

# Cloud Computing

Setup the repository for app, 建立應用程式的資源庫.

Requirements

---

1. [git \(https://git-scm.com/\)](https://git-scm.com/) tool for version control system, 版本管理系統;
2. [github \(https://github.com/\)](https://github.com/) site of repository stored; register an account , 儲存資源庫的網站 , 先註冊帳號
3. [smartgit \(https://www.syntevo.com/smartgit/\)](https://www.syntevo.com/smartgit/) gui git client app, 圖形化 git 應用程式,

In [1]:

```
1 from IPython.display import Image
```

```
In [4]: ▼ 1 # Create a new repository on GitHub, for instance 2023-test
2 # 首先在 GitHub 上面新增一個資源庫, 叫做 2023-test
3
4 Image("github/create_repo.png",width=1000)
```

Out[4]:

Welcome to Brave

Settings - About Brave

IMathAS-2022

Create a New Repository

github.com/new

Create a new repository

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

Owner \*

Repository name \*

cchuang2009


/

2023-test


Great repository names are short and memorable. Need inspiration? How about **refactored-spork**?

Description (optional)

How to establish connection between local box and github official site

☒  **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:

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

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

.gitignore template: None

streamlit-plotl....zip

smartgit-lin....tar.gz

Show all

```
In [6]: ▼ 1 # 2. after created repo, it should tell you how to go next, 接著 github 會告訴你如何進行下一步  
2  
3 Image("github/init_repol.png",width=1000)  
4
```

Out[6]:

The screenshot shows a web browser window with the GitHub repository page for `cchuang2009/2023-test`. The browser's address bar shows the URL `github.com/cchuang2009/2023-test`. The repository page has a dark header with the GitHub logo, a search bar, and navigation links: `Pull requests`, `Issues`, `Codespaces`, `Marketplace`, and `Explore`. Below the header, the repository name `cchuang2009 / 2023-test` is displayed, along with buttons for `Pin`, `Unwatch` (1), `Fork` (0), and `Star` (0). A secondary navigation bar includes links for `Code`, `Issues`, `Pull requests`, `Actions`, `Projects`, `Wiki`, `Security`, `Insights`, and `Settings`. The main content area features a light blue box with the heading **Quick setup — if you've done this kind of thing before**. It provides two options: `HTTPS` and `SSH`, with the `HTTPS` option selected. The corresponding URL `https://github.com/cchuang2009/2023-test.git` is shown in a text box with a copy icon. Below this, a message states: "Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#)." Further down, the section **...or create a new repository on the command line** is shown, with a code block containing the following commands: 

```
echo "# 2023-test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/cchuang2009/2023-test.git
git push -u origin main
```

 A copy icon is present next to the code block. Below this, the section **...or push an existing repository from the command line** is shown, with a code block containing the command: 

```
git remote add origin https://github.com/cchuang2009/2023-test.git
```

 A copy icon is also present next to this code block. At the bottom of the page, there is a file upload bar showing two files: `streamlit-plotl....zip` and `smartgit-lin....tar.gz`, each with an upload icon. A `Show all` button and a close icon are on the right.

Now we have to create the directory for repository, 接下來我們將新增貯存資源庫的目錄.

1. execute terminal app , (press "Windows + R" and input `cmd` for MS Windows, 執行終端機程式，如果使用微軟 Windows 系統使用組合鍵 "Windows + R"，並輸入 `cmd` 啟動.
2. create directory, named "2023-test", and enter to the directory, 新增目錄，"2023-test"，並進入該目錄.
3. To establish the connection between Github and local box, we have to generate "api token". Login `github` , and choose, 為了建立自己的電腦系統和 GitHub 的連結，我們必須新增 "api token"，進入網站 Github:

[Setting (Top right)] → [Developer setting (left bottom)]



[Personal access tokens (left column)] → [Fine-grained tokens] → [Create new token]

to create new token ,named `githubapi` for instance, 由上面的步驟產生新的 token，並叫做 `githubapi` .

4. Setup this api token for accessing, 設定相關事項,
  - Repository access : ● **All repositories**, 資源庫的存取, 選取全部.
  - Permissions : enable all the necessary options. If don't know, just enable all options. Late, you can update the setting if not work, 准許必要的服務選項，如果不知道怎麼選取，可以全部選取.如果仍然不能夠運作，還可以修正.
5. establish the connection between local box and Github as the last picture, 如上面最後一個圖示，建立電腦和 Github 之間的連結:

```
echo "# 3023-test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/GITHUB_USER_ACCOUNT/2023-test.git
git push -u origin main
```

Replace `GITHUB_USER_ACCOUNT` in line 6 above of **your account** in Github, 使用 Github 上面的帳號替換上述第六行的 `GITHUB_USER_ACCOUNT` . You can also modify `README.md` , site introduction displayed as enter, by markdown format, 你也可以使用 Markdown 語法修改 `README.md`，進入此網頁時的介紹.

In [7]: 1 Image("github/auth.png",width=1000)

Out[7]:

Welcome to Brave | Settings - About Brave | IMathAS-2022 | cchuang2009/2023-test | About remote repositories x

docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning...

GitHub Docs | Version: Free, Pro, & Team | Search GitHub Docs

Get started / Getting started with Git / About remote repositories

## Cloning with HTTPS URLs

The `https://` clone URLs are available on all repositories, regardless of visibility. `https://` clone URLs work even if you are behind a firewall or proxy.

When you `git clone`, `git fetch`, `git pull`, or `git push` to a remote repository using HTTPS URLs on the command line, Git will ask for your GitHub username and password. When Git prompts you for your password, enter your personal access token. Alternatively, you can use a credential helper like [Git Credential Manager](#). Password-based authentication for Git has been removed in favor of more secure authentication methods. For more information, see "[Creating a personal access token](#)."

If you are accessing an organization that uses SAML SSO and you are using a personal access token (classic), you must also authorize your personal access token to access the organization before you authenticate. For more information, see "[About authentication with SAML single sign-on](#)" and "[Authorizing a personal access token for use with SAML single sign-on](#)."

**Tips:**

- You can use a credential helper so Git will remember your GitHub credentials every time it talks to GitHub. For more information, see "[Caching your GitHub credentials in Git](#)."
- To clone a repository without authenticating to GitHub on the command line, you can use GitHub Desktop

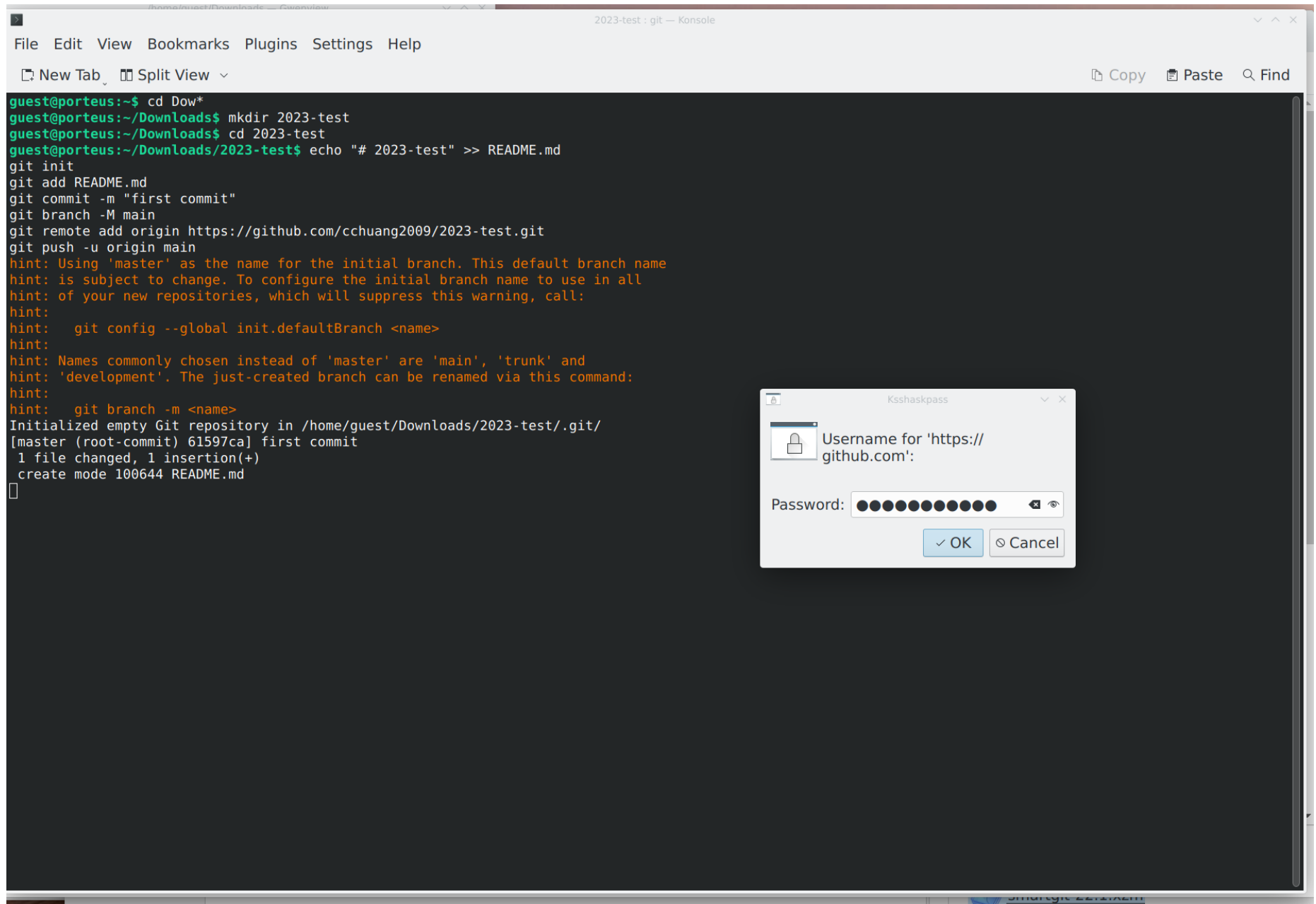
streamlit-plotl....zip ^ | smartgit-lin....tar.gz ^ | Show all x

6. While initializing the repository, 2023-test , run the shell commends above in directory, 2023-test : input the username in popup windows, the part before @gmail.com if use gmail account to register; and input **api token** created above in passwords input ( but not the passwords used for github.com). 在自己的電腦上面初始化資源庫， 2023-test ，在目錄中執行下列指令：並在彈跳的視窗中輸入使用者帳號，如果是用 gmail 設定 github 帳號, 輸入 @gmail.com 之前的使用者名稱即可，在輸入密碼的時候，輸入剛剛產生的 apitoken . 而不是輸入進入 github.com 的密碼



In [11]: 1 Image("github/init\_repo2.png",width=1000)

Out[11]:



6. Now startup `smartgit` to take over the left work of software developing by github: push, pull etc. Something like, 啟動取代文字介面的軟體開發，上傳或下載工作檔案目錄資料，就像下面的工作一樣:

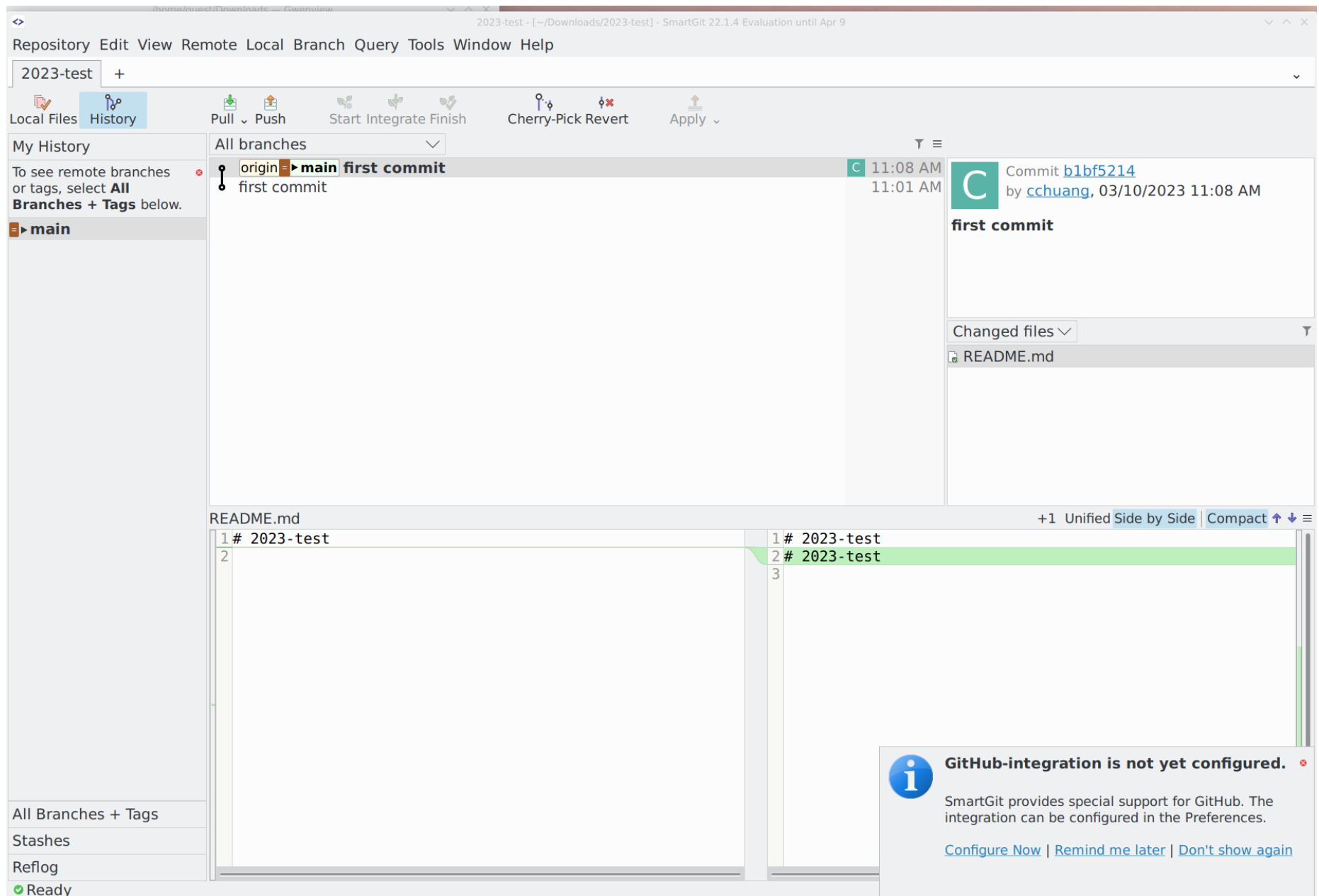
`copy files in the directory`, 將新增檔案放入目錄內

`[stage] → describe the change in the right region`, 工作重點描述, and `[commit] → [Push]`

Awhile, all the software maintenance will rely on such solution, 如此一來，所有的工作將藉由這個架構維護.

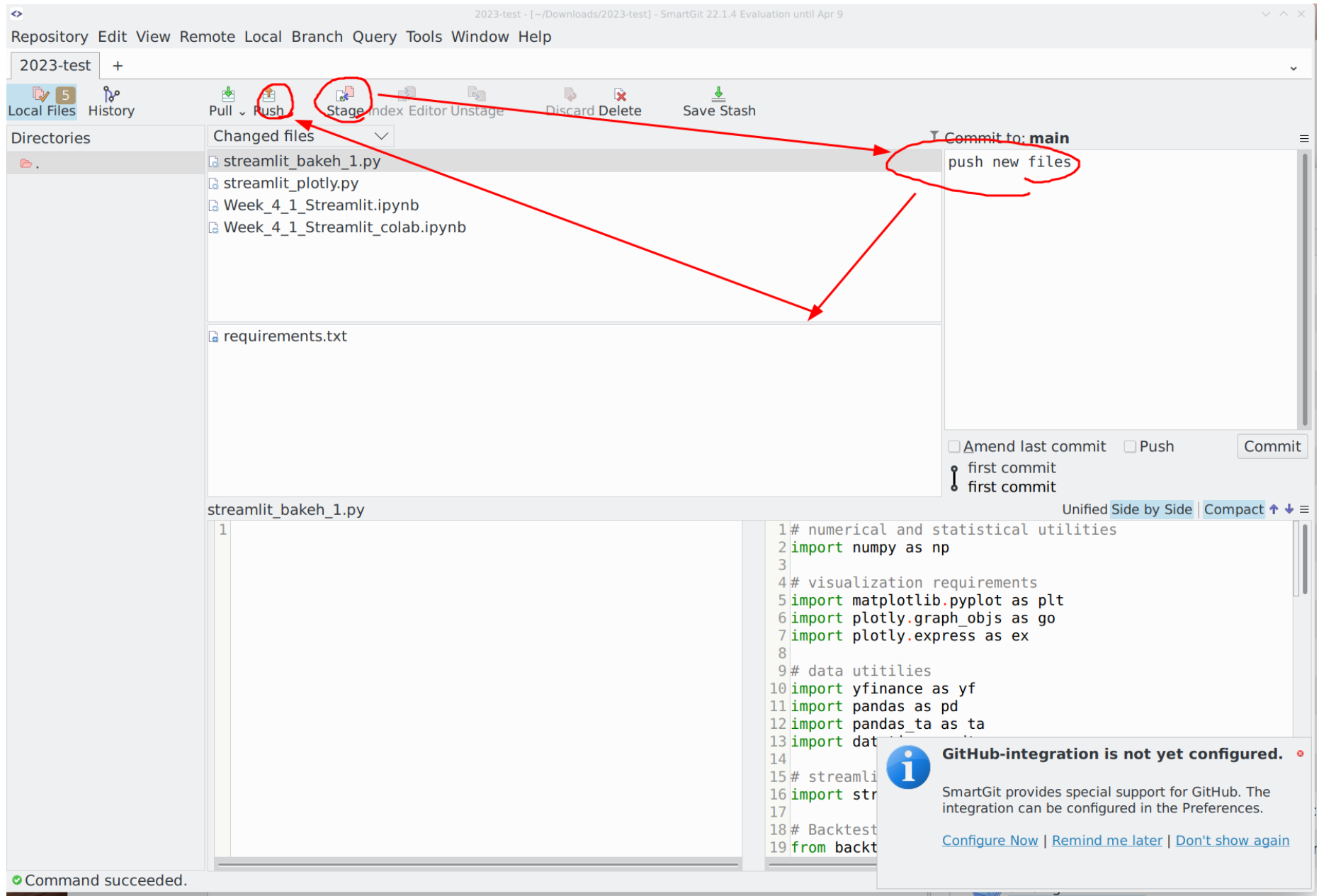
In [15]: 1 Image("github/smartgit-1.png",width=1000)

Out[15]:



In [14]: 1 Image("github/smartgit-2.png",width=1000)

Out[14]:



```
In [13]: 1 Image("github/push.png",width=1000)
```

Out[13]:

The screenshot shows a web browser displaying the GitHub repository page for `cchuang2009/2023-test`. The repository is public and has 1 branch (main) and 0 tags. The file list shows the following files and their commit history:

File	Commit	Time
README.md	first commit	32 minutes ago
Week_4_1_Streamlit.ipynb	push new files	now
Week_4_1_Streamlit_colab.ip...	push new files	now
requirements.txt	push new files	6 minutes ago
streamlit_bakeh_1.py	push new files	1 minute ago
streamlit_plotly.py	push new files	now

The repository description is "How to establish connection between local box and github official site". The repository has 0 stars, 1 watching, and 0 forks. The repository is currently empty, showing only the README.md file. The repository name is **2023-test**.

The right sidebar shows the repository's metadata:

- About:** How to establish connection between local box and github official site
- Readme:** 0 stars, 1 watching, 0 forks
- Releases:** No releases published. [Create a new release](#)
- Packages:** No packages published. [Publish your first package](#)

The bottom of the page shows a file upload bar with two files: `streamlit-plotl....zip` and `smartgit-lin....tar.gz`. A "Show all" button is visible on the right.

```
In [25]: 1 !pip install -q pydrive
```

1. [Google Cloud console \(https://console.cloud.google.com/\)](https://console.cloud.google.com/) Login

≡ Google Cloud (Top Left) Select project/Create New Project, named 2023-2-python)



API APIs and Services (Left) → [Library]



[Google Workplaces] → [Library] → [Google Drive API] ENABLE  
→ [credential] → [Google Drive API] ENABLE

Create Client ID credential

In [1]:

```
1 from pydrive.auth import GoogleAuth
2 from pydrive.drive import GoogleDrive
3
4 gauth = GoogleAuth()
5 gauth.LocalWebserverAuth() # client_secrets.json need to be in the same directory as the script
6 drive = GoogleDrive(gauth)
```

```

-----
JSONDecodeError                                Traceback (most recent call last)
Cell In[1], line 5
      2 from pydrive.drive import GoogleDrive
      4 gauth = GoogleAuth()
----> 5 gauth.LocalWebserverAuth() # client_secrets.json need to be in the same directory as the script
      6 drive = GoogleDrive(gauth)

File ~/opt/anaconda3/lib/python3.8/site-packages/pydrive/auth.py:113, in CheckAuth.<locals>._decorated(self,
*args, **kwargs)
    111 self.LoadCredentials()
    112 if self.flow is None:
--> 113 self.GetFlow()
    114 if self.credentials is None:
    115 code = decoratee(self, *args, **kwargs)

File ~/opt/anaconda3/lib/python3.8/site-packages/pydrive/auth.py:443, in GoogleAuth.GetFlow(self)
    437 """Gets Flow object from client configuration.
    438
    439 :raises: InvalidConfigError
    440 """
    441 if not all(config in self.client_config \
    442             for config in self.CLIENT_CONFIGS_LIST):
--> 443 self.LoadClientConfig()
    444 constructor_kwargs = {
    445     'redirect_uri': self.client_config['redirect_uri'],
    446     'auth_uri': self.client_config['auth_uri'],
    447     'token_uri': self.client_config['token_uri'],
    448 }
    449 if self.client_config['revoke_uri'] is not None:

File ~/opt/anaconda3/lib/python3.8/site-packages/pydrive/auth.py:366, in GoogleAuth.LoadClientConfig(self, ba
ckend)
    364 raise InvalidConfigError('Please specify client config backend')
    365 if backend == 'file':
--> 366 self.LoadClientConfigFile()
    367 elif backend == 'settings':
    368 self.LoadClientConfigSettings()

File ~/opt/anaconda3/lib/python3.8/site-packages/pydrive/auth.py:386, in GoogleAuth.LoadClientConfigFile(sel
f, client_config_file)

```



```

384 client_config_file = self.settings['client_config_file']
385 try:
--> 386 client_type, client_info = clientsecrets.loadfile(client_config_file)
387 except clientsecrets.InvalidClientSecretsError as error:
388     raise InvalidConfigError('Invalid client secrets file %s' % error)

```

File ~/opt/anaconda3/lib/python3.8/site-packages/oauth2client/clientsecrets.py:165, in loadfile(filename, cache)

```

162 _SECRET_NAMESPACE = 'oauth2client:secrets#ns'
164 if not cache:
--> 165     return _loadfile(filename)
167 obj = cache.get(filename, namespace=_SECRET_NAMESPACE)
168 if obj is None:

```

File ~/opt/anaconda3/lib/python3.8/site-packages/oauth2client/clientsecrets.py:122, in \_loadfile(filename)

```

120 try:
121     with open(filename, 'r') as fp:
--> 122         obj = json.load(fp)
123 except IOError as exc:
124     raise InvalidClientSecretsError('Error opening file', exc.filename,
125                                     exc.strerror, exc.errno)

```

File ~/opt/anaconda3/lib/python3.8/json/\_\_init\_\_.py:293, in load(fp, cls, object\_hook, parse\_float, parse\_int, parse\_constant, object\_pairs\_hook, \*\*kw)

```

274 def load(fp, *, cls=None, object_hook=None, parse_float=None,
275          parse_int=None, parse_constant=None, object_pairs_hook=None, **kw):
276     """Deserialize ``fp`` (a ``.read()``-supporting file-like object containing
277     a JSON document) to a Python object.
278
279     (...)
291     kwarg; otherwise ``JSONDecoder`` is used.
292     """
--> 293     return loads(fp.read(),
294                  cls=cls, object_hook=object_hook,
295                  parse_float=parse_float, parse_int=parse_int,
296                  parse_constant=parse_constant, object_pairs_hook=object_pairs_hook, **kw)

```

File ~/opt/anaconda3/lib/python3.8/json/\_\_init\_\_.py:357, in loads(s, cls, object\_hook, parse\_float, parse\_int, parse\_constant, object\_pairs\_hook, \*\*kw)

```

352     del kw['encoding']
354 if (cls is None and object_hook is None and
355     parse_int is None and parse_float is None and

```

```

356         parse_constant is None and object_pairs_hook is None and not kw):
--> 357     return _default_decoder.decode(s)
358 if cls is None:
359     cls = JSONDecoder

```

File ~/opt/anaconda3/lib/python3.8/json/decoder.py:337, in JSONDecoder.decode(self, s, \_w)

```

332 def decode(self, s, _w=WHITESPACE.match):
333     """Return the Python representation of ``s`` (a ``str`` instance
334     containing a JSON document).
335
336     """
--> 337     obj, end = self.raw_decode(s, idx=_w(s, 0).end())
338     end = _w(s, end).end()
339     if end != len(s):

```

File ~/opt/anaconda3/lib/python3.8/json/decoder.py:355, in JSONDecoder.raw\_decode(self, s, idx)

```

353     obj, end = self.scan_once(s, idx)
354 except StopIteration as err:
--> 355     raise JSONDecodeError("Expecting value", s, err.value) from None
356 return obj, end

```

JSONDecodeError: Expecting value: line 1 column 1 (char 0)

In [ ]:

1