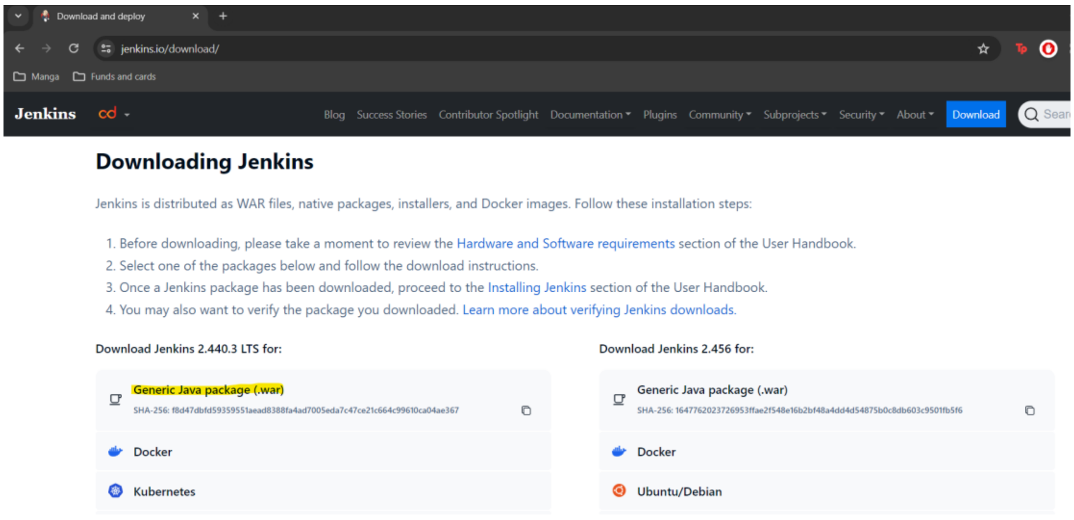
Jenkins

Initial setup for Jenkins:

- Go to URL: https://www.jenkins.io/download/ and download the war file as highlighted below:



Downloading Jenkins

Jenkins is distributed as WAR files, native packages, installers, and Docker images. Follow these installation steps:

1. Before downloading, please take a moment to review the Hardware and Software requirements section of the User Handbook.
2. Select one of the packages below and follow the download instructions.
3. Once a Jenkins package has been downloaded, proceed to the Installing Jenkins section of the User Handbook.
4. You may also want to verify the package you downloaded. Learn more about verifying Jenkins downloads.

Download Jenkins 2.440.3 LTS for:

Generic Java package (.war)
SHA-256: f8d47dbfd59359551aead8388fa4ad7005eda7c47ce21c664c99610ca04ae367

Docker

Kubernetes

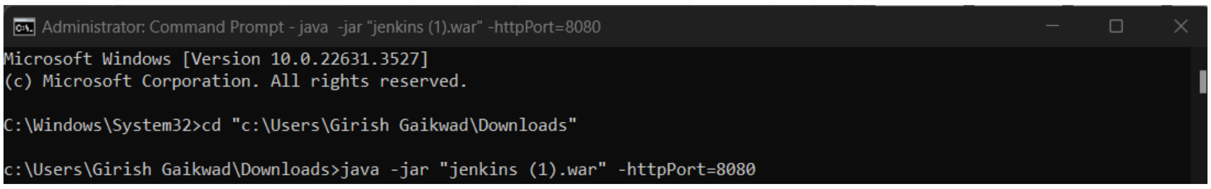
Download Jenkins 2.456 for:

Generic Java package (.war)
SHA-256: 1647762023726953ffae2f548e16b2bf48a4dd4d54875b0c8db603c9501fb5f6

Docker

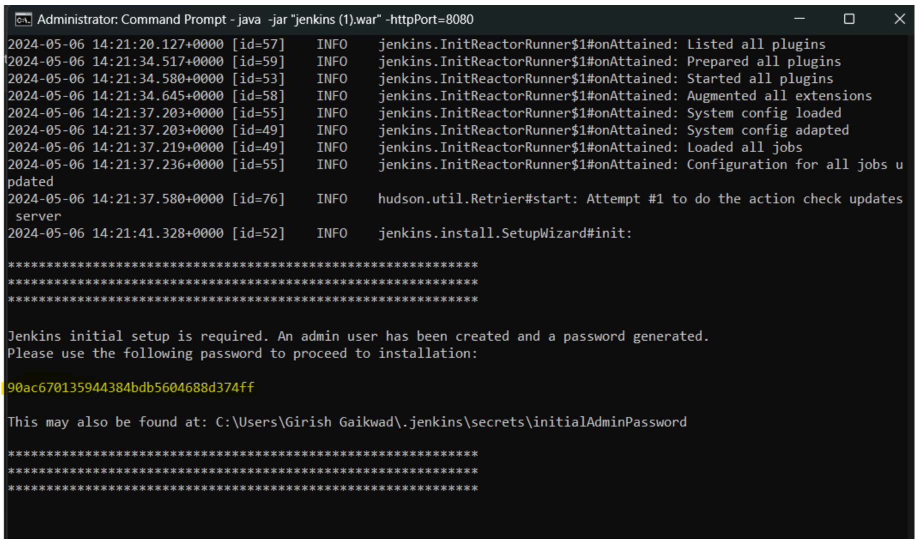
Ubuntu/Debian

- Once file is downloaded, open command prompt as admin and go to the file download location and run command "**java -jar jenkins.war -httpPort=8080**"



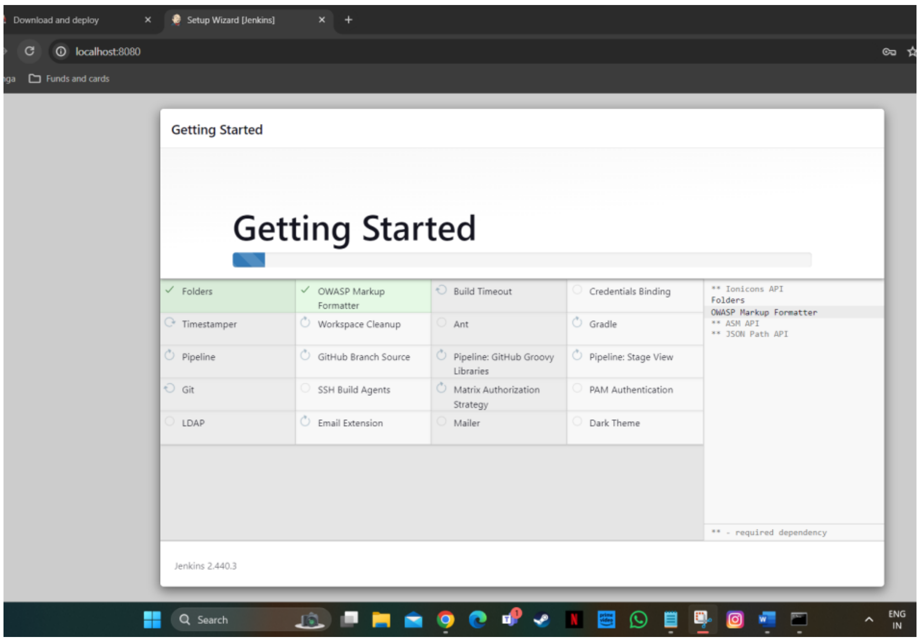
```
Administrator: Command Prompt - java -jar "jenkins (1).war" -httpPort=8080
Microsoft Windows [Version 10.0.22631.3527]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd "c:\Users\Girish Gaikwad\Downloads"

c:\Users\Girish Gaikwad\Downloads>java -jar "jenkins (1).war" -httpPort=8080
```

- Once the command is successful, it will give you one admin password copy it and keep it in notepad.



```
Administrator: Command Prompt - java -jar "jenkins (1).war" -httpPort=8080
2024-05-06 14:21:20.127+0000 [id=57]    INFO    jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2024-05-06 14:21:34.517+0000 [id=59]    INFO    jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2024-05-06 14:21:34.580+0000 [id=53]    INFO    jenkins.InitReactorRunner$1#onAttained: Started all plugins
2024-05-06 14:21:34.645+0000 [id=58]    INFO    jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2024-05-06 14:21:37.203+0000 [id=55]    INFO    jenkins.InitReactorRunner$1#onAttained: System config loaded
2024-05-06 14:21:37.203+0000 [id=49]    INFO    jenkins.InitReactorRunner$1#onAttained: System config adapted
2024-05-06 14:21:37.219+0000 [id=49]    INFO    jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2024-05-06 14:21:37.236+0000 [id=55]    INFO    jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs u
pdated
2024-05-06 14:21:37.580+0000 [id=76]    INFO    hudson.util.Retrier#start: Attempt #1 to do the action check updates
 server
2024-05-06 14:21:41.328+0000 [id=52]    INFO    jenkins.install.SetupWizard#init:

*************************************************************
*************************************************************
*************************************************************

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

90ac670135944384bdb5604688d374ff

This may also be found at: C:\Users\Girish Gaikwad\.jenkins\secrets\initialAdminPassword

*************************************************************
*************************************************************
*************************************************************
```

- To open jenkins, go to browser and hit url http://localhost:8080/, it will ask for admin password which you just copied, put it there and hit continue.
- Now click on "**Install suggested plugins**" option. This step will take some time as it will install multiple things that are needed for running jenkins jobs.



Getting Started

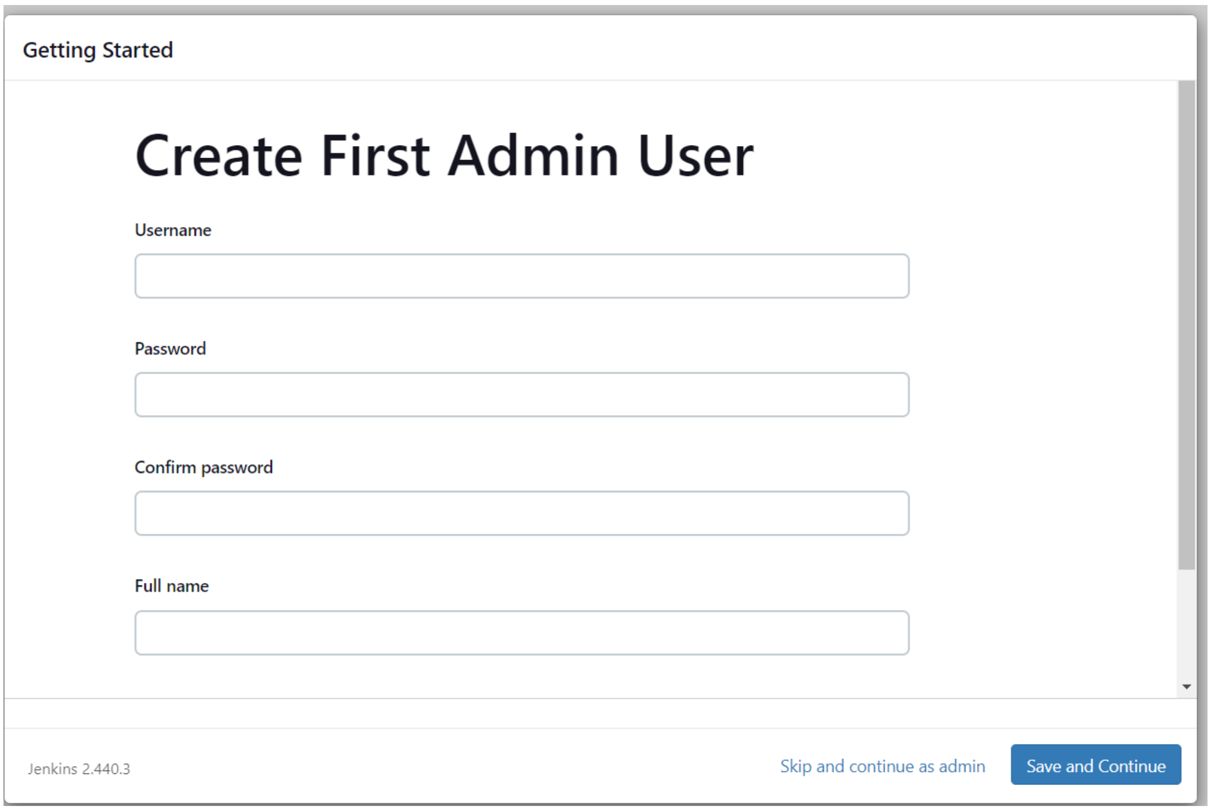
Getting Started

✓ Folders	✓ OWASP Markup Formatter	Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	Dark Theme

```
** Ionicons API
Folders
OWASP Markup Formatter
** ASM API
** JSON Path API

** - required dependency
```

Jenkins 2.440.3

- Once above installation is complete, below page will show up, just fill the details, also scroll so that Email option will show up as well and hit "**Save and continue**".



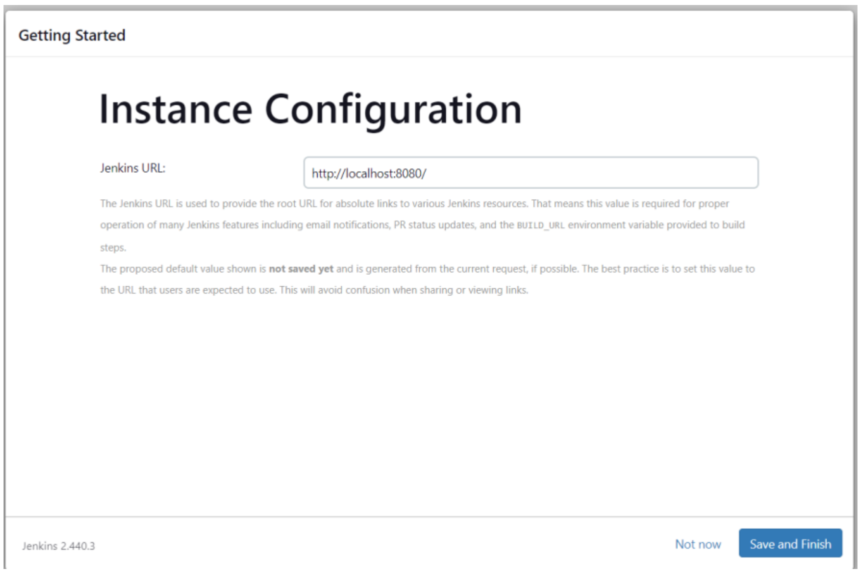
Getting Started

Create First Admin User

Username	
Password	
Confirm password	
Full name	

Jenkins 2.440.3

- Just hit "**Save and Finish**" on below page:



Getting Started

Instance Configuration

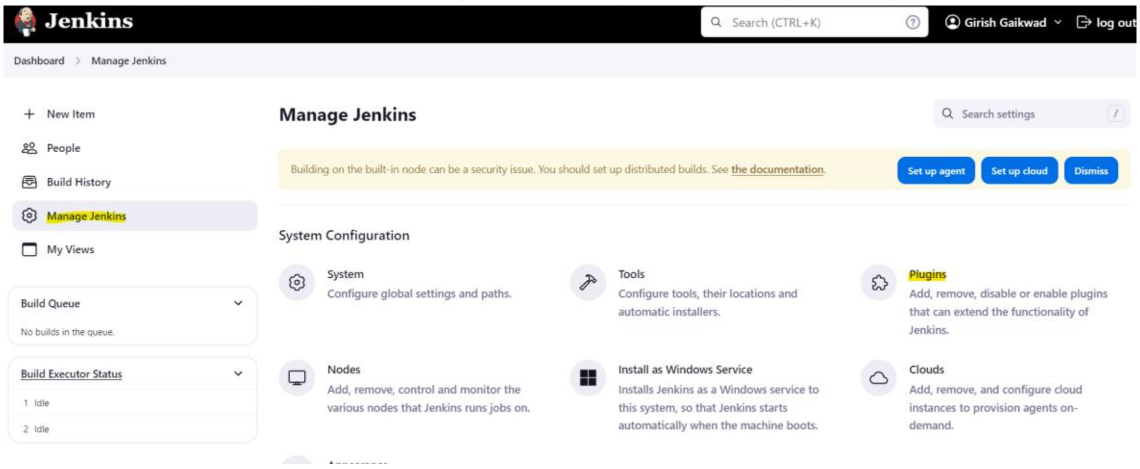
Jenkins URL: http://localhost:8080/

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 2.440.3

- Once everything is done, go to "**Manage Jenkins**" Option. In that go to Plugins.



Manage Jenkins

No builds in the queue.

1 Idle
2 Idle

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.

System Configuration

System — Configure global settings and paths.

Tools — Configure tools, their locations and automatic installers.

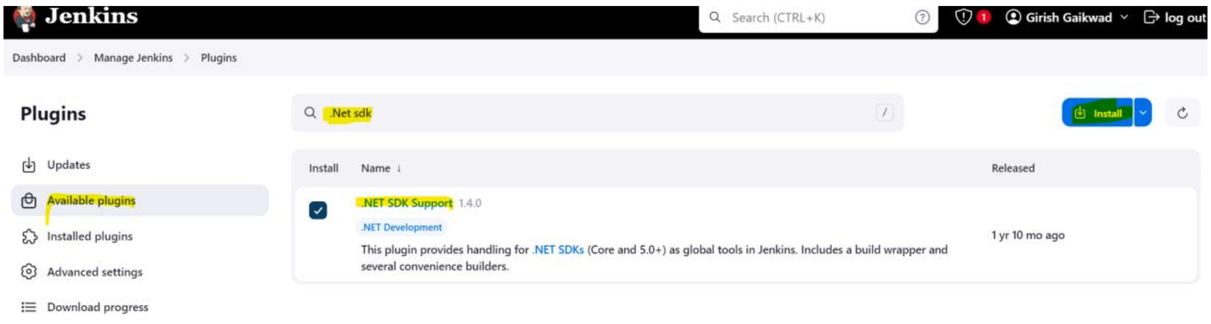
Plugins — Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Nodes — Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Install as Windows Service — Installs Jenkins as a Windows service to this system, so that Jenkins starts automatically when the machine boots.

Clouds — Add, remove, and configure cloud instances to provision agents on-demand.

- Then go to "**Available Plugins**" and search for "**.NET SDK**". Select it and then hit install. This is needed for running c# codes.

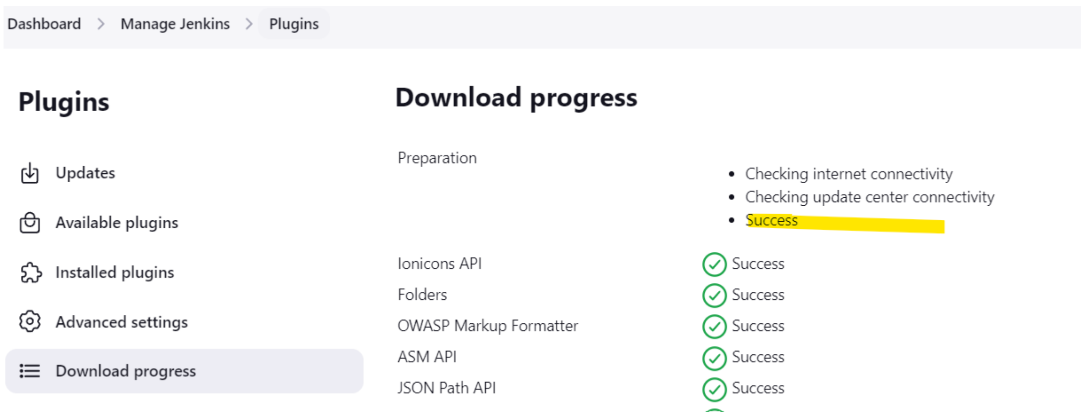


Plugins

.Net sdk

Install	Name ↓	Released
✓	.NET SDK Support 1.4.0 .NET Development This plugin provides handling for .NET SDKs (Core and 5.0+) as global tools in Jenkins. Includes a build wrapper and several convenience builders.	1 yr 10 mo ago

- Once you see success here you can go back to Dashboard (http://localhost:8080/).


Dashboard > Manage Jenkins > Plugins

Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Download progress

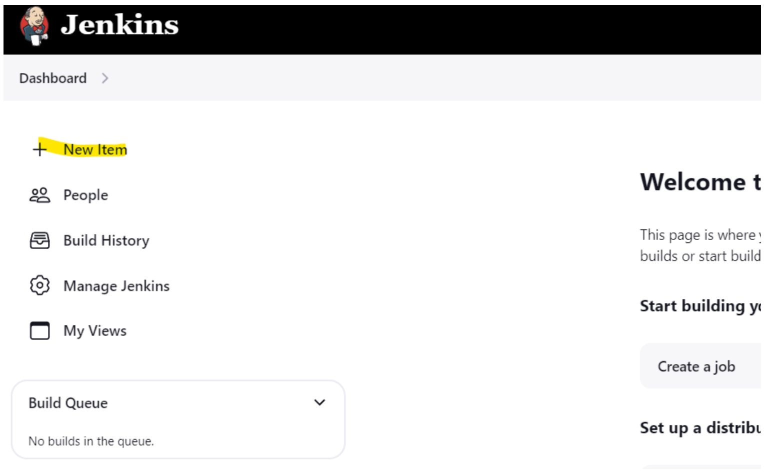
Preparation
- Checking internet connectivity
- Checking update center connectivity
- Success

Ionicons API	✓ Success
Folders	✓ Success
OWASP Markup Formatter	✓ Success
ASM API	✓ Success
JSON Path API	✓ Success

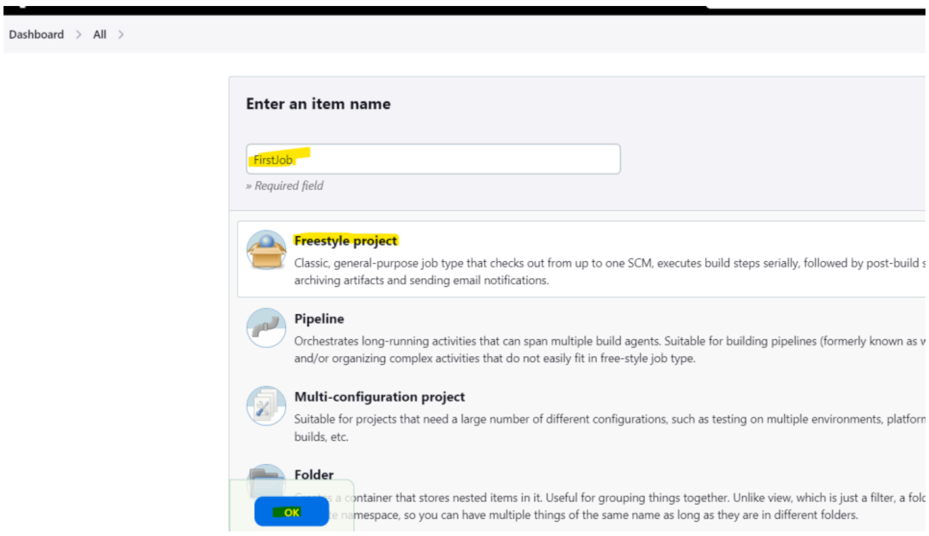
That's it, this is the Jenkins setup for running C# test automation codes.

Running Jenkins using local code path:

- Select the project that you want to run in Jenkins and go to its folder location.
- On Jenkins, select "**New Item**"



- Give any name to the item and select as per below, we will do FreeStyle project and this is commonly used for C# codes. Click Ok.


Dashboard > All >

Enter an item name

FirstJob

» Required field

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

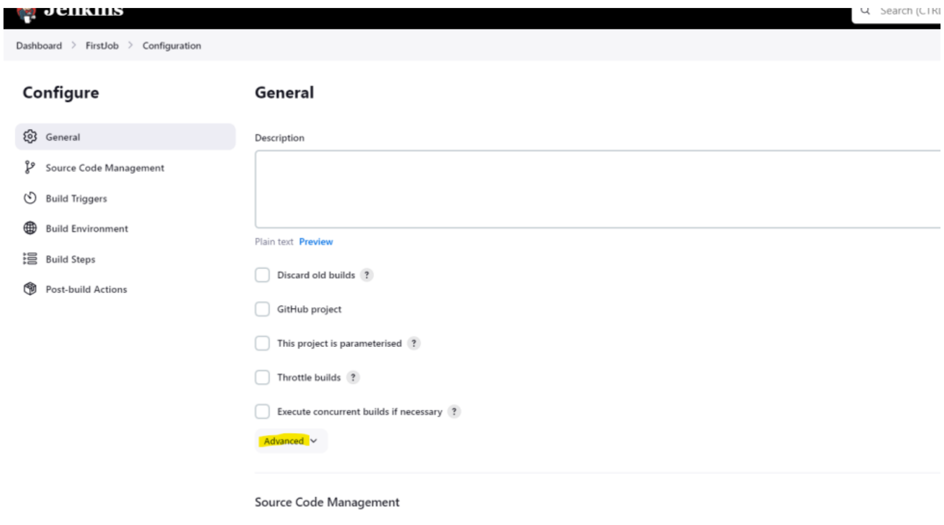
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as w
and/or organizing complex activities that do not easily fit in free-style job type.

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

Folder
ntainer that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folc
mespace, so you can have multiple things of the same name as long as they are in different folders.

OK

- Now expand the Advanced button. Select Use Custom workspace and give path of the project that you want to run.


Dashboard > FirstJob > Configuration

Configure

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

General

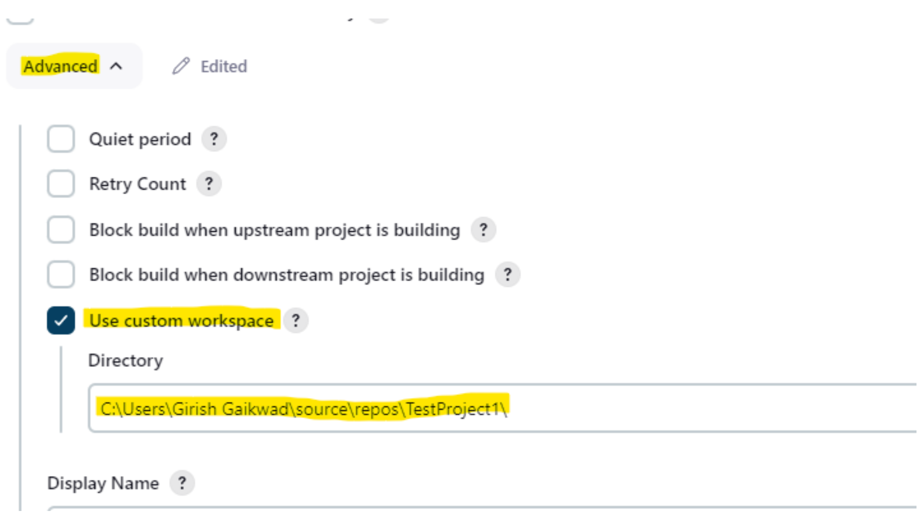
Description

Plain text Preview

- [ ] Discard old builds
- [ ] GitHub project
- [ ] This project is parameterised
- [ ] Throttle builds
- [ ] Execute concurrent builds if necessary

Advanced

Source Code Management


Advanced ^ Edited

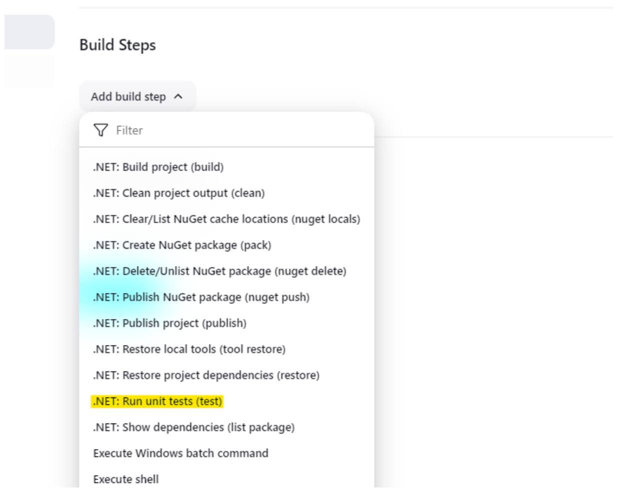
- [ ] Quiet period
- [ ] Retry Count
- [ ] Block build when upstream project is building
- [ ] Block build when downstream project is building
- [x] Use custom workspace

Directory

C:\Users\Girish Gaikwad\source\repos\TestProject1\

Display Name

- Now scroll down and you will see “**Build Steps**” and there will be multiple options, we can choose as per our need. Here we will select “**.NET run unit tests**”. You can keep the settings default here and click on Apply.

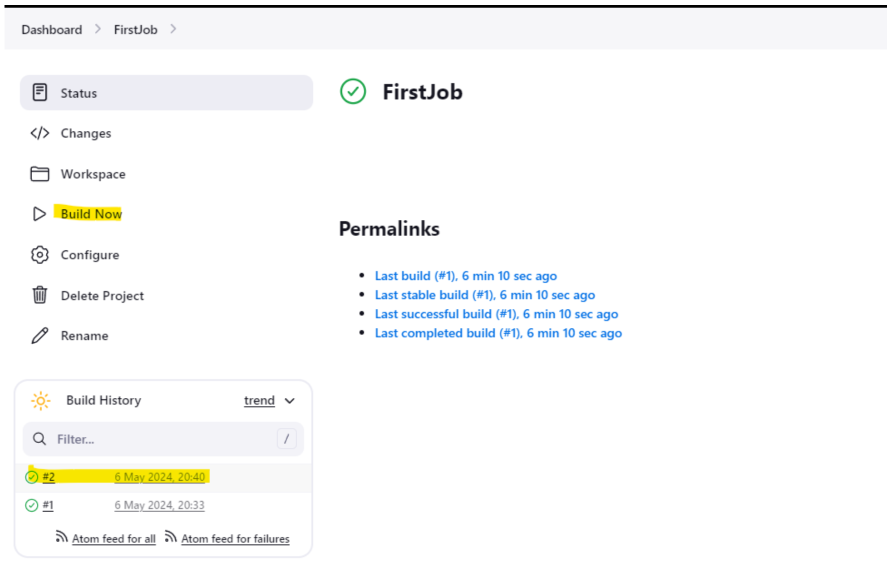

Build Steps

Add build step

Filter

.NET: Build project (build)
.NET: Clean project output (clean)
.NET: Clear/List NuGet cache locations (nuget locals)
.NET: Create NuGet package (pack)
.NET: Delete/Unlist NuGet package (nuget delete)
.NET: Publish NuGet package (nuget push)
.NET: Publish project (publish)
.NET: Restore local tools (tool restore)
.NET: Restore project dependencies (restore)
.NET: Run unit tests (test)
.NET: Show dependencies (list package)
Execute Windows batch command
Execute shell

- Now click here and you will see options like below, in that select "**Build Now**", and you will see build is triggered and if it's a selenium test cases then browser will get opened and you will start seeing the execution.


Dashboard > FirstJob >

FirstJob

Status
Changes
Workspace
Build Now
Configure
Delete Project
Rename

Permalinks

- Last build (#1), 6 min 10 sec ago
- Last stable build (#1), 6 min 10 sec ago
- Last successful build (#1), 6 min 10 sec ago
- Last completed build (#1), 6 min 10 sec ago

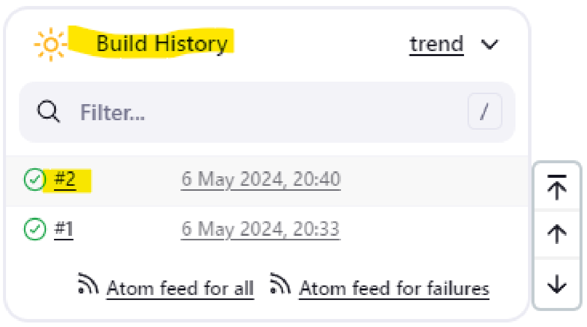
Build History trend

Filter...

#2	6 May 2024, 20:40
#1	6 May 2024, 20:33

Atom feed for all Atom feed for failures

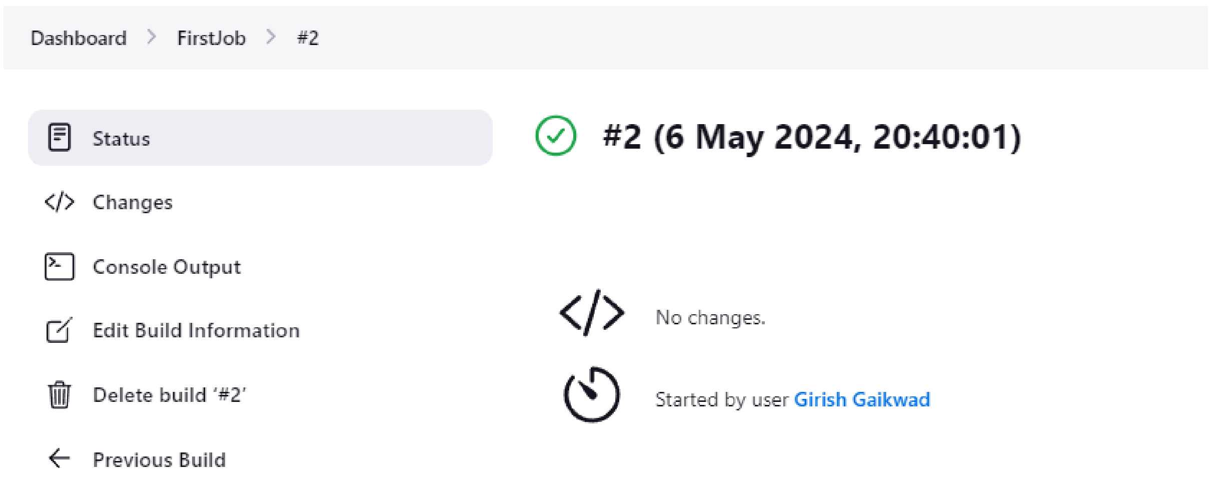
- If you want to see the build details and everything, you can click on the build number in build history and you will see options like below through which you can go through.


Build History trend

Filter...

#2	6 May 2024, 20:40
#1	6 May 2024, 20:33

Atom feed for all Atom feed for failures


Dashboard > FirstJob > #2

Status

Changes

Console Output

Edit Build Information

Delete build '#2'

Previous Build

#2 (6 May 2024, 20:40:01)

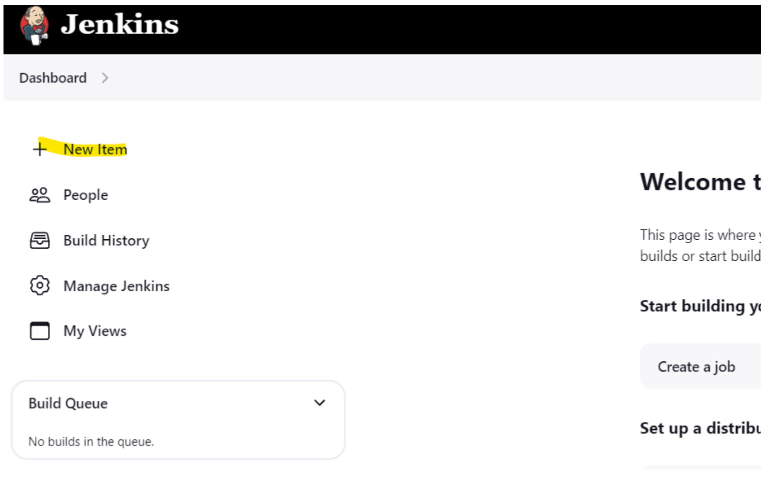
No changes.

Started by user Girish Gaikwad

This completes our first job with using local project.

Jenkins using GitHub project

- Select the project that you want to run in Jenkins and go to its folder location.
- On Jenkins, select "**New Item**"



No builds in the queue.

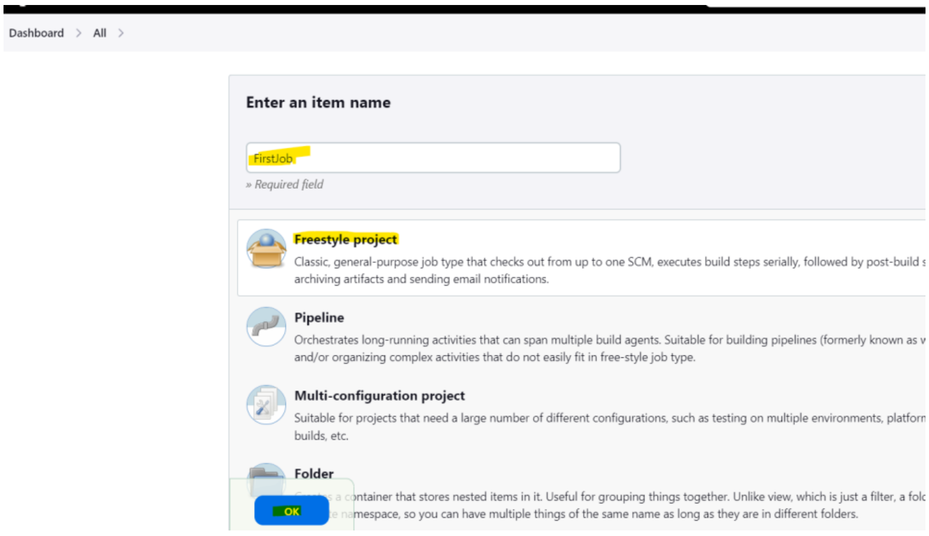
Welcome t

This page is where y
builds or start build

Start building yo

Set up a distribu

- Give any name to the item and select as per below, we will do FreeStyle project and this is commonly used for C# codes. Click Ok.



Enter an item name

FirstJob

» Required field

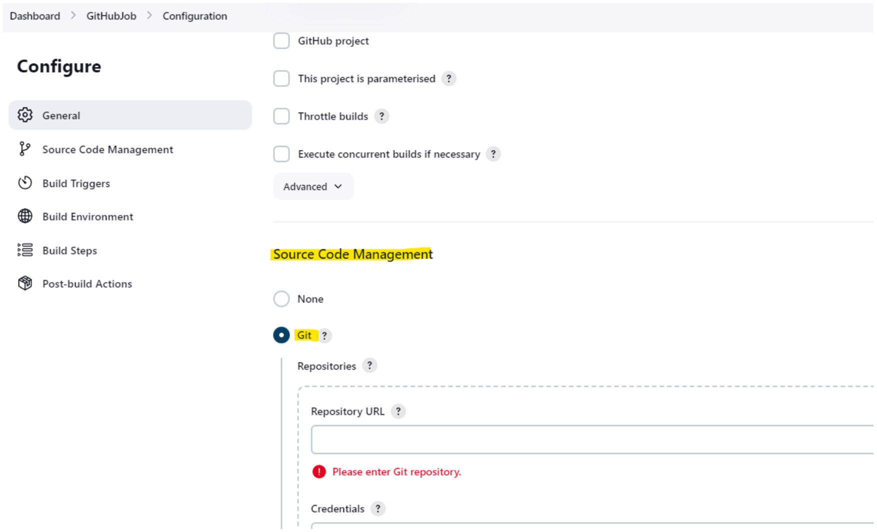
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build s
archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as w
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
builds, etc.

Folder
ntainer that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a fold
mespace, so you can have multiple things of the same name as long as they are in different folders.

- Here, slight change will be that we will select "**Git**" option in "**Source Code Management**".



Dashboard > GitHubJob > Configuration

Configure

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

☐ GitHub project
☐ This project is parameterised ?
☐ Throttle builds ?
☐ Execute concurrent builds if necessary ?

Advanced

Source Code Management

○ None
● Git ?

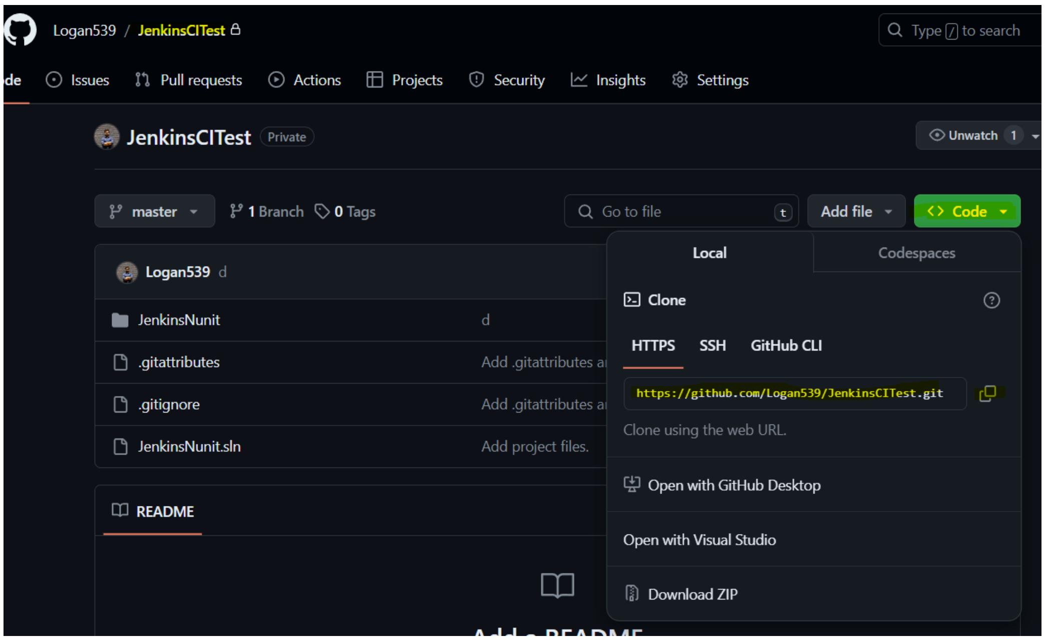
Repositories ?

Repository URL ?

Please enter Git repository.

Credentials ?

- To get Git repo url, go to your github and select the project repository that you want to run through Jenkins. Go into that repo and select code and copy the https path as shown below. Add that path in your jenkins job.



Logan539 / JenkinsCITest

Issues · Pull requests · Actions · Projects · Security · Insights · Settings

JenkinsCITest Private

master · 1 Branch · 0 Tags · Go to file · Add file · Code

Logan539 d	
JenkinsNunit	d
.gitattributes	Add .gitattributes a
.gitignore	Add .gitattributes a
JenkinsNunit.sln	Add project files.

README

Local · Codespaces

Clone

HTTPS · SSH · GitHub CLI

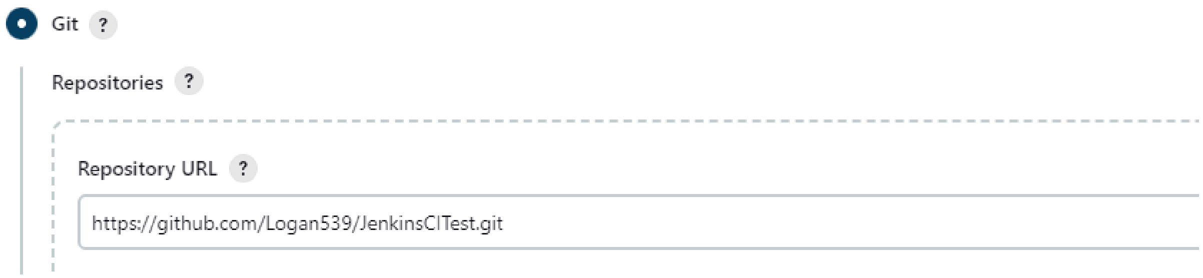
```
https://github.com/Logan539/JenkinsCITest.git
```

Clone using the web URL.

Open with GitHub Desktop

Open with Visual Studio

Download ZIP



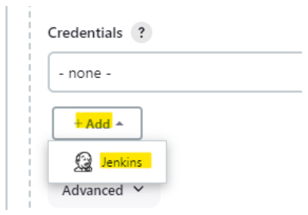
● Git ?

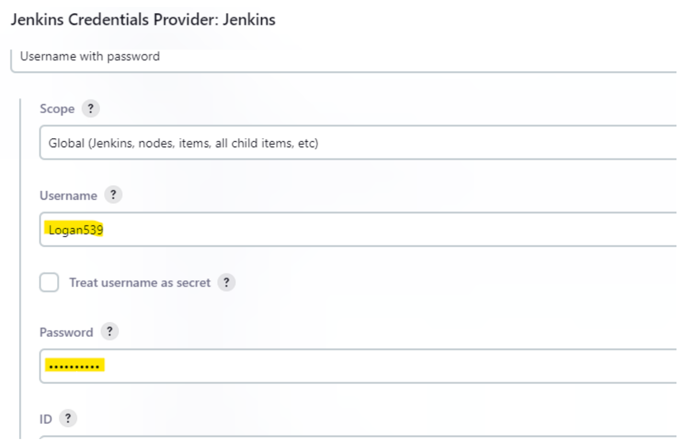
Repositories ?

Repository URL ?

https://github.com/Logan539/JenkinsCITest.git

- Now we need to add credentials of your github in jenkins so select Add then Jenkins. Add Username and password and hit Add.




Jenkins Credentials Provider: Jenkins

Username with password

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

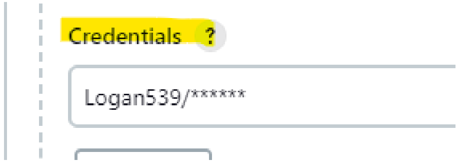
Username: Logan539

Treat username as secret

Password: ••••••••••

ID

- Once you have added Github creds, you will see the username in the dropdown in Credentials, select it.


Credentials

Logan539/******

- No need to change anything else, just select the build steps as how we did for local project path and hit apply.
- Then follow the steps of local build path and you will be able to run a jenkins job for github project.

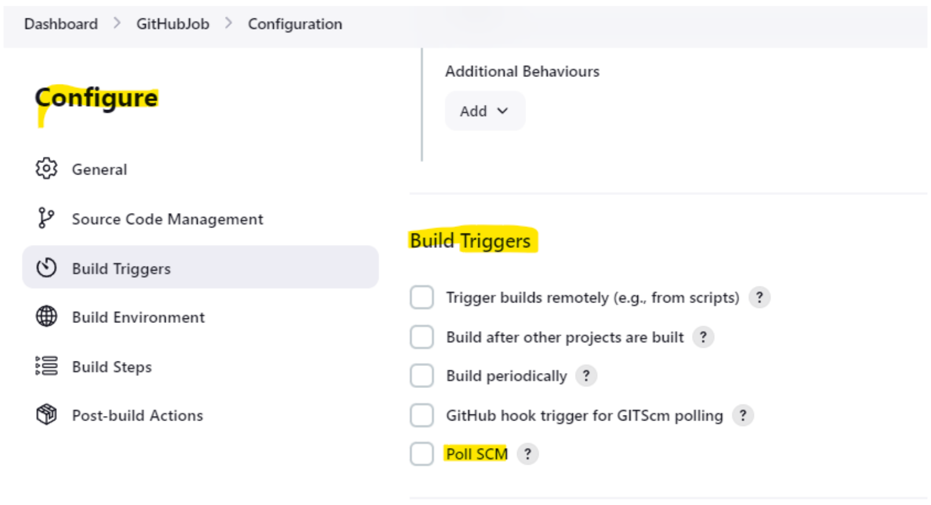
That's it for running github project through jenkins.

If you want to make any changes to you job you can select Configure option and do the changes. There are multiple options you can try but I have added the ones that we will use for C#.

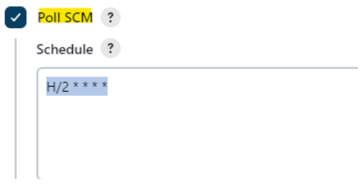
Now lets look at how we can schedule our jobs to run. First we will see a fixed interval and second we will see build getting triggered every time we push any changes to github.

Running Jenkins on fixed interval:

- Go to any existing job in Jenkins, select Configure option and Go to Build triggers. In this select "**Poll SCM**" as that's what we will use.



- You can put value as "H/2 * * * *" and this will run the build every two minutes. You can try multiple combinations here which you can find on internet.



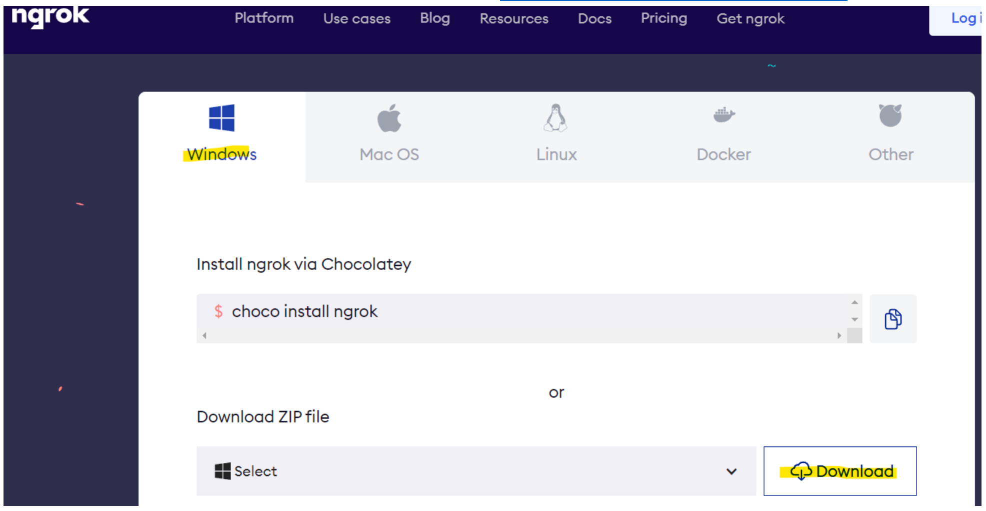
Now just click and save and your job will run every two minutes.

Running Jenkins for every Github push:

This is bit different as for this we will need a public IP which we don't have for our local setup, in ideal scenario you will have a public ip or a CMG type gateway which you can use to run this.

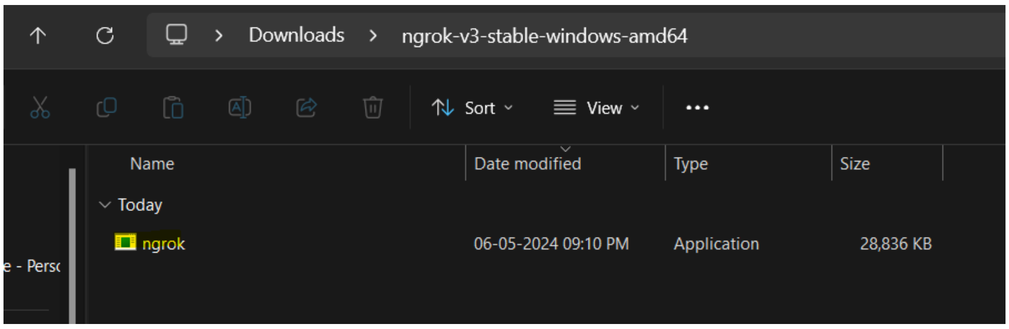
Just to show how we can achieve this, I have added a way to create a public IP and use it to run a Jenkins job for every github push.

- For this Download the files from this link: https://ngrok.com/download

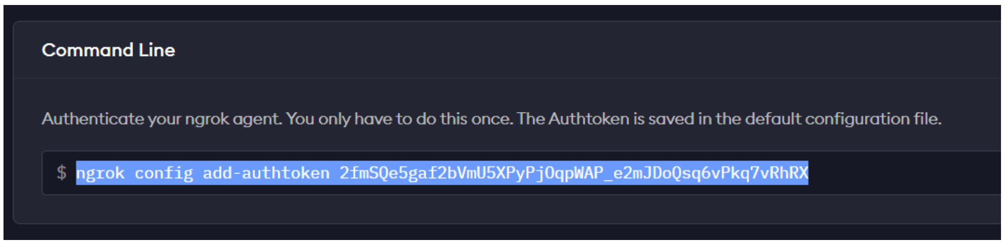


```
$ choco install ngrok
```

- Once file downloaded, extract the files and click on the application



- You will see a command prompt open up. Here first we need to add authentication token of ngrok from this url: https://dashboard.ngrok.com/get-started/your-authtoken (you might need to sign up for this if not done already) Use this command and run it.

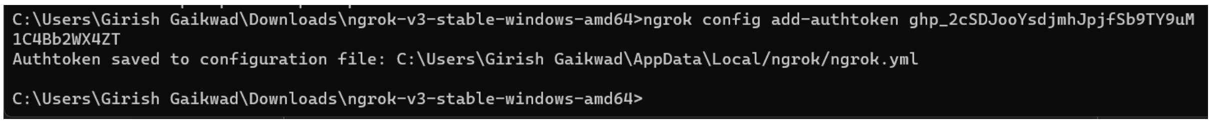


Command Line

Authenticate your ngrok agent. You only have to do this once. The Authtoken is saved in the default configuration file.

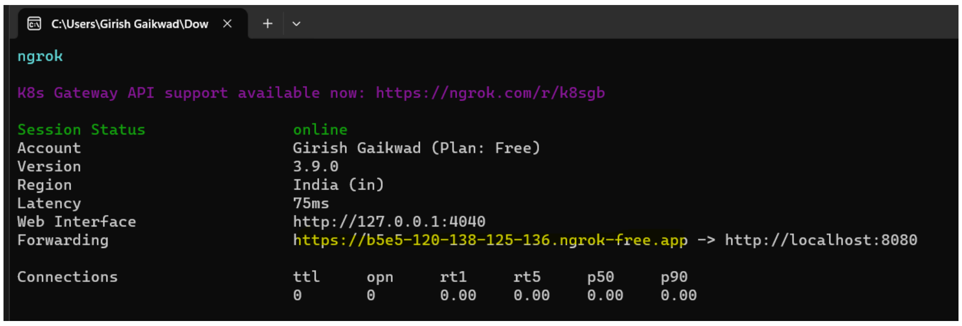
```
$ ngrok config add-authtoken 2fmSQe5gaf2bVmU5XPyPjOqpWAP_e2mJDoQsq6vPkq7vRhRX
```

Output should look like below.



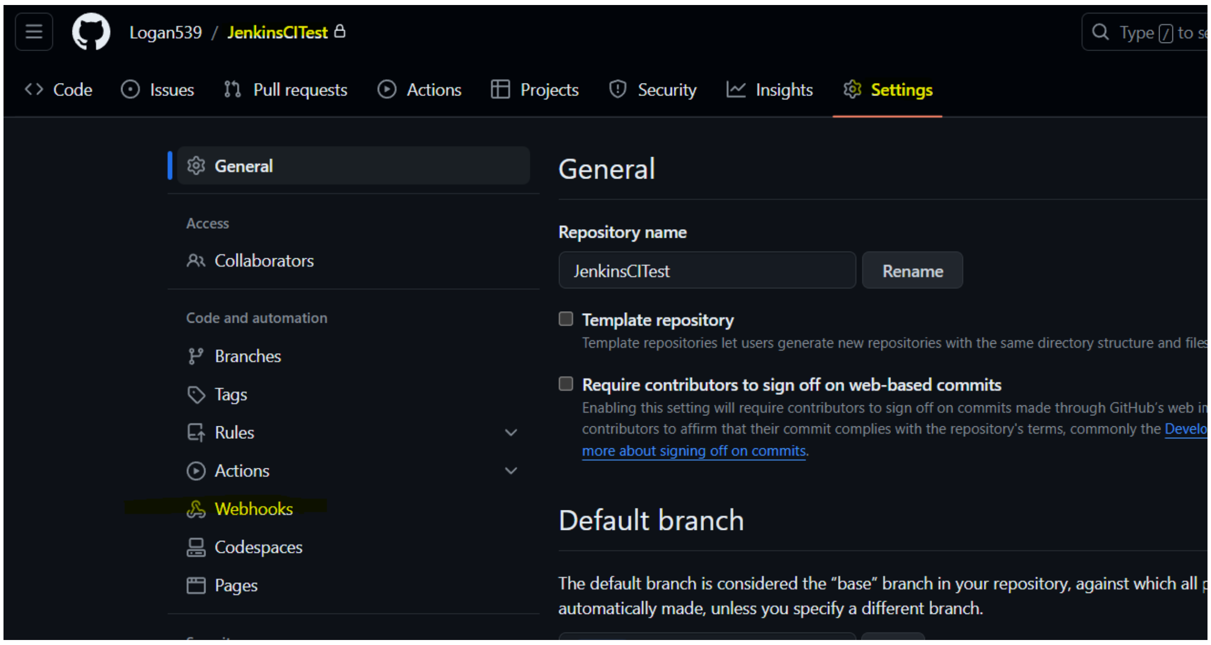
```
C:\Users\Girish Gaikwad\Downloads\ngrok-v3-stable-windows-amd64>ngrok config add-authtoken ghp_2cSDJooYsdjmhJpjfSb9TY9uM
1C4Bb2WX4ZT
Authtoken saved to configuration file: C:\Users\Girish Gaikwad\AppData\Local/ngrok/ngrok.yml

C:\Users\Girish Gaikwad\Downloads\ngrok-v3-stable-windows-amd64>
```

- Now run **ngrok http 8080** command. And you should get a public IP for your localhost as below. Copy the highlighted URL.



```
ngrok

K8s Gateway API support available now: https://ngrok.com/r/k8sgb

Session Status                online
Account                       Girish Gaikwad (Plan: Free)
Version                       3.9.0
Region                        India (in)
Latency                       75ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://b5e5-120-138-125-136.ngrok-free.app -> http://localhost:8080

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

- Now go to Github where your project repo is, in that go to its settings and select Webhooks.


Logan539 / JenkinsCITest

Code Issues Pull requests Actions Projects Security Insights Settings

General

Access
Collaborators

Code and automation
Branches
Tags
Rules
Actions
Webhooks
Codespaces
Pages

General

Repository name

JenkinsCITest

Rename

Template repository

Template repositories let users generate new repositories with the same directory structure and files

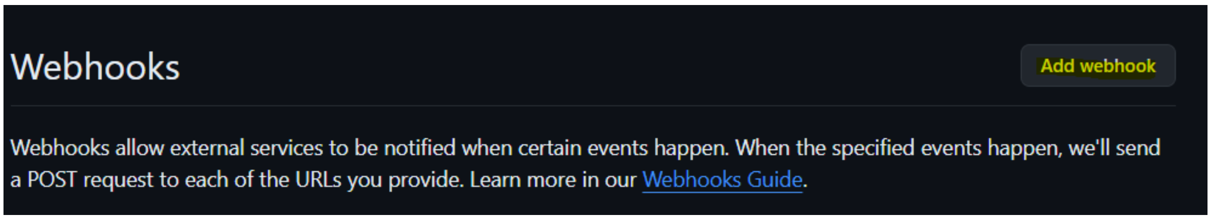
Require contributors to sign off on web-based commits

Enabling this setting will require contributors to sign off on commits made through GitHub's web in contributors to affirm that their commit complies with the repository's terms, commonly the Develo more about signing off on commits.

Default branch

The default branch is considered the "base" branch in your repository, against which all p automatically made, unless you specify a different branch.

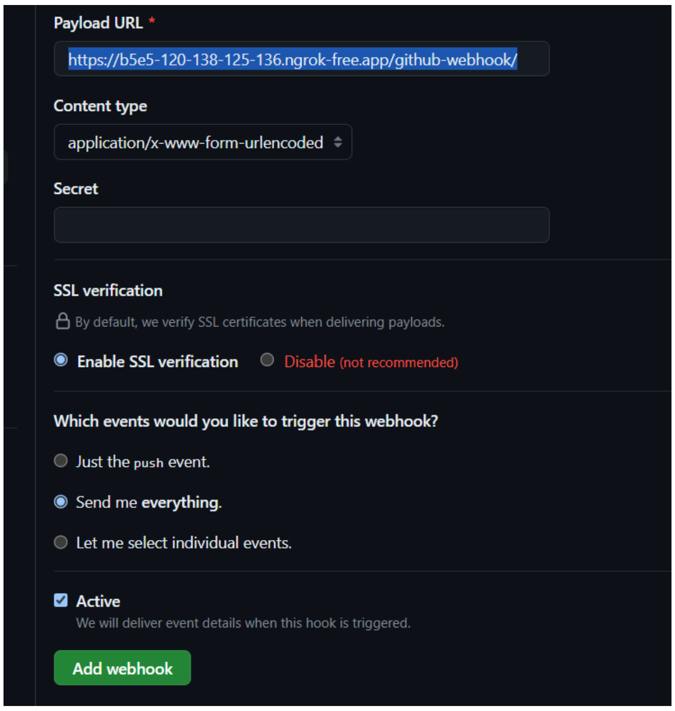
- Select Add Webhook.


Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

- Add the Payload Url as the url which was created with ngrok and append github-webhook to it as below:
  https://b5e5-120-138-125-136.ngrok-free.app/github-webhook/
- Keep the options as per screenshot and hit add webhook


Payload URL *

https://b5e5-120-138-125-136.ngrok-free.app/github-webhook/

Content type

application/x-www-form-urlencoded

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

[✓ Enable SSL verification] Disable (not recommended)

Which events would you like to trigger this webhook?

Just the push event.

[✓ Send me everything.]

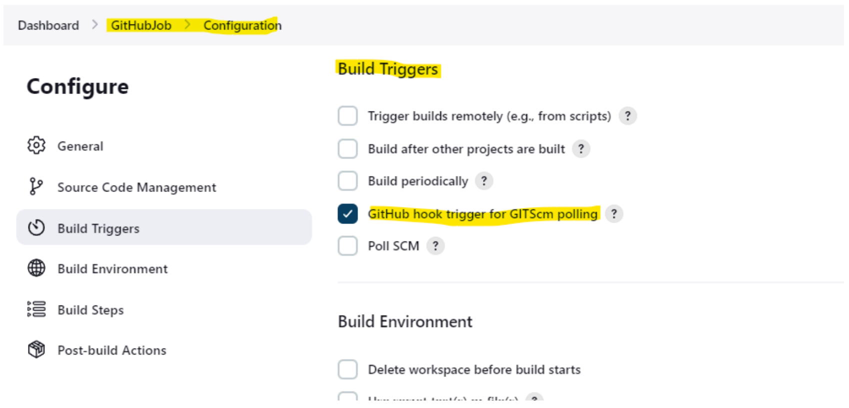
Let me select individual events.

[✓ Active]

We will deliver event details when this hook is triggered.

Add webhook

- Now go to the Jenkins and open the exisiting job, in that go to Configure and select below option as per screenshot in Build triggers and hit apply.


Dashboard > GitHubJob > Configuration

Configure

Build Triggers

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

Build Environment

- [ ] Delete workspace before build starts

Once this is done, any push you will do the github will trigger the build and you will be able to see the build running.