

Day 24 Task: Complete Jenkins CI/CD Project:

Let's make a beautiful CI/CD Pipeline for your Node JS Application 🥰

Task-01:

Fork [this](#) repository:

The screenshot shows a GitHub repository page for 'node-todo-cicd', which is a fork of 'LondheShubham153/node-todo-cicd'. The repository is currently on the 'master' branch, which is 4 commits ahead of the upstream master. The repository contains several files and folders, including 'views', '.gitignore', 'Dockerfile', 'Jenkinsfile', 'README.md', 'app.js', 'docker-compose.yaml', 'package-lock.json', 'package.json', and 'test.js'. The repository has 30 commits, 0 stars, 0 watching, and 509 forks. The repository is also listed in the 'Releases' and 'Packages' sections, with no releases or packages published yet.

File/Folder	Description	Time
views	Update todo.ejs	8 hours ago
.gitignore	added code and dockerfile	last month
Dockerfile	added tests	2 weeks ago
Jenkinsfile	Added jenkinsfile for agent specific	2 weeks ago
README.md	Update README.md	last month
app.js	added code and dockerfile	last month
docker-compose.yaml	Update docker-compose.yaml	8 hours ago
package-lock.json	added code and dockerfile	last month
package.json	added tests	2 weeks ago
test.js	added tests	2 weeks ago

Create a connection to your Jenkins job and your GitHub Repository via GitHub Integration.

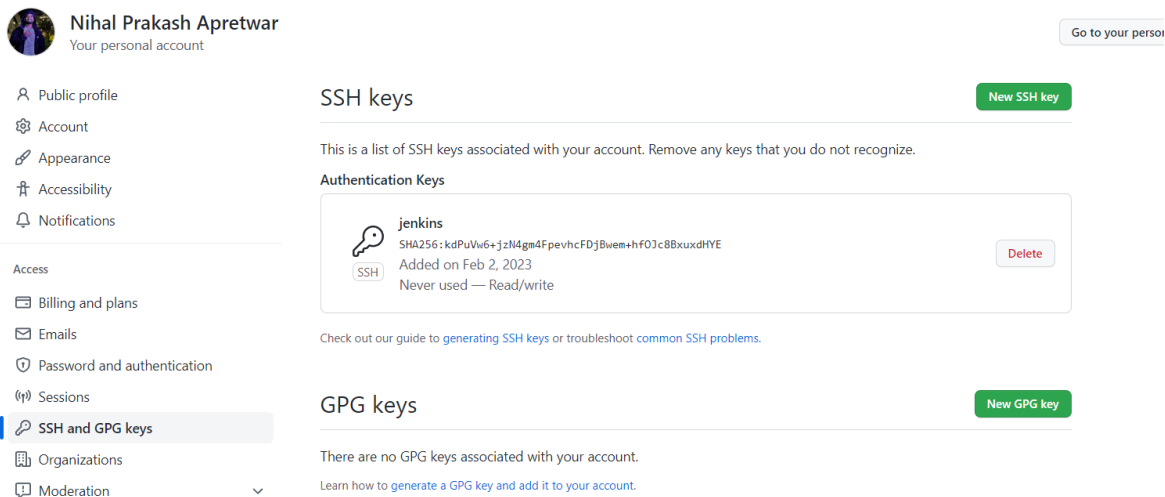
- Generate the SSH keys for integrating your Jenkins project with your git repository. Use ssh-keygen command to create public and private key.

```

ubuntu@ip-172-31-55-25:~$
ubuntu@ip-172-31-55-25:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
/home/ubuntu/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:shhDbYlcAUPGgV7i6xdxAu98L5bDDFgD3GGBETzyX8k ubuntu@ip-172-31-55-25
The key's randomart image is:
+---[RSA 3072]-----+
|      =B0=o      |
|    +.BoB.      |
| o =.*.o . .    |
| oo=o+. E       |
| +=+=S+         |
| . +*.oX        |
| . .o.o .       |
| .              |
| .              |
+---[SHA256]-----+
ubuntu@ip-172-31-55-25:~$
ubuntu@ip-172-31-55-25:~$ cd .ssh/
ubuntu@ip-172-31-55-25:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@ip-172-31-55-25:~/.ssh$ |

```

- Now, go to your GitHub account settings.
- Go to **SSH and GPG keys**, Add public key that we created using ssh-keygen and select key-type Authentication key.



The screenshot shows the GitHub account settings page for Nihal Prakash Apretwar. The left sidebar contains navigation links: Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, **SSH and GPG keys** (selected), Organizations, and Moderation. The main content area is titled 'SSH keys' and includes a 'New SSH key' button. Below the title, it states: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' Under the 'Authentication Keys' section, there is one key named 'jenkins' with a key icon, SHA256 fingerprint, and a 'Delete' button. Below this, there is a link to a guide on generating SSH keys. The 'GPG keys' section below it shows 'There are no GPG keys associated with your account.' and a link to learn how to generate a GPG key.

For GitHub-Webhook:

- Go to your GitHub repository and click on **Settings**.
- Click on **Webhooks** and then click on **Add webhook**.
- In the 'Payload URL' field, paste your Jenkins environment URL. At the end of this URL add /github-webhook/. In the 'Content type' select: 'application/json' and leave the 'Secret' field empty.

nihala19 / node-todo-cicd Public

forked from LondheShubham153/node-todo-cicd

< Code Pull requests Actions Projects Wiki Security Insights **Settings**

General
Access
Collaborators
Moderation options
Code and automation
Branches
Tags
Actions
Webhooks
Environments
Codespaces
Pages
Security
Code security and analysis
Deploy keys
Secrets and variables

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

Content type

Secret

Which events would you like to trigger this webhook?

☒ Just the push event.

☐ Send me **everything**.

☐ Let me select individual events.

☒ **Active**

We will deliver event details when this hook is triggered

- Now for Installing GitHub Integration plugin in Jenkins
- Open your jenkins dashboard.
- Click on the Manage Jenkins button on your Jenkins dashboard
- Click on Manage Plugins
- Install GitHub Integration plugin

Jenkins Search (CTRL+K) [?] [bell] [1] [shield] [1] Nihal Apretwar

Dashboard > Manage Jenkins > Plugin Manager

Updates
Available plugins
Installed plugins
Advanced settings

Plugins

Name	Enabled
GitHub Integration Plugin 0.4.0 GitHub Integration Plugin for Jenkins	<input checked="" type="checkbox"/>

- Configuring Jenkins:
- Create a jenkins job

Jenkins Search (CTRL+K) [?] [bell] [shield] [1] Nihal Apretwar log out

Dashboard >

+ New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views

All +

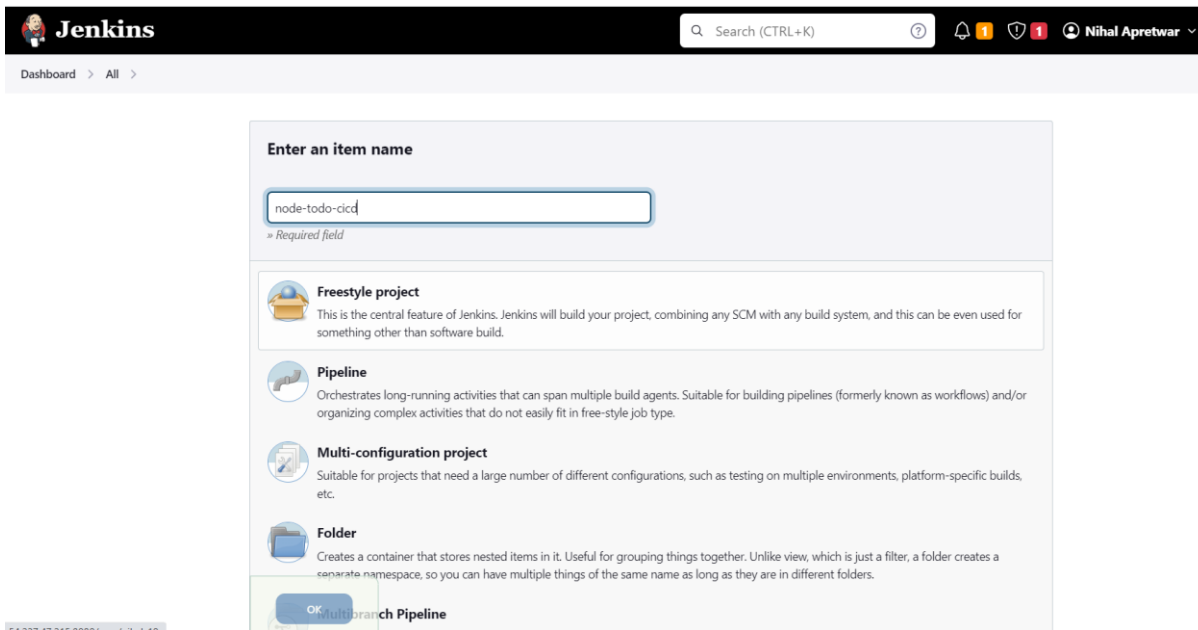
S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	django-app-pipeline	8 hr 55 min #29	1 day 9 hr #21	32 sec
✓	☀	my_pipeline	2 days 23 hr #1	N/A	0.26 sec

Icon: S M L
Atom legend
Atom feed for all
Atom feed for failures
Atom feed for just latest builds

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

- Create node-todo-app freestyle project



The screenshot shows the Jenkins Dashboard with the 'Enter an item name' form. The text 'node-todo-cicd' is entered in the input field. Below the input field, there are four project type options: Freestyle project, Pipeline, Multi-configuration project, and Folder. The 'Freestyle project' option is selected. At the bottom, there is a blue 'OK' button and a link to 'Create Pipeline'.

Enter an item name

node-todo-cicd

» 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.

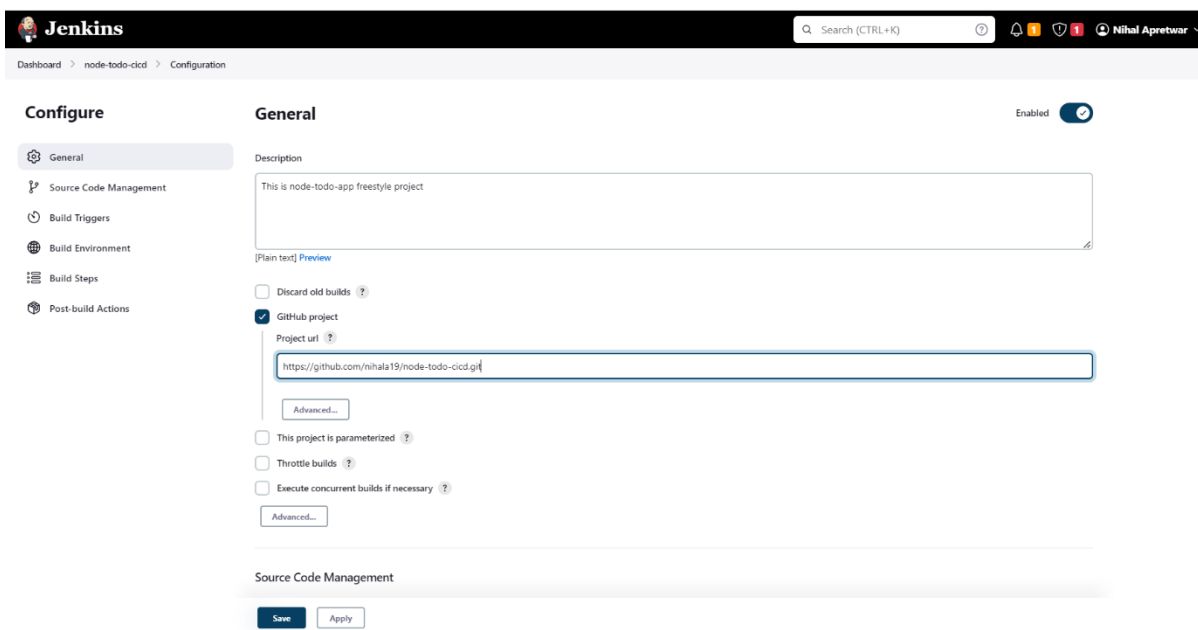
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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 Create Pipeline

- In Configure, GitHub project URL write your project GitHub URL



The screenshot shows the Jenkins Configuration page for the 'node-todo-cicd' project. The 'General' tab is selected. The 'Description' field contains 'This is node-todo-app freestyle project'. The 'Discard old builds' checkbox is unchecked. The 'GitHub project' checkbox is checked. The 'Project url' field contains 'https://github.com/nihala19/node-todo-cicd.git'. The 'Advanced...' button is visible. The 'Source Code Management' section is partially visible at the bottom.

Configure

General

Enabled

Description

This is node-todo-app freestyle project

[Plain text] Preview

☐ Discard old builds

☒ GitHub project

Project url

https://github.com/nihala19/node-todo-cicd.git

Advanced...

☐ This project is parameterized

☐ Throttle builds

☐ Execute concurrent builds if necessary

Advanced...

Source Code Management

Save Apply

- In Git, add credentials for Jenkins

Dashboard > node-todo-cicd > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Source Code Management

None

Git

Repositories

Repository URL

https://github.com/nihala19/node-todo-cicd.git

Credentials

none

+ Add

Jenkins

Advanced Jenkins Credentials Provider

Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/master

Add Branch

Save

Apply

- Click on the 'Build Triggers' tab and then on the 'GitHub hook trigger for GITScm polling'.

Dashboard > node-todo-cicd > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build Triggers

Trigger builds remotely (e.g., from scripts)

Build after other projects are built

Build periodically

GitHub Branches

GitHub Pull Requests

☒
 GitHub hook trigger for GITScm polling

Poll SCM

Build Environment

Delete workspace before build starts

Use secret text(s) or file(s)

Add timestamps to the Console Output

Inspect build log for published build scans

Terminate a build if it's stuck

With Ant

Build Steps

Add build step

Save

Apply

Task 2:

In the Execute shell run the application using Docker compose

Dashboard > node-todo-cicd > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

☐ With Ant ?

Build Steps

Execute shell ?

Command

See the list of available environment variables

```
docker-compose down
docker-compose up --no-deps --build -d
```

Advanced...

Add build step ▾

Post-build Actions

Add post-build action ▾

Save **Apply**

- You will have to make a Docker Compose file for this Project
- After build you can check console output.

Jenkins

Dashboard > node-todo-cicd > #1

Status

</> Changes

Console Output

View as plain text

Edit Build Information

Delete build '#1'

Git Build Data

✓ Console Output

Started by user Nihal Apretwar
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/node-todo-cicd
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/nihala19/node-todo-cicd.git
> git init /var/lib/jenkins/workspace/node-todo-cicd # timeout=10
Fetching upstream changes from https://github.com/nihala19/node-todo-cicd.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/nihala19/node-todo-cicd.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/nihala19/node-todo-cicd.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision b22e8a800b207f542dc455d6b64142b9d0f167c7 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f b22e8a800b207f542dc455d6b64142b9d0f167c7 # timeout=10
Commit message: "Update todo.ejs"
First time build. Skipping changelog.
[node-todo-cicd] \$ /bin/sh -xe /tmp/jenkins12907936104208052216.sh
+ docker-compose down
Removing network node-todo-cicd_default
Network node-todo-cicd_default not found.
+ docker-compose up --no-deps --build -d
Creating network "node-todo-cicd_default" with the default driver
Building web
Sending build context to Docker daemon 513kB

Step 1/7 : FROM node:12.2.0-alpine
--> f391dabf9dce

- Using docker ps command, you can see container is created.

```
ubuntu@ip-172-31-55-25:~/ssh$ cd /var/lib/jenkins/workspace/node-todo-cicd/
ubuntu@ip-172-31-55-25:~/ssh$ cd /var/lib/jenkins/workspace/node-todo-cicd$ sudo docker image
image images
ubuntu@ip-172-31-55-25:~/ssh$ cd /var/lib/jenkins/workspace/node-todo-cicd$ sudo docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
node-todo-cicd_web   latest          7e4a96b2d8cf   4 minutes ago   104MB
django-app-pipeline_web   latest          87e67c9fd9d2   9 hours ago     104MB
<none>               <none>          cb3bc624ccac   9 hours ago     104MB
<none>               <none>          776d03798a83   9 hours ago     104MB
django-app           latest          a2ef211e18b9   34 hours ago    986MB
<none>               <none>          24a14b037330   35 hours ago    986MB
python               3              63490c269128   10 days ago     934MB
trainwithshubham/node-todo-test   latest          05bc821cb8fd   12 days ago     104MB
node                 12.2.0-alpine   f391dabf9dce    3 years ago     77.7MB
ubuntu@ip-172-31-55-25:~/ssh$ cd /var/lib/jenkins/workspace/node-todo-cicd$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  STATUS    CREATED      PORTS                    NAMES
3fe3836267bb   node-todo-cicd_web   "node app.js"           Up 4 minutes    Up 4 minutes    0.0.0.0:8000->8000/tcp, :::8000->8000/tcp   node-todo-cicd_web_1
ubuntu@ip-172-31-55-25:~/ssh$ cd /var/lib/jenkins/workspace/node-todo-cicd$
```

- Browse public IP address with port no.8000

Yeah 🥰🥰🥰🥰🥰🥰🥰 DevOps Community is Super Awesome 😄😄😄😄😄😄

What should I do?

Thank you for reading!

