudo apt install git -y
```

```
git --version
```

```
ubuntu@ip-172-31-26-4:~$ git --version
git version 2.43.0
ubuntu@ip-172-31-26-4:~$
```

Install npm

```
Sudo curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
```

Install Node.js

```
sudo apt install nodejs -y
```

```
node -v
```

```
ubuntu@ip-172-31-26-4:~$ node -v
v18.19.1
ubuntu@ip-172-31-26-4:~$
```

```
sudo npm install -g pm2
```

```
ubuntu@ip-172-31-81-149:~$ sudo npm install -g pm2
added 133 packages in 8s
13 packages are looking for funding
  run `npm fund` for details
npm notice
npm notice New major version of npm available! 10.8.2 -> 11.10.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.10.0
npm notice To update run: npm install -g npm@11.10.0
npm notice
ubuntu@ip-172-31-81-149:~$
```

```
sudo apt install git
```

Verify :

```
node -v
```

```
npm -v
```

```
pm2 -v
```

STEP 2 — Install Jenkins Server

Install Java

```
sudo apt update
```

```
sudo apt install fontconfig openjdk-21-jre
```

```
java -version
```

Install Jenkins

```
sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
```

```
https://pkg.jenkins.io/debian-stable/jenkins.io-2026.key
```

```
echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc]" \
```

```
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
```

```
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
sudo apt update
```

```
sudo apt install Jenkins
```

```
sudo systemctl start jenkins
```

```
sudo systemctl enable Jenkins
```

```
sudo systemctl status jenkins
```

```
ubuntu@ip-172-31-26-4:~$ sudo systemctl start jenkins
sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
ubuntu@ip-172-31-26-4:~$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
ubuntu@ip-172-31-26-4:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Wed 2026-02-18 11:26:52 UTC; 11min ago
     Main PID: 3981 (java)
       Tasks: 37 (limit: 1008)
      Memory: 299.4M (peak: 306.2M)
         CPU: 20.792s
        CGroup: /system.slice/jenkins.service
                └─3981 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache
```

Open in browser:

Public ip - 3.86.138.82:8080

Unlock using: `sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

.....

Continue

Getting Started

Organization and Administration

Build Features

Build Tools

Build Analysis and Reporting

Pipelines and Continuous Delivery

Source Code Management

Distributed Builds

User Management and Security

Notifications and Publishing

Appearance

Languages

All | None | Suggested

Selected (0/52)

Note that the full list of plugins is not shown here. Additional plugins can be installed in the **Plugin Manager** once the initial setup is complete. [See the documentation for more information.](#)

Organization and Administration (0/4)

☐ Dashboard View

Customizable dashboard that can present various views of job information.

28 >

☐ Folders

This plugin allows users to create "folders" to organize jobs. Users can define custom taxonomies (like by project type, organization type etc). Folders are nestable and you can define views within folders. Maintained by CloudBees, Inc.

2 >

☐ Configuration as Code

This plugin allows configuration of Jenkins based on human-readable declarative configuration files.

17 >

☐ OWASP Markup Formatter

Uses the [OWASP Java HTML Sanitizer](#) to allow safe-seeming HTML markup to be entered in project descriptions and the like.

Build Features (0/9)

☐ Build Name and Description Setter

This plug-in sets the display name and description of a build to something other than #1, #2, #3, ...

31 >

☐ Build Timeout

7 >

Jenkins 2.541.1

Back

Install

Create First Admin User

Username

admin

Password

.....

Confirm password

.....

Full name

Arati Yadav

Jenkins 2.541.1

[Skip and continue as admin](#)

[Save and Continue](#)

Instance Configuration

Jenkins URL:

http://54.144.80.115:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.541.1

[Not now](#)

[Save and Finish](#)

Jenkins is ready!

Your Jenkins setup is complete.

Start using Jenkins


STEP 3 — Install Required Jenkins Plugins

Inside Jenkins → Manage Plugins:

Install:

- Git Plugin
- Pipeline Plugin
- SSH Agent Plugin
- NodeJS Plugin (optional)



 Jenkins is restarting

Your browser will reload automatically when Jenkins is ready.

Safe Restart
Builds on agents can usually continue.

STEP 4 — Setup SSH Access from Jenkins to App EC2

In Jenkins:

1. Go to **Manage Jenkins** → **Credentials**
2. Add Credentials:
 - Kind: SSH Username with private key
 - Username: ubuntu
 - Private Key
 - ID: ec2-key

Add Credentials



Select a type of credential



Username with password

Commonly used for authentication to services like Git, APIs, or registries.



SSH Username with private key



Secret file



Secret text



Certificate

Upload a PKCS#12 or PEM encoded certificate and private key.

Next

<

Add SSH Username with private key

×

ubuntu

☐

Treat username as secret

?

Private Key

☒ Enter directly

Key

Enter New Secret Below

7Lr1A00yL
UZgihVCZ7gmkemP2Dop5zGKqumd+kuRvQE3oPTyx602v1vyFvqCLkih
hwctmeZdv
4jJ7kENeYntkg630hVgJJibpJ60w0u8ihAPChiDX+X96I802E2UTcBC
/2CdiPl1Q
8bw7/wKBgGXN1Va0wMgvqPsEufdG8YBDCPFBifpwN3oNFfhZvTEojyv
AnHgQTksR
Qj/tVZYRnvY58ChnCyxRNOb6l4HT5U19XLh1IMmcfkDAnEa/fHZPry8
ySAoHPyEq
IU+9vs69dTTNAJeHPNfB+Dw89maYwQQw1C14UCFEC04sDP/Gd6wz
-----END RSA PRIVATE KEY-----

Passphrase

Create

STEP 5 — Create Jenkins Pipeline job

New Item

Enter an item name

Nodejs-pipeline

Select an item type



Pipeline

Build, test, and deploy using pipelines. Supports stages, parallel work, and running on multiple agents.



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.



Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

OK

```
pipeline {
```

```
    agent any
```

```
    environment {
```

```
        EC2_HOST = "3.86.138.82"
```

```
        EC2_USER = "ubuntu"
```

```
        APP_DIR = "/home/ubuntu/nodeapp"
```

```
    }
```

```
    stages {
```

```
        stage('Clone Code') {
```

```
            steps {
```

```
                git branch: 'main',
```

```
                url: 'https://github.com/yourusername/your-repo.git'
```

```
            }
```

```
        }
```

```
        stage('Install Dependencies') {
```

```

    steps {
        sh 'npm install'
    }
}

stage('Build') {
    steps {
        sh 'echo "Build Step (if needed)"'
    }
}

stage('Deploy to EC2') {
    steps {
        sshagent(['ec2-key']) {
            sh """
            ssh -o StrictHostKeyChecking=no $EC2_USER@$EC2_HOST '
                mkdir -p $APP_DIR
            '

            scp -r * $EC2_USER@$EC2_HOST:$APP_DIR

            ssh $EC2_USER@$EC2_HOST '
                cd $APP_DIR
                npm install
                pm2 delete nodeapp || true
                pm2 start app.js --name nodeapp
                pm2 save
            '
            """
        }
    }
}

```

```
}  
  
}  
  
}  
  
}
```

Step 6: Configure GitHub Webhook (Auto Deployment)

In GitHub:

1. Repo → Settings → Webhooks
2. Add webhook:
3. [http:// 3.86.138.82:8080/github-webhook/](http://3.86.138.82:8080/github-webhook/)
4. Choose: Push events

Output:

Job Successfully Build.

The screenshot shows the Jenkins web interface for a pipeline named 'Nodejs-pipeline'. The left sidebar contains navigation options: Status, Changes, Console Output (selected), Edit Build Information, Delete build '#3', Git Build Data, Restart from Stage, Replay, Pipeline Steps, Workspaces, and Previous Build. The main area displays the 'Console' output for build #3. The output shows the pipeline starting with a 'node' step, running on Jenkins in the workspace. It then executes a series of git commands to clone the repository, fetch changes, and checkout the main branch. The output also shows the installation of npm and the execution of 'npm install'.

```
Started by user Arati Yadav  
[Pipeline] Start of Pipeline  
[Pipeline] node  
Running on Jenkins in /var/lib/jenkins/workspace/Nodejs-pipeline  
[Pipeline] {  
[Pipeline] withEnv  
[Pipeline] {  
[Pipeline] stage  
[Pipeline] { (Clone Code)  
[Pipeline] git  
The recommended git tool is: NONE  
No credentials specified  
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Nodejs-pipeline/.git # timeout=10  
Fetching changes from the remote git repository  
> git config remote.origin.url https://github.com/AratiYadav21/aws-nodejs-deploy.git # timeout=10  
Fetching upstream changes from https://github.com/AratiYadav21/aws-nodejs-deploy.git  
> git --version # timeout=10  
> git --version # 'git version 2.43.0'  
> git fetch --tags --force --progress -- https://github.com/AratiYadav21/aws-nodejs-deploy.git +refs/heads/*:refs/remotes/origin/* # timeout=10  
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10  
Checking out Revision d7b4927f68ec17e25081bc044ca927e1fed308da (refs/remotes/origin/main)  
> git config core.sparsecheckout # timeout=10  
> git checkout -f d7b4927f68ec17e25081bc044ca927e1fed308da # timeout=10  
> git branch -a -v --no-abbrev # timeout=10  
  
> git branch -D main # timeout=10  
> git checkout -b main d7b4927f68ec17e25081bc044ca927e1fed308da # timeout=10  
Commit message: "Add files via upload"  
> git rev-list --no-walk d7b4927f68ec17e25081bc044ca927e1fed308da # timeout=10  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (Install Dependencies)  
[Pipeline] sh  
+ npm install  
  
up to date, audited 60 packages in 1s  
  
9 packages are looking for funding  
run 'npm fund' for details  
  
7 vulnerabilities (3 low, 4 high)  
  
To address all issues, run:  
npm audit fix  
  
Run 'npm audit' for details.  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (Build)  
[Pipeline] sh  
+ echo Build Step (if needed)
```

	<pre> Build Step (if needed) [Pipeline] } [Pipeline] // stage [Pipeline] stage [Pipeline] { (Deploy to EC2) [Pipeline] sshagent [ssh-agent] Using credentials ubuntu \$ ssh-agent SSH_AUTH_SOCK=/tmp/ssh-sklqCkkl3v/agent.6554 SSH_AGENT_PID=6557 Running ssh-add (command line suppressed) Identity added: /var/lib/jenkins/workspace/Nodejs-pipeline@tmp/private_key_16323703221816698274.key (/var/lib/jenkins/workspace/Nodejs-pipeline@tmp/private_key_16323703221816698274.key) [ssh-agent] Started. [Pipeline] { [Pipeline] sh + ssh -o StrictHostKeyChecking=no ubuntu@3.86.138.82 mkdir -p /home/ubuntu/nodeapp Warning: Permanently added '3.86.138.82' (ED25519) to the list of known hosts. + ssh ubuntu@3.86.138.82 cd /home/ubuntu/nodeapp npm install pm2 delete nodeapp true pm2 start app.js --name nodeapp pm2 save </pre>
--	--

 Jenkins / Nodejs-pipeline ▾ / #3 ▾ <div>    </div>	<pre> npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.10.0 npm notice To update run: npm install -g npm@11.10.0 npm notice [PM2][ERROR] Process or Namespace nodeapp not found [PM2][ERROR] Script not found: /home/ubuntu/nodeapp/app.js [PM2] Saving current process list... [PM2][WARN] PM2 is not managing any process, skipping save... [PM2][WARN] To force saving use: pm2 save --force [Pipeline] } \$ ssh-agent -k unset SSH_AUTH_SOCK; unset SSH_AGENT_PID; echo Agent pid 6557 killed; [ssh-agent] Stopped. [Pipeline] // sshagent [Pipeline] } [Pipeline] // stage [Pipeline] } [Pipeline] // withEnv [Pipeline] } [Pipeline] // node [Pipeline] End of Pipeline Finished: SUCCESS </pre>
--	--