Vidyavardhaka Sangha®, Mysore

VIDYAVARDHAKA COLLEGE OF ENGINEERING

Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi

(Approved by AICTE, New Delhi & Government of Karnataka)

Accredited by NBA (CV, CS, EE, EC, IS & ME) | NAAC with 'A' Grade

P.B. No. 206, Gokulam III Stage, Mysuru-570 002, Karnataka, India

Phone: +91 821 4276201 /202 /225, **Fax:** +91 824 2510677

Web: http://www.vvce.ac.in @vvceofficial

PRACTICAL RECORD

Department of Computer Science & Engineering

DevOps-21CS62

Name : NAGESH S

USN : 4VV21CS103

Vidyavardhaka Sangha®, Mysore

VIDYAVARDHAKA COLLEGE OF ENGINEERING

Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi
(Approved by AICTE, New Delhi & Government of Karnataka)
Accredited by NBA (CV, CS, EE, EC, IS & ME) | NAAC with 'A' Grade
P.B. No. 206, Gokulam III Stage, Mysuru-570 002, Karnataka, India
Phone: +91 821 4276201 /202 /225, **Fax:** +91 824 2510677
Web: http://www.vvce.ac.in @vvceofficial

Department of Computer Science & Engineering

DevOps-21CS62

Year : 2023-24

Name : NAGESH S

USN : 4VV21CS103

Class Marks	Test Marks	OEE Marks	CIA Marks

Signature of Faculty

Signature of HOD

A1. Demonstrate and Create project in local and remote repository using GitBash and GitHub and apply init, status, log, add, commit, push, config, clone and reset commands on repository.

Git **init** command:

Description;

init is the Git command that allows you to create a local repository.This command creates a .git directory that contains all of the folders and configuration files of the local repository.

Syntax :git init

Example :git init

Git **status** command:

Description:

The status command is used to display the state of the working directory and the staging area. It also lists the files that you've changed and those you still need to add or commit.

Syntax :git status

Example :git status

Git **log** command:

Description:

This command is used to check the commit history.

Syntax :git log

Example :git log

Git **add** command:

Description:

Making a commit (which we will see next) is to archive our changes in our local repository. When we edit files, we can choose which ones will be included in the next commit; it's a staged concept. The other files not selected will be set aside for a later commit.

Syntax :git add <files path to add>

Example :git add main.py

Git **commit** command:

Description:

A commit is a Git entity that contains a list of changes made to files and that have been registered in the local repository. Making a commit, therefore, consists of archiving changes made to files that have been previously selected with the add command.

Syntax :git commit -m "<your commit message>"

Example :git commit -m “first commit”

Git **push** command:

Description:

When we make commits, they are stored in the local repository, and when we are ready to share them with the rest of the team for validation or deployment, we must publish them to the remote repository. To update a remote repository from commits made on a local repository, we use this command.

Syntax :git push <alias> <branch>

Example :git push origin master

Git **config** command:

Description:

Git configuration requires us to configure our username and email,

Syntax :

git config --global user.name "<your username>"

git config --global user.email "<your email>"

Example :

git config --global user.name "nagesh-s"

git config --global user.email "nagesh-s@gmail.com”

Git **clone** command:

Description:

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

Example :git clone https://github.com/nagesh-s-03/Jenkins_JAVA.git

Git **Reset** command:

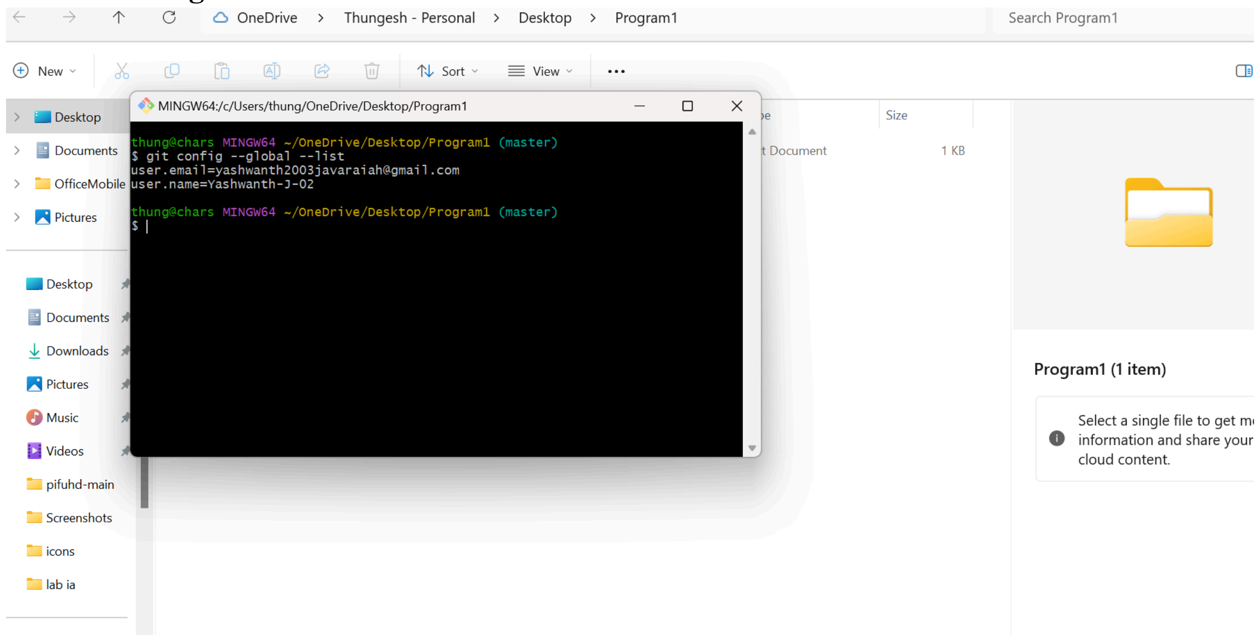
Description:

The term reset stands for undoing changes. The git reset command is used to reset the changes.

Syntax : git reset --hard <id>

Example :git reset –hard 6567b5e3e06421edf403e6bded67732563ef0985

Git user configuration details



MINGW64:/c/Users/thung/OneDrive/Desktop/Program1

```
thung@chars MINGW64 ~/OneDrive/Desktop/Program1 (master)
$ git config --global --list
user.email=yashwanth2003javaraiah@gmail.com
user.name=Yashwanth-J-02

thung@chars MINGW64 ~/OneDrive/Desktop/Program1 (master)
$ |
```

Configuration	Execution	Viva	Total	Verified By

A2. Demonstrate to create a project in remote repository and apply fork, merge, diff, merge conflict, branch and pull request concepts on repository using GitHub

Git **fork** command:

Description:

A fork is a rough copy of a repository. Forking a repository allows you to freely test and debug with changes without affecting the original project.

It is a straight-forward process. Steps for forking the repository are as follows:

o Login to the GitHub account.

o Find the GitHub repository which you want to fork.

o Click the Fork button on the upper right side of the repository's page.

Git **merge** command:

Description:

By default, when creating a repository, the code is placed in the main branch called master. In order to be able to isolate the developments of the master branch—for example, to develop a new feature, fix a bug, or even make technical experiments—we can create new branches from other branches and merge them together when we want to merge their code.

Syntax :git merge <branch name>

Example :git merge testing

Git **diff** command:

Description:

It compares the different versions of data sources. The version control system stands for working with a modified version of files. So, the diff command is a useful tool for working with Git.

Syntax :git diff <branch name>

Example :git diff master

Git **branch** command:

Description

A branch is a version of the repository that diverges from the main working project. It is a feature available in most modern version control systems. A Git project can have more than one branch.You can create a new branch with the help of the git branch command. This command will be used.

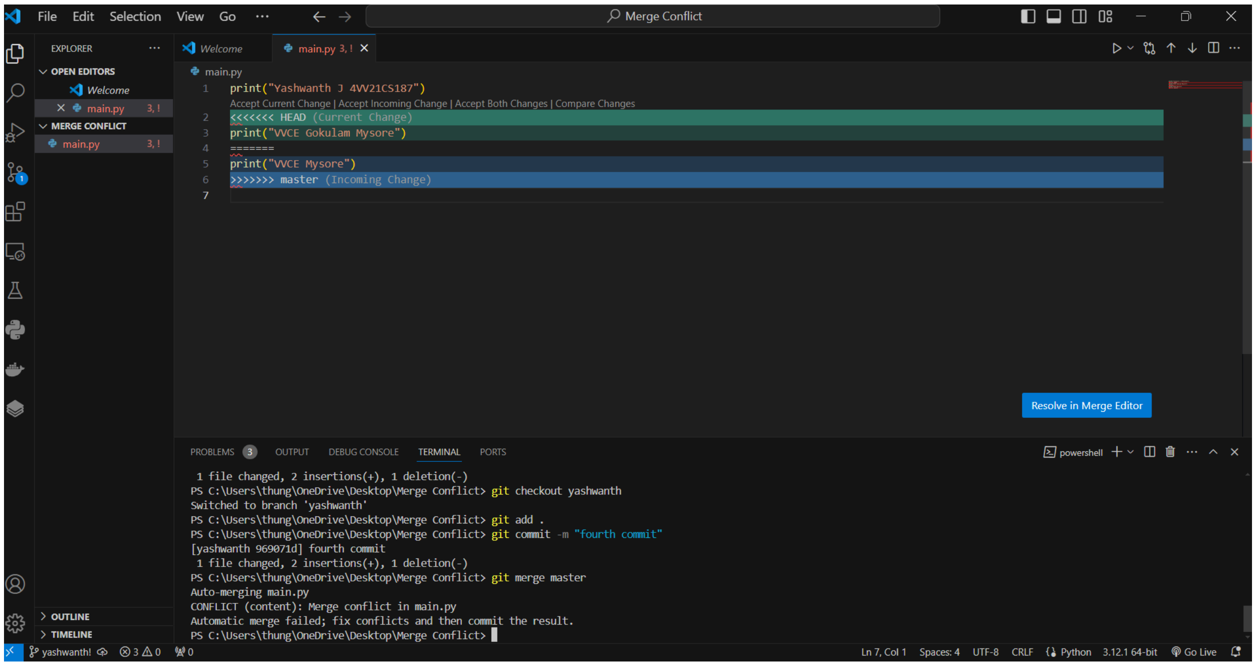
Syntax :git branch <branch name>

Example :git branch testing

Git **merge conflict** demonstration :

Description:

When two branches are trying to merge, and both are edited at the same time and in the same file, Git won't be able to identify which version is to take for changes. Such a situation is called merge conflict.



```
main.py
print("Yashwanth J 4VV21CS187")
<<<<<<< HEAD (Current Change)
print("VVCE Gokulam Mysore")
=======
print("VVCE Mysore")
>>>>>>> master (Incoming Change)
```

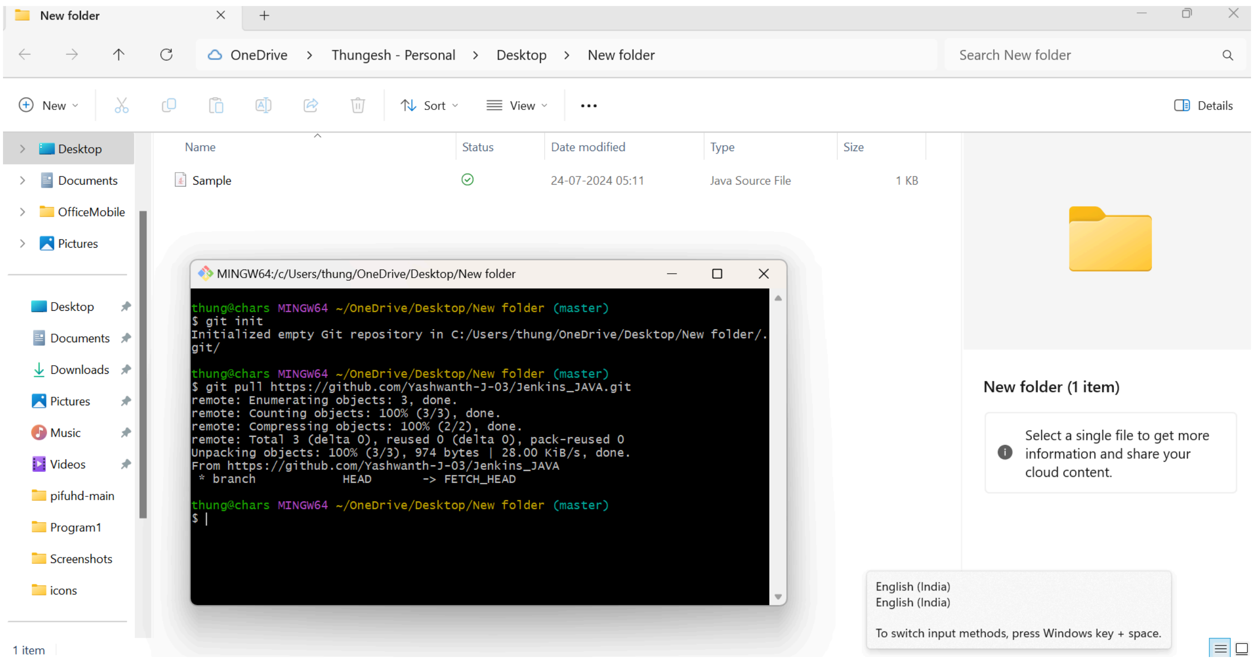
```
 1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\thung\OneDrive\Desktop\Merge Conflict> git checkout yashwanth
Switched to branch 'yashwanth'
PS C:\Users\thung\OneDrive\Desktop\Merge Conflict> git add .
PS C:\Users\thung\OneDrive\Desktop\Merge Conflict> git commit -m "fourth commit"
[yashwanth 969071d] fourth commit
 1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\thung\OneDrive\Desktop\Merge Conflict> git merge master
Auto-merging main.py
CONFLICT (content): Merge conflict in main.py
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\thung\OneDrive\Desktop\Merge Conflict>
```

Create **pull request** demonstration:

Description:

The term pull is used to receive data from GitHub. It fetches and merges changes from the remote server to your working directory. The git pull command is used to pull a repository.

Example :



Name	Status	Date modified	Type	Size
Sample	✓	24-07-2024 05:11	Java Source File	1 KB

MINGW64:/c/Users/thung/OneDrive/Desktop/New folder

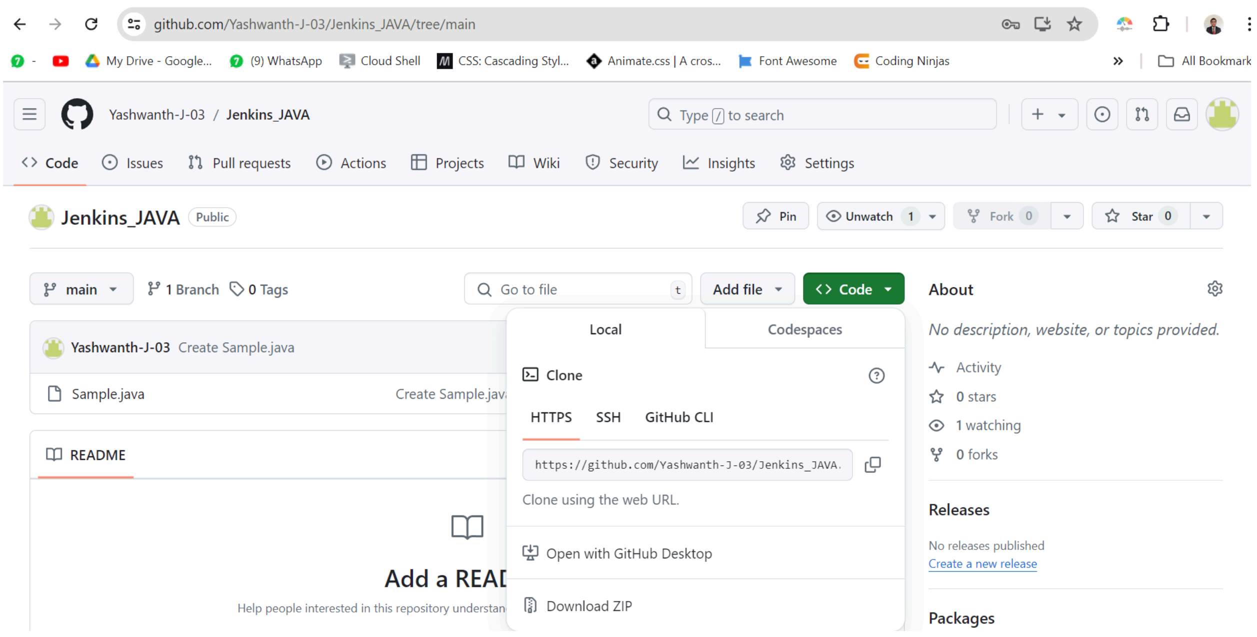
```
thung@chars MINGW64 ~/OneDrive/Desktop/New folder (master)
$ git init
Initialized empty Git repository in C:/Users/thung/OneDrive/Desktop/New folder/.
git/

thung@chars MINGW64 ~/OneDrive/Desktop/New folder (master)
$ git pull https://github.com/Yashwanth-J-03/Jenkins_JAVA.git
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 974 bytes | 28.00 KiB/s, done.
From https://github.com/Yashwanth-J-03/Jenkins_JAVA
 * branch            HEAD       -> FETCH_HEAD

thung@chars MINGW64 ~/OneDrive/Desktop/New folder (master)
$ |
```

New folder (1 item)

Select a single file to get more information and share your cloud content.

English (India)
English (India)

To switch input methods, press Windows key + space.


github.com/Yashwanth-J-03/Jenkins_JAVA/tree/main

Yashwanth-J-03 / Jenkins_JAVA

Jenkins_JAVA Public

main · 1 Branch · 0 Tags

Yashwanth-J-03 Create Sample.java

Sample.java — Create Sample.java

Clone — HTTPS

```
https://github.com/Yashwanth-J-03/Jenkins_JAVA.
```

Clone using the web URL.

About: No description, website, or topics provided. 0 stars, 1 watching, 0 forks. Releases: No releases published.

Configuration	Execution	Viva	Total	Verified By

A3. Demonstrate the process of integration github repository with Jenkins to automate the project execution in CI/CD pipeline.

Git Hub

Source code :

```
class Sample
{
    public static void main(String[]args)
    {
        System.out.println("nagesh-s");
        System.out.println("4vv21cs103");
    }
}
```

Jenkins Configuration

Required Plugin: Git

SCM

Git url : https://github.com/nagesh-s-03/Jenkins_JAVA.git

Branch Specifier: */main

Build Triggers

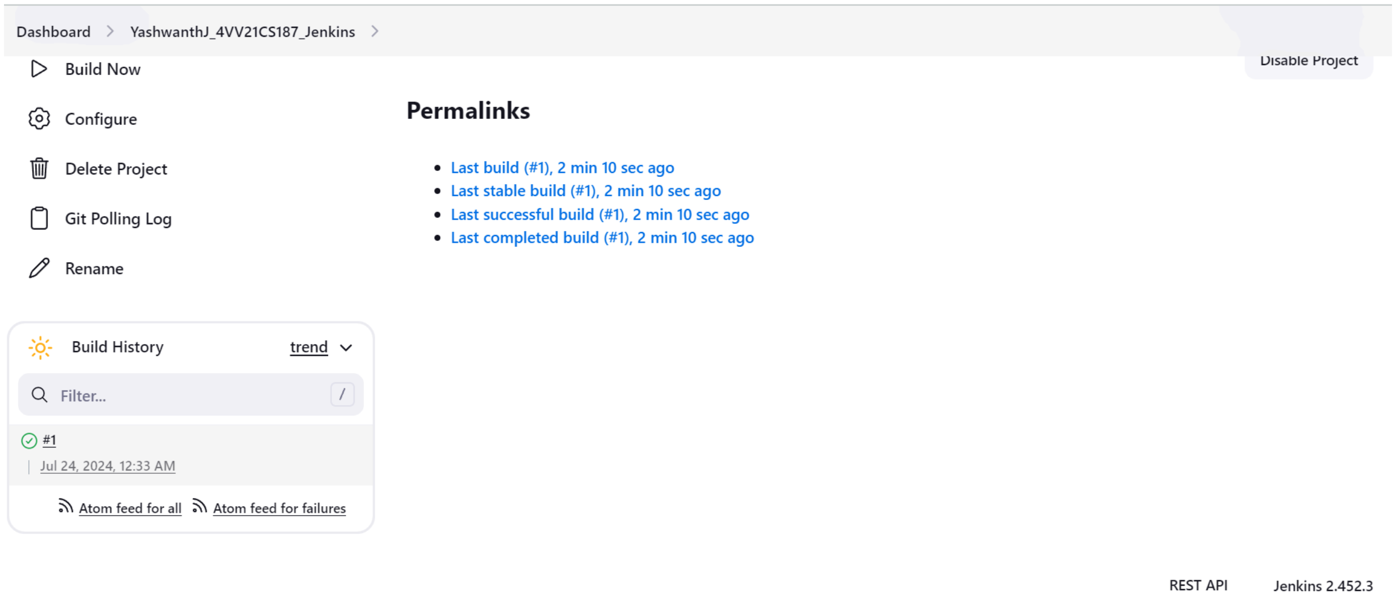
Poll SCM Schedule : * * * * *

Build Steps

Execute Windows batch command

Command : java Sample.java

Console Output:


Dashboard > YashwanthJ_4VV21CS187_Jenkins >

Disable Project

- Build Now
- Configure
- Delete Project
- Git Polling Log
- Rename

Permalinks

- Last build (#1), 2 min 10 sec ago
- Last stable build (#1), 2 min 10 sec ago
- Last successful build (#1), 2 min 10 sec ago
- Last completed build (#1), 2 min 10 sec ago

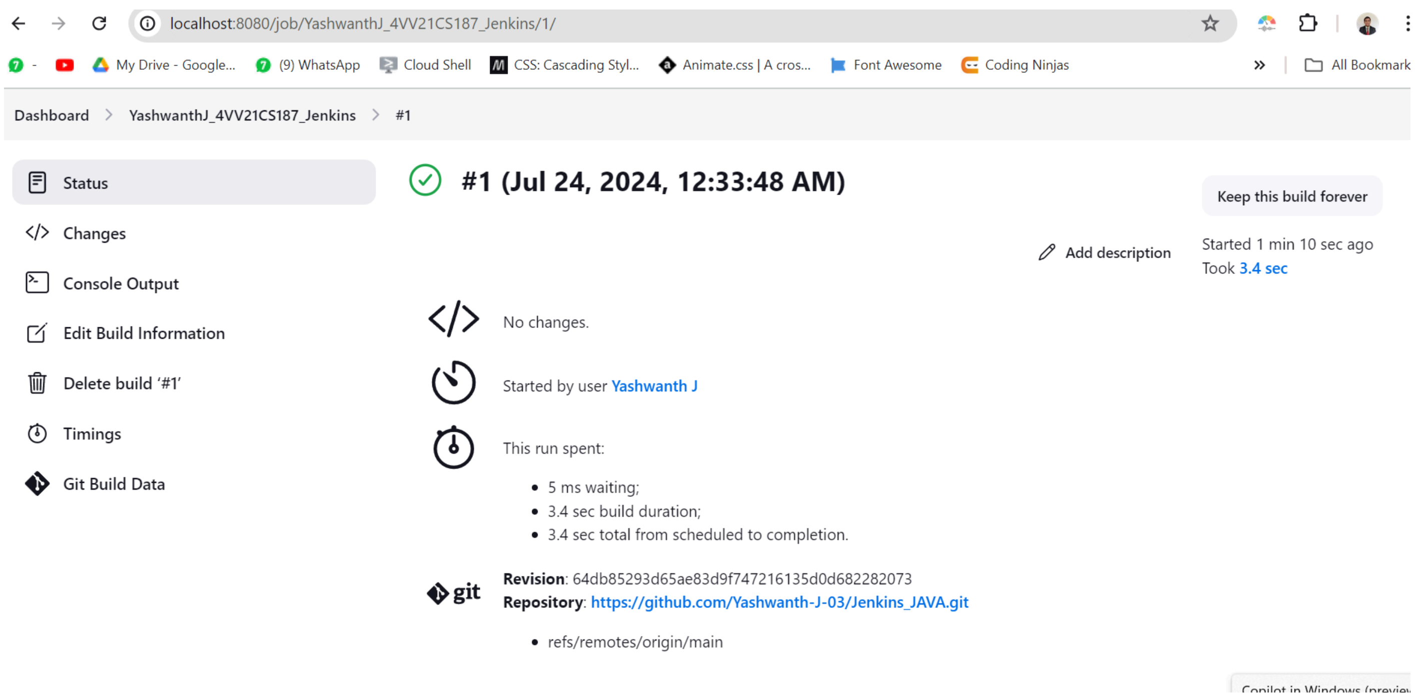
Build History trend

Filter...

#1
Jul 24, 2024, 12:33 AM

Atom feed for all Atom feed for failures

REST API Jenkins 2.452.3


localhost:8080/job/YashwanthJ_4VV21CS187_Jenkins/1/

Dashboard > YashwanthJ_4VV21CS187_Jenkins > #1

- Status
- Changes
- Console Output
- Edit Build Information
- Delete build '#1'
- Timings
- Git Build Data

#1 (Jul 24, 2024, 12:33:48 AM)

Keep this build forever

Add description

Started 1 min 10 sec ago
Took 3.4 sec

No changes.

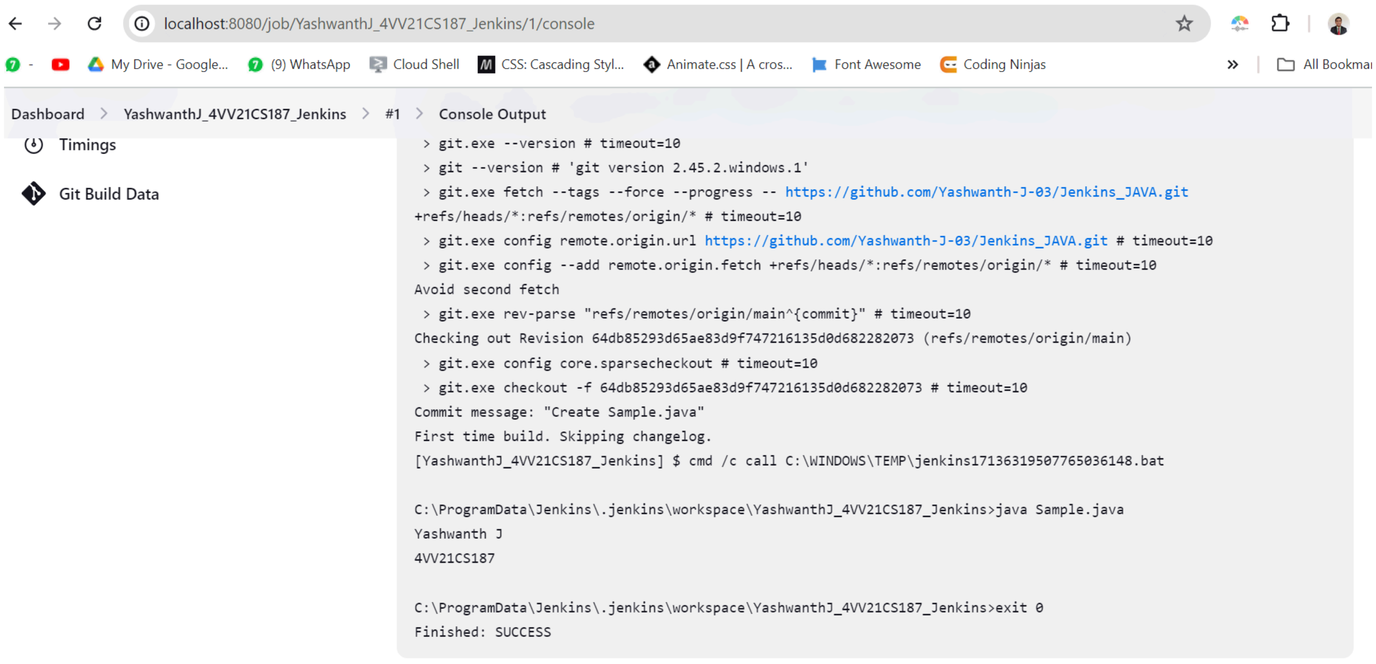
Started by user Yashwanth J

This run spent:

- 5 ms waiting;
- 3.4 sec build duration;
- 3.4 sec total from scheduled to completion.

Revision: 64db85293d65ae83d9f747216135d0d682282073
Repository: https://github.com/Yashwanth-J-03/Jenkins_JAVA.git

- refs/remotes/origin/main



```
 > git.exe --version # timeout=10
 > git --version # 'git version 2.45.2.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/Yashwanth-J-03/Jenkins_JAVA.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe config remote.origin.url https://github.com/Yashwanth-J-03/Jenkins_JAVA.git # timeout=10
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 64db85293d65ae83d9f747216135d0d682282073 (refs/remotes/origin/main)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 64db85293d65ae83d9f747216135d0d682282073 # timeout=10
Commit message: "Create Sample.java"
First time build. Skipping changelog.
[YashwanthJ_4VV21CS187_Jenkins] $ cmd /c call C:\WINDOWS\TEMP\jenkins17136319507765036148.bat

C:\ProgramData\Jenkins\.jenkins\workspace\YashwanthJ_4VV21CS187_Jenkins>java Sample.java
Yashwanth J
4VV21CS187

C:\ProgramData\Jenkins\.jenkins\workspace\YashwanthJ_4VV21CS187_Jenkins>exit 0
Finished: SUCCESS
```

Configuration	Execution	Viva	Total	Verified By

B1. Create a docker image for an application stored in local repository and run the application using docker image.

Application Source code

```
class Sample
{
    public static void main(String[]args)
    {
        System.out.println("nagesh-s");
        System.out.println("4vv21cs103");
    }
}
```

Dockerfile:

```
      FROM openjdk
      WORKDIR /app
      COPY . /app
      RUN javac Sample.java
      CMD ["java","Sample"]
```

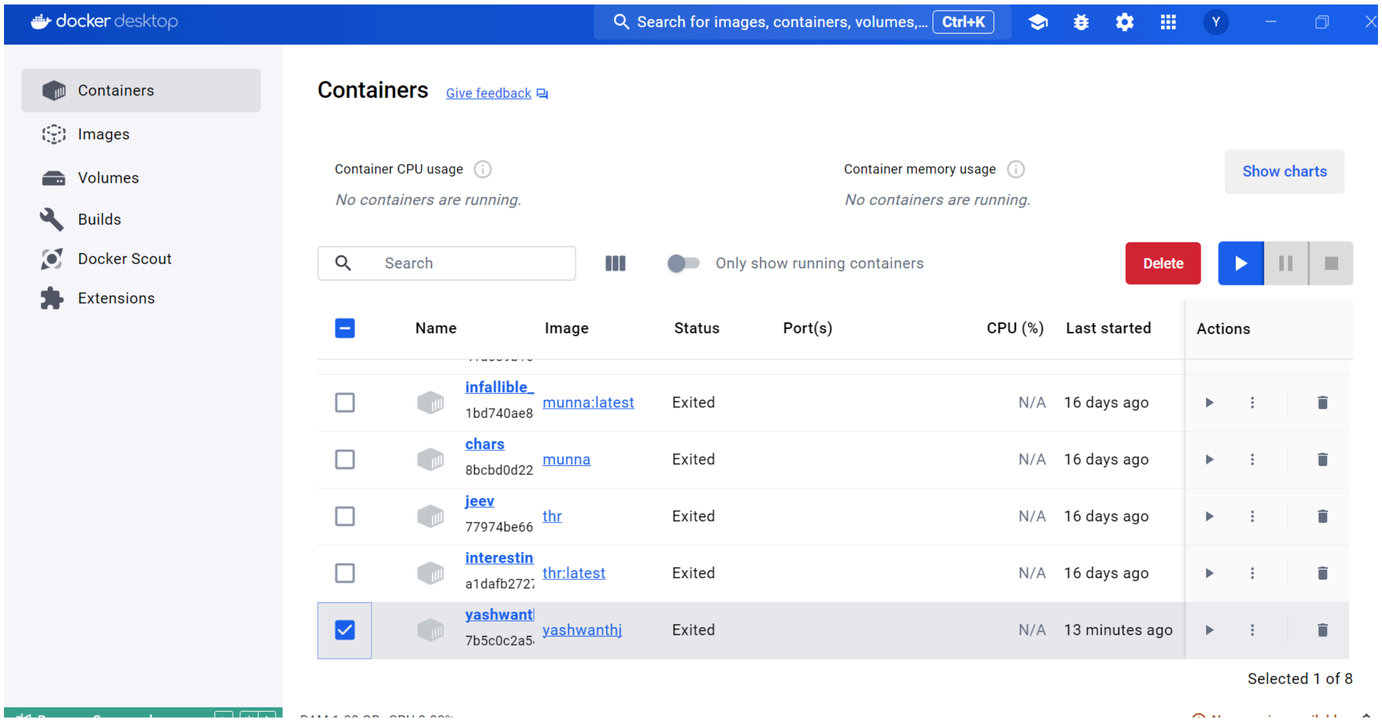
Command to Build image: docker build -t <name of image> .

docker build -t javaapp.

Command to run the image: docker run –name <container name> <name of image>

Docker run –name javaapp javaapp

Docker container



Containers

Container CPU usage
No containers are running.

Container memory usage
No containers are running.

	Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
☐	infallible_ 1bd740ae8	munna:latest	Exited		N/A	16 days ago	
☐	chars 8bcbd0d22	munna	Exited		N/A	16 days ago	
☐	jeev 77974be66	thr	Exited		N/A	16 days ago	
☐	interestin a1dafb2727	thr:latest	Exited		N/A	16 days ago	
☑	yashwantl 7b5c0c2a5	yashwanthj	Exited		N/A	13 minutes ago	

Selected 1 of 8

Configuration	Execution	Viva	Total	Verified By

B2. Create and configure Jenkins files for workflow and build of an application and push the image

Application Source code:

```
class Sample
{
    public static void main(String[]args)
    {
        System.out.println("nagesh-s");
        System.out.println("4vv21cs103");
    }
}
```

Dockerfile

```
      FROM openjdk
      WORKDIR /app
      COPY . /app
      RUN javac Sample.java
      CMD ["java","Sample"]
```

Command to Build image: docker build -t <name of image> .

docker build -t nagesh-s .

Command to tag the image :

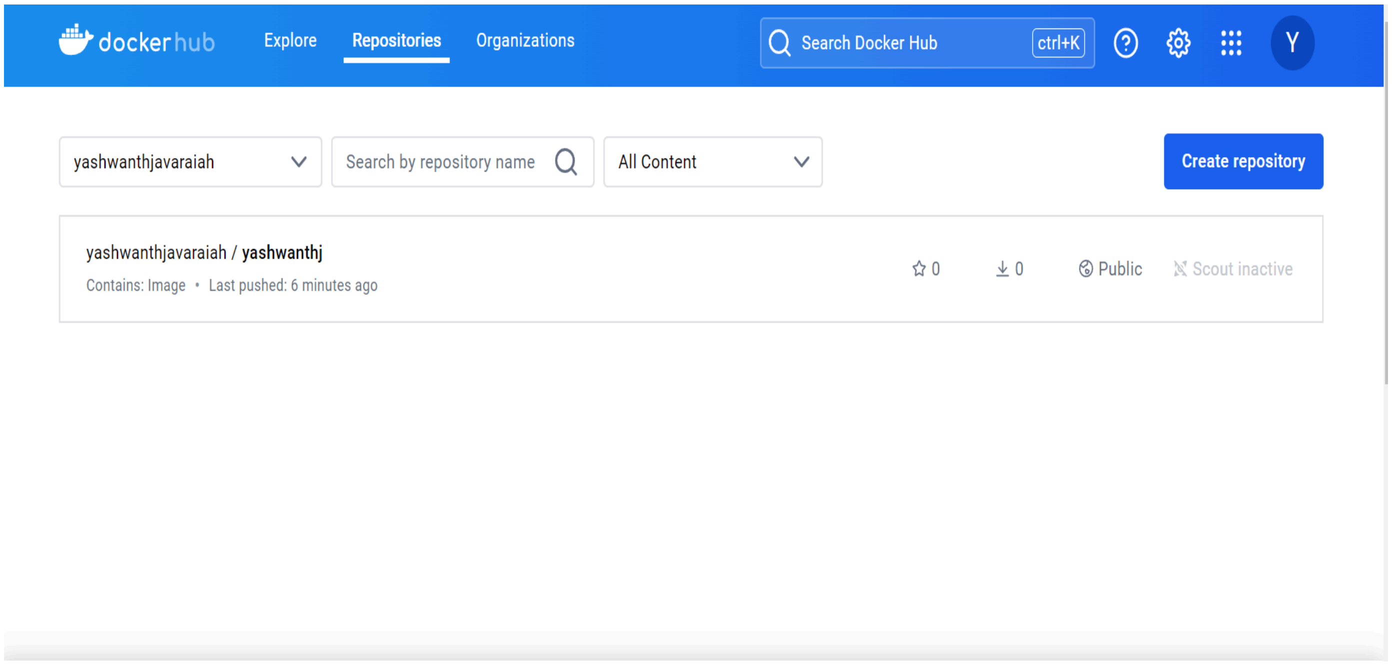
docker tag <Image id> <dockerhub login>/<name of image>:v1

docker tag 6e87198b9dfe nagesh-s/nagesh-s:v1

Command to push the image : docker push docker.io/<dockerhub login>/<name of image>:v1

docker push docker.io/nagesh-s/nagesh-s:v1

Docker Hub



yashwanthjavaraiah / **yashwanthj**

Contains: Image • Last pushed: 6 minutes ago

☆ 0 ⤓ 0 Public Scout inactive

Configuration	Execution	Viva	Total	Verified By

C1. Create a maven projects with all dependencies required for the application in CI/CD pipeline.

Maven Dependency:

```
<!-- https://mvnrepository.com/artifact/com.googlecode.json-simple/json-simple -->
<dependency>
   <groupId>com.googlecode.json-simple</groupId>
   <artifactId>json-simple</artifactId>
   <version>1.1.1</version>
</dependency>
```

JSON data:

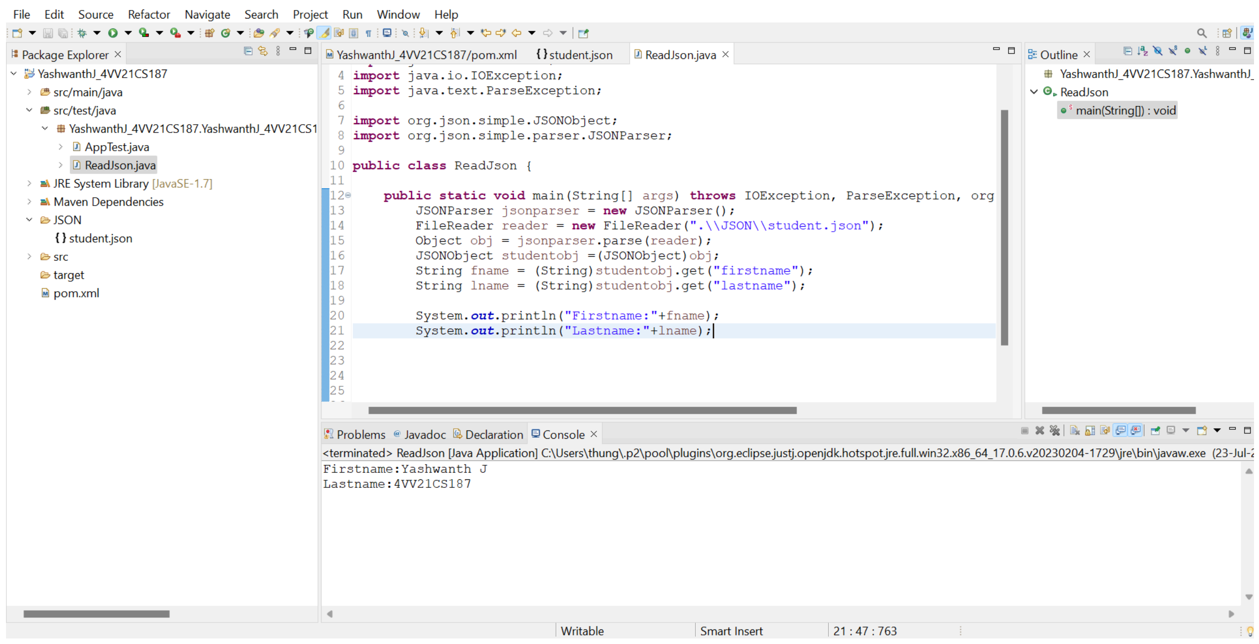
```
{"firstname":"nagesh-s",

"lastname":"4vv21cs103"}
```

Java Program to read data from JSON file

```
package nagesh-s_4vv21cs103.nagesh-s_4vv21cs103;

import java.io.FileReader;

import java.io.IOException;

import java.text.ParseException;

import org.json.simple.JSONObject;

import org.json.simple.parser.JSONParser;

public class ReadJson {

      public static void main(String[] args) throws IOException,
ParseException, org.json.simple.parser.ParseException{

            JSONParser jsonparser = new JSONParser();

            FileReader reader = new FileReader(".\\JSON\\student.json");

            Object obj = jsonparser.parse(reader);

            JSONObject studentobj =(JSONObject)obj;

            String fname = (String)studentobj.get("firstname");

            String lname = (String)studentobj.get("lastname");
```

```
                System.out.println("Firstname:"+fname);

                System.out.println("Lastname:"+lname);




        }

}
```

Program output:



```
import java.io.IOException;
import java.text.ParseException;

import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;

public class ReadJson {

    public static void main(String[] args) throws IOException, ParseException, org
        JSONParser jsonparser = new JSONParser();
        FileReader reader = new FileReader(".\\JSON\\student.json");
        Object obj = jsonparser.parse(reader);
        JSONObject studentobj =(JSONObject)obj;
        String fname = (String)studentobj.get("firstname");
        String lname = (String)studentobj.get("lastname");

        System.out.println("Firstname:"+fname);
        System.out.println("Lastname:"+lname);
```

```
<terminated> ReadJson [Java Application] C:\Users\thung\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (23-Jul-
Firstname:Yashwanth J
Lastname:4VV21CS187
```

Configuration	**Execution**	**Viva**	**Total**	**Verified By**

C2. Integrate communication channel with Jenkins for status of project and also enable email notification for a build.

Slack Group Name : vvcegroup

Slack Channel Name : #devops-demo

Slack plugins:Slack notification plugin

Steps to get secret text :

1. Goto Manage Jenkins
2. Goto plugins
3. Click on Slack Notification plugin
4. Goto https://my.slack.com/services/new/jenkins-ci
5. Select the channel
6. You will get secret key

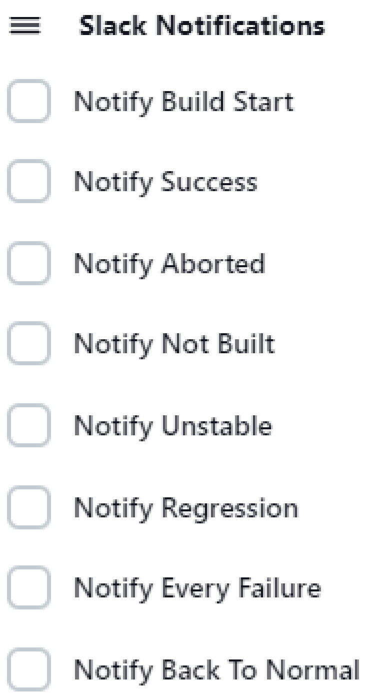
Secret Text :`jg3sifiR9GkRCEsPaLE8gewX`

Name of the project :nagesh-s_4vv21cs103

Windows Execution batch command: java –version

Enabled Post build actions:

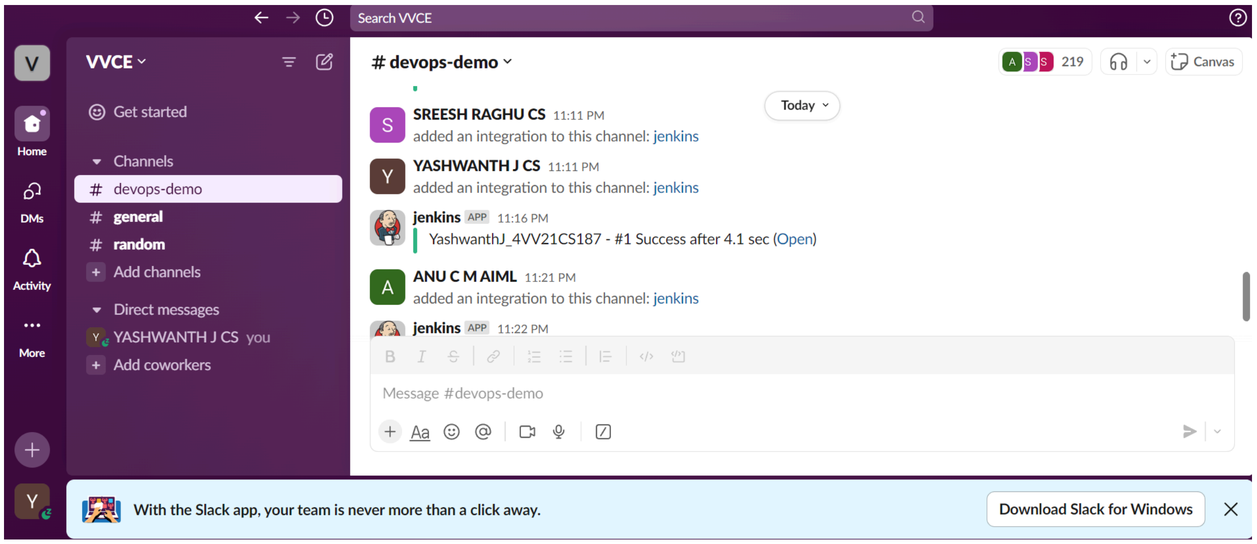
Notify Success
Notify Every Failure
Notify Unstable



Slack Notifications

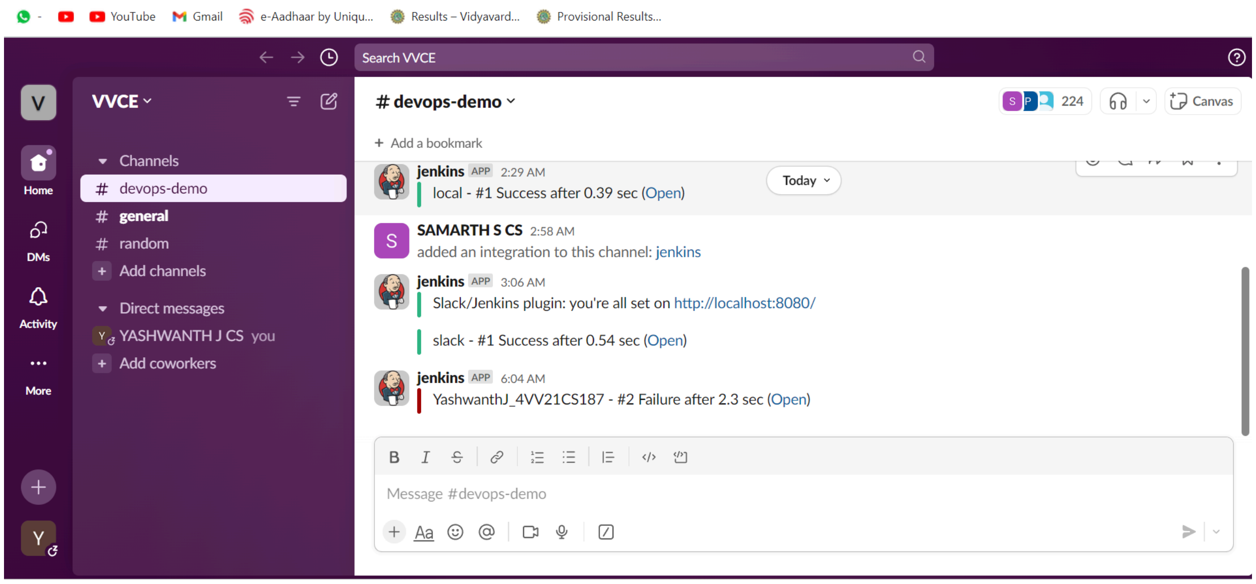
- ☐ Notify Build Start
- ☐ Notify Success
- ☐ Notify Aborted
- ☐ Notify Not Built
- ☐ Notify Unstable
- ☐ Notify Regression
- ☐ Notify Every Failure
- ☐ Notify Back To Normal

Program output:



```
SREESH RAGHU CS 11:11 PM
added an integration to this channel: jenkins
YASHWANTH J CS 11:11 PM
added an integration to this channel: jenkins
jenkins APP 11:16 PM
YashwanthJ_4VV21CS187 - #1 Success after 4.1 sec (Open)
ANU C M AIML 11:21 PM
added an integration to this channel: jenkins
jenkins APP 11:22 PM
```

With the Slack app, your team is never more than a click away.



jenkins APP 2:29 AM
local - #1 Success after 0.39 sec (Open)

SAMARTH S CS 2:58 AM
added an integration to this channel: jenkins

jenkins APP 3:06 AM
Slack/Jenkins plugin: you're all set on http://localhost:8080/

slack - #1 Success after 0.54 sec (Open)

jenkins APP 6:04 AM
YashwanthJ_4VV21CS187 - #2 Failure after 2.3 sec (Open)

Configuration	Execution	Viva	Total	Verified By