



Automating Deployment:

Node.js REST API CI/CD Pipeline on AWS EC2

with GitHub Actions

This document provides a comprehensive guide to setting up a complete CI/CD pipeline for deploying a Node.js REST API on an AWS EC2 instance using GitHub Actions. It covers every step, from configuring the AWS environment to automating build, test, and deployment processes.

Key highlights include:

- **EC2 Instance Setup:** Installing and configuring the necessary tools, such as Node.js, npm, and Git, on the AWS EC2 instance.
- **GitHub Actions Workflow:** Writing reusable and efficient workflows to automate the testing, building, and deployment stages of the Node.js application.
- **Environment Variables Management:** Ensuring secure handling of sensitive information like AWS credentials and instance details.
- **Seamless Integration:** Demonstrating how GitHub Actions integrate with AWS to streamline the deployment process.
- **Error Handling and Debugging:** Best practices for monitoring and resolving potential issues during deployment.

This guide is tailored for those seeking to understand the practical implementation of DevOps practices, focusing on continuous integration and delivery workflows for modern cloud environments.



THISARA KANDAGE

UNDERGRADUATE - SLIIT

[E-mail](#) [LinkedIn](#) [GitHub](#) [Website](#)

Created a GitHub repository

The screenshot shows a GitHub repository named 'nodejs-rest-api-EC2' by user 'Thisarak943'. The repository is in the 'main' branch. The file structure on the left includes:

- nodejs-rest-api-EC2
 - .github/workflows
 - controllers
 - db
 - middlewares
 - models
 - routes
 - .gitignore
 - README.md
 - package-lock.json
 - package.json
 - server.js
 - package-lock.json

The commit history table shows the following commits:

Name	Last commit message	Last commit date
..		
.github/workflows	first commit	1 hour ago
controllers	first commit	1 hour ago
db	done	12 minutes ago
middlewares	first commit	1 hour ago
models	first commit	1 hour ago
routes	first commit	1 hour ago
.gitignore	first commit	1 hour ago
README.md	first commit	1 hour ago
package-lock.json	done	12 minutes ago
package.json	done	12 minutes ago
server.js	first commit	1 hour ago

Create AWS EC2 instance with Ubuntu

The screenshot shows the AWS Management Console 'Launch instance' page. The breadcrumb navigation is 'EC2 > Instances > Launch an instance'. The page title is 'Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.'

Name and tags [Info](#)

Name: [Add additional tags](#)

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recents | **Quick Start**

Amazon Linux | macOS | **Ubuntu** | Windows | Red Hat | SUSE Linux | Debian

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

Summary

Number of instances [Info](#)

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0e2c8caa4b6378d8c

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

[Cancel](#) [Launch instance](#) [Preview code](#)



THISARA KANDAGE

UNDERGRADUATE - SLIIT

[E-mail](#) [LinkedIn](#) [GitHub](#) [Website](#)

[Alt+S]

[EC2](#) > [Instances](#) > Launch an instance

✓ **Success**
Successfully initiated launch of instance ([i-0bf69e46d6a1543f9](#))

▶ [Launch log](#)

Connect via SSH

[Alt+S] N. Virginia Thisara Kandage

[EC2](#) > [Instances](#) > [i-0bf69e46d6a1543f9](#) > Connect to instance

Connect to instance [Info](#)
Connect to your instance i-0bf69e46d6a1543f9 (AWSdeploy1) using any of these options

EC2 Instance Connect Session Manager **SSH client** EC2 serial console

Instance ID
 [i-0bf69e46d6a1543f9](#) (AWSdeploy1)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is AWSLinuxAssign.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 `chmod 400 "AWSLinuxAssign.pem"`
4. Connect to your instance using its Public DNS:
 `ec2-18-207-206-209.compute-1.amazonaws.com`

✓ Command copied

`ssh -i "AWSLinuxAssign.pem" ubuntu@ec2-18-207-206-209.compute-1.amazonaws.com`

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

[Cancel](#)



THISARA KANDAGE

UNDERGRADUATE - SLIIT

[E-mail](#) [LinkedIn](#) [GitHub](#) [Website](#)

```
PS C:\Users\Thisara K\Downloads\LinuxTest> ssh -i "AWSLinuxAssign.pem" ubuntu@ec2-18-207-206-209.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sun Dec 29 07:50:18 UTC 2024

System load:  0.0          Processes:      105
Usage of /:   24.6% of 6.71GB Users logged in: 0
Memory usage: 20%         IPv4 address for enx0: 172.31.92.212
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-92-212:~$
```

Create Self Hosted Runner on GitHub

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

General

Runners

Webhooks

Environments

Codespaces

Pages

Security

Code security

Deploy keys

Secrets and variables

Runners / Add new self-hosted runner · Thisarak943/AWS-deploy1

Adding a self-hosted runner requires that you download, configure, and execute the GitHub Actions Runner. By downloading and configuring the GitHub Actions Runner, you agree to the [GitHub Terms of Service](#) or [GitHub Corporate Terms of Service](#), as applicable.

Runner image

macOS

Linux

Windows

Architecture

x64

Download

```
# Create a folder
$ mkdir actions-runner && cd actions-runner

# Download the latest runner package
$ curl -o actions-runner-linux-x64-2.321.0.tar.gz -L
https://github.com/actions/runner/releases/download/v2.321.0/actions-runner-linux-x64-2.321.0.tar.gz

# Optional: Validate the hash
$ echo "ba46ba7ce3a4d7236b16f6e44419fb453bc08f866b24f04d549ec89f1722a29e" actions-runner-linux-x64-
2.321.0.tar.gz | shasum -a 256 -c

# Extract the installer
$ tar xzf ./actions-runner-linux-x64-2.321.0.tar.gz
```



Download Part

```
ubuntu@ip-172-31-92-212:~$ mkdir actions-runner && cd actions-runner
ubuntu@ip-172-31-92-212:~/actions-runner$ curl -o actions-runner-linux-x64-2.321.0.tar.gz -L https://github.com/actions/runner/releases/download/v2.321.0/actions-runner-linux-x64-2.321.0.tar.gz
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00 0
100 113M 100 113M 0 0 82.4M 0 0:00:01 0:00:01 --:--:-- 111M
ubuntu@ip-172-31-92-212:~/actions-runner$ echo "ba46ba7ce3a4d7236b16fbe44419fb453bc08f866b24f04d549ec89f1722a29e actions-runner-linux-x64-2.321.0.tar.gz" | shasum -a 256 -c
actions-runner-linux-x64-2.321.0.tar.gz: OK
ubuntu@ip-172-31-92-212:~/actions-runner$ tar xzf ./actions-runner-linux-x64-2.321.0.tar.gz
ubuntu@ip-172-31-92-212:~/actions-runner$
```

Configuration part

```
ubuntu@ip-172-31-92-212:~/actions-runner$ ./config.sh --url https://github.com/Thisarak943/AWS-deploy1 --token BGDYHC4SSFMBQVPLKDL2RGDHOEG20
Self-hosted runner registration

# Authentication

✓ Connected to GitHub

# Runner Registration

Enter the name of the runner group to add this runner to: [press Enter for Default]

Enter the name of runner: [press Enter for ip-172-31-92-212]

This runner will have the following labels: 'self-hosted', 'Linux', 'X64'
Enter any additional labels (ex. label-1,label-2): [press Enter to skip]

✓ Runner successfully added
✓ Runner connection is good
```

Runners

New self-hosted runner

Host your own runners and customize the environment used to run jobs in your GitHub Actions workflows. [Learn more about self-hosted runners.](#)

Runners	Status
 ip-172-31-92-212 self-hosted Linux X64	● Offline ...



THISARA KANDAGE

UNDERGRADUATE - SLIIT

[E-mail](#) [LinkedIn](#) [GitHub](#) [Website](#)

Add new Repository Secret

Actions secrets / New secret

Name *

MONGO_DB_URL

Secret *

MONGO_DB_URL=mongodb+srv://thisarak943:thisara123@cluster0.avmbe.mongodb.net/?
retryWrites=true&w=majority&appName=Cluster0

Add secret

Continuous integration

[View all](#)

SLSA Generic generator

By Open Source Security
Foundation (OpenSSF)

Generate SLSA3 provenance for your
existing release workflows

Configure

Go

Node.js

By GitHub Actions

Build and test a Node.js project with npm.

Configure

JavaScript

Datadog Synthetics

By Datadog

Run Datadog Synthetic tests within your
GitHub Actions workflow

Configure

JavaScript

Symfony

By GitHub Actions

Test a Symfony project.

Configure

PHP



Node.js.yml File

name: Node.js CI/CD

on:

push:

branches: ["main"]

jobs:

build:

runs-on: self-hosted

strategy:

matrix:

node-version: [20.x]

See supported Node.js release schedule at <https://nodejs.org/en/about/releases/>

steps:

- uses: actions/checkout@v3

- name: Use Node.js \${{ matrix.node-version }}

uses: actions/setup-node@v3

with:

node-version: \${{ matrix.node-version }}

cache: 'npm'

- run: npm ci

- run: |

touch .env

echo "\${{ secrets.MONGO_DB_URL }}" > .env

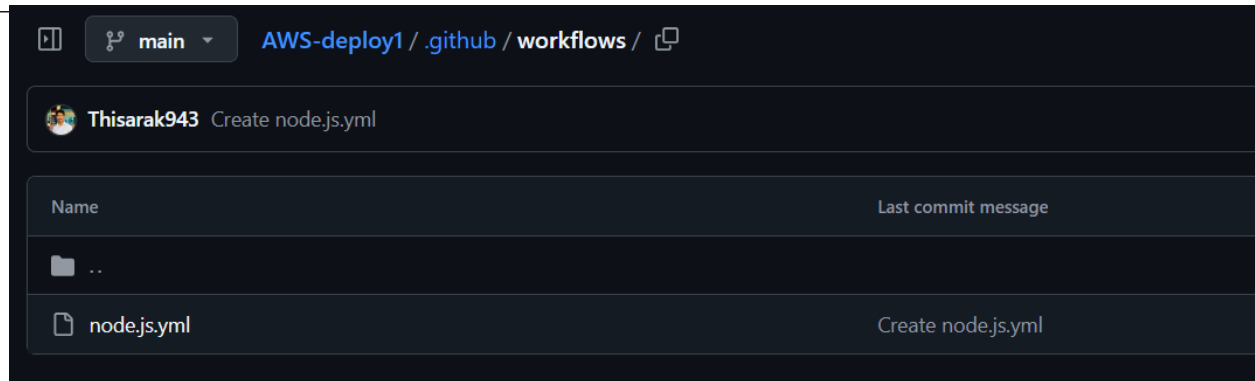
- run: pm2 restart BackendAPI



THISARA KANDAGE

UNDERGRADUATE - SLIIT

[E-mail](#) [LinkedIn](#) [GitHub](#) [Website](#)



Environmental Setup in Ubuntu

Ensure Node.js and Nginx are installed on my Ubuntu instance.

```
ubuntu@ip-172-31-92-212:~/actions-runner$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [572 kB]

ubuntu@ip-172-31-92-212:~/actions-runner$ curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -
2024-12-29 08:25:56 - Installing pre-requisites
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
```



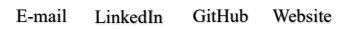

THISARA KANDAGE

UNDERGRADUATE - SLIIT

[E-mail](#) [LinkedIn](#) [GitHub](#) [Website](#)

```
ubuntu@ip-172-31-92-212:~/actions-runner$ sudo apt-get install -y nodejs nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  nginx nginx-common nodejs
0 upgraded, 3 newly installed, 0 to remove and 55 not upgraded.
Need to get 36.7 MB of archives.
```

```
ubuntu@ip-172-31-92-212:~/actions-runner$ node -v
v22.12.0
ubuntu@ip-172-31-92-212:~/actions-runner$ npm -v
10.9.0
ubuntu@ip-172-31-92-212:~/actions-runner$
```



A 10x10 grid of 100 small, stylized line drawings. Each drawing is composed of simple lines and dots, creating a variety of abstract forms. The shapes are arranged in a grid, with each row containing 10 shapes and each column containing 10 shapes. The shapes are diverse, including lines, dots, and combinations of both, forming various geometric and organic patterns.



THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

Start the Runners

```
ubuntu@ip-172-31-92-212:~/actions-runner$ sudo ./svc.sh install
Creating launch runner in /etc/systemd/system/actions.runner.Thisarak943-AWS-deploy1.ip-172-31-92-212.service
Run as user: ubuntu
Run as uid: 1000
gid: 1000
Created symlink /etc/systemd/system/multi-user.target.wants/actions.runner.Thisarak943-AWS-deploy1.ip-172-31-92-212.service → /etc/systemd/system/actions.runner.Thisarak943-AWS-deploy1.ip-172-31-92-212.service.
ubuntu@ip-172-31-92-212:~/actions-runner$ sudo ./svc.sh start


/etc/systemd/system/actions.runner.Thisarak943-AWS-deploy1.ip-172-31-92-212.service
● actions.runner.Thisarak943-AWS-deploy1.ip-172-31-92-212.service - Github Actions Runner (Thisarak943-AWS-deploy1.ip-172-31-92-212)
   Loaded: loaded (/etc/systemd/system/actions.runner.Thisarak943-AWS-deploy1.ip-172-31-92-212.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-12-29 08:34:36 UTC; 14ms ago
     Main PID: 2794 (runsv.sh)
       Tasks: 1 (limit: 1130)
      Memory: 236.0K (peak: 756.0K)
         CPU: 3ms
    CGroup: /system.slice/actions.runner.Thisarak943-AWS-deploy1.ip-172-31-92-212.service
            └─2794 /bin/bash /home/ubuntu/actions-runner/runsv.sh
              2796 cat .path

Dec 29 08:34:36 ip-172-31-92-212 systemd[1]: Started actions.runner.Thisarak943-AWS-deploy1.ip-172-31-92-212.service - Github Actions Runner (Thisarak943-AWS-deploy1.ip-172-31-92-212).
Dec 29 08:34:36 ip-172-31-92-212 runsv.sh[2794]: .path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
Hint: Some lines were ellipsized, use -l to show in full.
ubuntu@ip-172-31-92-212:~/actions-runner$
```

Runners

New self-hosted runner

Host your own runners and customize the environment used to run jobs in your GitHub Actions workflows. [Learn more about self-hosted runners.](#)

Runners	Status
<div><div> ip-172-31-92-212</div><div>self-hosted</div><div>Linux</div><div>X64</div></div>	<div><div></div> Idle</div> <div>...</div>



PM2 logs in terminal

```
/home/ubuntu/.pm2/logs/server-out.log last 15 lines:
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | Server is running on port: 8000
0|server    | MongoDB connection established
```

Creating Nginx reverse proxy

```
Make pm2 auto-boot at server restart.
GNU nano 7.2 /etc/nginx/sites-available/default *
#       fastcgi_pass 127.0.0.1:9000;
#}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny all;
#}
}

# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.
#
#server {
#    listen 80;
#    listen [::]:80;
#
#    server_name example.com;
}
GNU nano 7.2 /etc/nginx/sites-available/default *
#       location / {
#           try_files $uri $uri/ =404;
#       }
#}
location /api {
    rewrite ^/api/(.*)$ /api/$1 break;
    proxy_pass http://localhost:5000;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
}
```



Conclusion

1. Challenges Faced:

- Configuring the AWS EC2 instance for deployments, including setting up security groups and SSH access.
- Managing AWS credentials securely within the CI/CD pipeline.
- Debugging GitHub Actions workflows due to misconfigurations in YAML files and incorrect environment variables.

2. Solutions Implemented:

- Researched AWS documentation to properly configure EC2 instances.
- Used GitHub Secrets to securely store and manage sensitive information.
- Analyzed GitHub Actions logs and applied systematic debugging to resolve pipeline issues.

3. Key Learnings:

- Gained practical knowledge of cloud infrastructure setup and management.
- Improved understanding of CI/CD processes, automation, and workflow optimization.
- Developed skills in securing pipelines and handling environment variables.

4. Impact on Knowledge:

- Enhanced technical proficiency in deploying scalable and reliable applications.
- Strengthened problem-solving abilities and confidence in tackling DevOps challenges.
- Acquired hands-on experience with tools and practices critical to modern DevOps workflows.