INDEX

Sr. No	Topic	Date	Page No	Sign
1	Setting up and Managing Git Repository a) Set up a Git repository and perform basic Git operations (clone, commit, push, pull). b) Create and manage branches in Git. c) Resolve merge conflicts in Git.			
2	Implementing CI using Jenkins and CircleCI a) Set up a Jenkins server and create a basic CI pipeline. b) Implement a CI pipeline using CircleCI.			
3	Implementing CI and CD a) Implement CD using Jenkins			
4	Securing Jenkins using DevSecOps			
5	Containerization using Docker			
6	Set up Kubernetes and deploy a sample application			
7	Configuration Management using Puppet			

Practical 1:

Setting up and Managing Git Repository

Q1. What is a Version Control System? Define Git. How Git is used in the Version Control System.

A version control system is a software that tracks changes to a file or set of files over time so that you can recall specific versions later. It also allows you to work together with other programmers.

The version control system is a collection of software tools that help a team to manage changes in a source code. It uses a special kind of database to keep track of every modification to the code.

Developers can compare earlier versions of the code with an older version to fix the mistakes.

Git is an open-source distributed version control system. It is designed to handle minor to major projects with high speed and efficiency. It is developed to coordinate the work among the developers.

Git is the foundation of many services like GitHub and GitLab, but we can use Git without using any other Git services. Git can be used privately and publicly.

Q2. Explain and Implement all the staging and commit commands in Git.

1) **Git config command**

This command configures the user. The Git config command is the first and necessary command used on the Git command line. This command sets the author name and email address to be used with your commits. Git config is also used in other scenarios.

Syntax

$ git config --global user.name "ImDwivedi1"

$ git config --global user.email "Himanshudubey481@gmail.com"

2) **Git Init command**

This command is used to create a local repository.

Syntax

$ git init Demo

The init command will initialize an empty repository. See the below screenshot.

3) **Git clone command**

This command is used to make a copy of a repository from an existing URL. If I want a local copy of my repository from GitHub, this command allows creating a local copy of that repository on your local directory from the repository URL

Syntax

$ git clone URL

4) **Git add command**

This command is used to add one or more files to staging (Index) area.

Syntax

To add one file

$ git add Filename

To add more than one file

$ git add*

5) **Git commit command**

This is used in two scenarios. They are as follows.

Git commit -m

This command changes the head. It records or snapshots the file permanently in the version history with a message.

Syntax

$ git commit -m " Commit Message"

Git commit -a

This command commits any files added in the repository with git add and also commits any files you've changed since then.

Syntax

$ git commit -a

6) **Git status command**

The status command is used to display the state of the working directory and the staging area. It allows you to see which changes have been staged, which haven't, and which files aren?t

being tracked by Git. It does not show you any information about the committed project history. For this, you need to use the git log. It also lists the files that you've changed and those you still need to add or commit.

Syntax

$ git status

7) **Git push Command**

It is used to upload local repository content to a remote repository. Pushing is an act of transferring commits from your local repository to a remote repo. It's the complement to git fetch, but whereas fetching imports commits to local branches on comparatively pushing exports commits to remote branches. Remote branches are configured by using the git remote command. Pushing is capable of overwriting changes, and caution should be taken when pushing.

Git push command can be used as follows.

Git push origin master

This command sends the changes made on the master branch, to your remote repository.

Syntax

$ git push [variable name] master

Git push -all

This command pushes all the branches to the server repository.

Syntax

$ git push --all

8) **Git pull command**

This is used to receive data from GitHub. It fetches and merges changes on the remote server to your working directory.

Syntax

$ git pull URL

9) **Git Branch Command**

This command lists all the branches available in the repository.

Syntax

$ git branch

10**) Git Merge Command**

This command is used to merge the specified branch history into the current branch.

Syntax

$ git merge BranchName

11) **Git log Command**

This command is used to check the commit history.

Syntax

$ git log

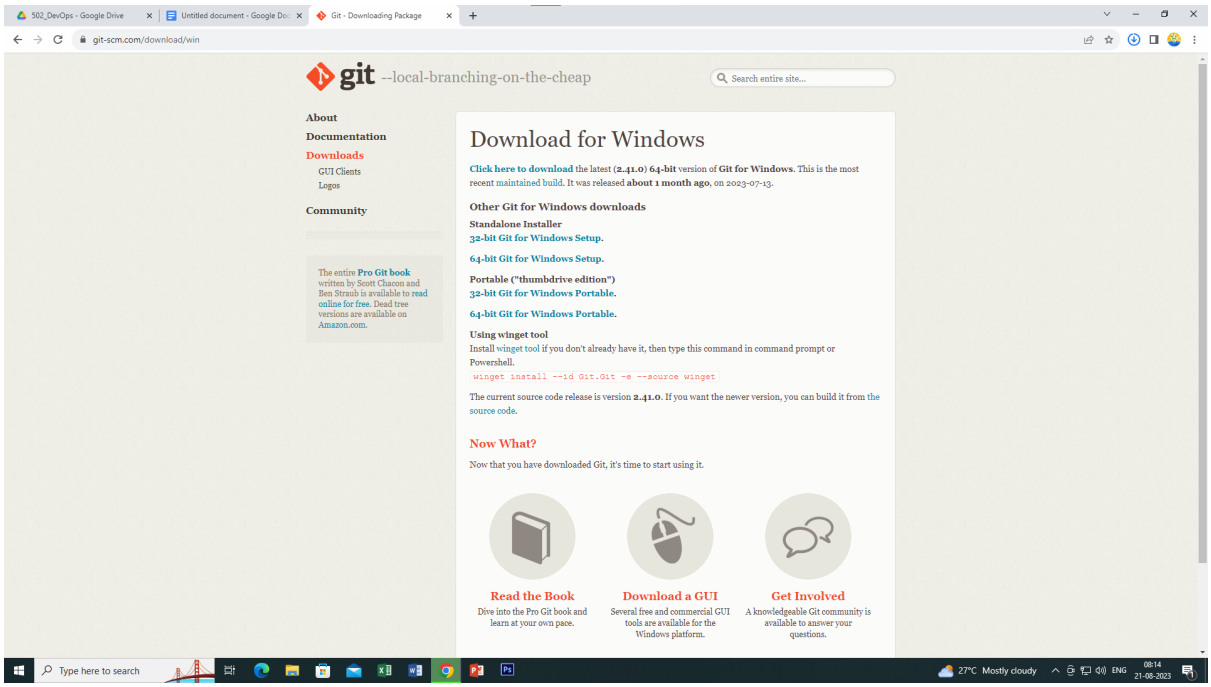
By default, if no argument passed, Git log shows the most recent commits first. We can limit the number of log entries displayed by passing a number as an option, such as -3 to show only the last three entries.

$ git log -3

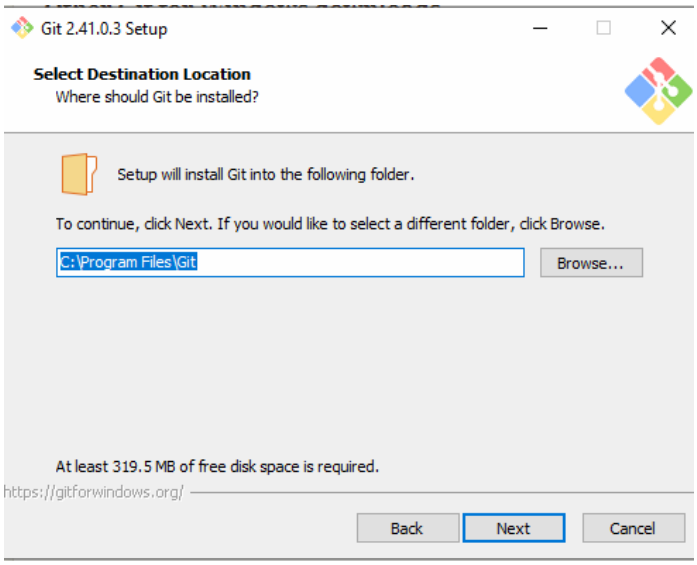
12) **Git remote Command**

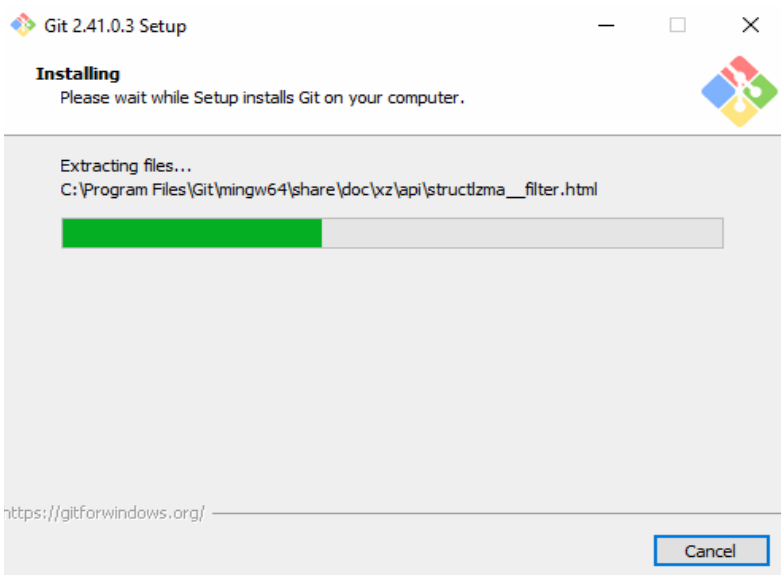
This is used to connect your local repository to the remote server. This command allows you to create, view, and delete connections to other repositories. These connections are more like bookmarks rather than direct links to other repositories. This command doesn't provide real-time access to repositories.

Installation:




Git 2.41.0.3 Setup
Information
Please read the following important information before continuing.
When you are ready to continue with Setup, click Next.
GNU General Public License
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Preamble
The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change
https://gitforwindows.org/
Next
Cancel


Git 2.41.0.3 Setup
Select Destination Location
Where should Git be installed?
Setup will install Git into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program Files\Git
Browse...
At least 319.5 MB of free disk space is required.
https://gitforwindows.org/
Back
Next
Cancel


Git 2.41.0.3 Setup
Installing
Please wait while Setup installs Git on your computer.
Extracting files...
C:\Program Files\Git\mingw64\share\doc\xz\api\structlzma__filter.html
https://gitforwindows.org/
Cancel

CG-CS-PC01@CG-CS-PC01 MINGW64 ~

$ git config --global user.name "shwethavenkat4"

CG-CS-PC01@CG-CS-PC01 MINGW64 ~

$ git config --global user. email "shwetha. venkat03@gmail. com"

CG-CS-PC01@CG-CS-PC01 MINGW64 ~

$ git clone https://github.com/shwethavenkat4/php-payro11-management-git

Cloning into 'php-payroll-management'... remote: Enumerating objects: 41, done.

emote: Counting objects: 100% (41/41), done.

emote: Compressing objects: 100% (25/25), done.

emote: Total 41 (delta 15),

reused 41 (delta 18), pack-reused OReceiving object

Receiving objects: 87% (36/41)

Receiving objects: 100% (41/41), 153.50 KiB | 11.81 MiB/s, done.

Resolving deltas: 100% (15/15), done.

CG-CS-PC01@CG-CS-PC01 MINGW64 ~/dvPrac1 (master)

$ git add test1. py

CG-CS-PC01@CG-CS-PC01 MINGW64 ~/dvPrac1 (master)

$ ls

test1. py

CG-CS-PC01@CG-CS-PC01 MINGW64 ~/dvPrac1 (master)

$ git status

On branch master

No commits yet

Changes to be committed:

"git rm --cached «file>..." to unstage)

new file:

test1. py

CG-CS-PC01@CG-CS-PC01 MINGW64 ~/dvPraci (master)

§ git commit -m "first version of test code'

[master (root-commit) 64bbdf9] first version of test code

1 file changed, 1 insertion(+) create mode 100644 test1py

CG-CS-PC01@CG-CS-PC01 MINGW64

$ git status

On branch master

nothing to commit, working tree clean

~/dvPrac1 (master)

CG-CS-PC01@CG-CS-PC01 MINGW64 ~/dvPrac1 (master)

$ git log

commit 64bbdf9bf24b632627dbba9a25c6adcf552f044 (HEAD -> master)

Author: shwethavenkat4 <shwetha.venkat03@gmail.com

Date:

Mon Aug 21 09:22:24 2023 +0530

first version of test code

CG-CS-PCO1@CG-CS-PC01 MINGW64 ~/dvPraci (main)

§ git push

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 12 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 312 bytes | 312.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

To https://github.com/shwethavenkat4/dvPraci.git

42c13f7..a0caa6f main -> main

test1.py

CG-CS-PC01@CG-CS-PCO1 MINGW64 ~/dvPracI (main)

§ git add testi.py

CG-CS-PC01@CG-CS-PC01 MINGW64 ~/dvPrac1 (main)

$ git status

On branch main

Your branch is up to date with 'origin/main'

Changes to be committed:

(use

"git restore --staged «file›..." to unstage)

modified:

test1.py

CG-CS-PC01@CG-CS-PC01 MINGW64 ~/dvPrac1 (main)

§ git commit -m 'changed code to take input from the user'

1 file changed, 2 insertions(+), 1 deletion (-)

CG-CS-PC01@CG-CS-PC01 MINGW64 ~/dvPraci (main)

§ git push

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Delta compression using up to 12 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 350 bytes | 350.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused O To https://github.com/shwethavenkat4/dvPrac1.git

a0caa6f..77944e7 main -> main

CG-CS-pco1@CG-CS-PCO1 MINGW64 ~/dvPracI (main)

$ git branch

main

CG-CS-PC01@CG-CS-PC01 MINGW64 ~/dvPraci (main)

$ git checkout main

Already on

'main'

Your branch is up to date with 'origin/main'

'CG-CS-PC01@CG-CS-PC01 MINGW64 ~/dvPraci (main)

$ git checkout -b test

Switched to a new branch 'test'

CG-CS-PC01@CG-CS-PC01 MINGW64 ~/dvPraci (test)

§ git checkout main

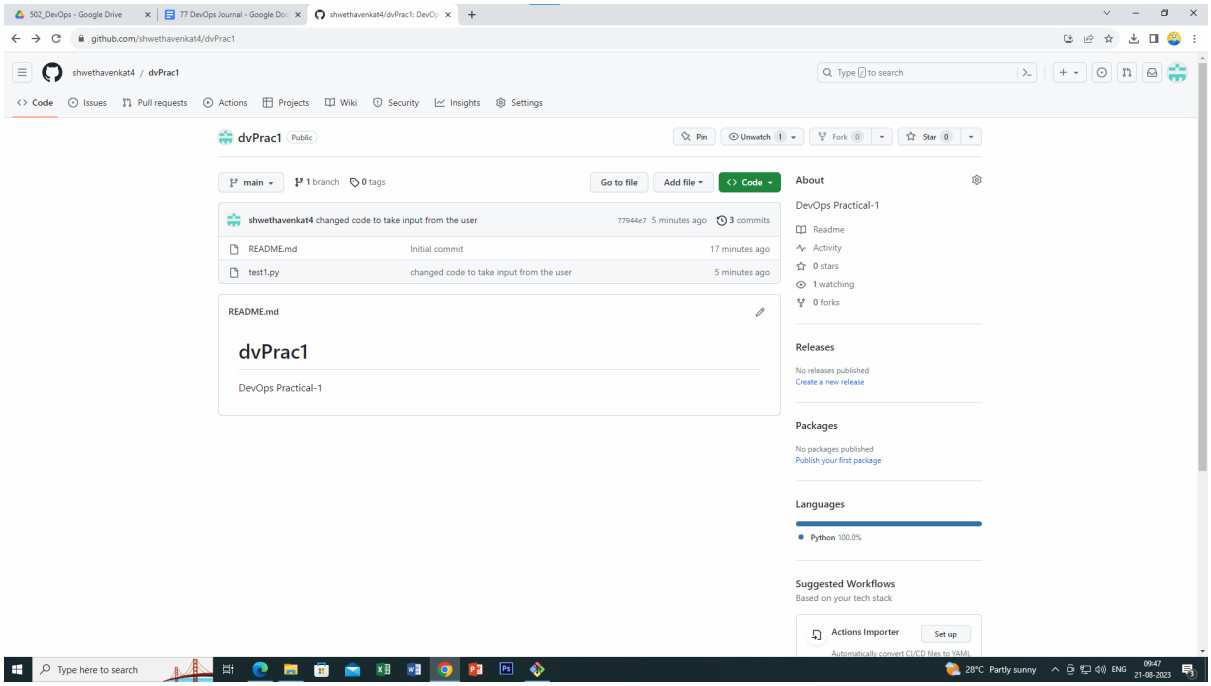
Switched to branch 'main'

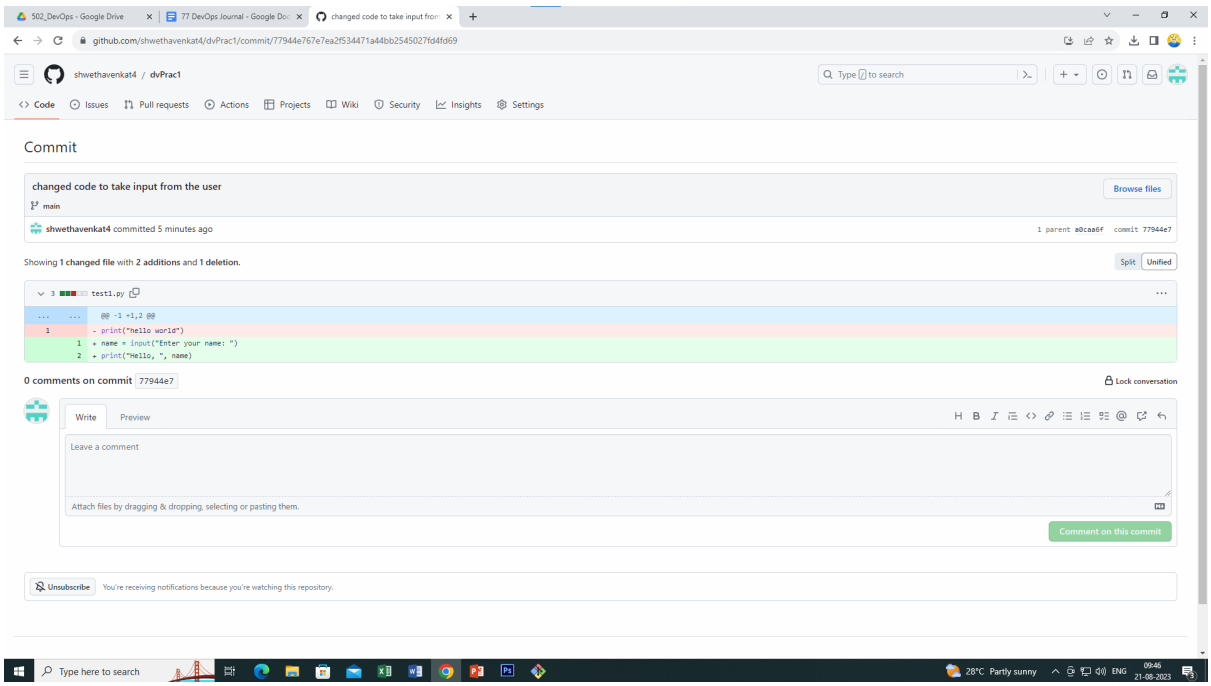
Your branch is up to date with 'origin/main'

CG-CS-PC01@CG-CS-PC01 MINGW64 ~/dvPraci (main)

S git merge test

Already up to date.


github.com/shwethavenkat4/dvPrac1
shwethavenkat4 / dvPrac1
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
dvPrac1
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 branch
0 tags
Go to file
Add file
Code
shwethavenkat4 changed code to take input from the user
77944e7 5 minutes ago
3 commits
README.md
Initial commit
17 minutes ago
test1.py
changed code to take input from the user
5 minutes ago
README.md
dvPrac1
DevOps Practical-1
About
DevOps Practical-1
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
Python 100.0%
Suggested Workflows
Based on your tech stack
Actions Importer
Set up
Type here to search
28°C Partly sunny
ENG
09:47
21-08-2023


github.com/shwethavenkat4/dvPrac1/commit/77944e767e7ea2f534471a44bb2545027fd4fd69
shwethavenkat4 / dvPrac1
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Commit
changed code to take input from the user
main
Browse files
shwethavenkat4 committed 5 minutes ago
1 parent a0caa6f commit 77944e7
Showing 1 changed file with 2 additions and 1 deletion.
Split
Unified
test1.py
@@ -1 +1,2 @@
- print("hello world")
+ name = input("Enter your name: ")
+ print("Hello, ", name)
0 comments on commit 77944e7
Lock conversation
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Comment on this commit
Unsubscribe
You're receiving notifications because you're watching this repository.
Type here to search
28°C Partly sunny
ENG
09:46
21-08-2023

Practical 2a:

Implementing CI using Jenkins

Q1. Integrate GitHub with Jenkins by fetching the source code from GitHub and build it using Jenkins.

Jenkins is a CI (Continuous Integration) server and this means that it needs to check out source code from a source code repository and build code. Jenkins has outstanding support for various source code management systems like Subversion, CVS etc.

Q2. Install and setup Jenkins.

Jenkins is an open source automation tool written in Java programming language that allows continuous integration.

Jenkins builds and tests our software projects which continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.

It also allows us to continuously deliver our software by integrating with a large number of testing and deployment technologies. Jenkins offers a straightforward way to set up a continuous integration or continuous delivery environment for almost any combination of languages and source code repositories using pipelines, as well as automating other routine development tasks.

With the help of Jenkins, organizations can speed up the software development process through automation. Jenkins adds development life-cycle processes of all kinds, including build, document, test, package, stage, deploy static analysis and much more.

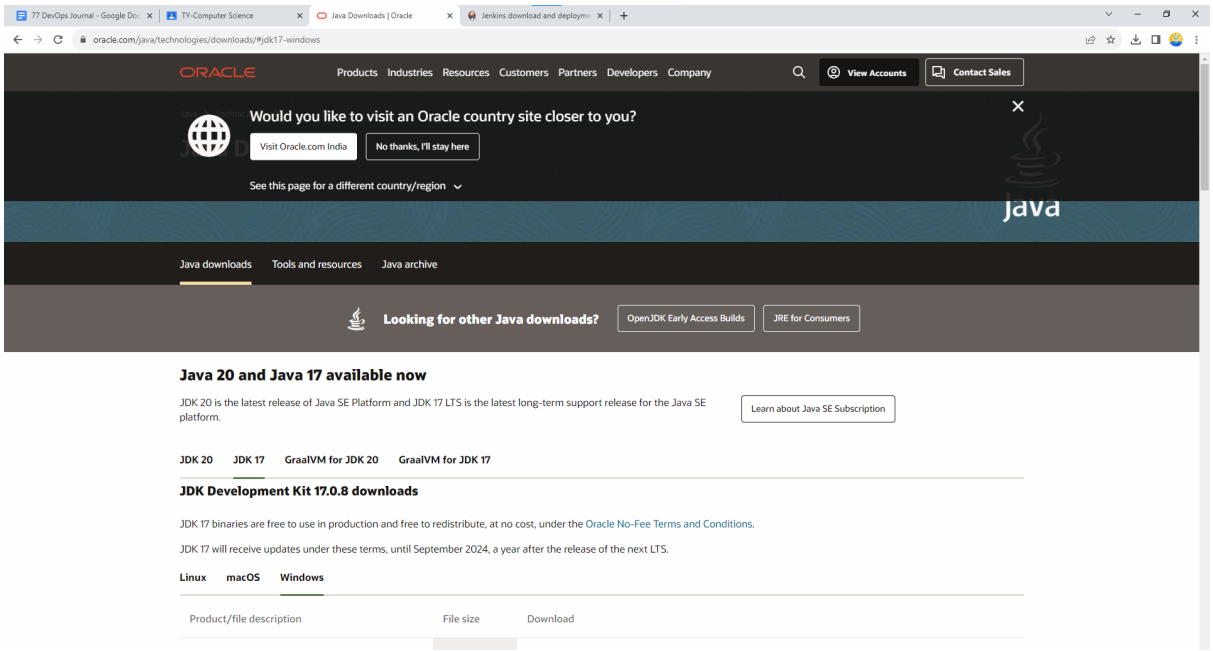
Jenkins achieves CI (Continuous Integration) with the help of plugins. Plugins is used to allow the integration of various DevOps stages. If you want to integrate a particular tool, you have to install the plugins for that tool. For example: Maven 2 Project, Git, HTML Publisher, Amazon EC2, etc.

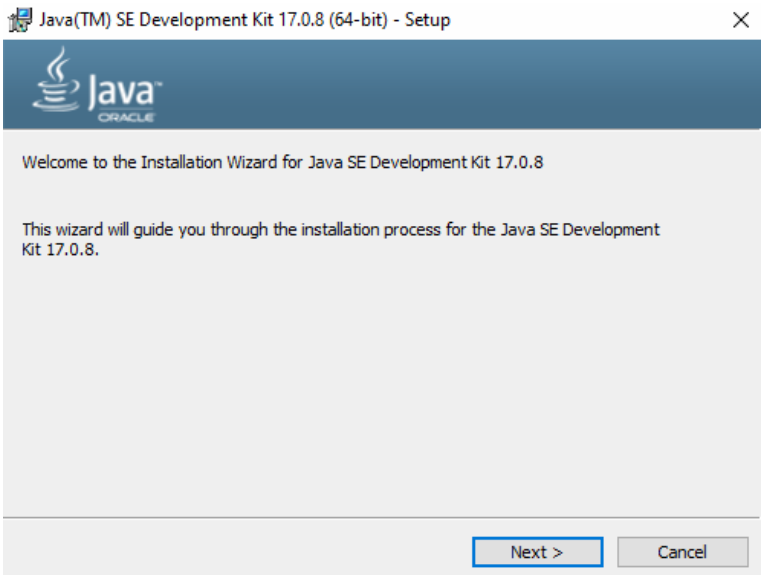
For example: If any organization is developing a project, then Jenkins will continuously test your project builds and show you the errors in early stages of your development.

Possible steps executed by Jenkins are for example:

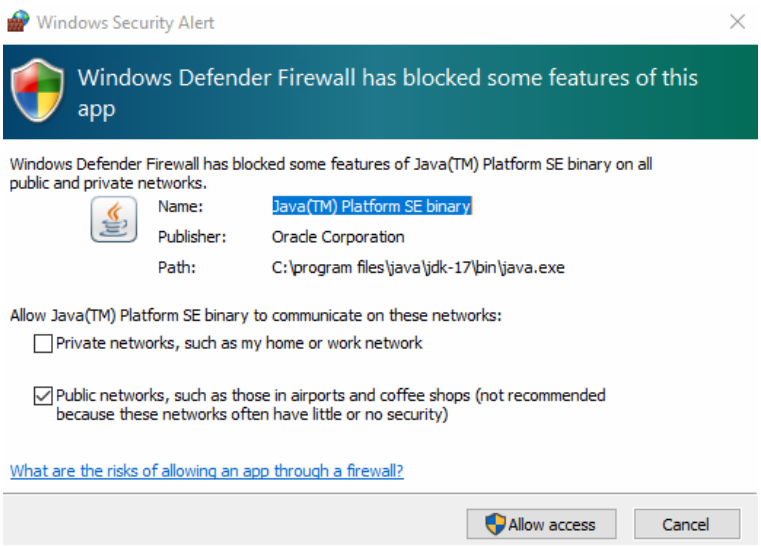
- Perform a software build using a build system like Gradle or Maven Apache
- Execute a shell script
- Archive a build result
- Running software tests

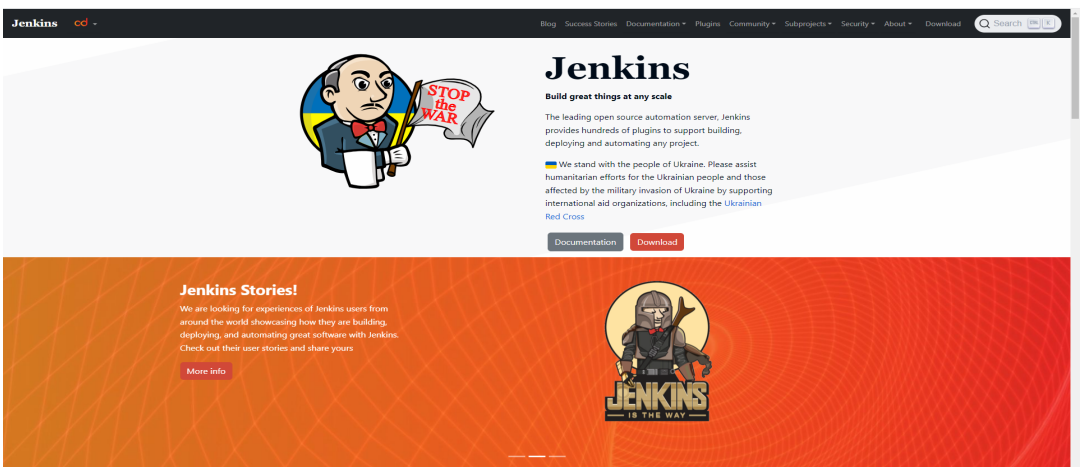
Part-1: Installing JDK and Jenkins

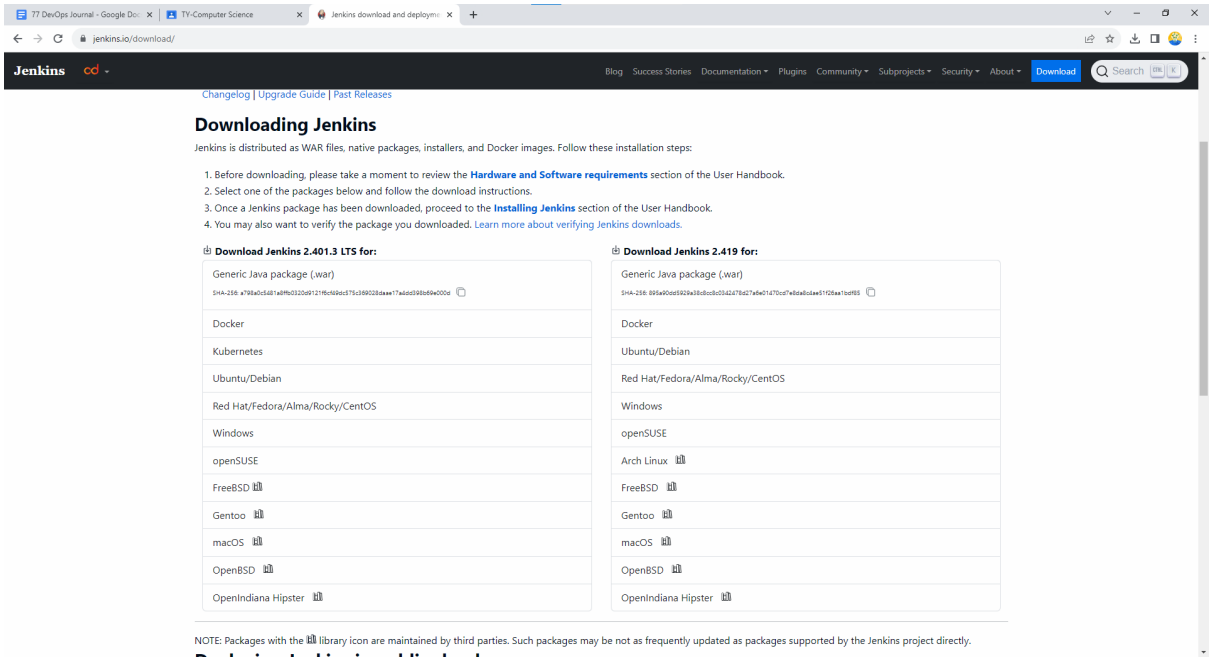


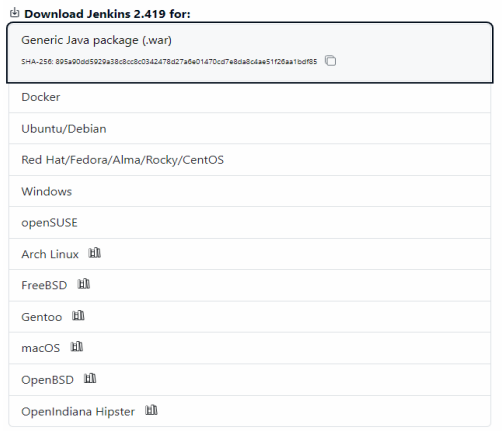


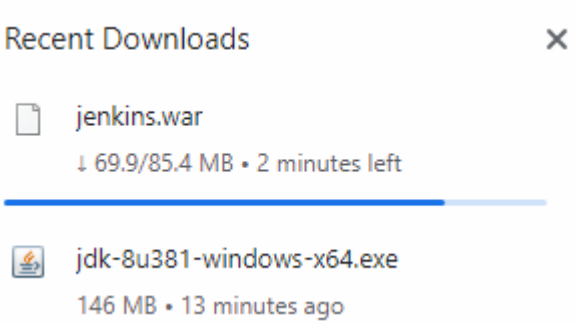

Java(TM) SE Development Kit 17.0.8 (64-bit) - Complete
Java
ORACLE
Java(TM) SE Development Kit 17.0.8 (64-bit) Successfully Installed
Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.
Next Steps
Close

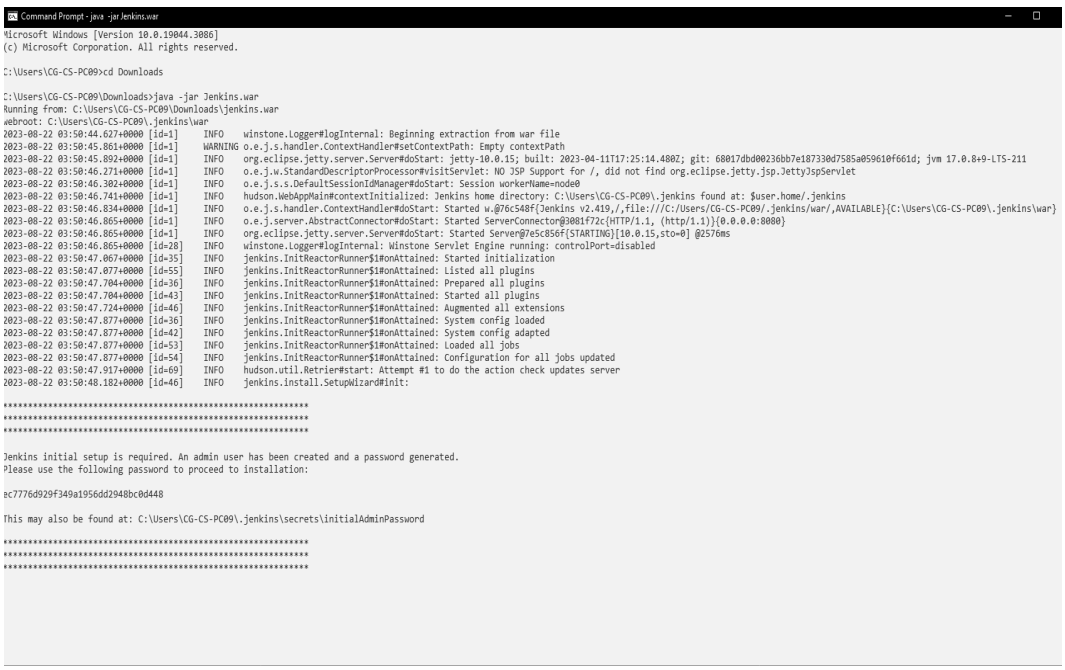

Windows Security Alert
Windows Defender Firewall has blocked some features of this app
Windows Defender Firewall has blocked some features of Java(TM) Platform SE binary on all public and private networks.
Name: Java(TM) Platform SE binary
Publisher: Oracle Corporation
Path: C:\program files\java\jdk-17\bin\java.exe
Allow Java(TM) Platform SE binary to communicate on these networks:
Private networks, such as my home or work network
Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)
What are the risks of allowing an app through a firewall?
Allow access
Cancel


Jenkins
Blog
Success Stories
Documentation
Plugins
Community
Subprojects
Security
About
Download
Search
STOP the WAR
Jenkins
Build great things at any scale
The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.
We stand with the people of Ukraine. Please assist humanitarian efforts for the Ukrainian people and those affected by the military invasion of Ukraine by supporting international aid organizations, including the Ukrainian Red Cross
Documentation
Download
Jenkins Stories!
We are looking for experiences of Jenkins users from around the world showcasing how they are building, deploying, and automating great software with Jenkins. Check out their user stories and share yours
More info
JENKINS IS THE WAY


77 DevOps Journal - Google Doc
TY-Computer Science
Jenkins download and deployme
jenkins.io/download/
Jenkins
Blog
Success Stories
Documentation
Plugins
Community
Subprojects
Security
About
Download
Search
Changelog | Upgrade Guide | Past Releases
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 2.401.3 LTS for:
Generic Java package (.war)
SHA-256: a798a0c5481a8ffb0320d9121f6c69dc575c369028daaae17addd398b69e000d
Docker
Kubernetes
Ubuntu/Debian
Red Hat/Fedora/Alma/Rocky/CentOS
Windows
openSUSE
FreeBSD
Gentoo
macOS
OpenBSD
OpenIndiana Hipster
Download Jenkins 2.419 for:
Generic Java package (.war)
SHA-256: 895a90cdd5929a38c8cc8c0342478d27a6e01470cd7e8da8c4ae51f26aa1bdf85
Docker
Ubuntu/Debian
Red Hat/Fedora/Alma/Rocky/CentOS
Windows
openSUSE
Arch Linux
FreeBSD
Gentoo
macOS
OpenBSD
OpenIndiana Hipster
NOTE: Packages with the library icon are maintained by third parties. Such packages may be not as frequently updated as packages supported by the Jenkins project directly.


Download Jenkins 2.419 for:
Generic Java package (.war)
SHA-256: 895a90cdd5929a38c8cc8c0342478d27a6e01470cd7e8da8c4ae51f26aa1bdf85
Docker
Ubuntu/Debian
Red Hat/Fedora/Alma/Rocky/CentOS
Windows
openSUSE
Arch Linux
FreeBSD
Gentoo
macOS
OpenBSD
OpenIndiana Hipster

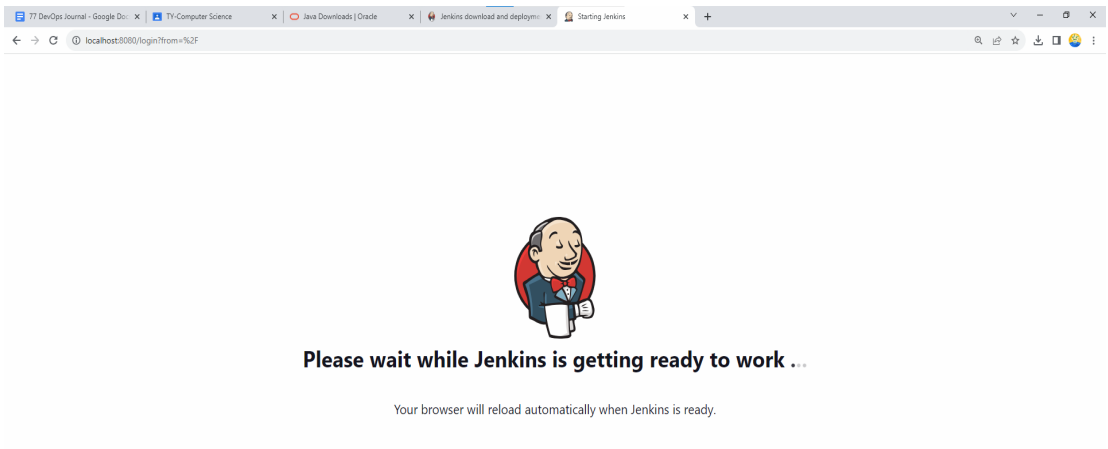

Recent Downloads
jenkins.war
↓ 69.9/85.4 MB • 2 minutes left
jdk-8u381-windows-x64.exe
146 MB • 13 minutes ago

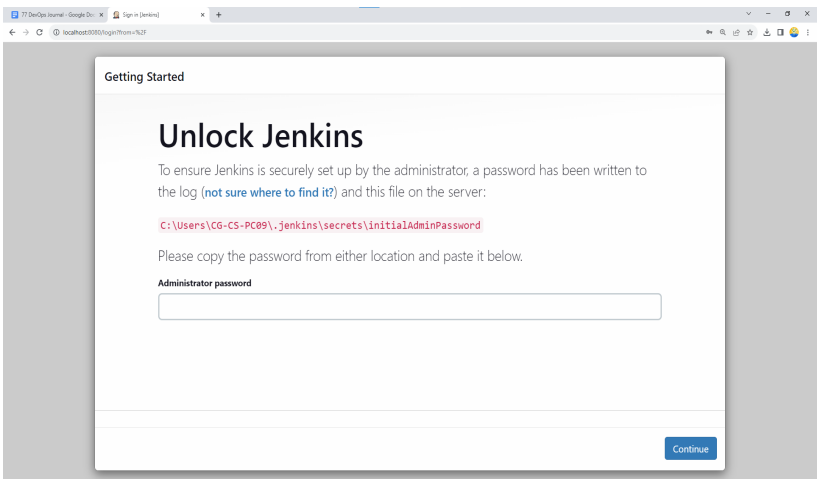

Command Prompt - java -jar Jenkins.war
Microsoft Windows [Version 10.0.19044.3086]
(c) Microsoft Corporation. All rights reserved.
C:\Users\CG-CS-PC09>cd Downloads
C:\Users\CG-CS-PC09\Downloads>java -jar Jenkins.war
Running from: C:\Users\CG-CS-PC09\Downloads\jenkins.war
webroot: C:\Users\CG-CS-PC09\.jenkins\war
2023-08-22 03:50:44.627+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2023-08-22 03:50:45.861+0000 [id=1] WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2023-08-22 03:50:45.892+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-10.0.15; built: 2023-04-11T17:25:14.480Z; git: 68017dbd00236bb7e187330d7585a059610f661d; jvm 17.0.8+9-LTS-211
2023-08-22 03:50:46.271+0000 [id=1] INFO o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
2023-08-22 03:50:46.302+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: Session workerName=node0
2023-08-22 03:50:46.741+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: C:\Users\CG-CS-PC09\.jenkins found at: $user.home/.jenkins
2023-08-22 03:50:46.834+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started w.@76c548f{Jenkins v2.419,/,file:///C:/Users/CG-CS-PC09/.jenkins/war/,AVAILABLE}{C:\Users\CG-CS-PC09\.jenkins\war}
2023-08-22 03:50:46.865+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started ServerConnector@3081f72c{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
2023-08-22 03:50:46.865+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started Server@7e5c856f{STARTING}[10.0.15,sto=0] @2576ms
2023-08-22 03:50:46.865+0000 [id=28] INFO winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2023-08-22 03:50:47.067+0000 [id=35] INFO jenkins.InitReactorRunner$1#onAttained: Started initialization
2023-08-22 03:50:47.077+0000 [id=55] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2023-08-22 03:50:47.704+0000 [id=36] INFO jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2023-08-22 03:50:47.704+0000 [id=43] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins
2023-08-22 03:50:47.724+0000 [id=46] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2023-08-22 03:50:47.877+0000 [id=36] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
2023-08-22 03:50:47.877+0000 [id=42] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
2023-08-22 03:50:47.877+0000 [id=53] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2023-08-22 03:50:47.877+0000 [id=54] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2023-08-22 03:50:47.917+0000 [id=69] INFO hudson.util.Retrier#start: Attempt #1 to do the action check updates server
2023-08-22 03:50:48.182+0000 [id=46] 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:
ec7776d929f349a1956dd2948bc0d448
This may also be found at: C:\Users\CG-CS-PC09\.jenkins\secrets\initialAdminPassword

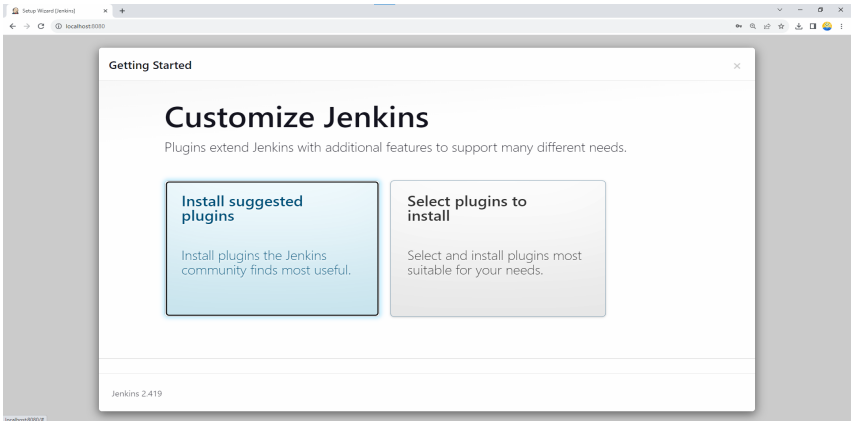
```
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

ec7776d929f349a1956dd2948bc0d448

This may also be found at: C:\Users\CG-CS-PC09\.jenkins\secrets\initialAdminPassword

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

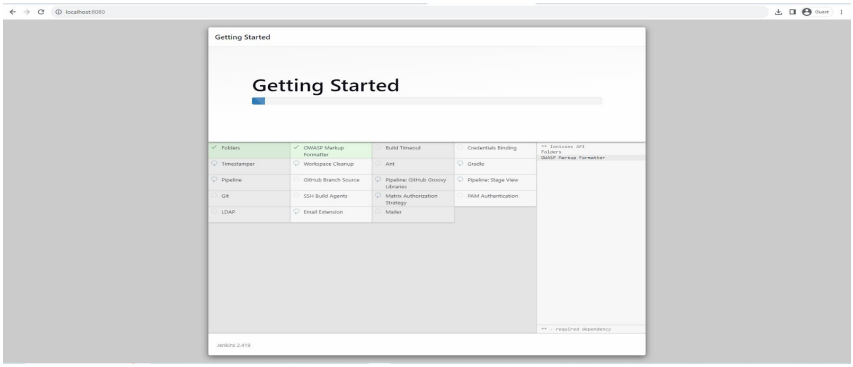
Jenkins Password: ec7776d929f349a1956dd2948bc0d448

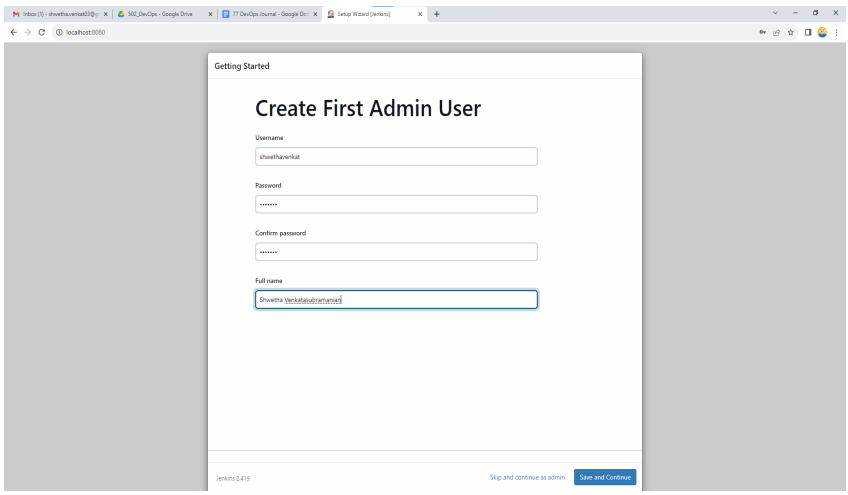
Setup Jenkins



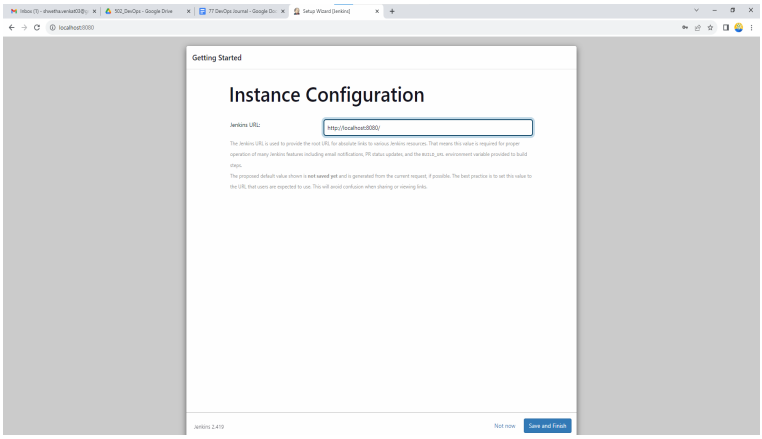


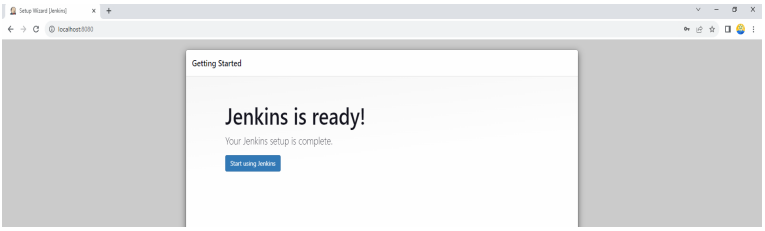


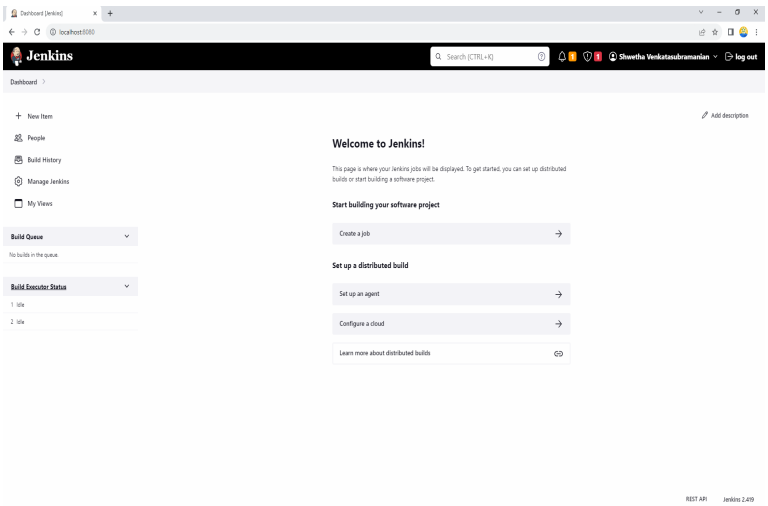




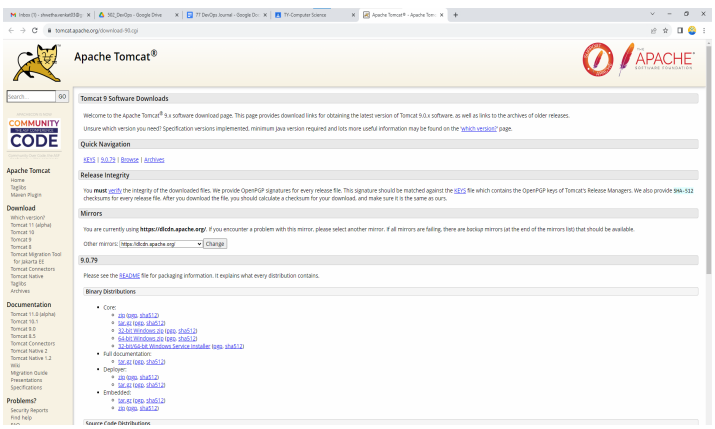
Password: PSP@123

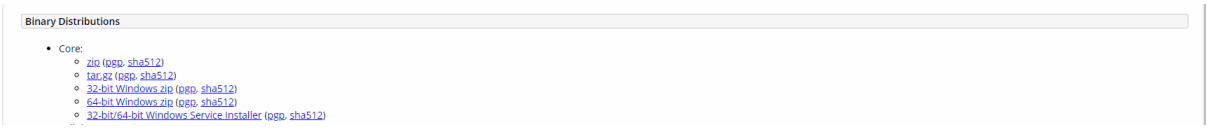


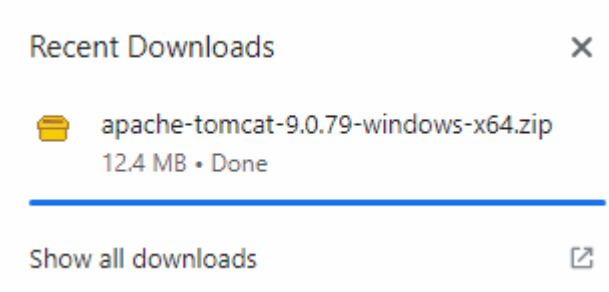




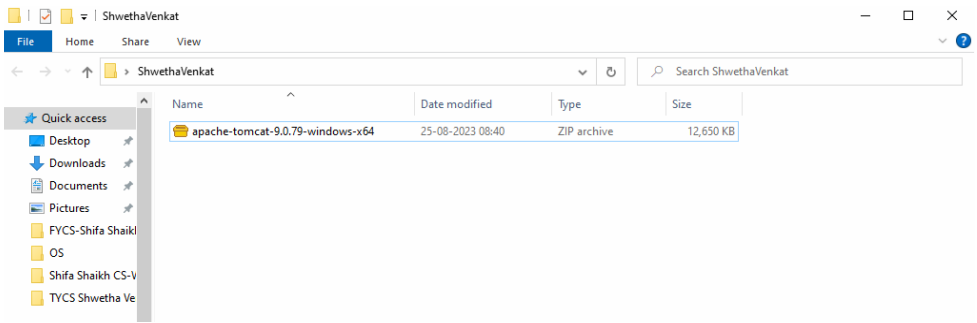
Part-2: Installing Tomcat Server

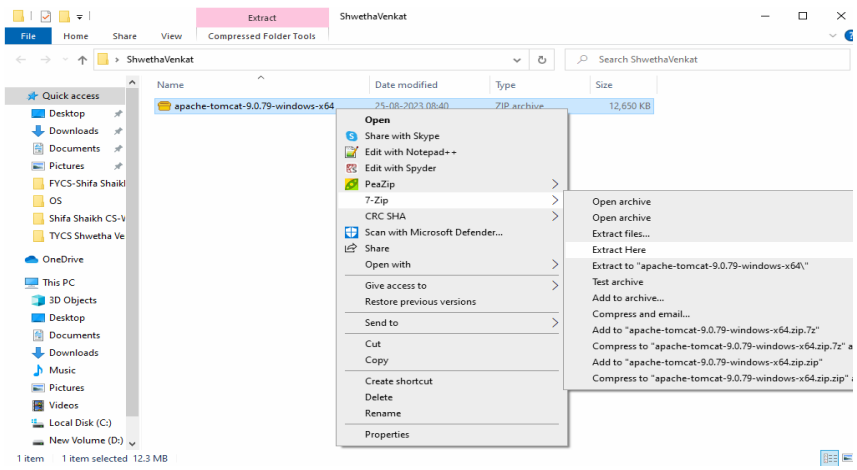


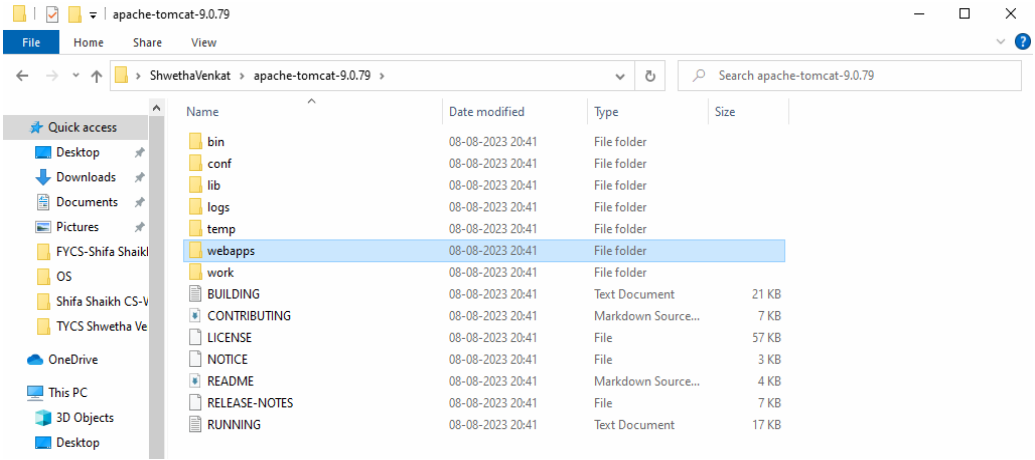




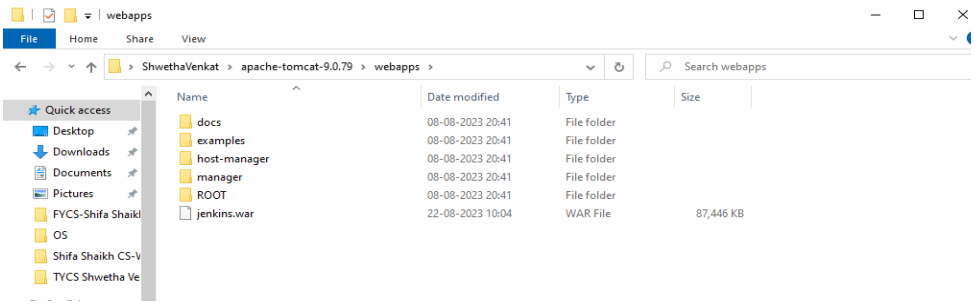
Unzip the Apache Tomcat zip file inside the folder with your name

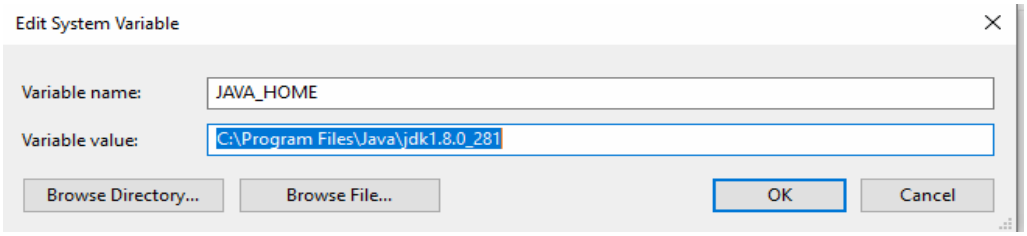


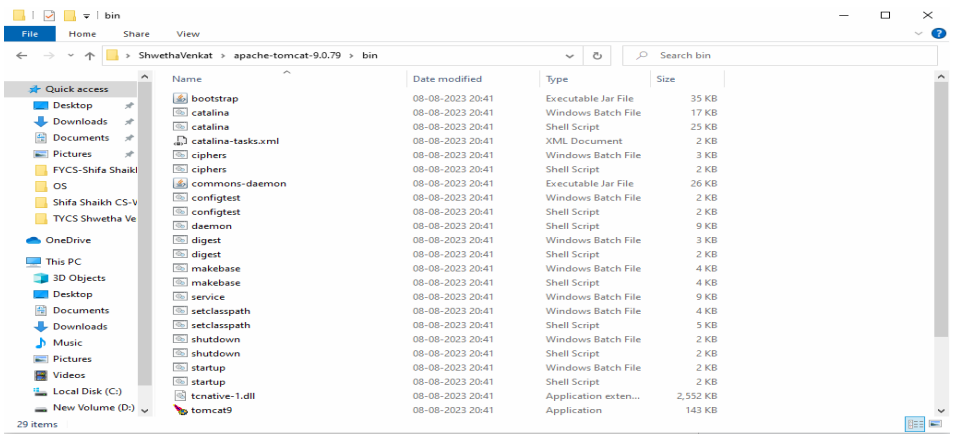


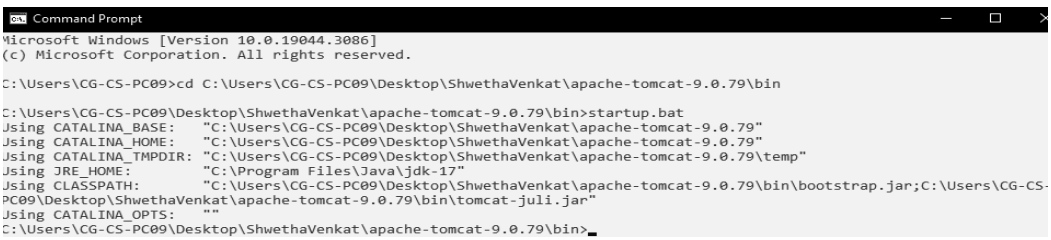


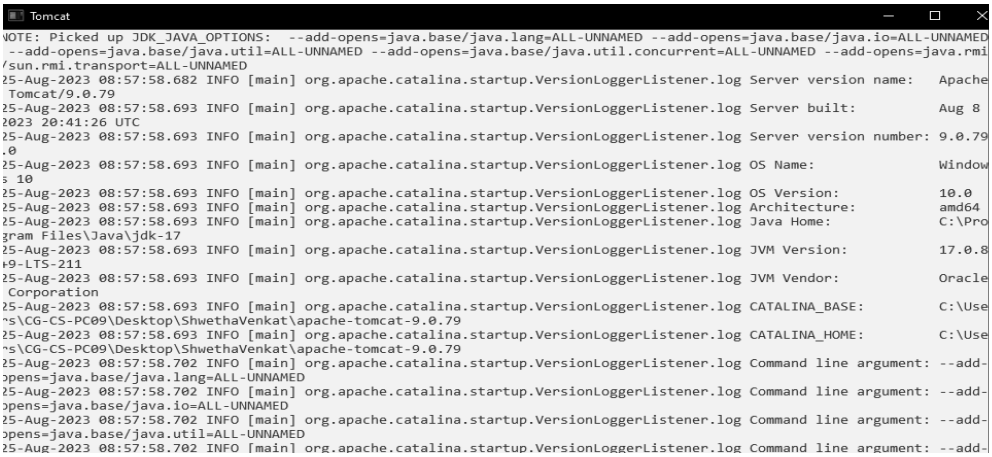
Copy and Paste jenkins.war file inside the webapps folder

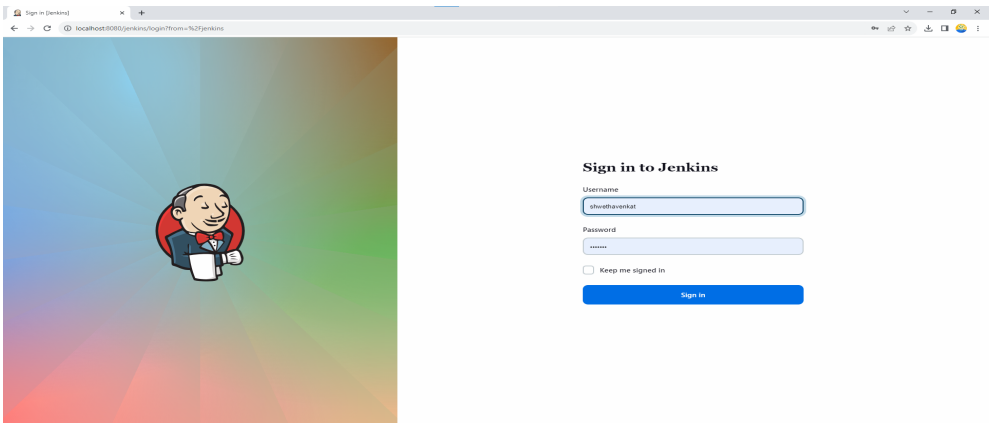






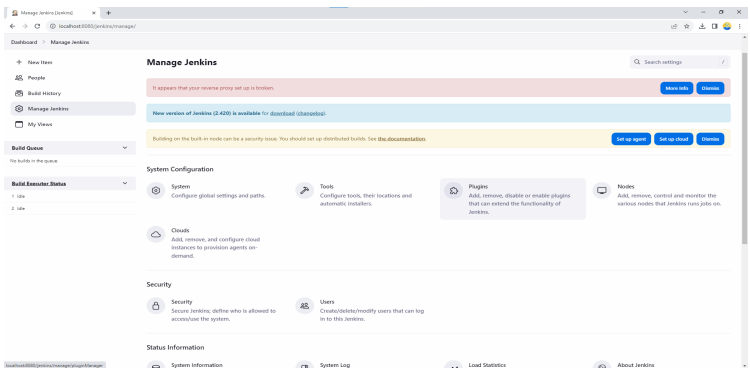




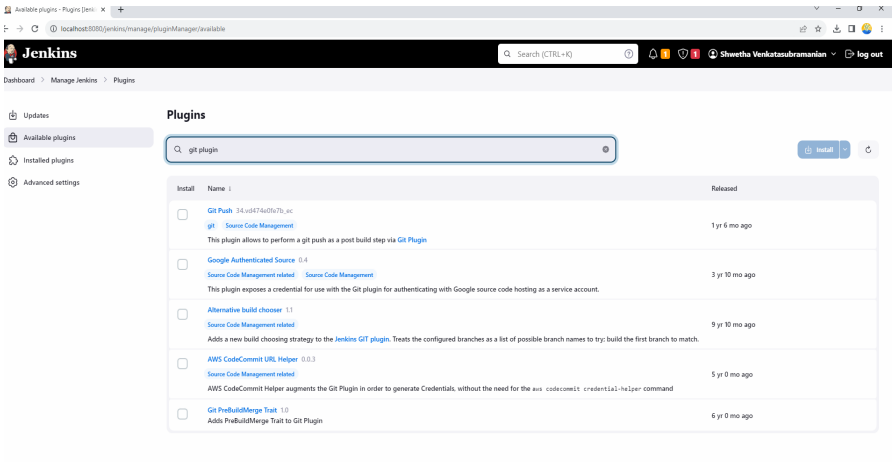


Continuous Integration with Jenkins

1. Go to Manage Jenkins > Plugins > Available Plugins

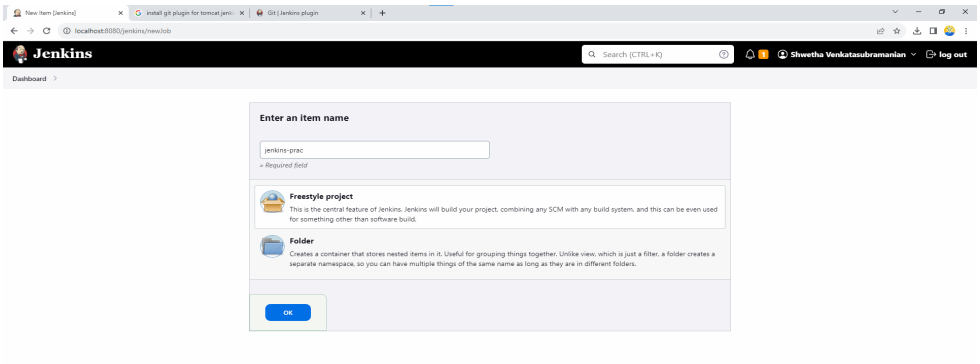


2. Search for Git Plugin

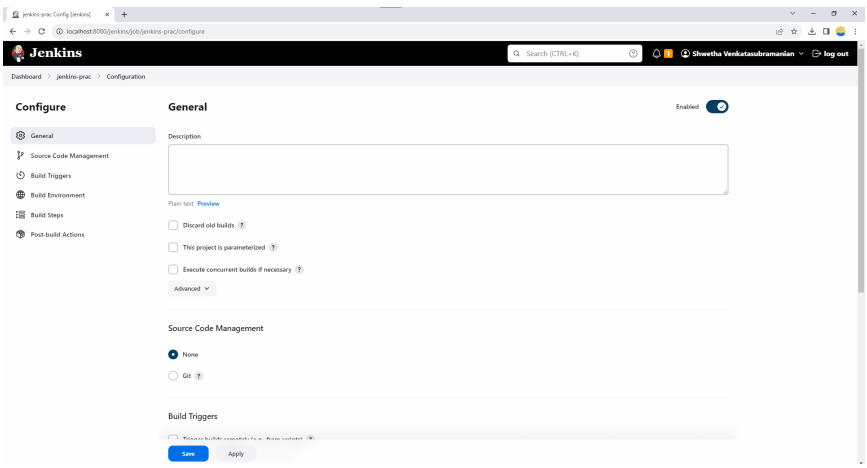


3. Select the checkmark and click on install if it is not already installed.

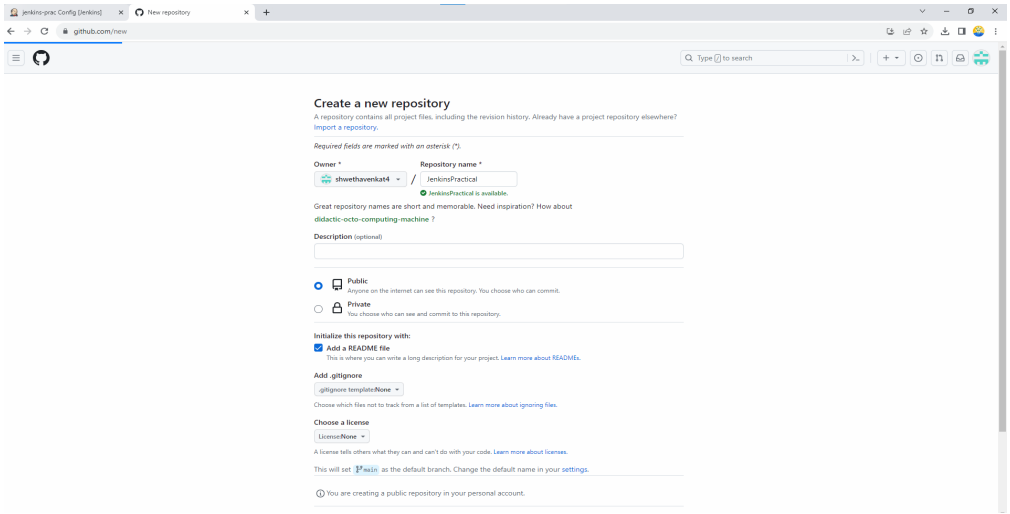
4. Create a new Freestyle Project

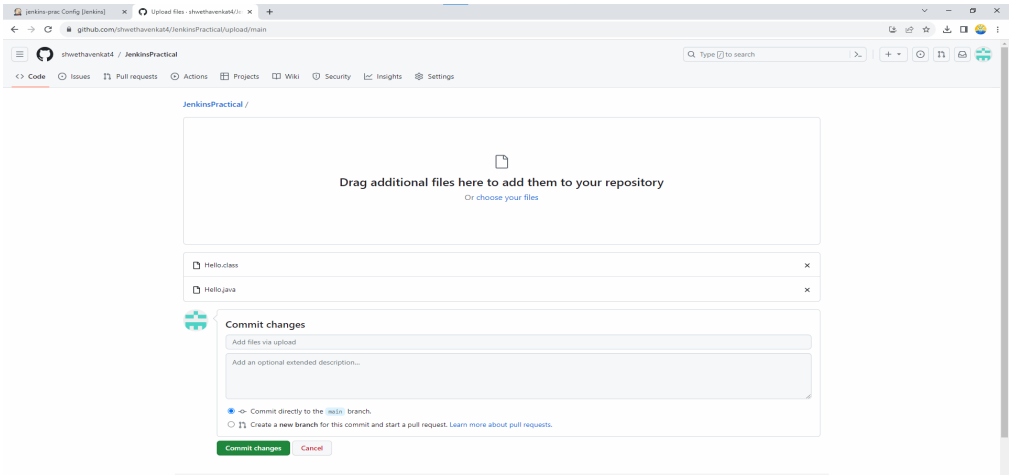


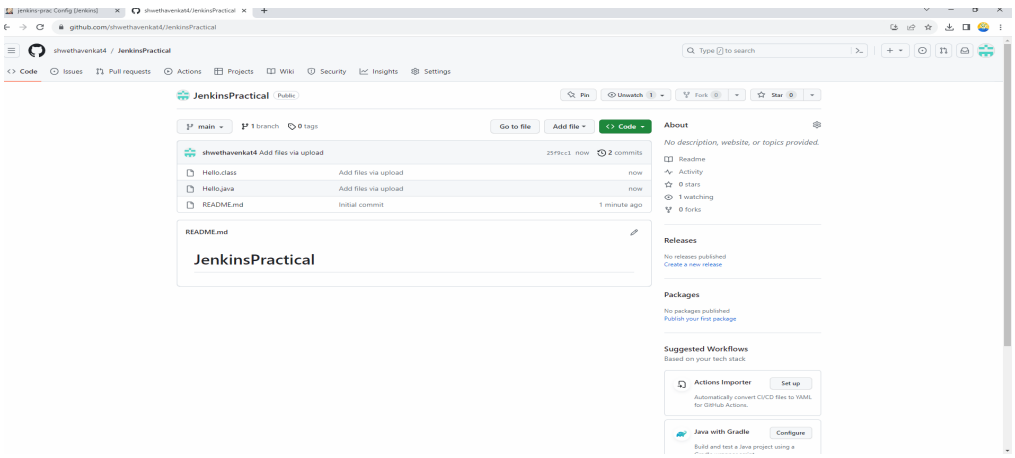
5. Select Git under the Source Code Management section



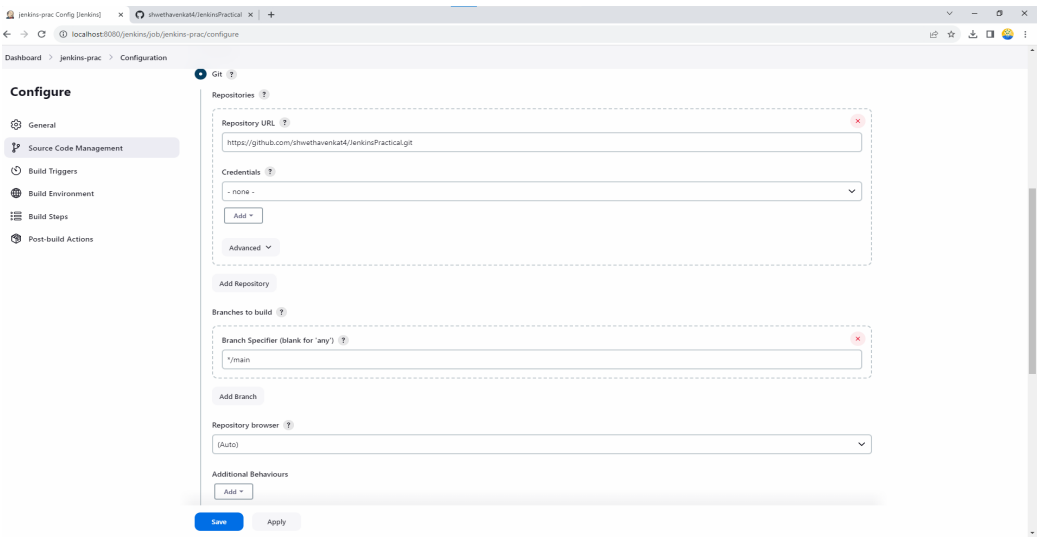
6. Create a new repository on Git



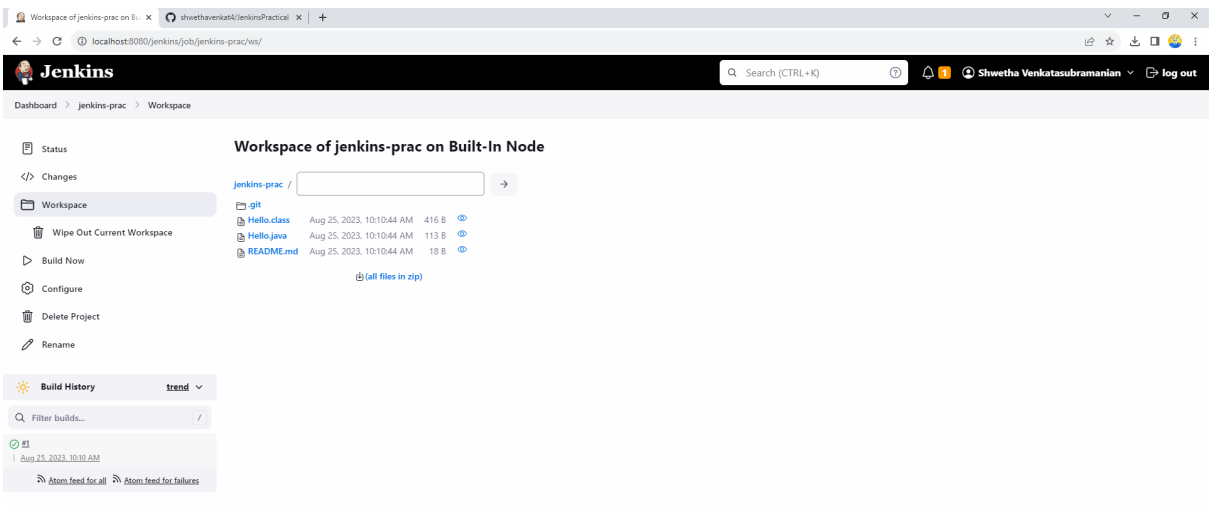


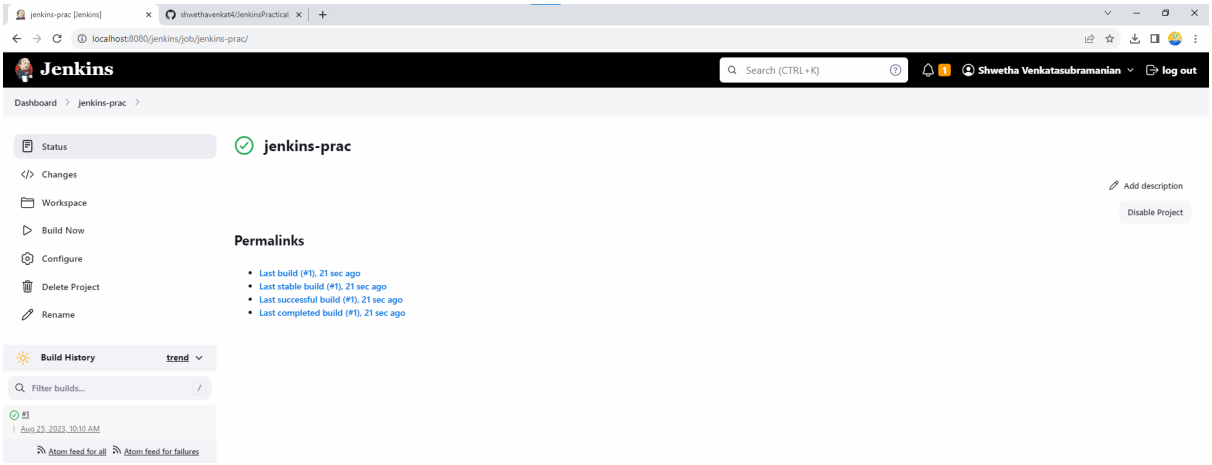


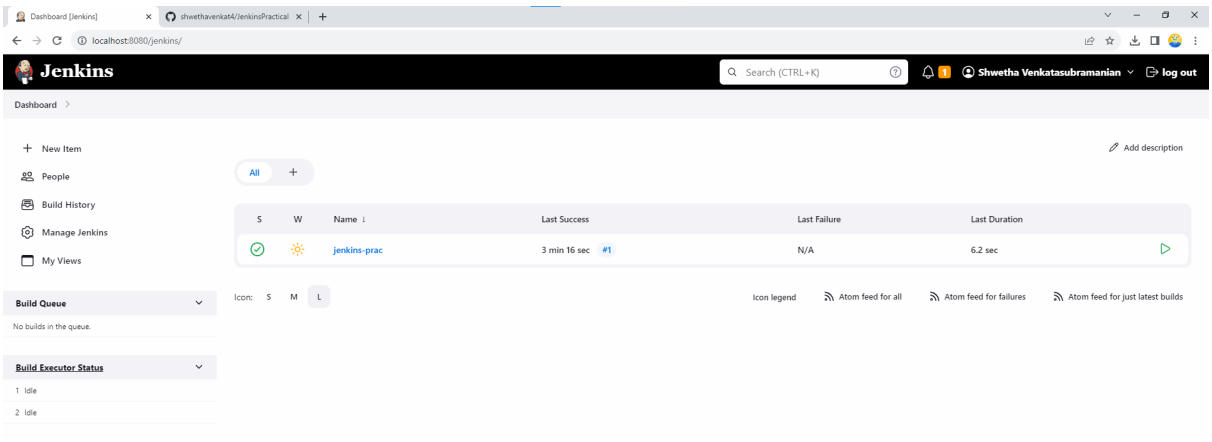
7. Copy the link to the source code and paste it on Repository URL and click on save



8. Click on Build

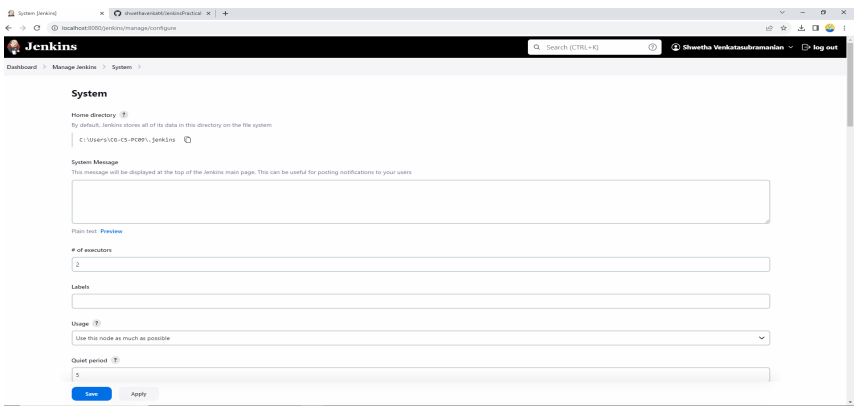




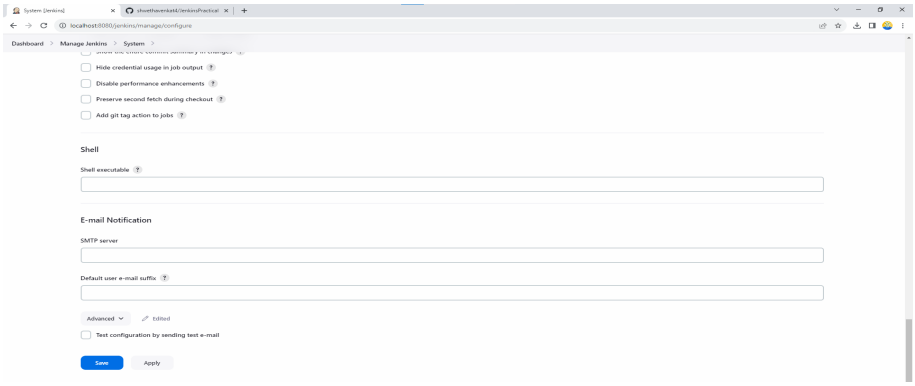


Setting up Email Notifications

1. Go to Manage Jenkins > System

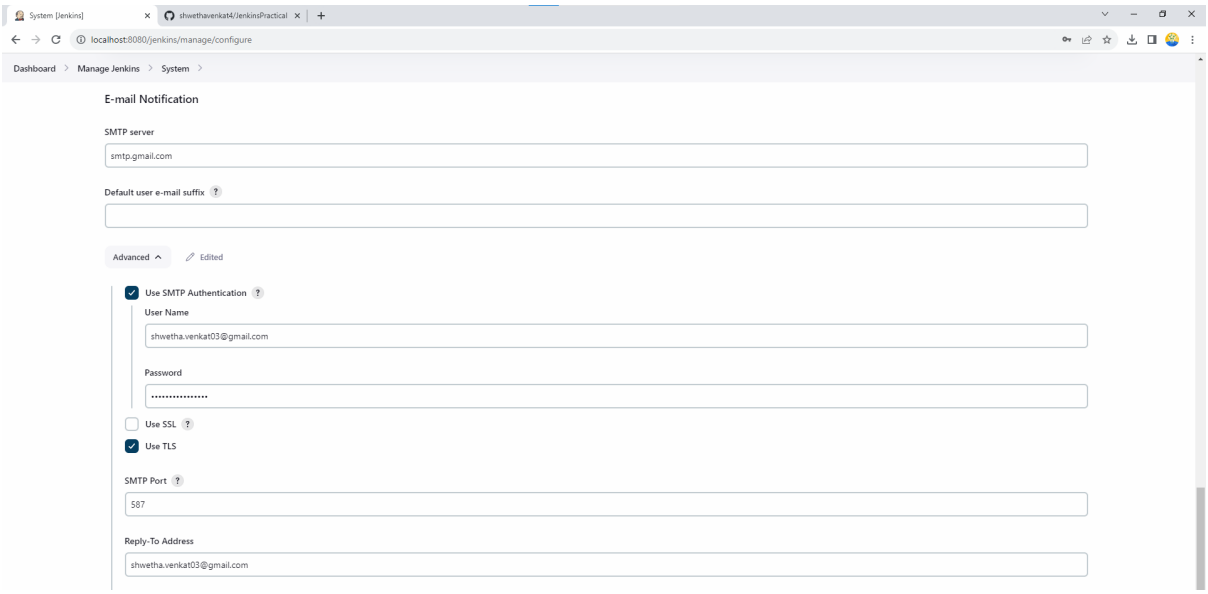


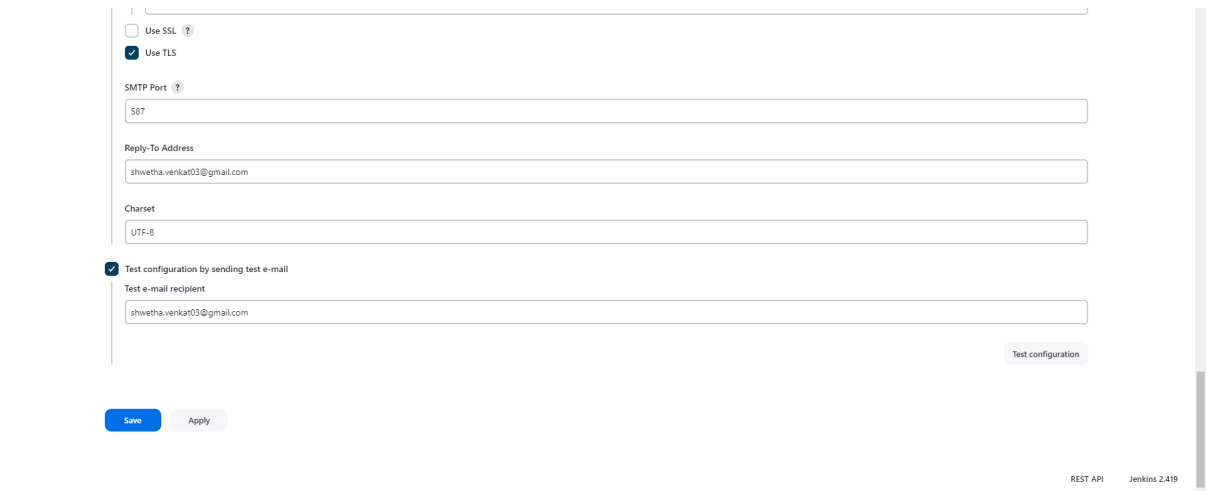
2. Scroll down to Email Notifications

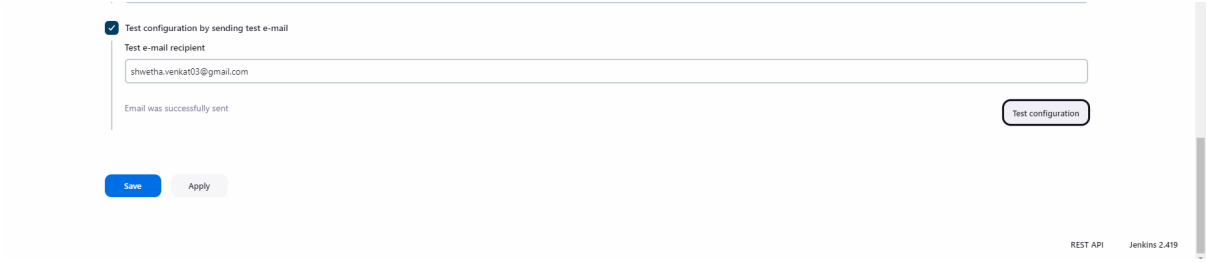


3. Create an App password in your Google Account under Manage Account > Security > 2-Factor Authentication

4. Enter your email ID and the App Password as shown below and click on save

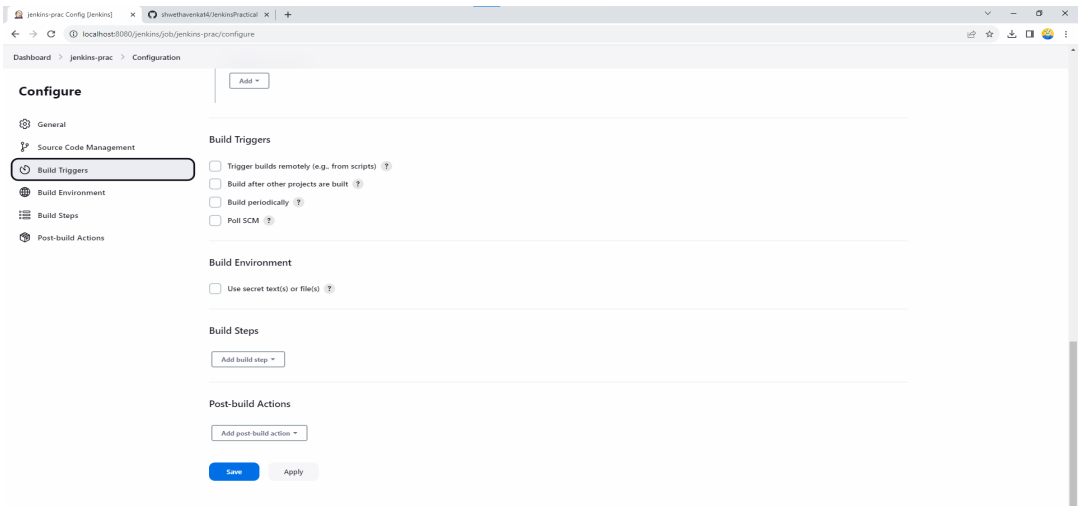






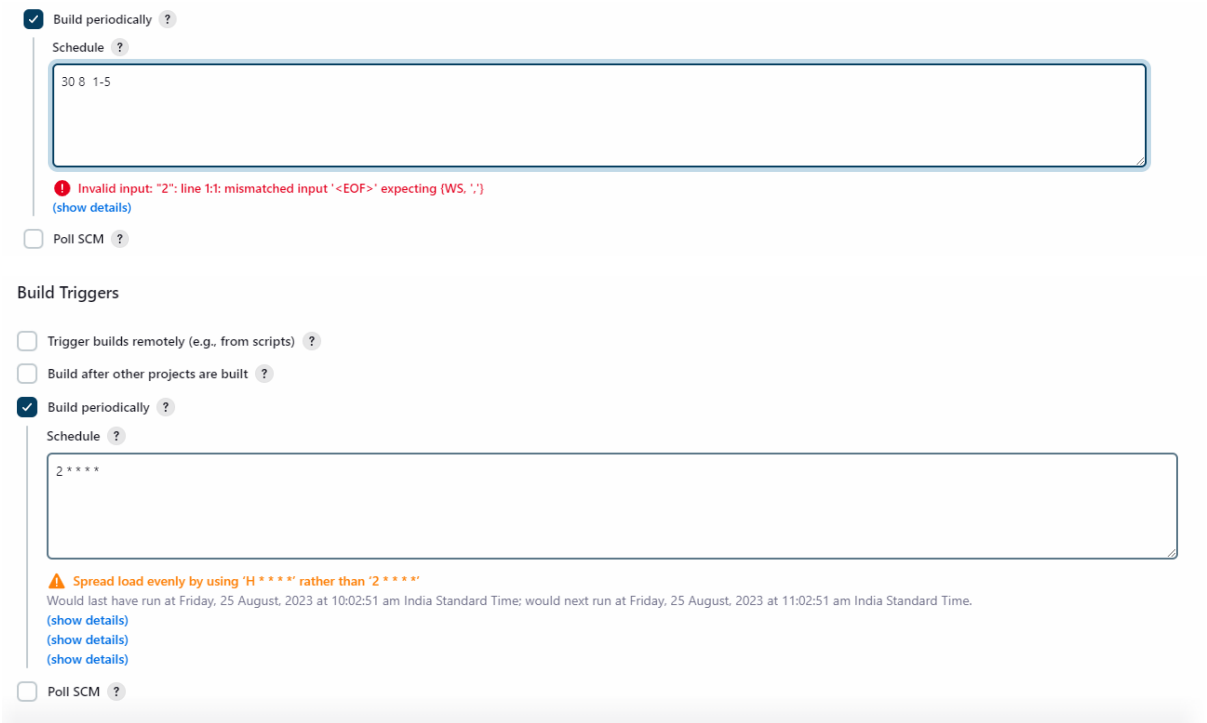
Scheduling Automated Builds

1. Go to Your Project Name > Build Triggers

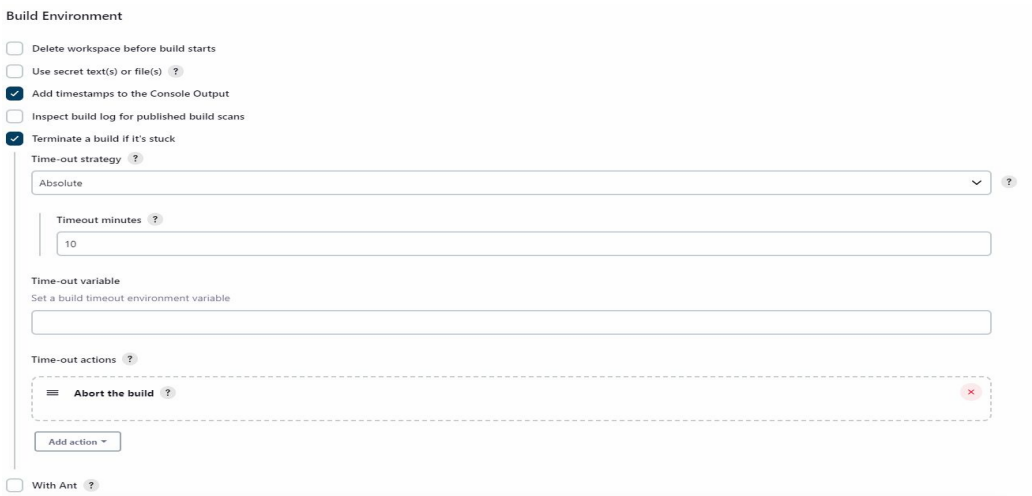


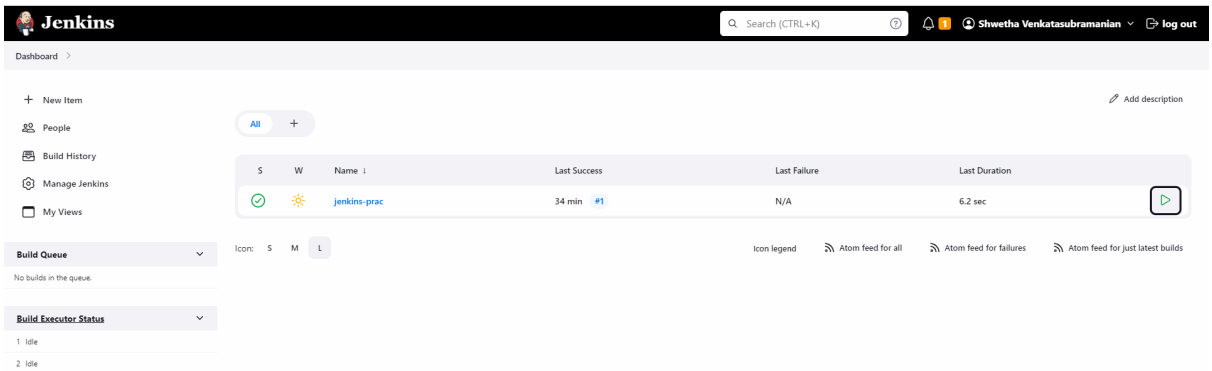
2. Check the build periodically option

3. Enter a build schedule



4. Check the Add Timestamps and Terminate a build if it's stuck options

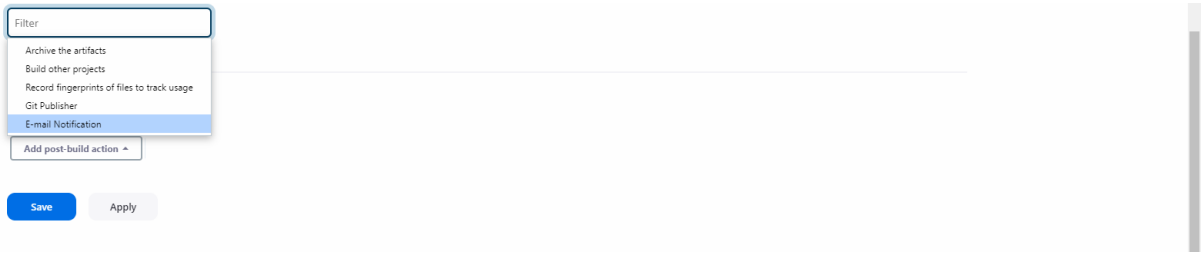




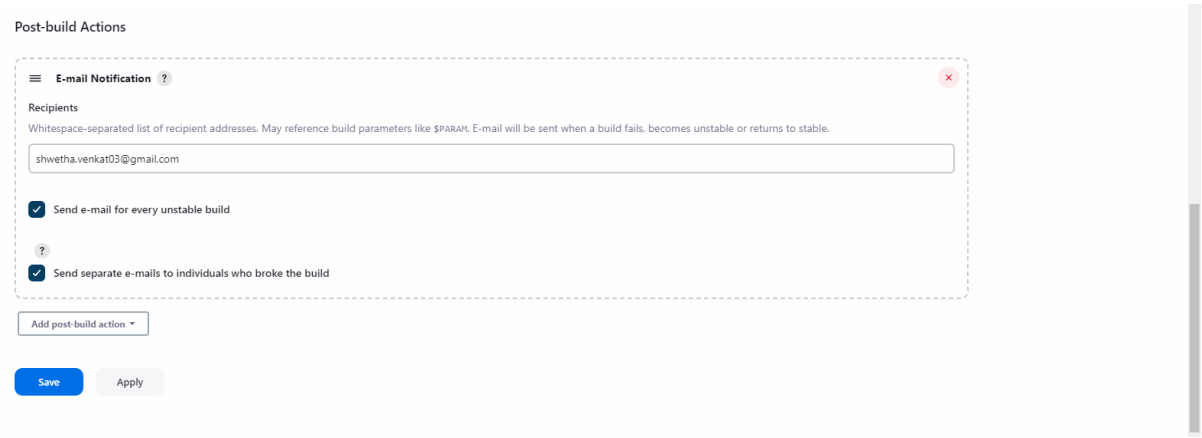
Email Notifications for Automated Builds

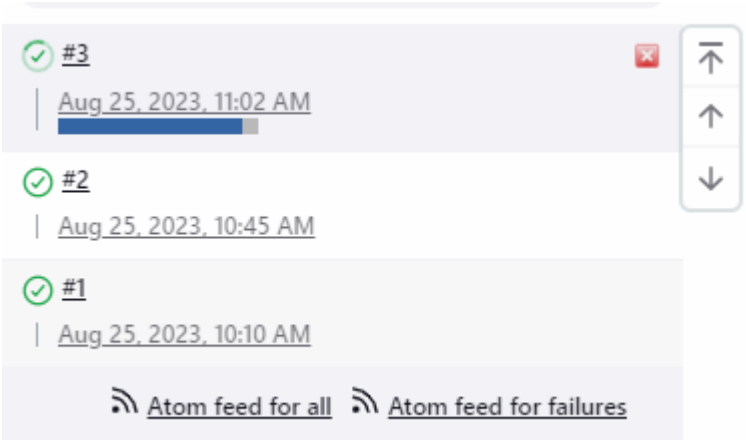
1. Go to Your Project Name > Post Build Actions

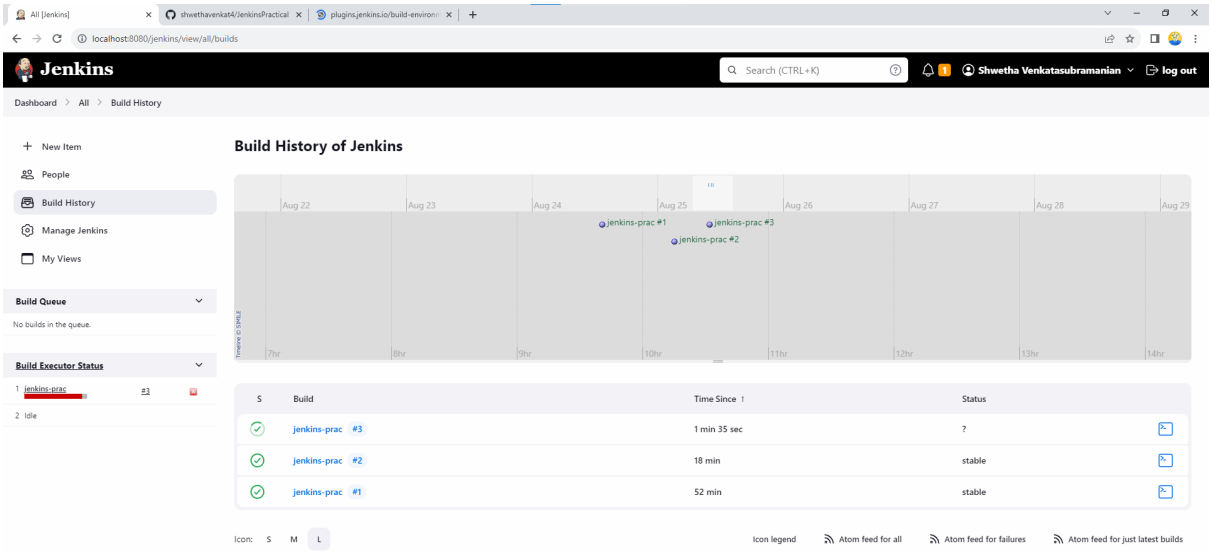
2. Select Email Notifications from the drop-down menu



3. Enter your Email ID as shown below






All [Jenkins]
shwethavenkat4/JenkinsPractical
plugins.jenkins.io/build-environm
localhost:8080/jenkins/view/all/builds
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
All
Build History
New Item
People
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 jenkins-prac #3
2 Idle
Build History of Jenkins
Aug 22
Aug 23
Aug 24
Aug 25
Aug 26
Aug 27
Aug 28
Aug 29
jenkins-prac #1
jenkins-prac #3
jenkins-prac #2
7hr
8hr
9hr
10hr
11hr
12hr
13hr
14hr
S
Build
Time Since
Status
jenkins-prac #3
1 min 35 sec
?
jenkins-prac #2
18 min
stable
jenkins-prac #1
52 min
stable
Icon: S M L
Icon legend
Atom feed for all
Atom feed for failures
Atom feed for just latest builds

Practical 2b:

Implementing CI using Circle CI

Circle CI is a powerful Continuous Integration tool that automates the entire build and deployment process. CircleCI leverages containers to run multiple builds and deployments in parallel.

Circle CI continuously deploys your code to the docker container registry and enables you to track the status of your builds. Apply frontend, backend, and serverless architecture patterns by leveraging Circle CI's intuitive and powerful CI/CD features that automate everything from managing Docker images and trays, to deploying services on Kubernetes.

CircleCI is a popular tool for building and deploying software. It offers a simple and flexible way to automate your workflow, allowing you to focus on writing code instead of managing infrastructure. In this blog, we will walk through the steps to create CI/CD pipeline using CircleCI

Key Features of CircleCI

- **Configuration as Code**
    - CircleCI uses YAML configuration files (.circleci/config.yml) to define the steps and workflows of the CI/CD pipeline.
- **Support for Multiple Platforms**
    - Supports building on macOS, Linux, Windows, and Docker containers.
- **Integration with Version Control Systems**
    - Integrates with GitHub, Bitbucket, and GitLab, allowing automatic triggering of workflows on code commits or pull requests.
- **Parallelism and Performance**
    - Offers parallel execution to speed up the build and test process.
    - Provides insights and metrics to optimize build performance.
- **Environment Variables and Secrets Management**
    - Allows the use of environment variables for configuring builds and managing sensitive information securely.
- **Workflows**
    - Enables the orchestration of multiple jobs with dependencies and scheduling.
- **Caching and Artifacts**
    - Supports caching dependencies and storing build artifacts, reducing redundant work and speeding up builds.

Step 1: Download demo project from https://gitlab.com/cci-devrel-demos/python-app-base/-/tree/main?ref_type=heads

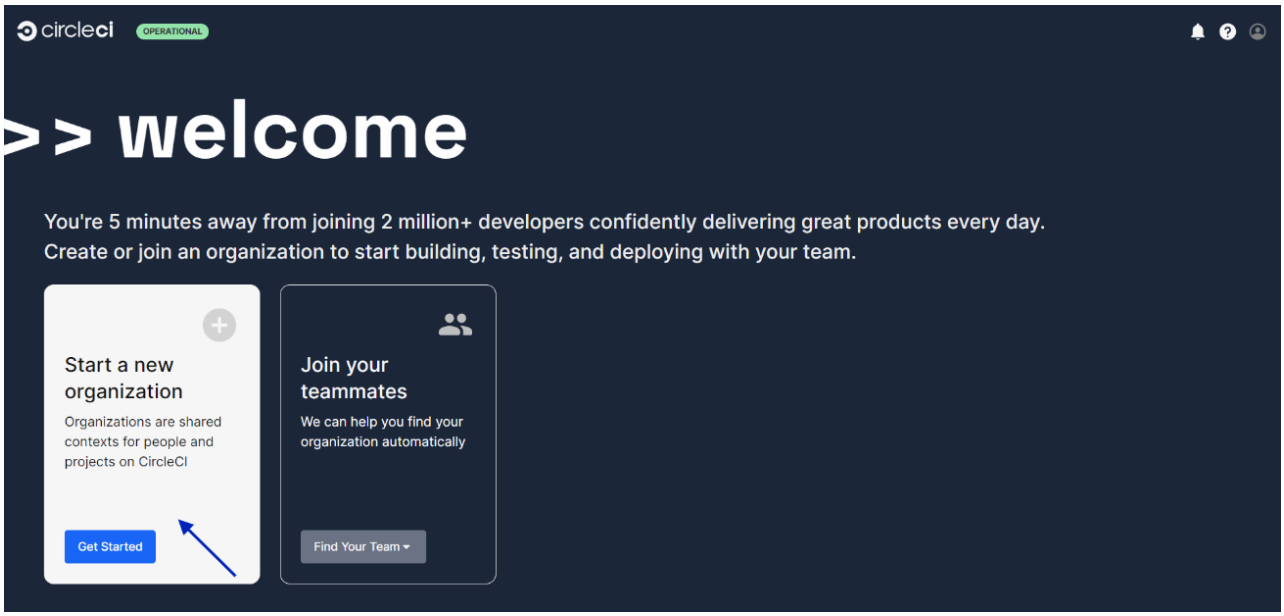
Step 2: Create account on gitlab

Step 3: Create a new project and add readme file. This project should be public.

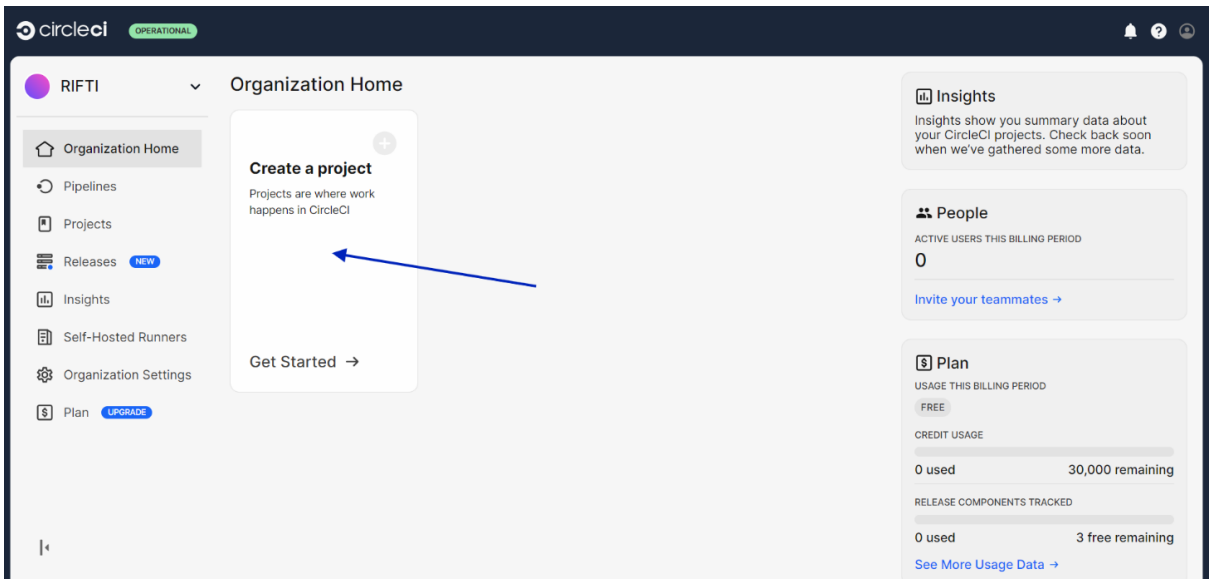
Step 4: Upload files to gitlab project.

Step 5: Create a new directory .circleci in the top level of the project. Now, create a new file config.yml in that directory (.circleci/config.yml) — this is your main configuration file for CircleCI.

Step 6: create account on circle ci: https://circleci.com/signup/



Step 7: Give any name of organization



Enter project details

We'll help you set up a project to build, test, and deploy a software application.

What would you like to name your project?

CircleCI Practical

Project names must:
- be 3-40 characters long
- contain only letters, numbers, or - _ . : ! & + [ ]
- begin with a letter

← Go back Next: set up a pipeline →

What would you like to do in your project?

Build, test, and deploy your software application

Build, test, **evaluate** and deploy your **GenAI-enabled** software application

Something else

Enter project details

We'll help you set up a project to build, test, and deploy a software application.

What would you like to name your project?

CircleCI Practical

Project names must:
- be 3-40 characters long
- contain only letters, numbers, or - _ . : ! & + [ ]
- begin with a letter

← Go back Next: set up a pipeline →

CircleCI projects get work done by running **pipelines.**

Pipeline
Pipelines orchestrate executable commands and scripts for your CI/CD process

Code
Pipelines can checkout code from VCS repositories

Config
Config files use YAML to define what runs during your pipeline

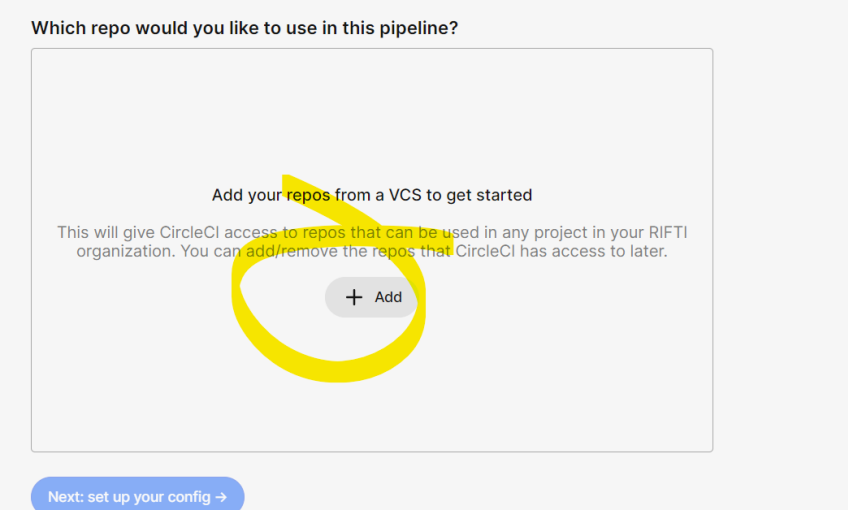
Triggers
Triggers automatically run your pipelines when an event occurs

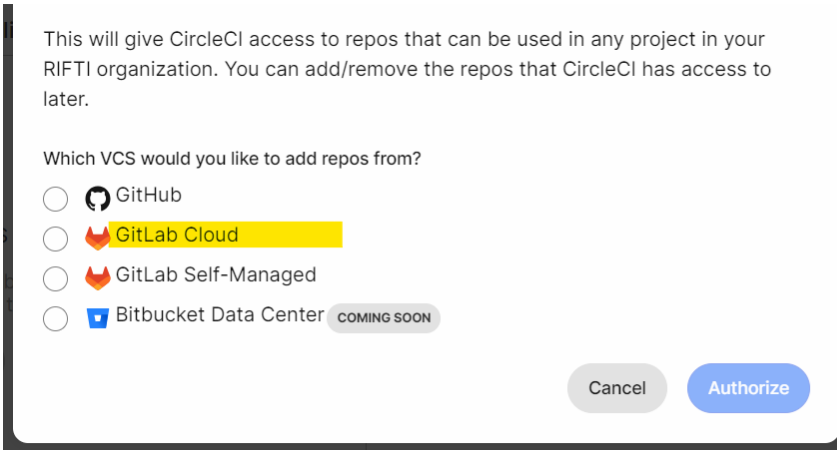
What would you like to name your project's first pipeline?

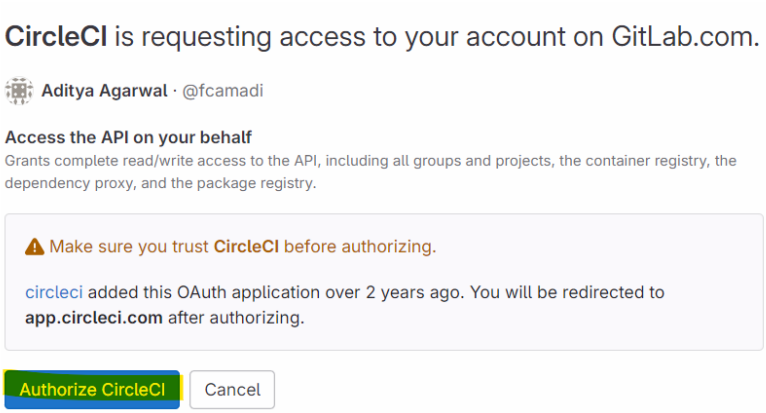
build-and-test

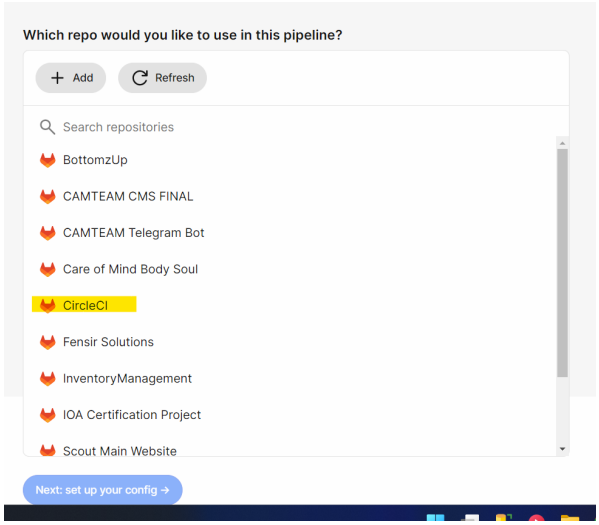
We suggest naming pipelines after the work they will perform, e.g. integration-tests

Next: choose a repo →


Which repo would you like to use in this pipeline?
Add your repos from a VCS to get started
This will give CircleCI access to repos that can be used in any project in your RIFTI organization. You can add/remove the repos that CircleCI has access to later.
Add
Next: set up your config →


This will give CircleCI access to repos that can be used in any project in your RIFTI organization. You can add/remove the repos that CircleCI has access to later.
Which VCS would you like to add repos from?
GitHub
GitLab Cloud
GitLab Self-Managed
Bitbucket Data Center
COMING SOON
Cancel
Authorize


CircleCI is requesting access to your account on GitLab.com.
Aditya Agarwal · @fcamadi
Access the API on your behalf
Grants complete read/write access to the API, including all groups and projects, the container registry, the dependency proxy, and the package registry.
Make sure you trust CircleCI before authorizing.
circleci added this OAuth application over 2 years ago. You will be redirected to app.circleci.com after authorizing.
Authorize CircleCI
Cancel



Create New Project

Connect to your GitLab.com repository and start running pipelines.

Connect to

GitLab.com — Connected to fcamadi

Repository

CircleCI

`.circleci/config.yml` found

We will listen for changes and setup a configuration file in this repository. Not seeing a repository? Check that you have the correct level of permissions on the repository.

Project Name

CircleCI

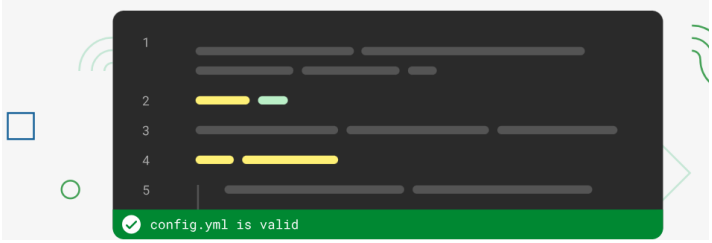
Enter 3-40 letters, numbers, or - _ . : ! & + [ ]. Project name cannot begin with a number or special character.

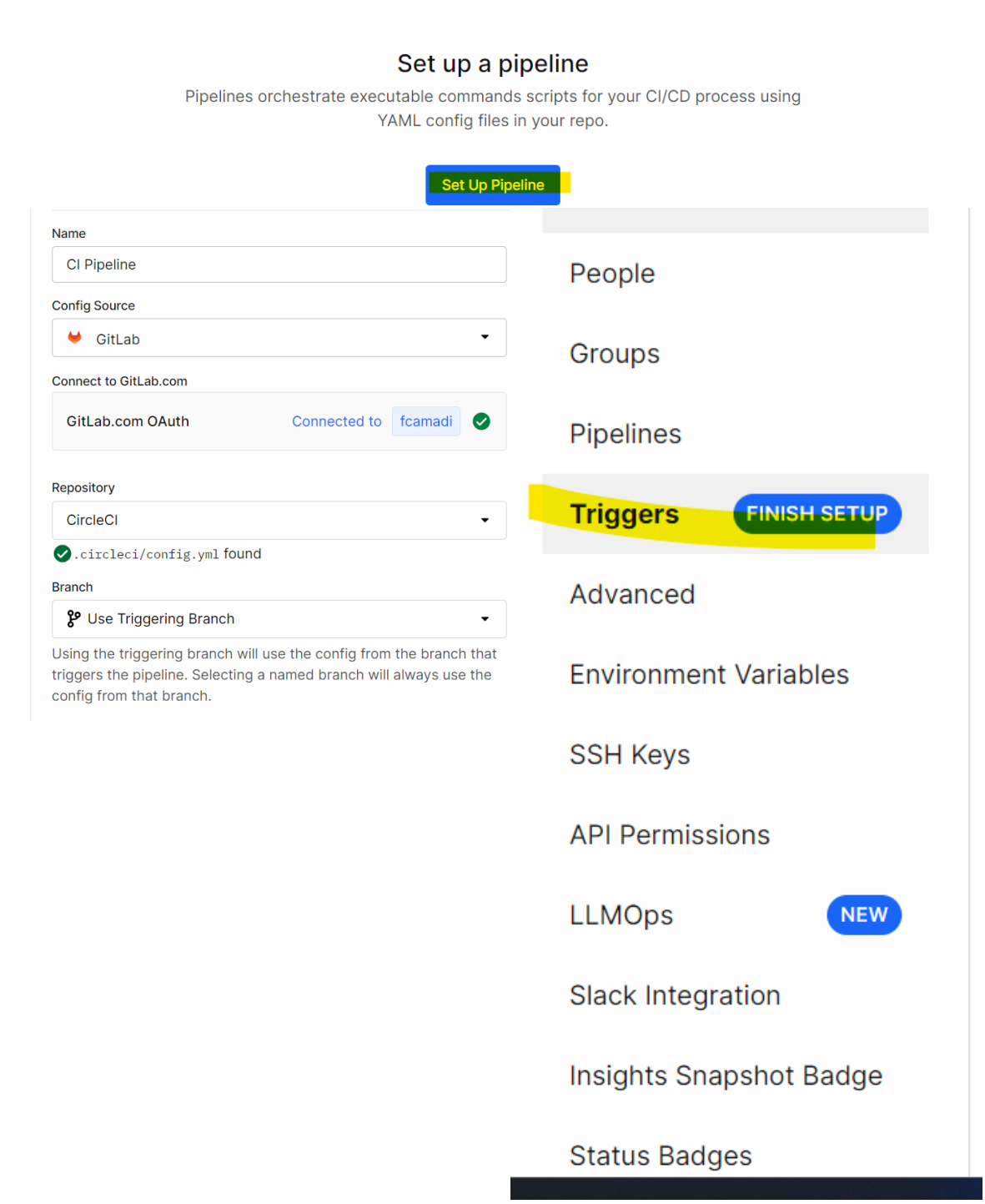
Create Project | Cancel

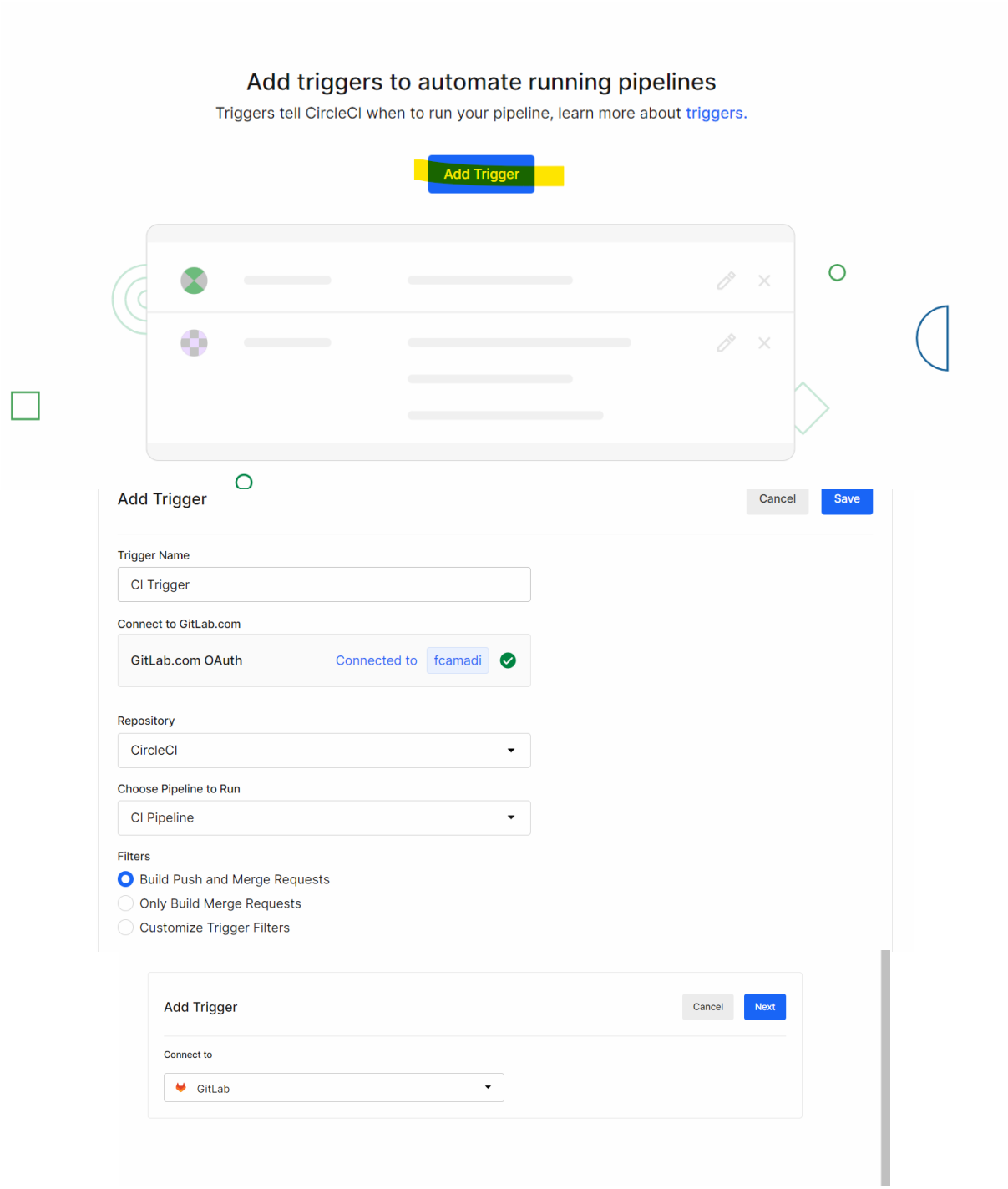
Set up configuration to start running pipelines

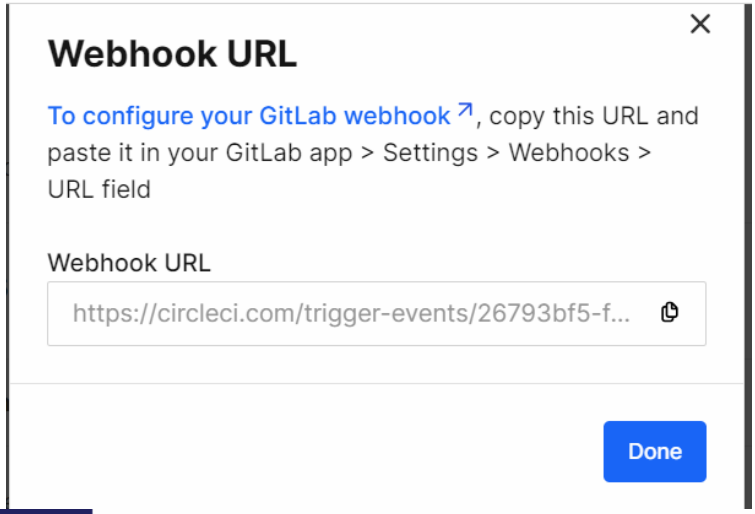
The configuration file determines how your pipelines will run. Learn more about configuration.

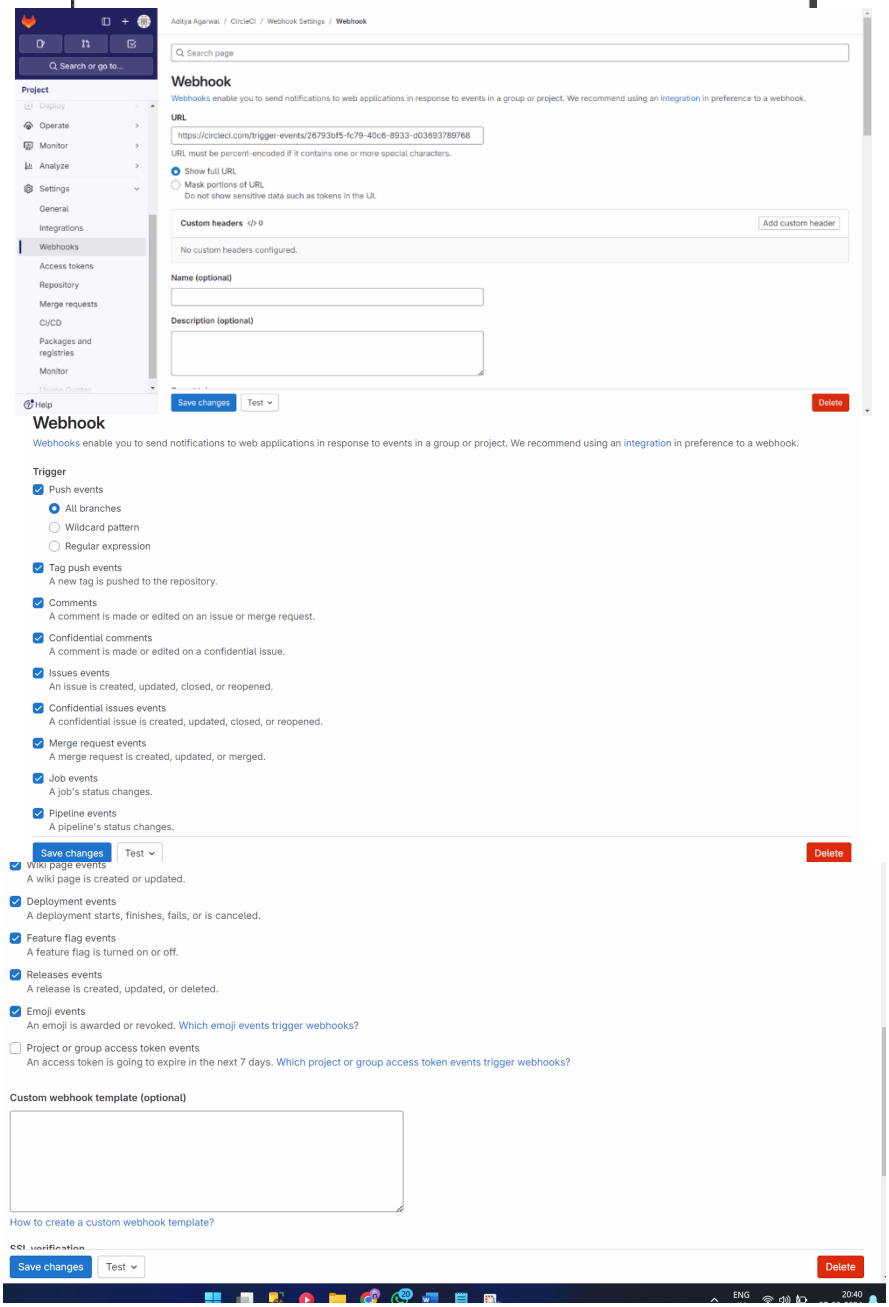
Set Up Configuration




Set up a pipeline
Pipelines orchestrate executable commands scripts for your CI/CD process using YAML config files in your repo.
Set Up Pipeline
Name
CI Pipeline
Config Source
GitLab
Connect to GitLab.com
GitLab.com OAuth
Connected to
fcamadi
Repository
CircleCI
.circleci/config.yml found
Branch
Use Triggering Branch
Using the triggering branch will use the config from the branch that triggers the pipeline. Selecting a named branch will always use the config from that branch.
People
Groups
Pipelines
Triggers
FINISH SETUP
Advanced
Environment Variables
SSH Keys
API Permissions
LLMOps
NEW
Slack Integration
Insights Snapshot Badge
Status Badges


Add triggers to automate running pipelines
Triggers tell CircleCI when to run your pipeline, learn more about triggers.
Add Trigger
Add Trigger
Cancel
Save
Trigger Name
CI Trigger
Connect to GitLab.com
GitLab.com OAuth
Connected to
fcamadi
Repository
CircleCI
Choose Pipeline to Run
CI Pipeline
Filters
Build Push and Merge Requests
Only Build Merge Requests
Customize Trigger Filters
Add Trigger
Cancel
Next
Connect to
GitLab


Webhook URL
To configure your GitLab webhook, copy this URL and paste it in your GitLab app > Settings > Webhooks > URL field
Webhook URL
https://circleci.com/trigger-events/26793bf5-f...
Done


Webhook
Webhooks enable you to send notifications to web applications in response to events in a group or project. We recommend using an integration in preference to a webhook.
URL
https://circleci.com/trigger-events/26793bf5-fc79-40c6-8933-d03693789768
URL must be percent-encoded if it contains one or more special characters.
Show full URL
Mask portions of URL
Do not show sensitive data such as tokens in the UI.
Custom headers
Add custom header
No custom headers configured.
Name (optional)
Description (optional)
Save changes
Test
Delete
Webhook
Webhooks enable you to send notifications to web applications in response to events in a group or project. We recommend using an integration in preference to a webhook.
Trigger
Push events
All branches
Wildcard pattern
Regular expression
Tag push events
A new tag is pushed to the repository.
Comments
A comment is made or edited on an issue or merge request.
Confidential comments
A comment is made or edited on a confidential issue.
Issues events
An issue is created, updated, closed, or reopened.
Confidential issues events
A confidential issue is created, updated, closed, or reopened.
Merge request events
A merge request is created, updated, or merged.
Job events
A job's status changes.
Pipeline events
A pipeline's status changes.
Save changes
Test
Delete
Wiki page events
A wiki page is created or updated.
Deployment events
A deployment starts, finishes, fails, or is canceled.
Feature flag events
A feature flag is turned on or off.
Releases events
A release is created, updated, or deleted.
Emoji events
An emoji is awarded or revoked. Which emoji events trigger webhooks?
Project or group access token events
An access token is going to expire in the next 7 days. Which project or group access token events trigger webhooks?
Custom webhook template (optional)
How to create a custom webhook template?
Save changes
Test
Delete

Hello.py

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"
```

==========

Tests.py

```
from hello import app

with app.test_client() as c:
   response = c.get('/')
   assert response.data == b'Hello World!'
   assert response.status_code == 200
```

---

config.yml

```
version: 2.1

jobs:
  test:
    docker:
      - image: cimg/python:3.10.11
    steps:
      - checkout
      - restore_cache:
          key: deps1-{{ .Branch }}-{{ checksum "requirements.txt" }}
      - run:
          name: Install dependencies
          command: |
            python3 -m venv venv
            . venv/bin/activate
            pip install -r requirements.txt
      - save_cache:
```

```
      key: deps1-{{ .Branch }}-{{ checksum "requirements.txt" }}
      paths:
        - "venv"
    - run:
      name: Running tests
      command: |
        . venv/bin/activate
        python3 tests.py
    - store_artifacts:
      path: test-reports/
      destination: python_app

workflows:
  run-tests:
    jobs:
      - test
```

Practical 3:

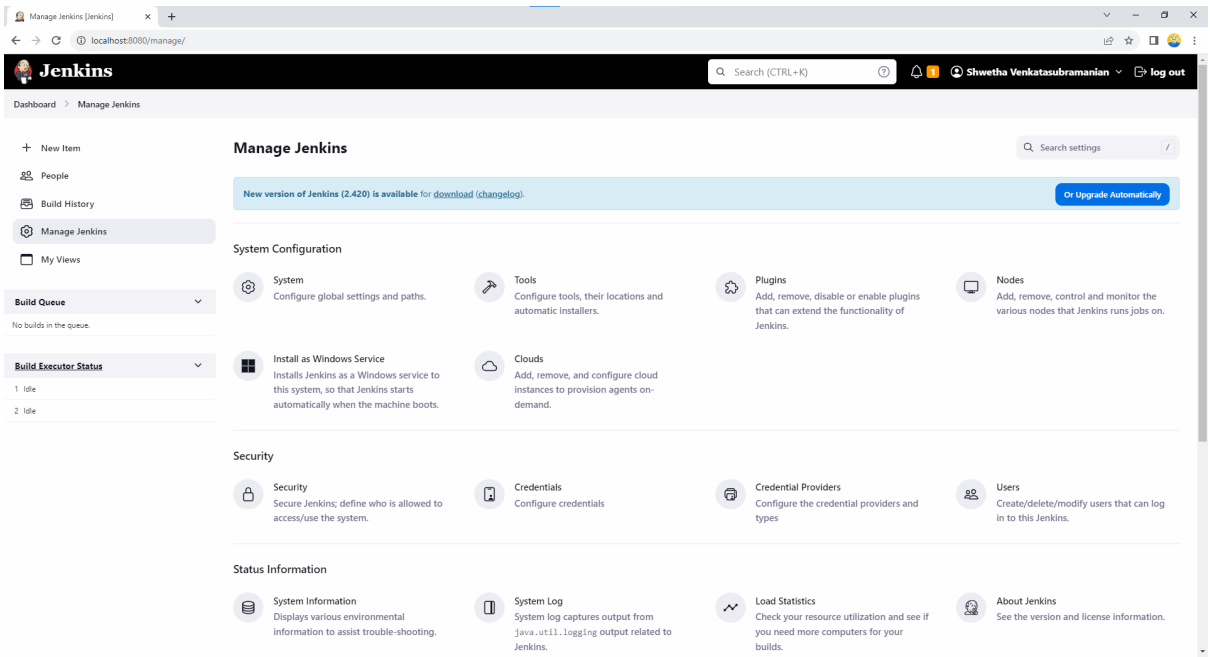
Continuous Delivery Using Jenkins

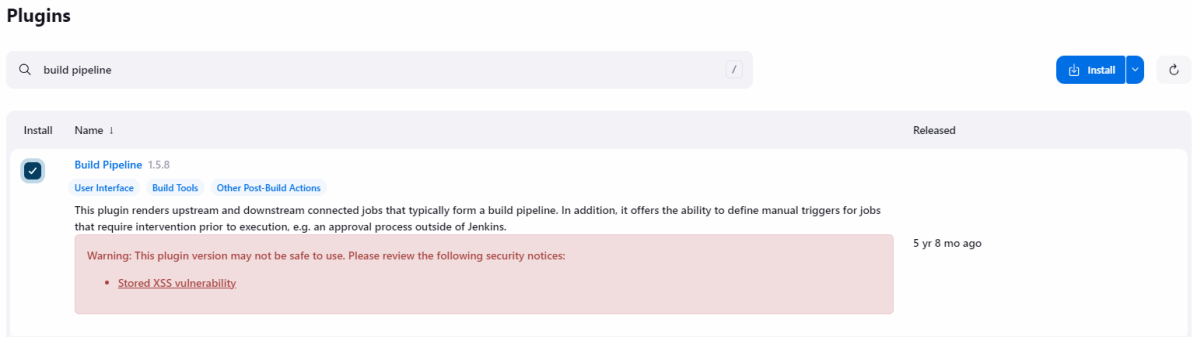
In Jenkins, a pipeline is a collection of events or jobs that are interlinked with one another in a sequence.

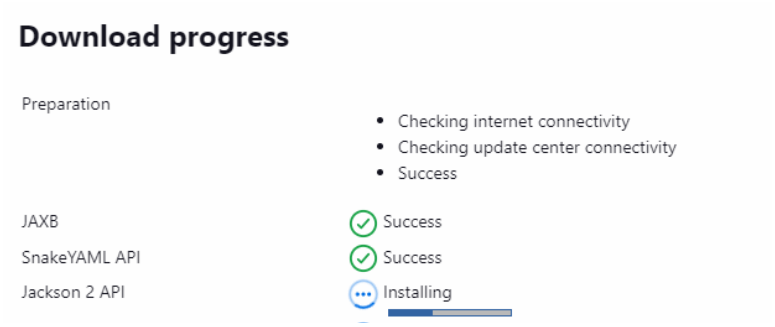
It is a combination of plugins that support the integration and implementation of continuous delivery pipelines using Jenkins.

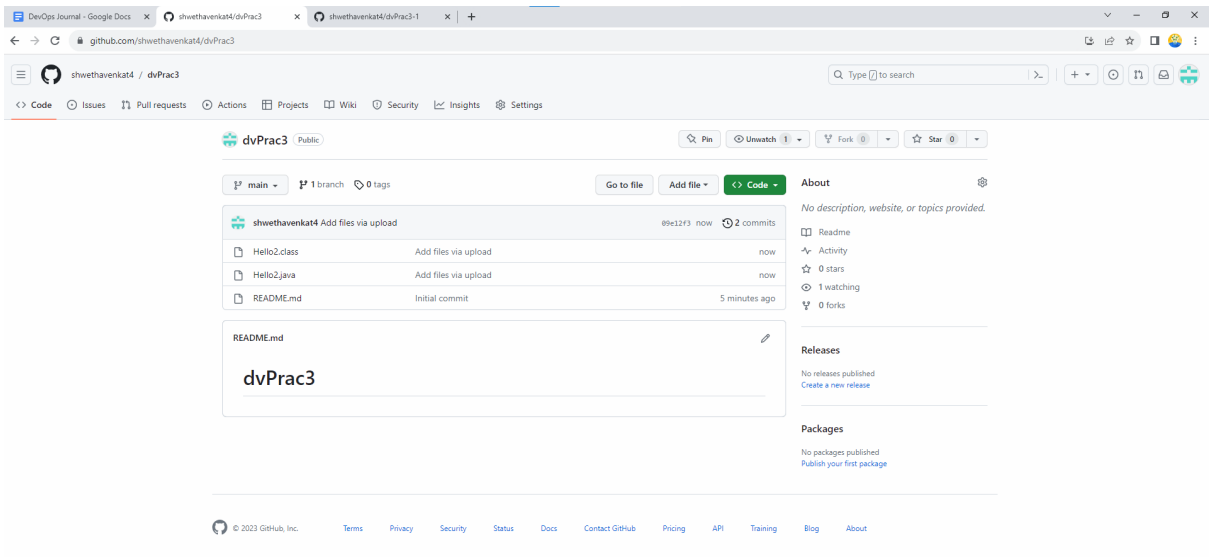
In other words, a Jenkins Pipeline is a collection of jobs or events that brings the software from version control into the hands of the end users by using automation tools. It is used to incorporate continuous delivery in our software development workflow.

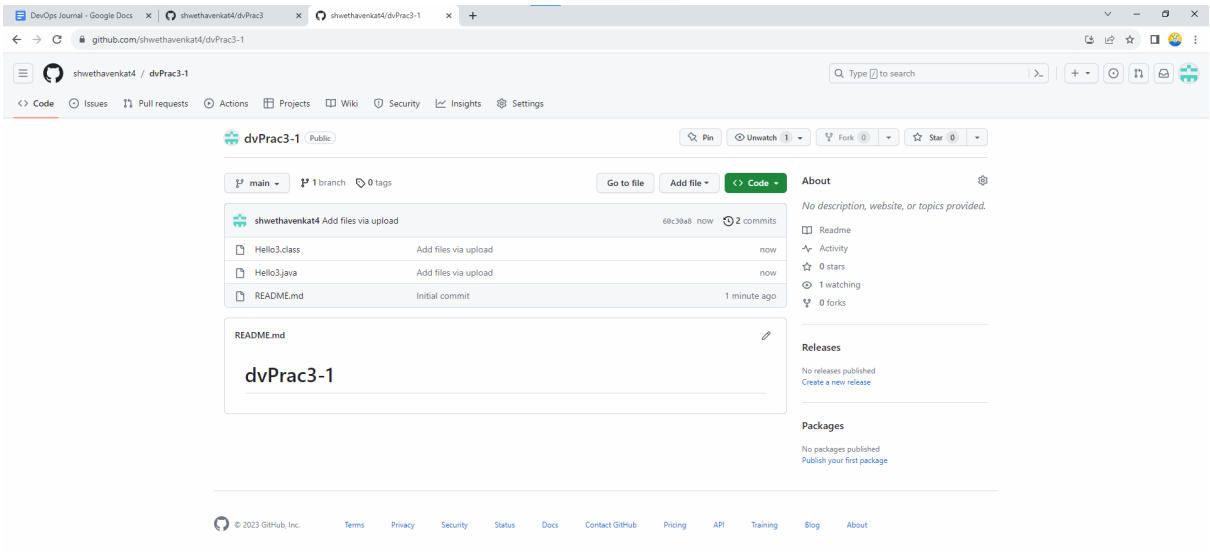
A pipeline has an extensible automation server for creating simple or even complex delivery pipelines "as code", via DSL (Domain-specific language).

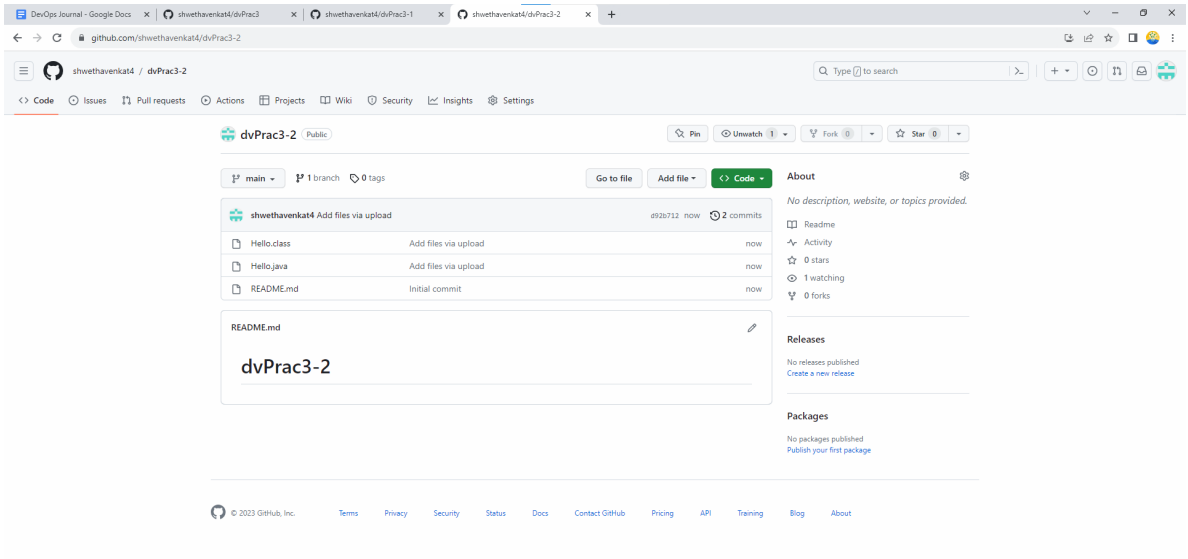


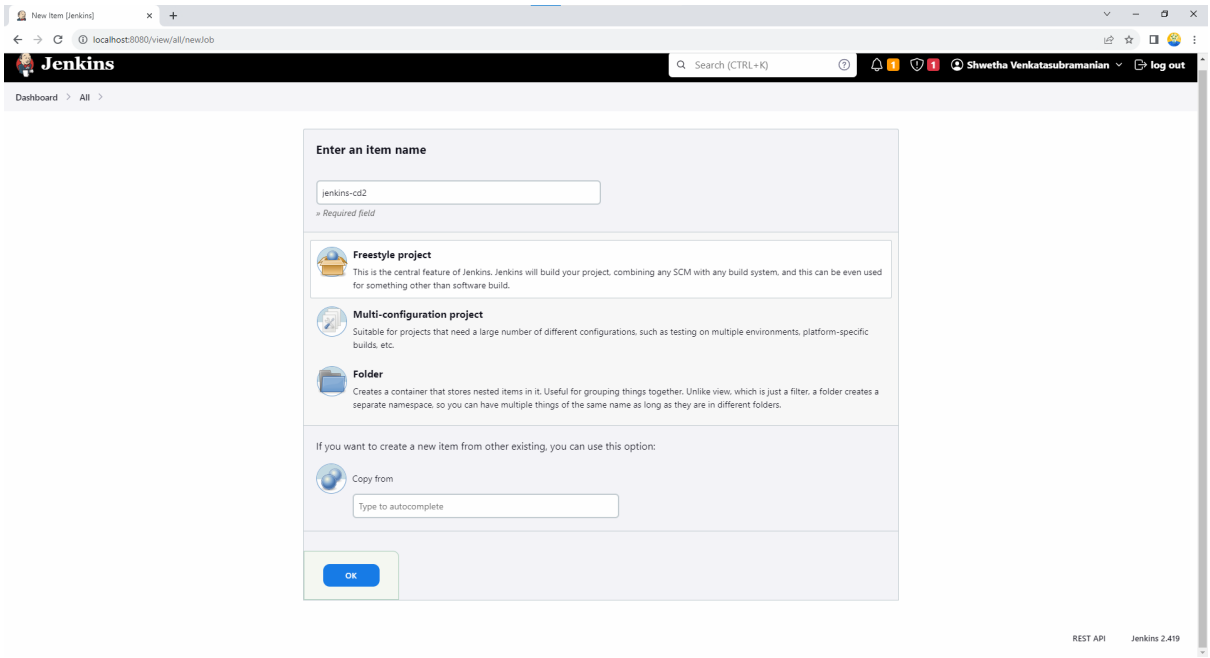


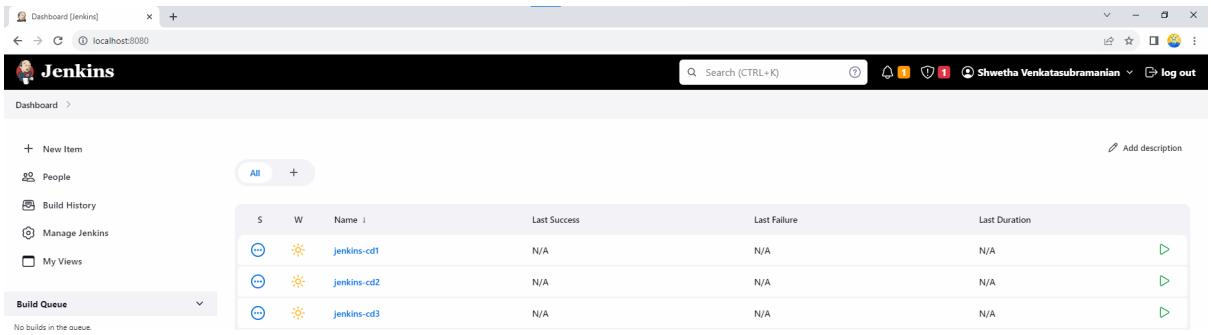

Download progress
Preparation
Checking internet connectivity
Checking update center connectivity
Success
JAXB
Success
SnakeYAML API
Success
Jackson 2 API
Installing

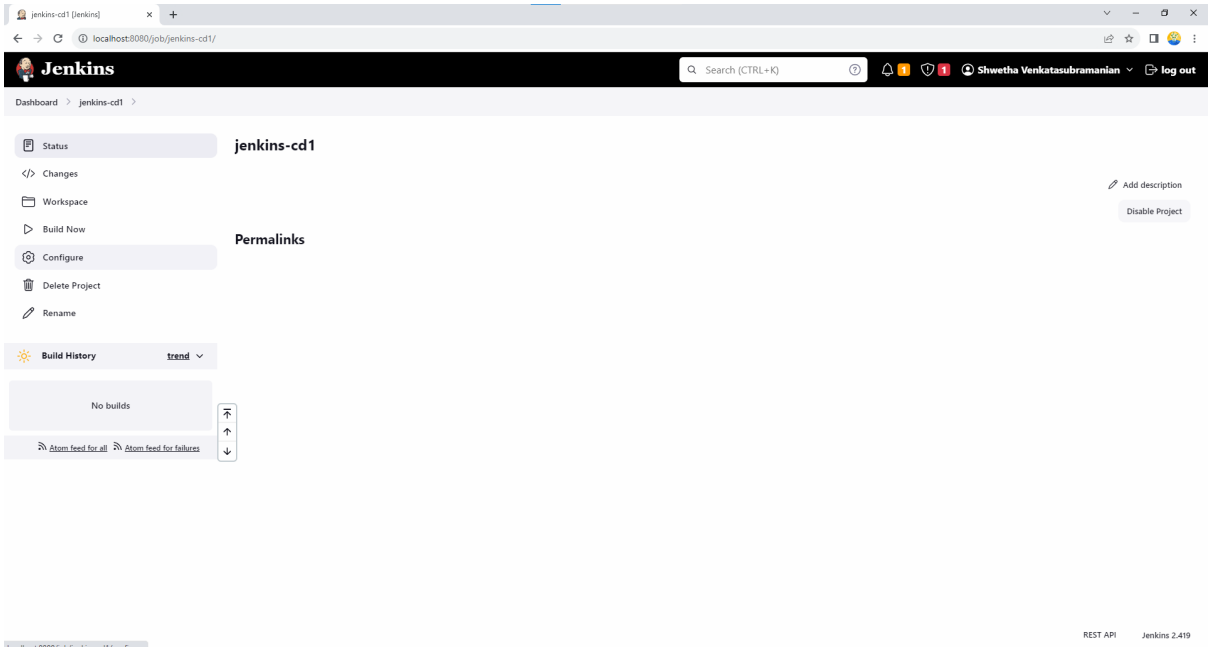

DevOps Journal - Google Docs
shwethavenkat4/dvPrac3
shwethavenkat4/dvPrac3-1
github.com/shwethavenkat4/dvPrac3
shwethavenkat4 / dvPrac3
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
dvPrac3
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 branch
0 tags
Go to file
Add file
Code
About
No description, website, or topics provided.
shwethavenkat4 Add files via upload
09e12f3 now
2 commits
Hello2.class
Add files via upload
now
Hello2.java
Add files via upload
now
README.md
Initial commit
5 minutes ago
Readme
Activity
0 stars
1 watching
0 forks
README.md
dvPrac3
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
© 2023 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About


DevOps Journal - Google Docs
shwethavenkat4/dvPrac3
shwethavenkat4/dvPrac3-1
github.com/shwethavenkat4/dvPrac3-1
shwethavenkat4 / dvPrac3-1
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
dvPrac3-1
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 branch
0 tags
Go to file
Add file
Code
About
No description, website, or topics provided.
shwethavenkat4 Add files via upload
60c30a8 now
2 commits
Hello3.class
Add files via upload
now
Hello3.java
Add files via upload
now
README.md
Initial commit
1 minute ago
Readme
Activity
0 stars
1 watching
0 forks
README.md
dvPrac3-1
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
© 2023 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About

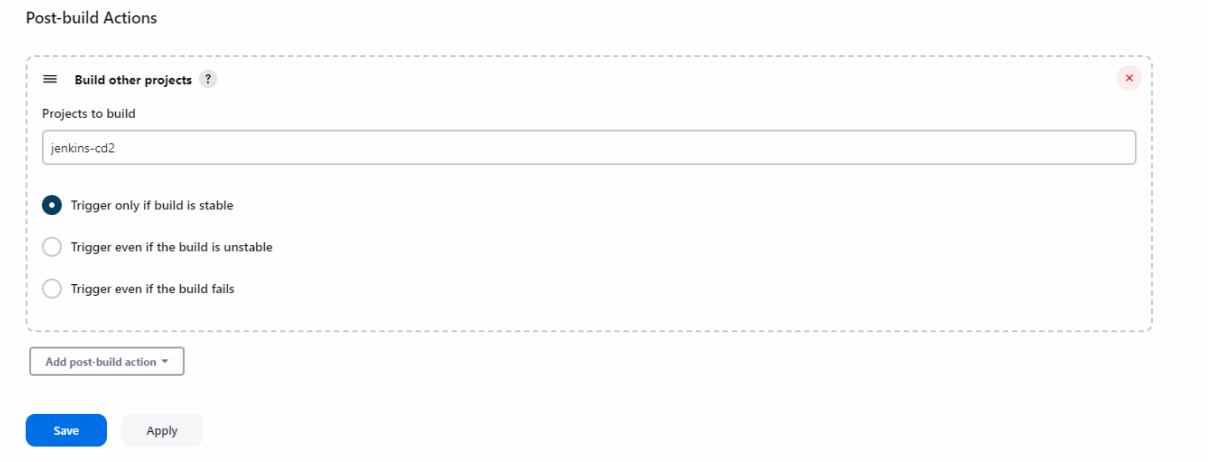

DevOps Journal - Google Docs
shwethavenkat4/dvPrac3
shwethavenkat4/dvPrac3-1
shwethavenkat4/dvPrac3-2
github.com/shwethavenkat4/dvPrac3-2
shwethavenkat4 / dvPrac3-2
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
dvPrac3-2
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 branch
0 tags
Go to file
Add file
Code
About
No description, website, or topics provided.
shwethavenkat4 Add files via upload
d92b712 now
2 commits
Hello.class
Add files via upload
now
Hello.java
Add files via upload
now
README.md
Initial commit
now
README.md
dvPrac3-2
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
© 2023 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About

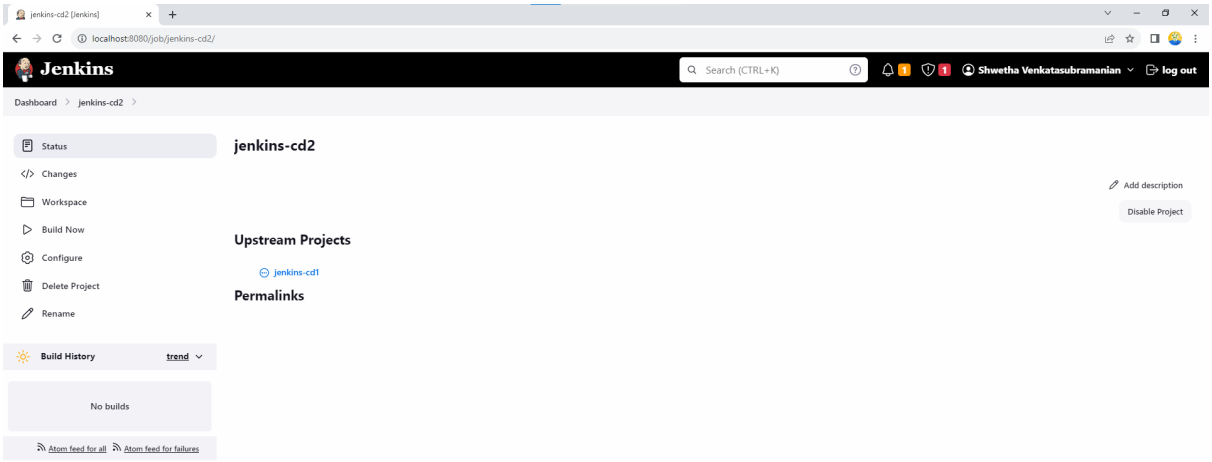

New Item [Jenkins]
localhost:8080/view/all/newJob
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
All
Enter an item name
jenkins-cd2
» Required field
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
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.
If you want to create a new item from other existing, you can use this option:
Copy from
Type to autocomplete
OK
REST API
Jenkins 2.419

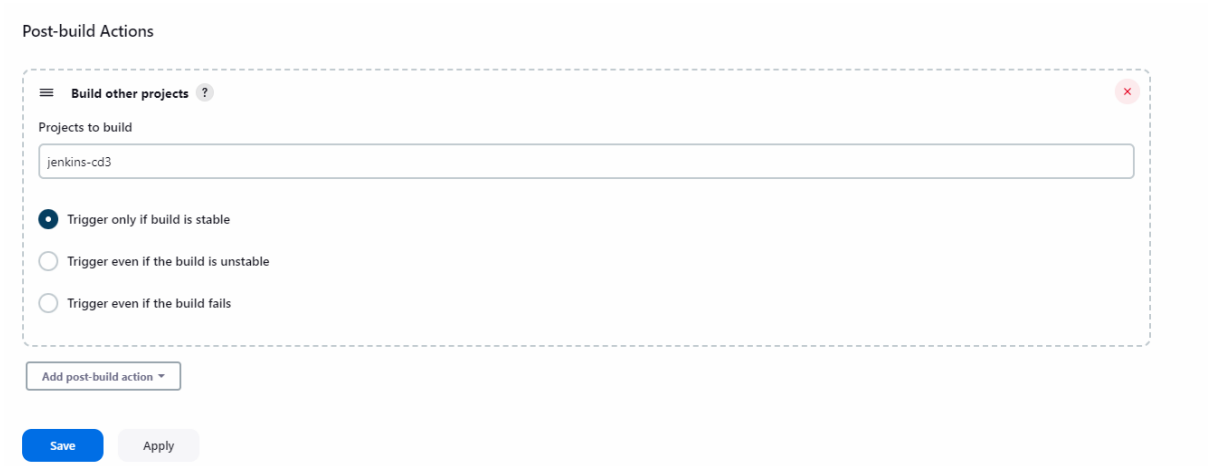

Dashboard [Jenkins]
localhost:8080
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
New Item
People
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Add description
All
S
W
Name
Last Success
Last Failure
Last Duration
jenkins-cd1
N/A
N/A
N/A
jenkins-cd2
N/A
N/A
N/A
jenkins-cd3
N/A
N/A
N/A

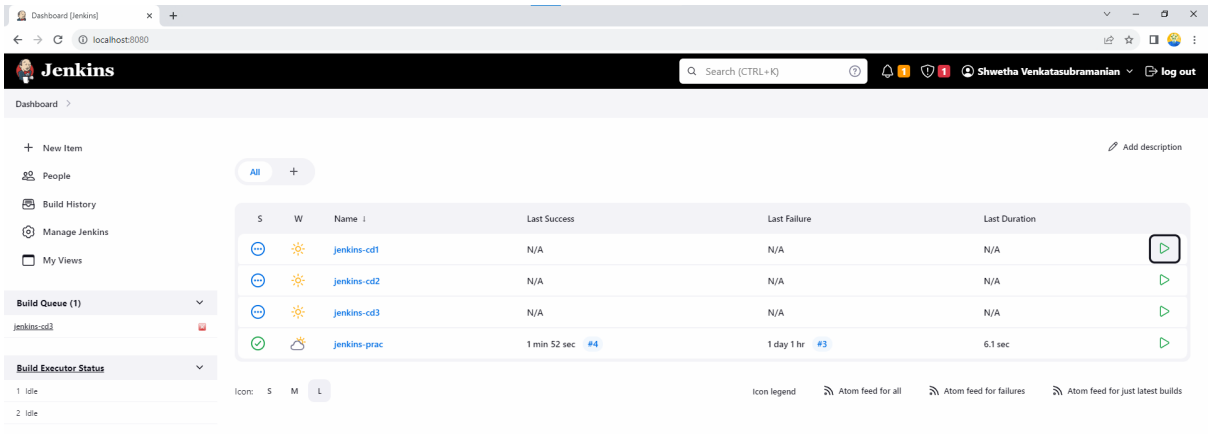

jenkins-cd1 [Jenkins]
localhost:8080/job/jenkins-cd1/
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
jenkins-cd1
Status
Changes
Workspace
Build Now
Configure
Delete Project
Rename
Build History
trend
No builds
Atom feed for all
Atom feed for failures
jenkins-cd1
Add description
Disable Project
Permalinks
REST API
Jenkins 2.419

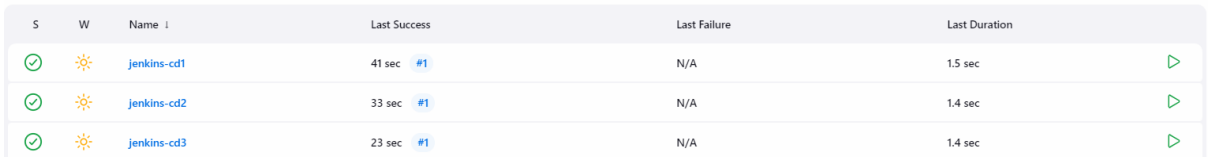

Post-build Actions
Add post-build action
Save
Apply

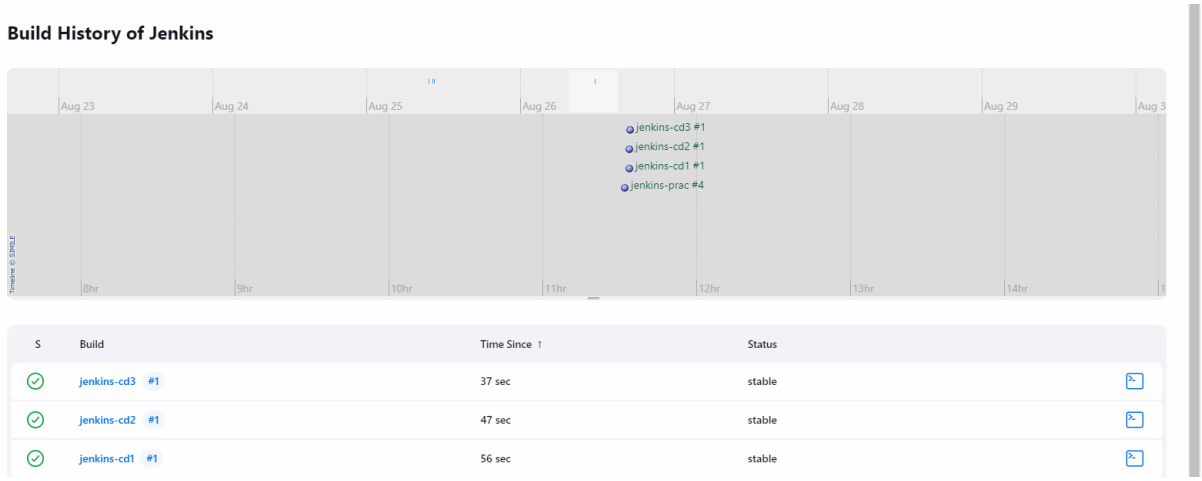

Post-build Actions
Build other projects
Projects to build
jenkins-cd2
Trigger only if build is stable
Trigger even if the build is unstable
Trigger even if the build fails
Add post-build action
Save
Apply

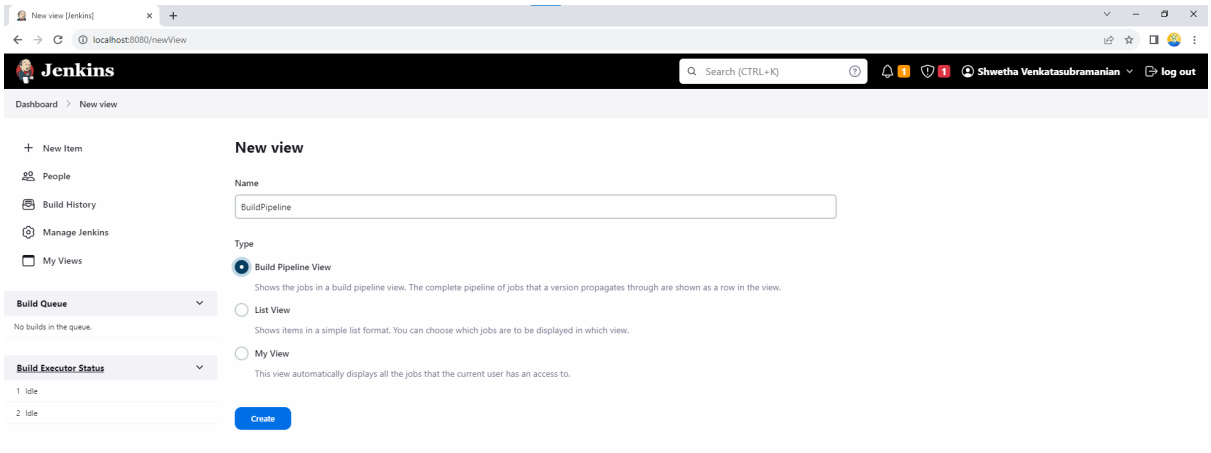

jenkins-cd2 [Jenkins]
localhost:8080/job/jenkins-cd2/
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
jenkins-cd2
Status
Changes
Workspace
Build Now
Configure
Delete Project
Rename
Build History
trend
No builds
Atom feed for all
Atom feed for failures
jenkins-cd2
Add description
Disable Project
Upstream Projects
jenkins-cd1
Permalinks

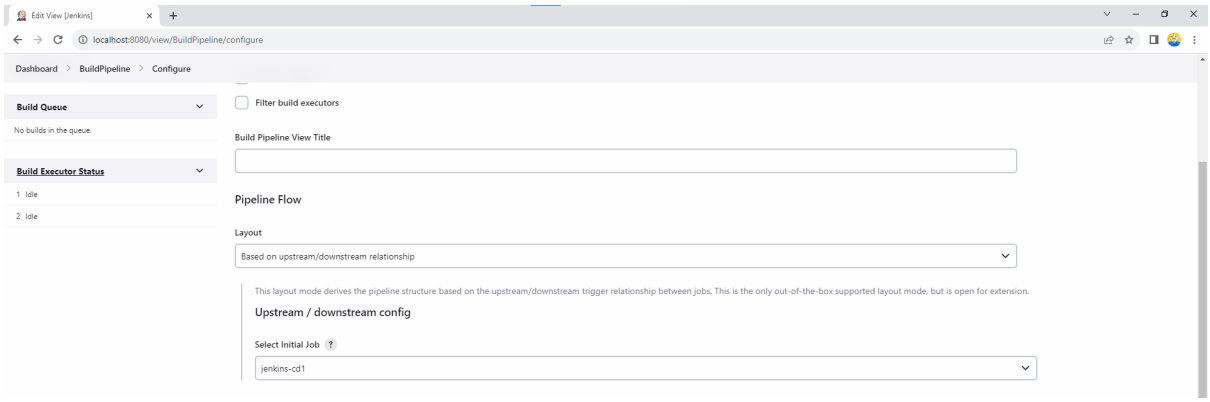

Post-build Actions
Build other projects
Projects to build
jenkins-cd3
Trigger only if build is stable
Trigger even if the build is unstable
Trigger even if the build fails
Add post-build action
Save
Apply

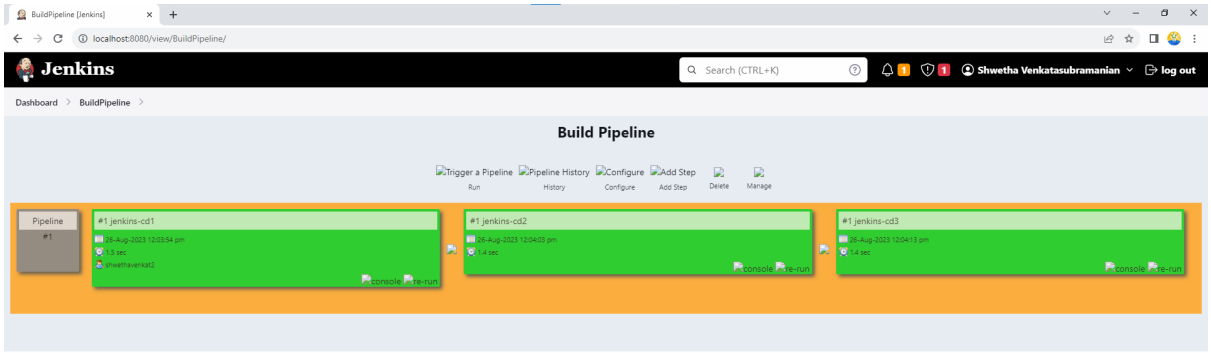

Dashboard [Jenkins]
localhost:8080
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
New Item
People
Build History
Manage Jenkins
My Views
Build Queue (1)
jenkins-cd3
Build Executor Status
1 Idle
2 Idle
Add description
All
S
W
Name
Last Success
Last Failure
Last Duration
jenkins-cd1
N/A
N/A
N/A
jenkins-cd2
N/A
N/A
N/A
jenkins-cd3
N/A
N/A
N/A
jenkins-prac
1 min 52 sec #4
1 day 1 hr #3
6.1 sec
Icon: S M L
Icon legend
Atom feed for all
Atom feed for failures
Atom feed for just latest builds

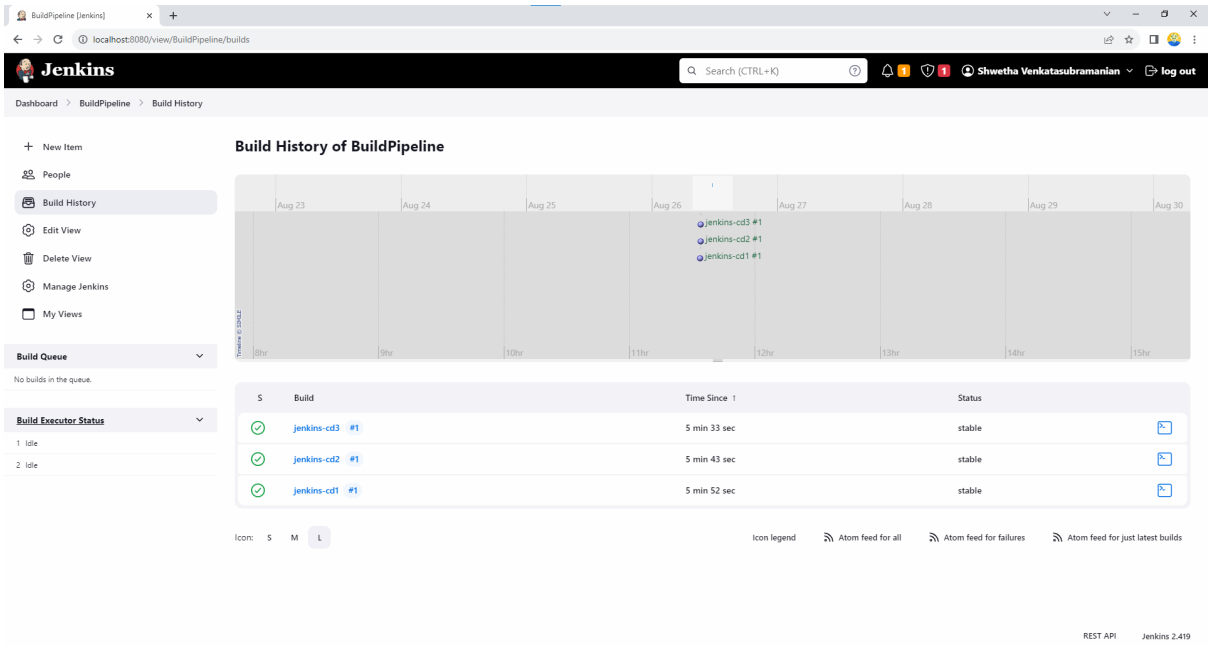

S
W
Name
Last Success
Last Failure
Last Duration
jenkins-cd1
41 sec #1
N/A
1.5 sec
jenkins-cd2
33 sec #1
N/A
1.4 sec
jenkins-cd3
23 sec #1
N/A
1.4 sec

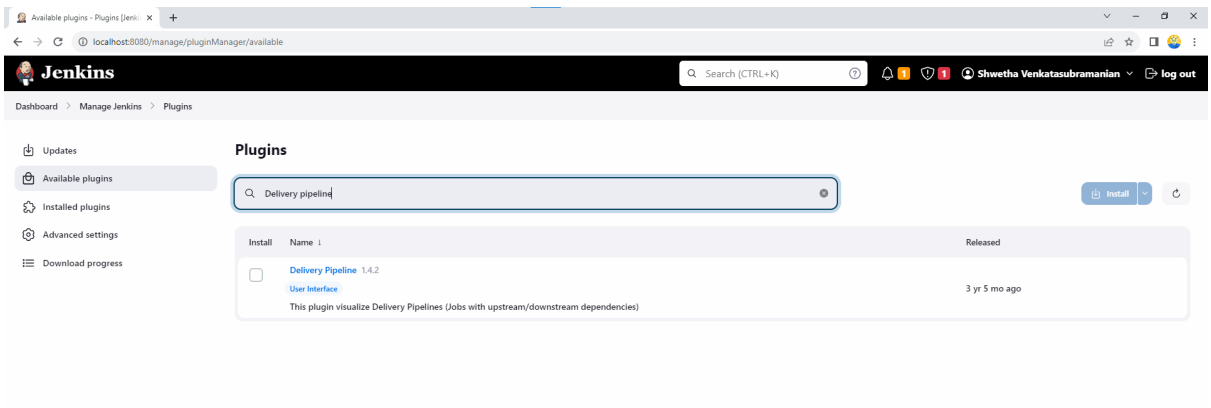

Build History of Jenkins
Aug 23
Aug 24
Aug 25
Aug 26
Aug 27
Aug 28
Aug 29
jenkins-cd3 #1
jenkins-cd2 #1
jenkins-cd1 #1
jenkins-prac #4
8hr
9hr
10hr
11hr
12hr
13hr
14hr
S
Build
Time Since
Status
jenkins-cd3 #1
37 sec
stable
jenkins-cd2 #1
47 sec
stable
jenkins-cd1 #1
56 sec
stable

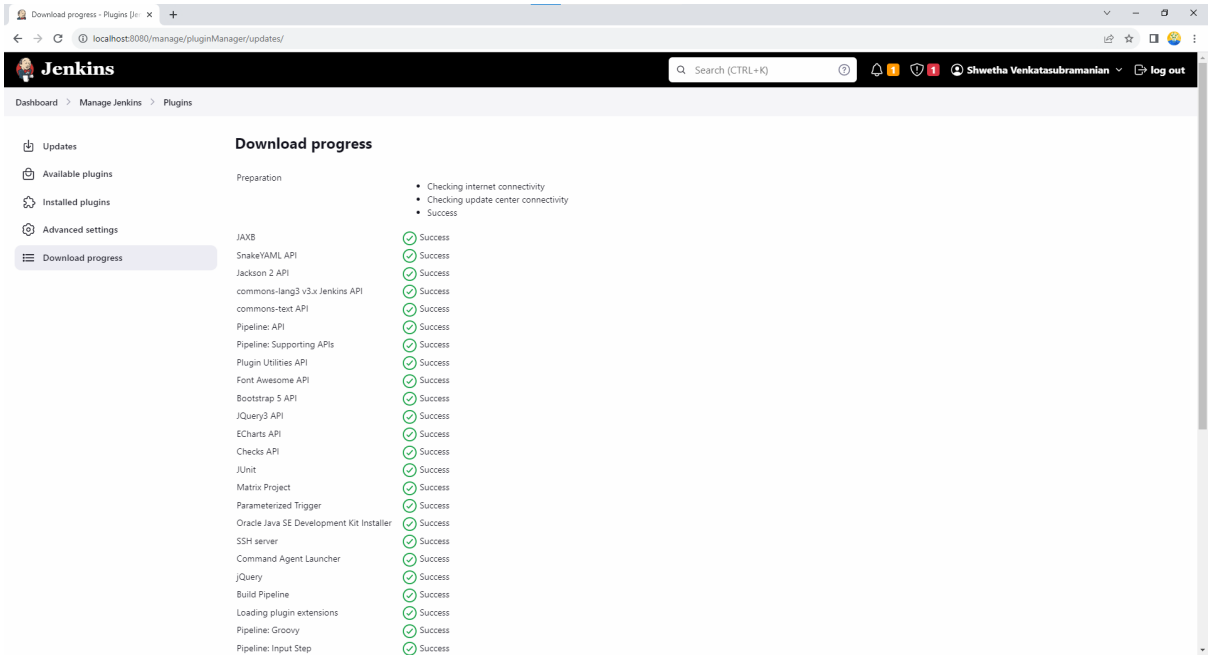

New view [Jenkins]
localhost:8080/newView
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
New view
New Item
People
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
New view
Name
BuildPipeline
Type
Build Pipeline View
Shows 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.
Create

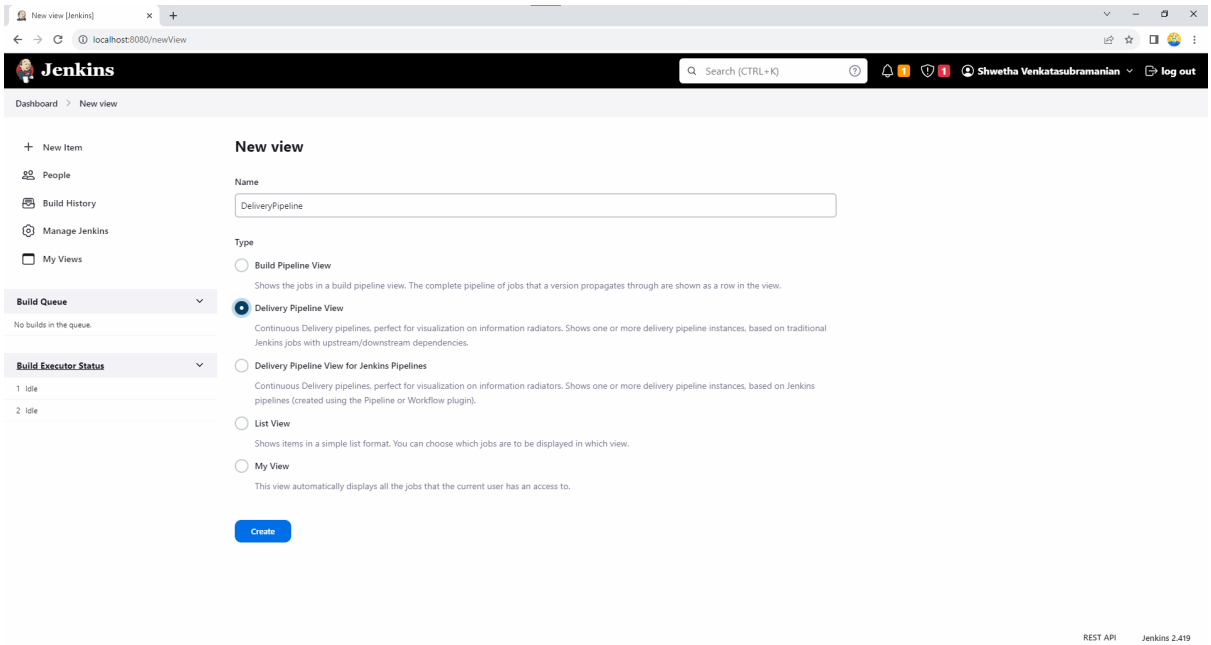

Edit View [Jenkins]
localhost:8080/view/BuildPipeline/configure
Dashboard
BuildPipeline
Configure
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Filter build executors
Build Pipeline View Title
Pipeline Flow
Layout
Based on upstream/downstream relationship
This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for extension.
Upstream / downstream config
Select Initial Job
jenkins-cd1

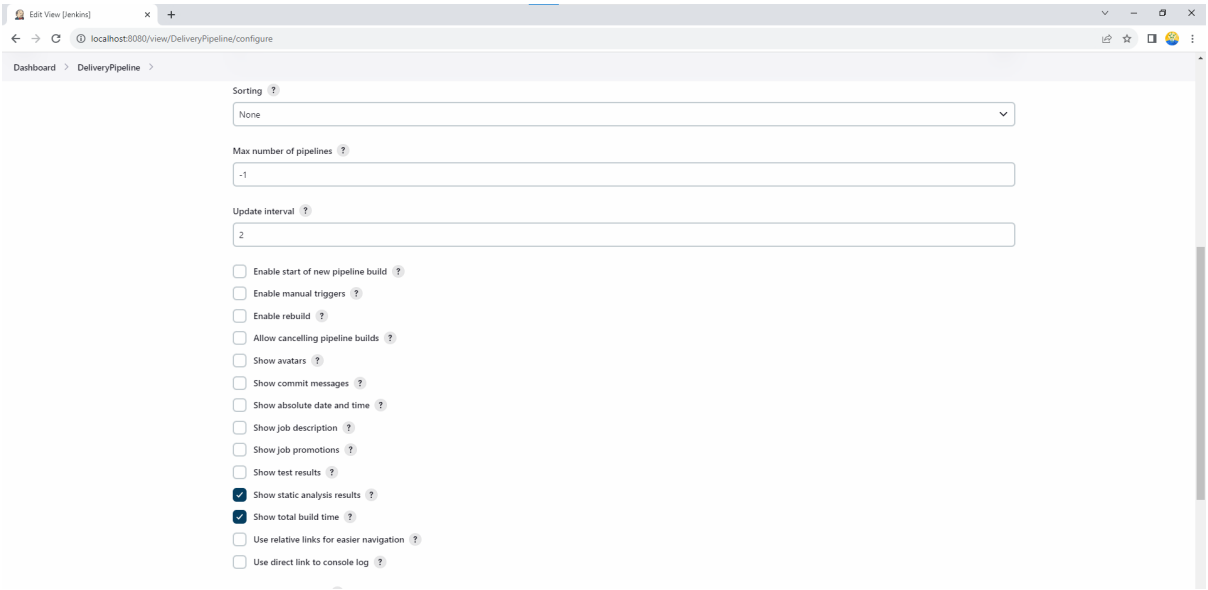

BuildPipeline [Jenkins]
localhost:8080/view/BuildPipeline/
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
BuildPipeline
Build Pipeline
Trigger a Pipeline
Run
Pipeline History
History
Configure
Configure
Add Step
Add Step
Delete
Delete
Manage
Manage
Pipeline
#1
#1 jenkins-cd1
#1 jenkins-cd2
#1 jenkins-cd3
console
re-run

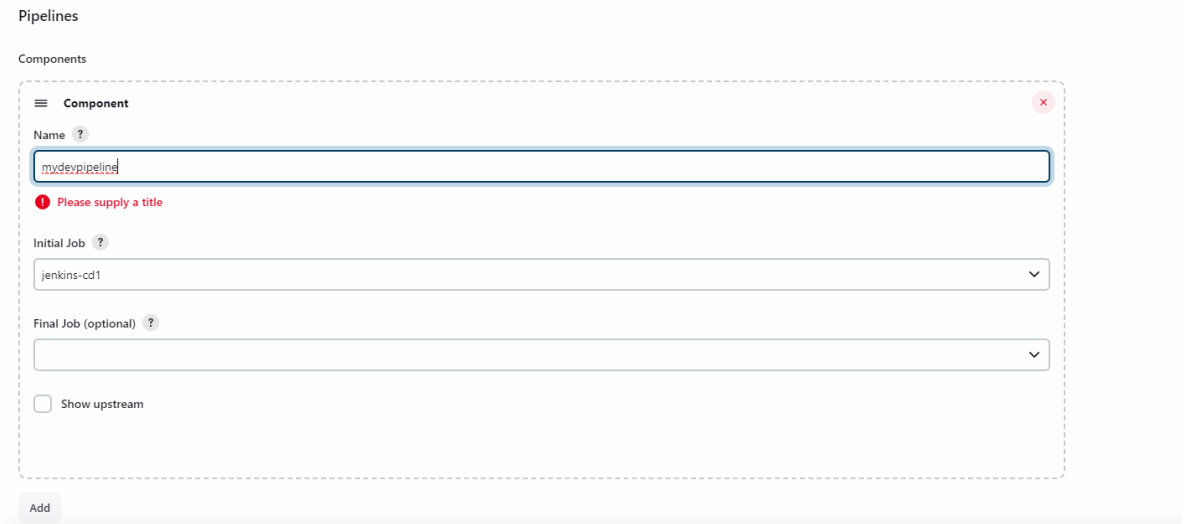

BuildPipeline [Jenkins]
localhost:8080/view/BuildPipeline/builds
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
BuildPipeline
Build History
New Item
People
Build History
Edit View
Delete View
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Build History of BuildPipeline
Aug 23
Aug 24
Aug 25
Aug 26
Aug 27
Aug 28
Aug 29
Aug 30
jenkins-cd3 #1
jenkins-cd2 #1
jenkins-cd1 #1
8hr
9hr
10hr
11hr
12hr
13hr
14hr
15hr
S
Build
Time Since
Status
jenkins-cd3 #1
5 min 33 sec
stable
jenkins-cd2 #1
5 min 43 sec
stable
jenkins-cd1 #1
5 min 52 sec
stable
Icon: S M L
Icon legend
Atom feed for all
Atom feed for failures
Atom feed for just latest builds
REST API
Jenkins 2.419

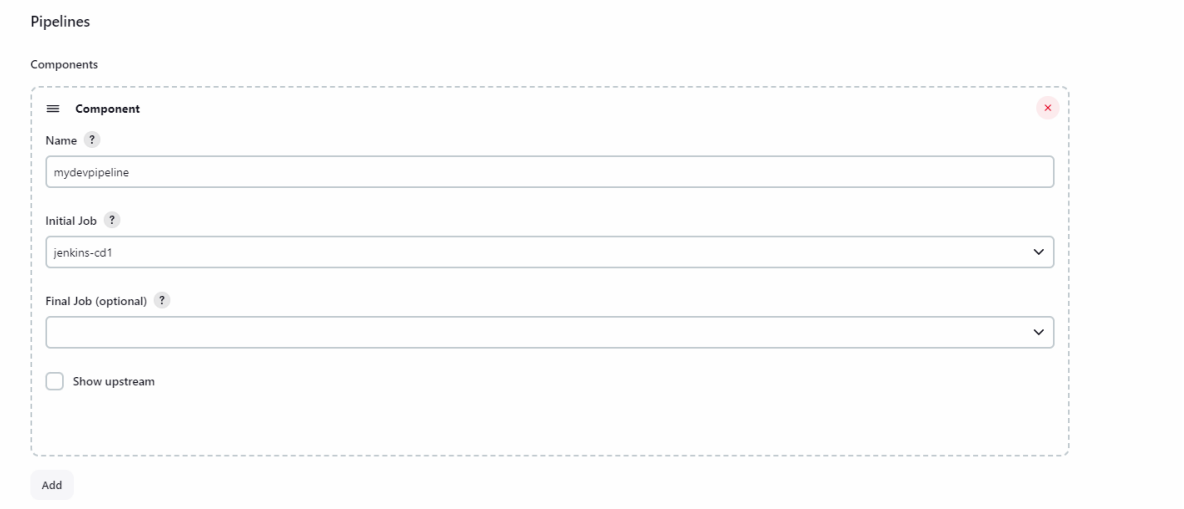

Available plugins - Plugins [Jenki
localhost:8080/manage/pluginManager/available
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
Manage Jenkins
Plugins
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Plugins
Delivery pipeline
Install
Install
Name
Released
Delivery Pipeline 1.4.2
User Interface
This plugin visualize Delivery Pipelines (Jobs with upstream/downstream dependencies)
3 yr 5 mo ago

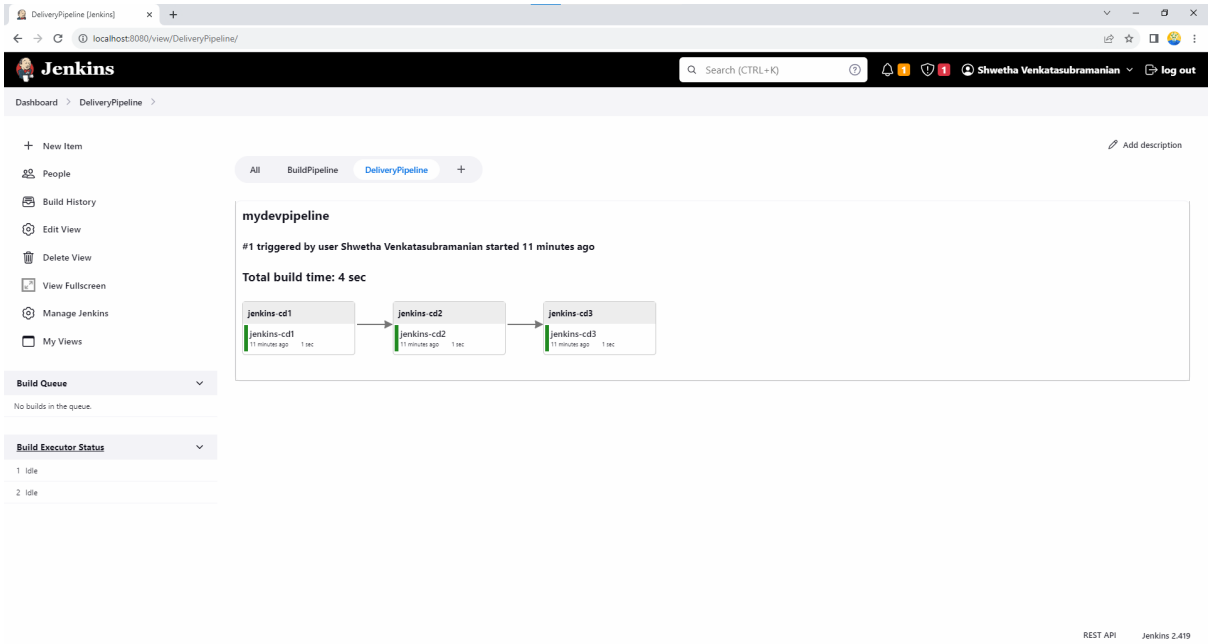

Download progress - Plugins [Je
localhost:8080/manage/pluginManager/updates/
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
Manage Jenkins
Plugins
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Download progress
Preparation
Checking internet connectivity
Checking update center connectivity
Success
JAXB Success
SnakeYAML API Success
Jackson 2 API Success
commons-lang3 v3.x Jenkins API Success
commons-text API Success
Pipeline: API Success
Pipeline: Supporting APIs Success
Plugin Utilities API Success
Font Awesome API Success
Bootstrap 5 API Success
JQuery3 API Success
ECharts API Success
Checks API Success
JUnit Success
Matrix Project Success
Parameterized Trigger Success
Oracle Java SE Development Kit Installer Success
SSH server Success
Command Agent Launcher Success
jQuery Success
Build Pipeline Success
Loading plugin extensions Success
Pipeline: Groovy Success
Pipeline: Input Step Success

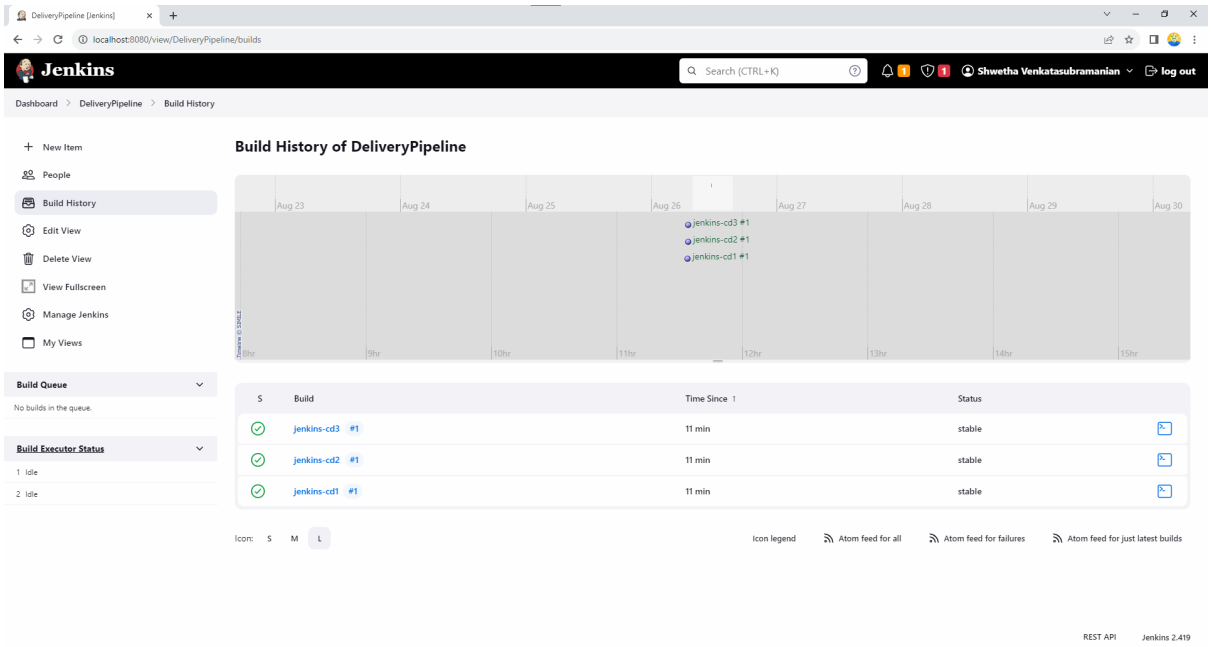

New view [Jenkins]
localhost:8080/newView
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
New view
New Item
People
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
New view
Name
DeliveryPipeline
Type
Build Pipeline View
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.
Delivery Pipeline View
Continuous Delivery pipelines, perfect for visualization on information radiators. Shows one or more delivery pipeline instances, based on traditional Jenkins jobs with upstream/downstream dependencies.
Delivery Pipeline View for Jenkins Pipelines
Continuous Delivery pipelines, perfect for visualization on information radiators. Shows one or more delivery pipeline instances, based on Jenkins pipelines (created using the Pipeline or Workflow plugin).
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.
Create
REST API
Jenkins 2.419


Edit View [Jenkins]
localhost:8080/view/DeliveryPipeline/configure
Dashboard
DeliveryPipeline
Sorting
None
Max number of pipelines
-1
Update interval
2
Enable start of new pipeline build
Enable manual triggers
Enable rebuild
Allow cancelling pipeline builds
Show avatars
Show commit messages
Show absolute date and time
Show job description
Show job promotions
Show test results
Show static analysis results
Show total build time
Use relative links for easier navigation
Use direct link to console log


Pipelines
Components
Component
Name
mydevpipeline
Please supply a title
Initial Job
jenkins-cd1
Final Job (optional)
Show upstream
Add


Pipelines
Components
Component
Name
mydevpipeline
Initial Job
jenkins-cd1
Final Job (optional)
Show upstream
Add


DeliveryPipeline [Jenkins]
localhost:8080/view/DeliveryPipeline/
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
DeliveryPipeline
New Item
People
Build History
Edit View
Delete View
View Fullscreen
Manage Jenkins
My Views
Add description
All
BuildPipeline
DeliveryPipeline
mydevpipeline
#1 triggered by user Shwetha Venkatasubramanian started 11 minutes ago
Total build time: 4 sec
jenkins-cd1
jenkins-cd2
jenkins-cd3
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
REST API
Jenkins 2.419


DeliveryPipeline [Jenkins]
localhost:8080/view/DeliveryPipeline/builds
Jenkins
Search (CTRL+K)
Shwetha Venkatasubramanian
log out
Dashboard
DeliveryPipeline
Build History
New Item
People
Build History
Edit View
Delete View
View Fullscreen
Manage Jenkins
My Views
Build History of DeliveryPipeline
Aug 23
Aug 24
Aug 25
Aug 26
Aug 27
Aug 28
Aug 29
Aug 30
jenkins-cd3 #1
jenkins-cd2 #1
jenkins-cd1 #1
S
Build
Time Since
Status
jenkins-cd3 #1
11 min
stable
jenkins-cd2 #1
11 min
stable
jenkins-cd1 #1
11 min
stable
Icon: S M L
Icon 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
REST API
Jenkins 2.419

Practical 4:

Applying DevSecOps on Jenkins

DevSecOps stands for Development, Security, and Operations. It's an approach that integrates security practices within the DevOps process, ensuring that security is a shared responsibility throughout the development lifecycle rather than an afterthought. The goal of DevSecOps is to deliver secure software faster by embedding security into every stage of the continuous integration and continuous deployment (CI/CD) pipeline.

Advantages of DevSecOps

- Security by design.
- Faster recovery if security issues are raised.
- Reduce the vulnerability in the application.
- Maintain and ensure compliance.
- Offers more speed and agility to the security team.
- Increase observability and transparency.
- Helps to maintain the security and compliance in the pipeline.

DevSecOps Tools

There are various tools available in market to implement the DevSecOps practice , some of those are as follows.

- Hadolint: Docker file linter, validate inline bash, written in Haskell.
- GitSecret: Checks the any sensitive information sent to repository
- Checkov: To check the file following standard best security practice
- Trivy: To scan the vulnerabilities in image, file system
- Owasp: Penetration testing tool
- Falco: Runtime application self protection
- ECR scanning: Inbuilt tool by AWS to check image vulnerability

DevSecOps Security Check stages

To implement DevSecOps, it needs to be implemented in stages which are as follows.

- Secret Analysis
- Static application security testing
- Dynamic application security testing
- Runtime application self protection

Practical 5:

Containerization using Docker

Docker is a centralised platform for packaging, deploying, and running applications. Before Docker, many users faced the problem that a particular code is running in the developer's system but not in the user's system. So, the main reason to develop docker is to help developers to develop applications easily, ship them into containers and can be deployed anywhere.

Docker is an open-source centralised platform designed to create, deploy, and run applications. Docker uses a container on the host's operating system to run applications. It allows applications to use the same Linux kernel as a system on the host computer, rather than creating a whole virtual operating system. Containers ensure that our application works in any environment like development, test, or production.

Docker includes components such as Docker client, Docker server, Docker machine, Docker hub, Docker compose, etc.

Docker Containers

Docker containers are lightweight alternatives to the virtual machine. It allows developers to package up the application with all its libraries and dependencies, and ship it as a single package. The advantage of using a docker container is that you don't need to allocate any RAM and disk space for the applications. It automatically generates storage and space according to the application requirement.

Docker Engine

- It is a client-server application that contains the following major components.
- A server is a type of long-running program called a daemon process.
- The REST API is used to specify interfaces that programs can use to talk to the daemon and instruct it on what to do.
- A command line interface client.

1. Install Ubuntu in Machine from MS Store
2. In cmd wsl –update
3. In docker-> settings-> resources -> wsl integration-> enable ubuntu -> apply and restart
4. In c-> users -> computerfolder-> create folder mydockerimages
5. Put hello.java and hello.class file
6. Open powershell
7. wsl --update
8. wsl --set-default-version 2
9. cd C:\Users\CG-CS-PC21\my_docker_images
10. set-content C:\Users\CG-CS-PC21\my_docker_images\newdockerfile

FROM ubuntu:22.04
RUN sed -i 's|http://archive.ubuntu.com/ubuntu/|http://mirrors.kernel.org/ubuntu/|g' /etc/apt/sources.list
RUN apt-get update && apt-get install -y default-jdk
COPY . .
RUN javac Hello.java
CMD ["java", "Hello"]

```
FROM ubuntu:22.04

# Change to a different mirror
RUN sed -i 's|http://archive.ubuntu.com/ubuntu/|http://mirrors.kernel.org/ubuntu/|g' /etc/apt/sources.list

# Install default JDK
RUN apt-get update && apt-get install -y default-jdk

# Copy the local files into the container
COPY . .

# Compile the Java program
RUN javac Hello.java

# Run the Java program
CMD ["java", "Hello"]
```

1. docker run -it --rm ubuntu:22.04 /bin/bash
1. apt-get update
1. apt-get install -y default-jdk
1. Exit
1. docker build --no-cache -f newdockerfile .
1. Sign in to docker
1. docker images
1. Take the image id from recent image
1. docker run 53a843653cbc
1. Docker pull ubuntu

1. docker run -it -d ubuntu
1. Docker ps -a
1. docker start
1. docker exec -it b0f98530bed3 bash
1. Echo hello
1. Exit

Practical 6:

Containerization using Kubernetes

Kubernetes is also known as 'k8s'. This word comes from the Greek language, which means pilot or helmsman.

Kubernetes is an extensible, portable, and open-source platform designed by Google in 2014. It is mainly used to automate the deployment, scaling, and operations of container-based applications across the cluster of nodes. It is also designed for managing the services of containerized apps using different methods that provide scalability, predictability, and high availability.

It is actually an enhanced version of 'Borg' for managing long-running processes and batch jobs. Nowadays, many cloud services offer a Kubernetes-based infrastructure on which it can be deployed as a platform-providing service. This technique or concept works with many container tools, like docker, and follows the client-server architecture.

The following are the key objects that exist in Kubernetes:

Pod

It is the smallest and simplest basic unit of the Kubernetes application. This object indicates the processes which are running in the cluster.

Node

A **node** is nothing but a single host, which is used to run the virtual or physical machines. A node in the Kubernetes cluster is also known as a minion.

Service

A **service** in Kubernetes is a logical set of pods, which works together. With the help of services, users can easily manage load-balancing configurations.

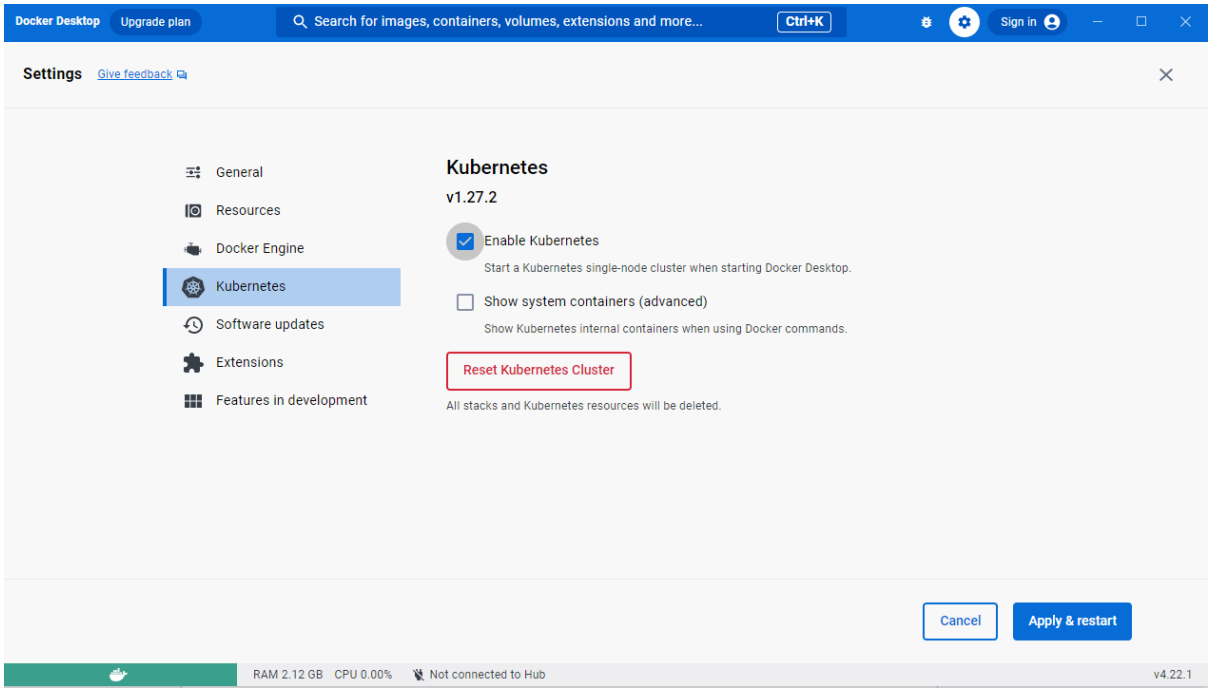
ReplicaSet

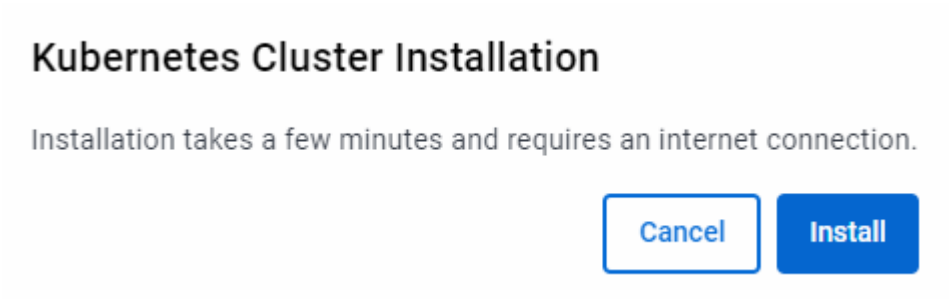
A **ReplicaSet** in the Kubernetes is used to identify the particular number of pod replicas are running at a given time. It replaces the replication controller because it is more powerful and allows a user to use the "set-based" label selector.

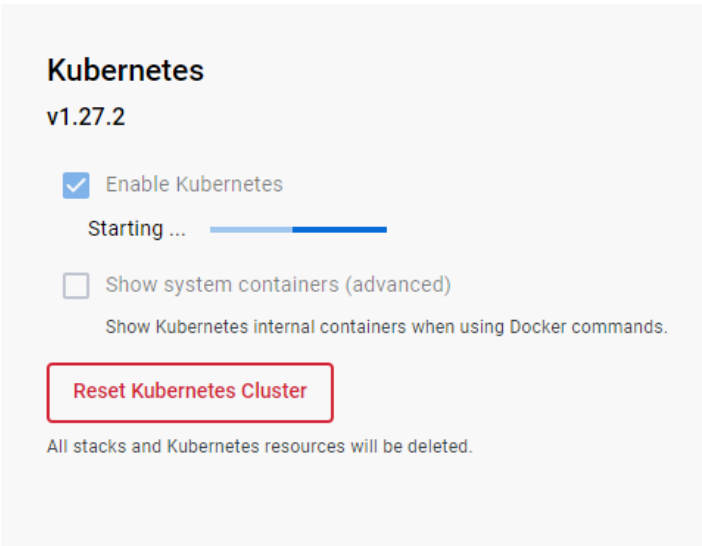
Namespace

Kubernetes supports various virtual clusters, which are known as namespaces. It is a way of dividing the cluster resources between two or more users.

1. Install Kubernetes







PS C:\Windows\system32> kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.6.1/aio/deploy/recommended.yaml

namespace/kubernetes-dashboard created

serviceaccount/kubernetes-dashboard created

service/kubernetes-dashboard created

secret/kubernetes-dashboard-certs created

secret/kubernetes-dashboard-csrf created

secret/kubernetes-dashboard-key-holder created

configmap/kubernetes-dashboard-settings created

role.rbac.authorization.k8s.io/kubernetes-dashboard created

clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created

rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created

clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created

deployment.apps/kubernetes-dashboard created

service/dashboard-metrics-scraper created

deployment.apps/dashboard-metrics-scraper created

PS C:\Windows\system32> cd C:\Users\CG-CS-PC09\Desktop

PS C:\Users\CG-CS-PC09\Desktop> kubectl apply -f dashboard-adminuser.yaml

serviceaccount/admin-user created

PS C:\Users\CG-CS-PC09\Desktop> kubectl -n kubernetes-dashboard create token admin-user

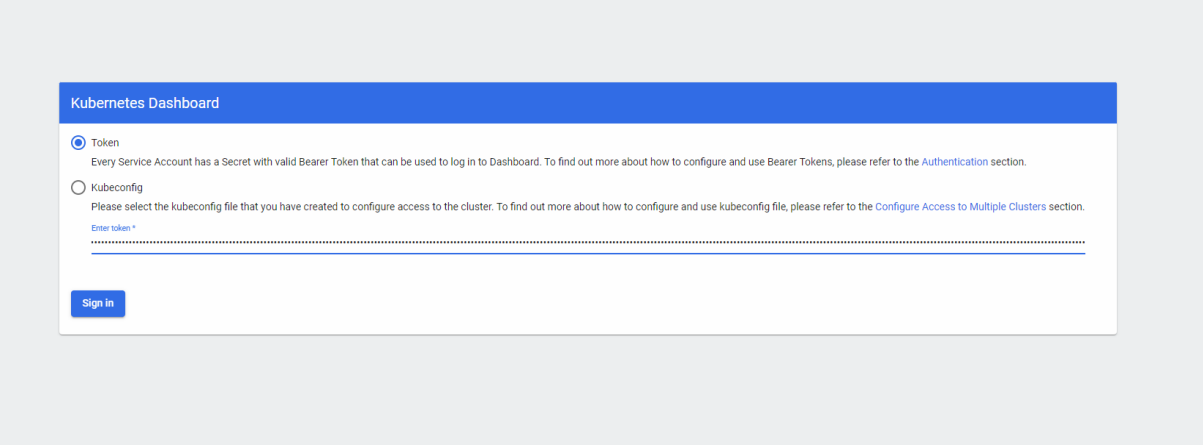
eyJhbGciOiJSUzI1NiIsImtpZCI6InZTeWdndnIySVlvRWFpT2V3TEFyb05hMjZEeklDOUQybm03OThqRkNTZncifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjoxNjkzMjg1MTM0LCJpYXQiOjE2OTMyODE1MzQsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJhZG1pbi11c2VyIiwidWlkIjoiMTk4YzdlYmYtY2E0Mi00NmQxLWJmMmUtOGM1MzdlMDgyMzk3In19LCJuYmYiOjE2OTMyODE1MzQsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDprdWJlcm5ldGVzLWRhc2hib2FyZDphZG1pbi11c2VyIn0.qICdGSdyUQ1Om6VER526N1kmlVyMWEM5keDTaYag9T14VituxCko027CoJ9s0Qmxl6wy6gLArrRAhjOppe0ErwGz0t50J4-RZWIp2wsohAtD_e8KA9jsS_sFiQKbRKjJgifwIYSlOtPUR2nHrSmh0wT7Eq4O9zN31ooiKTgpBsl1-Q-

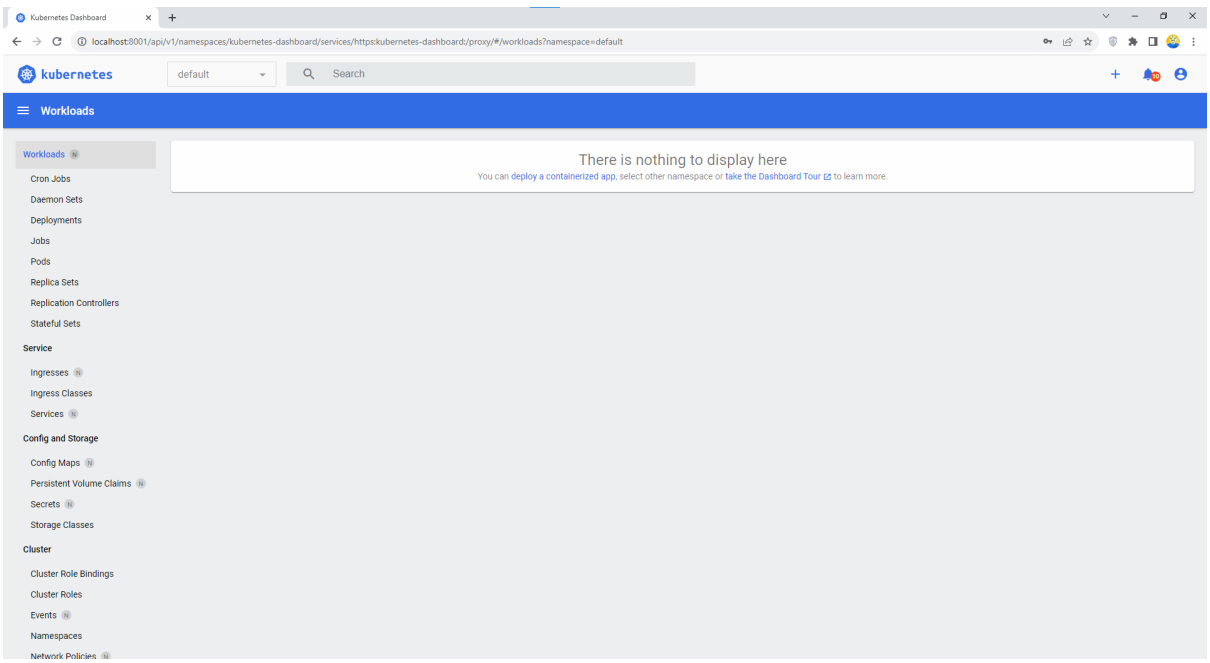
r1wDRLf6XJKMYiHB2PQHPZlhAWIAXljJKGJdQmLLhp51_K3F_YwXjaYYJ854cEHHXdDQz9cEHzEt6ZWjqkaAa_b0TUYQaoPYtrH_QW5OoZen6YnG8ZYqFCAiQBwBCiq8otYzoUniGpdlOiHQeMD2WdivZ7GbY_iUGc_kntw

PS C:\Users\CG-CS-PC09\Desktop> kubectl proxy

Starting to serve on 127.0.0.1:8001

Link: http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/login





PS C:\Users\CG-CS-PC09\Desktop> kubectl create -f myDeployment.yml

deployment.apps/nginx created

PS C:\Users\CG-CS-PC09\Desktop> kubectl get deployment

NAME READY UP-TO-DATE AVAILABLE AGE

nginx 3/3 3 3 5m18s

PS C:\Users\CG-CS-PC09\Desktop> kubectl get replicaset

NAME DESIRED CURRENT READY AGE

nginx-55f598f8d 3 3 3 6m9s

PS C:\Users\CG-CS-PC09\Desktop> kubectl get pod

NAME READY STATUS RESTARTS AGE

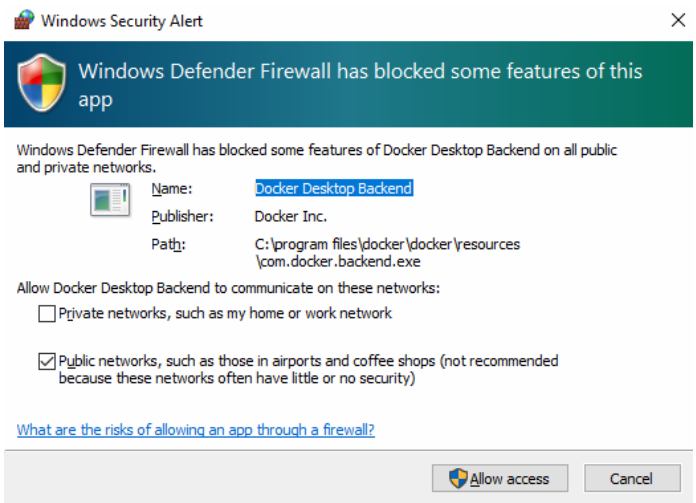
nginx-55f598f8d-8w6x4 1/1 Running 0 7m20s

nginx-55f598f8d-dsclr 1/1 Running 0 7m20s

nginx-55f598f8d-n2gsb 1/1 Running 0 7m20s

PS C:\Users\CG-CS-PC09\Desktop> kubectl create -f myService.yml

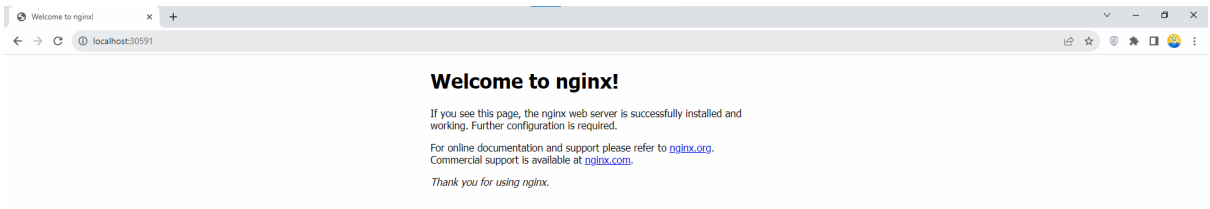
service/nginx created



PS C:\Users\CG-CS-PC09\Desktop> kubectl get service nginx

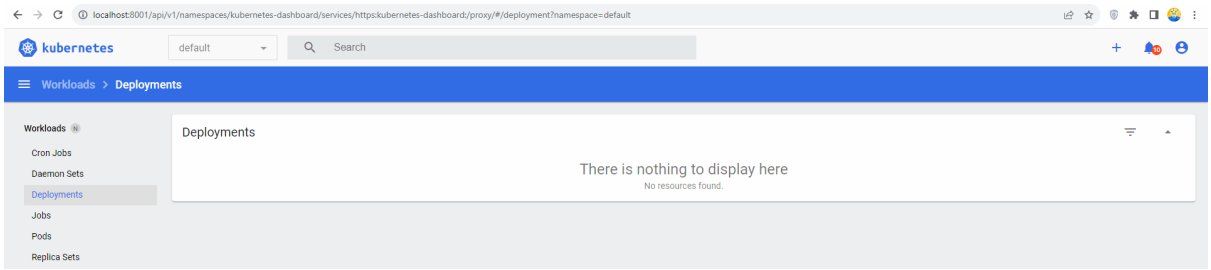
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

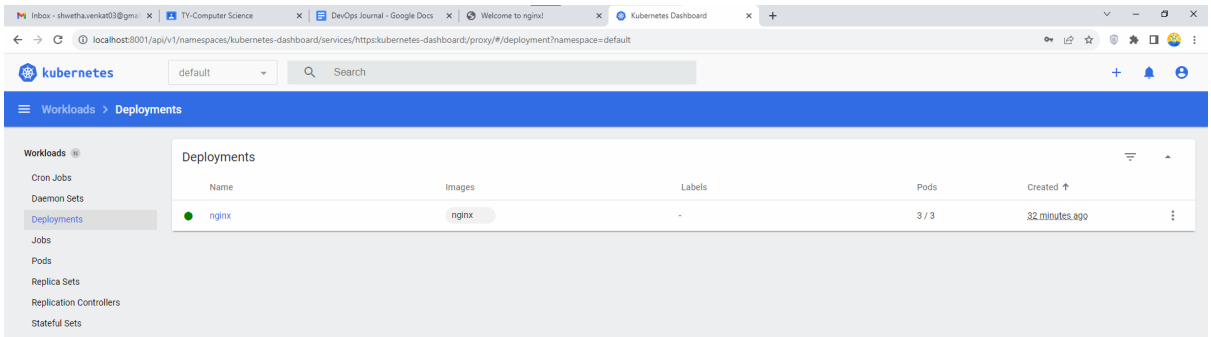
nginx NodePort 10.99.122.169 <none> 80:30591/TCP 2m4s

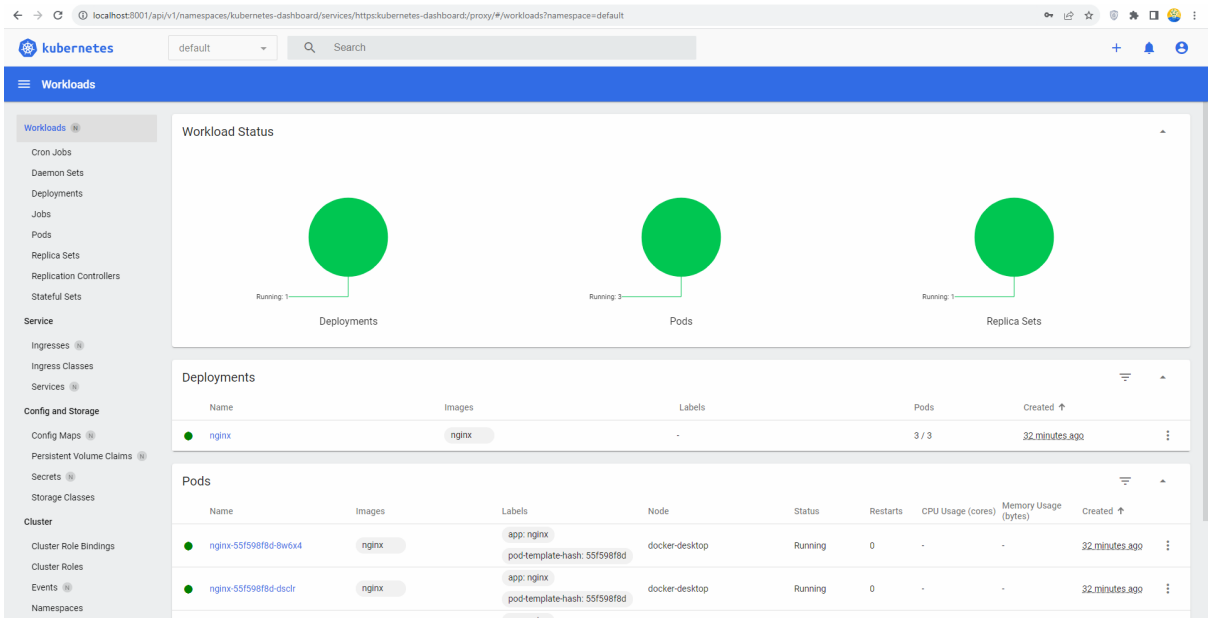


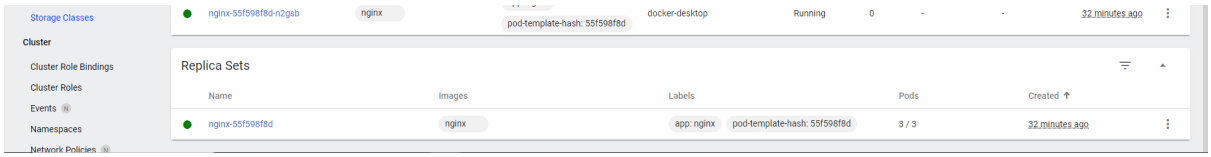
PS C:\Users\CG-CS-PC09\Desktop> kubectl proxy

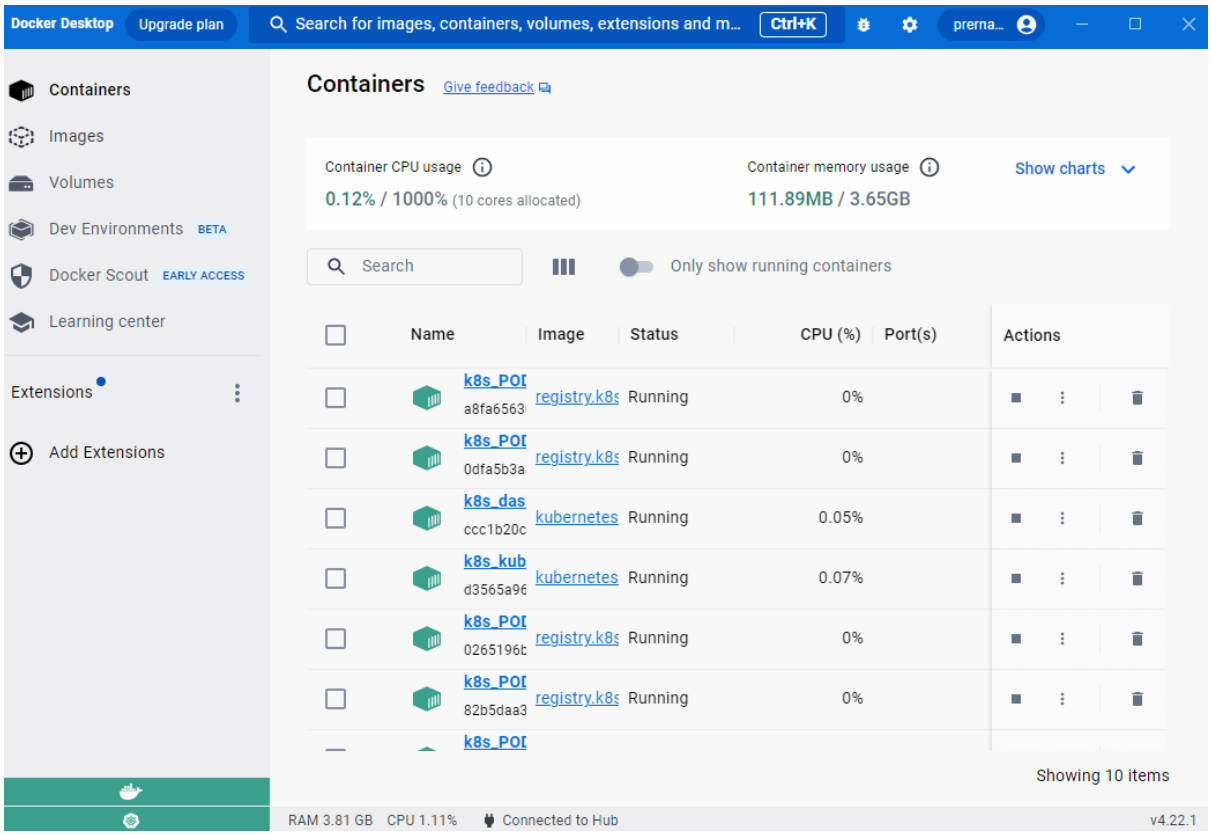
Starting to serve on 127.0.0.1:8001






kubernetes
default
Search
Workloads
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Storage Classes
Cluster
Cluster Role Bindings
Cluster Roles
Events
Namespaces
Workload Status
Deployments
Pods
Replica Sets
Deployments
Name
Images
Labels
Pods
Created
nginx
nginx
3 / 3
32 minutes ago
Pods
Name
Images
Labels
Node
Status
Restarts
CPU Usage (cores)
Memory Usage (bytes)
Created
nginx-55f598f8d-8w6x4
nginx
app: nginx
pod-template-hash: 55f598f8d
docker-desktop
Running
0
32 minutes ago
nginx-55f598f8d-dsclr
nginx
app: nginx
pod-template-hash: 55f598f8d
docker-desktop
Running
0
32 minutes ago


Storage Classes
Cluster
Cluster Role Bindings
Cluster Roles
Events
Namespaces
Network Policies
nginx-55f598f8d-n2gsb
nginx
pod-template-hash: 55f598f8d
docker-desktop
Running
0
32 minutes ago
Replica Sets
Name
Images
Labels
Pods
Created
nginx-55f598f8d
nginx
app: nginx
pod-template-hash: 55f598f8d
3 / 3
32 minutes ago


Docker Desktop
Upgrade plan
Search for images, containers, volumes, extensions and m...
Ctrl+K
prerna...
Containers
Images
Volumes
Dev Environments
BETA
Docker Scout
EARLY ACCESS
Learning center
Extensions
Add Extensions
Containers
Give feedback
Container CPU usage
0.12% / 1000% (10 cores allocated)
Container memory usage
111.89MB / 3.65GB
Show charts
Search
Only show running containers
Name
Image
Status
CPU (%)
Port(s)
Actions
k8s_POD
a8fa6563
registry.k8s
Running
0%
k8s_POD
0dfa5b3a
registry.k8s
Running
0%
k8s_das
ccc1b20c
kubernetes
Running
0.05%
k8s_kub
d3565a96
kubernetes
Running
0.07%
k8s_POD
0265196b
registry.k8s
Running
0%
k8s_POD
82b5daa3
registry.k8s
Running
0%
k8s_POD
Showing 10 items
RAM 3.81 GB
CPU 1.11%
Connected to Hub
v4.22.1

Practical 7:

Configuration Management using puppet

Configuration management occurs when a configuration platform is used to automate, monitor, design, and manage otherwise manual configuration processes. System-wide changes take place across servers and networks, storage, applications, and other managed systems.

An important function of configuration management is defining the state of each system. By orchestrating these processes with a platform, organizations can ensure consistency across integrated systems and increase efficiency. The result is that businesses can scale more readily without hiring additional IT management staff. Companies that otherwise wouldn't have the resources can grow by deploying a DevOps approach.

Configuration management is closely associated with change management, and as a result, the two terms are sometimes confused. Configuration management is most readily described as the automation, management, and maintenance of configurations at each state, while change management is the process by which configurations are redefined and changed to meet the conditions of new needs and dynamic circumstances.

A number of tools are available for those seeking to implement configuration management in their organizations. Puppet has carried the torch in pioneering configuration management, but other companies like Chef and Red Hat also offer intriguing suites of products to enhance configuration management processes. Proper configuration management is at the core of continuous testing and delivery, two key benefits of DevOps.

Based on what we've discussed, you may have already gleaned that configuration management takes on the primary responsibility for three broad categories required for DevOps transformation: identification, control, and audit processes.

Identification:
The process of finding and cataloging system-wide configuration needs.

Control:
During configuration control, we see the importance of change management at work. It's highly likely that configuration needs will change over time, and configuration control allows this to happen in a controlled way so as to not destabilize integrations and existing infrastructure.

Audit:
Like most audit processes, a configuration audit is a review of the existing systems to ensure that it stands up to compliance regulations and validations.

Like DevOps, configuration management is spread across both operational and development buckets within an organization. This is by design. There are primary components that go into the comprehensive configuration management required for DevOps:

- Artifact repository
- Source code repository
- Configuration management data architecture

What is Puppet?

- Puppet is a **DevOps configuration management tool**. This is developed by Puppet Labs and is available for both open-source and enterprise versions. It is used to centralize and automate the procedure of configuration management.
- This tool is developed using Ruby DSL (domain-specific language), which allows you to change a complete infrastructure in code format and can be easily managed and configured.
- Puppet tool deploys, configures, and manages the servers. This is used particularly for the automation of hybrid infrastructure delivery and management.
- With the help of automation, Puppet enables system administrators to operate easier and faster.
- Puppet can also be used as a deployment tool as it can deploy software on the system automatically. Puppet implements infrastructure as a code, which means that you can test the environment for accurate deployment.
- Puppet supports many platforms such as Microsoft Windows, Debian/Ubuntu, Red Hat/CentOS/Fedora, MacOS X, etc.
- Puppet uses the client-server paradigm, where one system in any cluster works as the server, called the puppet master, and other works as a client on nodes called a slave.

Puppet Blocks

Puppet provides the flexibility to integrate Reports with third-party tools using Puppet APIs.

Four types of Puppet building blocks are

1. Resources
2. Classes
3. Manifest
4. Modules

Puppet Resources:

Puppet Resources are the building blocks of Puppet.

Resources are the inbuilt functions that run at the back end to perform the required operations in Puppet.

Puppet Classes:

A combination of different resources can be grouped together into a single unit called a class.

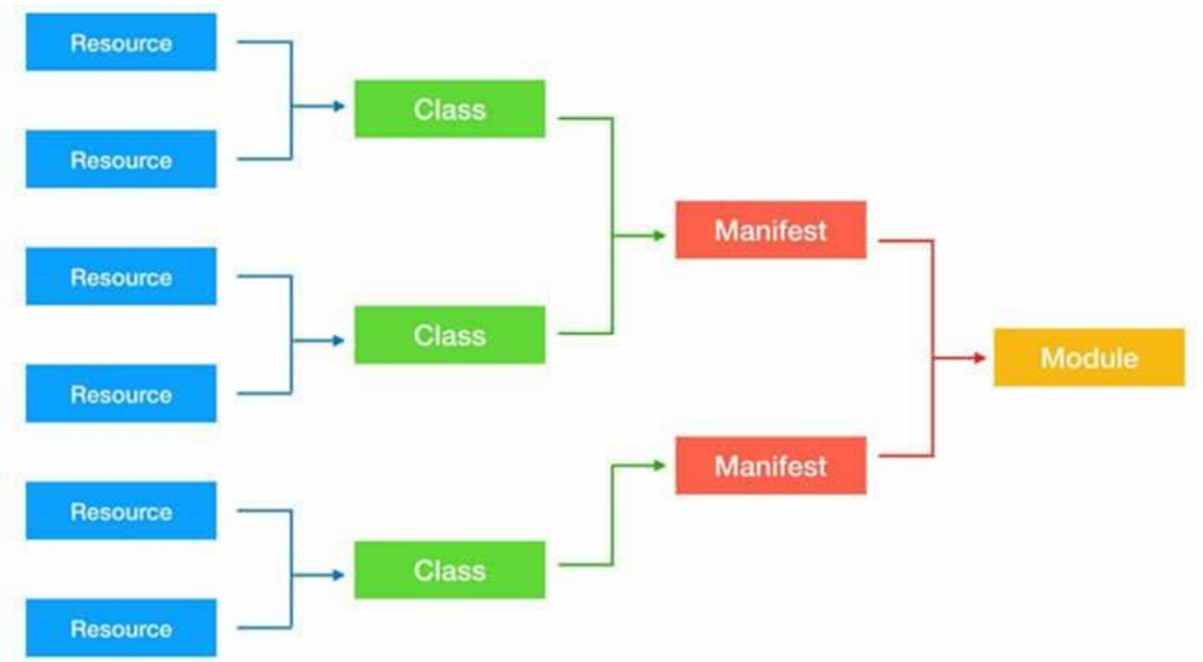
Puppet Manifest:

Manifest is a directory containing puppet DSL files. Those files have a .pp extension. The .pp extension stands for puppet program. The puppet code consists of definitions or declarations of Puppet Classes.

Puppet Modules:

Modules are a collection of files and directories such as Manifests and class definitions. They are the reusable and shareable units in Puppet.

For example, the MySQL module to install and configure MySQL or the Jenkins module to manage Jenkins, etc.

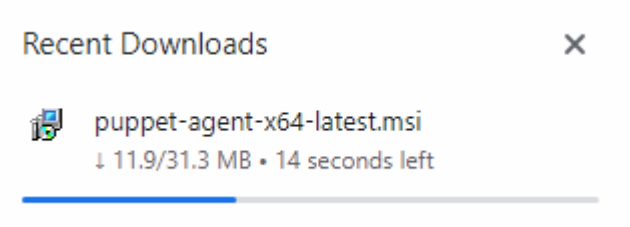


Windows nodes cannot serve as puppet master servers.

- If your Windows nodes will be fetching configurations from a puppet master, you will need a *nix server to run as a puppet master at your site.
- If your Windows nodes will be compiling and applying configurations locally with Puppet apply, you should disable the Puppet agent service on them after installing Puppet.

1. Download Puppet from this link:

http://downloads.puppetlabs.com/windows/puppet-agent-x64-latest.msi

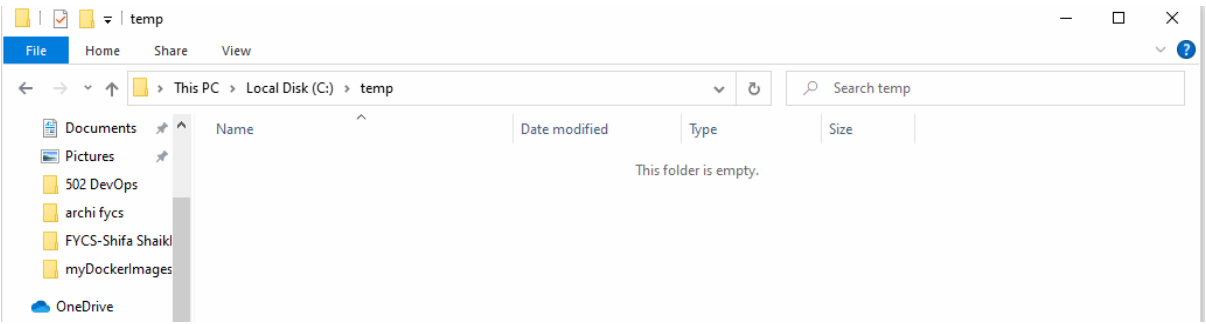


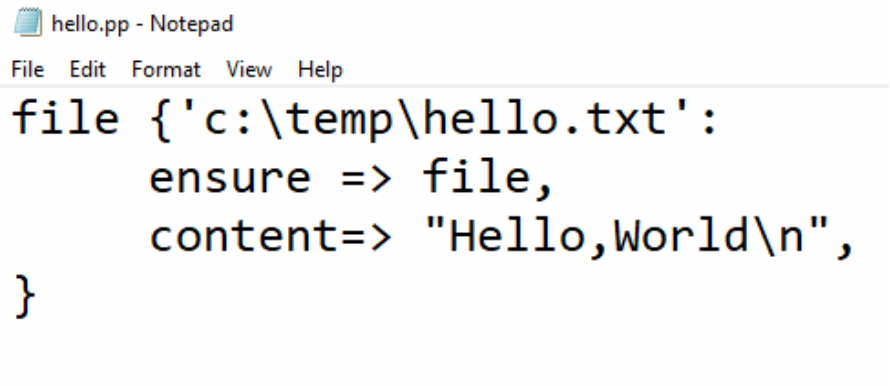
PS C: \Windows \system32> puppet config print config

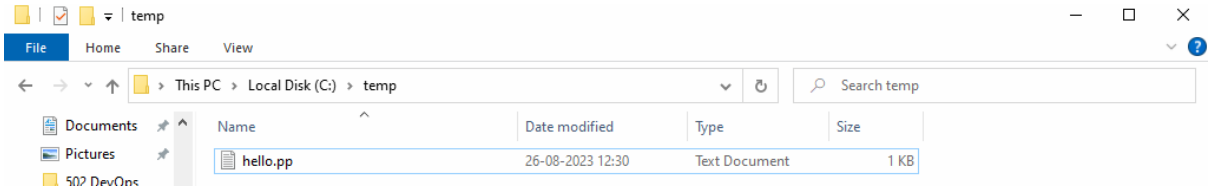
C:/ProgramData/PuppetLabs/puppet/etc/puppet.conf

PS C: \Windows\system32> puppet config print modulepath

C:/ProgramData/PuppetLabs/code/environments/production/modules;C:/ProgramData/PuppetLabs/code/modules







PS C: \Windows \system32> cd C: \temp

PS C: \temp> puppet apply hello.pp

Notice: Compiled catalog for cg-cs-pc01 in environment production in 0.09 seconds

Notice: /Stage [main]/Main/File[c: \temp\hello.txt]/ensure: defined content as "{md5\78d5333e735ae15f5192e768386728*

Notice: Applied catalog in 0.05 seconds