EXPERIMENT 1:

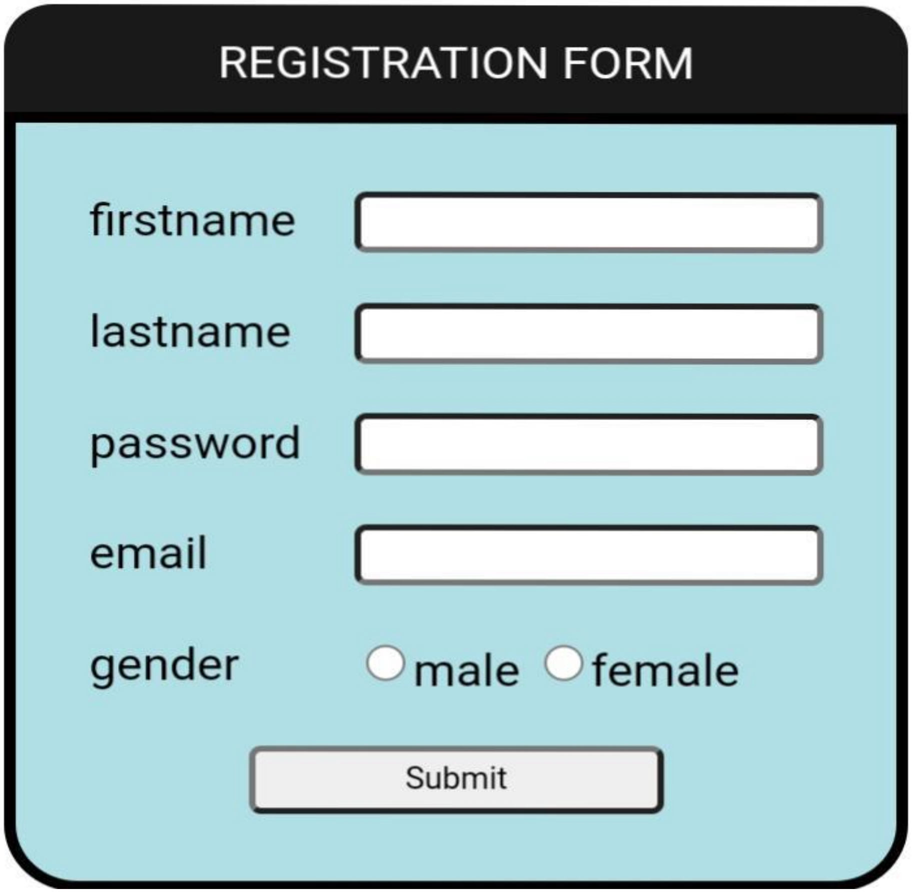
AIM : To write a code for a simple user registration form for an event.

PROGRAM :

```
<!DOCTYPE htm>
 <html>
 <head>
<title> registration </title>
 <style>
 table{margin:100px auto; border:5px solid black; border-bottom-right-radius:30px; border-bottom-left-radius:30px; padding:5%; background-color:powderblue; }
td{padding:10px;}
input{border-radius:5px;border-width:3px; height:20px;border-color:5px solid black;}
input:hover{background-color:#ff1f00;opacity:1; color:white;font-weight:bold; background-clip:border-box;}
caption{border:5px solid black; border-bottom:none; border-top-left-radius:20px; border-top-right-radius:20px; padding:10px; background-color:black; color:white; opacity:0.9;}
</style> </head>
<body style="font-size:20px;">
```

```
 <table>
<caption>REGISTRATION FORM</caption>
<tr> <td id="1st">firstname</td> <td><input type="text"></td> </tr>
<tr> <td>lastname</td> <td><input type="text"></td> </tr>
<tr> <td>password</td> <td><input type="password"></td> </tr>
<tr> <td>email</td> <td><input type="email"></td> </tr>
 <tr> <td>gender</td> <td> <input id="opt2" type="radio" name="1"
><label for="opt2">male</label>
<input id="op3" type="radio" name="1"><label
for="op3">female</label></td>
</tr>
<tr>
<td colspan="2" style="text-align:center;"><input type="submit"
style="width:170px;height:30px;">
</td>
</tr>
</table>
</body>
</html>
```

OUTPUT :



REGISTRATION FORM

firstname

lastname

password

email

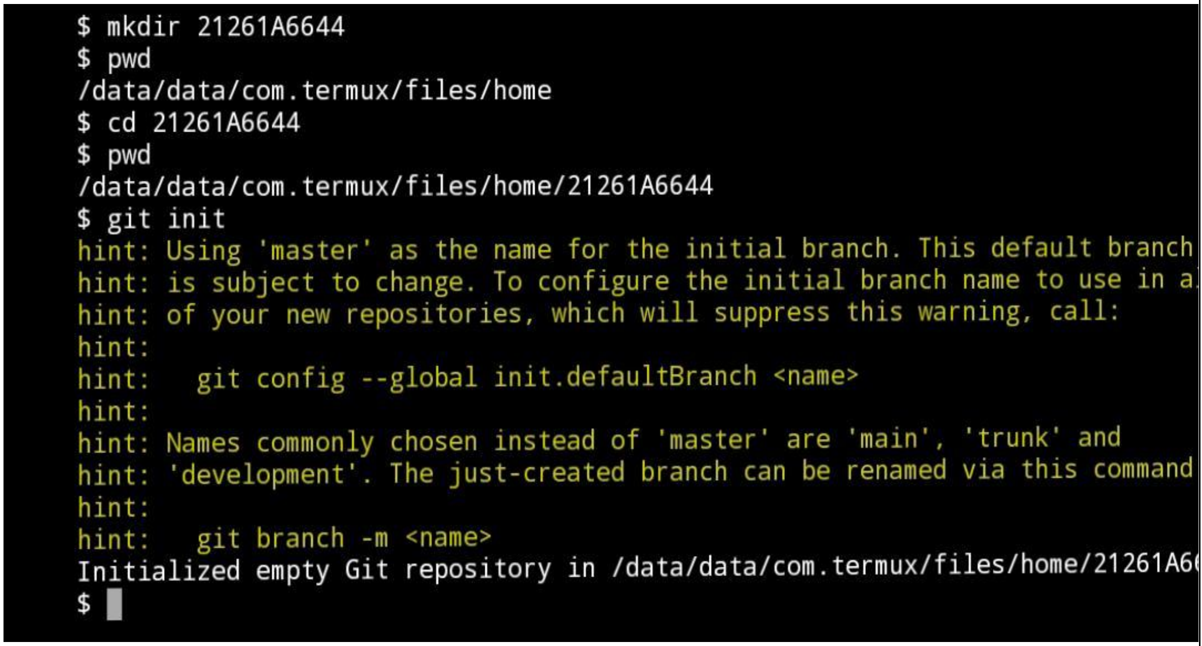
gender ○male ○female

Submit

EXPERIMENT 2:

AIM: Exploring Git and Github Commands

- Specify the global parameters in git.
- Now in our local repository we can initialize git by the command “git init”.



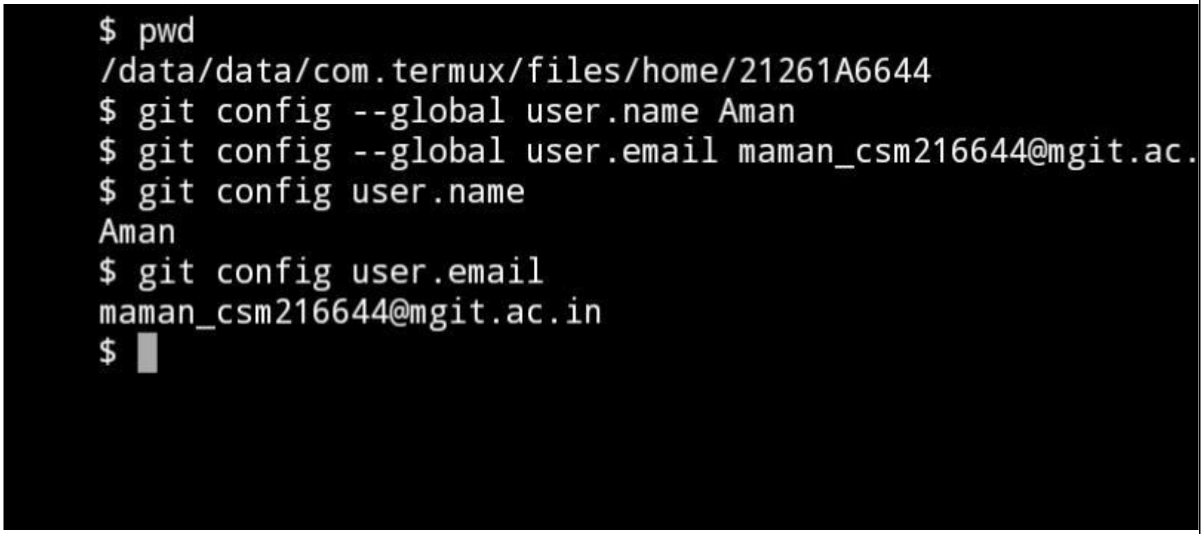
```
$ mkdir 21261A6644
$ pwd
/data/data/com.termux/files/home
$ cd 21261A6644
$ pwd
/data/data/com.termux/files/home/21261A6644
$ git init
hint: Using 'master' as the name for the initial branch. This default branch
hint: is subject to change. To configure the initial branch name to use in a
hint: of 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 /data/data/com.termux/files/home/21261A6
$ 
```

•After saving your code, type the command “git status” what this essentially does is, it checkswhether our files are there on the staging area or not

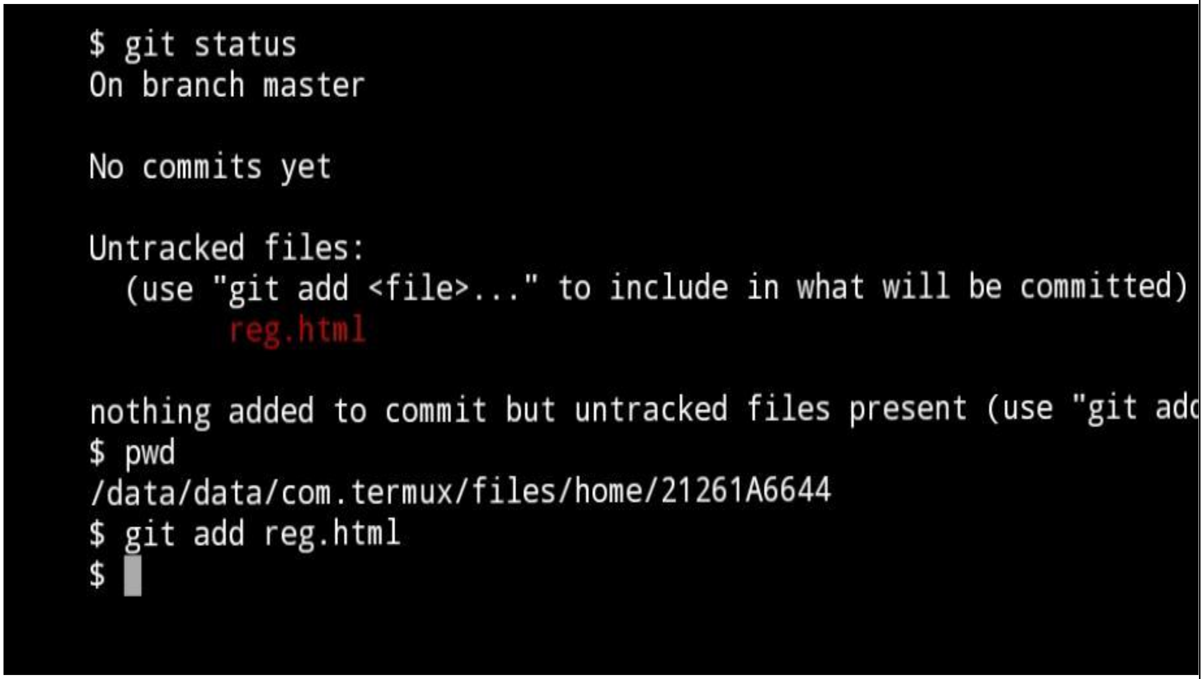
and if further any modifications are done to the file, then also we have to use this.

- Now do the “git add &lt;file-name&gt;” or “git add .” command for adding

the files to the stagingarea. fter that check your status!



```
$ pwd
/data/data/com.termux/files/home/21261A6644
$ git config --global user.name Aman
$ git config --global user.email maman_csm216644@mgit.ac.
$ git config user.name
Aman
$ git config user.email
maman_csm216644@mgit.ac.in
$ █
```



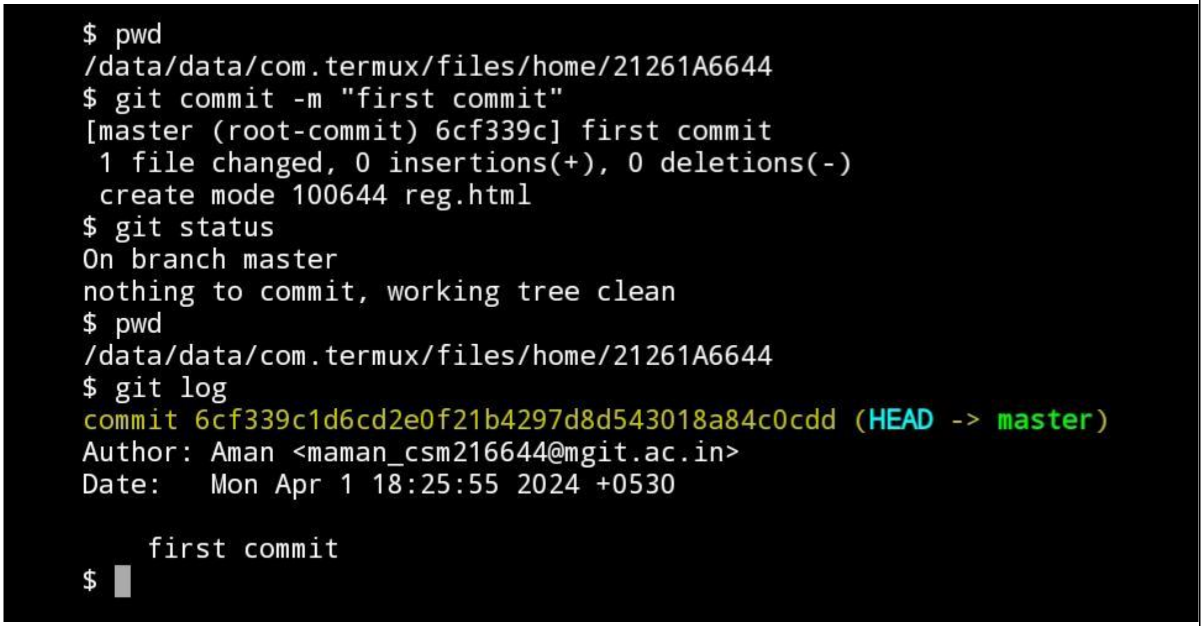
```
$ git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add
$ pwd
/data/data/com.termux/files/home/21261A6644
$ git add reg.html
$ █
```

- Our files are currently in the staging area!
- To commit the changes made we use the command git commit -m “Your message”



```
$ pwd
/data/data/com.termux/files/home/21261A6644
$ git commit -m "first commit"
[master (root-commit) 6cf339c] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 reg.html
$ git status
On branch master
nothing to commit, working tree clean
$ pwd
/data/data/com.termux/files/home/21261A6644
$ git log
commit 6cf339c1d6cd2e0f21b4297d8d543018a84c0cdd (HEAD -> master)
Author: Aman <maman_csm216644@mgit.ac.in>
Date:   Mon Apr 1 18:25:55 2024 +0530

    first commit
$ 
```

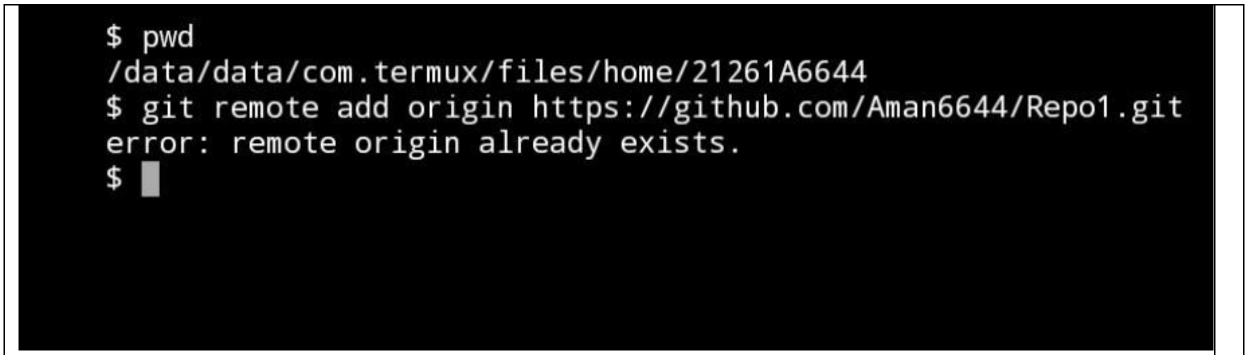
To push these files to our Github repository, execute the following commands

Step-1: Copy your remote repository's URL from GitHub, Mine is https://github.com/bharath205/devops-lab.git

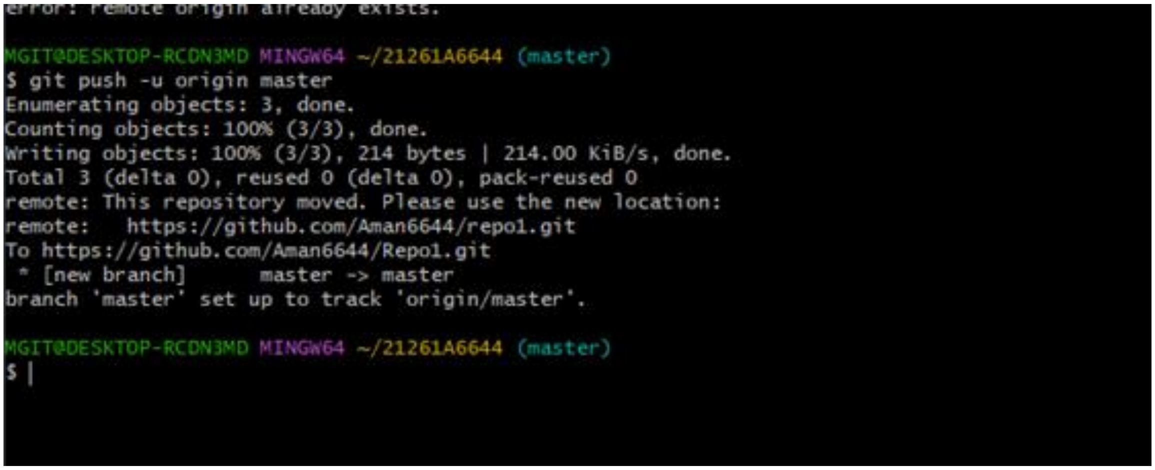
git remote add origin https://github.com/bharath205/devops-lab.git

origin just specifies your repo’s address.

Step-2: Next task is to push it into Github Command is - git push origin master Enter your login credentials the same as your github.



```
$ pwd
/data/data/com.termux/files/home/21261A6644
$ git remote add origin https://github.com/Aman6644/Repo1.git
error: remote origin already exists.
$ 
```

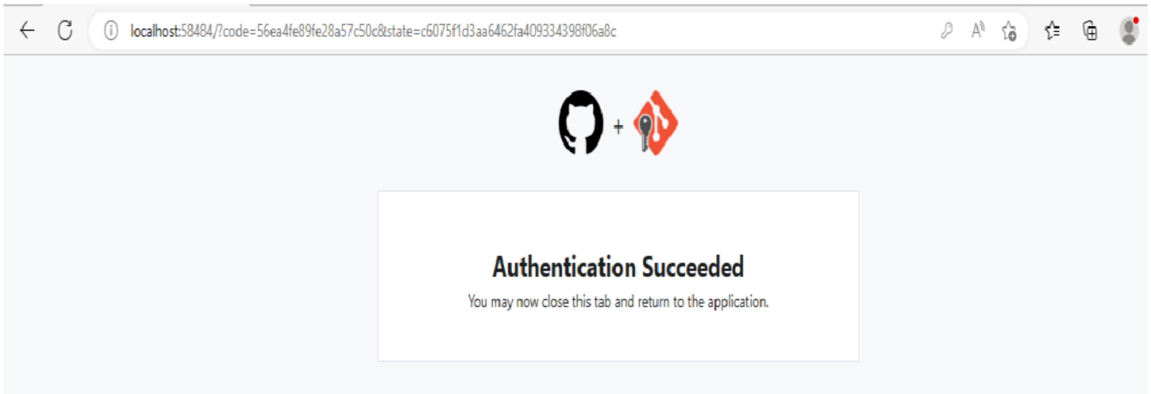


```
error: remote origin already exists.

MGIT@DESKTOP-RCDN3MD MINGW64 ~/21261A6644 (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 214 bytes | 214.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: This repository moved. Please use the new location:
remote:   https://github.com/Aman6644/repo1.git
To https://github.com/Aman6644/Repo1.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

MGIT@DESKTOP-RCDN3MD MINGW64 ~/21261A6644 (master)
$ |
```

- Finally verify the reflected changes in your github repository!

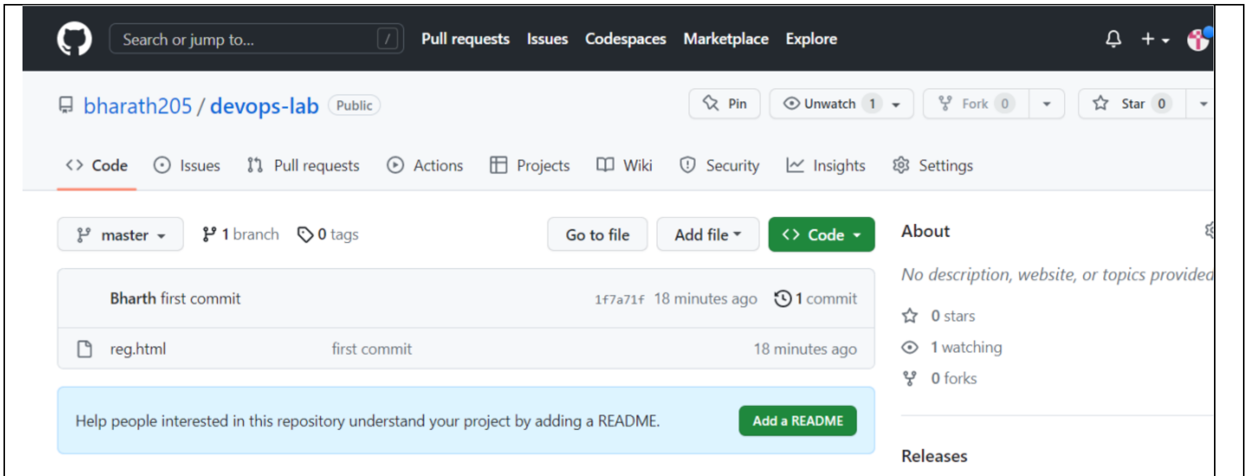


localhost:58484/?code=56ea4fe89fe28a57c50c&state=c6075f1d3aa6462fa409334398f06a8c

+

Authentication Succeeded

You may now close this tab and return to the application.



bharath205 / devops-lab Public

Bharth first commit 1f7a71f 18 minutes ago 1 commit

reg.html first commit 18 minutes ago

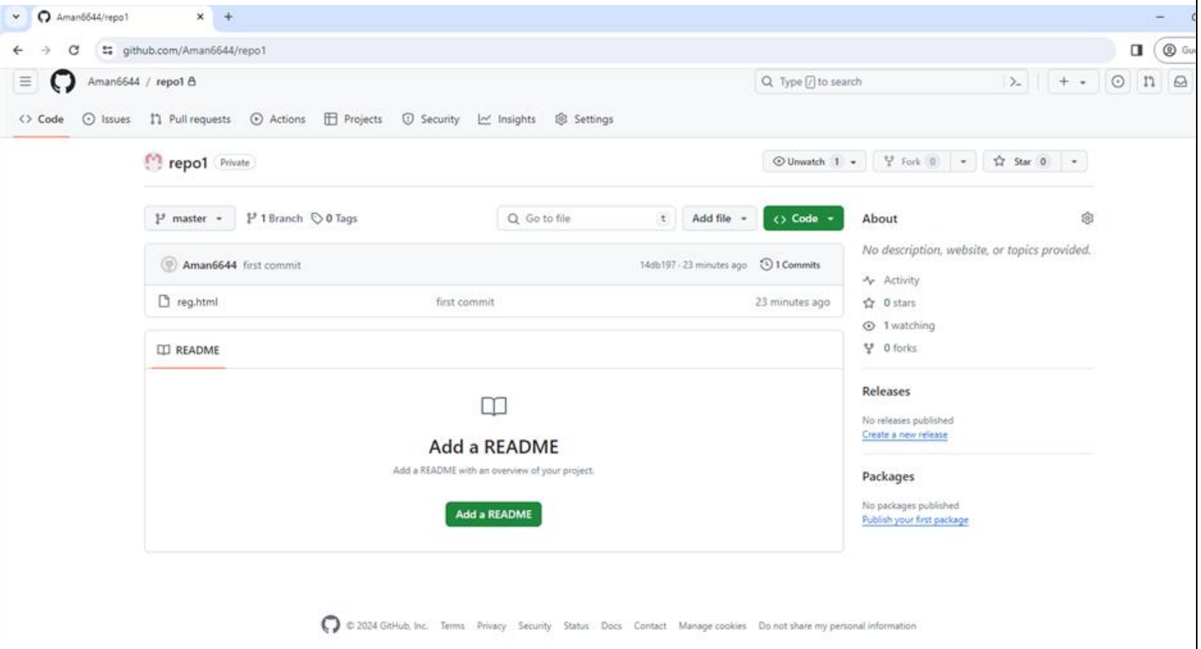
Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided

0 stars
1 watching
0 forks

Releases



Aman6644/repo1

github.com/Aman6644/repo1

Aman6644 / repo1

repo1 Private

Aman6644 first commit 14db197 · 23 minutes ago 1 Commits

reg.html first commit 23 minutes ago

Add a README

Add a README with an overview of your project.

About

No description, website, or topics provided.

Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

EXPERIMENT 3:

AIM : Practice Source code management on GitHub . Experiment with the source code written in exercise 1

Description:
To practice source code management on GitHub, you can follow these steps:

1. Creating a new repository

You need to create a new repository on GitHub .Click on the plus sign.

Fill up all the required details, i.e., repository name, description and also make the repository public this time as it is free.

2. Open your Git Bash

- Git Bash can be downloaded in **here**, and it is a shell used to interface with the operating system which follows the UNIX command.

3. Create your local project in your desktop directed towards a current working directory

`pwd` stands for 'print working directory', which is used to print the current directory.

Move to the specific path in your local computer by cd `'path_name'`. The cd

commands stand for 'change directory' and it is used to change to the working directory in your operating system, and to locate your file, `'path_name'` `i.e. E:\Devlab`

4. Initialize the git repository

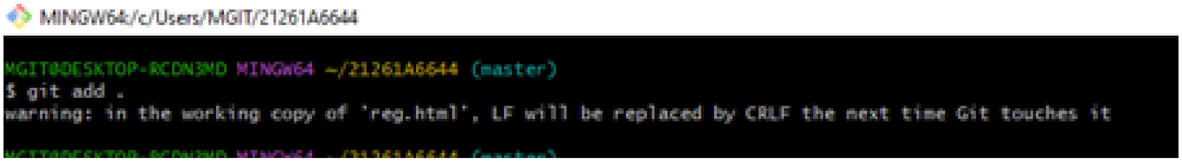
Run `git init` to initialize the repository.

It is used to create a new empty repository or directory consisting of files' with the hidden directory. `'.git'` is created at the top level of your project, which places all of the revision information in one place.

5. Add the file to the new local repository

Use `git add .` in your bash to add all the files to the given folder.

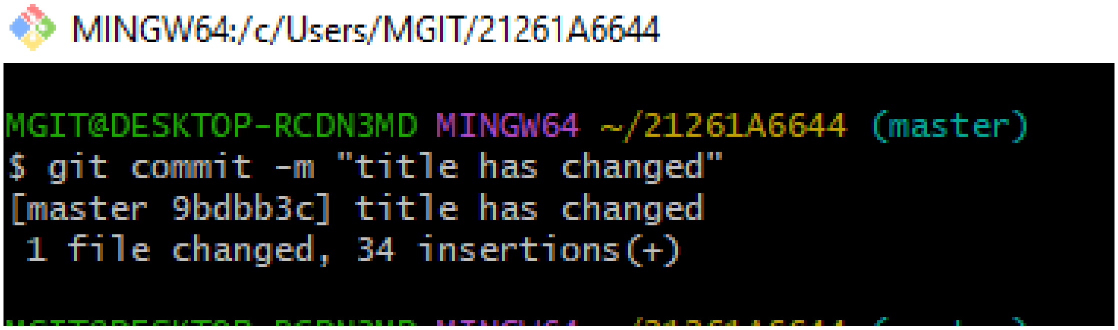
Use `git status` in your bash to view all the files which are going to be staged to the first commit.



```
MGIT@DESKTOP-RCDN3MD MINGW64 ~/21261A6644 (master)
$ git add .
warning: in the working copy of 'reg.html', LF will be replaced by CRLF the next time Git touches it
```

6. Commit the files staged in your local repository by writing a commit message

You can create a commit message by `git commit -m 'your message'`, which adds the change to the local repository.



```
MGIT@DESKTOP-RCDN3MD MINGW64 ~/21261A6644 (master)
$ git commit -m "title has changed"
[master 9bdbb3c] title has changed
 1 file changed, 34 insertions(+)
```

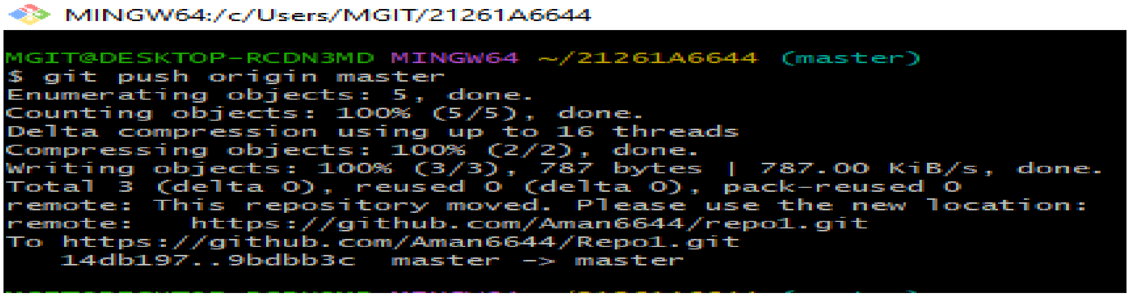
7. Copy your remote repository's URL from GitHub

The HTTPS or URL is copied from the given GitHub account, which is the place of the remote repository.

8. Add the URL copied, which is your remote repository to where your local content from your repository is pushed

```
git remote add origin 'your_url_name'
```

if previously you have added url of another repository and now you want to change the repo URL then firstly delete the previous repo origin by using the command



MINGW64:/c/Users/MGIT/21261A6644

```
MGIT@DESKTOP-RCDN3MD MINGW64 ~/21261A6644 (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 787 bytes | 787.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: This repository moved. Please use the new location:
remote:   https://github.com/Aman6644/repo1.git
To https://github.com/Aman6644/Repo1.git
   14db197..9bdbb3c  master -> master
```

git remote remove origin

9. Push the code in your local repository to GitHub

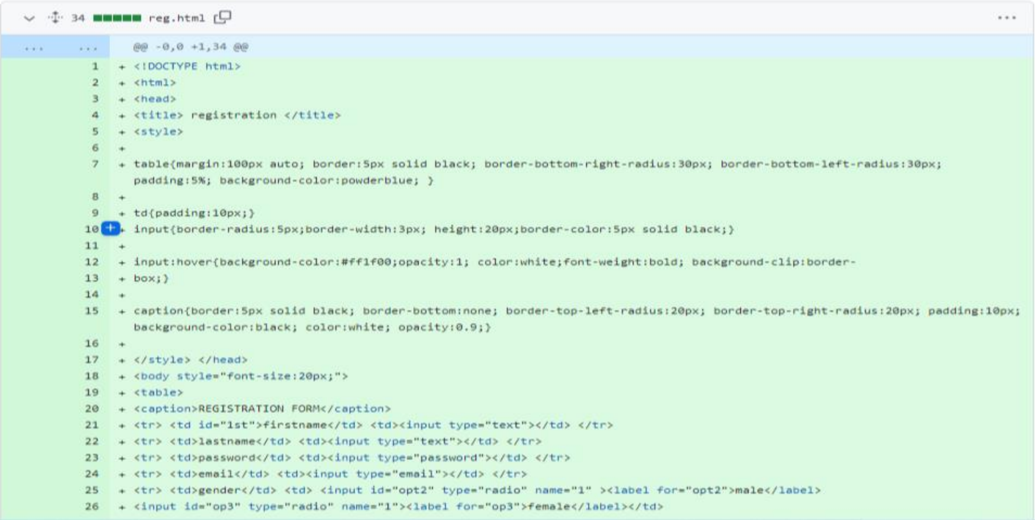
`git push -u origin master` is used for pushing local content to GitHub.

In the code, the origin is your default remote repository name and '-u' flag is upstream, which is equivalent to '-set-upstream.' and the master is the branch, name.upstream is the repository that we have cloned the project.

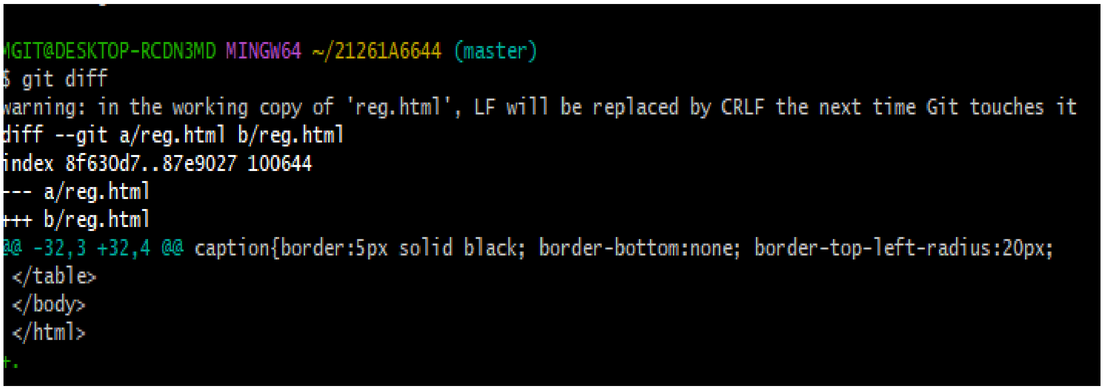
These steps demonstrate how to use GitHub for source code management. You can use the same steps to manage any source code projects on GitHub.

Additionally, you can also explore GitHub features such as pull requests, code review, and branch management to enhance your source code management workflow

OUTPUT :



```
@@ -0,0 +1,34 @@
+ <!DOCTYPE html>
+ <html>
+ <head>
+ <title> registration </title>
+ <style>
+
+ table{margin:100px auto; border:5px solid black; border-bottom-right-radius:30px; border-bottom-left-radius:30px;
  padding:5%; background-color:powderblue; }
+
+ td{padding:10px;}
+ input{border-radius:5px;border-width:3px; height:20px;border-color:5px solid black;}
+
+ input:hover{background-color:#ff1f00;opacity:1; color:white;font-weight:bold; background-clip:border-
+ box;}
+
+ caption{border:5px solid black; border-bottom:none; border-top-left-radius:20px; border-top-right-radius:20px; padding:10px;
  background-color:black; color:white; opacity:0.9;}
+
+ </style> </head>
+ <body style="font-size:20px;">
+ <table>
+ <caption>REGISTRATION FORM</caption>
+ <tr> <td id="1st">firstname</td> <td><input type="text"></td> </tr>
+ <tr> <td>lastname</td> <td><input type="text"></td> </tr>
+ <tr> <td>password</td> <td><input type="password"></td> </tr>
+ <tr> <td>email</td> <td><input type="email"></td> </tr>
+ <tr> <td>gender</td> <td> <input id="opt2" type="radio" name="1" ><label for="opt2">male</label>
+ <input id="op3" type="radio" name="1"><label for="op3">female</label></td>
```



```
MGIT@DESKTOP-RCDN3MD MINGW64 ~/21261A6644 (master)
$ git diff
warning: in the working copy of 'reg.html', LF will be replaced by CRLF the next time Git touches it
diff --git a/reg.html b/reg.html
index 8f630d7..87e9027 100644
--- a/reg.html
+++ b/reg.html
@@ -32,3 +32,4 @@ caption{border:5px solid black; border-bottom:none; border-top-left-radius:20px;
 </table>
 </body>
 </html>
+.
```

EXPERIMENT NO: 4

AIM : Jenkins installation and setup, explore the environment

DESCRIPTION :

Jenkins is a popular open-source tool for Continuous Integration and Continuous Deployment (CI/CD) in software development. Here are the steps to install and set up Jenkins:

Download and install Jenkins:

- Download the Jenkins package for your operating system from the Jenkins website.
- Follow the installation instructions for your operating system to install Jenkins.

Unlock Jenkins

Start the Jenkins service:

- On Windows, use the Windows Services Manager to start the Jenkins service.
- On Linux, use the following command to start the Jenkins service:

```
$ sudo service jenkins start
```

Access the Jenkins web interface:

- Open a web browser and navigate to http://localhost:8080 to access the Jenkins web interface.
- If the Jenkins service is running, you will see the Jenkins login page.

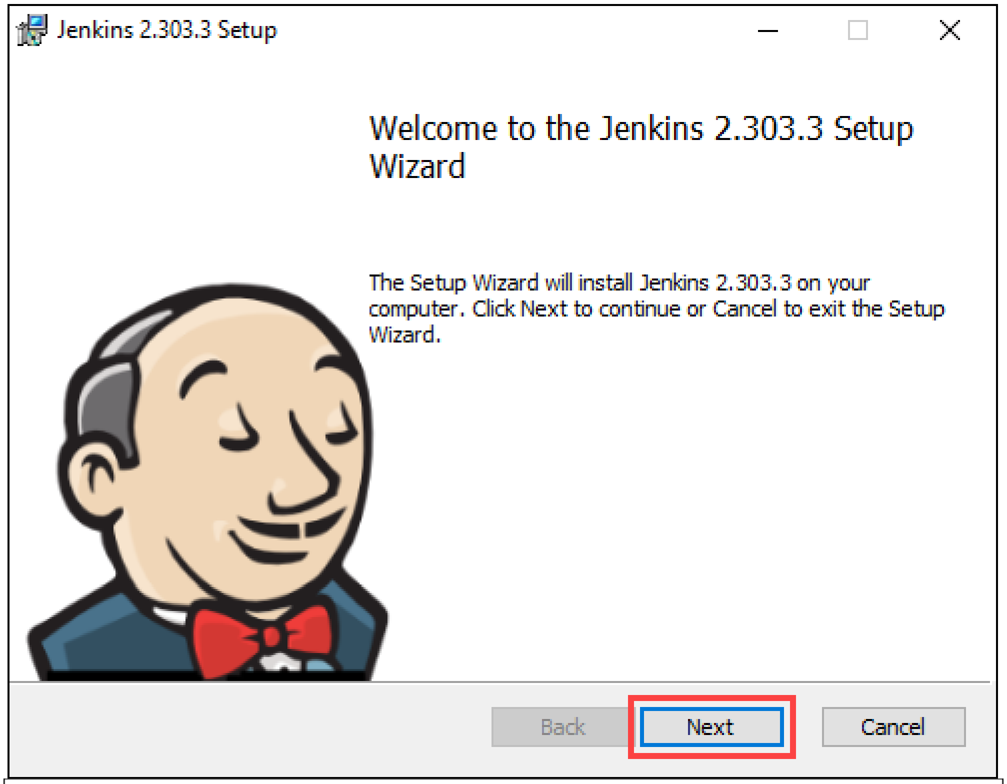
Initialize the Jenkins environment:

- Follow the instructions on the Jenkins setup wizard to initialize the Jenkins environment.
- This process involves installing recommended plugins, setting up security, and creating the first admin user.

Explore the Jenkins environment:

- Once the Jenkins environment is set up, you can explore the various features and functionalities available in the web interface.
- Jenkins has a rich user interface that provides access to features such as build history, build statistics, and system information.

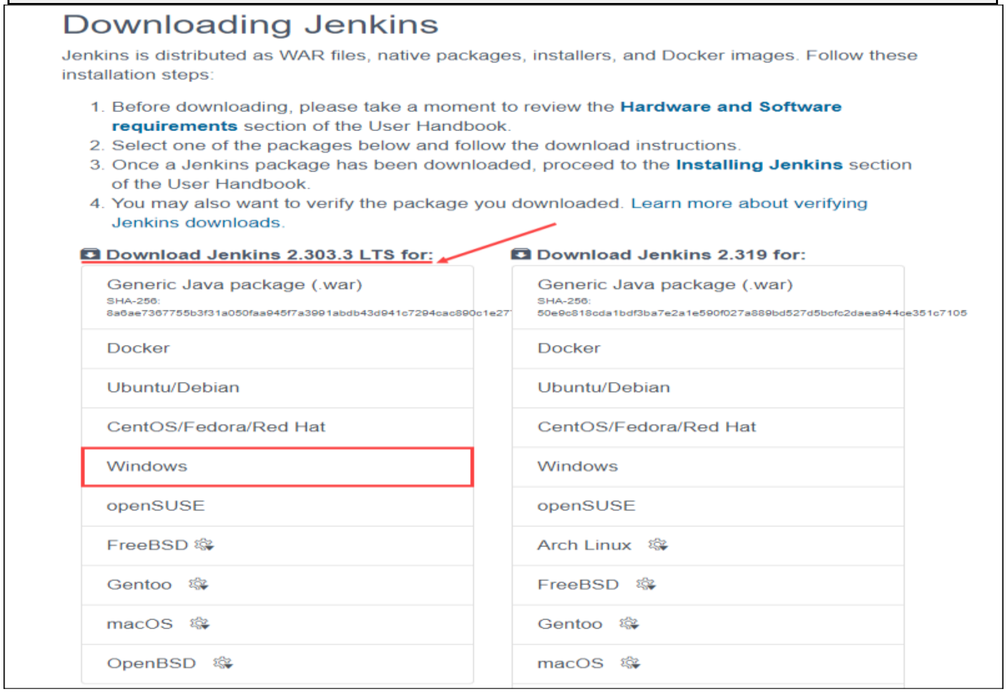
These are the basic steps to install and set up Jenkins. Depending on your use case, you may need to customize your Jenkins environmentfurther.



Jenkins 2.303.3 Setup

Welcome to the Jenkins 2.303.3 Setup Wizard

The Setup Wizard will install Jenkins 2.303.3 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.



Downloading Jenkins

Jenkins is distributed as WAR files, native packages, installers, and Docker images. Follow these installation steps:

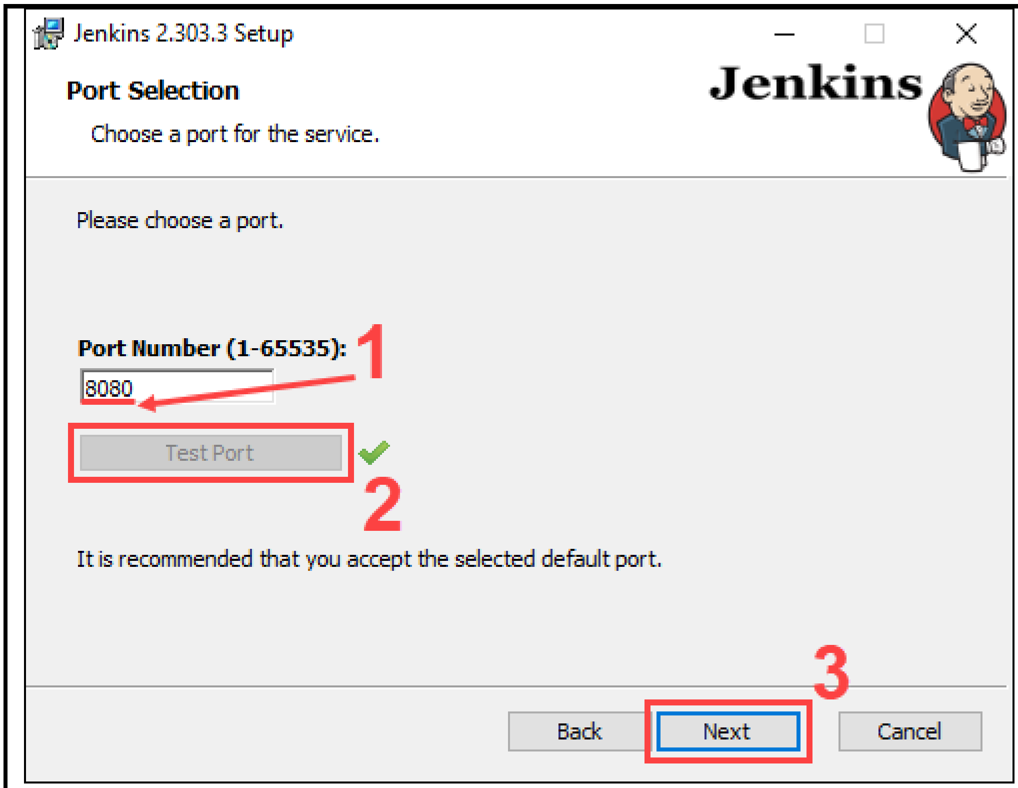
1. Before downloading, please take a moment to review the **Hardware and Software requirements** section of the User Handbook.
2. Select one of the packages below and follow the download instructions.
3. Once a Jenkins package has been downloaded, proceed to the **Installing Jenkins** section of the User Handbook.
4. You may also want to verify the package you downloaded. Learn more about verifying Jenkins downloads.

Download Jenkins 2.303.3 LTS for:

- Generic Java package (.war)
  SHA-256:
  8a6ae7367755b3f31a050faa945f7a3991abdb43d941c7294cac890c1e27
- Docker
- Ubuntu/Debian
- CentOS/Fedora/Red Hat
- Windows
- openSUSE
- FreeBSD
- Gentoo
- macOS
- OpenBSD

Download Jenkins 2.319 for:

- Generic Java package (.war)
  SHA-256:
  50e9c818cda1bdf3ba7e2a1e590f027a889bd527d5bcfc2daea944ce351c7105
- Docker
- Ubuntu/Debian
- CentOS/Fedora/Red Hat
- Windows
- openSUSE
- Arch Linux
- FreeBSD
- Gentoo
- macOS



Jenkins 2.303.3 Setup

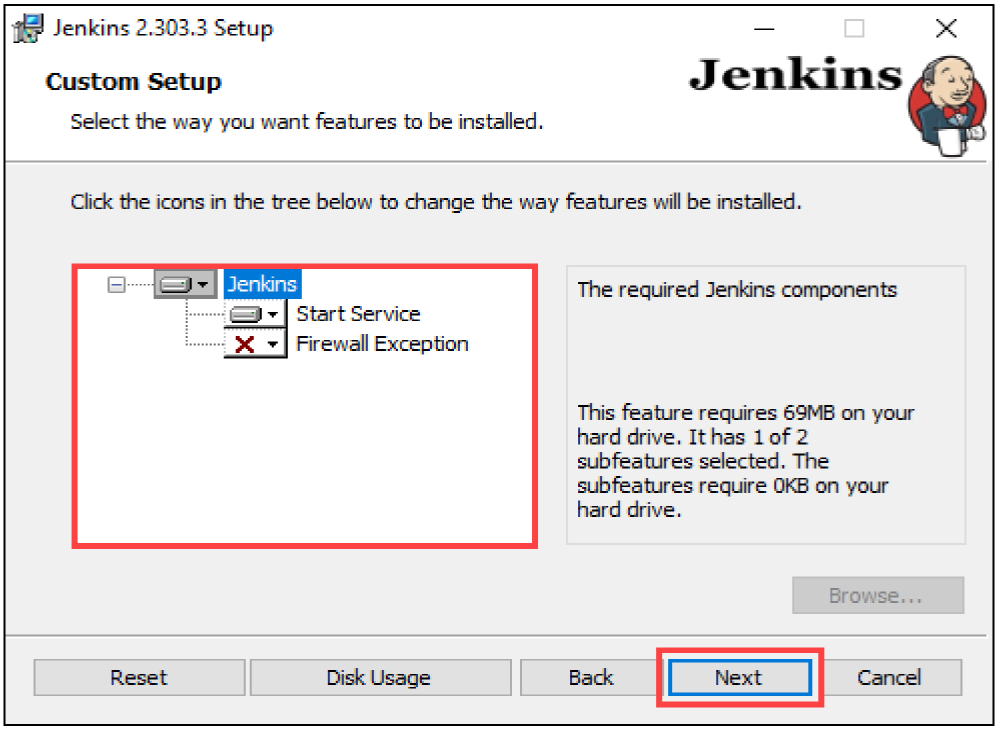
Port Selection

Choose a port for the service.

Please choose a port.

Port Number (1-65535): 8080

It is recommended that you accept the selected default port.



Jenkins 2.303.3 Setup

Custom Setup

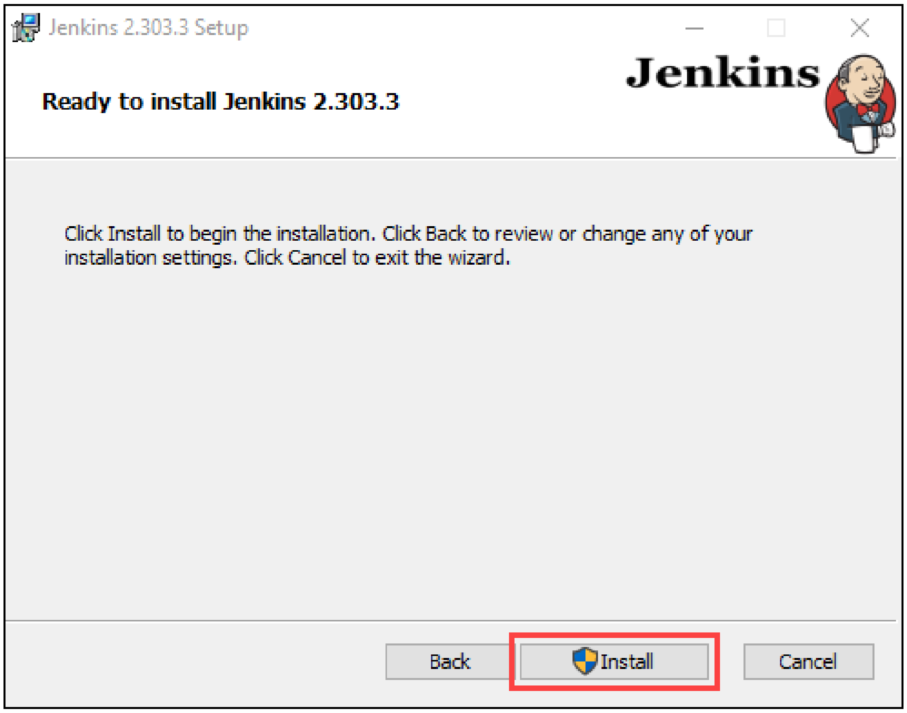
Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

- Jenkins
  - Start Service
  - Firewall Exception

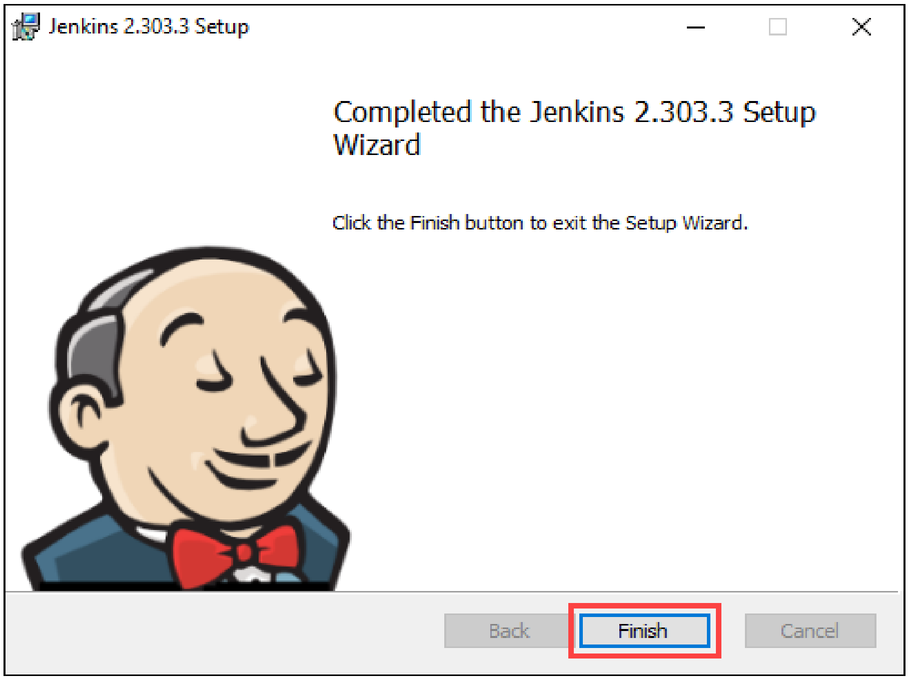
The required Jenkins components

This feature requires 69MB on your hard drive. It has 1 of 2 subfeatures selected. The subfeatures require 0KB on your hard drive.



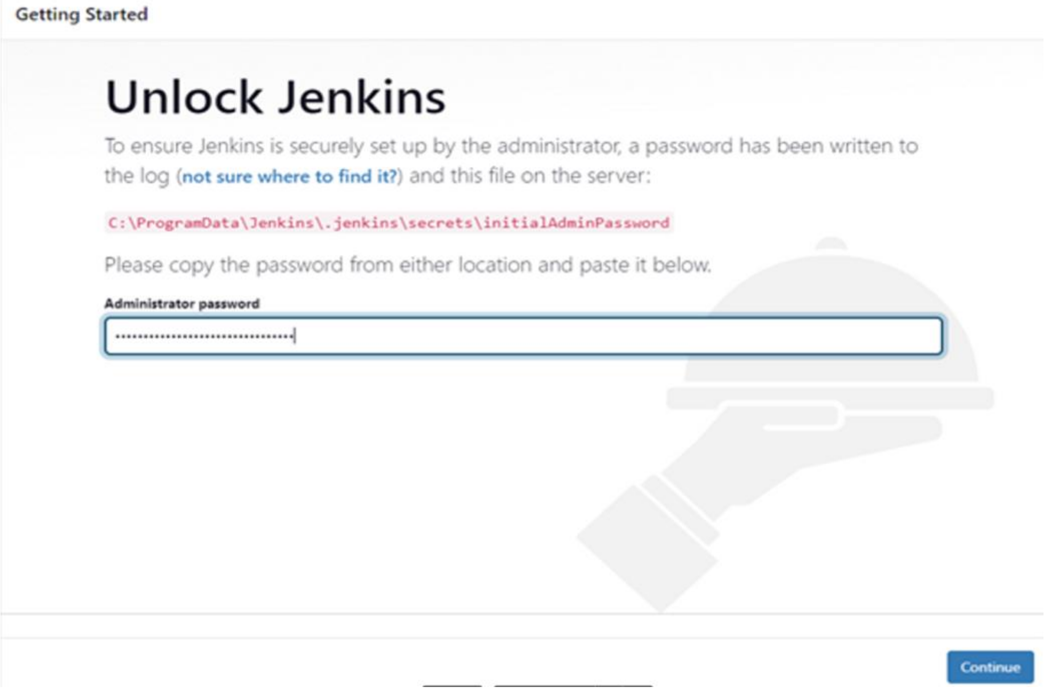
Ready to install Jenkins 2.303.3

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.



Completed the Jenkins 2.303.3 Setup Wizard

Click the Finish button to exit the Setup Wizard.



Getting Started

Unlock Jenkins

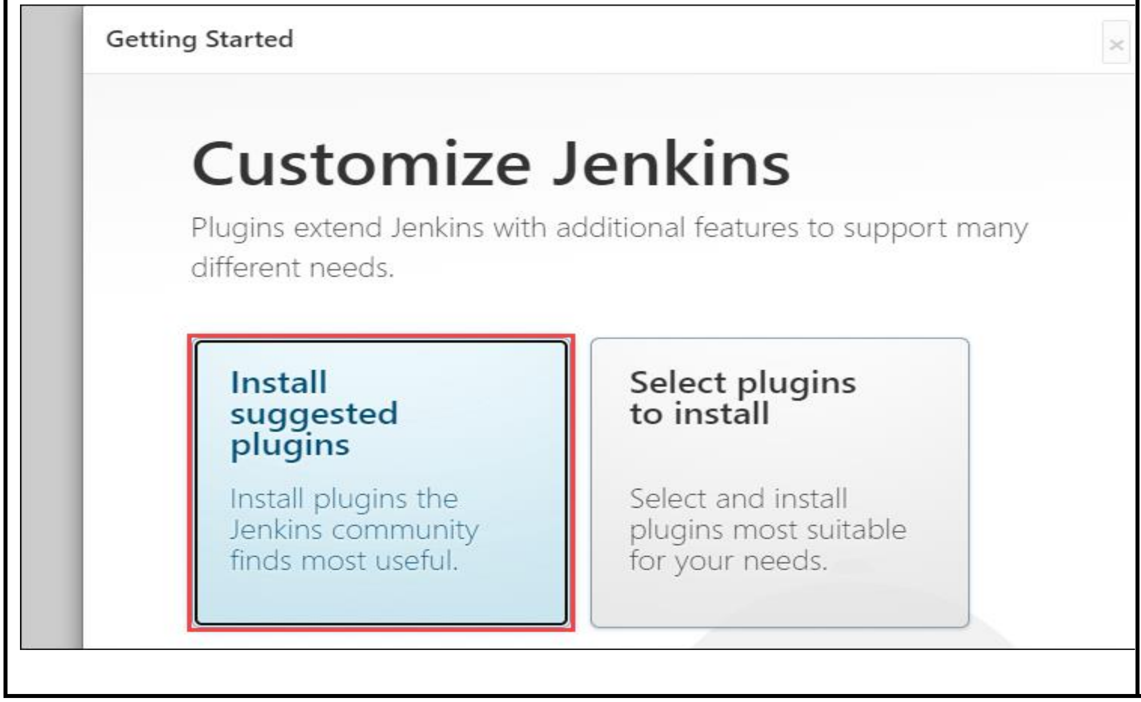
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

································



Getting Started

Customize Jenkins

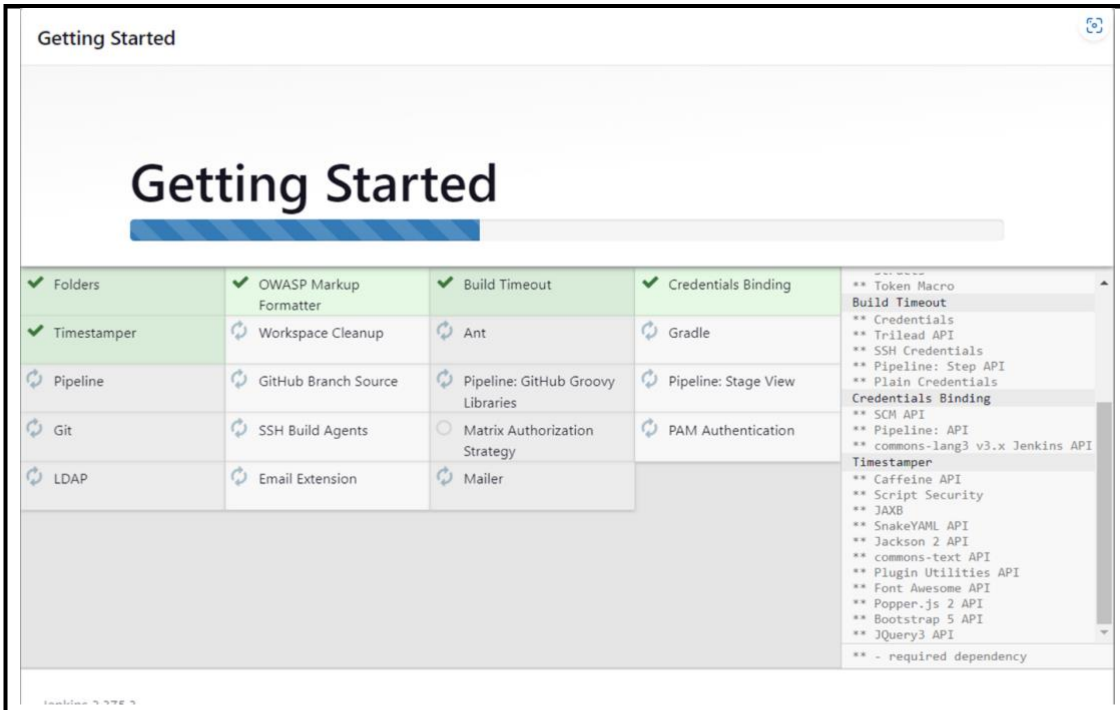
Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

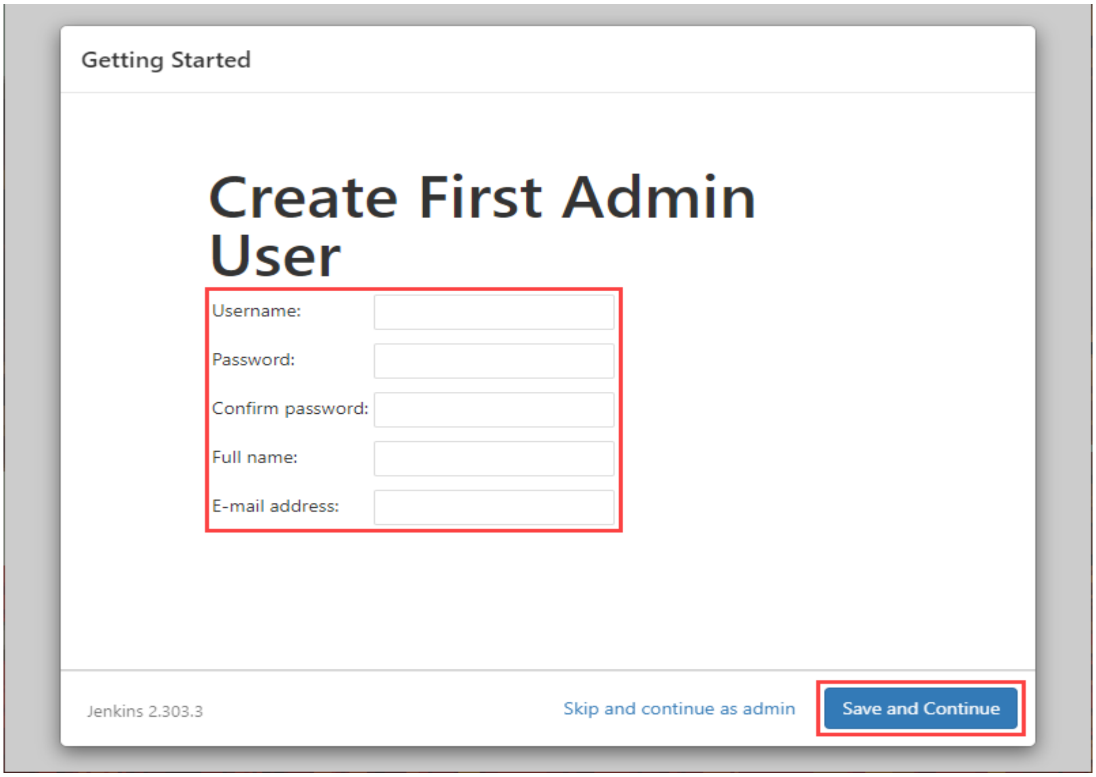


Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	↻ Workspace Cleanup	↻ Ant	↻ Gradle
↻ Pipeline	↻ GitHub Branch Source	↻ Pipeline: GitHub Groovy Libraries	↻ Pipeline: Stage View
↻ Git	↻ SSH Build Agents	○ Matrix Authorization Strategy	↻ PAM Authentication
↻ LDAP	↻ Email Extension	↻ Mailer	

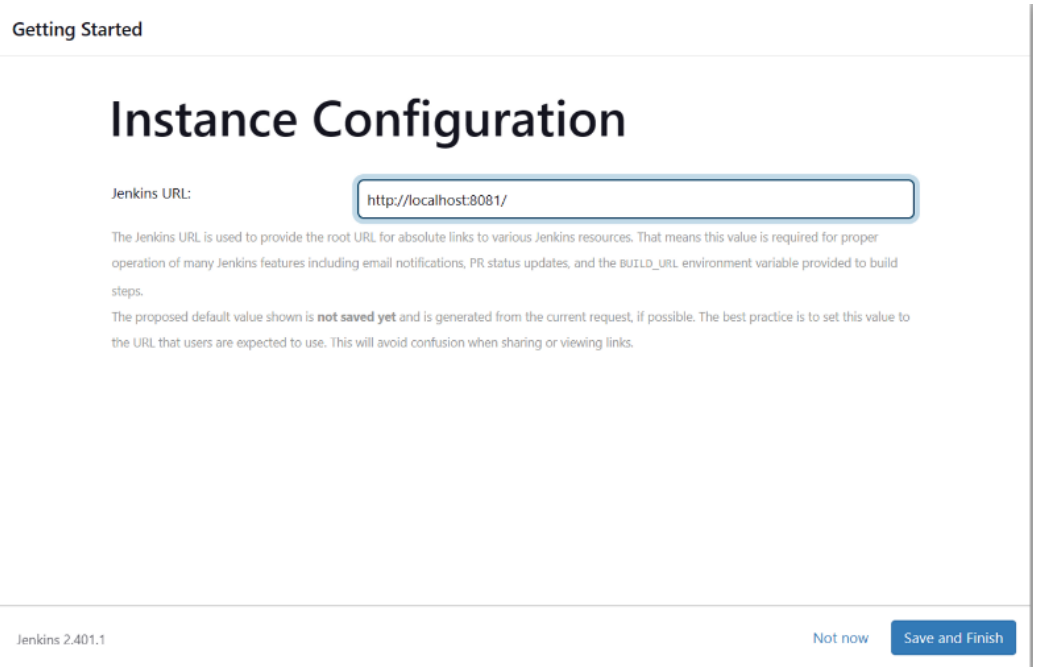
```
** Token Macro
Build Timeout
** Credentials
** Trilead API
** SSH Credentials
** Pipeline: Step API
** Plain Credentials
Credentials Binding
** SCM API
** Pipeline: API
** commons-lang3 v3.x Jenkins API
Timestamper
** Caffeine API
** Script Security
** JAXB
** SnakeYAML API
** Jackson 2 API
** commons-text API
** Plugin Utilities API
** Font Awesome API
** Popper.js 2 API
** Bootstrap 5 API
** JQuery3 API
** - required dependency
```



Getting Started

Create First Admin User

Username:	
Password:	
Confirm password:	
Full name:	
E-mail address:	



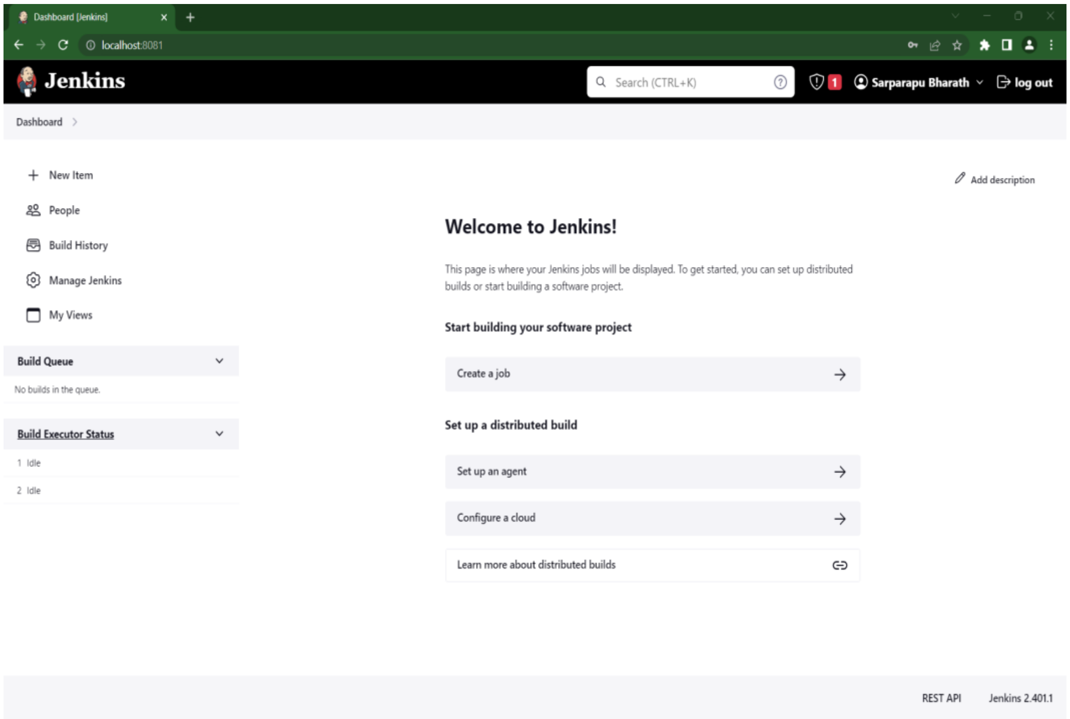
Getting Started

Instance Configuration

Jenkins URL: http://localhost:8081/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.



No builds in the queue.

1 Idle

2 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds