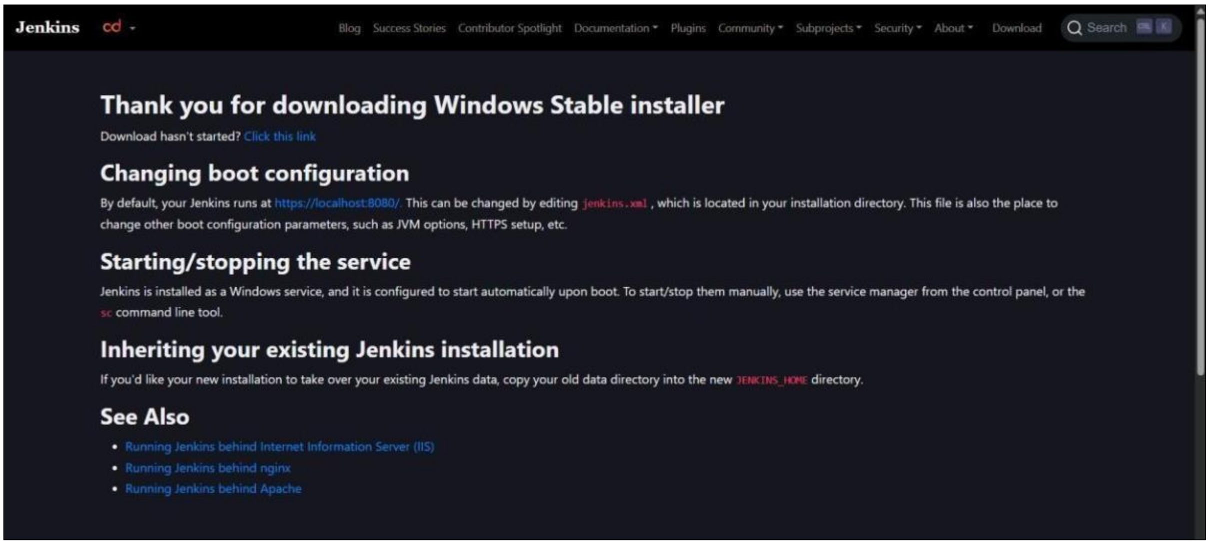
Experiment No.4

Install Jenkins and configure it using setup wizard



Thank you for downloading Windows Stable installer

Download hasn't started? Click this link

Changing boot configuration

By default, your Jenkins runs at https://localhost:8080/. This can be changed by editing `jenkins.xml` , which is located in your installation directory. This file is also the place to change other boot configuration parameters, such as JVM options, HTTPS setup, etc.

Starting/stopping the service

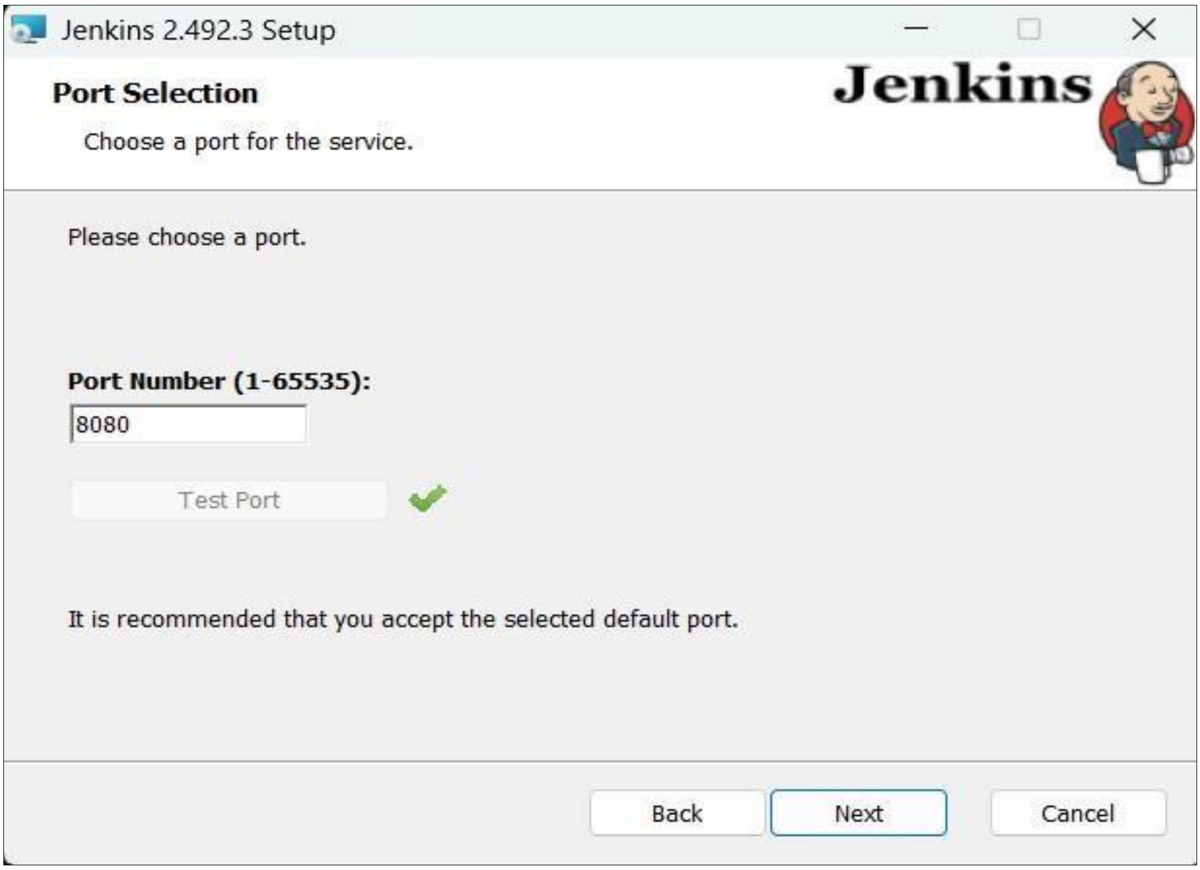
Jenkins is installed as a Windows service, and it is configured to start automatically upon boot. To start/stop them manually, use the service manager from the control panel, or the `sc` command line tool.

Inheriting your existing Jenkins installation

If you'd like your new installation to take over your existing Jenkins data, copy your old data directory into the new `JENKINS_HOME` directory.

See Also

- Running Jenkins behind Internet Information Server (IIS)
- Running Jenkins behind nginx
- Running Jenkins behind Apache



Port Selection

Choose a port for the service.

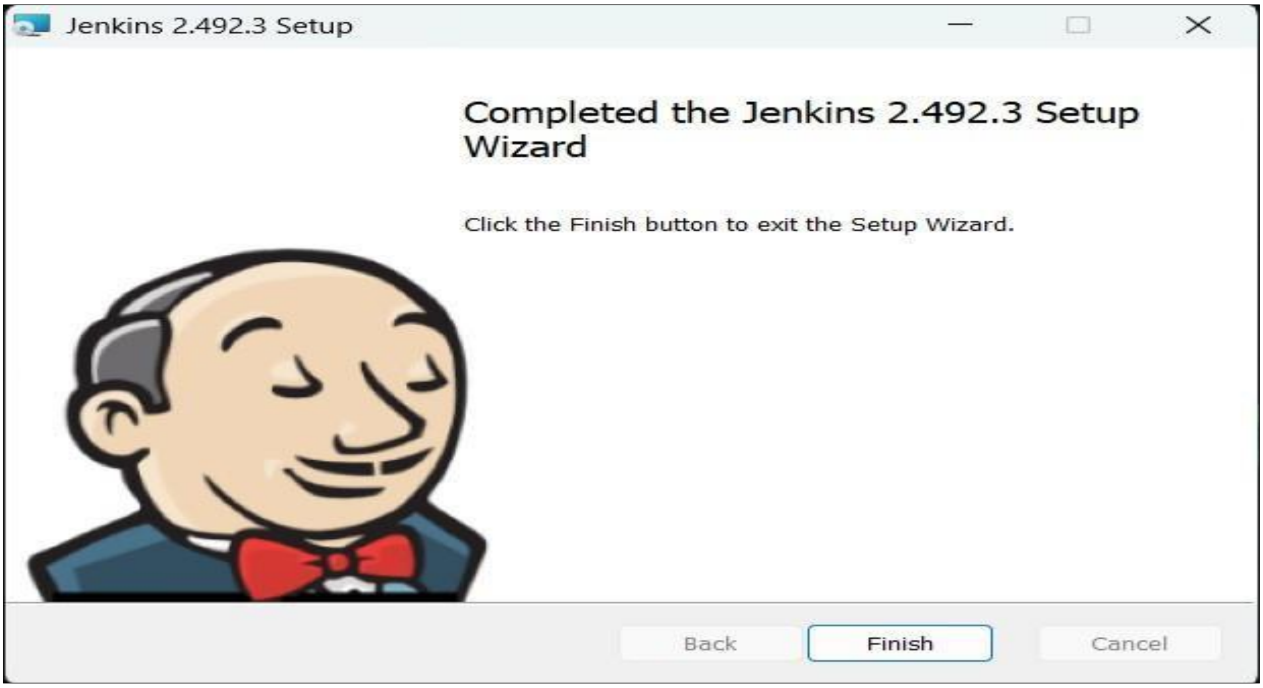
Please choose a port.

Port Number (1-65535): 8080

✓

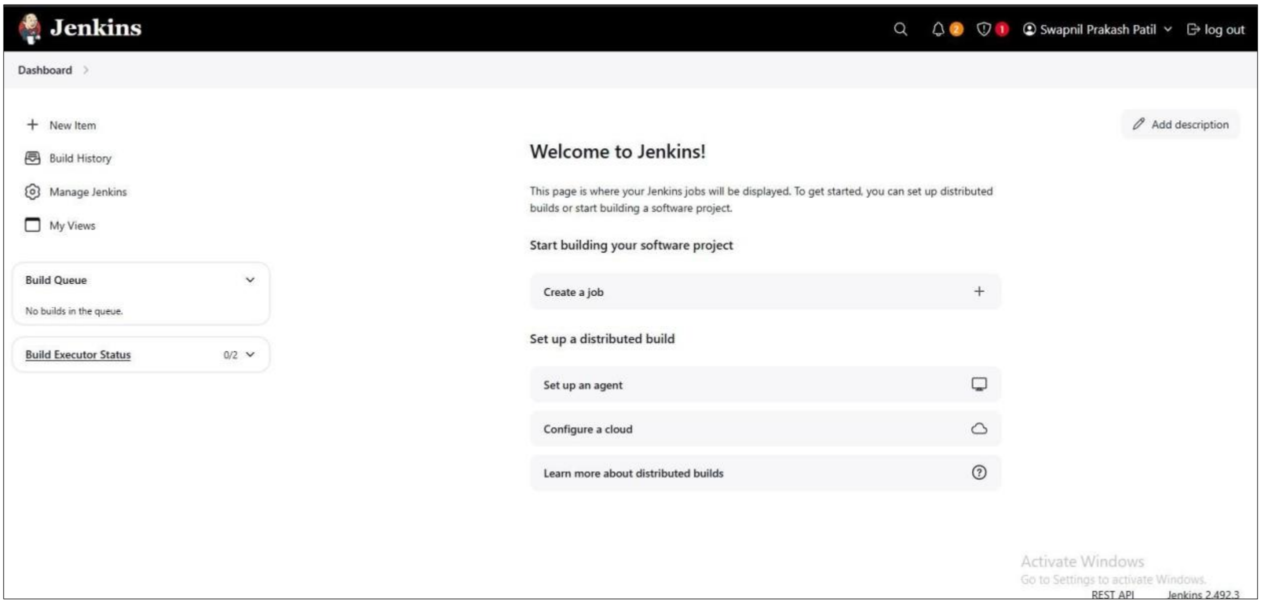
It is recommended that you accept the selected default port.

Experiment No.4



Completed the Jenkins 2.492.3 Setup Wizard

Click the Finish button to exit the Setup Wizard.



No builds in the queue.

Welcome to Jenkins!

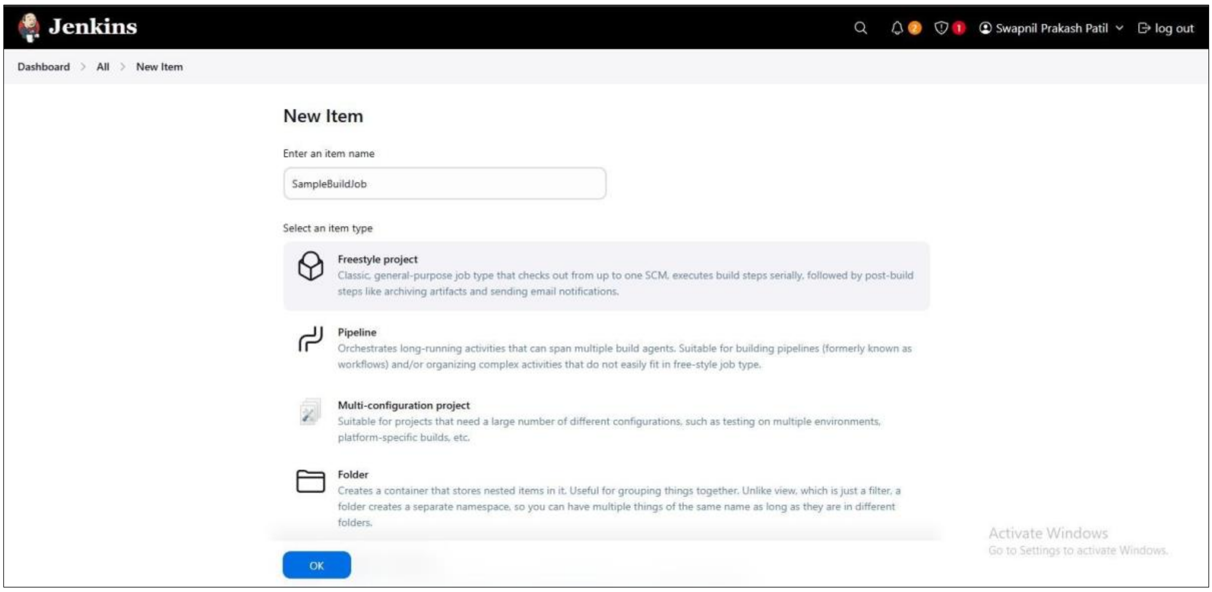
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

Experiment No.4

Create new Free style project



New Item

Enter an item name

SampleBuildJob

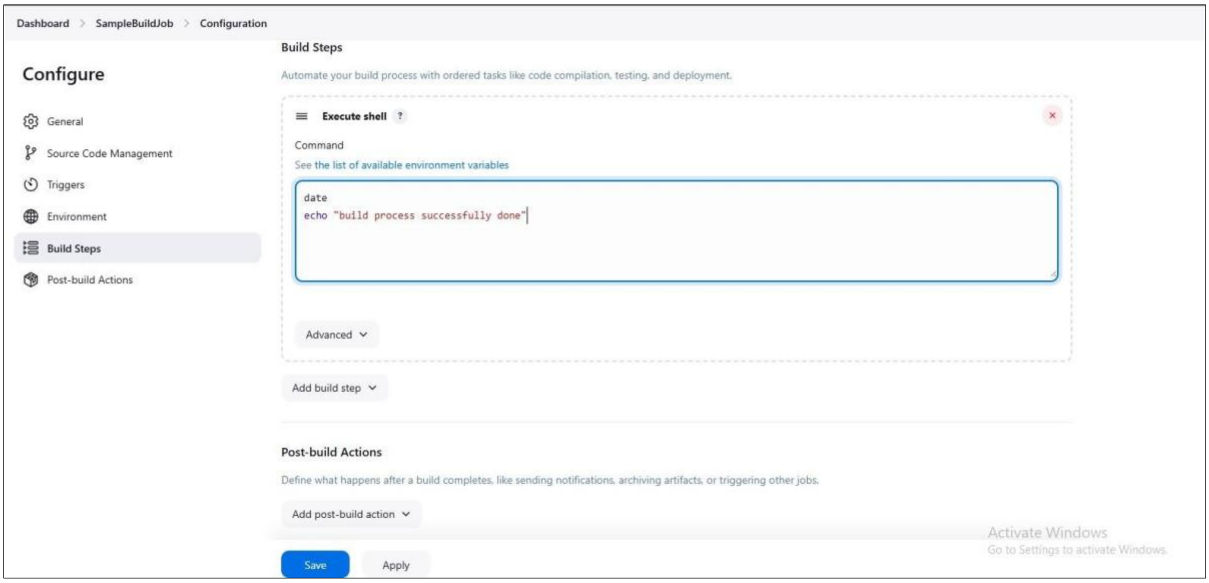
Select an item type

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.

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.



Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute shell

Command

See the list of available environment variables

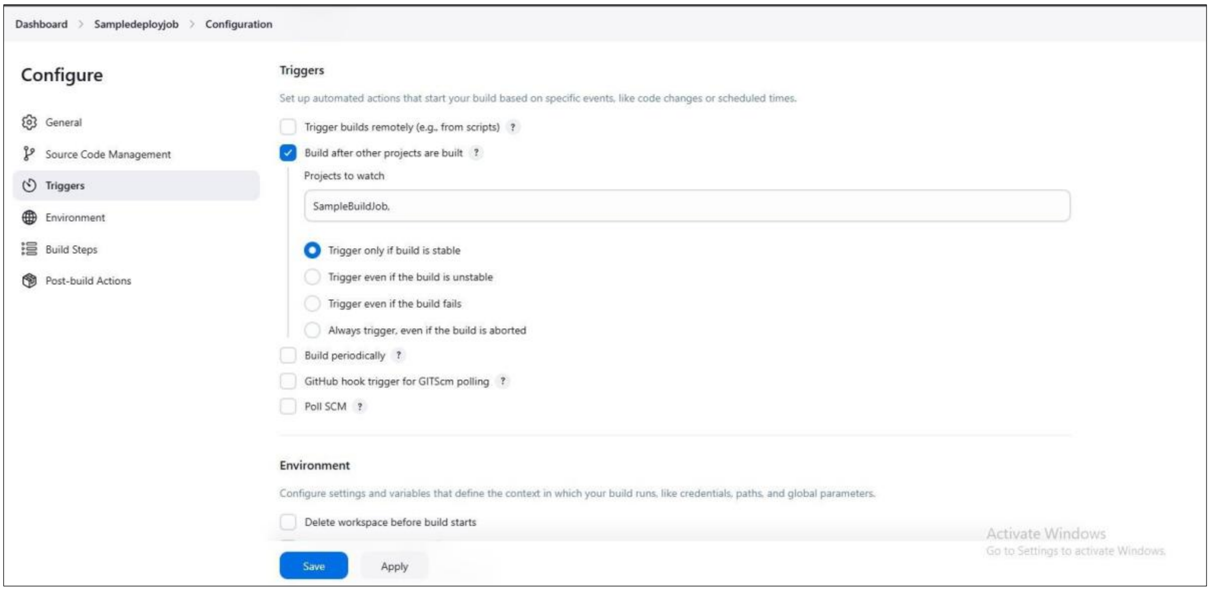
```
date
echo "build process successfully done"
```

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Experiment No.4

Trigger the build



Dashboard > Sampledeployjob > Configuration

Configure

Triggers

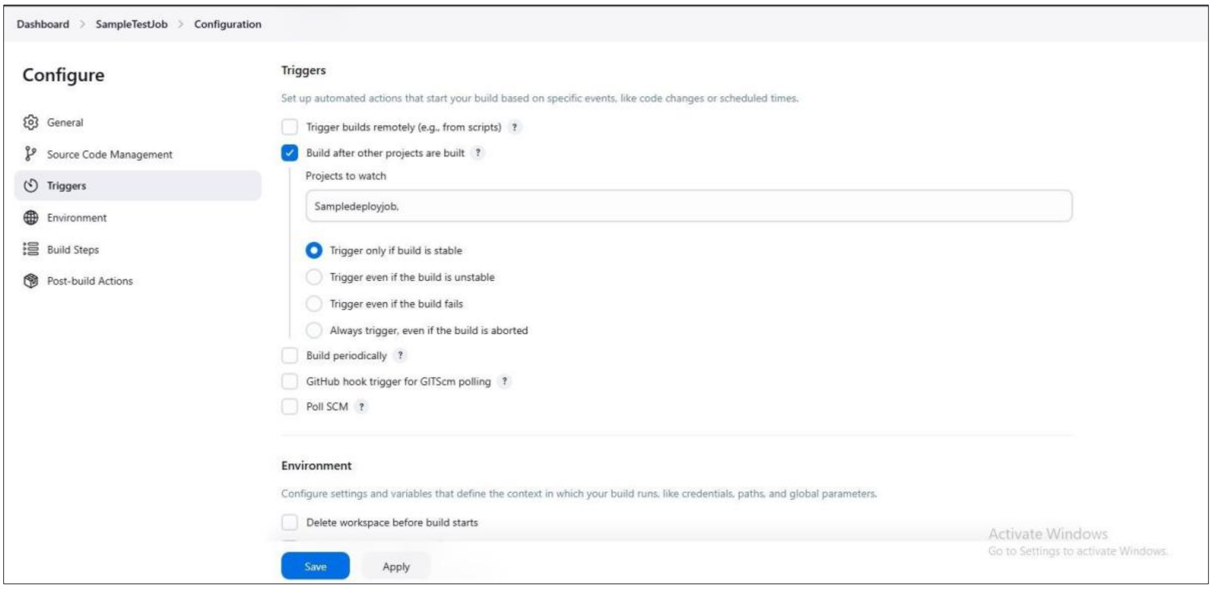
Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- □ Trigger builds remotely (e.g., from scripts)
- ☑ Build after other projects are built
  - Projects to watch: SampleBuildJob,
  - ● Trigger only if build is stable
  - ○ Trigger even if the build is unstable
  - ○ Trigger even if the build fails
  - ○ Always trigger, even if the build is aborted
- □ Build periodically
- □ 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.

- □ Delete workspace before build starts



Dashboard > SampleTestJob > Configuration

Configure

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

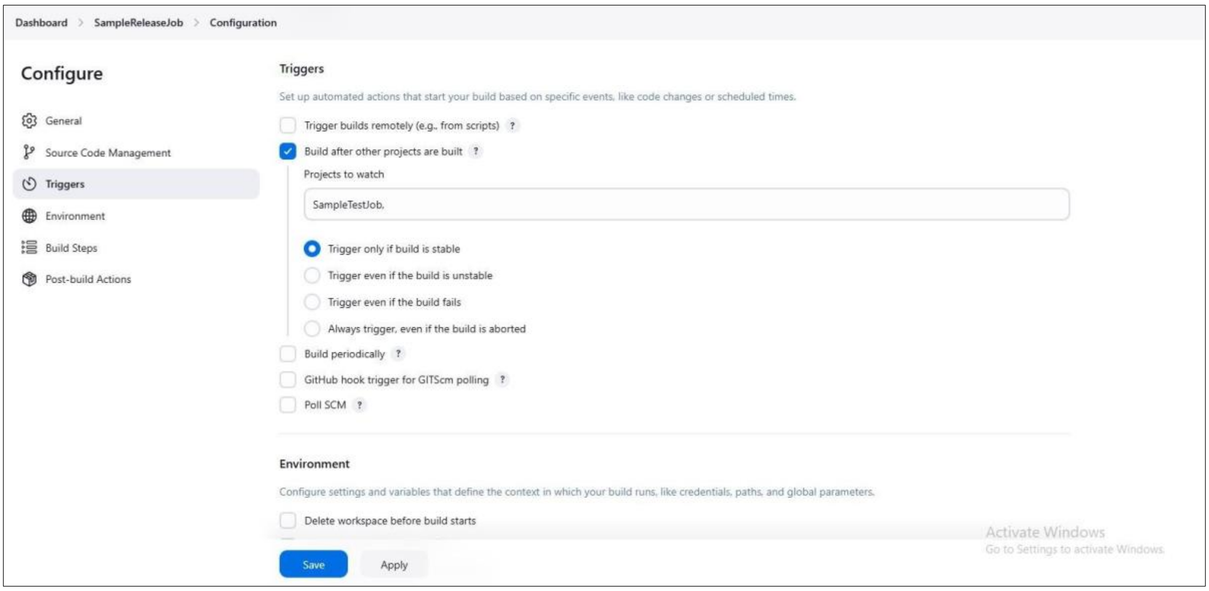
- □ Trigger builds remotely (e.g., from scripts)
- ☑ Build after other projects are built
  - Projects to watch: Sampledeployjob,
  - ● Trigger only if build is stable
  - ○ Trigger even if the build is unstable
  - ○ Trigger even if the build fails
  - ○ Always trigger, even if the build is aborted
- □ Build periodically
- □ 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.

- □ Delete workspace before build starts

Experiment No.4



Dashboard > SampleReleaseJob > Configuration

Configure

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

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

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

☑ Build after other projects are built ?

Projects to watch

SampleTestJob,

● Trigger only if build is stable

○ Trigger even if the build is unstable

○ Trigger even if the build fails

○ Always trigger, even if the build is aborted

□ Build periodically ?

□ GitHub hook trigger for GITScm polling ?

□ Poll SCM ?

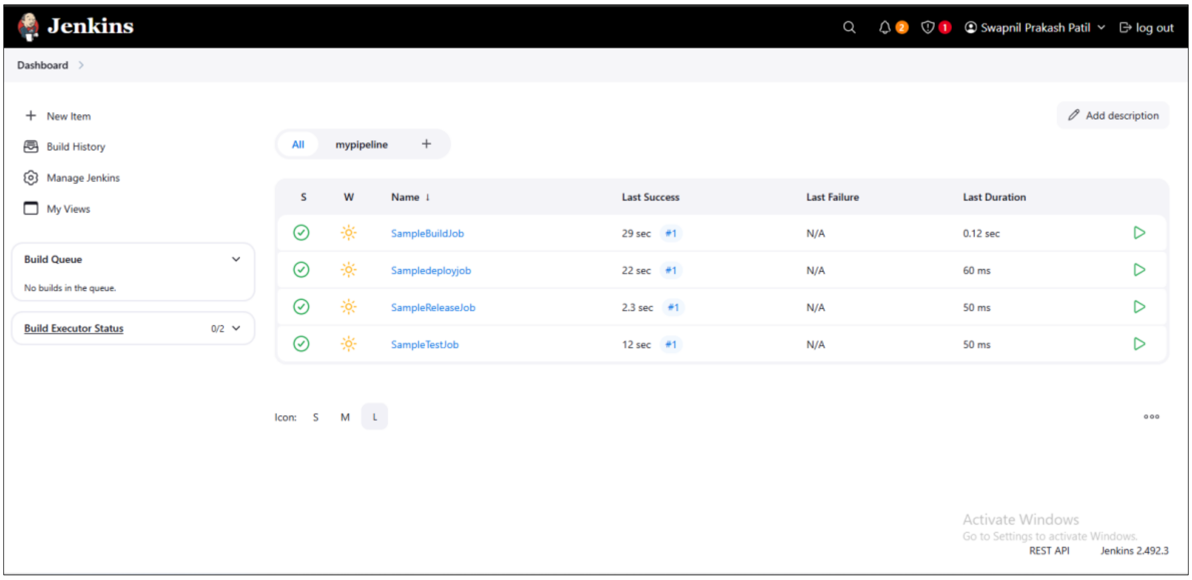
Environment

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

□ Delete workspace before build starts

Save Apply

Activate Windows
Go to Settings to activate Windows.



Jenkins — Swapnil Prakash Patil — log out

Dashboard >

- New Item
- Build History
- Manage Jenkins
- My Views

Build Queue

No builds in the queue.

Build Executor Status 0/2

Add description

All | mypipeline | +

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	SampleBuildJob	29 sec #1	N/A	0.12 sec
✓	☀	Sampledeployjob	22 sec #1	N/A	60 ms
✓	☀	SampleReleaseJob	2.3 sec #1	N/A	50 ms
✓	☀	SampleTestJob	12 sec #1	N/A	50 ms

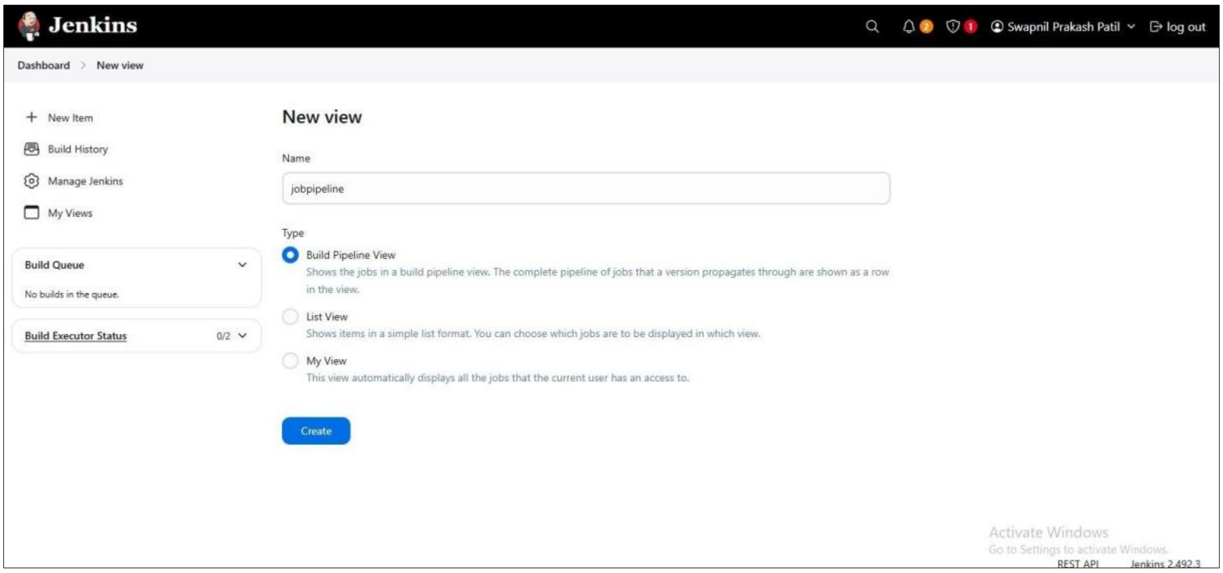
Icon: S M L

Activate Windows
Go to Settings to activate Windows.

REST API Jenkins 2.492.3

Experiment No.4

Create pipeline of jobs



New view

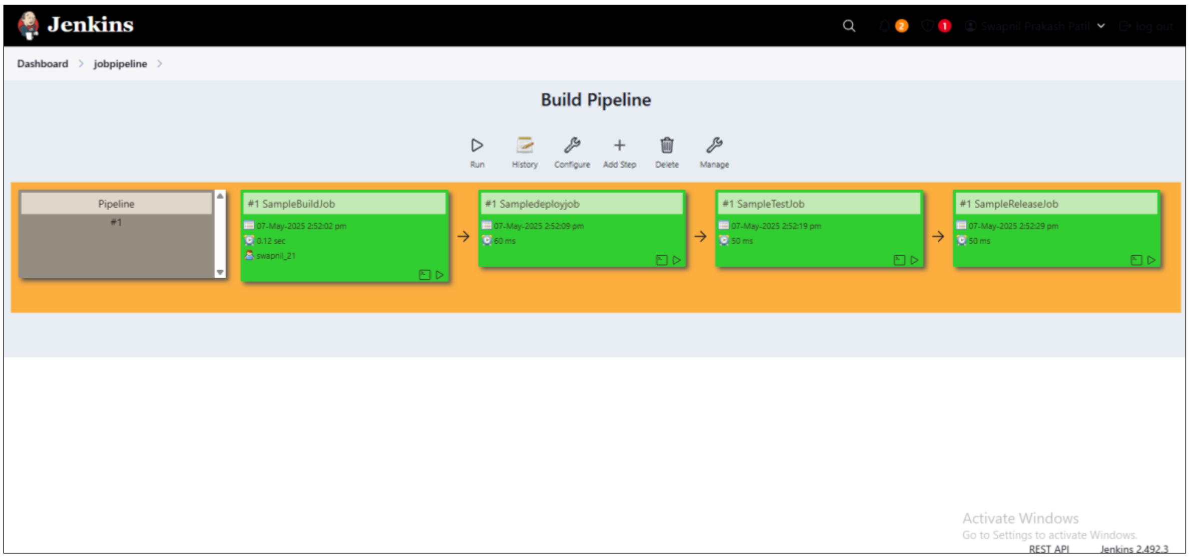
Name: jobpipeline

Type

● Build Pipeline View
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.

○ List View
Shows items in a simple list format. You can choose which jobs are to be displayed in which view.

○ My View
This view automatically displays all the jobs that the current user has an access to.



Build Pipeline

Pipeline #1

#1 SampleBuildJob
07-May-2025 2:52:02 pm
0.12 sec
swapnil_21

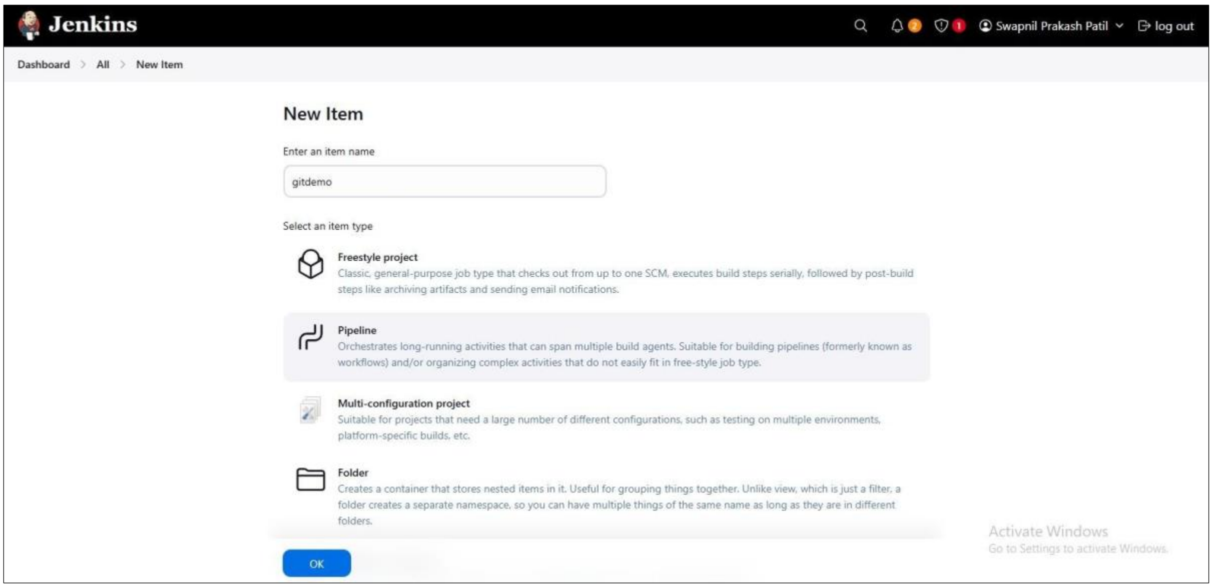
→ #1 Sampledeployjob
07-May-2025 2:52:09 pm
60 ms

→ #1 SampleTestJob
07-May-2025 2:52:19 pm
50 ms

→ #1 SampleReleaseJob
07-May-2025 2:52:29 pm
50 ms

Experiment No.4

Plugin git to Jenkin



New Item

Enter an item name

gitdemo

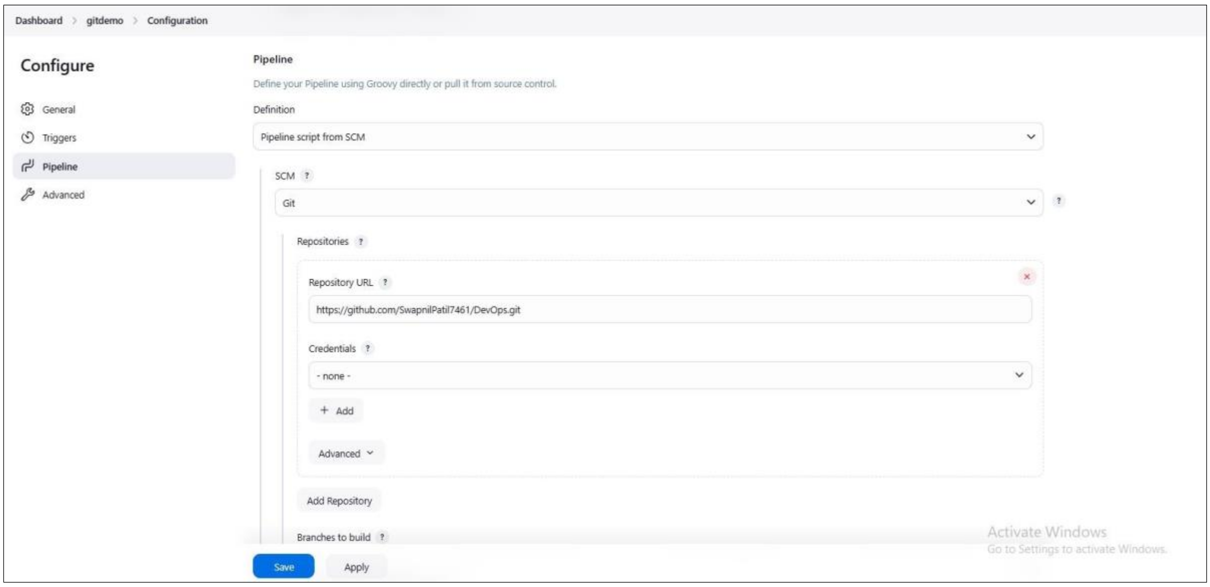
Select an item type

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.

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.



Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition: Pipeline script from SCM

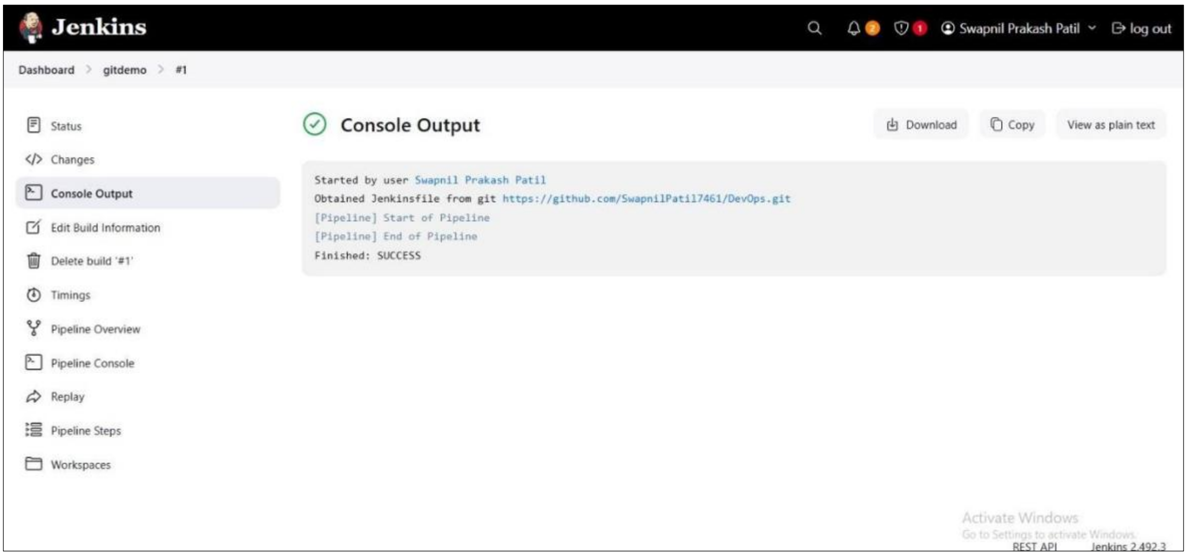
SCM: Git

Repositories

Repository URL: https://github.com/SwapnilPatil7461/DevOps.git

Credentials: - none -

Experiment No.4



Console Output

```
Started by user Swapnil Prakash Patil
Obtained Jenkinsfile from git https://github.com/SwapnilPatil7461/DevOps.git
[Pipeline] Start of Pipeline
[Pipeline] End of Pipeline
Finished: SUCCESS
```