

Introduction to DevOps

Assignment 1 - GitHub Assignment

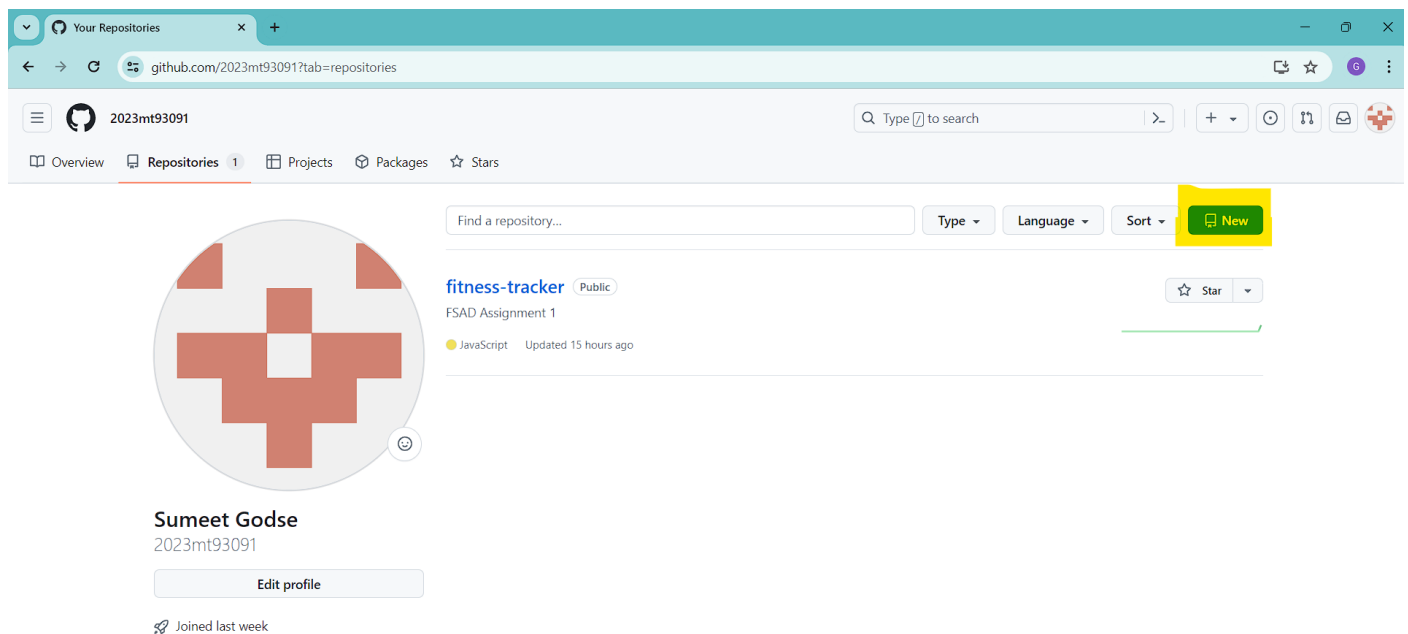
By - Godse Sumeet Deelip (2023mt93091)

GitHub - <https://github.com/2023mt93091/devops-assignment-1>

Problem Statement - ABC Organization would like to opt for the distributed version control system to upgrade their environment; where Git has been selected as the solution. You been assigned as a consultant to educate the migration process to move their Source Code from Centralized to Distributed systems. As a phase one, you would like to go ahead with a workshop to demonstrate below operation to make the ABC team comfortable.

Part 1 -

1. Open GitHub and create a repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 2023mt93091 ▾

Repository name *



/ devops-assignment-1

✔ devops-assignment-1 is available.

Great repository names are short and memorable. Need inspiration? How about **cautious-broccoli** ?

Description (optional)

Introduction to DevOps - Assignment 1 - GitHub and Jenkins demo

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

- ☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: None ▾


Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

The screenshot shows the GitHub interface for a repository named 'devops-assignment-1' by user '2023mt93091'. The repository is public and has 1 commit. The main branch is 'main'. The README file is visible, titled 'devops-assignment-1', with the content 'Introduction to DevOps - Assignment 1 - GitHub and Jenkins demo'. The right sidebar contains sections for 'About' (Introduction to DevOps - Assignment 1 - GitHub and Jenkins demo), 'Readme', 'Activity', '0 stars', '1 watching', '0 forks', 'Releases' (No releases published, Create a new release), and 'Packages' (No packages published, Publish your first package).

2. Add two or more directories and some raw code files to the repository. I will be creating a react based application here and adding the same to the repository.

```
sumee@LAPTOP-I8BBG50J MINGW64 /d/DevProjects
$ git clone https://github.com/2023mt93091/devops-assignment-1.git
Cloning into 'devops-assignment-1'...
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
Receiving objects: 100% (3/3), done.

sumee@LAPTOP-I8BBG50J MINGW64 /d/DevProjects
$ cd devops-assignment-1/

sumee@LAPTOP-I8BBG50J MINGW64 /d/DevProjects/devops-assignment-1 (main)
$ ls
README.md
```

```
sumee@LAPTOP-I8BBG50J MINGW64 /d/DevProjects/devops-assignment-1 (main)
$ yarn create vite
yarn create v1.22.10
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Installed "create-vite@5.2.3" with binaries:
- create-vite
- cva
✓ Project name: ... devops
✓ Select a framework: » React
✓ Select a variant: » JavaScript

Scaffolding project in D:\DevProjects\devops-assignment-1\devops...

Done. Now run:

  cd devops
  yarn
  yarn dev

Done in 14.54s.
```

DEVOPS-ASSIGNMENT-1

devops

> public

> src

🔗 .eslintrc.cjs

🔗 .gitignore

🔗 index.html

📄 package.json

📄 README.md

JS vite.config.js

📄 README.md

sumee@LAPTOP-I8BBG50J MINGW64 /d/DevProjects/devops-assignment-1/devops (main)

\$ git add *

warning: LF will be replaced by CRLF in devops/README.md.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in devops/index.html.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in devops/package.json.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in devops/src/App.css.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in devops/src/App.jsx.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in devops/src/index.css.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in devops/src/main.jsx.

The file will have its original line endings in your working directory

warning: LF will be replaced by CRLF in devops/vite.config.js.

The file will have its original line endings in your working directory

sumee@LAPTOP-I8BBG50J MINGW64 /d/DevProjects/devops-assignment-1/devops (main)

\$ git commit -m "add a sample react project with two directories"

[main 2d2e2d5] add a sample react project with two directories

10 files changed, 211 insertions(+)

create mode 100644 devops/README.md

create mode 100644 devops/index.html

create mode 100644 devops/package.json

create mode 100644 devops/public/vite.svg

create mode 100644 devops/src/App.css

create mode 100644 devops/src/App.jsx

create mode 100644 devops/src/assets/react.svg

create mode 100644 devops/src/index.css

create mode 100644 devops/src/main.jsx

create mode 100644 devops/vite.config.js

sumee@LAPTOP-I8BBG50J MINGW64 /d/DevProjects/devops-assignment-1/devops (main)

\$ git push

info: please complete authentication in your browser...

Enumerating objects: 17, done.

Counting objects: 100% (17/17), done.

Delta compression using up to 12 threads

Compressing objects: 100% (14/14), done.

Writing objects: 100% (16/16), 5.91 KiB | 2.95 MiB/s, done.

Total 16 (delta 0), reused 0 (delta 0), pack-reused 0

To https://github.com/2023mt93091/devops-assignment-1.git

22118c4..2d2e2d5 main -> main

devops-assignment-1 / devops /

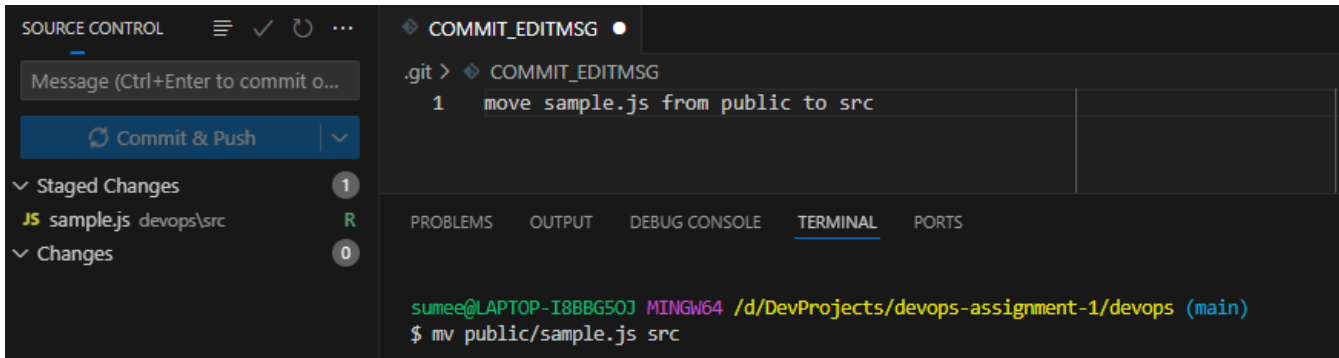
Add file ...

sumeetgodse add a sample react project with two directories

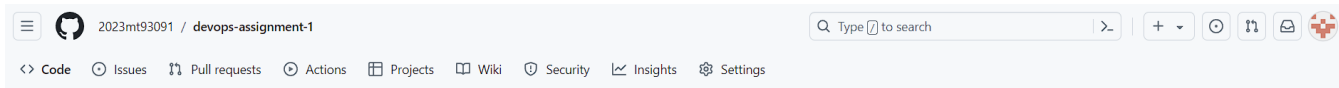
2d2e2d5 · 1 minute ago History

Name	Last commit message	Last commit date
..		
public	add a sample react project with two directories	1 minute ago
src	add a sample react project with two directories	1 minute ago
README.md	add a sample react project with two directories	1 minute ago
index.html	add a sample react project with two directories	1 minute ago
package.json	add a sample react project with two directories	1 minute ago
vite.config.js	add a sample react project with two directories	1 minute ago

3. Move code from one directory to another. I will be moving sample.js from public to src.



The screenshot shows the VS Code interface. On the left, the 'SOURCE CONTROL' panel is open, showing a commit message 'move sample.js from public to src'. The 'TERMINAL' panel at the bottom shows the command `$ mv public/sample.js src` being executed in a terminal window.

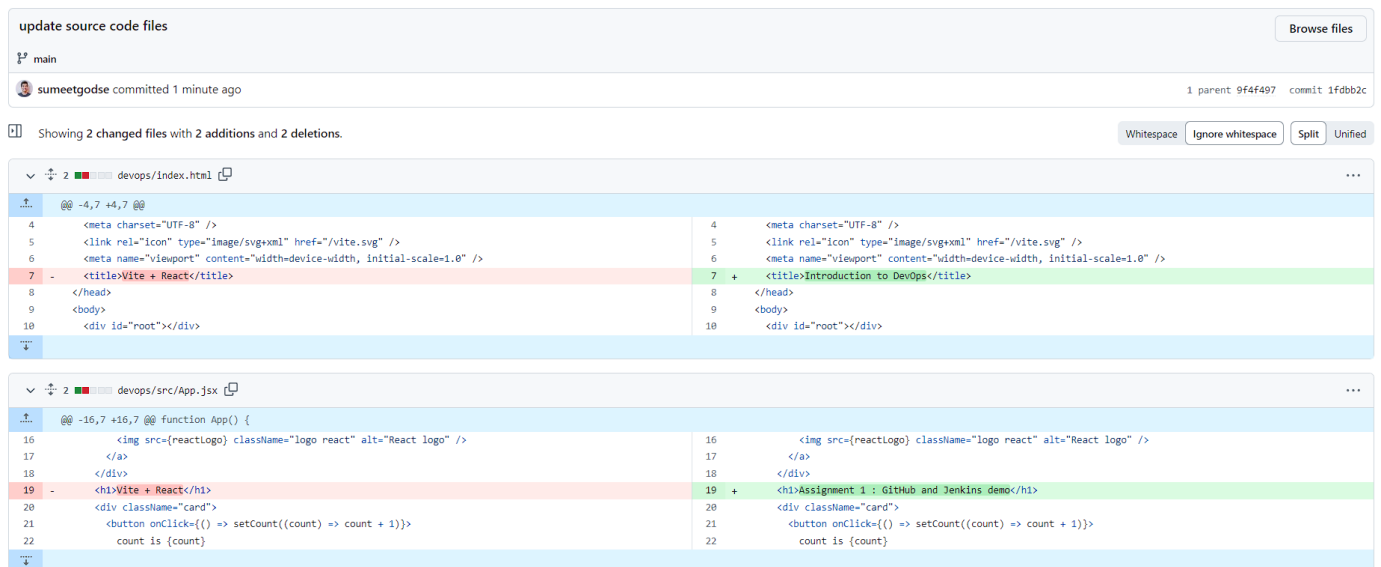


Commit



The screenshot shows the GitHub commit page. The commit message is 'move sample.js from public to src'. The commit is by 'sumeetgodse' and is 1 minute old. The commit shows 1 parent (5599f88) and the commit hash is 9f4f497. The commit shows 1 changed file with 0 additions and 0 deletions. The file is 'devops/public/sample.js' which has been renamed to 'devops/src/sample.js'.

4. Update source code files and display the difference.



The screenshot shows the GitHub commit page for the commit 'update source code files'. The commit is by 'sumeetgodse' and is 1 minute old. The commit shows 1 parent (9f4f497) and the commit hash is 1fdbb2c. The commit shows 2 changed files with 2 additions and 2 deletions. The files are 'devops/Index.html' and 'devops/src/App.jsx'. The differences are shown in a diff view, with changes highlighted in green (additions) and red (deletions).

5. Create a branch

```
sumee@LAPTOP-I8BBG50J MINGW64 /d/DevProjects/devops-assignment-1/devops (main)
$ git checkout -b branch-1
Switched to a new branch 'branch-1'

sumee@LAPTOP-I8BBG50J MINGW64 /d/DevProjects/devops-assignment-1/devops (branch-1)
$ git push -u origin branch-1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branch-1' on GitHub by visiting:
remote:   https://github.com/2023mt93091/devops-assignment-1/pull/new/branch-1
remote:
To https://github.com/2023mt93091/devops-assignment-1.git
 * [new branch]      branch-1 -> branch-1
Branch 'branch-1' set up to track remote branch 'branch-1' from 'origin'.
```



devops-assignment-1

Public

branch-1 2 Branches 0 Tags

Switch branches/tags



Find or create a branch...

Branches

Tags

main

default

✓ branch-1

View all branches

6. Add some raw code to the branch

some raw changes Browse files

branch-1


sumeetgodse committed now
 1 parent 1fdbb2c commit 13c9dbb

Showing 1 changed file with 1 addition and 1 deletion. Whitespace Ignore whitespace Split Unified

devops/src/App.jsx


<pre> 22 count is {count} 23 </button> 24 <p> 25 - Edit <code>src/App.jsx</code> and save to test HMR 26 </p> 27 </div> 28 <p className="read-the-docs"> </pre>	<pre> 22 count is {count} 23 </button> 24 <p> 25 + Add some raw code to the newly created branch : branch-1 26 </p> 27 </div> 28 <p className="read-the-docs"> </pre>
---	---

7. Merge the branch with main line. We can create a pull request to do this.

 **branch-1** had recent pushes 37 seconds ago

Compare & pull request

base: main ← compare: branch-1 ✓ **Able to merge.** These branches can be automatically merged.

 Add a title

some raw changes

Add a description

Write Preview H B I ≡ <> 🔗 ☰ ≡ ≡ ☎ @ ↗ ↶ ↷

some raw changes


Markdown is supported Paste, drop, or click to add files

Create pull request

some raw changes #1

Open 2023mt93091 wants to merge 1 commit into main from branch-1

Conversation 0 Commits 1 Checks 0 Files changed 1


 2023mt93091 commented now

Owner ...

some raw changes

some raw changes 13c9d6b

Add more commits by pushing to the **branch-1** branch on [2023mt93091/devops-assignment-1](#).

 Require approval from specific reviewers before merging

Rulesets ensure specific people approve pull requests before they're merged.

Add rule ×

Continuous integration has not been set up


GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.


✓ This branch has no conflicts with the base branch


Merging can be performed automatically.


Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).


some raw changes #1


 Merged


2023mt93091 merged 1 commit into `main` from `branch-1`  now

 Conversation 0

 Commits 1

 Checks 0


 Files changed 1





2023mt93091 commented 1 minute ago

Owner ...


some raw changes







 some raw changes


13c9d6b




 2023mt93091 merged commit `adca056` into `main` now

Revert






 2023mt93091 deleted the `branch-1` branch now

Restore branch


Merge pull request #1 from 2023mt93091/branch-1 


some raw changes

 `adca056`  <>

 2023mt93091 committed now

some raw changes

 sumeetgodse committed 4 minutes ago

13c9d6b  <>

Advantages of moving from Centralized Source Code to Distributed Version Control.

Migrating from a centralized source code management system (SCM) to a distributed version control system (DVCS) like Git offers several advantages:

- Offline Work:**
With DVCS, developers can work offline since they have a local copy of the entire repository. They can commit changes, create branches, and perform other operations without needing a constant connection to a central server.
- Redundancy and Backup:**
Every developer has a complete copy of the repository, providing redundancy. This means if a central server goes down, developers can continue working and push changes once the server is back up. Additionally, each developer's local repository serves as a backup.
- Flexible Workflows:**
DVCS systems like Git support various workflow models, such as Gitflow, GitHub flow, or GitLab flow. Teams can choose the workflow that best fits their development process.
- Easy Branching and Merging:**
DVCS systems excel at branching and merging. Developers can create branches for features, bug

fixes, or experiments without affecting the main codebase. Merging is also easier and less error-prone compared to centralized systems.

5. **Parallel Development:**

Since each developer works on their own local copy, multiple developers can work in parallel without stepping on each other's toes. This fosters faster development cycles and better collaboration.

6. **Scalability:**

DVCS scales better as the project and team size grow. Since most operations are performed locally, the performance doesn't degrade as the repository size increases or when there are many concurrent users.

7. **Forking and Pull Requests:**

DVCS encourages contribution from external developers through forking and pull requests. Forking allows developers to create their own copy of the repository to work on a feature or fix independently. Pull requests enable developers to propose changes and request them to be merged into the main repository.

8. **Security:**

With DVCS, developers can sign commits, ensuring the authenticity of code changes. Additionally, access control mechanisms can be enforced both locally and on remote repositories, enhancing security.

9. **Decentralization:**

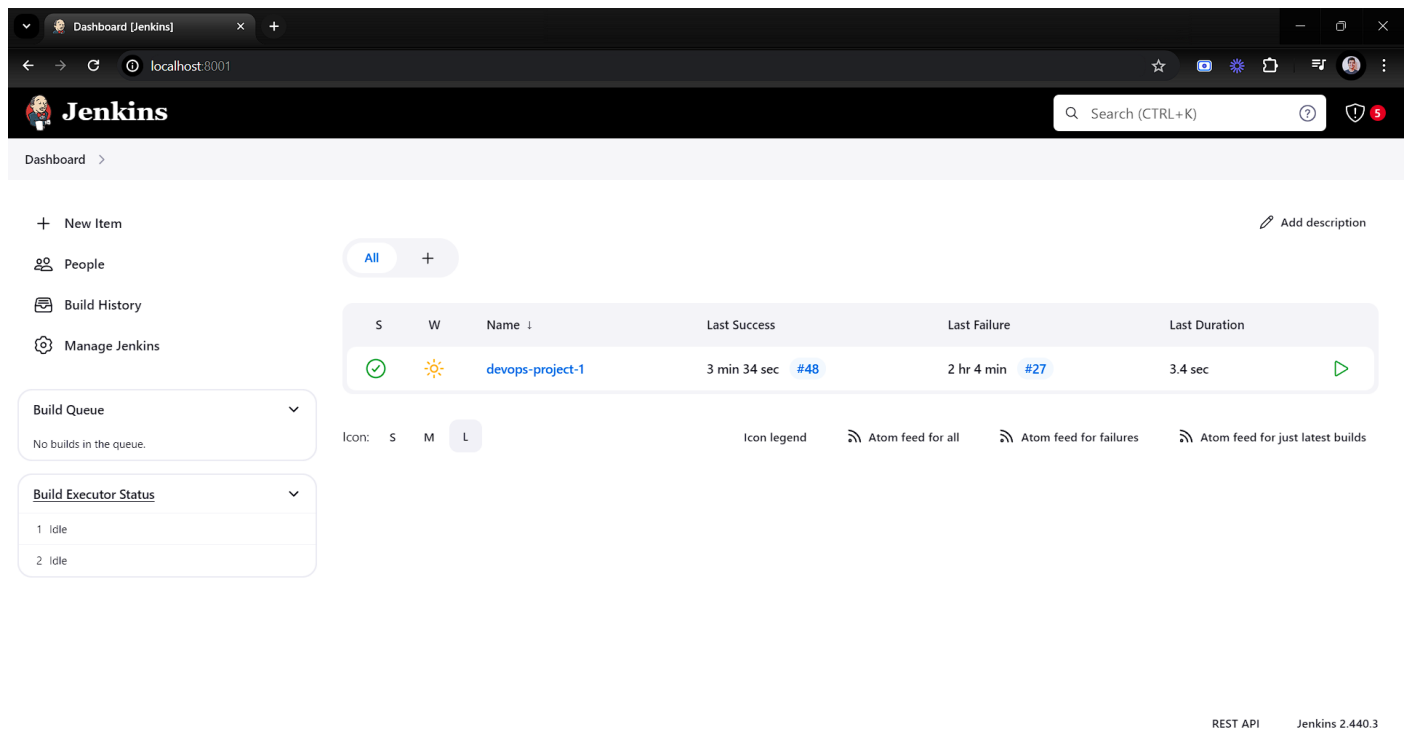
DVCS reduces the reliance on a single central server, distributing the control and responsibility among all developers. This decentralization can lead to increased agility and resilience in large, distributed teams.

Overall, migrating from a centralized source code system to a distributed version control system brings numerous benefits, improving collaboration, productivity, and flexibility in software development workflows.

Part 2 -

Create the Jenkins pipeline, which is checking out the code and build and compiling it on every commit automatically.

1. Install Jenkins on local machine, by referring the documentation. You must also have compatible java installed in your machine. Specify Jenkins port number etc. Once all setup is done Jenkins will run as a service which can be accessed using the services section of control panel. The GUI can be accessed on `http://localhost:{PORT_THAT_YOU_SET}` (I gave 8001 as port number)



The screenshot shows the Jenkins Dashboard in a web browser. The browser address bar shows `localhost:8001`. The Jenkins logo is in the top left, and a search bar is in the top right. The main content area has a sidebar on the left with links: '+ New Item', 'People', 'Build History', and 'Manage Jenkins'. The main area displays a table of builds. The table has columns: 'S' (Success), 'W' (Warning), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. There is one build listed: 'devops-project-1' with a success status (green checkmark), a warning icon (yellow sun), and a duration of '3.4 sec'. Below the table, there are links for 'Icon legend', 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. On the left, there are two expandable sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing '1 Idle' and '2 Idle').

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	devops-project-1	3 min 34 sec #48	2 hr 4 min #27	3.4 sec

Icon: S M L


REST API Jenkins 2.440.3

2. Create a new freestyle project. Do the configuration as below.


Enter an item name

devops-project


» Required field


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

OK

3. Update configuration - Source Code Management. Add the path to your local git repository.

Git ?

Repositories ?

Repository URL ?

file:///D:/DevProjects/devops-assignment-1

Credentials ?

- none -

+ Add ▾

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

**

4. Update configuration - Build Steps. Add the steps for your project compilation and build

Build Steps

Execute Windows batch command ?

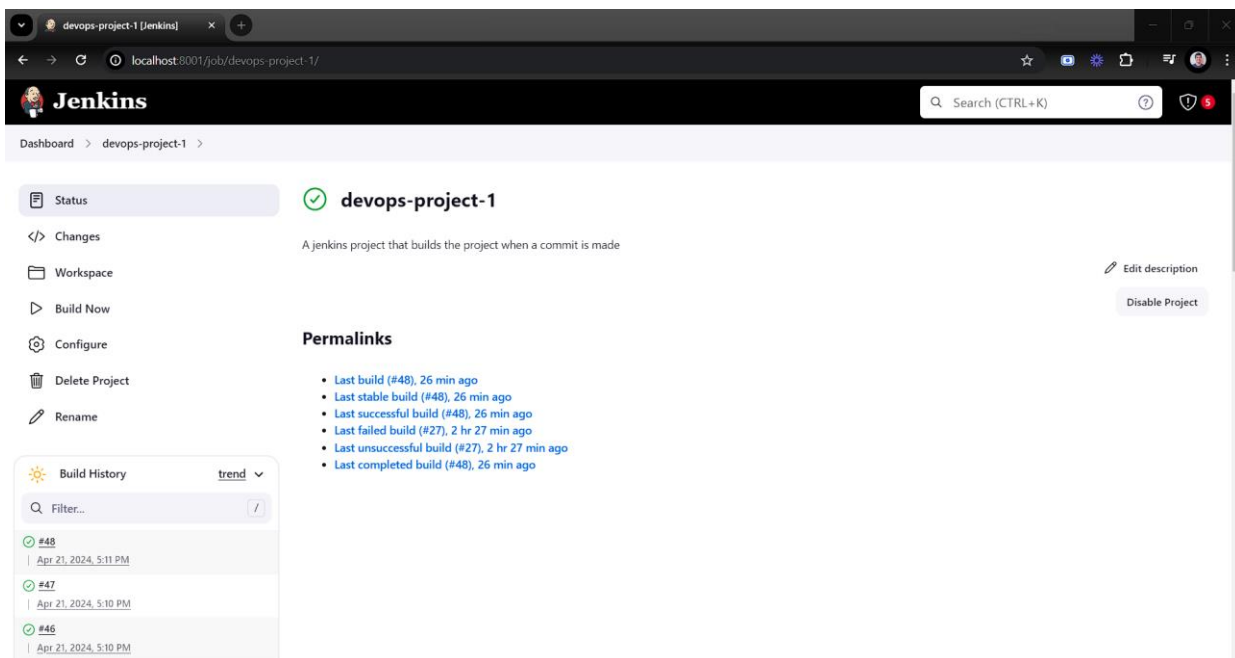
Command

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

D:
cd DevProjects\devops-assignment-1\devops
npm run build

Advanced ▾

5. Keep all other settings as default and hit Save button. You will see a new project created.



6. Now create a new executable file called 'post-commit' inside your projects `.git/hooks` directory. It should contain a curl command that makes a request to your Jenkins project to run a new build.

curl -X POST <http://localhost:8001/job/{project-name}/build>

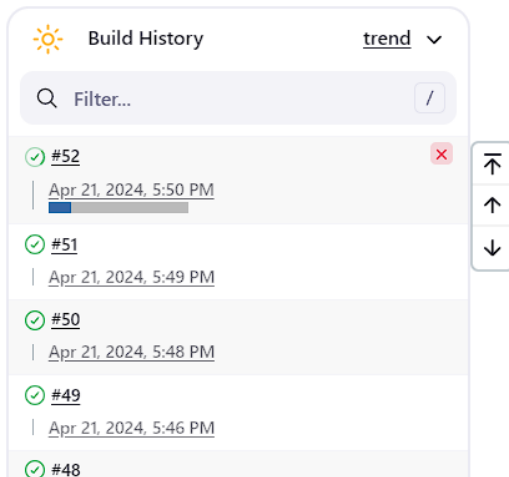
```
sumee@LAPTOP-I8BBG5OJ MINGW64 /d/DevProjects/devops-assignment-1/.git/hooks (GIT_DIR!)
$ ls -lart |grep post-commit
-rwxr-xr-x 1 sumee 197609 73 Apr 21 17:09 post-commit

sumee@LAPTOP-I8BBG5OJ MINGW64 /d/DevProjects/devops-assignment-1/.git/hooks (GIT_DIR!)
$ cat post-commit
#!/bin/sh
curl -X POST http://localhost:8001/job/devops-project-1/build
```

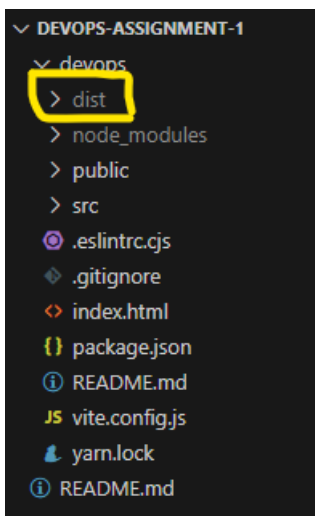
7. Now we are all set to go. Make a new commit in your GitHub repository. A new build will be then triggered in Jenkins.

```
sumee@LAPTOP-I8BBG50J MINGW64 /d/DevProjects/devops-assignment-1/devops (main)
$ git add src/sample.js

sumee@LAPTOP-I8BBG50J MINGW64 /d/DevProjects/devops-assignment-1/devops (main)
$ git commit -m "This commit will trigger a new build"
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
             Dload  Upload   Total     Spent    Left     Speed
0           0          0     0         0      0         0      0      0 --:--:-- --:--:-- --:--:--     0
[main e14bf92] This commit will trigger a new build
1 file changed, 1 insertion(+), 1 deletion(-)
```



8. This will compile and build our application code and create a new folder named 'dist' in our project which can be served to end users.



9. You can click on the build number check the console output.

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#52'

Git Build Data

Previous Build

Console Output

```
Started by user unknown or anonymous
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\devops-project-1
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\devops-project-1\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url file:///D:/DevProjects/devops-assignment-1 # timeout=10
Fetching upstream changes from file:///D:/DevProjects/devops-assignment-1
> git.exe --version # timeout=10
> git --version # 'git version 2.29.0.windows.1'
> git.exe fetch --tags --force --progress -- file:///D:/DevProjects/devops-assignment-1 +refs/heads/*:refs/remotes/origin/* # timeout=10
Seen branch in repository origin/branch-1
Seen branch in repository origin/main
Seen 2 remote branches
> git.exe show-ref --tags -d # timeout=10
Checking out Revision 15cae13ae5f42f3f8bdef315b9d331cc72e5111 (origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 15cae13ae5f42f3f8bdef315b9d331cc72e5111 # timeout=10
Commit message: "This commit will trigger a new build"
> git.exe rev-list --no-walk e204f8c8cfc44a77577a75db89eb841cd6eb0788 # timeout=10
[devops-project-1] $ cmd /c call C:\WINDOWS\TEMP\jenkins4041212600903363643.bat

C:\ProgramData\Jenkins\jenkins\workspace\devops-project-1>D:

D:\>cd DevProjects\devops-assignment-1\devops

D:\DevProjects\devops-assignment-1\devops>npm run build

> devops@0.0.0 build
> vite build
```

Hence, in this way we can leverage Jenkins to checkout the code and do automated build and compilation on every commit.