

Qu1: Integration with Jenkins and GitHub

Automate test execution via CI/CD.

Push a Selenium project to GitHub.

Setup a Jenkins pipeline to trigger on every push.

Generate and email test reports using ExtentReports

Ans:

Create a selenium Test File in a Local repository

Eg: sele.py:

```
import pytest

from selenium import webdriver

from selenium.webdriver.chrome.service import Service

from webdriver_manager.chrome import ChromeDriverManager


def test_open_site():

    driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))

    driver.get('https://selenium.dev/')

    driver.quit()
```

Create a requirements.txt file with content:

```
selenium
webdriver-manager
pytest
pytest-extent-reporter
```

Create a pytest.ini with content:

```
[pytest]
```

```
addopts =--extent-html=report.html
```

Create JenkinsFile with content:

```
pipeline {  
    agent any  
  
    stages {  
        stage('Checkout Code') {  
            steps {  
                git branch: 'main', url: 'https://github.com/ShivRaiGithub/DevopsAssess.git'  
            }  
        }  
        stage('Install Dependencies') {  
            steps {  
                sh 'pip3 install-r requirements.txt'  
            }  
        }  
        stage('Run Selenium Test') {  
            steps {  
                sh 'pytest sele.py'  
            }  
        }  
        stage('Archive Test Report') {
```

```
    steps {
        archiveArtifacts artifacts: 'report.html', fingerprint: true
    }
}
stage('Send Email') {
    steps {
        emailext(
            subject: 'Selenium Test Report',
            body: 'Test report.',
            attachmentsPattern: 'report.html',
            to: 'sshaktirai@gmail.com'
        )
    }
}
}
```

Create a new github repo

Push local repo to github using steps:

```
git init
```

```
git add .
```

```
git commit-m "first commit"
```


```
git branch-M main
```

```
git remote add origin https://github.com/ShivRaiGithub/DevopsAssess.git
```

git push-u origin main

DevopsAssessPublic


PinUnwatch1Fork0Star0



### Start coding with Codespaces

Add a README file and start coding in a secure, configurable, and dedicated development environment.

Create a codespace



### Add collaborators to this repository

Search for people using their GitHub username or email address.

Invite collaborators

#### Quick setup — if you've done this kind of thing before

Set up in DesktoporHTTPSSSHhttps://github.com/ShivRaiGithub/DevopsAssess.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

#### ...or create a new repository on the command line

```
echo "# DevopsAssess" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ShivRaiGithub/DevopsAssess.git
git push -u origin main
```

#### ...or push an existing repository from the command line

```
git remote add origin https://github.com/ShivRaiGithub/DevopsAssess.git
git branch -M main
git push -u origin main
```

DevopsAssessPublic

PinUnwatch1

main1 Branch0 Tags

Go to filetAdd fileCode

ShivRaiGithubJenkinsc22277c · 5 minutes ago4 Commits

JenkinsFile	Jenkins	5 minutes ago
pytest.ini	Jenkins	5 minutes ago
requirements.txt	Jenkins	5 minutes ago
sele.py	Jenkins	5 minutes ago

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● shiv@Shiv:~/Codes/college/DevsAssess$ git init
Reinitialized existing Git repository in /home/shiv/Codes/college/DevsAssess/.git/
● shiv@Shiv:~/Codes/college/DevsAssess$ git add .
● shiv@Shiv:~/Codes/college/DevsAssess$ git commit -m "First commit"
[main (root-commit) 8e9bcc0] First commit
 1 file changed, 6 insertions(+)
 create mode 100644 sele.py
● shiv@Shiv:~/Codes/college/DevsAssess$ git remote add origin https://github.com/ShivRaiGithub/DevopsAssess.git
● shiv@Shiv:~/Codes/college/DevsAssess$ git push
fatal: The current branch main has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin main

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
● shiv@Shiv:~/Codes/college/DevsAssess$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 305 bytes | 305.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ShivRaiGithub/DevopsAssess.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
○ shiv@Shiv:~/Codes/college/DevsAssess$
```

Qu2: Version control basics and collaboration using Git.

Initialize a Git repository and perform basic commands (init, add, commit, status, log).

Clone a repo, create a new branch, and simulate a merge conflict.

Resolve the conflict and push changes to GitHub.

Ans:

Create a folder

Create a new file file.txt with content:

Hello world

Welcome to Devops

Push to Github using commands:

```
git init
```

```
git add .
```

```
git commit-m "first commit"
```

```
git branch-M main
```

```
git remote add origin https://github.com/ShivRaiGithub/DevopsOnCloud.git
```

```
git push-u origin main
```

ShivRaiGithub/DevopsOnCloud

github.com/ShivRaiGithub/DevopsOnCloud

ShivRaiGithub / DevopsOnCloud

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

DevopsOnCloud

Public

Pin

Unwatch 1

Fork 0

Star 0

**Start coding with Codespaces**  
Add a README file and start coding in a secure, configurable, and dedicated development environment.  
[Create a codespace](#)

**Add collaborators to this repository**  
Search for people using their GitHub username or email address.  
[Invite collaborators](#)

**Quick setup — if you've done this kind of thing before**

Set up in Desktop

 or 

HTTPS

SSH

`https://github.com/ShivRaiGithub/DevopsOnCloud.git`

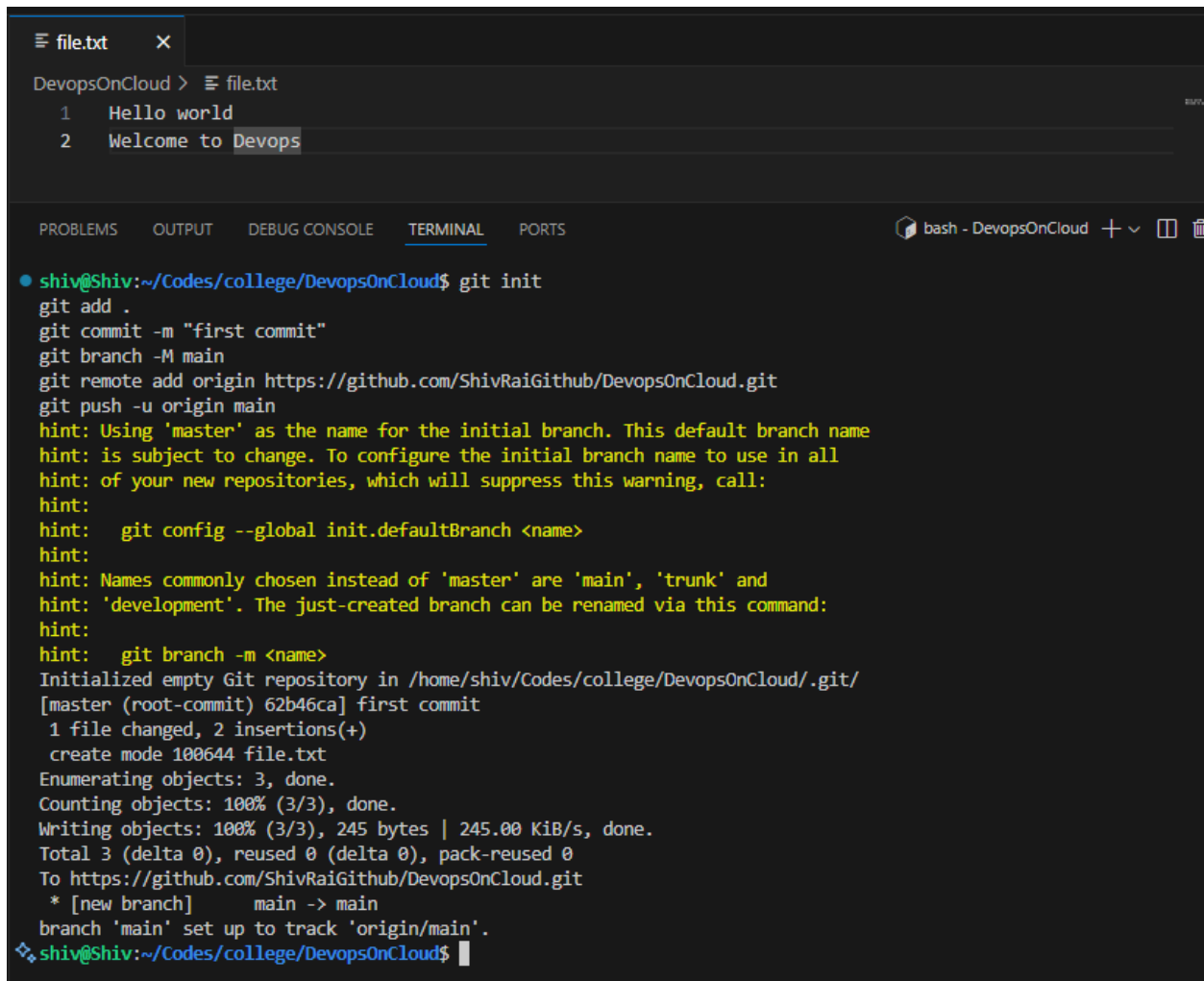
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# DevopsOnCloud" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ShivRaiGithub/DevopsOnCloud.git
git push -u origin main
```

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/ShivRaiGithub/DevopsOnCloud.git
git branch -M main
git push -u origin main
```



```
file.txt x
DevopsOnCloud > file.txt
1 Hello world
2 Welcome to Devops

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - DevopsOnCloud + v

● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git init
git add .
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ShivRaiGithub/DevopsOnCloud.git
git push -u origin main
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/shiv/Codes/college/DevopsOnCloud/.git/
[master (root-commit) 62b46ca] first commit
1 file changed, 2 insertions(+)
create mode 100644 file.txt
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 245 bytes | 245.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ShivRaiGithub/DevopsOnCloud.git
* [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
❖ shiv@Shiv:~/Codes/college/DevopsOnCloud$
```

Use following commands to check the status and previous commits:

git status

git log



```

● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git status
On branch main
Your branch is up to date with 'origin/main'.

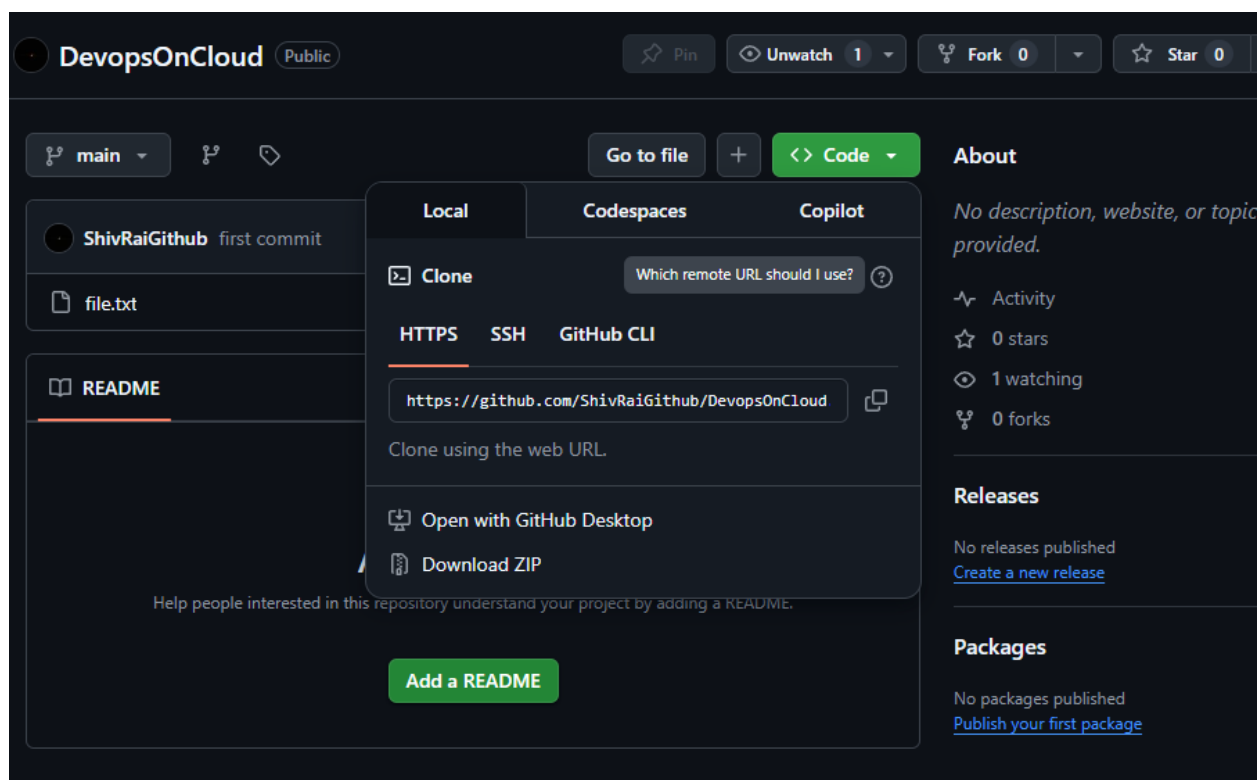
nothing to commit, working tree clean
● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git log
commit 62b46caab09a7e340c6a31a6d2a8fca2f0b776f7 (HEAD -> main, origin/main)
Author: ShivRaiGithub <sshaktirai@gmail.com>
Date: Sat Apr 26 08:01:14 2025 +0000

    first commit
○ shiv@Shiv:~/Codes/college/DevopsOnCloud$

```

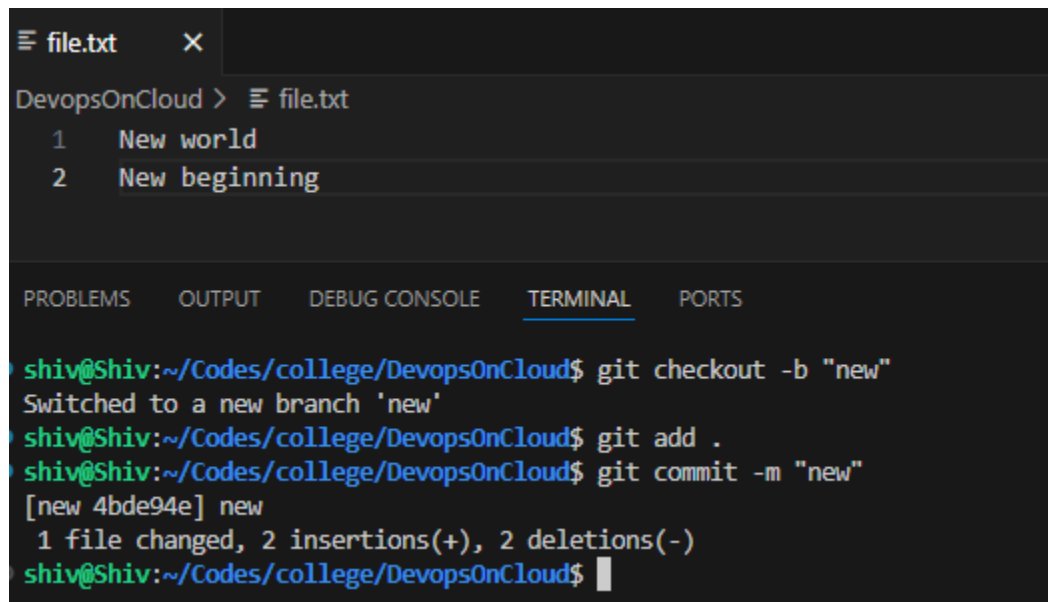
Use the following to clone:

git clone <https://github.com/ShivRaiGithub/DevopsOnCloud.git>



```
● shiv@Shiv:~/Codes/college/temp$ git clone https://github.com/ShivRaiGithub/DevopsOnCloud.git
Cloning into 'DevopsOnCloud'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
○ shiv@Shiv:~/Codes/college/temp$
```

Make a new branch, do some changes and commit them:



```
file.txt x
DevopsOnCloud > file.txt
1 New world
2 New beginning

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git checkout -b "new"
Switched to a new branch 'new'
● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git add .
● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git commit -m "new"
[new 4bde94e] new
1 file changed, 2 insertions(+), 2 deletions(-)
● shiv@Shiv:~/Codes/college/DevopsOnCloud$
```

On the main branch do some changes and commit them:

```
file.txt x
DevOpsOnCloud > file.txt
1 Hello world
2 Welcome to this world

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

shiv@Shiv:~/Codes/college/DevOpsOnCloud$ git add .
shiv@Shiv:~/Codes/college/DevOpsOnCloud$ git commit -m "this world"
[main 99a0507] this world
1 file changed, 1 insertion(+), 1 deletion(-)
shiv@Shiv:~/Codes/college/DevOpsOnCloud$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 276 bytes | 276.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ShivRaiGithub/DevOpsOnCloud.git
62b46ca..99a0507 main -> main
shiv@Shiv:~/Codes/college/DevOpsOnCloud$
```

Switch to new branch and merge the new branch with main. Conflicts will occur. Resolve them

```
file.txt ! x
DevOpsOnCloud > file.txt
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
1 <<<<<< HEAD (Current Change)
2 New world
3 New beginning
4 =====
5 Hello world
6 Welcome to this world
7 >>>>>> main (Incoming Change)
8

Resolve in Merge Editor

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash - DevOpsOnCloud

shiv@Shiv:~/Codes/college/DevOpsOnCloud$ git checkout "new"
Switched to branch 'new'
Your branch is up to date with 'origin/new'.
shiv@Shiv:~/Codes/college/DevOpsOnCloud$ git merge main
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.
shiv@Shiv:~/Codes/college/DevOpsOnCloud$
```

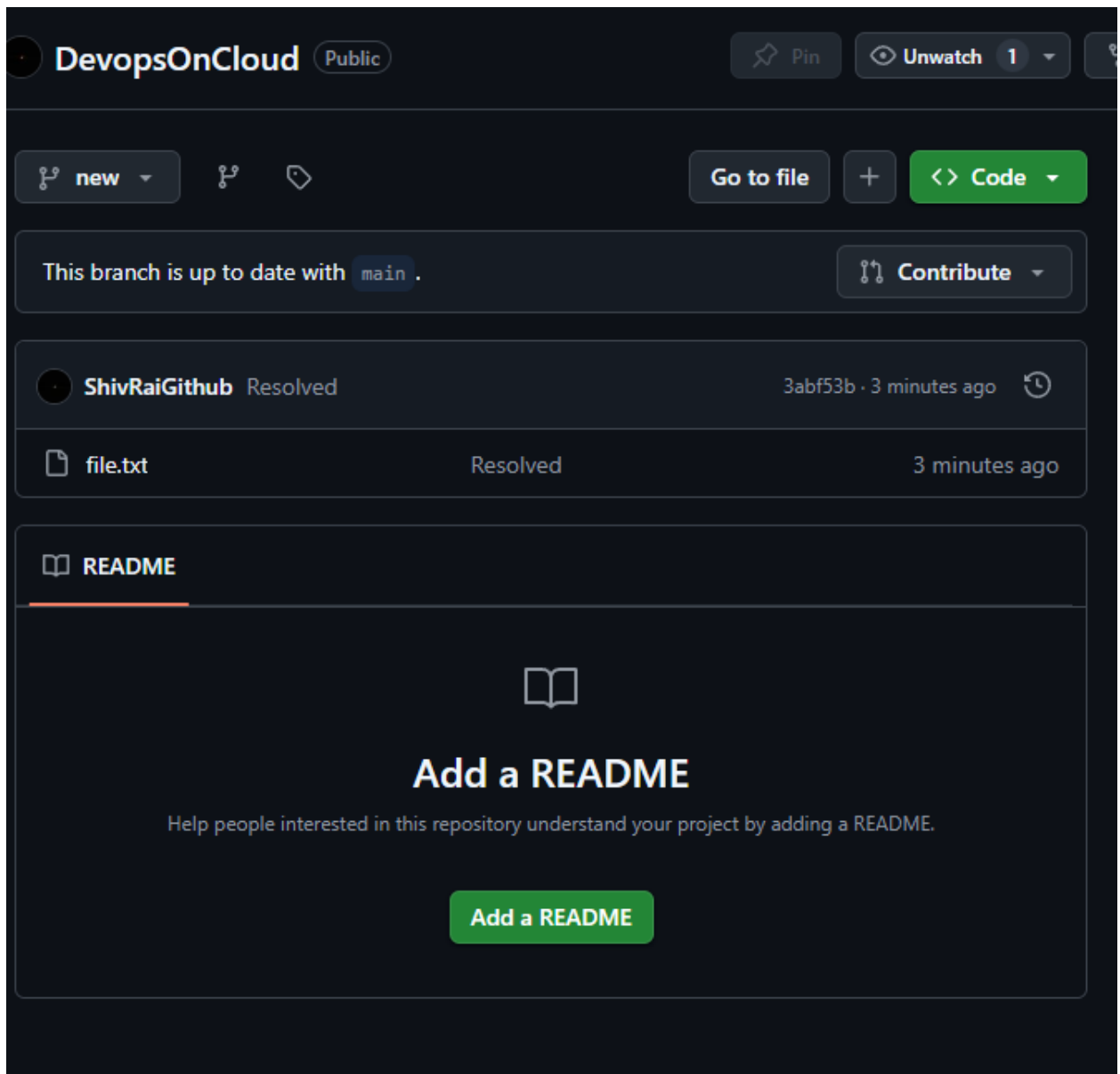
After resolution, push the changes

```
● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git branch
  main
* new
● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git add .
● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git commit -m "Resolved"
[new 3abf53b] Resolved
● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 322 bytes | 322.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ShivRaiGithub/DevopsOnCloud.git
4bde94e..3abf53b  new -> new
```

Push changes of main branch too:

```
● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git branch
  main
* new
● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git checkout "main"
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git merge new
Updating 99a0507..3abf53b
Fast-forward
 file.txt | 6 +++++-
1 file changed, 5 insertions(+), 1 deletion(-)
⊗ shiv@Shiv:~/Codes/college/DevopsOnCloud$ git commit -m "Changes on main"
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
● shiv@Shiv:~/Codes/college/DevopsOnCloud$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ShivRaiGithub/DevopsOnCloud.git
99a0507..3abf53b  main -> main
○ shiv@Shiv:~/Codes/college/DevopsOnCloud$ █
```



Qu3: Automate pulling code from GitHub and building it.

Link Jenkins with your GitHub repo.

Use Git plugin in the Source Code Management section.

Add a shell script in the build step to print a list of files (ls -la).

Trigger the job manually.

Ans:

Go to Jenkins Dashboard

Make a New Item:

## New Item

Enter an item name

DevopsAssess

Select an item type



### Freestyle project

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



### Pipeline

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



### Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



### Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



### Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.



### Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

OK

In source code management of Configuration, select git and provide github repo details:

Dashboard > DevopsAssess > Configuration

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

None

Git

Repositories

Repository URL

https://github.com/ShivRaiGithub/DevopsAssess.git

Credentials

- none -

+ Add

Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any')

\*/main

Add Branch

Add the build command to show the files in Build Steps:



## Build Steps

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

Add build step ^

Filter

Execute Windows batch command

Execute shell

Invoke Ant

Invoke Gradle script

Invoke top-level Maven targets

Set build status to "pending" on GitHub commit

sending notifications, archiving

## Build Steps

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

≡ **Execute shell** ?

×

Command

See [the list of available environment variables](#)

```
ls -a
```

Advanced ▾

Save the Configuration.

Go to dashboard, click on job, click on Build Now.

Check console logs.

### ✓ Console Output



```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/DevopsAssess
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/ShivRaiGithub/DevopsAssess.git
> git init /var/lib/jenkins/workspace/DevopsAssess # timeout=10
Fetching upstream changes from https://github.com/ShivRaiGithub/DevopsAssess.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/ShivRaiGithub/DevopsAssess.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/ShivRaiGithub/DevopsAssess.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision c22277c4b5b1fcab81d6a0231ced7a455343a863 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f c22277c4b5b1fcab81d6a0231ced7a455343a863 # timeout=10
Commit message: "Jenkins"
First time build. Skipping changelog.
[DevopsAssess] $ /bin/sh -xe /tmp/jenkins15579841157969413485.sh
+ ls -la
.
..
.git
JenkinsFile
pytest.ini
requirements.txt
sele.py
Finished: SUCCESS
```

Qu4: Scheduled Jobs (CRON Trigger) Automate job execution at intervals.

Set Build Trigger using Jenkins' cron syntax (H/15 \* \* \* \* for every 15 mins).

Print the system time during execution to confirm the job is running.

Ans:

Go to Jenkins Dashboard, select job.

Go to Configure, select Build Periodically under Triggers. Put:

H/15 \* \* \* \*

Dashboard > DevopsAssess > Configuration

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

Triggers

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


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

☐ Build after other projects are built ?

☒ Build periodically ?

Schedule ?

H/15 \* \* \* \*

 No schedules so will never run

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

Within Build Steps, add line:

echo "System Time: \$(date)"

## Configure

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

☐ With Ant ?

### Build Steps

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

#### Execute shell ?

Command

See [the list of available environment variables](#)

```
ls -a
echo "System Time: $(date)"
```

Advanced ▾

Add build step ▾

Save the configuration.

The output of next build will show the files of the repo as well as system time

Status

Changes

**Console Output**

Edit Build Information

Delete build '#3'

Timings

Git Build Data

Previous Build

### Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/DevopsAssess
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/DevopsAssess/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/ShivRaiGithub/DevopsAssess.git # timeout=10
Fetching upstream changes from https://github.com/ShivRaiGithub/DevopsAssess.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/ShivRaiGithub/DevopsAssess.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision c2277c4b5b1fcab81d6a0231ced7a455343a863 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f c2277c4b5b1fcab81d6a0231ced7a455343a863 # timeout=10
Commit message: "Jenkins"
> git rev-list --no-walk c2277c4b5b1fcab81d6a0231ced7a455343a863 # timeout=10
[DevopsAssess] $ /bin/sh -xe /tmp/jenkins17282299698920406285.sh
+ ls -a
.
..
.git
JenkinsFile
pytest.ini
requirements.txt
sele.py
+ date
+ echo System Time: Sat Apr 26 08:33:01 UTC 2025
System Time: Sat Apr 26 08:33:01 UTC 2025
Finished: SUCCESS
```