

Continuous Integration for a Node.js Application using Git, Jenkins, and AWS Elastic Beanstalk

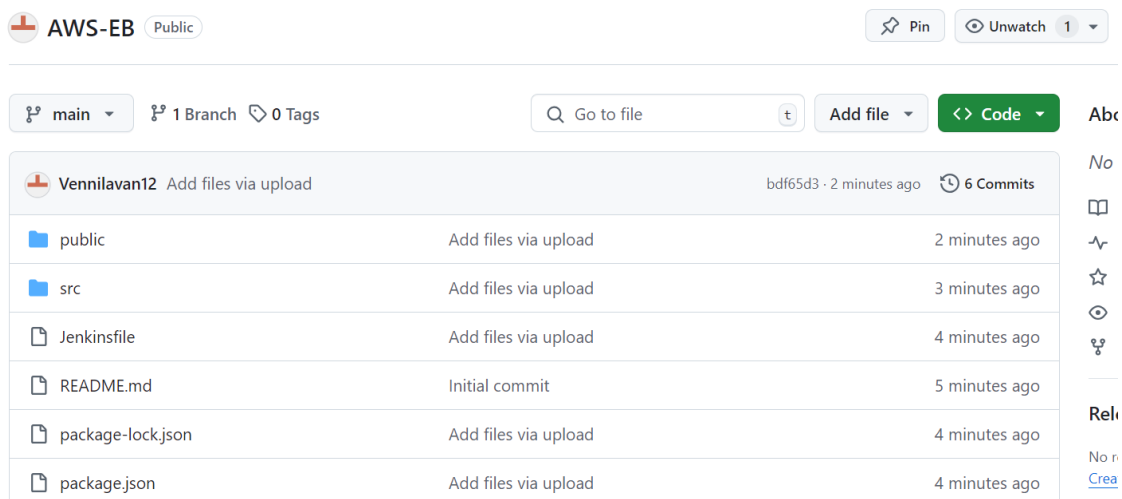
Solution

Project Requirements:

1. EC2
2. Git
3. Jenkins
4. AWS Elastic Beanstalk

STEP 1: Create a sample Node.js application with source code in a Git repository

- Create a new repository into our github account.



- Clone the repository into local using the git clone method.

git clone “Repo url”

- Once the code has been ready we can move into the next step.

STEP 2: Configure Jenkins to clone the Git repository, install dependencies, run tests, and package the application as a ZIP file

Requirements :

1. EC2
2. Jenkins

EC2

- Create an EC2 instance in AWS for installing Jenkins server.
- Login to AWS Management console



Sign in

☐ **Root user**
Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☒ **IAM user**
User within an account that performs daily tasks. [Learn more](#)

Account ID (12 digits) or account alias

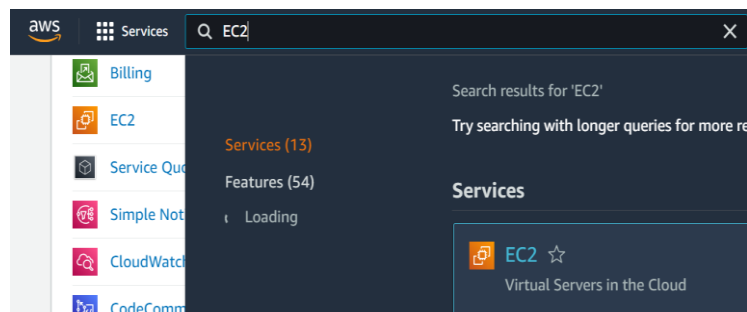
Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

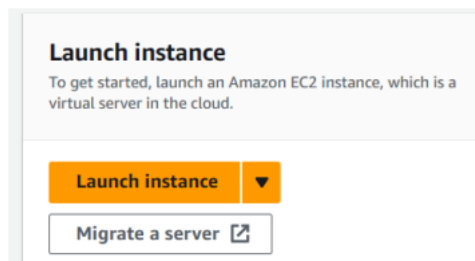
New to AWS?

Create a new AWS account

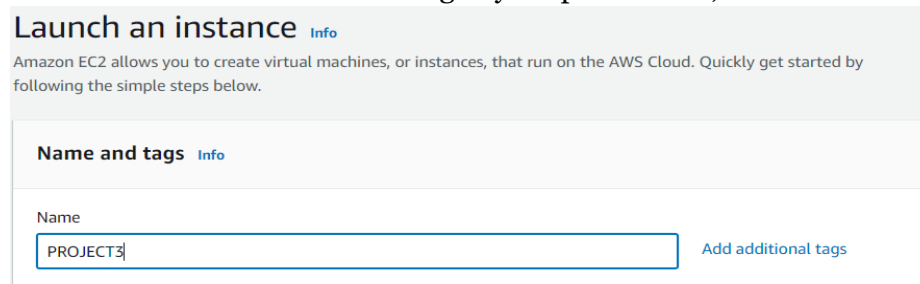
- Choose EC2 and click launch instance



- Then under the EC2 management console we could able to find the Launch instance option, click that one for creating an instance:



- Then name the instance according to your preferences,



- Then we need to select an AMI Image for the instance, choose AMI image according to your preferences: Here I am selecting Ubuntu 20.04 AMI Image.

Recents | **Quick Start**

Amazon Linux
aws

macOS
Mac

Ubuntu
ubuntu

Windows
Microsoft

Red Hat
Red Hat

SUSE Li
SUSE

[Browse more AMIs](#)
 Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type
 ami-0a7cf821b91bccbc (64-bit (x86)) / ami-025a235c91853ccbe (64-bit (Arm))
 Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

- Then we need to select the instance type

▼ **Instance type** [Info](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
 On-Demand Linux base pricing: 0.0124 USD per Hour
 On-Demand Windows base pricing: 0.017 USD per Hour
 On-Demand RHEL base pricing: 0.0724 USD per Hour
 On-Demand SUSE base pricing: 0.0124 USD per Hour

[Additional costs apply for AMIs with pre-installed software](#)

- Then we need to select the keypair for security authentication purpose of the instance

▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select

[Create new key pair](#)

Q Linux X

- Then under network settings, I am selecting default VPC, subnet no preference option, auto-assign public is enabled by default for default VPC & Subnet.

▼ **Network settings** [Info](#)

VPC - *required* [Info](#)

vpc-04dc687e3ffd22a68
172.31.0.0/16

(default) ▼

Subnet [Info](#)

No preference

▼

[Create new subnet](#) [↗](#)

Auto-assign public IP [Info](#)

Enable

▼

- Configure storage and launch a instance.

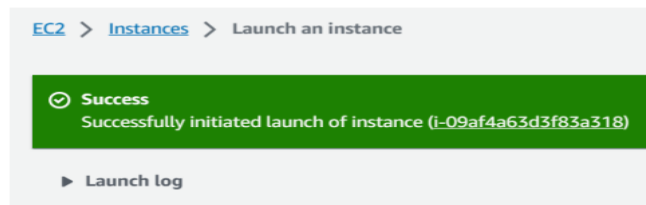
▼ **Configure storage** [Info](#) [Advanced](#)

1x GiB Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

[Add new volume](#)

0 x File systems [Edit](#)



Jenkins

- Install Jenkins in our EC2 machine.

Prerequisites

Minimum hardware requirements:

- 256 MB of RAM
- 1 GB of drive space (although 10 GB is a recommended minimum if running Jenkins as a Docker container)
- Jenkins requires Java to run, so we need to install java using this commands.

sudo apt update

```
ubuntu@ip-172-31-19-109:~$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1149 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [245 kB]
```

sudo apt install fontconfig openjdk-17-jre

```
ubuntu@ip-172-31-19-109:~$ sudo apt install fontconfig openjdk-17-jre
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
fontconfig is already the newest version (2.13.1-4.2ubuntu5).
fontconfig set to manually installed.
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf ca-certificates-java fonts-dejavu-extra java-common libasound2 libasound2-data libatk-wrappers1.0 libatk1.0-0 libatspi2.0-0 libcairo-gobject2 libcairo2 libdatrie1 libfontconfig1 libfreetype6 libgdk-pixbuf2.0 libglib2.0-0 libgtk-3-0 libharfbuzz0b libidn2-0 libjpeg-turbo8 libjpeg8 liblcms2-2 libltdl7 libnss3 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpixman-1-0 libpng16-16 libpsl5 libpython3.10 librtmp1 libsecret-1-0 libsharpyuv0 libsm6 libssl3 libtcl8.6 libthai0 libtiff5 libunistring2 libvorbis0a libwebp6 libx11-6 libx11-data libx11-xcb1 libxcb1 libxcomposite1 libxcursor1 libxdamage1 libxfixes3 libxi6 libxinerama1 libxkbcommon0 libxrandr2 libxrender1 libxsharpev1 libxss1 libxt6 libxtst6 libxv1 libxxf86vm1 libzstd1
Suggested packages:
  default-jre libasound2-plugins alsa-utils pcscd libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zencai ttf-bitstream-vera
The following NEW packages will be installed:
  alsa-topology-conf alsa-ucm-conf ca-certificates-java fonts-dejavu-extra java-common libasound2 libasound2-data libatk-wrappers1.0 libatk1.0-0 libatspi2.0-0 libcairo-gobject2 libcairo2 libdatrie1 libfontconfig1 libfreetype6 libgdk-pixbuf2.0 libglib2.0-0 libgtk-3-0 libharfbuzz0b libidn2-0 libjpeg-turbo8 libjpeg8 liblcms2-2 libltdl7 libnss3 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpixman-1-0 libpng16-16 libpsl5 libpython3.10 librtmp1 libsecret-1-0 libsharpyuv0 libsm6 libssl3 libtcl8.6 libthai0 libtiff5 libunistring2 libvorbis0a libwebp6 libx11-6 libx11-data libx11-xcb1 libxcb1 libxcomposite1 libxcursor1 libxdamage1 libxfixes3 libxi6 libxinerama1 libxkbcommon0 libxrandr2 libxrender1 libxsharpev1 libxss1 libxt6 libxtst6 libxv1 libxxf86vm1 libzstd1
Need to get 51.2 MB of archives.
After this operation, 204 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 alsa-topology-conf all 1.2.5.1-2 [15.5 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libasound2-data all 1.2.6.1-1ubuntu1 [19.1 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libasound2 amd64 1.2.6.1-1ubuntu1 [390 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 alsa-ucm-conf all 1.2.6.3-1ubuntu1.8 [43.3 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 java-common all 0.72build2 [6782 B]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpcsc-lite amd64 1.9.5-3ubuntu1 [19.8 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jre-headless amd64 17.0.8.1+1-0ubuntu0~22.04.1 [114.1 MB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ca-certificates-java all 20190909ubuntu1.2 [11.4 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-extra all 2.37-2build1 [2041 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libatlas3-base amd64 3.10.4-4ubuntu1 [13.1 kB]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libgif7 amd64 5.1.9-2build2 [33.8 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libglib2.0-0 amd64 2.76.6-0ubuntu0.22.04.1 [1212 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jre amd64 17.0.8.1+1-0ubuntu0~22.04.1 [114.1 MB]
```

java -version

```
ubuntu@ip-172-31-19-109:~$ java -version
openjdk version "17.0.8.1" 2023-08-24
OpenJDK Runtime Environment (build 17.0.8.1+1-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 17.0.8.1+1-Ubuntu-0ubuntu122.04, mixed mode, sharing)
ubuntu@ip-172-31-19-109:~$
```

- Once java installation completes next start installing Jenkins using this commands.

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc
\https://pkg.jenkins.io/debian/jenkins.io-2023.key
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
ubuntu@ip-172-31-19-109:~$ sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \https://pkg.jenkins.io/debian/jenkins.io-2023.key
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
--2023-11-04 11:41:37-- https://pkg.jenkins.io/debian/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.78.133, 2a04:4e42:83::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)[146.75.78.133]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins-keyring.asc 100%[=====]
2023-11-04 11:41:37 (25.0 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]
```

sudo apt-get update sudo apt-get install Jenkins

```
sudo apt-get install jenkins
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:5 https://pkg.jenkins.io/debian binary/ InRelease
Get:6 https://pkg.jenkins.io/debian binary/ Release [2044 B]
Get:7 https://pkg.jenkins.io/debian binary/ Release.gpg [833 B]
Get:8 https://pkg.jenkins.io/debian binary/ Packages [57.4 kB]
Fetched 60.3 kB in 1s (64.7 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 21 not upgraded.
Need to get 89.2 MB of archives.
After this operation, 90.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Get:2 https://pkg.jenkins.io/debian binary/ jenkins 2.430 [89.0 MB]
Fetched 89.2 MB in 5s (16.5 MB/s)
Selecting previously unselected package net-tools.
(Reading database ... 124560 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../archives/jenkins_2.430_all.deb ...
Unpacking jenkins (2.430) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.430) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
```

- Once Jenkins installation finished we need to open port (8080) for accessing our Jenkins using aws security groups.
- Go to instance security groups and edit inbound rules and add 8080 with custom ip range and save.

Edit inbound rules [info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type info	Protocol info	Port range info	Source info	Description - optional info	
sg-08f50871e2c3558	All traffic	All	All	Custom	Q	Delete
sg-0d4cb1790dfa73eb	Custom TCP	TCP	8080	Custom	Q	Delete

- Next Accessing our Jenkins server using our instance public ip follows with port 8080.

<http://IP:8080>



- Getting started, the Jenkins page will open and paste our admin password.
- Use this command to get Jenkins admin password

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](#)

```
ubuntu@ip-172-31-19-109:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
b8d4fff8d5ce460089f8e6f191ac8115
ubuntu@ip-172-31-19-109:~$
```

- Copy and paste the password from terminal Jenkins page next click install suggested plugins.

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

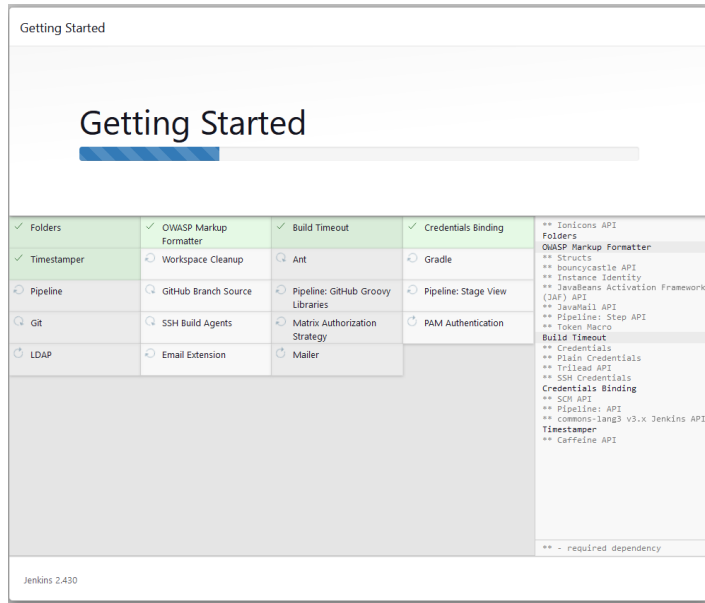
Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.430

- It will install all plugins



- Create a user to work with Jenkins or continue as admin.

The screenshot shows the 'Create First Admin User' form in Jenkins. It includes input fields for Username, Password, Confirm password, Full name, and E-mail address. At the bottom, there are two buttons: 'Skip and continue as admin' and 'Save and Continue'.

- Click save and continue to enter Jenkins dashboard.

Getting Started

Instance Configuration

Jenkins URL:

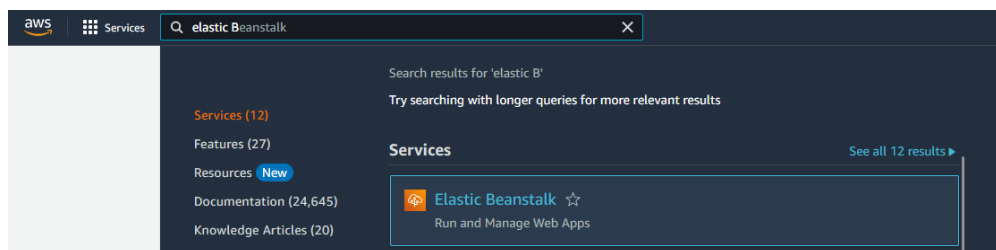
The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

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.

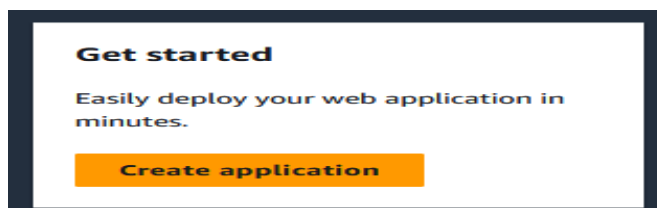
Jenkins 2.430 Not now Save and Finish

AWS Elastic Beanstalk

- Create and configure Elastic Beanstalk application in aws to deploy our nodejs application.
- First we need to login aws management console.
- Search and select elastic beanstalk service.



- Click Create application button



- First we need to configure our environment as a web server environment.

Configure environment [Info](#)

Environment tier [Info](#)
 Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

☒ **Web server environment**
 Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

☐ **Worker environment**
 Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

- Give name to our Application and also give environment name.

Application information [Info](#)

Application name

Maximum length of 100 characters.

► **Application tags (optional)**

Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain

.us-east-2.elasticbeanstalk.com

Check availability

Environment description

- Select Platform for our application and choose version for our application.

Platform [Info](#)

Platform type

☒ **Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

☐ **Custom platform**
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Platform branch

Platform version

- Choose sample code for our application.

Application code [Info](#)

☒ **Sample application**

☐ **Existing version**
Application versions that you have uploaded.

☐ **Upload your code**
Upload a source bundle from your computer or copy one from Amazon S3.

- Choose presets for our application as per needs and click next.

Presets [Info](#)

Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets

- ☒ Single instance (free tier eligible)
- ☐ Single instance (using spot instance)
- ☐ High availability
- ☐ High availability (using spot and on-demand instances)
- ☐ Custom configuration

Cancel **Next**

- Configure service step we need to choose our service role with permissions of **EC2, Elastic Beanstalk and S3**.
- We can attach those permissions as a role policies in IAM console.

Configure service access [Info](#)

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

- ☐ Create and use new service role
- ☒ Use an existing service role

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

EC2role

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

linux

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

EC2role

[View permission details](#)

Cancel Skip to review Previous **Next**

- Next we need to configure networks for this to choose our VPC.

Set up networking, database, and tags - optional [Info](#)

Virtual Private Cloud (VPC)

VPC

Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. [Learn more](#)

vpc-05fd04c5db0441f92 | (172.31.0.0/16)

[Create custom VPC](#)

- Select public ip checkbox as well as select subnets for launch our application.

Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. [Learn more](#)

Public IP address
Assign a public IP address to the Amazon EC2 instances in your environment.
☒ **Activated**

Instance subnets

<input type="checkbox"/>	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	us-east-2c	subnet-026b11f48...	172.31.32.0/20	
<input type="checkbox"/>	us-east-2b	subnet-0878a931c...	172.31.16.0/20	
<input type="checkbox"/>	us-east-2a	subnet-0bb100bf4...	172.31.0.0/20	

- Choose others that are optional and click next.
- Next configure instance details as go with default values.

Configure instance traffic and scaling - *optional* [Info](#)

▼ **Instances** [Info](#)
Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type
(Container default) ▼

Size
The number of gigabytes of the root volume attached to each instance.
8 GB

IOPS
Input/output operations per second for a provisioned IOPS (SSD) volume.
100 IOPS

Throughput
The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance
125 MiB/s

Amazon CloudWatch monitoring
The time interval between when metrics are reported from the EC2 instances

Monitoring interval
5 minute ▼

- Choose our security group.

Instance metadata service (IMDS)
Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. [Learn more](#)

IMDSv1
With the current setting, the environment enables only IMDSv2.
☒ **Deactivated**

EC2 security groups
Select security groups to control traffic.

EC2 security groups (4) [Refresh](#)

<input type="checkbox"/>	Group name	Group ID	Name
<input type="checkbox"/>	ansible	sg-0b4104ccb5d57ca17	
<input type="checkbox"/>	default	sg-07f2d090400319516	
<input type="checkbox"/>	launch-wizard-1	sg-00c39b1bbe82caae8	
<input type="checkbox"/>	launch-wizard-2	sg-01057229e54f082b6	

- Select our capacity for instance choose single or multiple instance.

▼ **Capacity** [Info](#)

Configure the compute capacity of your environment and auto scaling settings to optimize the number of instances used.

Auto scaling group

Environment type
Select a single-instance or load-balanced environment. You can develop and test an application in a single-instance environment to save costs and then upgrade to a load-balanced environment when the application is ready for production. [Learn more](#)

Single instance ▼

Instances

1 Min

1 Max

Fleet composition
Spot instances are launched at the lowest available price. [Learn more](#)

☒ On-Demand instance

☐ Spot instance

- Also choose architecture and instance type (ex: t2.micro) and AMI ID and click next.

Architecture
The processor architecture determines the instance types that are made available. You can't change this selection after you create the environment. [Learn more](#)

☒ x86_64
This architecture uses x86 processors and is compatible with most third-party tools and libraries.

☐ arm64 - new
This architecture uses AWS Graviton2 processors. You might have to recompile some third-party tools and libraries.

Instance types
Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. [Learn more](#)

Choose x86 instance types ▼

t2.micro ✕

AMI ID
Elastic Beanstalk selects a default Amazon Machine Image (AMI) for your environment based on the Region, platform version, and processor architecture that you choose. [Learn more](#)

ami-0d9d9dd5a14e0043b

Availability Zones
Number of Availability Zones (AZs) to use.

Any ▼

Placement
Specify Availability Zones (AZs) to use.

Choose Availability Zones (AZs) ▼

Scaling cooldown

360 seconds

Cancel Skip to review Previous **Next**

- Next, configure the monitoring process and select basic monitoring.

Configure updates, monitoring, and logging - optional [Info](#)

▼ **Monitoring** [Info](#)

Health reporting
Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#)

System

☒ Basic

☐ Enhanced

- Choose our server proxy as Nginx or Apache and choose other options as default and click next.

▼ **Platform software** [Info](#)
 Configure the options available to your specific platform. These include the proxy server and OS environment properties. [Learn more](#)

Container options

Proxy server

Nginx	▲
Apache	
Nginx	✓

Amazon X-Ray is a service that collects data about the requests and responses that your application serves and receives. You can use the tools that X-Ray offers to view and filter the data that it provides to identify potential issues and optimization opportunities.

- Final step should be review and submit. It will create a nodejs application on beanstalk.

Jenkins Configuration for Nodejs application deployment

- First Login to Jenkins dashboard and install required plugins.
- **Manage Jenkins >> Plugins >> Available Plugins**
 1. NodeJS Plugin
 2. AWSEB Deployment Plugin
 3. Publish to AWS beanstalk plugin

Name	Enabled
Amazon Web Services SDK :: Elastic Beanstalk 1.12.610-428.v849169a_01b_a_5 Elastic Beanstalk module for the AWS SDK for Java.	<input checked="" type="checkbox"/>
AWS Elastic Beanstalk Publisher Plugin 1.8.2 This Plugin allows you to deploy into AWS Elastic Beanstalk Report an issue with this plugin	<input checked="" type="checkbox"/>

Warning: The currently installed plugin version may not be safe to use. Please review the following security notices:

- [Credentials stored in plain text](#)

- Next we need to configure those plugins so follow the steps.
- Create credentials in the system and use your aws credentials and use the same ID name in jenkins file Deploy stage.

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted)

New credentials

Kind: AWS Credentials

Scope: Global (Jenkins, nodes, items, all child items, etc)

ID:

Description:

Access Key ID:

Secret Access Key:

IAM Role Support: ☐

- **Manage Jenkins >> System >> Deploy into AWS Elastic Beanstalk**

Deploy into AWS Elastic Beanstalk

Credentials

AWS credentials

Name:

AWS Access Key Id:

AWS Secret Access Key:

Advanced

Add

- Configure Name, AWS Access Key Id and AWS Secret Access Key.
- Install node and zip in our ec2 machine using below commands.

sudo apt install npm

sudo apt install zip

sudo apt install awscli

- Install aws cli and configure IAM credentials in terminal using below command.

aws configure

JOB Creation in Jenkins

- In this project we use a **Pipeline** as a Project in Jenkins.
- Create a **Pipeline** Project and config our application details.
- **Dashboard >> New item**

Enter an item name

jenkinsnode

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

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.

OK **Cancel** **Branch Pipeline**

- Give the name to the project and select Pipeline project and click ok to create it.
- Next we need to configure our project with our github code.

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-Build Actions

Description

Plain text [Preview](#)

☐ Commit agent's Docker container ?

☐ Define a Docker template

☐ Discard old builds ?

☒ **GitHub project**

Project url ?

<https://github.com/> [newnode.git/](#)

Advanced v

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

☐ Restrict where this project can be run ?

Advanced v

- Select Github Project and paste our github link.
- On Source code management select Git and paste the same link and select or branch as main.

Dashboard > nodepipe > Configuration **General** Enabled

Configure

General

Advanced Project Options

Pipeline

Description

Plain text [Preview](#)

☐ Discard old builds ?

☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

☒ **GitHub project**

Project url ?

<https://github.com/Vennilavan12/ebs.git/>

- Select our Build Triggers as GitHub hook triggers.

Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts) ?
- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

- Next we need to select pipeline script from scm and choose your repo link and Jenkinsfile.

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main


Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add +

 **Jenkins**

Dashboard > nodepipe >

Status

</> Changes

▷ Build Now

⚙️ Configure

🗑️ Delete Pipeline

🔍 Full Stage View

🔄 GitHub

✎ Rename

📖 Pipeline Syntax

Pipeline nodepipe

Stage View

	Checkout	Build	Package	Deploy to Elastic Beanstalk
Average stage times: (Average full run time: ~3min 39s)	4s	7s	1s	40s
Dec 07 01:42 No Changes	2s	8s	3s	3min 22s

Build History trend

```
upload: ./node_modules.zip to s3://elasticbeanstalk-us-east-2-006095271778/node_modules.zip
[Pipeline] sh
+ aws elasticbeanstalk create-application-version --application-name Nodejs --version-label 8 --source-bundle S3Bucket=elasticbeanstalk-us-east-2-006095271778,S3Key=node_modules.zip
{
  "ApplicationVersion": {
    "ApplicationVersionArn": "arn:aws:elasticbeanstalk:us-east-2:006095271778:applicationversion/Nodejs/8",
    "ApplicationName": "Nodejs",
    "VersionLabel": "8",
    "SourceBundle": {
      "S3Bucket": "elasticbeanstalk-us-east-2-006095271778",
      "S3Key": "node_modules.zip"
    },
    "DateCreated": "2023-12-07T06:46:06.656000+00:00",
    "DateUpdated": "2023-12-07T06:46:06.656000+00:00",
    "Status": "UNPROCESSED"
  }
}
[Pipeline] sh
+ aws elasticbeanstalk update-environment --application-name Nodejs --environment-name Nodejs-env --version-label 8
{
  "EnvironmentName": "Nodejs-env",
  "EnvironmentId": "e-pzxarbtg92",
  "ApplicationName": "Nodejs",
  "VersionLabel": "8",
  "SolutionStackName": "64bit Amazon Linux 2023 v6.0.3 running Node.js 18",
  "PlatformArn": "arn:aws:elasticbeanstalk:us-east-2:platform/Node.js 18 running on 64bit Amazon Linux 2023/6.0.3",
  "EndpointURL": "3.13.98.181",
  "CNAME": "Nodejs-env.eba-bftymmy.us-east-2.elasticbeanstalk.com",
  "DateCreated": "2023-12-07T06:10:32.974000+00:00",
  "DateUpdated": "2023-12-07T06:46:10.040000+00:00",
  "Status": "Updating",
  "AbortableOperationInProgress": true,

```



```

    "EnvironmentId": "e-pzxmrbtg92",
    "ApplicationName": "Nodejs",
    "VersionLabel": "8",
    "SolutionStackName": "64bit Amazon Linux 2023 v6.0.3 running Node.js 18",
    "PlatformArn": "arn:aws:elasticbeanstalk:us-east-2::platform/Node.js 18 running on 64bit Amazon Linux 2023/6.0.3",
    "EndpointURL": "3.13.98.161",
    "CNAMERecord": "Nodejs-env.eba-bftymvuy.us-east-2.elasticbeanstalk.com",
    "DateCreated": "2023-12-07T06:10:32.974000+00:00",
    "DateUpdated": "2023-12-07T06:46:19.840000+00:00",
    "Status": "Updating",
    "AbortableOperationInProgress": true,
    "Health": "Grey",
    "Tier": {
      "Name": "WebServer",
      "Type": "Standard",
      "Version": "1.0"
    },
    "EnvironmentArn": "arn:aws:elasticbeanstalk:us-east-2:006095271778:environment/Nodejs/Nodejs-env"
  },
  "Pipeline": {
    "Script": "node",
    "WithCredentials": true,
    "Stage": "Deploy",
    "WithEnv": true,
    "Node": "node"
  }
}
[Pipeline] End of Pipeline
Finished: SUCCESS

```

- Go to the AWS dashboard and check Elastic Beanstalk environment.

Elastic Beanstalk > Environments > Node-env

Node-env Info

🔄
Actions ▾
Upload and deploy

Environment overview

Health
⊖ Grey

Domain
Node-env.eba-xfemeacd.us-east-2.elasticbeanstalk.com [🔗](#)

Environment ID
e-exdmhdpanp

Application name
Node

Platform

Change version

Platform
Node.js 18 running on 64bit Amazon Linux 2023/6.0.2

Running version
1

Platform state
🟢 Supported

Events (12) Info

🔍 Filter events by text, property or value

Time	Type	Details
December 7, 2023 12:17:03 (UTC+5:30)	🔵 INFO	Environment update completed successfully.
December 7, 2023 12:17:03 (UTC+5:30)	🔵 INFO	New application version was deployed to running EC2 instances.
December 7, 2023 12:16:43 (UTC+5:30)	🔵 INFO	Instance deployment completed successfully.
December 7, 2023 12:16:34 (UTC+5:30)	🔵 INFO	Deploying new version to instance(s).
December 7, 2023 12:16:09 (UTC+5:30)	🔵 INFO	Environment update is starting.
December 7, 2023 11:44:06 (UTC+5:30)	🔵 INFO	Successfully launched environment: Nodejs-env
December 7, 2023 11:43:00 (UTC+5:30)	🔵 INFO	Instance deployment completed successfully.
December 7, 2023 11:41:55 (UTC+5:30)	🔵 INFO	Waiting for EC2 instances to launch. This may take a few minutes.
December 7, 2023 11:41:08 (UTC+5:30)	🔵 INFO	Created EIP: 3.13.98.161
December 7, 2023 11:40:52 (UTC+5:30)	🔵 INFO	Created security group named: awseb-e-pzxmrbtg92-stack-AWSECGroup-1LU7DHFV6N9N7
December 7, 2023 11:40:33 (UTC+5:30)	🔵 INFO	Using elasticbeanstalk-us-east-2-006095271778 as Amazon S3 storage bucket for environment data.
December 7, 2023 11:40:32 (UTC+5:30)	🔵 INFO	createEnvironment is starting.

- Check the Beanstalk environment to reach our website.



[Git](#) [Jenkins](#) [Docker](#) [Kubernetes](#)

Reference:

github url: <https://github.com/Vennilavan12/AWS-EB.git>