EXPERIMENT 4

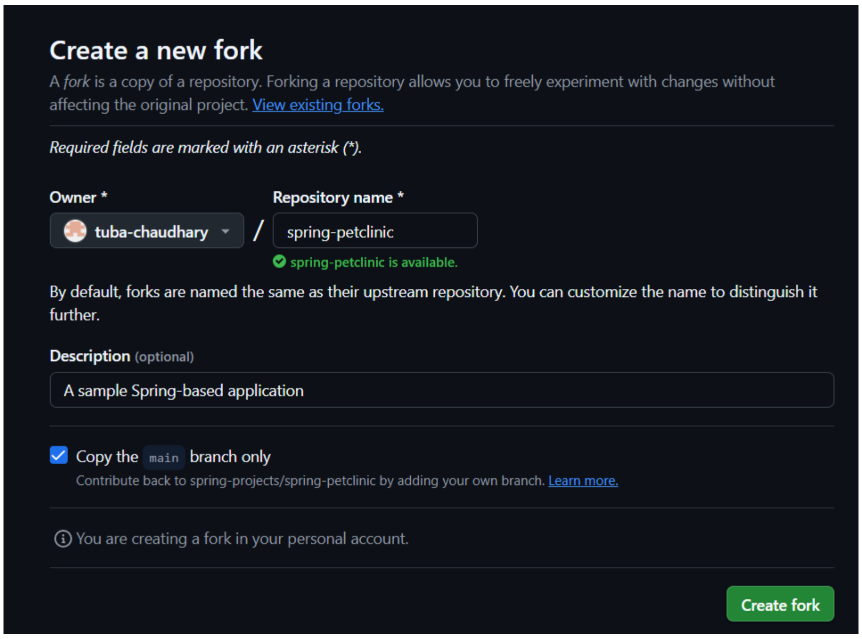
231410
TE IT

AIM : To integrate a GitHub repository with Jenkins and implement a scripted pipeline that automatically builds and tests the project.

STEPS :

1. Fork the Repository and Login to Jenkins.

Forking the spring-petclinic repository into your GitHub account.



Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. View existing forks.

Required fields are marked with an asterisk (*).

Owner *: tuba-chaudhary / Repository name *: spring-petclinic

spring-petclinic is available.

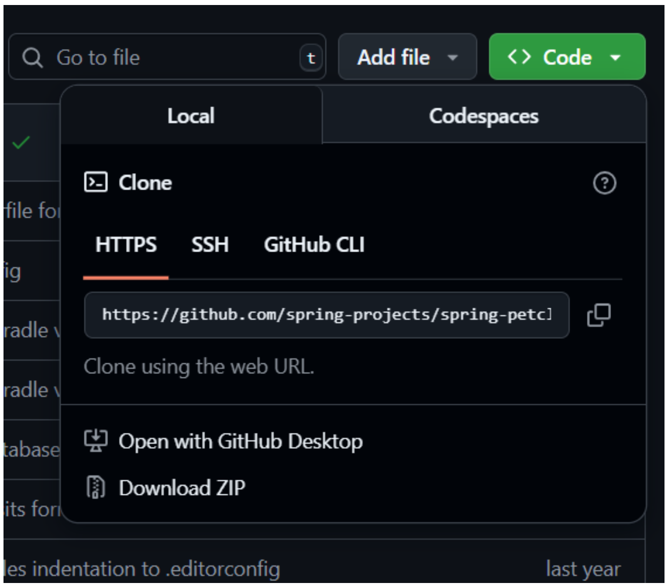
By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional): A sample Spring-based application

[✓] Copy the `main` branch only
Contribute back to spring-projects/spring-petclinic by adding your own branch. Learn more.

You are creating a fork in your personal account.

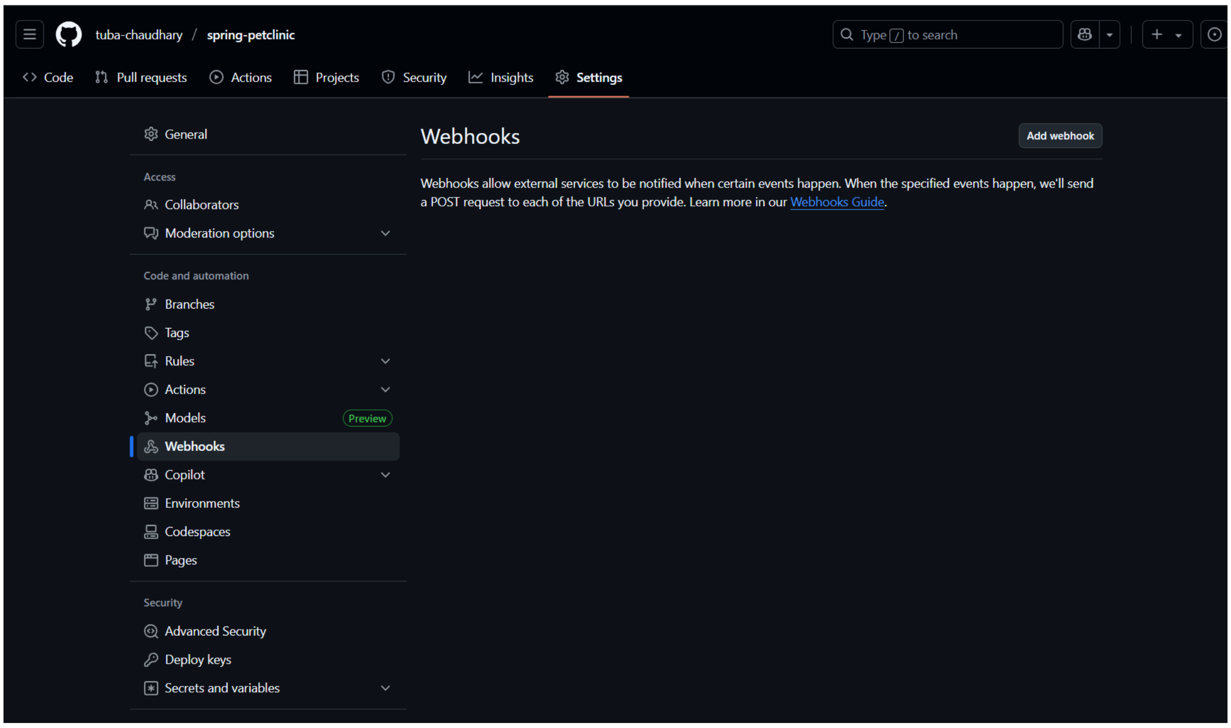
Click the **Code** button on GitHub and **copied the HTTPS URL** of the Spring PetClinic repository to clone it locally.



```
https://github.com/spring-projects/spring-petcl
```

Clone using the web URL.

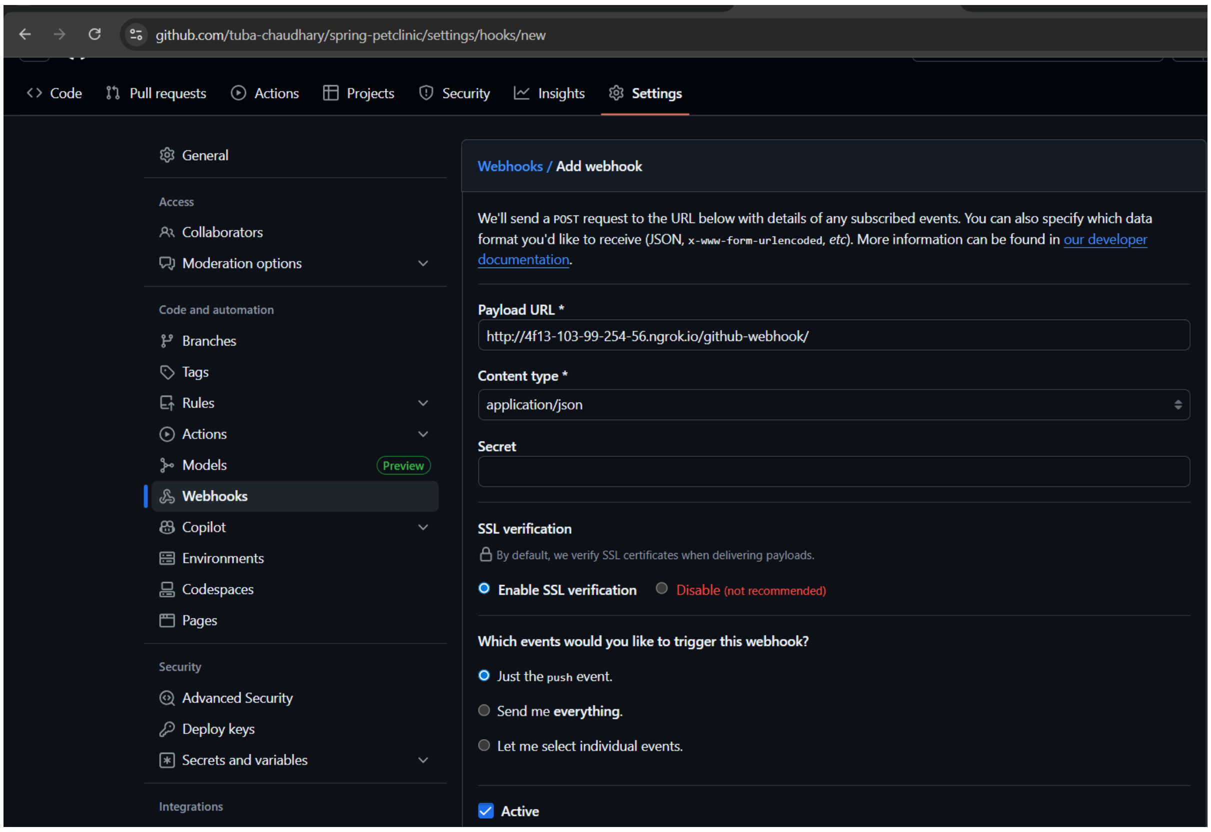
2. Opening **Webhooks settings** in the GitHub repository to add a webhook.



Webhooks

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.

Adding a **new webhook** in GitHub using your Ngrok public URL + /GitHub



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 *
http://4f13-103-99-254-56.ngrok.io/github-webhook/

Content type *
application/json

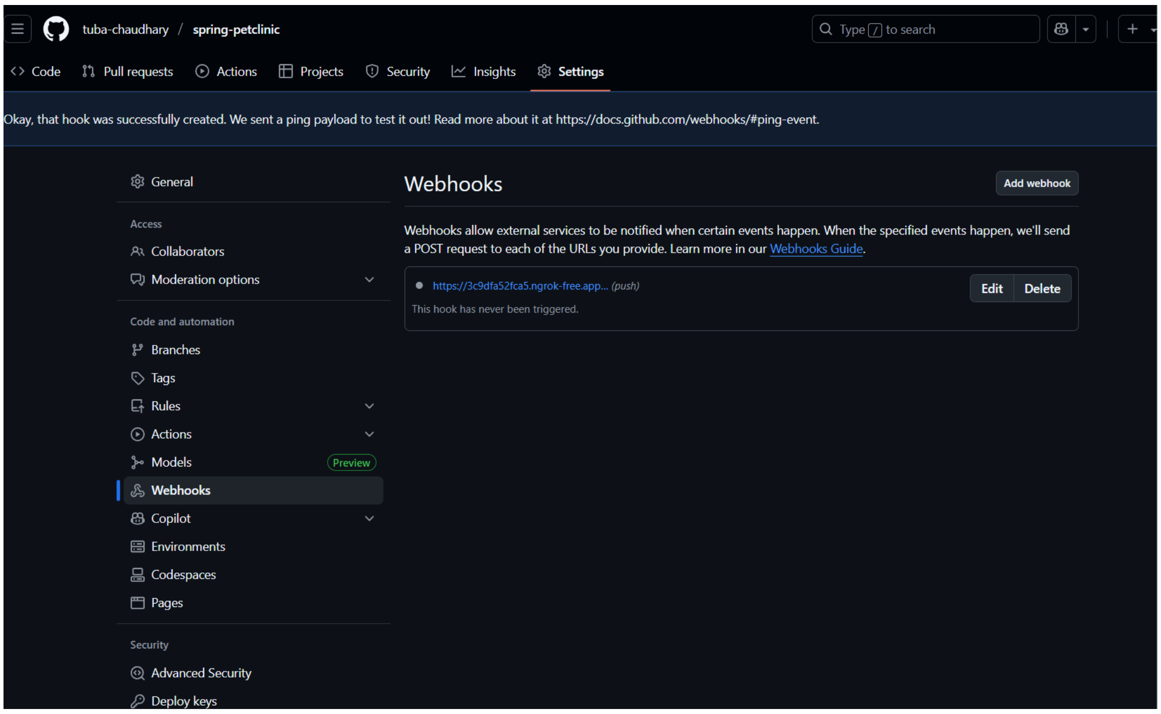
Secret

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]

A **GitHub webhook** has been successfully added, pointing to the **Ngrok public URL**, which will trigger Jenkins builds whenever changes are pushed to the repository.



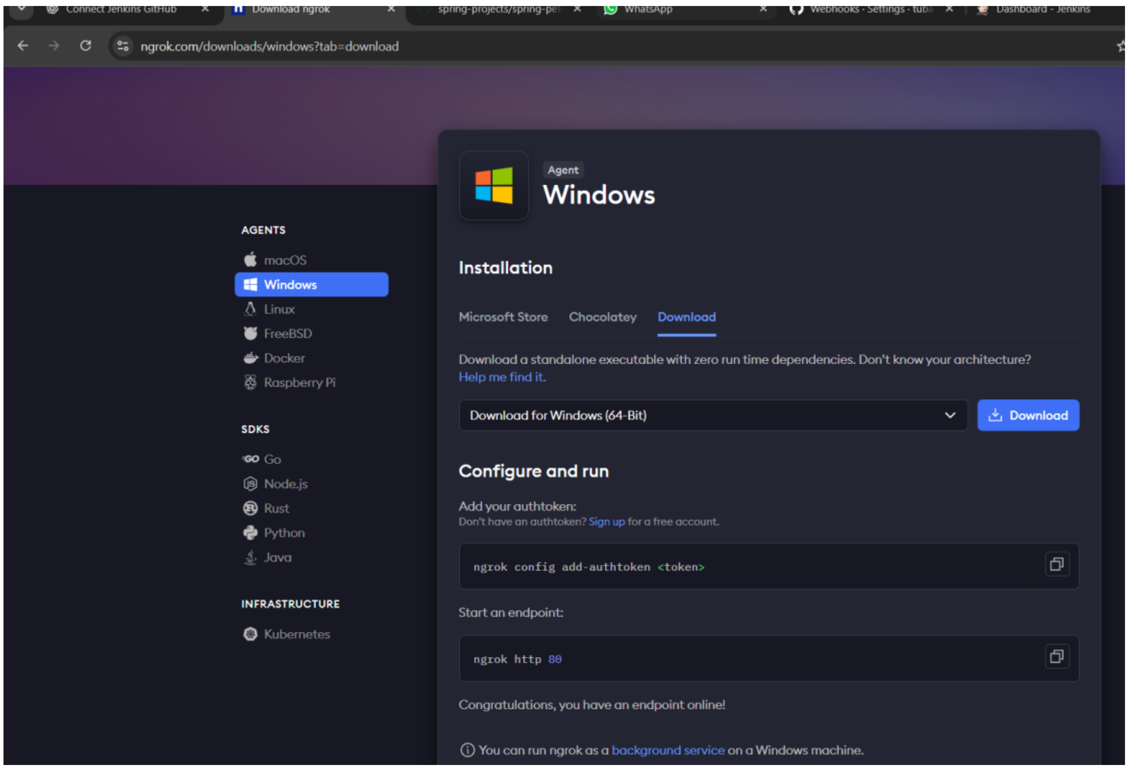
Okay, that hook was successfully created. We sent a ping payload to test it out! Read more about it at https://docs.github.com/webhooks/#ping-event.

Webhooks

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.

https://3c9dfa52fca5.ngrok-free.app... (push)
This hook has never been triggered.

3 . Downloading **ngrok** for Windows to enable external access to Jenkins.



Installation

Download a standalone executable with zero run time dependencies. Don't know your architecture? Help me find it.

Configure and run

Add your authtoken:
Don't have an authtoken? Sign up for a free account.

```
ngrok config add-authtoken <token>
```

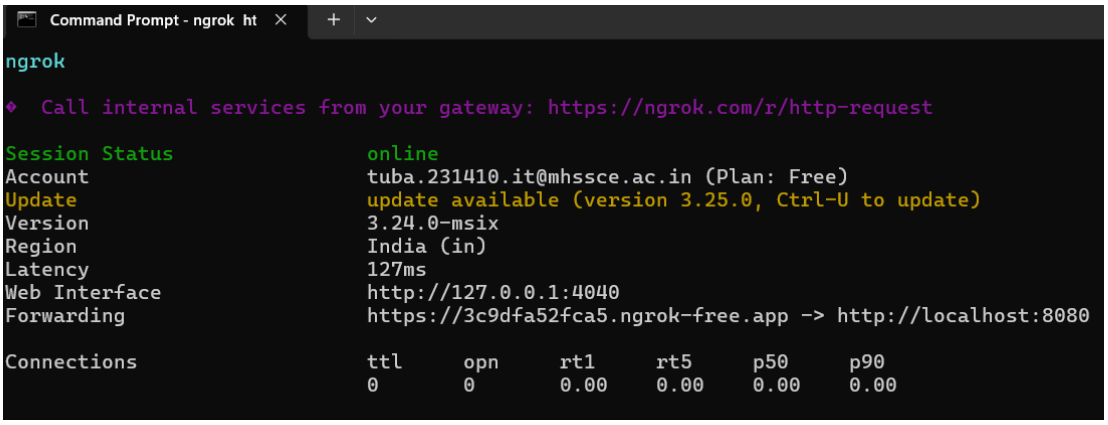
Start an endpoint:

```
ngrok http 80
```

Congratulations, you have an endpoint online!

You can run ngrok as a background service on a Windows machine.

Ngrok is **running and forwarding localhost:8080** (Jenkins) to a **public URL** so GitHub webhooks can reach your local Jenkins server.



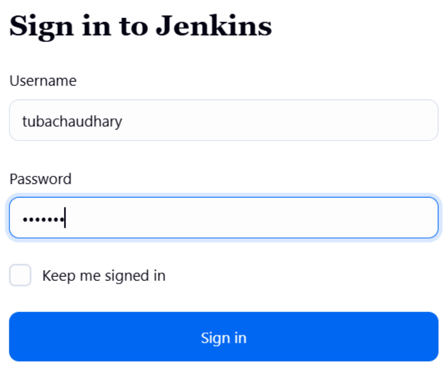
```
ngrok

◆  Call internal services from your gateway: https://ngrok.com/r/http-request

Session Status                online
Account                       tuba.231410.it@mhssce.ac.in (Plan: Free)
Update                        update available (version 3.25.0, Ctrl-U to update)
Version                       3.24.0-msix
Region                        India (in)
Latency                       127ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://3c9dfa52fca5.ngrok-free.app -> http://localhost:8080

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

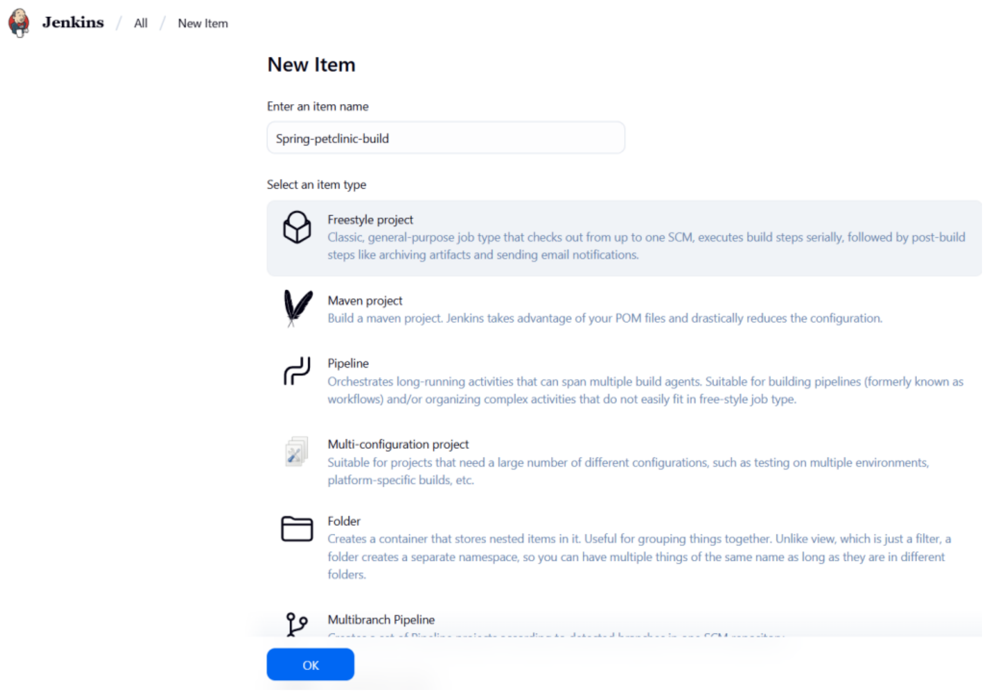
4. Logging in to Jenkins using your credentials at http://localhost:8080.


Sign in to Jenkins

Username: tubachaudhary

Password: •••••••

A **new Jenkins job** named Spring-petclinic-build is being created, and the **project type** (Freestyle) is being selected to configure the build.



New Item

Enter an item name: Spring-petclinic-build

Select an item type

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

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

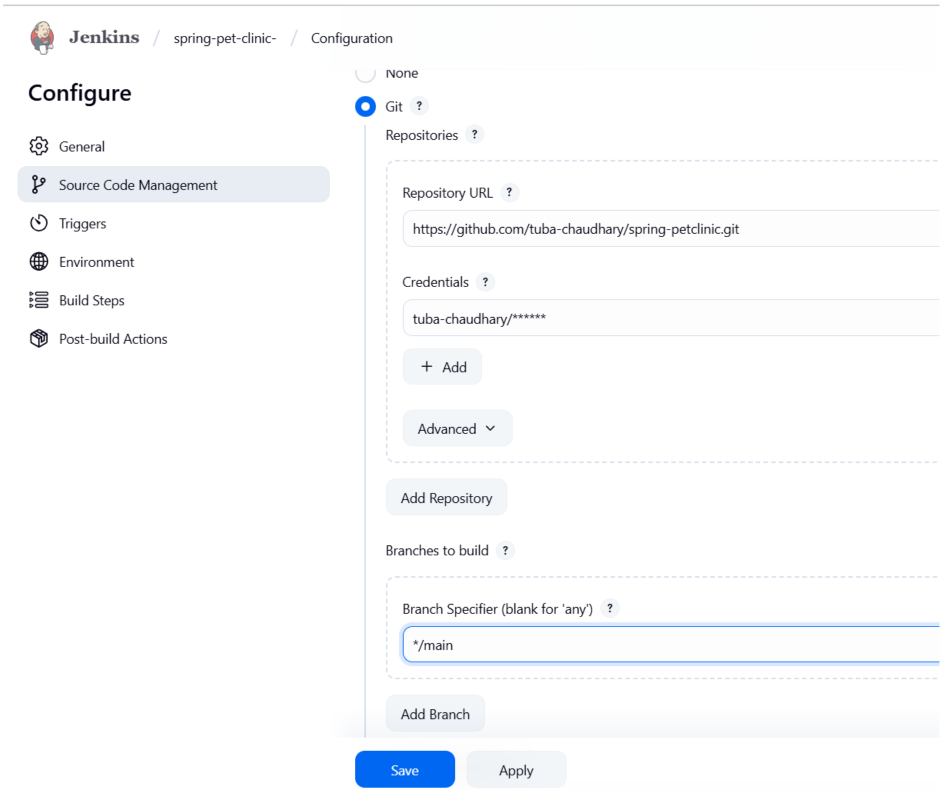
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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.

Multibranch Pipeline

The **Source Code Management** section of Jenkins is configured by **adding the GitHub repository URL**, providing **credentials**, and specifying the **branch to build (main)** for the pipeline.



Jenkins / spring-pet-clinic- / Configuration

Configure

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

None [✓ Git]

Repositories

Repository URL: https://github.com/tuba-chaudhary/spring-petclinic.git

Credentials: tuba-chaudhary/******

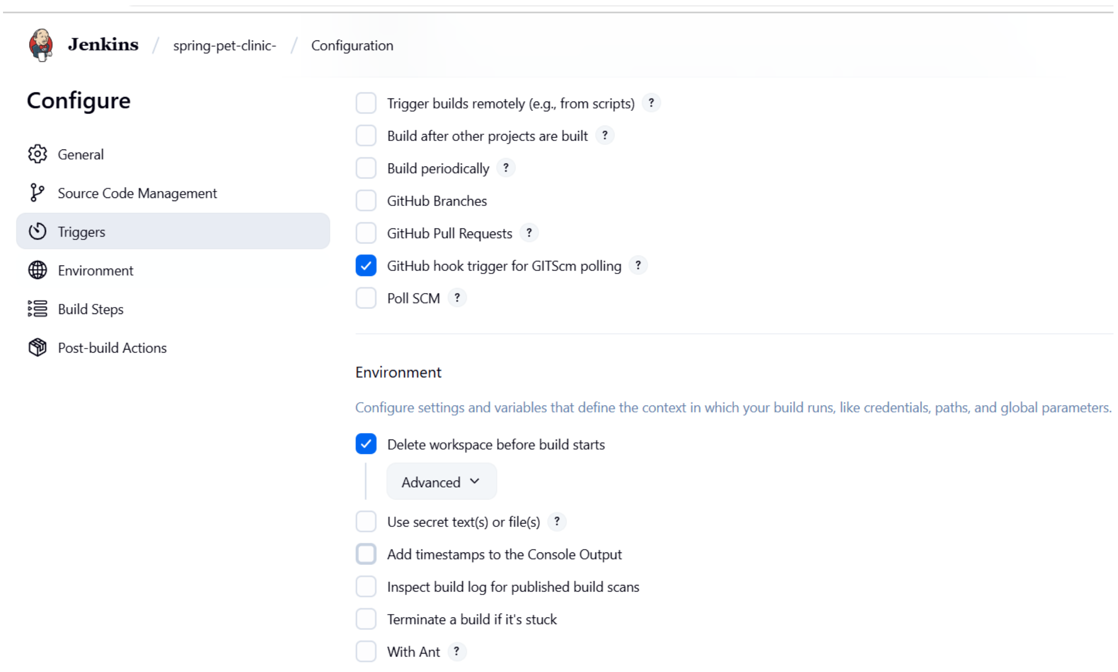
Add | Advanced | Add Repository

Branches to build

Branch Specifier (blank for 'any'): */main

Add Branch | Save | Apply

Jenkins is set to **start a build automatically when GitHub sends a webhook**, and it will **clear the old workspace before starting**.



Jenkins / spring-pet-clinic- / Configuration

Configure

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

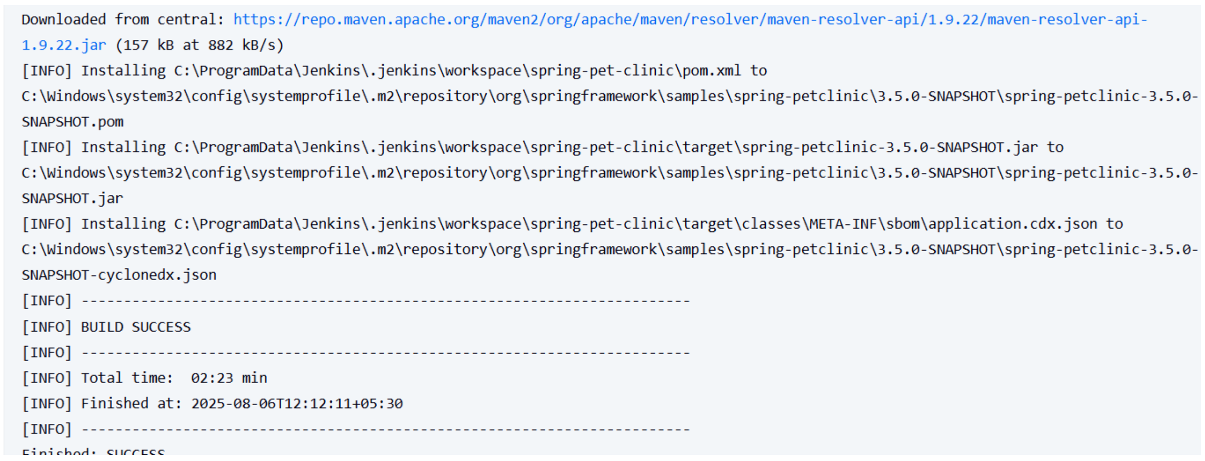
- [ ] Trigger builds remotely (e.g., from scripts)
- [ ] Build after other projects are built
- [ ] Build periodically
- [ ] GitHub Branches
- [ ] GitHub Pull Requests
- [x] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

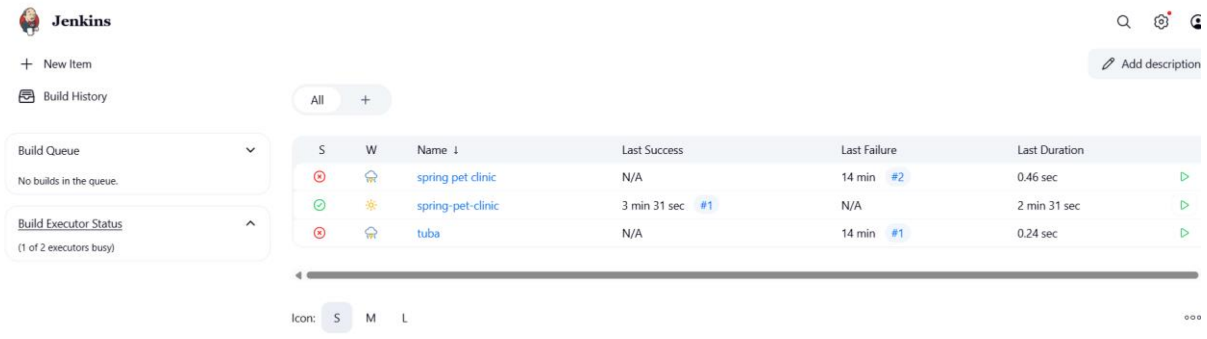
- [x] Delete workspace before build starts
  - Advanced
- [ ] Use secret text(s) or file(s)
- [ ] Add timestamps to the Console Output
- [ ] Inspect build log for published build scans
- [ ] Terminate a build if it's stuck
- [ ] With Ant

Jenkins successfully built the Spring Pet Clinic project using Maven, and the console output shows BUILD SUCCESS with the total build time



```
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-api/1.9.22/maven-resolver-api-
1.9.22.jar (157 kB at 882 kB/s)
[INFO] Installing C:\ProgramData\Jenkins\.jenkins\workspace\spring-pet-clinic\pom.xml to
C:\Windows\system32\config\systemprofile\.m2\repository\org\springframework\samples\spring-petclinic\3.5.0-SNAPSHOT\spring-petclinic-3.5.0-
SNAPSHOT.pom
[INFO] Installing C:\ProgramData\Jenkins\.jenkins\workspace\spring-pet-clinic\target\spring-petclinic-3.5.0-SNAPSHOT.jar to
C:\Windows\system32\config\systemprofile\.m2\repository\org\springframework\samples\spring-petclinic\3.5.0-SNAPSHOT\spring-petclinic-3.5.0-
SNAPSHOT.jar
[INFO] Installing C:\ProgramData\Jenkins\.jenkins\workspace\spring-pet-clinic\target\classes\META-INF\sbom\application.cdx.json to
C:\Windows\system32\config\systemprofile\.m2\repository\org\springframework\samples\spring-petclinic\3.5.0-SNAPSHOT\spring-petclinic-3.5.0-
SNAPSHOT-cyclonedx.json
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  02:23 min
[INFO] Finished at: 2025-08-06T12:12:11+05:30
[INFO] ------------------------------------------------------------------------
Finished: SUCCESS
```

The Jenkins dashboard shows the list of jobs with their build status, last success, last failure, and duration for the Spring Pet Clinic



Build Queue

No builds in the queue.

Build Executor Status

(1 of 2 executors busy)

S	W	Name ↓	Last Success	Last Failure	Last Duration
⊗	☂	spring pet clinic	N/A	14 min #2	0.46 sec
✓	☀	spring-pet-clinic	3 min 31 sec #1	N/A	2 min 31 sec
⊗	☂	tuba	N/A	14 min #1	0.24 sec

<u>CONCLUSION</u> : In this experiment, a GitHub repository was successfully connected to Jenkins using webhooks and Ngrok. Jenkins automatically triggered builds on new commits, and the pipeline ran successfully with BUILD SUCCESS.