

Organized Steps -> creating git repository and indicating over the mail to teammates whenever any new change added to that repository:

Here's a structured procedure you can follow based on your command prompt output and our previous discussions:

### ### Setting Up Git Locally:

1. Open your terminal or command prompt.
2. Navigate to the directory containing your Python script.
3. Initialize a Git repository:

```
```bash
git init
```
```

4. Add your files to the repository:

```
```bash
git add .
```
```

5. Commit the initial changes:

```
```bash
git commit -m "Initial commit"
```
```

### ### Creating a GitHub Repository:

1. Go to [GitHub](https://github.com/) and log in or create an account.
2. Click the "+" sign in the top right corner and select "New repository".
3. Give your repository a name and click "Create repository".
4. Follow the instructions to add a remote repository to your local Git:

```
```bash
git remote add origin <repository_url>
```
```

5. Push your initial commit to GitHub:

```
```bash
git push -u origin master
```
```

### ### Versioning with Tags:

1. After making significant changes, create a tag:

```
```bash
git tag -a v1.0 -m "Version 1.0"
```
```

2. Push the tags to GitHub:

```
```bash
git push --tags
```
```

### ### Collaboration:

1. Share the GitHub repository URL with your teammates.
2. They can clone the repository to their local machines:

```
```bash
git clone <repository_url>
```
```

### ### Notifications on GitHub:

1. On your GitHub repository page, go to "Settings" > "Notifications".
2. Configure your notification preferences, such as receiving notifications for all conversations.

### Automated Emails with GitHub Actions:

1. Create a ``.github/workflows/main.yml`` file in your repository with the following content:

```
```yaml
name: Notify Teammates

on:
  push:
    branches:
      - main

jobs:
  notify:
    runs-on: windows

    steps:
      - name: Notify Teammates
        uses: dawidd6/action-send-mail@v2
        with:
          server_address: smtp.gmail.com
          server_port: 587
          username: ${ secrets.EMAIL_USERNAME }
          password: ${ secrets.EMAIL_PASSWORD }
          subject: "New Version Available"
          to: "narendrakopperal23@gmail.com,
adithya.kunda369@gmail.com"
          body: "A new version of the Python stability script is
available. Please update your local copy."
```
```

2. Replace placeholder values with your actual email server information.  
3. Add your email credentials as secrets in your GitHub repository settings.

This structured guide should help you set up a version control system with Git, collaborate with teammates, and receive notifications via GitHub. If you need further assistance, feel free to ask!

command prompt output: C:\Users\HP\Naren\_python>git init.  
git: 'init.' is not a git command. See 'git --help'.

The most similar command is  
init

C:\Users\HP\Naren\_python>git init  
Initialized empty Git repository in C:/Users/HP/Naren\_python/.git/

C:\Users\HP\Naren\_python>git add ..  
fatal: ...: '..' is outside repository at 'C:/Users/HP/Naren\_python'

C:\Users\HP\Naren\_python>git add  
Nothing specified, nothing added.  
hint: Maybe you wanted to say 'git add .'?  
hint: Turn this message off by running  
hint: "git config advice.addEmptyPathsSpec false"

C:\Users\HP\Naren\_python>git add .

C:\Users\HP\Naren\_python>git commit -m "Initial commit"  
Author identity unknown

\*\*\* Please tell me who you are.

Run

```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```

to set your account's default identity.  
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'HP@Adhi369.(none)')

C:\Users\HP\Naren\_python>git commit -m "First change"  
Author identity unknown

\*\*\* Please tell me who you are.

Run

```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```

to set your account's default identity.  
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'HP@Adhi369.(none)')

C:\Users\HP\Naren\_python> git config --global user.email  
"adithya.kunda369@gmail.com"

C:\Users\HP\Naren\_python>git config --global user.name "Adhi"

C:\Users\HP\Naren\_python>git commit -m "Initial Comment"

```
[master (root-commit) 3c4ea38] Initial Comment
1 file changed, 1 insertion(+)
create mode 100644 hello.py
```

```
C:\Users\HP\Naren_python>git remote add origin
https://github.com/Adithya78204/Firstrepo
```

```
C:\Users\HP\Naren_python> git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 240 bytes | 240.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Adithya78204/Firstrepo
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

```
C:\Users\HP\Naren_python>git tag -a v1.0 -m "Version 1.0"
```

```
C:\Users\HP\Naren_python>git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 163 bytes | 163.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Adithya78204/Firstrepo
 * [new tag]         v1.0 -> v1.0
```

```
C:\Users\HP\Naren_python>git tag -a v1.0 -m "Version 2.0"
fatal: tag 'v1.0' already exists
```

```
C:\Users\HP\Naren_python>git tag -a v2.0 -m "Version 2.0"
```

```
C:\Users\HP\Naren_python>git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 164 bytes | 164.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Adithya78204/Firstrepo
 * [new tag]         v2.0 -> v2.0
```

reference procedure:

Certainly! Here's a more detailed step-by-step guide:

### ### 1. \*Setting Up Git Locally:\*

- Open your terminal or command prompt.
- Navigate to the directory containing your Python script.
- Initialize a Git repository: `git init`.
- Add your files to the repository: `git add ..`
- Commit the initial changes: `git commit -m "Initial commit"`.

### ### 2. \*Creating a GitHub Repository:\*

- Go to [GitHub](https://github.com/) and log in or create an account.
- Click the "+" sign in the top right corner and select "New repository".
- Give your repository a name and click "Create repository".

- Follow the instructions to add a remote repository to your local Git:

```
bash
git remote add origin <repository_url>
```

- Push your initial commit to GitHub:

```
bash
git push -u origin master
```

### ### 3. \*Versioning with Tags:\*

- After making significant changes, create a tag:

```
bash
git tag -a v1.0 -m "Version 1.0"
```

- Push the tags to GitHub:

```
bash
git push --tags
```

### ### 4. \*Collaboration:\*

- Share the GitHub repository URL with your teammates.

- They can clone the repository to their local machines:

```
bash
git clone <repository_url>
```

### ### 5. \*Notifications on GitHub:\*

- On your GitHub repository page, go to "Settings" > "Notifications".

- Configure your notification preferences, such as receiving notifications for all conversations.

- GitHub will notify you about issues, pull requests, and other activities related to your repository.

### ### 6. \*Automated Emails with GitHub Actions:\*

- Create a .github/workflows/main.yml file in your repository with the following content:

```
name: Notify Teammates
```

```
on:
```

```
  push:
    branches:
      - main
```

```
jobs:
```

```
  notify:
    runs-on: windows
```

```
  steps:
```

```
    - name: Notify Teammates
      uses: dawidd6/action-send-mail@v2
      with:
        server_address: smtp.gmail.com
        server_port: 587
        username: ${ secrets.EMAIL_USERNAME }}
        password: ${ secrets.EMAIL_PASSWORD }}
        subject: "New Version Available"
        to: "narendrakoppera123@gmail.com, adithya.kunda369@gmail.com"
```

```
body: "A new version of the Python stability script is
available. Please update your local copy."
```

- Replace placeholder values with your actual email server information.
- Add your email credentials as secrets in your GitHub repository settings.

This detailed guide should help you set up a version control system with Git, collaborate with teammates, and receive notifications via GitHub.

How to create yml:

Sure! You can create a ``.github/workflows/main.yml`` file directly within your GitHub repository by following these steps:

1. Navigate to your GitHub repository on the GitHub website.
2. Click on the "Actions" tab.
3. Click on the "New workflow" button.
4. Choose "Set up a workflow yourself" to create a new workflow file.
5. GitHub will open a text editor where you can define your workflow.
6. Copy and paste the following YAML content into the editor:

```
```yaml
name: CI

on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repository
        uses: actions/checkout@v2

      # Add your build steps here
      # For example, running tests
      - name: Run tests
        run: |
          # Insert your test commands here

      # Additional steps can be added as needed
...
```
```

7. Modify the content of the file as per your requirements. This example sets up a simple CI workflow that runs on every push to the main branch and on every pull request targeting the main branch. You can customize it to fit your specific needs.

8. Once you're done, click on the "Start commit" button at the bottom.
9. Provide a commit message, optionally describe your changes, and click on the "Commit new file" button.

You have successfully created a ``.github/workflows/main.yml`` file in your GitHub repository, setting up a CI workflow.

creating secret keys:

No, ``${{ secrets.EMAIL_USERNAME }}`` and ``${{ secrets.EMAIL_PASSWORD }}`` are not your GitHub credentials. They are placeholders that refer to secrets stored in your GitHub repository settings.

GitHub allows you to store sensitive information, such as API tokens, passwords, or any other confidential data, securely using encrypted secrets. These secrets can then be accessed by your workflows during execution.

To replace ``${{ secrets.EMAIL_USERNAME }}`` and ``${{ secrets.EMAIL_PASSWORD }}`` with your actual email username and password respectively, you need to follow these steps:

1. **\*\*Set up Secrets\*\***: Go to your GitHub repository page and navigate to the "Settings" tab.
2. **\*\*Manage Secrets\*\***: In the left sidebar, click on "Secrets" or "Secrets and keys", depending on the GitHub interface.
3. **\*\*Add New Secret\*\***: Click on the "New repository secret" or similar button.
4. **\*\*Enter Details\*\***: For the first secret, set the name as ``EMAIL_USERNAME`` and enter your email username as the value. Then, add another secret with the name ``EMAIL_PASSWORD`` and enter your email password as the value.
5. **\*\*Save Secrets\*\***: Click on "Add secret" to save each secret.

Once you've added these secrets, GitHub will securely store them and make them available to your workflows when they run. Your workflow can then reference these secrets using ``${{ secrets.EMAIL_USERNAME }}`` and ``${{ secrets.EMAIL_PASSWORD }}`` respectively, allowing your workflow to send emails using your email credentials without exposing them in the YAML file.

about smtp server:

To find the SMTP server address of your email provider, you typically need to refer to the documentation provided by your email service or consult their support resources. However, here are some general steps you can follow:

1. **\*\*Check Provider's Website\*\***: Visit the website of your email service provider and search for their SMTP server settings. Many providers have help or support sections where you can find this information.
2. **\*\*Search Online\*\***: You can search online for the SMTP server settings of your email provider. Many users and support forums discuss these settings, so you may find relevant information through a simple search.
3. **\*\*Contact Support\*\***: If you can't find the information you need, consider reaching out to your email service provider's customer support. They should be able to provide you with the SMTP server address and any other necessary details.

4. **\*\*Use Common Defaults\*\***: Some common email providers have well-known SMTP server addresses. For example:

- Gmail: smtp.gmail.com
- Yahoo Mail: smtp.mail.yahoo.com
- Outlook.com (formerly Hotmail): smtp.live.com

Remember, the SMTP server address may vary depending on whether you're using a free email service, a custom domain email, or a business email service. Always refer to the specific documentation or support resources provided by your email service provider for the most accurate information.