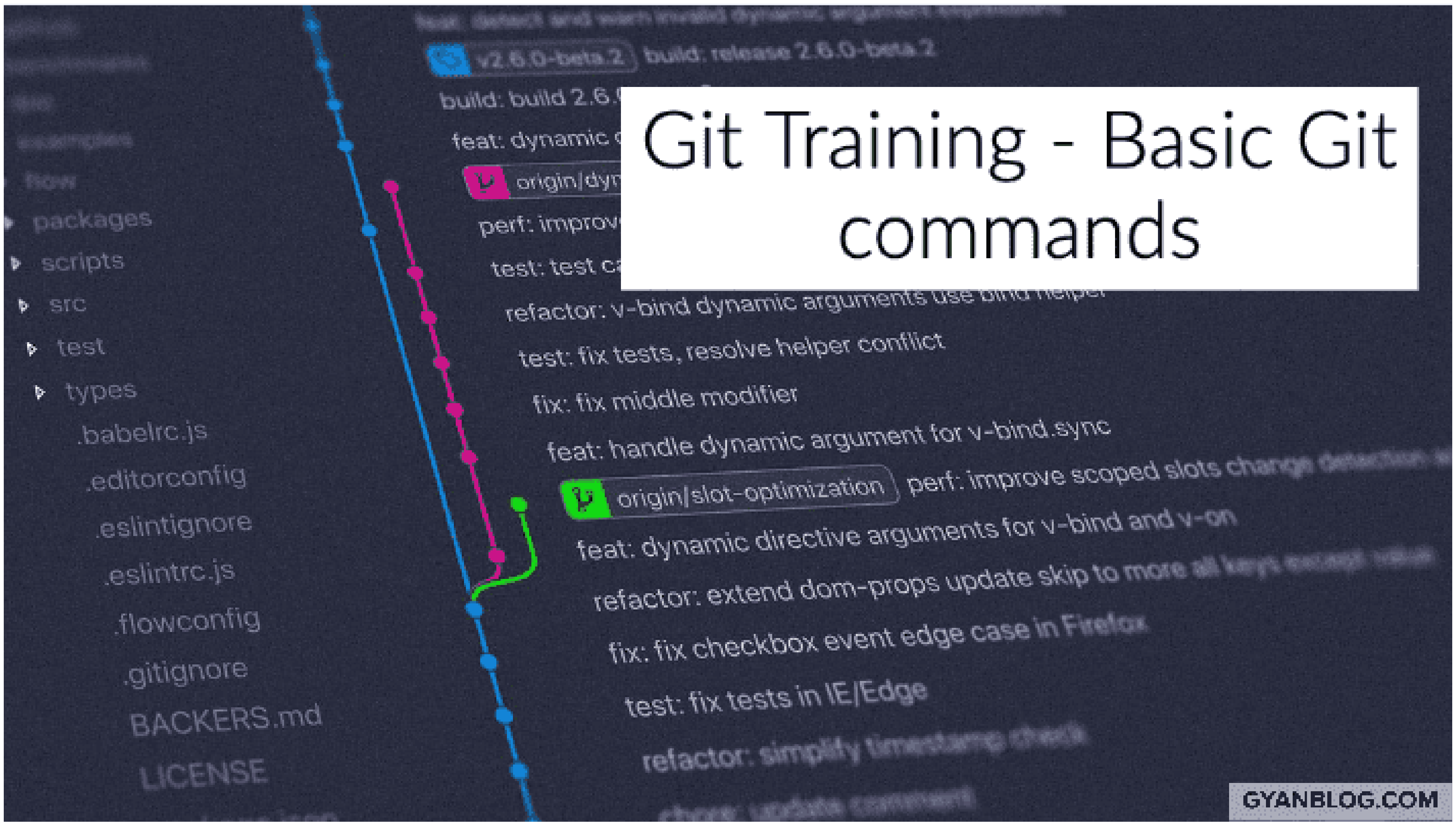
A Practical Guide on how to work with Git Basic Commands and workflows

July 07, 2020



Contents

Introduction

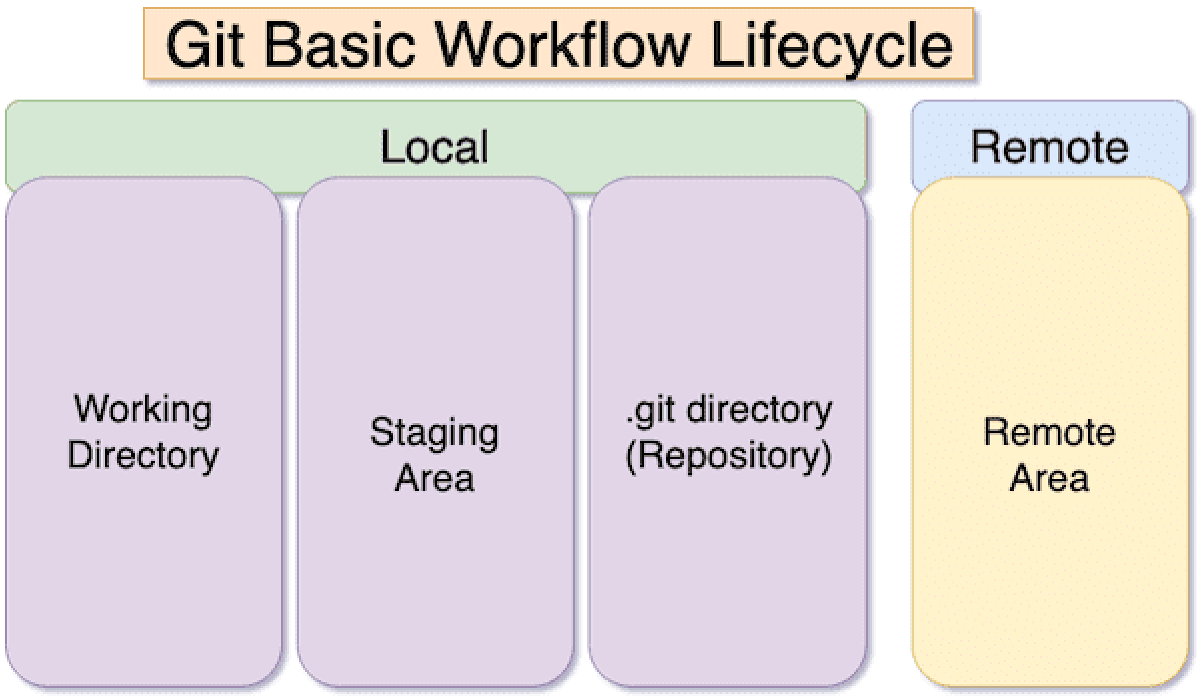
In this guide, we will see git basic commands, and fundamentals of git.

Pre-requisites

1. Have a github project, and setup ssh keys Login to Github.com, and create a test project. Example: `git-test`

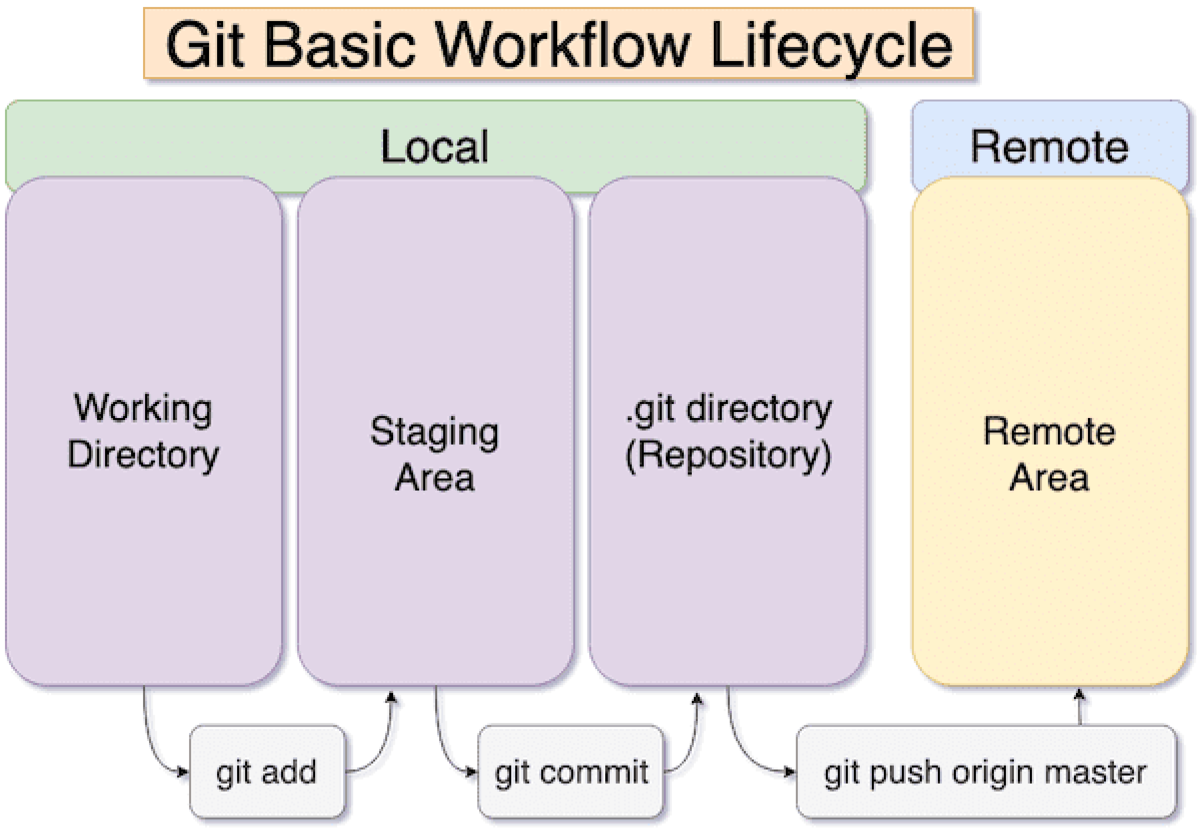
   And, setup your ssh keys from Setup Ssh Keys - Github{:target="_blank"}

2. Install command line git.



There are four different areas in lifecycle of our code in git.

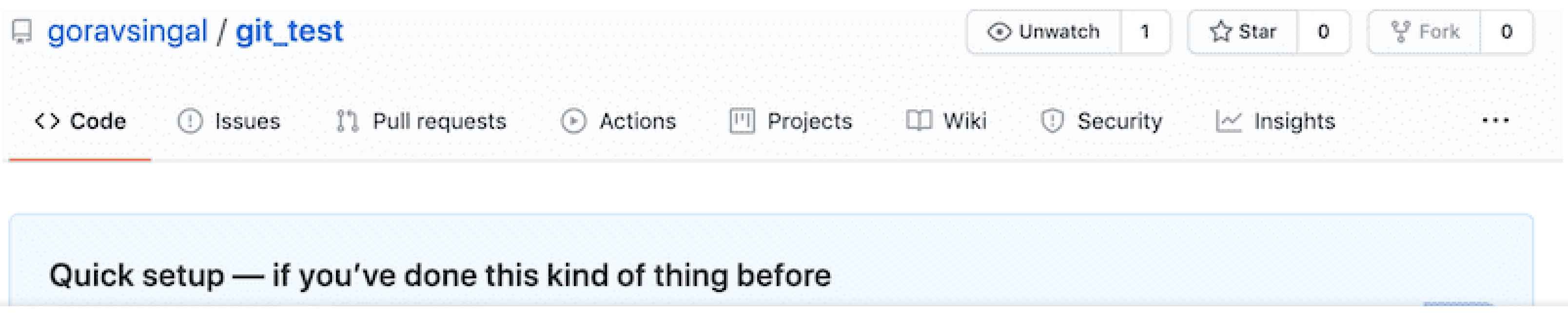
1. Working Directory Its your local copy of code where you can add/modify code.
2. Staging area It is an area where you save your commits, still locally. It is like you are crafting the whole code commits. You can add/remove files into this area. And when ready to commit, you do commit to further area. To add files, you do a `git add <file>`
3. Repository (.git folder) Its a stage just before going live. When you promote files from staging area, files comes here. You run command: `git commit` to move files here.
4. Remote (Actual git repository) Its the final destination of files, when they resides in your git code base. Where anyone who has access to your code can take from.



Workflow-1, Get the code from Github

(Or, to clone the code)

To get the code or project from Github, goto your project link in Github. Copy the git path, as shown in image:



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

Open terminal, run command:

```
git clone https://github.com/goravsingal/git_test.git
```

It should create a folder with your project name in the current folder. Lets go inside that folder.

Git ls-files

It lists all files tracked by git.

Workflow-2, Git status

Command `git status` tells you the status of your files. It tells in which branch I'm currently. It also tells which are the files which are unstaged.

```
$ git status
```

Workflow-3, Add a file

It involves three steps:

1. Add to stage area (`git add`)
2. Move to repository folder (`git commit`)
3. Move to remote (`git push`)

Add a file to staging area

Lets create a test file in the project folder. And, see its status.

```
$ echo "hi" > test1.txt

# Run git status
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test1.txt

nothing added to commit but untracked files present (use "git add" to track)
```

The files you just added will be treated as Untracked files. To add this file to staging area, run command:

```
$ git add test1.txt

# Run git status
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test1.txt
```

Add file to repository area (Just before going live)

Now our file is in staging area. Lets promote it to next level by committing it.

```
$ git commit -m "Commit message"

# Run git status
$ git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean
```

You should specify very nice commit message. This will help you and your team in future about why this commit happened.

Push file to go live

Push this file to master branch.

```
$ git push origin master

# run git status
$ git status

On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

You can verify this file by going to your github url in browser.

Example adding two commits in a single push

Lets add one file first in first commit:

```
# Adds one file
$ echo "hi" > t1
$ git add t1
$ git commit -m "t1"

# Adds second file
$ echo "hi" > t2
$ git add t2
$ git commit -m "t2"

$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

# note the "2 commits"

# push changes
$ git push origin master
```

In above example, we added two files in two separate commits. And, did a single push for both commits. If you look at commit history in your github url, you will find these two separate commits.

Revert file from Staging area

Consider you have created a new file or modified existing file. And, used `git add` to add the file to staging area. But, later you want that file to be removed from staging area. The important point is that we want file to be removed from staging area only, and the chanegs will be intact in the local file.

```
# run git status after adding file name t3
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   t3
```

Note, git status shows you the command to revert from stage. There are actually two ways to do this:

```
# 1
git reset HEAD <file>

# 2
git restore --staged <file>
```

Both does the same job. Your local file will still has the changes.

Revert local file changes and make it similar to what is live
You have done some changes to files. And, now want to revert that changes so that it become exactly same as what is at github.

```
$ git checkout -- <file>

# in our example, if we modified file named: t2
$ git checkout -- t2
```

Workflow-4, Rename a file

Method-1

use git command to rename file. (Not through system command)

```
git mv <old filename> <new filename>

# In my example, rename test2.txt to test3.txt
$ git mv test2.txt test3.txt

$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        renamed:    test2.txt -> test3.txt
```

You don’t need to rename file on disk. Git command will do it for you.

Method-2

Rename it via system command.

```
$ mv test2.txt test3.txt
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
        deleted:    test2.txt

Untracked files:
        test3.txt
```

In this case, git sees this as two separate operations. Delete and add the file. Lets see the magical command.

```
$ git status
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        renamed:    test2.txt -> test3.txt
```

Now, you can commit the changes by `git commit, git push` commands.

Reverting file rename

There is a simpler way to do this apart from `git reset`. Just do:

```
$ git mv test3.txt test2.txt
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

Workflow-5, Deleting file

delete a file using `git rm`

```
$ git rm <Filename>

# example
$ git rm t2
rm 't2'

$ git status
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        deleted:    t2
```

If you do an ls, you will not find that file in local folder. In `git status`, you will see that git has staged this change. But the change has not been committed and pushed yet.

Delete file by system command

If we delete file by system command, `rm <filename` Run a git status

```
$ git status

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    test2.txt
```

It detects the change, that this is not staged. To add this in stage area:

```
# Below command will add change for any file added/removed/updated
$ git add -A
```

How to revert deleted file from stage area back to local area

```
git reset HEAD <filename>

$ git reset HEAD t2

Unstaged changes after reset:
D       t2
```

Note, you need to type the filename correctly. Since, it is not in your local folder. And, even after running above command, if you see the file is not in your folder yet.

```
$ git status

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    t2

no changes added to commit (use "git add" and/or "git commit -a")
```

Finally, to bring it back.

```
$ git checkout -- t2

$ git status

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

Workflow-6, Ignore files or folders to commit

There can be several files that you do not want to commit in git. Commands like `git add -A` recursively add all the files in your folder. For example, there is a credential files placed in your folder that you do not want to commit. You might accidently commit it.

So, to save from these scenarios. There is a special file called `.gitignore`

Create this file under your project folder, and you can use file/folder name, pattern. Once you save this file, git will never commit those matched files and folders.

```
# example file .gitignore
```

It is important to note that you need to commit this file in git.

Please read continuous posts:

- Git log Command
- Creating Git command alias
- Understanding Git diff
- Git Best Practices

ALSO ON GYANBLOG

How to sync Mongodb data to …
4 years ago · 11 comments
Introduction This post is about syncing your mongodo database data …

Drupal 8 - How to show view user data to …
3 years ago · 2 comments
Introduction to Problem I have some data related to users. For example: …

How to connect to a running mysql …
3 years ago · 2 comments
Introduction I have a host running mysql (not on a container). I have to run …

Jquery validate submitHandler not …
3 years ago · 1 comment
Code that I have is: It was: I changed it to: So, I needed to change button type …

Drupal 8 - Adv usage of …
3 years ago · 1 cor
Introduction In m article, I explaine have set of fields

0 Comments Login ▾

G Start the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS ?

Name

Share Best Newest Oldest

Be the first to comment.

Subscribe Privacy Do Not Sell My Data

Similar Posts

How to solve java issue - could not load main class - Visual Studio Code

Assuming you have a java project and is using Visual Studio Code as IDE. All of…

Static Website Hosting with AWS S3 and Cloudflare

Static websites have several advantages over dyanamic websites. If you are…

[How to renew SSL certificate from Lets-encrypt when your website is using cloudflare]

[Drupal&#58; How to block a user by email programatically]

Many times, while administering your drupal website, you must have encountered...

[Mac showing strange incorrect month name]

Introduction to problem So, on my mac, I'ev set timezone to my local city i.e...

[How to trim Meta Description tag value generated by Metatag module, to max 160 characters]

I was using On page optimization of the article pages, and found that meta...

Latest Posts

Jenkins Pipeline with Jenkinsfile - How To Schedule Job on Cron and Not on Code Commit

Introduction In this post we will see following: How to schedule a job on cron...

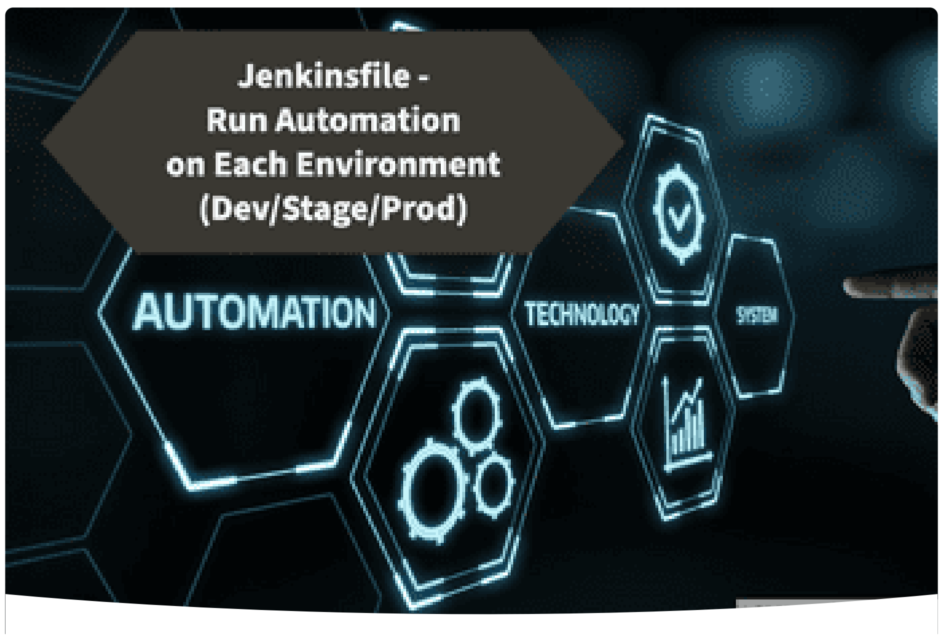
How to Git Clone Another Repository from Jenkin Pipeline in Jenkinsfile

Introduction There are some cases, where I need another git repository while...

Jenkinsfile - Create UI Widgets and Access their Values, Run Automation
AUTOMATION

How to Fetch Multiple Credentials and Expose them in Environment using Jenkinsfile pipeline

Introduction In this post, we will see how to fetch multiple credentials and...


Jenkinsfile - Run Automation on Each Environment (Dev/Stage/Prod)
AUTOMATION
TECHNOLOGY
SYSTEM

Jenkins Pipeline - How to run Automation on Different Environment (Dev/Stage/Prod), with Credentials

Introduction I have an automation script, that I want to run on different...

Jenkinsfile - Create UI Widgets and Access their Values, Run Automation

Jenkinsfile - How to Create UI Form Text fields, Drop-down and Run for Different Conditions

Introduction I had to write a CICD system for one of our project. I had to...

Java Log4j Logger - Programmatically Initialize JSON logger with customized keys in json logs

Introduction Java log4j has many ways to initialize and append the desired...