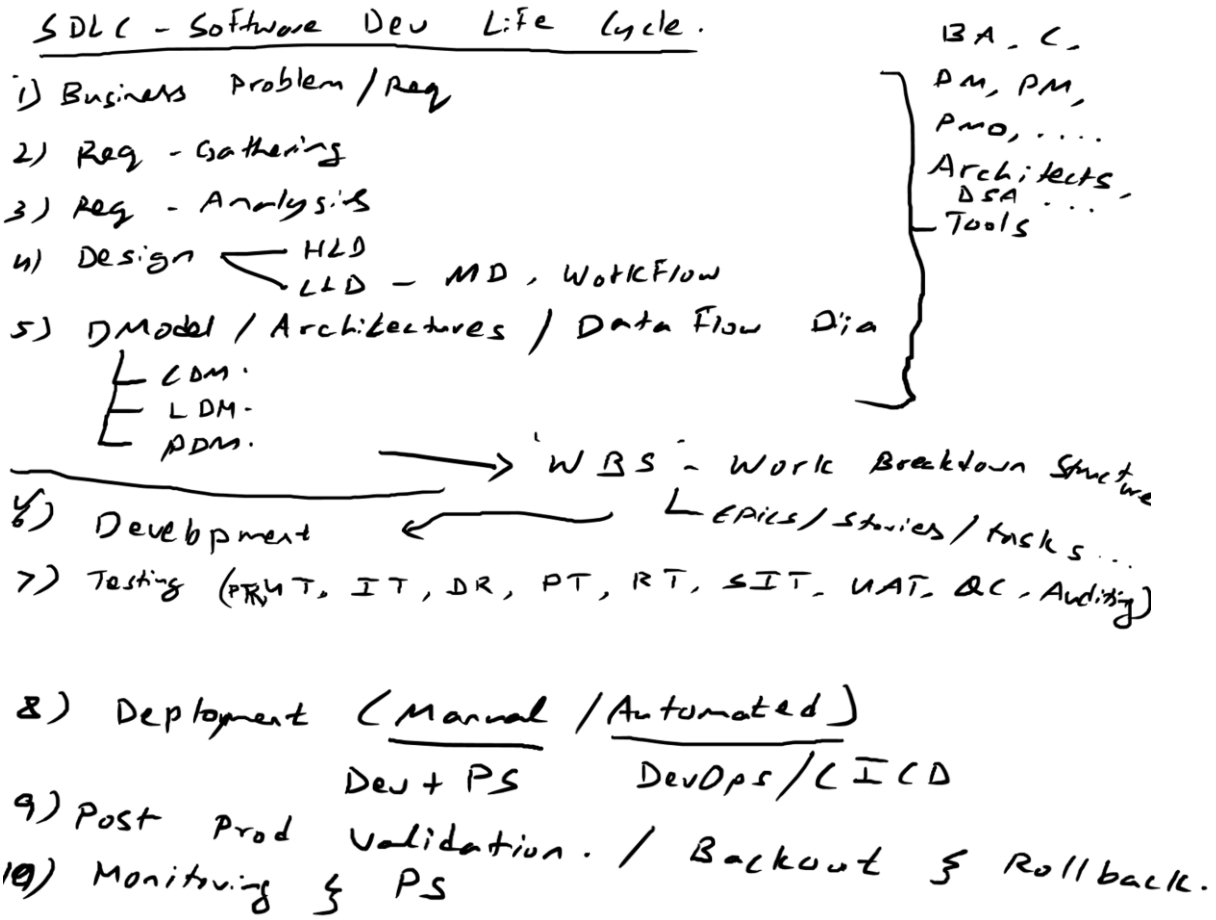
Web: Inceptez.com Mail: info@inceptez.com Call: 7871299810, 7871299817

Table of Contents

CICD – Continuous Integration & Continuous Delivery/Deployment (DEVOPS)

What SDLC model or how you are productionizing the end to end development code?

Tell all SDLC stories and importantly “After requirement, dev, testing, I commit & Push the code into GitHub and then Jenkins pipeline configured by our Devops team will move the code to GCP bucket, so that Airflow can pick it and run the Data pipelines in the Prod env on the predefined schedule... Please correct me.



Scrum Master/ Agile coach

1. SDLC model – **Agile (Sprint (releases) planning)** & Scrum/RAD/Waterfall/V-Model
2. Daily Standup calls we will have in the morning (getting work assigned/discuss about the work we are going to do) and evening (questions/doubts/challenges/show stoppers)

3. Release Cycle – Biweekly Sprints created with different Jira ticket – user story points (requirement) 3/5/8/13 (Work breakdown structure) – split into multiple sub tasks.. Stories can be put in the behind to execute later (backlog), spike story (came adhoc based on priority we will execute).

Informal (adhoc) –

Leads/Clients provide/Source systems/Design team/allocate some work mails/chats/adhoc calls -> Developer/Data Analyst/TL/Architech -> (extract data -> filtration -> curation) .hql/.sql/.py/.bql -> Validation/Review/UnitTesting/~~QA/Testing/UAT~~ -> upload the code Dev Server/GDrive/Mail/Chat window -> Prod Deployment team/Onshore move the code to -> Production -> Prod Deployment team use the Release document and start the code to run on a defined schedule.

Formal –

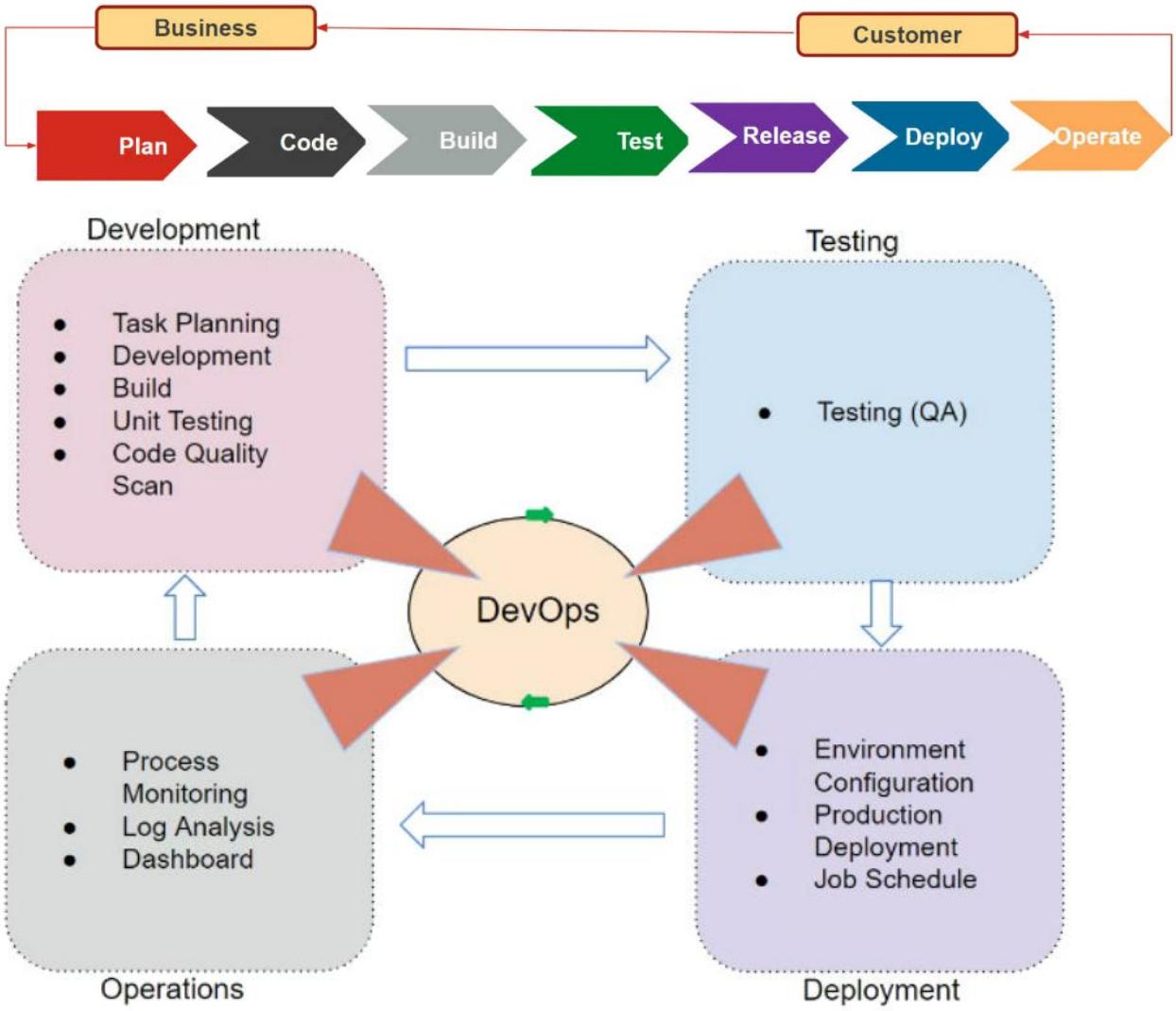
Confluence HLD (Business team requirements) -> LLD documents (Artifacts RACI charts, WBS, DFD, ER Diagrams, Mapping Documents etc.,) (Enablement team/Data modelers/Design team/DSA/TL/BA) ->

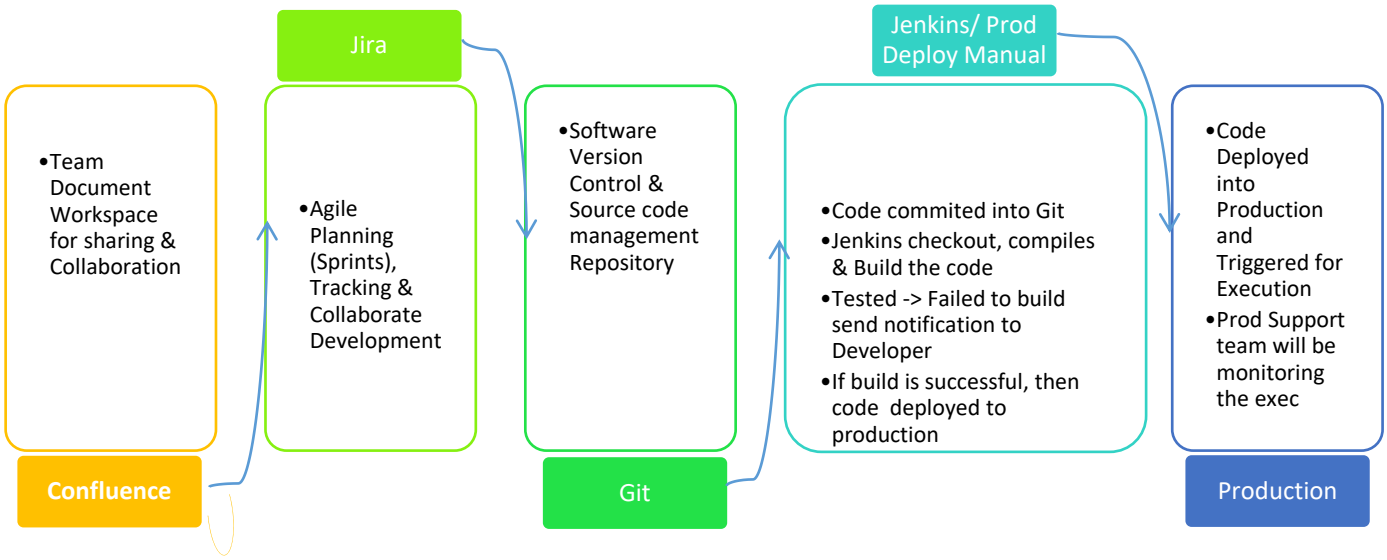
Jira stories (Epics -> Functional Stories & Technical Stories) assigned (PMO assigns Developers/Leads/Architects/Managers)-> Grooming call (Scrum Master or Agile coach will track the progress Managers/clients/architects/Developers) -> **Jira** Stories divided into (Tasks) (Tasks assigned to TLs/Architects/Data engineers/Data Analysts) -> Development (Data engineer) -> **Jira** tickets updated (open-> ready for iteration -> inprogress->cancel/ready for prod(RFP)->Done) (Data engineers updates comments and change status on a regular basis)->

Git/Github/VCS/SCM/bitbucket (**Dev/feature** branch) (Data engineer)-> Peer Review (Lead/Data engineer)-> Unit Testing/test evidence (Data engineer)-> dry run + Integration Testing (Data engineer/Testing team) -> Checkout code from **Git** and deploy into UAT/Pre Prod Server (Data engineer)-> Run the code in UAT (Data engineer)-> Pre Prod/UAT Validation (Design Team/Enablement Team/Data Solution Architects/Leads/Architects) -> Checkin the final code to **Git** (**Master/Main** branch) (Data engineer)->

Jenkins Pipeline (Checkout code from Git -> Build of code -> Test of code) (DevOps engineer team) -> **Jenkins Pipeline** (Shell Scripts/Terraform/AWS Code Deploy/GCP Workflow Manager/Azure Devops components) do a Prod Deployment (DevOps engineer)-> Production support team monitor

Leads/Clients provide/allocate some work through (Confluence) Atlassian ->



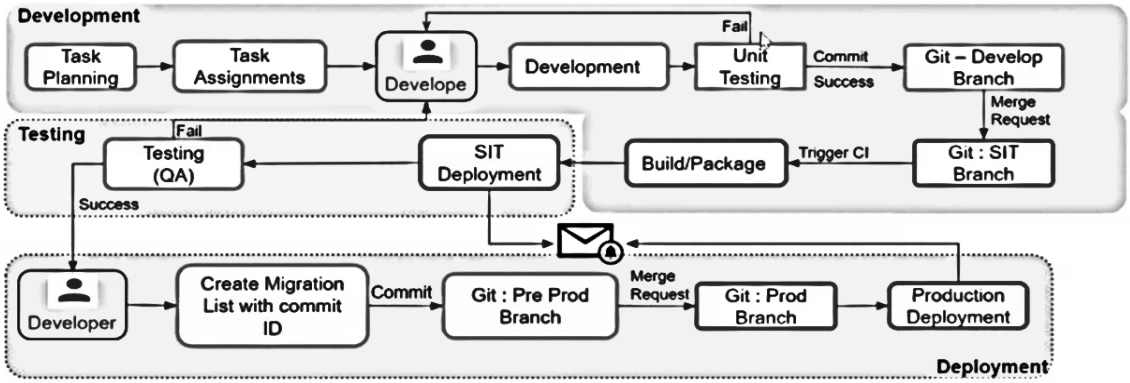


Confluence – Knowledge base (documents, stories, supportive content, release steps, back out scripts, support steps)

Jira - Epics (larger stories Dev/Maintenance/Enhance/Bugs) Stories --> Tasks/Issues -> Subtasks (Stories)

Git - GitHub is a code hosting platform for collaboration and version control. GitHub lets you (and others) work together on projects.

Jenkins is an open-source Continuous Integration and Continuous Deployment (CI/CD) tool for automating the software development life cycle (SDLC).



Agile – Sprints (release cycle) consist of stories and epic can roll over to multiple sprint. Sprint cycle can be 2 weeks or 5 days etc depending on project model

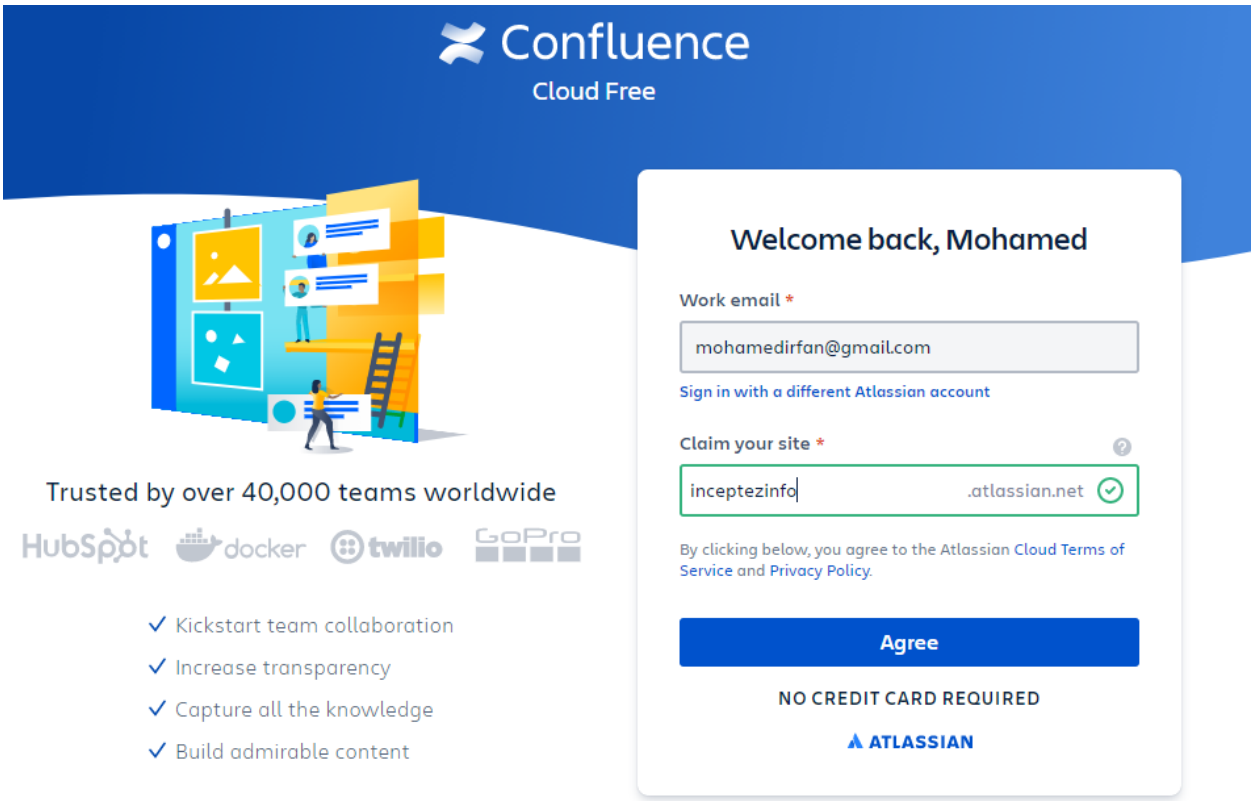
Prerequisites (Install the following tools in Windows):

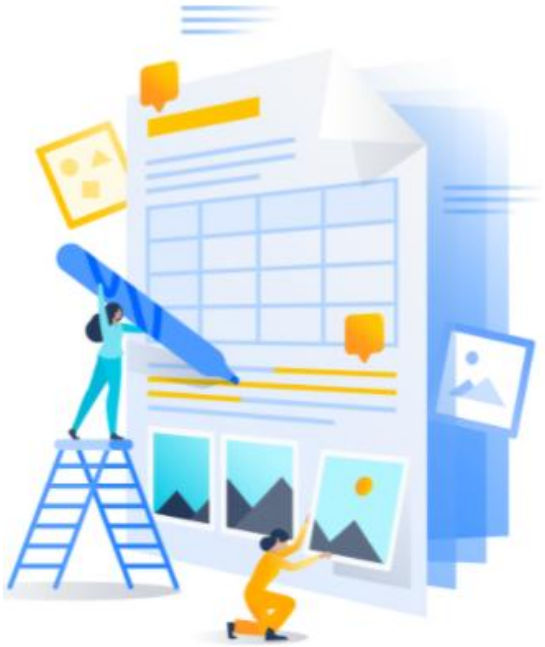
1. Install Github for desktop (Git Cli, Git Plugins for Pycharm, Git desktop, Git Web Gui)
   https://desktop.github.com/
2. Install java 11 & Jenkins
   https://drive.google.com/drive/folders/1eomjZZZ0sf9UvlgmVPD4DoCVa3_U5elt?usp=share_link
3. Install and Configure Pycharm for windows
   https://docs.google.com/document/d/1tiMiDEmsKav03khFLVxFVqbgHQx_GY-V/edit?usp=share_link&ouid=106781519779062697823&rtpof=true&sd=true
4. Link for Confluence and Jira
   https://www.atlassian.com/software/confluence/free
5. Link for Git
   https://github.com/mohamedirfan
6. Link for Jenkins
   https://www.jenkins.io/download/thank-you-downloading-windows-installer-stable

Confluence:

Confluence is a team workspace where knowledge and collaboration meet. Dynamic pages give your team a place to create, capture, and collaborate on any project or idea.

https://www.atlassian.com/software/confluence/free





Invite your teammates

Bring your team along for the ride!

mohamedirfan@gmail.com

info@inceptez.com

inceptezwebd35@gmail.com

Let my teammates invite other people

You can change these settings at any time. Continue

Create (custom/from template/import) -> Publish the project -> Share to users

Jira:

Jira is a software application used for issue tracking and project management. The tool, developed by the Australian software company Atlassian

Issue Types

Jira can be used to track many different types of issues. The currently defined issue types are listed below. In addition, you can add more in the administration section.

For Regular Issues

Epic	A big user story that needs to be broken down. Created by Jira Software - do not edit or delete.
Improvement	An improvement or enhancement to an existing feature or task.
Task	A small, distinct piece of work.
New Feature	A new feature of the product, which has yet to be developed.
Bug	A problem or error.

For Sub-Task Issues

Sub-task	

Priority Levels

An issue has a priority level which indicates its importance. The currently defined priorities are listed below. In addition, you can add more priority levels in the administration section.

Highest	This problem will block progress.
High	Serious problem that could block progress.
Medium	Has the potential to affect progress.
Low	Minor problem or easily worked around.
Lowest	Trivial problem with little or no impact on progress.

Statuses

Each issue has a status, which indicates the stage of the issue. In the default workflow, issues start as being Open, progressing to In Progress, Resolved and then Closed. Other workflows may have other status transitions.

OPEN	The issue is open and ready for the assignee to start work on it.
IN PROGRESS	This issue is being actively worked on at the moment by the assignee.
REOPENED	This issue was once resolved, but the resolution was deemed incorrect. From here issues are either marked assigned or resolved.
RESOLVED	A resolution has been taken, and it is awaiting verification by reporter. From here issues are either reopened, or are closed.
CLOSED	The issue is considered finished, the resolution is correct. Issues which are closed can be reopened.
BUILDING	Source code has been committed, and JIRA is waiting for the code to be built before moving to the next status.
BUILD BROKEN	The source code committed for this issue has possibly broken the build.
TO DO	
IN REVIEW	
DONE	

Git-Hub – WebUI, Desktop , IDE (Eclipse) & CLI

GIT (2005) – Version control

A system that keeps records of your changes

Allows for collaborative development

Allows you to know who made what changes and when

Allows you to revert any changes and go back to a previous state

Distributed version control

Users keep entire code and history on their location machines

- Users can make any changes without internet access
- (Except pushing and pulling changes from a remote server)

GITHUB - 2008

What is GitHub?

- www.github.com
- Largest web-based git repository hosting service
  - Aka, hosts 'remote repositories'
- Allows for code collaboration with anyone online
- Adds extra functionality on top of git
  - UI, documentation, bug tracking, feature requests, pull requests, *and more!*

Octocat!

https://desktop.github.com/

https://github.com/yourname

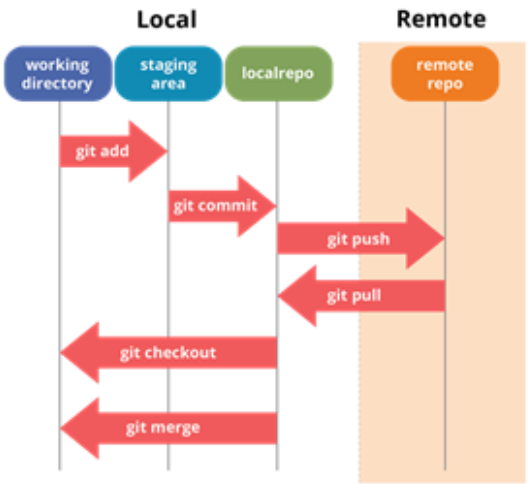
Developer create code and wanted to share the final code into Github directly to Master.

1. Create a repository
2. Clone into local
3. Open in Github Desktop
4. **Create a file and add inside the staging area from working directory**
5. **Do a local Commit to master**
6. **Push to github by clicking push to orgin**

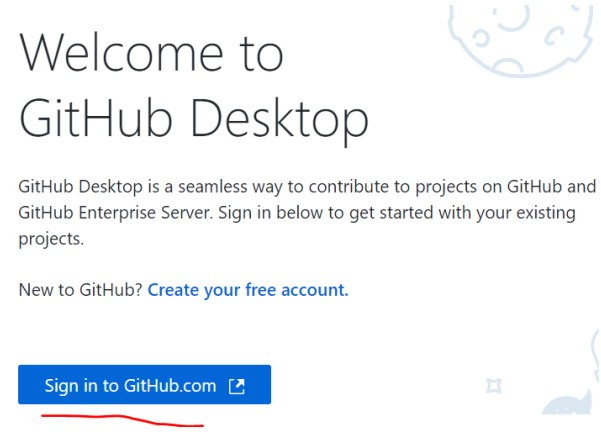
Developer modify the code and wanted to share the code into his own branch and get reviewed and merged to the master branch.

1. Create a Branch in local or remote, choose the new branch , go to repository -> pull
2. Modify the local code
3. Do a local commit to a new branch, then push origin to remote
4. Create a pull request (compare & pull request)
5. Review the code with comments.
6. Do merge (conform merge) and commit the changes from your branch to the master branch
7. Open and merge a pull request -> **branch to the master**

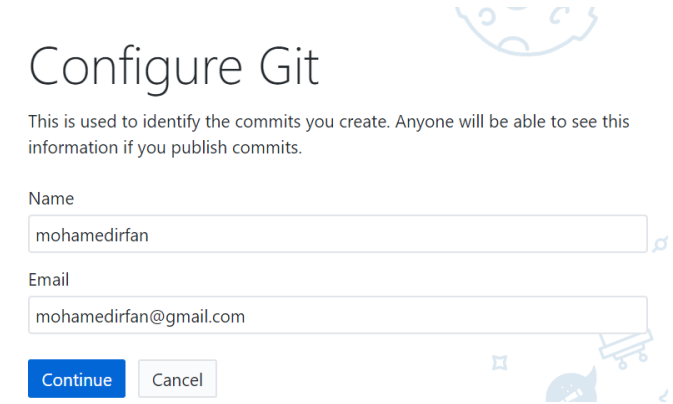
Working Directory - Working Directory which holds the actual files what you have coded.
A staging step (index) in git allows you to continue making changes to the working directory
Local Repo (Head) - when you decide to interact with version control repository (locally in your machine), it allows you to record changes in small commits.
Remote Repo - To send the changes to your remote repository, execute git push

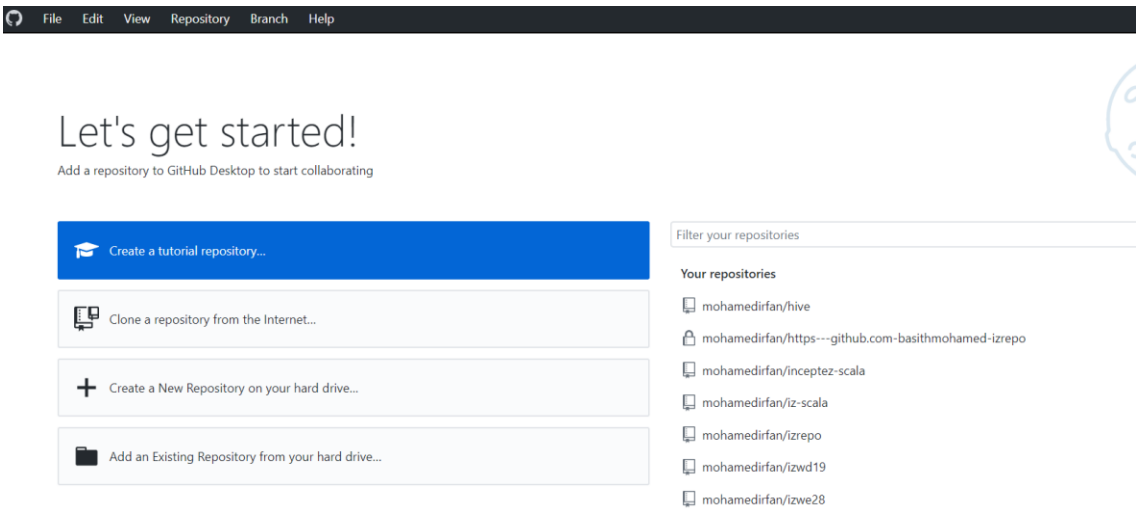


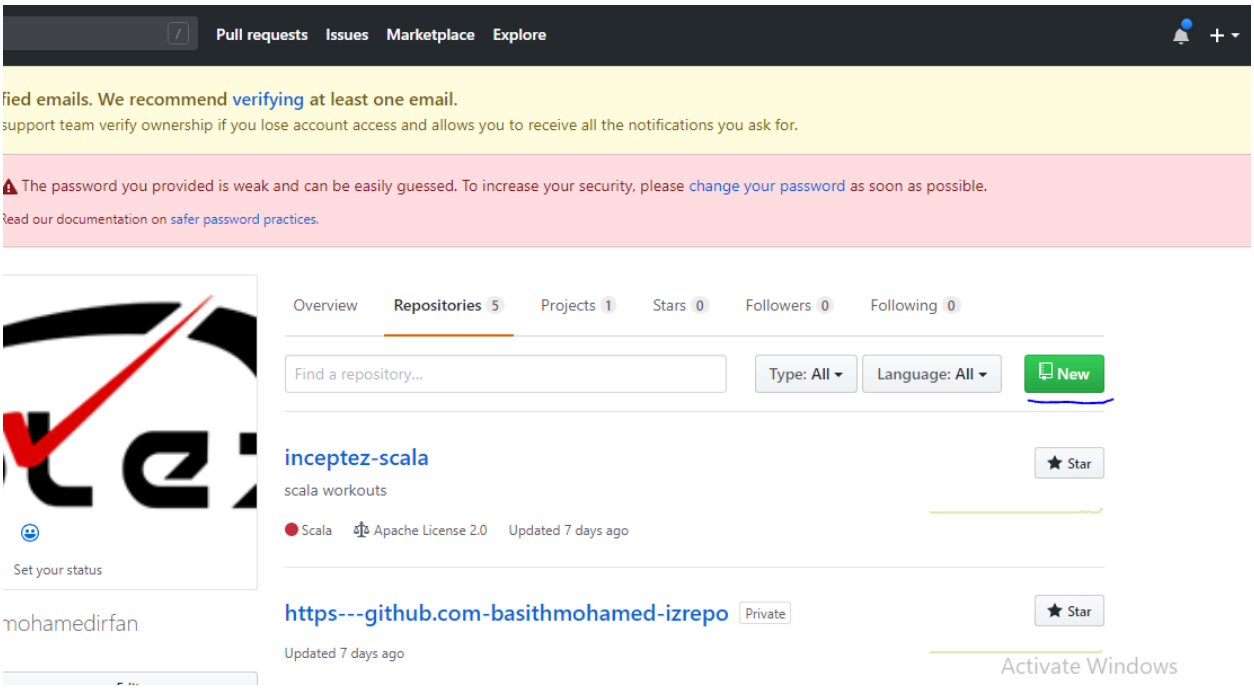
https://desktop.github.com/



After signin with your username and password






Pull requests
Issues
Marketplace
Explore
fied emails. We recommend verifying at least one email.
support team verify ownership if you lose account access and allows you to receive all the notifications you ask for.
The password you provided is weak and can be easily guessed. To increase your security, please change your password as soon as possible.
Read our documentation on safer password practices.
Set your status
mohamedirfan
Overview
Repositories 5
Projects 1
Stars 0
Followers 0
Following 0
Find a repository...
Type: All
Language: All
New
inceptez-scala
scala workouts
Star
Scala
Apache License 2.0
Updated 7 days ago
https---github.com-basithmohamed-izrepo
Private
Star
Updated 7 days ago
Activate Windows

https://github.com/new

Create a new repository

A repository contains all project files, including the revision history.

Owner / Repository name *

mohamedirfan / izrepo

Great repository names are short and memorable. Need inspiration? How about supreme-octo-adventure?

Description (optional)

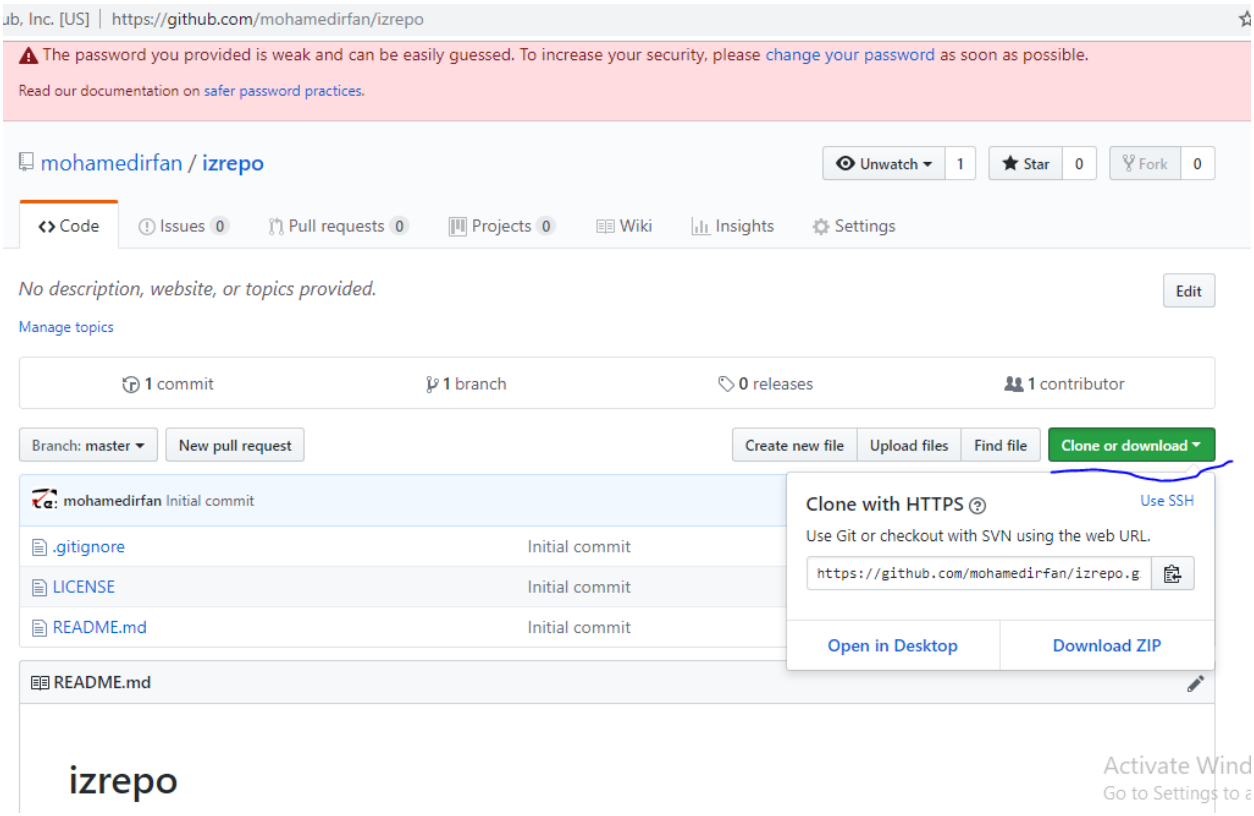
Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: Scala | Add a license: Apache License 2.0

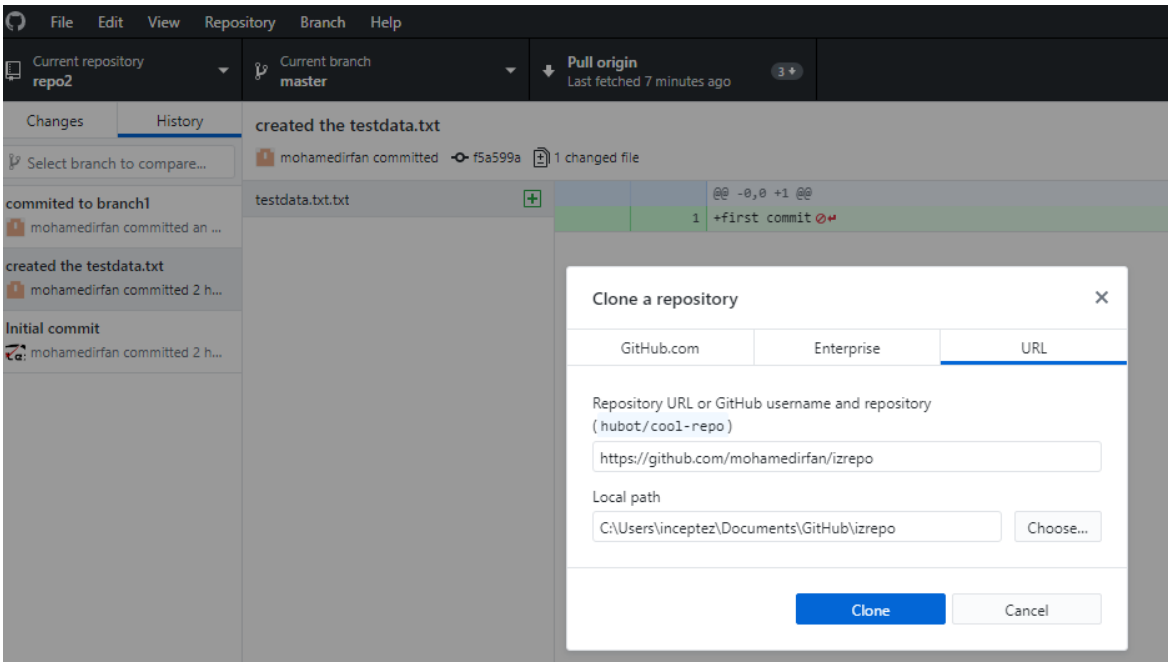
Create repository

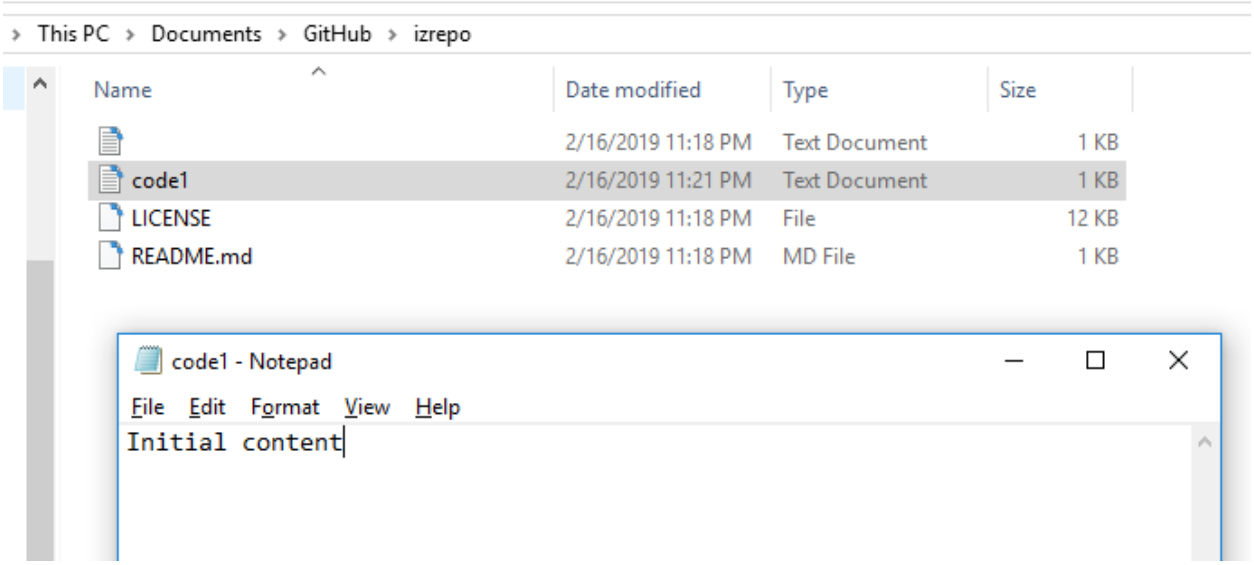


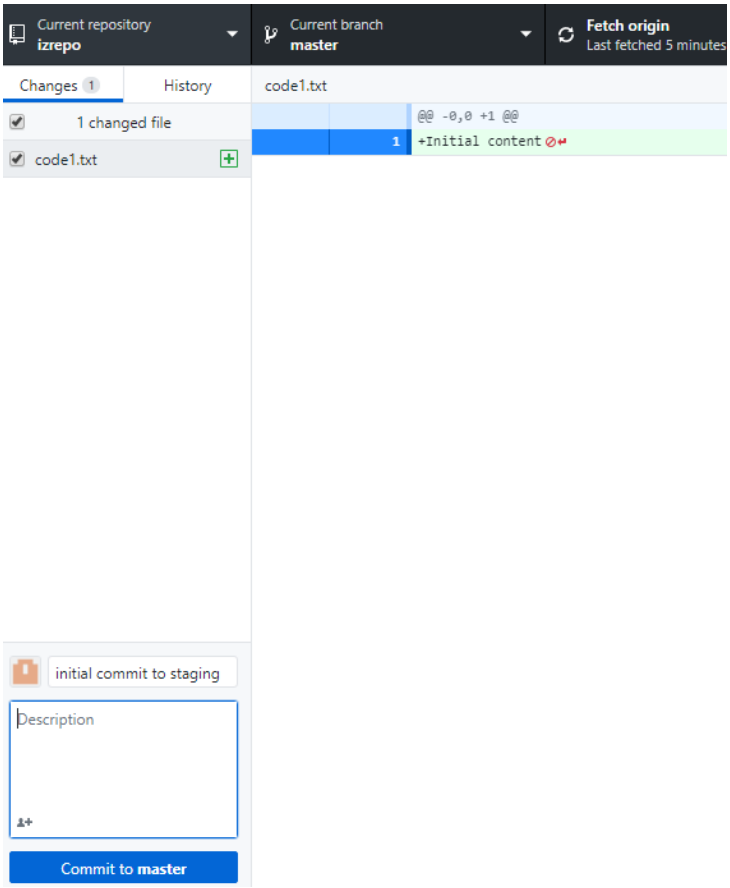
Open in Desktop

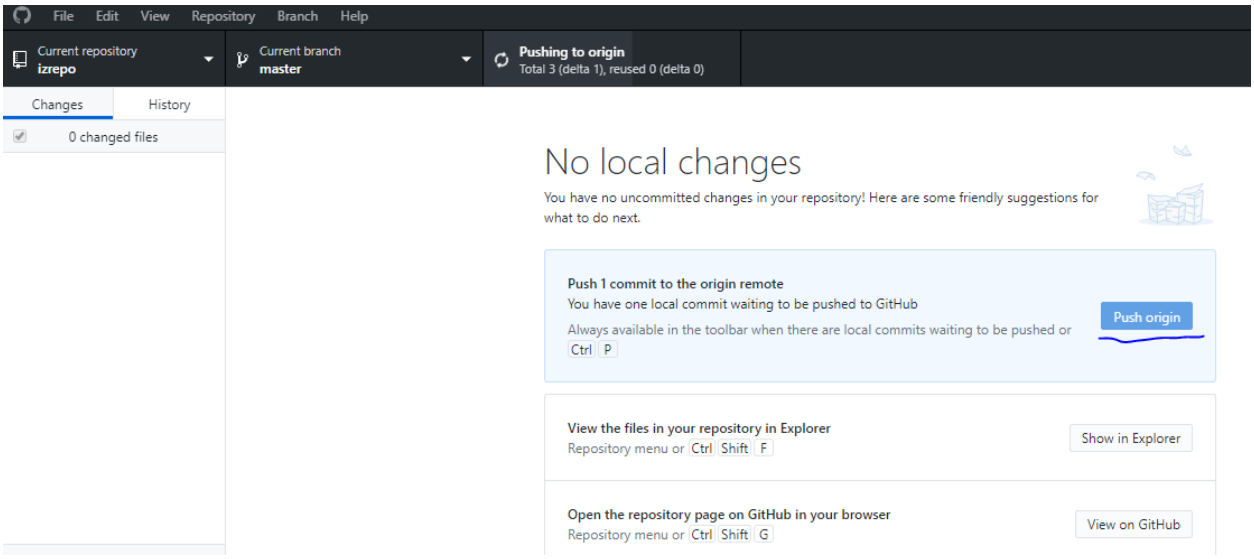
File -> options -> signin

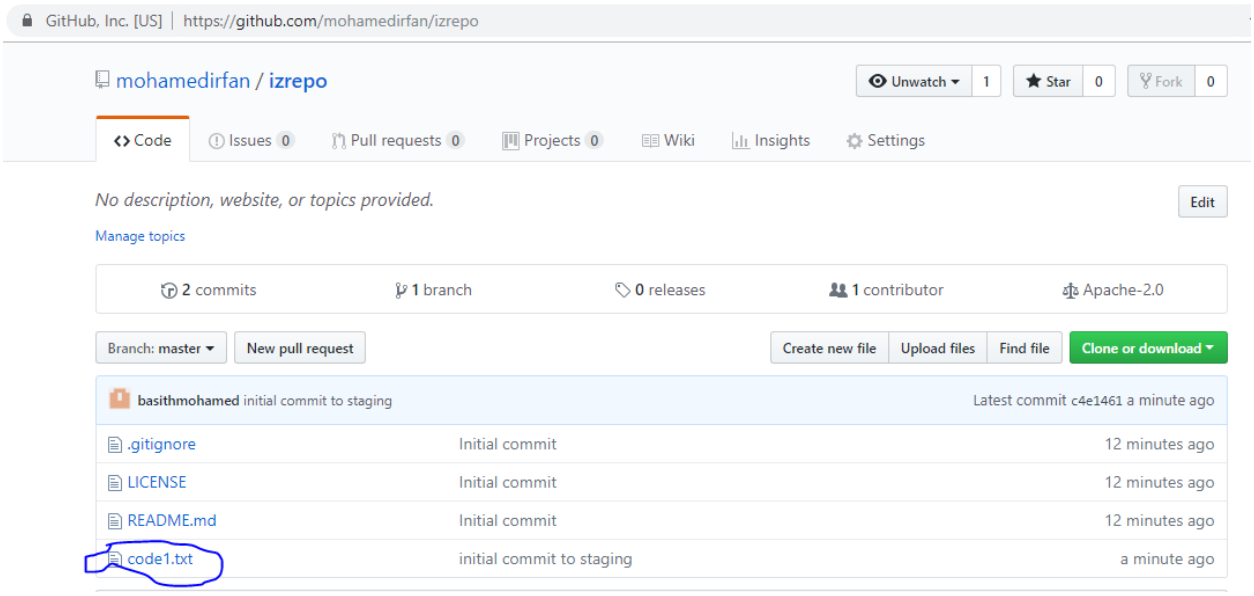
File -> clone repository -> url

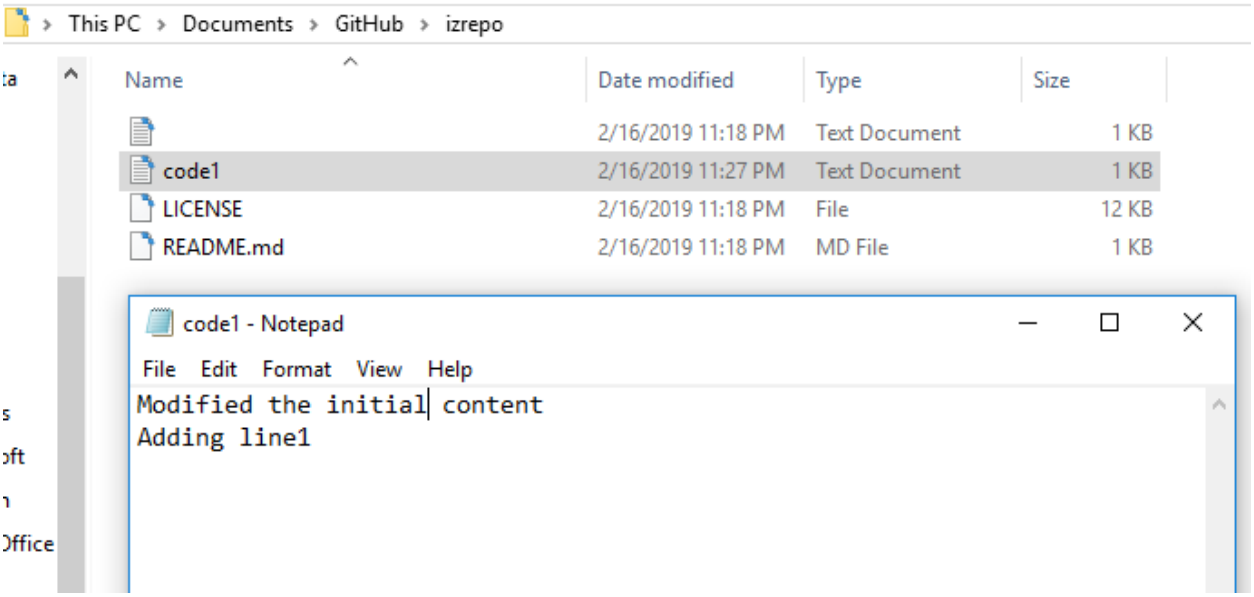


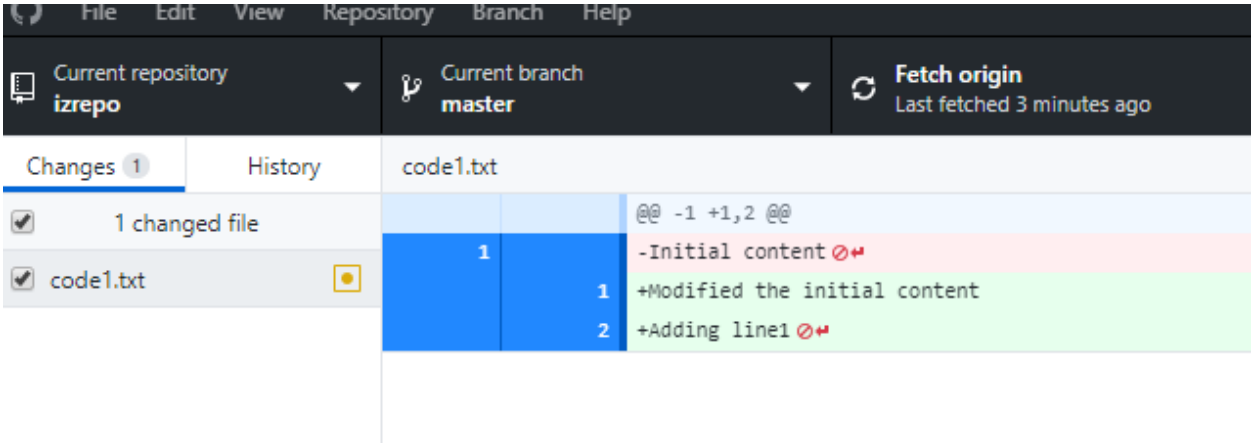

This PC › Documents › GitHub › izrepo
Name
Date modified
Type
Size
2/16/2019 11:18 PM Text Document 1 KB
code1
2/16/2019 11:21 PM Text Document 1 KB
LICENSE
2/16/2019 11:18 PM File 12 KB
README.md
2/16/2019 11:18 PM MD File 1 KB
code1 - Notepad
File Edit Format View Help
Initial content

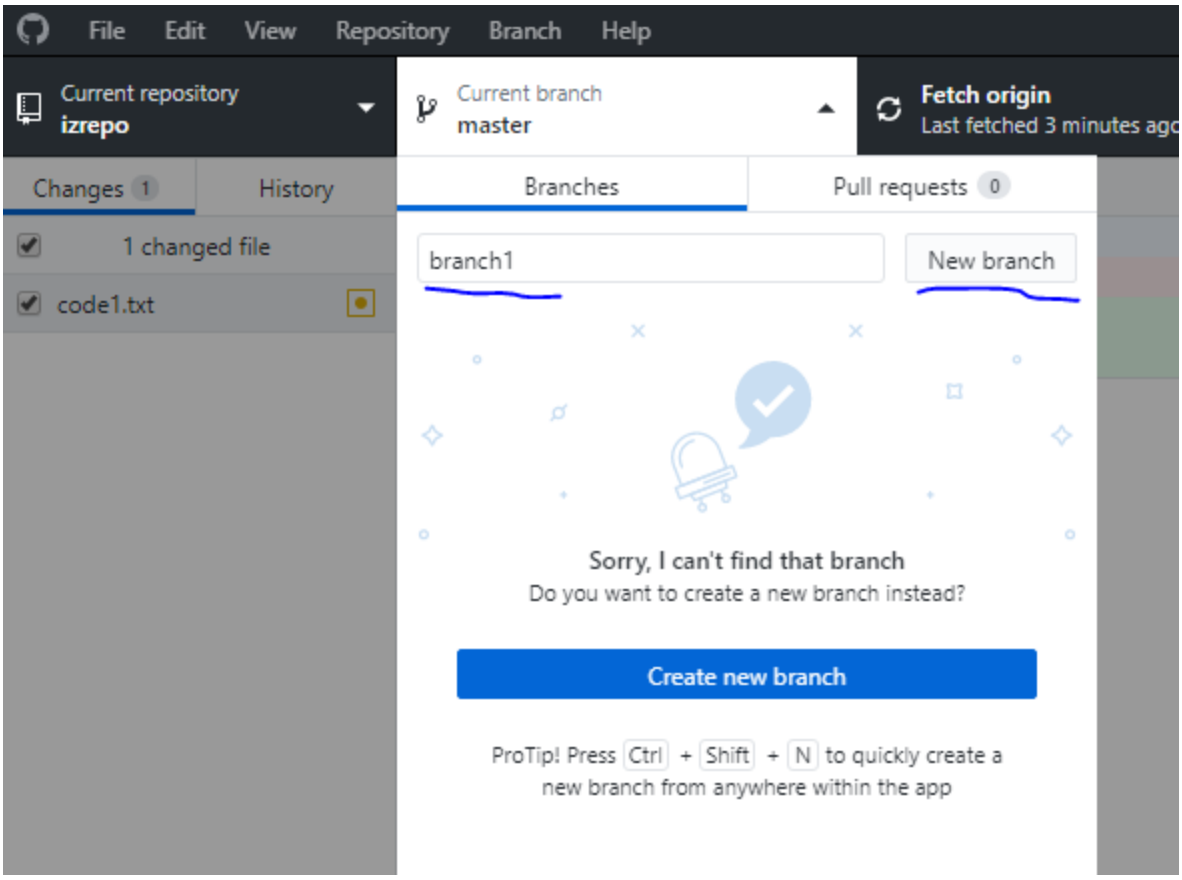

Current repository
izrepo
Current branch
master
Fetch origin
Last fetched 5 minutes
Changes 1
History
1 changed file
code1.txt
code1.txt
@@ -0,0 +1 @@
1
+Initial content
initial commit to staging
Description
Commit to master

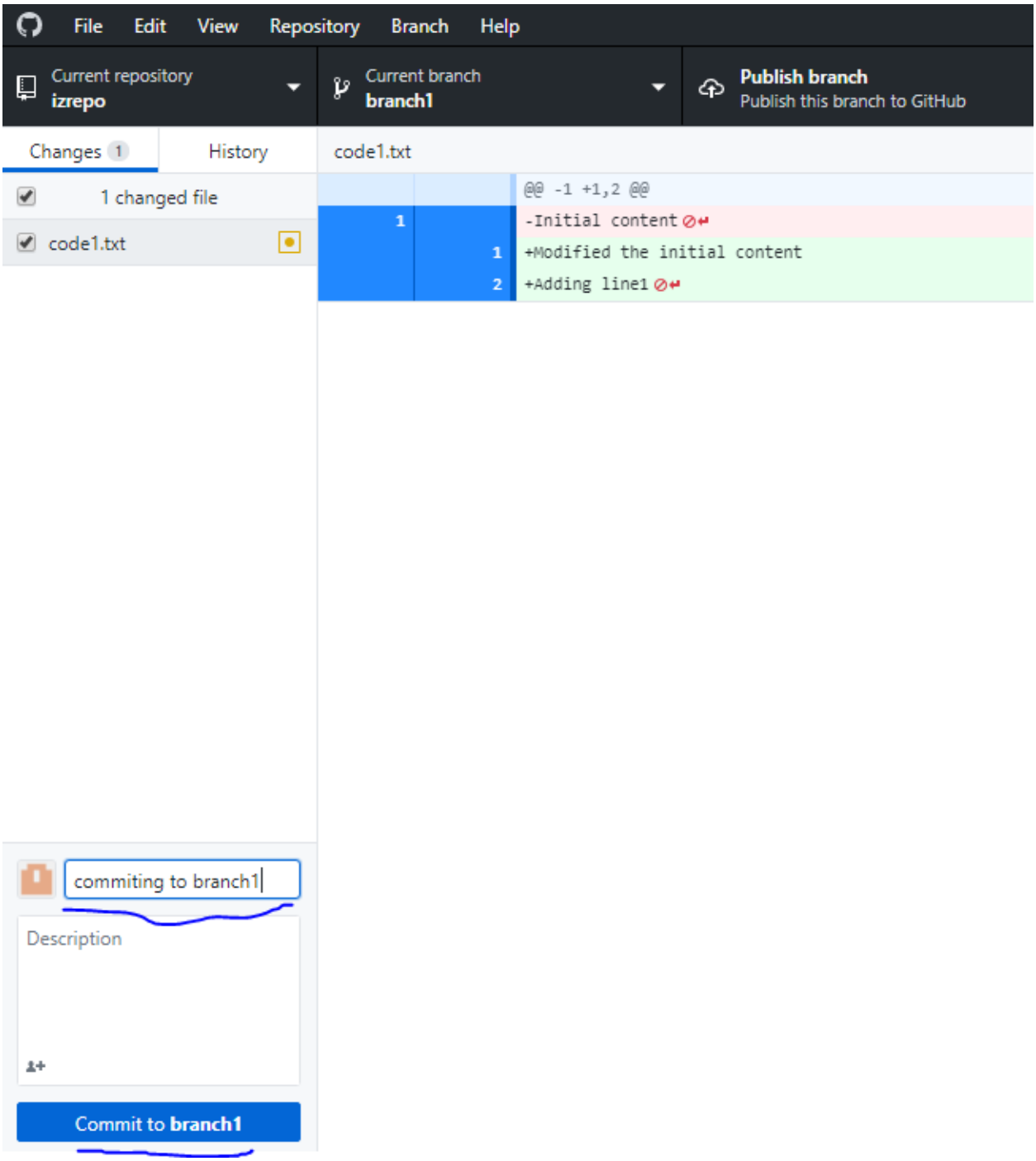

File Edit View Repository Branch Help
Current repository
izrepo
Current branch
master
Pushing to origin
Total 3 (delta 1), reused 0 (delta 0)
Changes
History
0 changed files
No local changes
You have no uncommitted changes in your repository! Here are some friendly suggestions for what to do next.
Push 1 commit to the origin remote
You have one local commit waiting to be pushed to GitHub
Always available in the toolbar when there are local commits waiting to be pushed or Ctrl P
Push origin
View the files in your repository in Explorer
Repository menu or Ctrl Shift F
Show in Explorer
Open the repository page on GitHub in your browser
Repository menu or Ctrl Shift G
View on GitHub

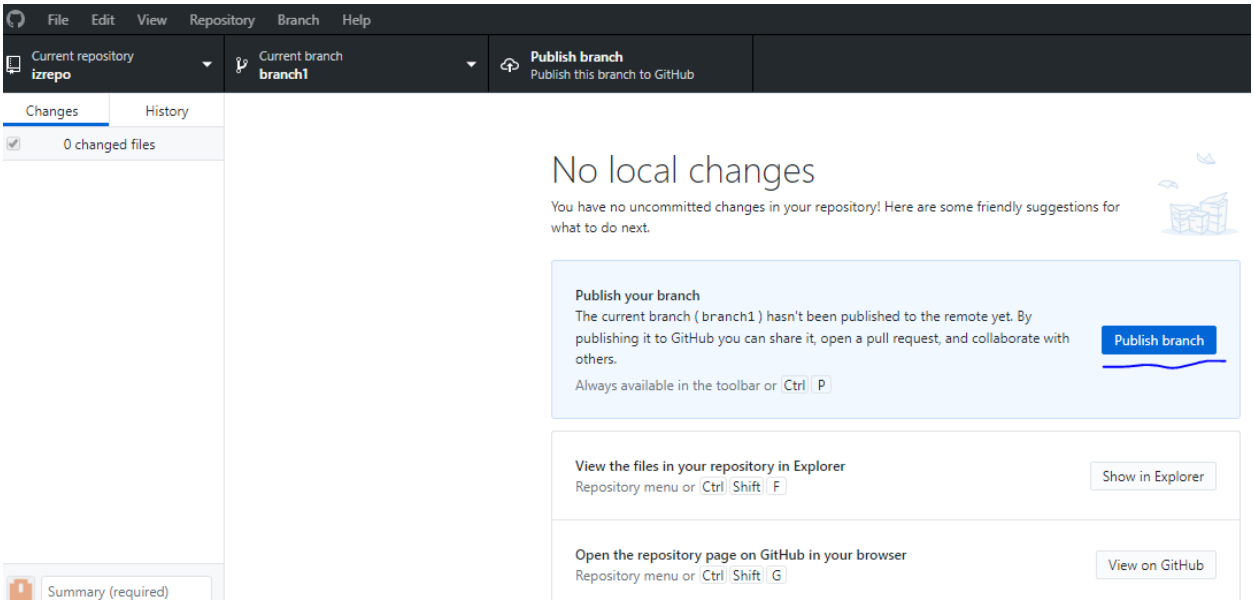

GitHub, Inc. [US] | https://github.com/mohamedirfan/izrepo
mohamedirfan / izrepo
Unwatch 1
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings
No description, website, or topics provided.
Edit
Manage topics
2 commits
1 branch
0 releases
1 contributor
Apache-2.0
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
basithmohamed initial commit to staging
Latest commit c4e1461 a minute ago
.gitignore Initial commit 12 minutes ago
LICENSE Initial commit 12 minutes ago
README.md Initial commit 12 minutes ago
code1.txt initial commit to staging a minute ago

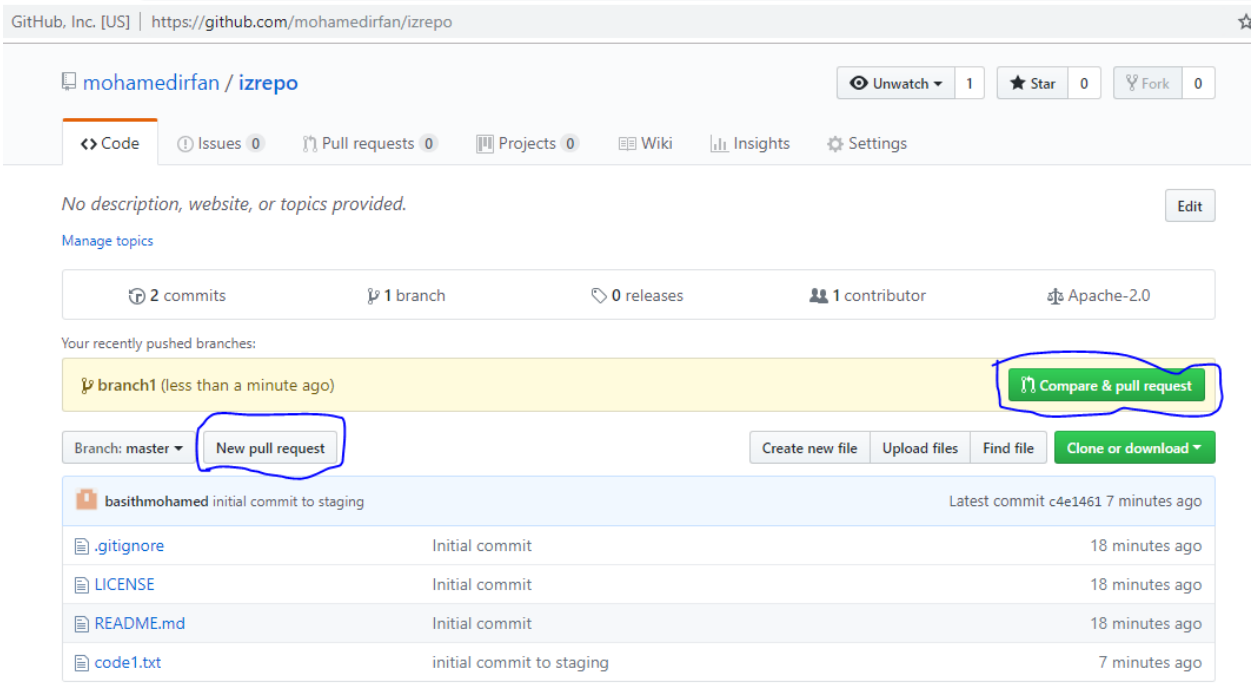

This PC › Documents › GitHub › izrepo
Name
Date modified
Type
Size
2/16/2019 11:18 PM Text Document 1 KB
code1
2/16/2019 11:27 PM Text Document 1 KB
LICENSE
2/16/2019 11:18 PM File 12 KB
README.md
2/16/2019 11:18 PM MD File 1 KB
code1 - Notepad
File Edit Format View Help
Modified the initial content
Adding line1

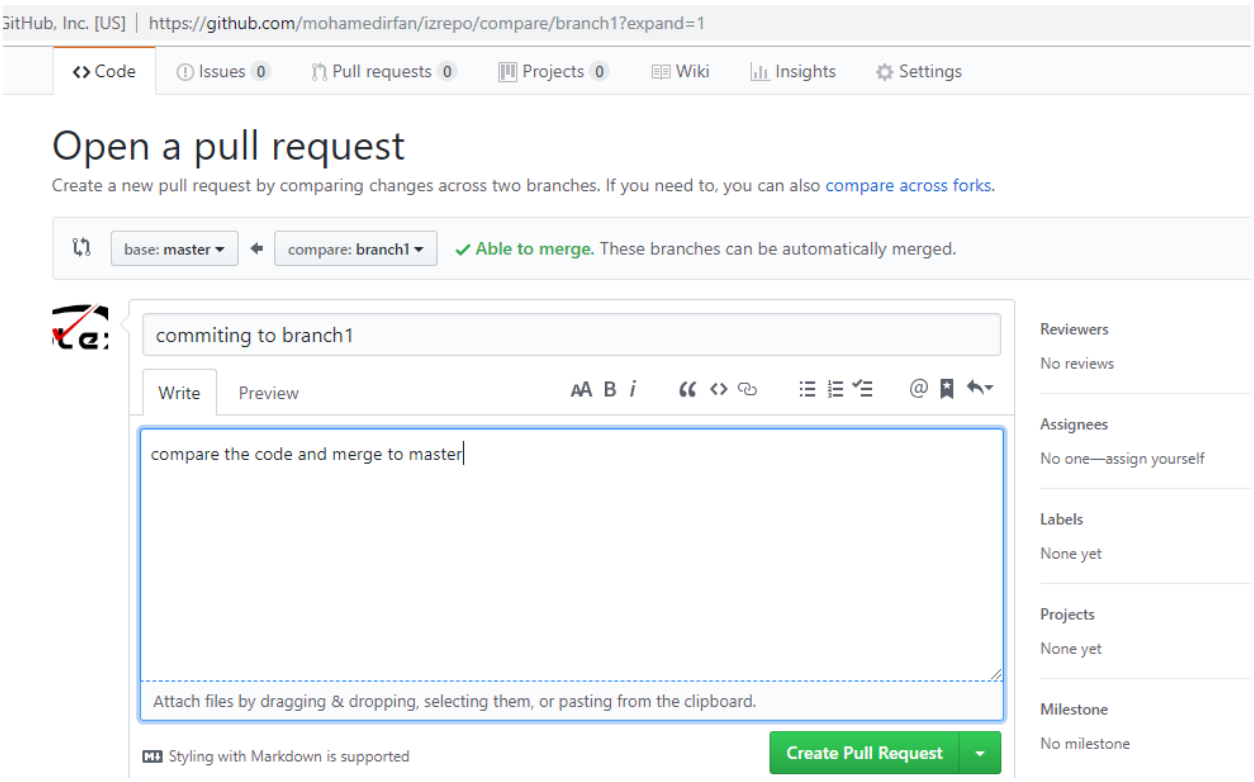

File Edit View Repository Branch Help
Current repository
izrepo
Current branch
master
Fetch origin
Last fetched 3 minutes ago
Changes 1
History
code1.txt
1 changed file
code1.txt
@@ -1 +1,2 @@
-Initial content
+Modified the initial content
+Adding line1

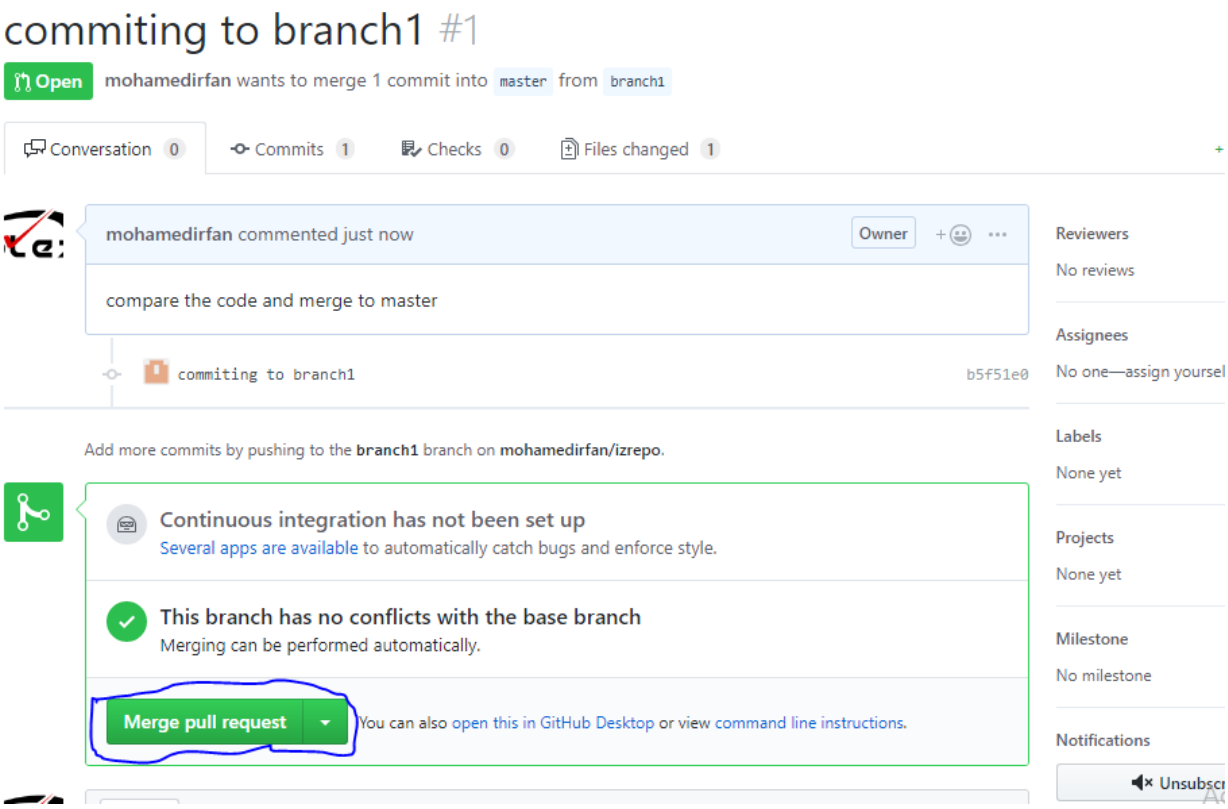

File
Edit
View
Repository
Branch
Help
Current repository
izrepo
Current branch
master
Fetch origin
Last fetched 3 minutes ago
Changes 1
History
Branches
Pull requests 0
1 changed file
code1.txt
branch1
New branch
Sorry, I can't find that branch
Do you want to create a new branch instead?
Create new branch
ProTip! Press Ctrl + Shift + N to quickly create a new branch from anywhere within the app

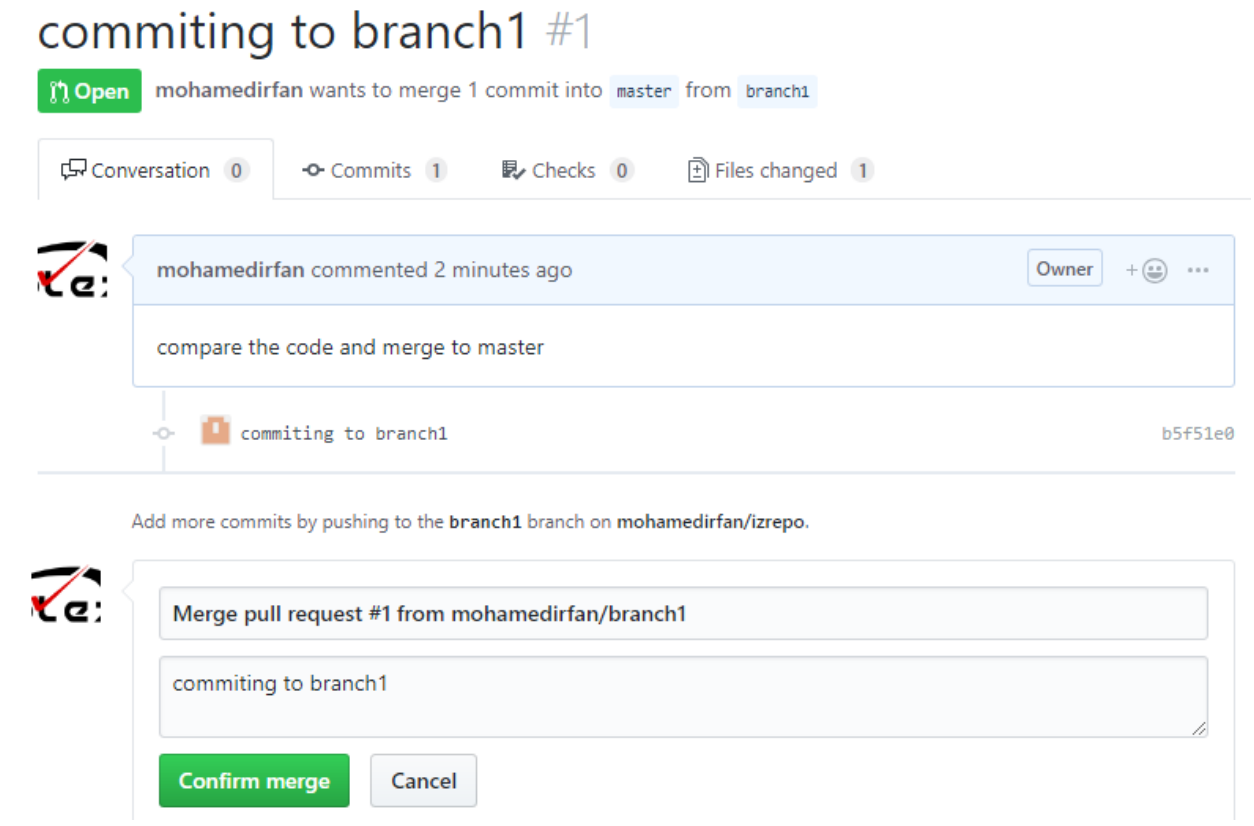

File
Edit
View
Repository
Branch
Help
Current repository
izrepo
Current branch
branch1
Publish branch
Publish this branch to GitHub
Changes 1
History
1 changed file
code1.txt
code1.txt
@@ -1 +1,2 @@
1
-Initial content
1
+Modified the initial content
2
+Adding line1
commiting to branch1
Description
Commit to branch1

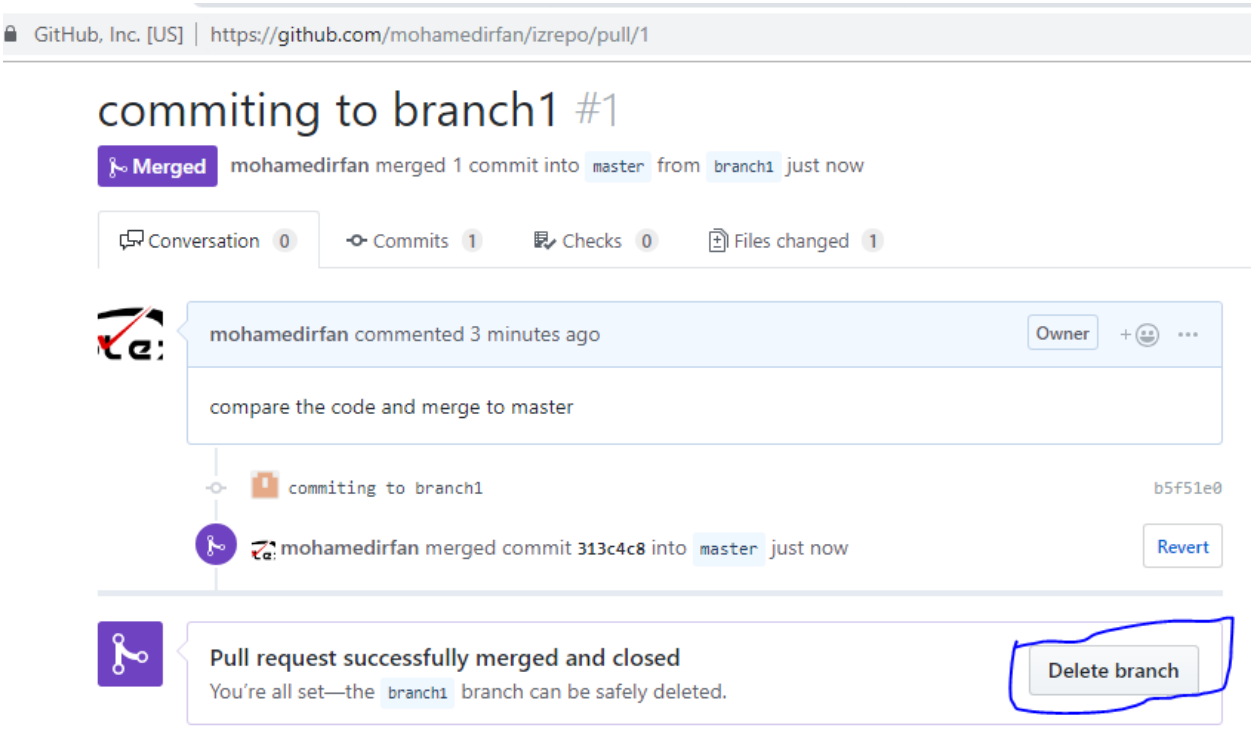

File Edit View Repository Branch Help
Current repository
izrepo
Current branch
branch1
Publish branch
Publish this branch to GitHub
Changes
History
0 changed files
No local changes
You have no uncommitted changes in your repository! Here are some friendly suggestions for what to do next.
Publish your branch
The current branch (branch1) hasn't been published to the remote yet. By publishing it to GitHub you can share it, open a pull request, and collaborate with others.
Always available in the toolbar or Ctrl P
Publish branch
View the files in your repository in Explorer
Repository menu or Ctrl Shift F
Show in Explorer
Open the repository page on GitHub in your browser
Repository menu or Ctrl Shift G
View on GitHub
Summary (required)


GitHub, Inc. [US] | https://github.com/mohamedirfan/izrepo
mohamedirfan / izrepo
Unwatch 1
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings
No description, website, or topics provided.
Edit
Manage topics
2 commits
1 branch
0 releases
1 contributor
Apache-2.0
Your recently pushed branches:
branch1 (less than a minute ago)
Compare & pull request
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
basithmohamed initial commit to staging
Latest commit c4e1461 7 minutes ago
.gitignore Initial commit 18 minutes ago
LICENSE Initial commit 18 minutes ago
README.md Initial commit 18 minutes ago
code1.txt initial commit to staging 7 minutes ago


GitHub, Inc. [US] | https://github.com/mohamedirfan/izrepo/compare/branch1?expand=1
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings
Open a pull request
Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.
base: master
compare: branch1
Able to merge. These branches can be automatically merged.
commiting to branch1
Write
Preview
compare the code and merge to master
Attach files by dragging & dropping, selecting them, or pasting from the clipboard.
Styling with Markdown is supported
Create Pull Request
Reviewers
No reviews
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone


commiting to branch1 #1
Open
mohamedirfan wants to merge 1 commit into master from branch1
Conversation 0
Commits 1
Checks 0
Files changed 1
mohamedirfan commented just now
Owner
compare the code and merge to master
commiting to branch1
b5f51e0
Add more commits by pushing to the branch1 branch on mohamedirfan/izrepo.
Continuous integration has not been set up
Several apps are available 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.
Reviewers
No reviews
Assignees
No one—assign yoursel
Labels
None yet
Projects
None yet
Milestone
No milestone
Notifications


commiting to branch1 #1
Open
mohamedirfan wants to merge 1 commit into master from branch1
Conversation 0
Commits 1
Checks 0
Files changed 1
mohamedirfan commented 2 minutes ago
Owner
compare the code and merge to master
commiting to branch1
b5f51e0
Add more commits by pushing to the branch1 branch on mohamedirfan/izrepo.
Merge pull request #1 from mohamedirfan/branch1
commiting to branch1
Confirm merge
Cancel

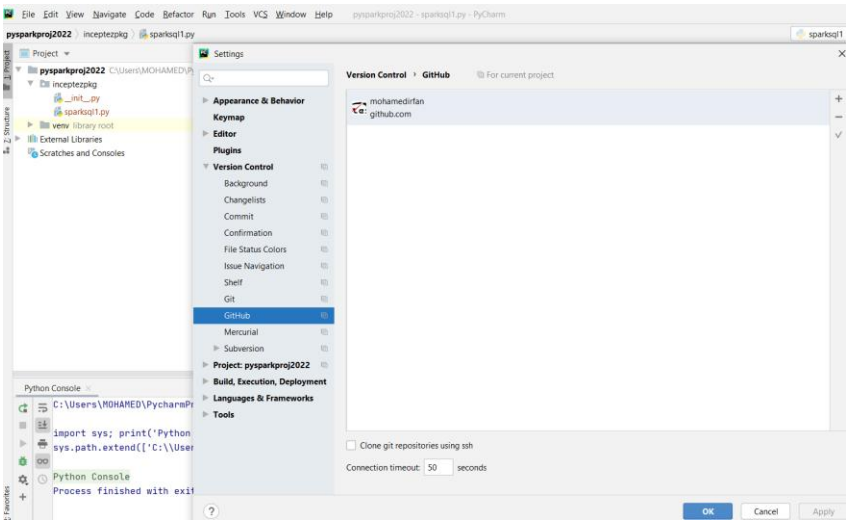

GitHub, Inc. [US] | https://github.com/mohamedirfan/izrepo/pull/1
commiting to branch1 #1
Merged
mohamedirfan merged 1 commit into master from branch1 just now
Conversation 0
Commits 1
Checks 0
Files changed 1
mohamedirfan commented 3 minutes ago
Owner
compare the code and merge to master
commiting to branch1
b5f51e0
mohamedirfan merged commit 313c4c8 into master just now
Revert
Pull request successfully merged and closed
You're all set—the branch1 branch can be safely deleted.
Delete branch

Git PyCharm Integration

1. **Login to https://github.com**
2. https://github.com/settings/tokens
3. **Generate new token (Classic) -> add some name to the note**

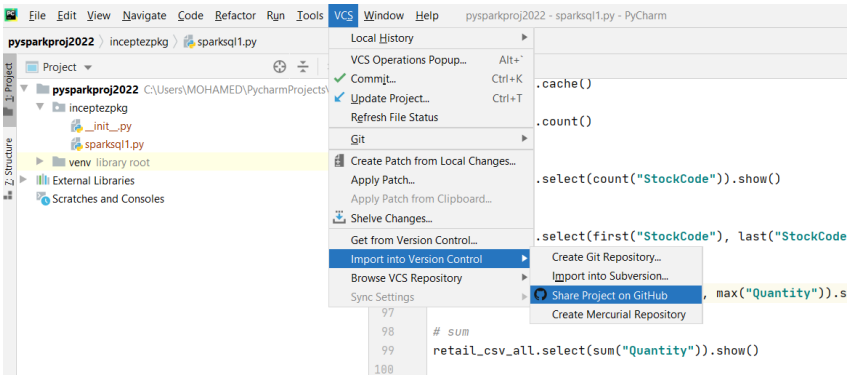
    1. **repo** - select everything
    2. **gist** - select everything
    3. **org** - select only **read:org**

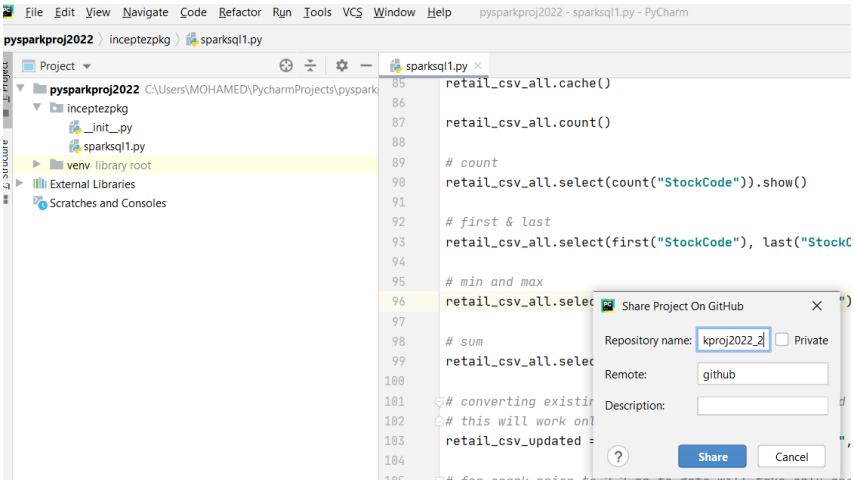
4. **Pycharm -> file -> setting -> version control -> github -> choose token option and enter the token**



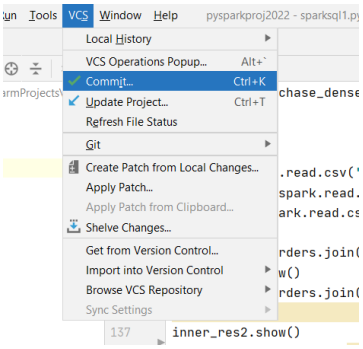
Share the project into Git:

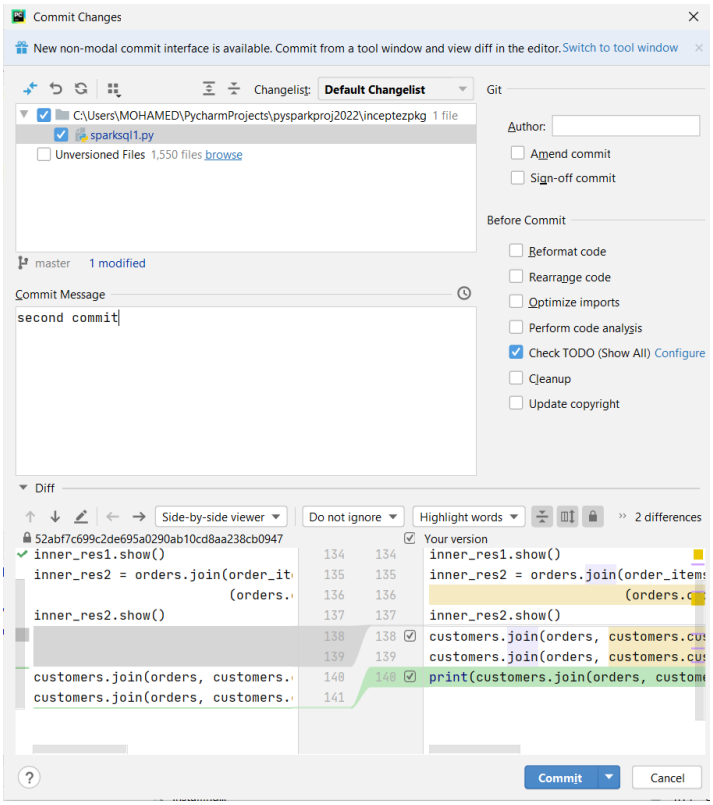
1. VCS -> Import into Version Control -> **Share Project on Github**



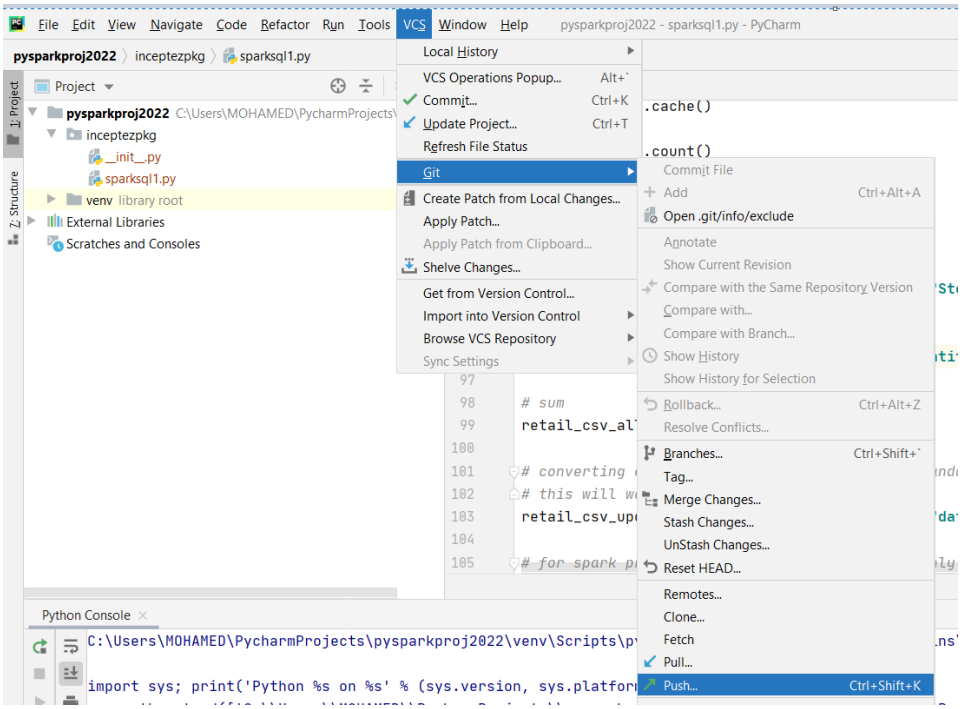


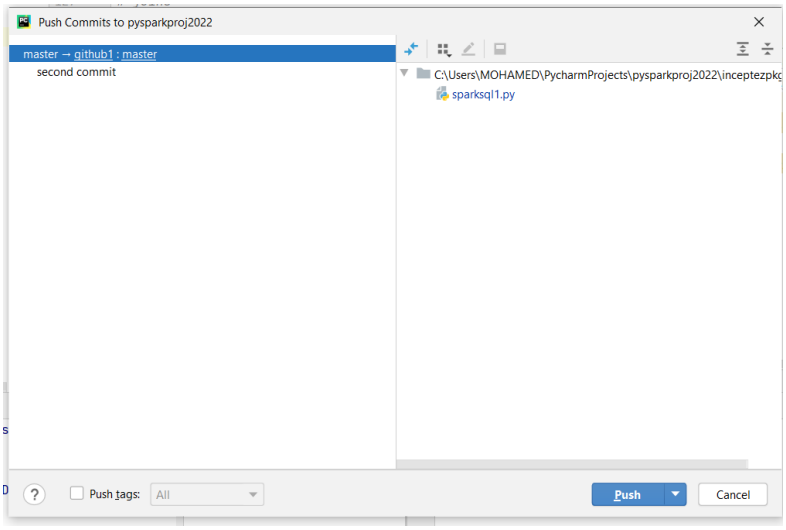
2. Do Some changes in the code -> VCS -> **execute the commit** (to commit the code to the staging area)




Commit Changes
New non-modal commit interface is available. Commit from a tool window and view diff in the editor. Switch to tool window
Changelist: Default Changelist
Git
C:\Users\MOHAMED\PycharmProjects\pysparkproj2022\inceptezpkg 1 file
sparksql1.py
Unversioned Files 1,550 files browse
Author:
Amend commit
Sign-off commit
Before Commit
master 1 modified
Commit Message
second commit
Reformat code
Rearrange code
Optimize imports
Perform code analysis
Check TODO (Show All) Configure
Cleanup
Update copyright
Diff
Side-by-side viewer
Do not ignore
Highlight words
2 differences
52abf7c699c2de695a0290ab10cd8aa238cb0947
Your version
inner_res1.show()
inner_res2 = orders.join(order_it
(orders.
inner_res2.show()
customers.join(orders, customers.
customers.join(orders, customers.
inner_res1.show()
inner_res2 = orders.join(order_item
(orders.c
inner_res2.show()
customers.join(orders, customers.cu
customers.join(orders, customers.cu
print(customers.join(orders, custom
Commit
Cancel

3. VCS -> **execute the Push** (to push the code to the internet Git repo)

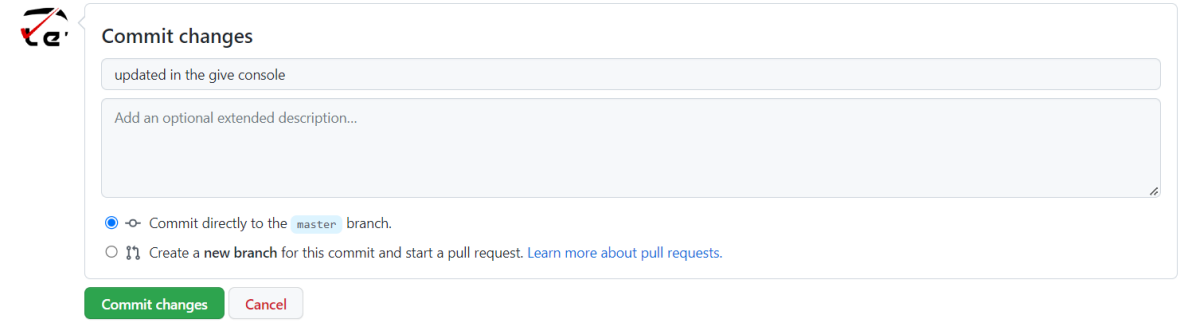




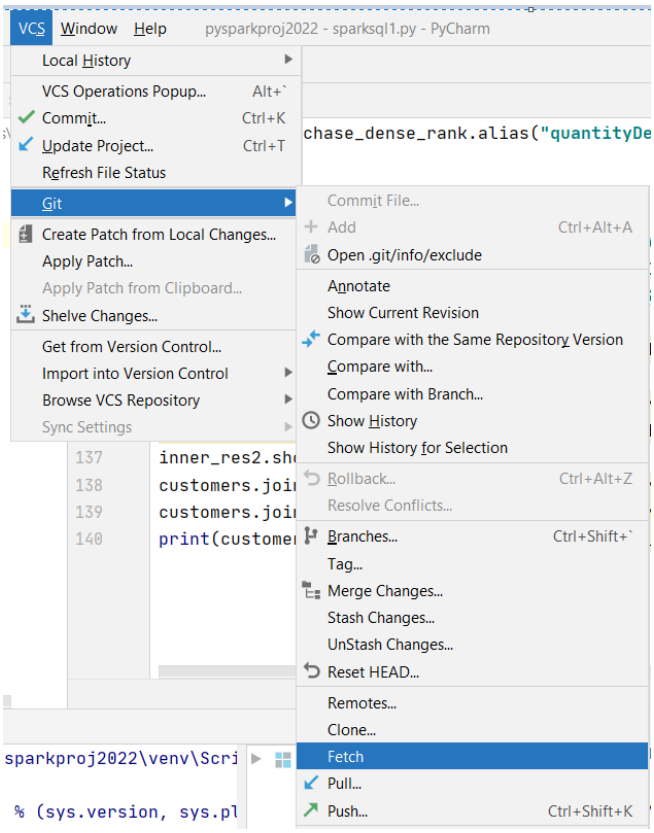
Pulling the Code from Git:

1. Edit the code in the Github console

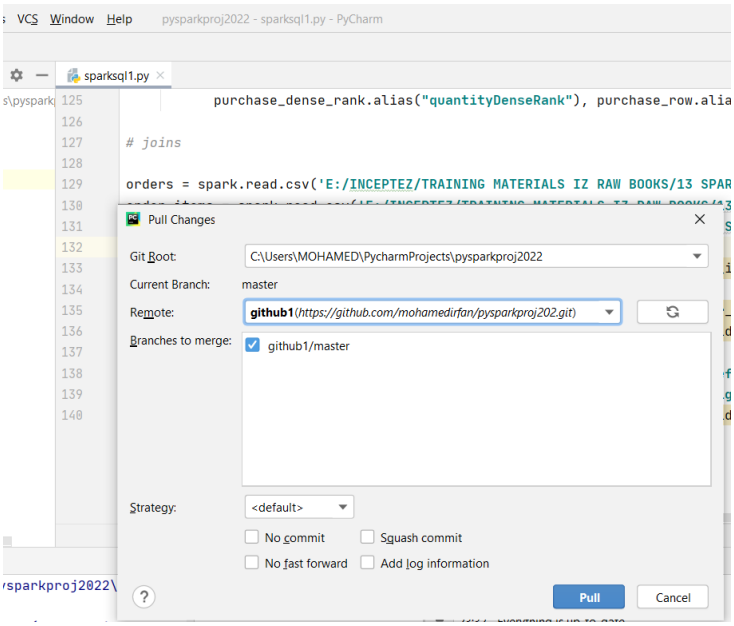
```
129  orders = spark.read.csv('E:/INCEPTEZ/TRAINING MATERIALS IZ RAW BOOKS/13 SPARK/PYSPARK/PYSPARK_2022/data/orders.csv',header=True,inferSchema=True)
130  order_items = spark.read.csv('E:/INCEPTEZ/TRAINING MATERIALS IZ RAW BOOKS/13 SPARK/PYSPARK/PYSPARK_2022/data/order_items.csv',header=True,inferSchema=True)
131  customers = spark.read.csv('E:/INCEPTEZ/TRAINING MATERIALS IZ RAW BOOKS/13 SPARK/PYSPARK/PYSPARK_2022/data/customers.csv',header=True,inferSchema=True)
132
133  inner_res1 = orders.join(order_items, orders.order_id == order_items.order_item_order_id)
134  inner_res1.show()
135  inner_res2 = orders.join(order_items, (orders.order_id == order_items.order_item_order_id) &
136                           (orders.order_id == order_items.order_item_order_id))
137  inner_res2.show()
138  customers.join(orders, customers.customer_id==orders.order_customer_id, 'left').show()
139  customers.join(orders, customers.customer_id==orders.order_customer_id, 'right').show()
140  print(customers.join(orders, customers.customer_id==orders.order_customer_id, 'left').count())
141  print(customers.join(orders, customers.customer_id==orders.order_customer_id, 'right').count())
```



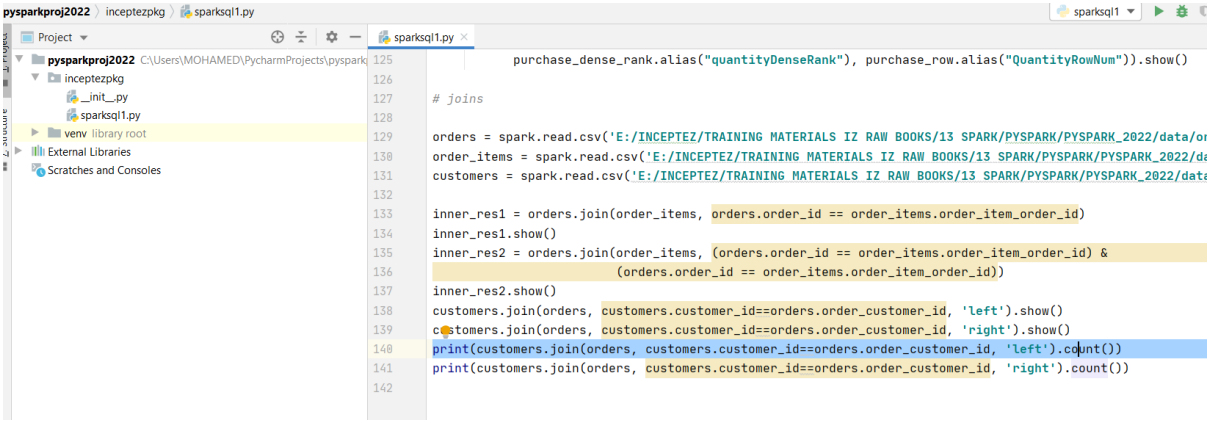
2. VCS -> Git -> **Fetch**(to fetch the code from the Git)



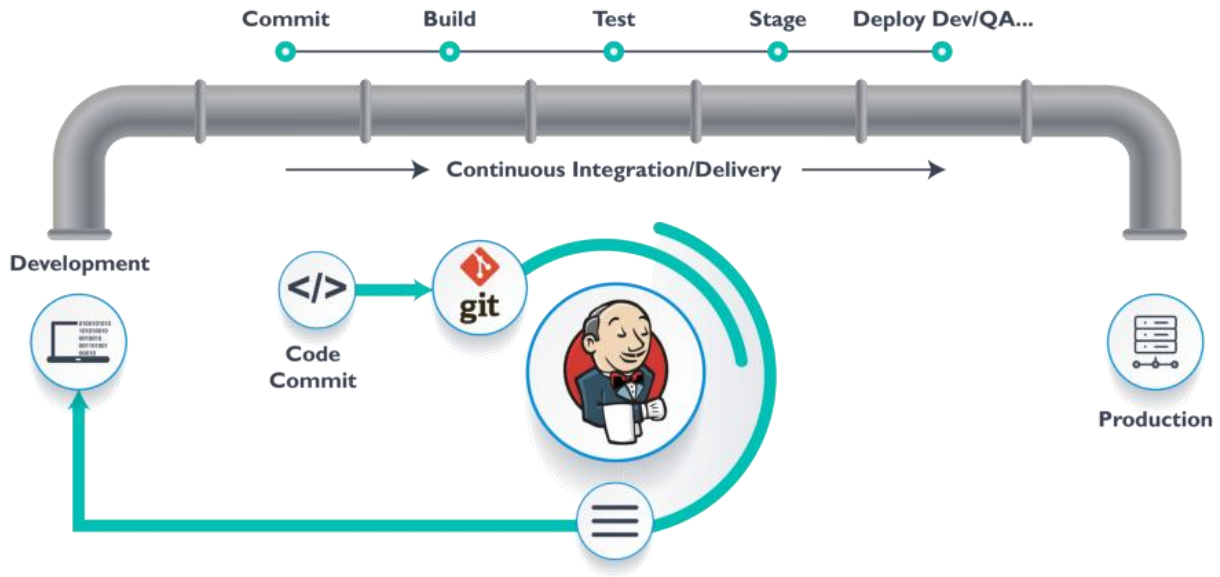
3. VCS -> Git -> **Pull**(to Pull the code from the Git)



4. See the modified code in the Pycharm



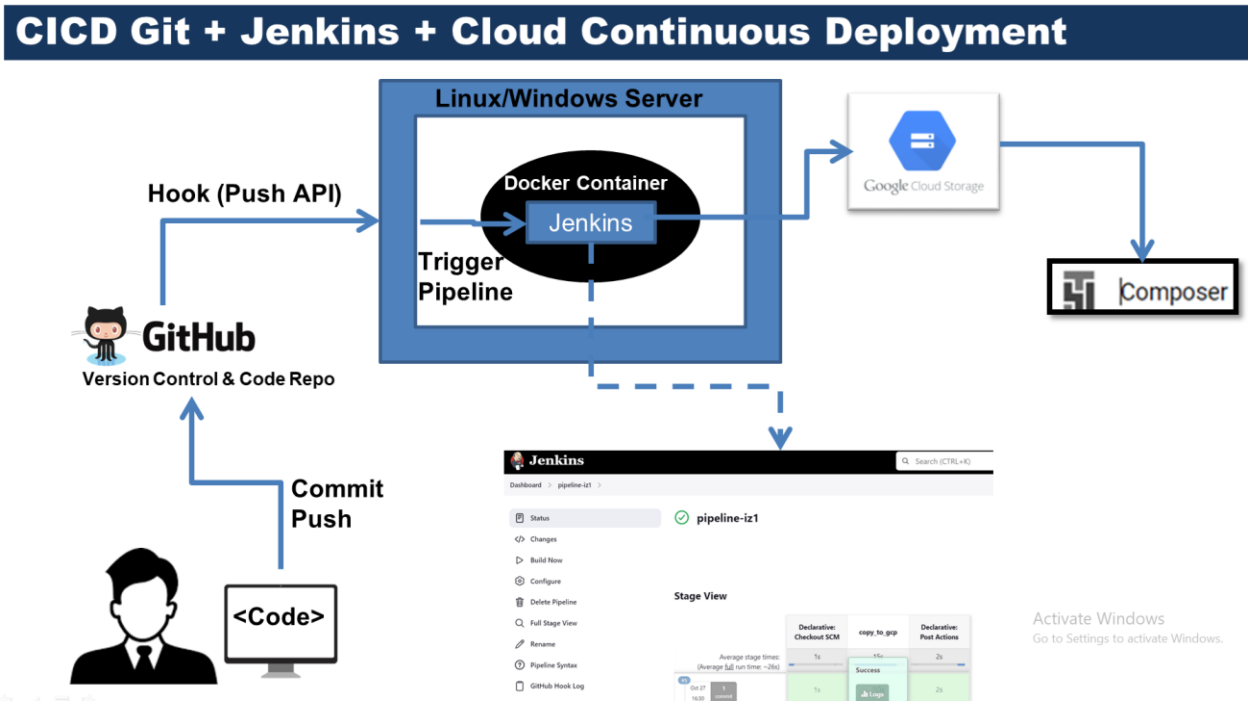
Jenkins



Download Jenkins and Java 11

https://www.jenkins.io/download/thank-you-downloading-windows-installer-stable

https://www.techspot.com/downloads/downloadnow/5553/?evp=cc1002b8f24231deaf82043061a06561&file=6310



Check these documents for installing Jenkins and configure:

Installing and Configuring Jenkins on Windows with Git and GCS Deployment

Step 1: Install Jenkins on Windows

1. Download Jenkins from the official site: https://www.jenkins.io/download/ (choose the Windows MSI installer). https://www.jenkins.io/download/thank-you-downloading-windows-installer-stable/

2. Run the installer (jenkins.msi) and install Jenkins as a Windows service.
3. Open Jenkins in the browser at http://localhost:8080
4. Unlock Jenkins using the initial password stored at:
   C:\Program Files\Jenkins\secrets\initialAdminPassword
5. Install recommended plugins and create an admin user. (To log in, use the username: "admin" and the administrator password you used to access the setup wizard.)

Step 2: Install Required Plugins

Go to: Manage Jenkins → Plugins → Available Plugins and install:
- Git Plugin
- Google Cloud Storage Plugin
- Pipeline Plugin

Step 3: Configure Git in Jenkins

1. Install Git on Windows from: https://git-scm.com/download/win
   2. In Jenkins → Manage Jenkins → Global Tool Configuration → Git → Add Git installation.
2. ngrok config add-authtoken 33ITQCgKImLF4nbdvsTw7V63HMt_5aPX29cVDxqB3FesbAozp

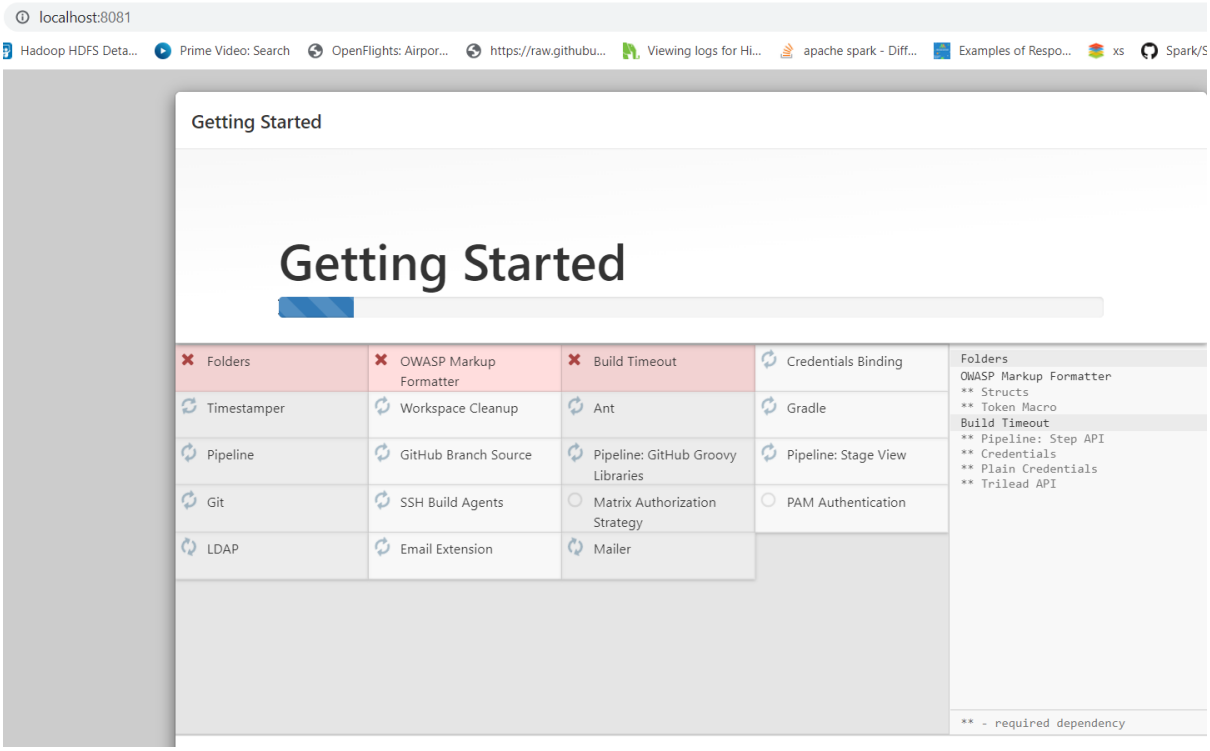
Step 4: Configure GCP Credentials in Jenkins

1. Install Google Cloud SDK on Windows: https://cloud.google.com/sdk/docs/install
2. Authenticate with GCP:
   gcloud auth login
   gcloud auth application-default login
3. Create a Service Account in GCP with Storage Admin role and download the JSON key.
4. In Jenkins: Manage Jenkins → Credentials → Global → Add Credentials → Secret file → Upload JSON key.

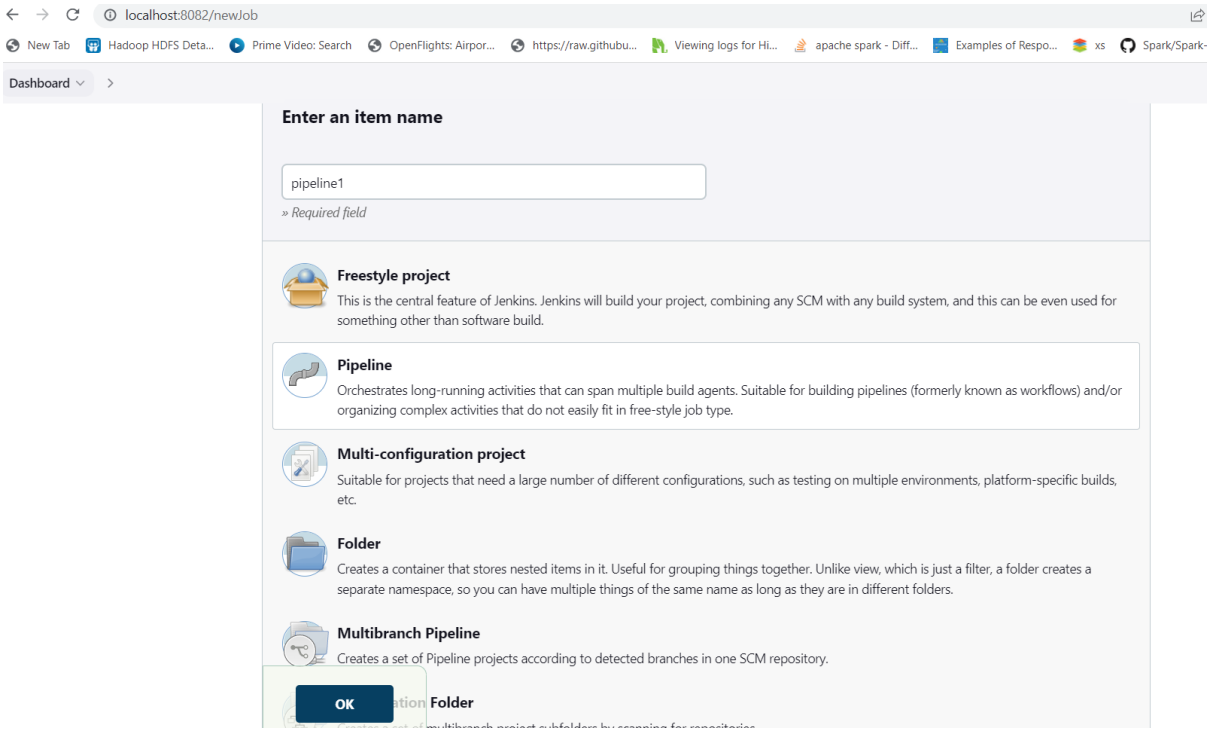
Get the initial login password of Jenkins:

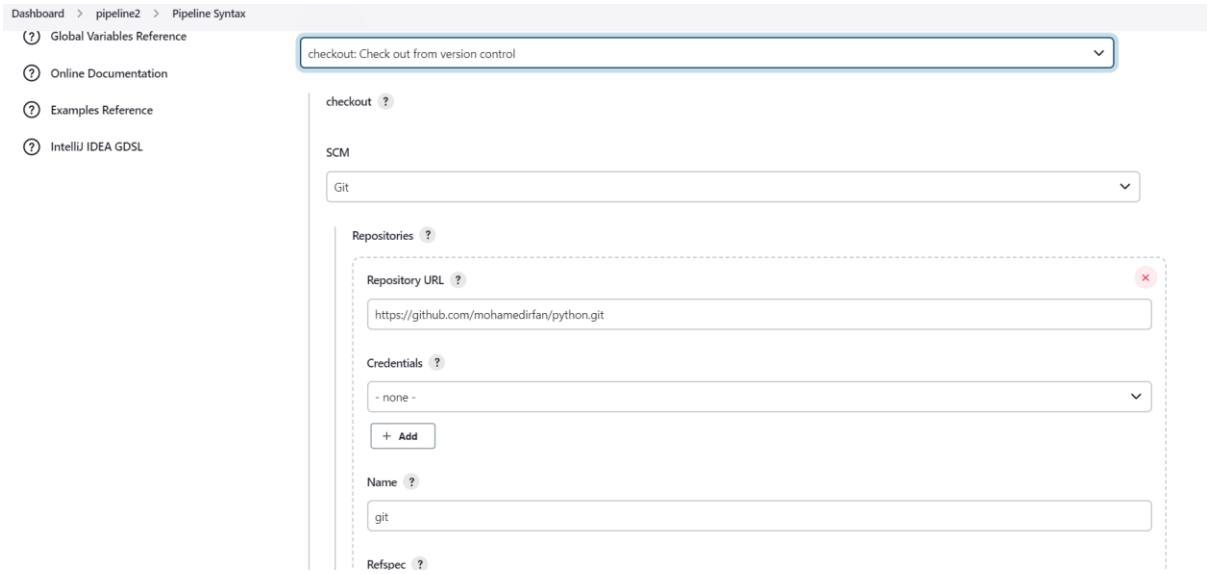
C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword

1. After installation -> Open chrome browser -> localhost:8082
2. Select (Install suggested plugins)



Create Pipeline:


localhost:8082/newJob
New Tab
Hadoop HDFS Deta...
Prime Video: Search
OpenFlights: Airpor...
https://raw.githubu...
Viewing logs for Hi...
apache spark - Diff...
Examples of Respo...
xs
Spark/Spark-
Dashboard
Enter an item name
pipeline1
» Required field
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
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.
OK
Folder


Dashboard
pipeline2
Pipeline Syntax
Global Variables Reference
Online Documentation
Examples Reference
IntelliJ IDEA GDSL
checkout: Check out from version control
checkout
SCM
Git
Repositories
Repository URL
https://github.com/mohamedirfan/python.git
Credentials
- none -
Add
Name
git
Refspec

Steps

Sample Step

bat: Windows Batch Script

bat

Batch Script ?

```
python codewewd.py
```

Advanced...

Generate Pipeline Script

```
bat 'python codewewd.py'
```

Jenkins

Search (C

Dashboard > pipeline2 >

- Status
- Changes
- Build Now
- Configure
- Delete Pipeline
- Full Stage View
- Rename
- Pipeline Syntax

Build History trend

Atom feed for all Atom feed for failures

Pipeline pipeline2

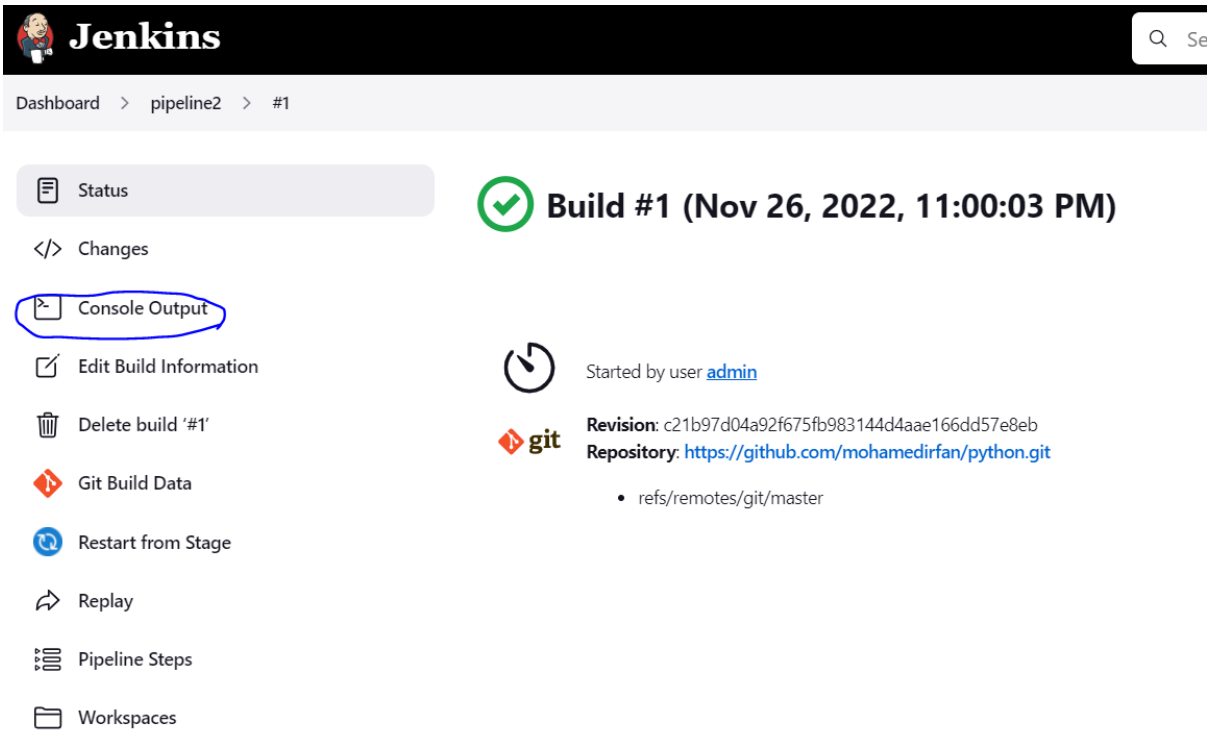
python script build and deploy

Stage View

	checkout	build
Average stage times: (Average full run time: ~12s)	4s	2s
#1 Nov 26 23:00 No Changes	4s	2s

Permalinks

- Last build (#1), 5 min 40 sec ago
- Last stable build (#1), 5 min 40 sec ago
- Last successful build (#1), 5 min 40 sec ago
- Last completed build (#1), 5 min 40 sec ago




```
pipeline {
  agent any

  stages {
    stage('checkout') {
      steps {
        echo "checkout is happening"
        checkout([$class: 'GitSCM', branches: [[name: '*/master']], extensions: [], userRemoteConfigs: [[name: 'git', url: 'https://github.com/mohamedirfan/python.git']]])
      }
    }
    stage('build') {
      steps {
        echo "build is happening"
        git 'https://github.com/mohamedirfan/python.git'
      }
    }
    stage('deploy') {
      steps {
        echo "deploy is happening"
        bat 'python codewewd.py'
      }
    }
```

```
  }
}
```

Github CLI Commands:

1. Install Git in Linux:

sudo yum install git

2. Configure Git:

git config --global user.name "inceptez"

git config --global user.email "info@inceptez.com"

git config --list

3. Clone an existing repository from the Git:

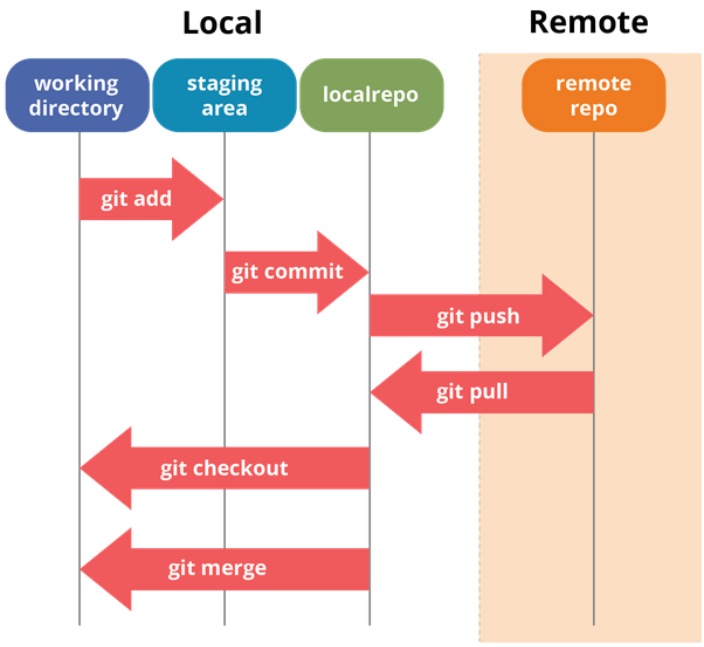
git clone https://github.com/incepteztechnologies/de2025.git

cd de2025

4. Set the existing remote repository URL

git remote set-url origin https://github.com/incepteztechnologies/de2025.git

5. Adding your code to Github



local code -> add -> stage -> commit -> local git repo

Create your code and place it inside the firstrepo folder

echo hello > ~/de2025/code1

Add the local file to the Stage

git add ~/de2025/code1.py

Commit the staged file to the local git repo

git commit -m "inital commit"

Push the local git file to the remote git

https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token

git push -u origin **main** -f

Authentication Key –

~~Username for 'https://github.com': mohamedirfan@gmail.com~~

~~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~~

~~**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**~~

Shows the current status of anything to commit

git status

Latest log of all activities happened in Git

git log

To know the remote repository

git remote -v

6. <u>Pulling the code from Github</u>

Goto github.com -> modify the file3

To pull all the changes from the remote to local directory only, not to my working directory.

git fetch

we dont see any changes until merge

cat /c/Users/moham/de2025/code1.py

Update the files from local to working directory

git merge origin/main

We can see changes now

cat /c/Users/moham/de2025/code1

Pull command basically runs 2 command, git fetch and git merge(first fetch to the local repo, do compare the code with the working directory then merge the local code with the working code).

git pull

List the files from Git Repo

git ls-files

Few Additional Commands:

Create a branch1

git branch branchwewd

git pull

switch to a different branch

git checkout branchwewd

To see which branch we are in

git branch

Checkout different versions

git checkout b9aed8bbeabaa41d5bc4a0b3b40f8d3863b0b727