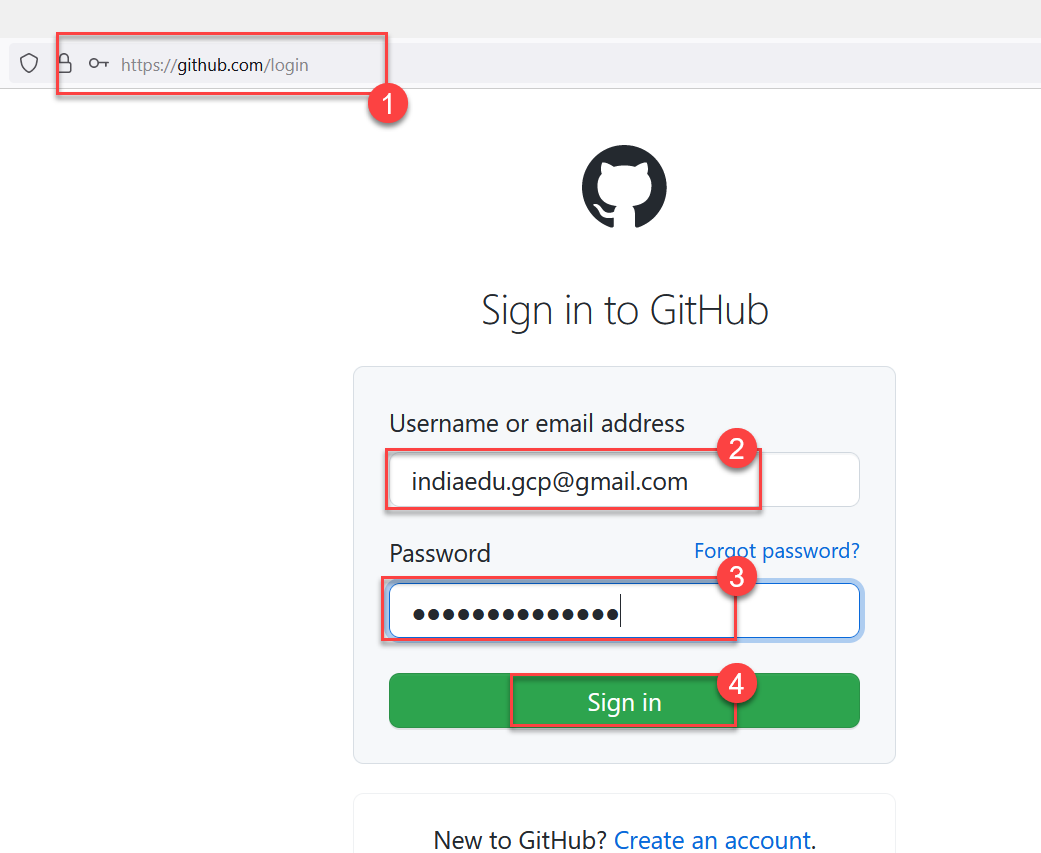
Demo 4 : CI/CD Using GitHub Actions

NOTE: Please complete all the three demos first.

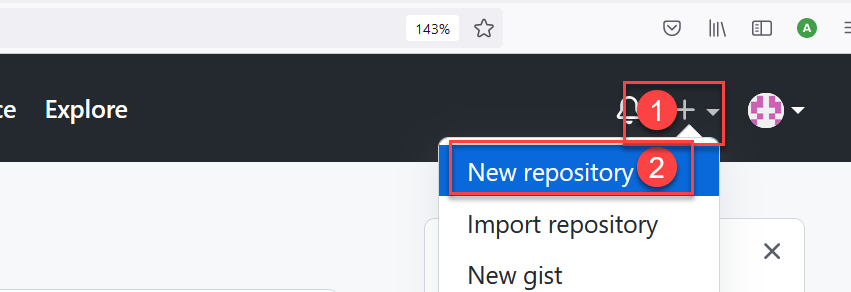
# PHASE 1 : Setup Source control

1. Login in to your GitHub account (linked to Student pack)

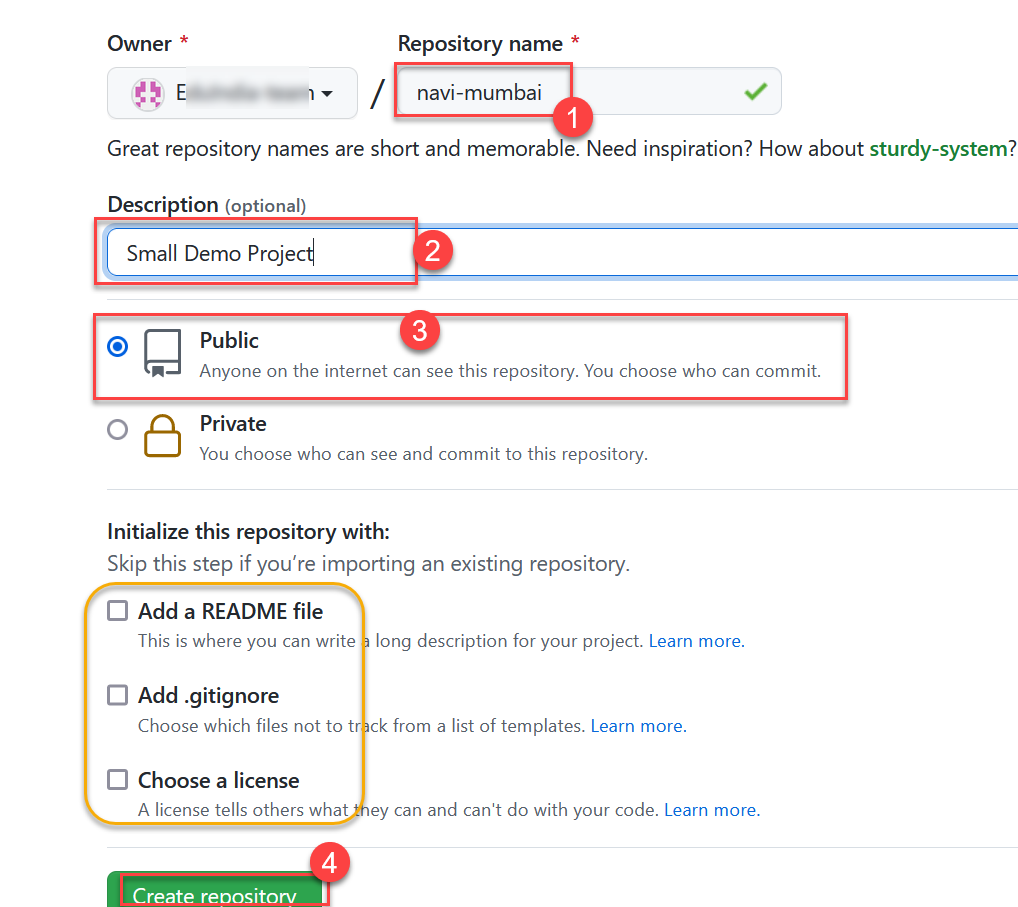


***You might be prompted for token sent to your registered email id. Please go to index to get it and use it for login.***

1. Now Create a new repository.

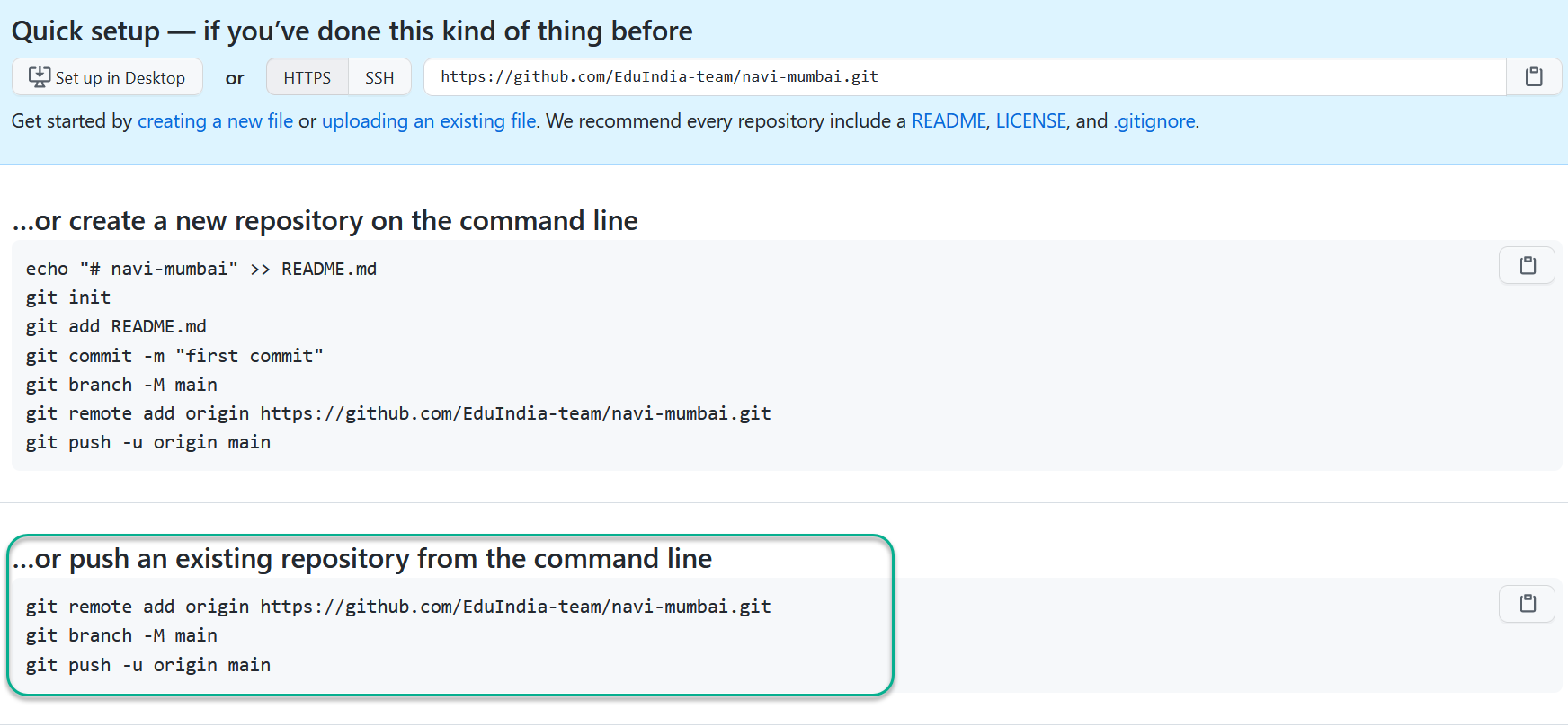


1. Create new “bare” repository (no readme, no gitignore, no license file )



Now, you must have “GIT For Windows” Installed on your system to try next steps.

1. You should get instructions on how to add local repository like this one. Just keep in Open, you need it later!



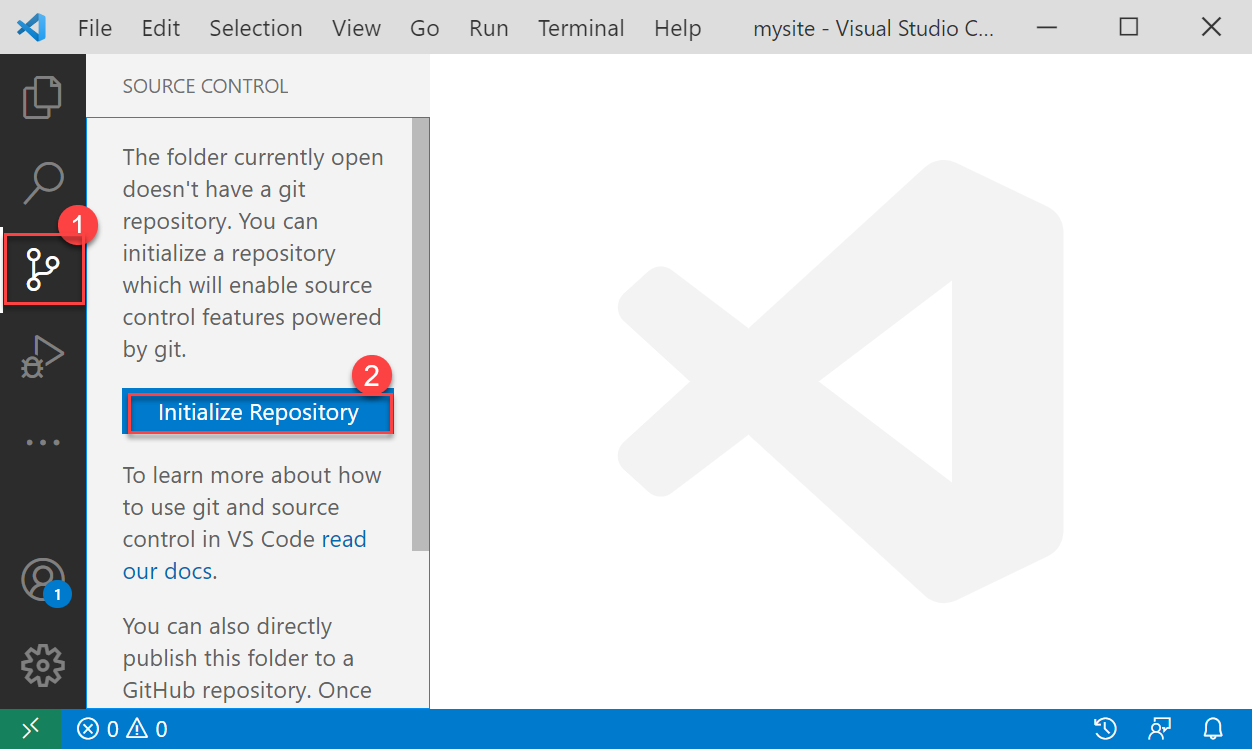
Please keep this browser window OPEN, the text highlighted would be used for setting up local repository.

1. Now, lets get back to local filesystem folder where you have extracted the bootstrap website. (Refer to Demo3: bootstrap) I had created a folder D:\mysite
2. Use “Visual Studio Code” to open the project directory (d:\mysite for me)

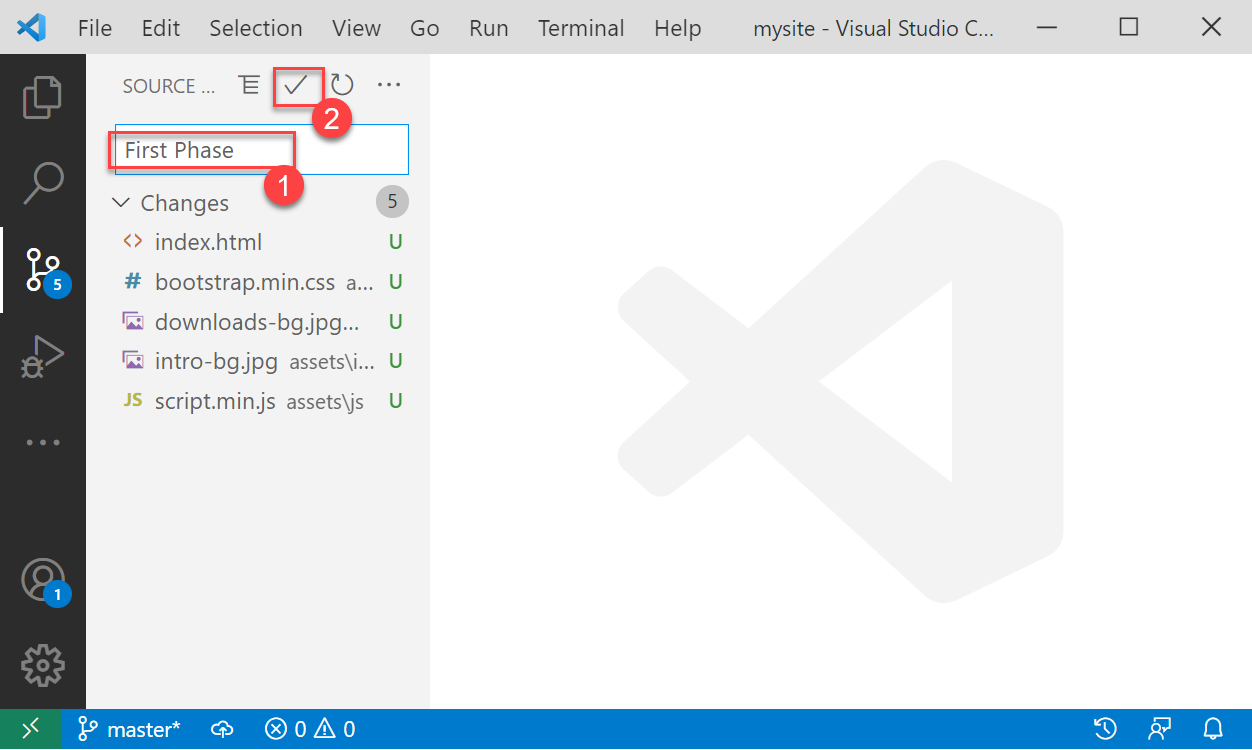
You can use following command to launch VS Code into desired directory from command prompt

$ code d:\mysite

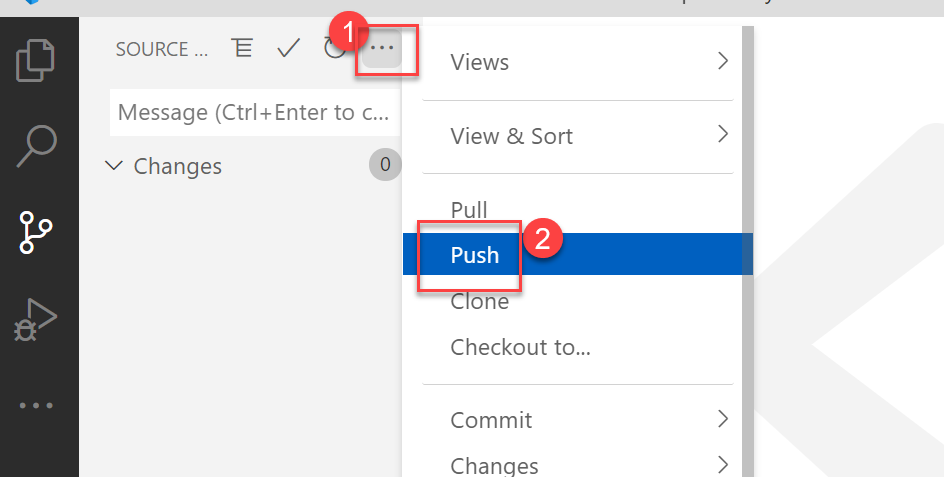
1. VS Code has an option of GIT Repository, just use “Initialize Repository” button to start creating a repository.



1. Now, Accept all the changes, use new commit message “First Phase” and click the commit button.

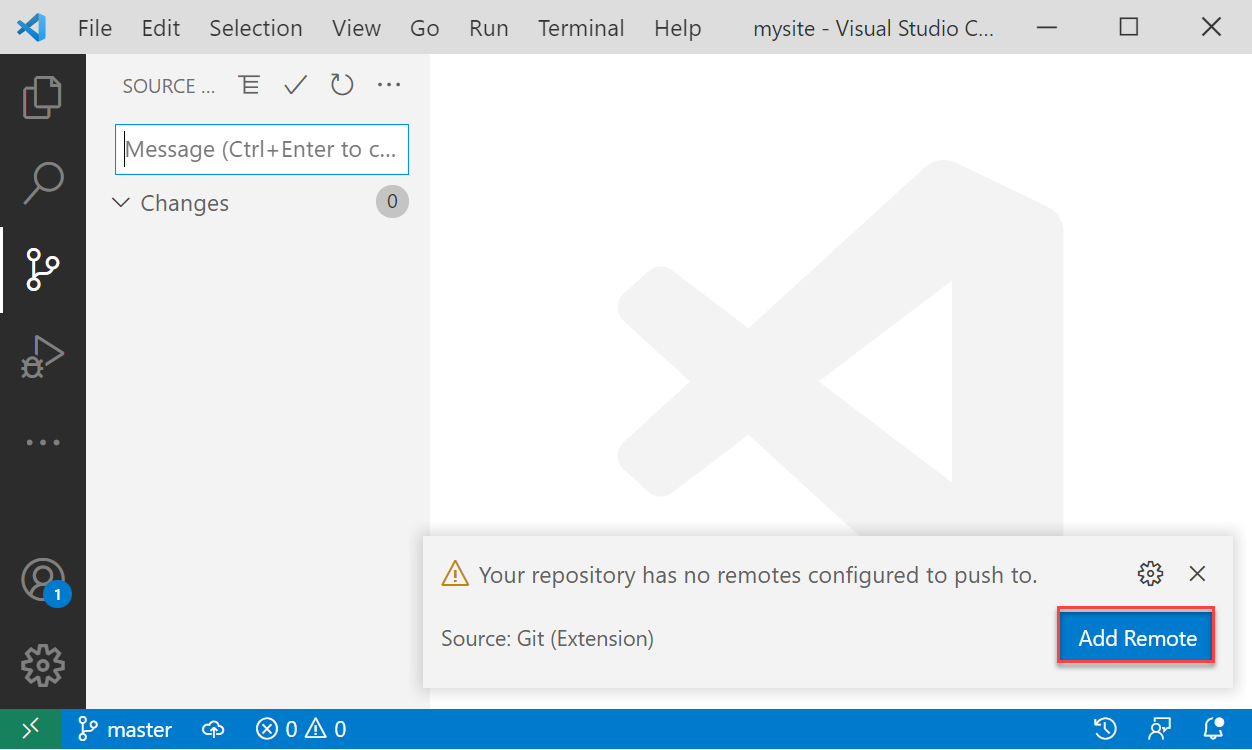


1. Now, Try pushing changes to remote repository.

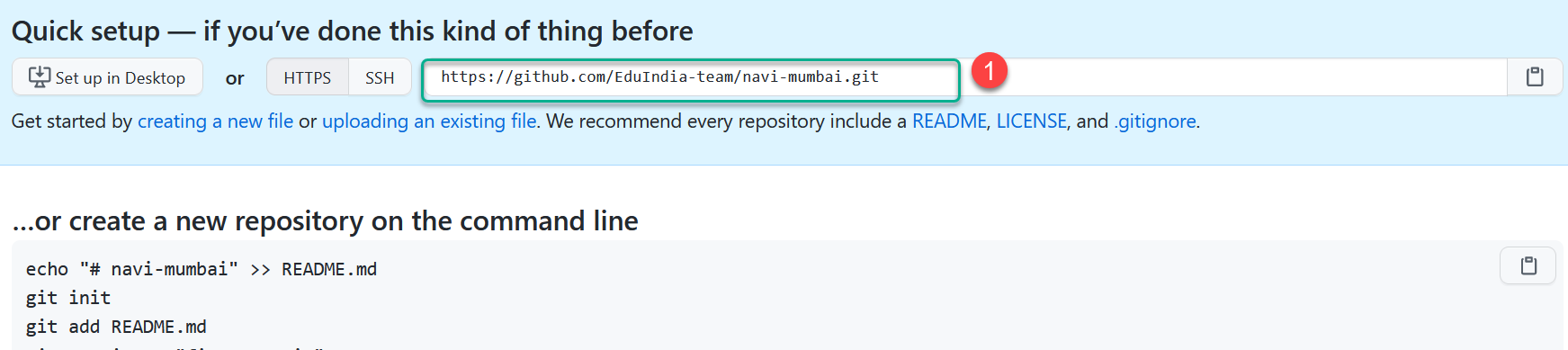


1. Now, VSCode should complain that you have no REMOTE REPOSITORY!

Please click on “Add Remote” button



1. Now, You need to get your remote repository URL from browser window you kept open earlier.



1. Now use command prompt and move to directory where you have extracted your website.



1. You can now, convert this one into a local repository with following commands (ignore “$” )

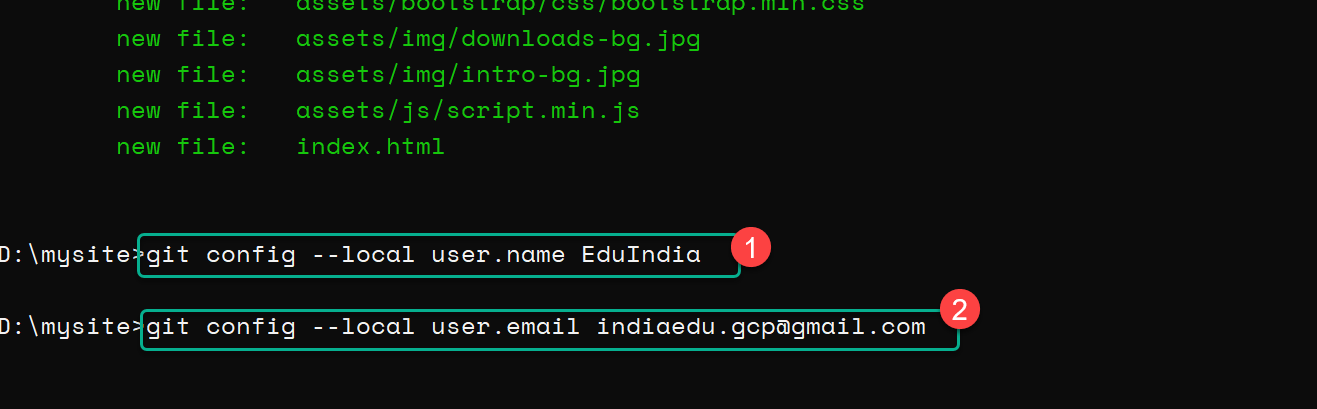
FIRST TWO LINES HAVE “DOT” at the end which is LINK to current directory

$ git init .

$ git config --local user.name **YOURNAME**

$ git config --local user.email **YOURGITHUBEMAIL**

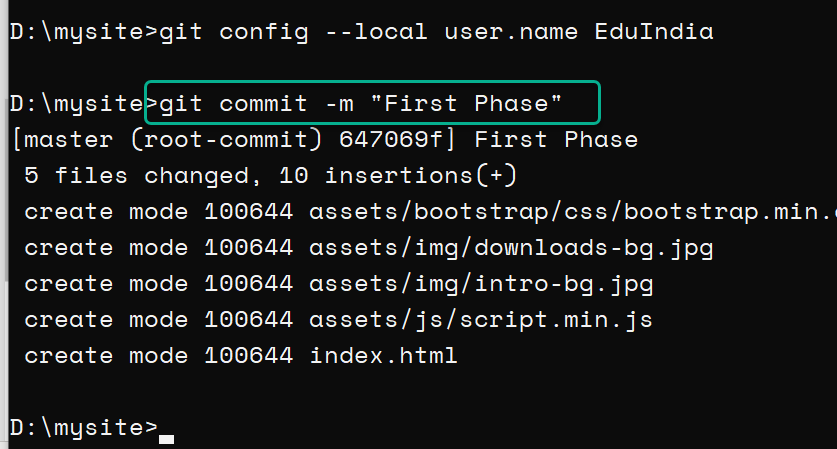
***NOTE: Please replace the text in UPPERCASE with indented values***



1. Now, make a new commit using following command (DOT at the end of first command)

$ git add .

$ git commit -m “First Phase”

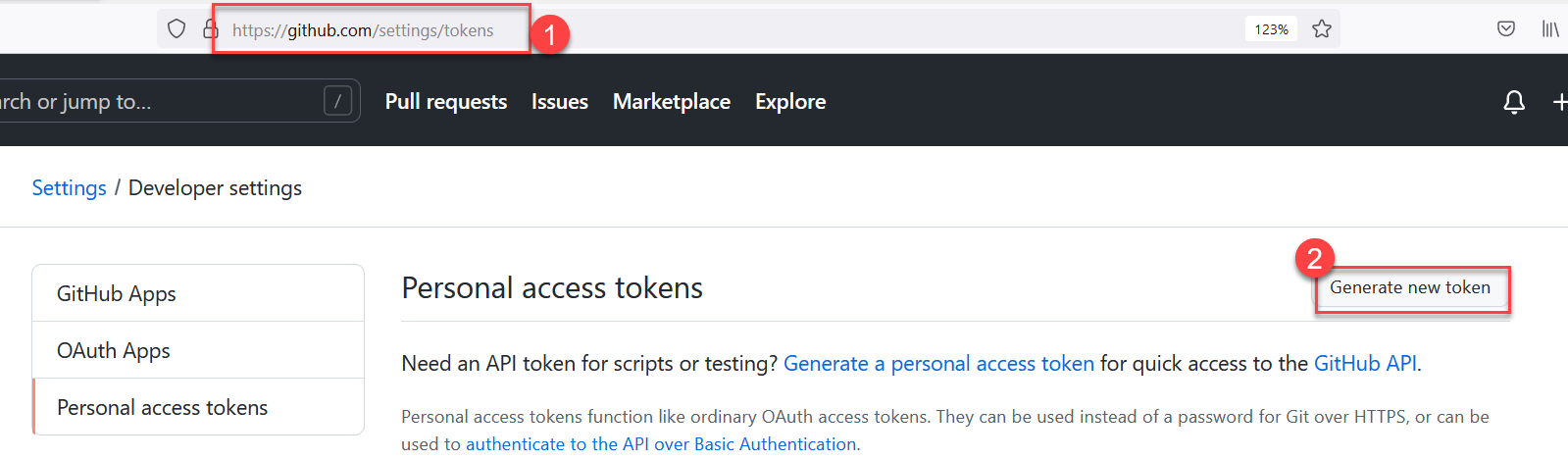


1. Lets force GIT not to use any EXISTING CREDENTIALS for current repository and force login on push operation.

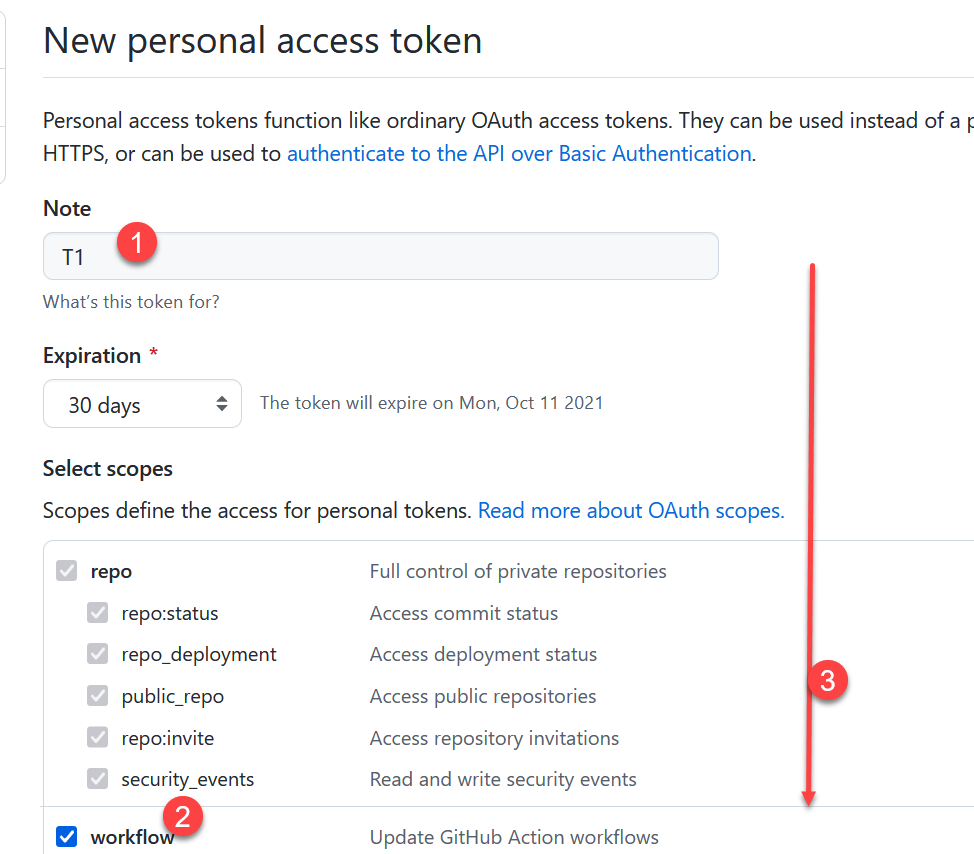
$ git config --local credential.helper ""

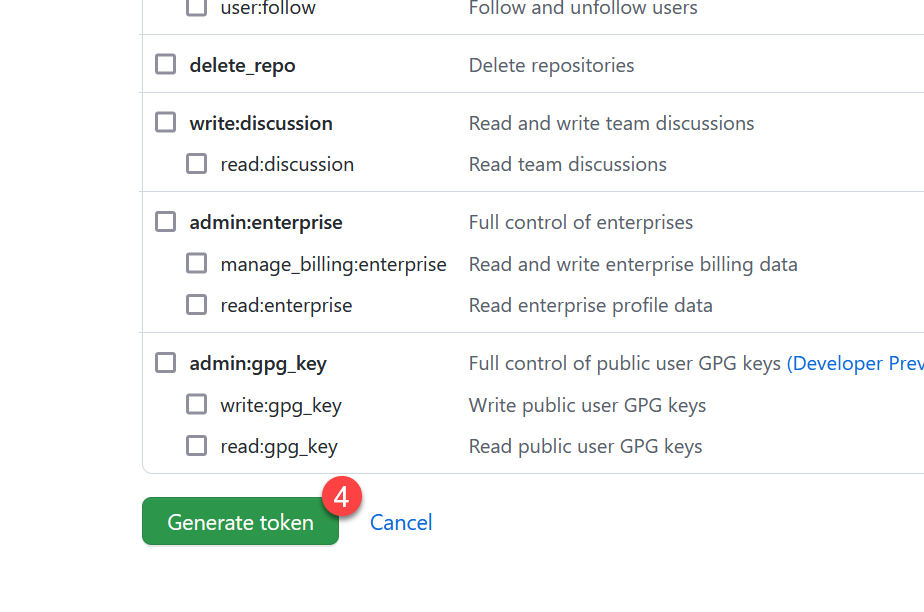
1. Now, You need a new Personal Access Token (PAT) to authorize the Git CLI. Go back to browser window where you have logged-in into GitHub. And Try this url: <https://github.com/settings/tokens>

Click on “Generate New token” button



1. Provide a new name and make sure it allows you to edit “GitHub Actions/Workflows”





1. Copy the generated token immediately (use notepad to save this token for later use).

***TOKENs cannot be viewed later***

1. Now, back to command prompt, Lets add the remote repository URL (These command can be found in step# 4

$ git remote add origin [**https://github.com/EduIndia-team/navi-mumbai.git**](https://github.com/EduIndia-team/navi-mumbai.git)

$ git branch -M main

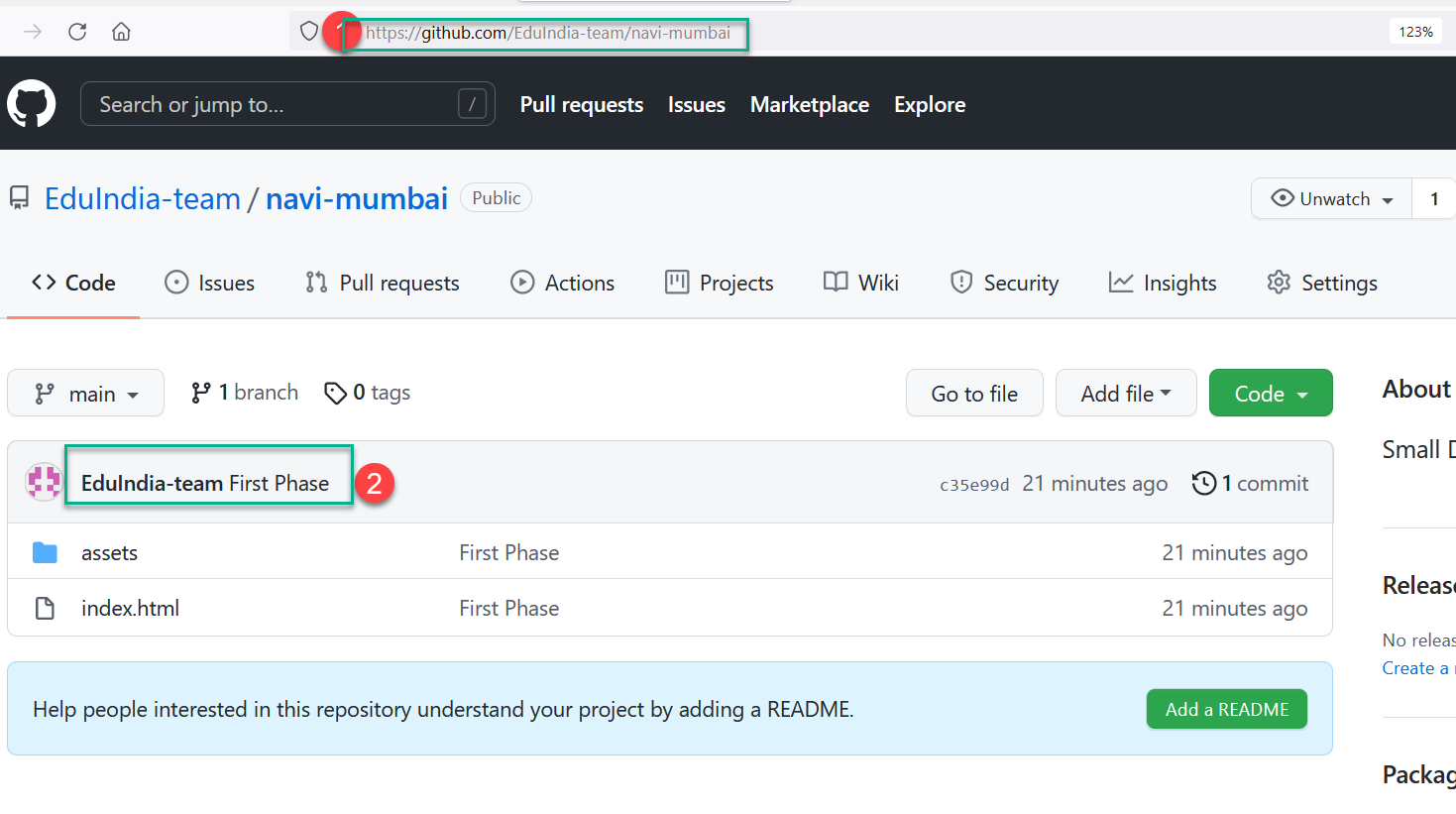
$ git push -u origin main

**NOTE: Please replace the URL with your repository.**

1. When prompted, just enter your github email and access token

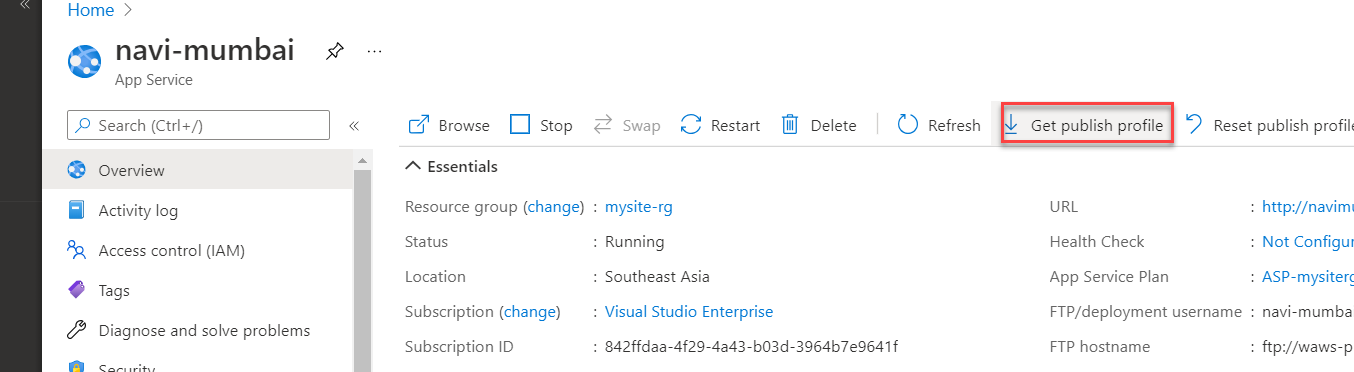


1. Just go back and check your remote repository in web browser.

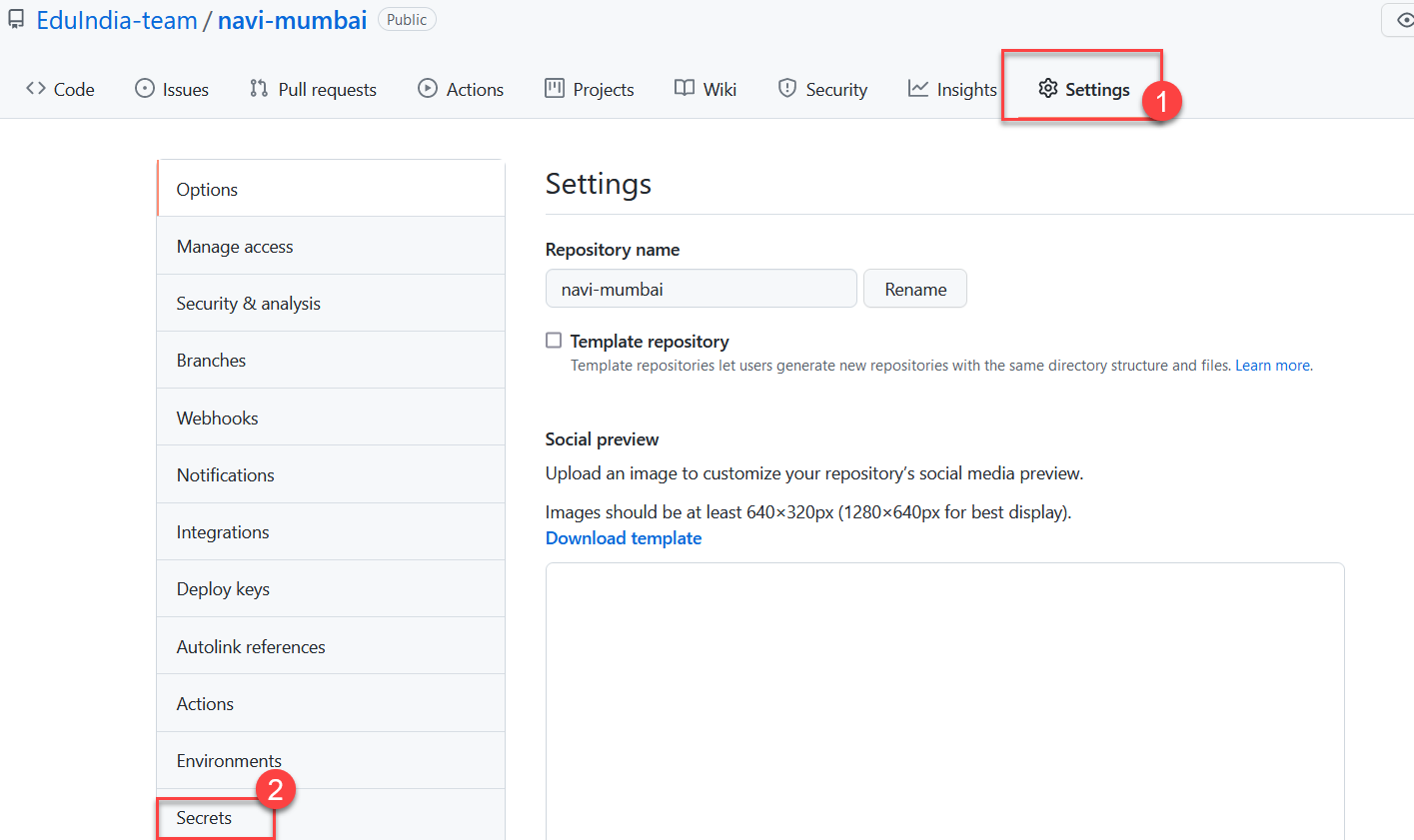


## PHASE 2 : Setup GitHub Action

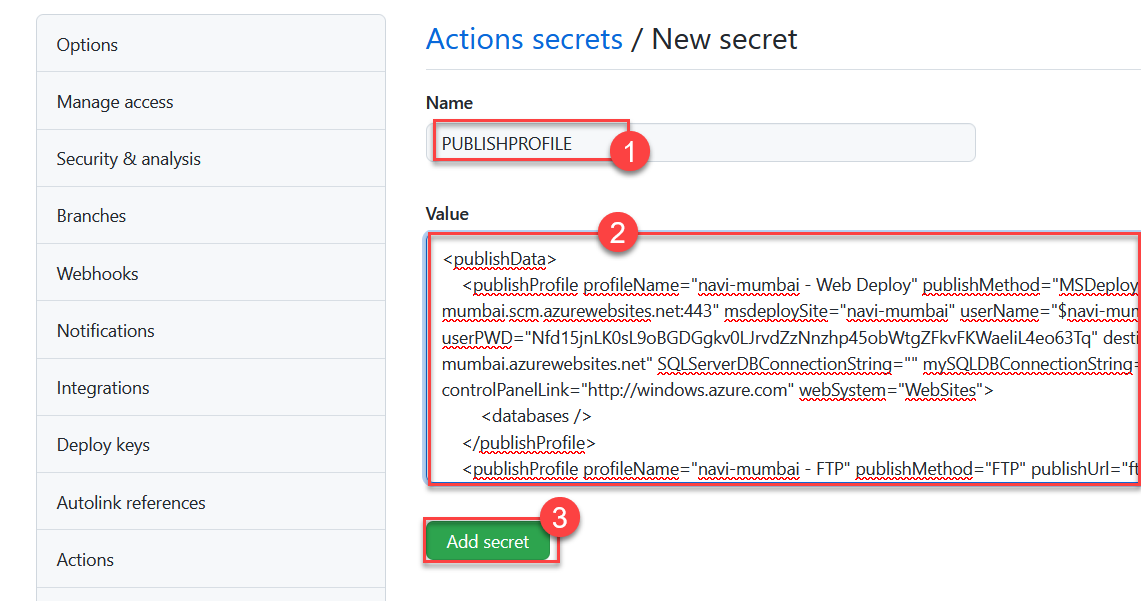
1. Login into azure portal and then download the “Publish Profile” for your App Service.



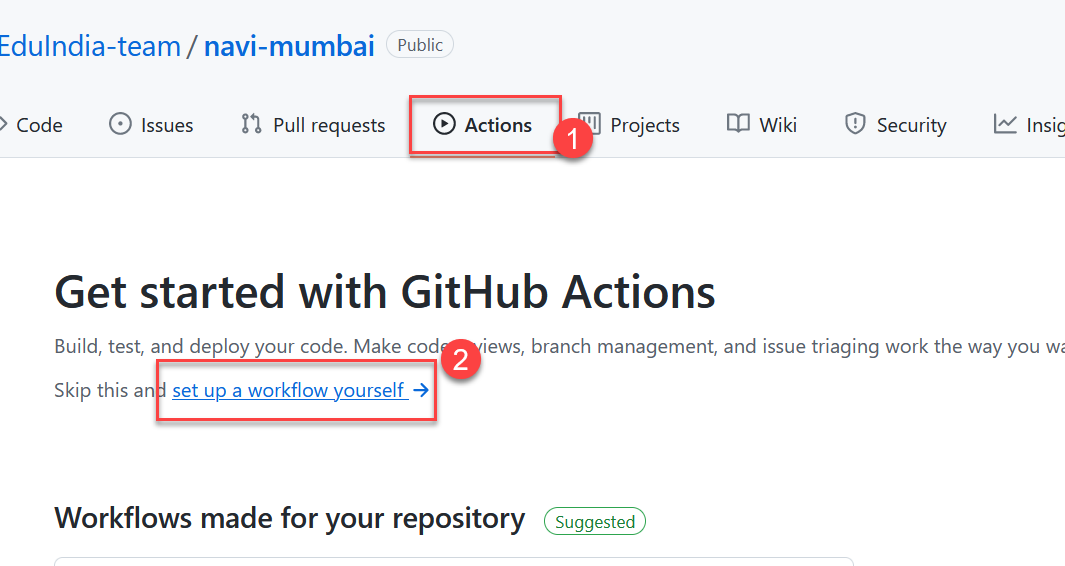
1. Now, Visit your Github repository in web browser and create a new “SECRET” to be used in “Github Action”
2. Goto your repository and then click on “Settings” for repository, then choose “Secrets”



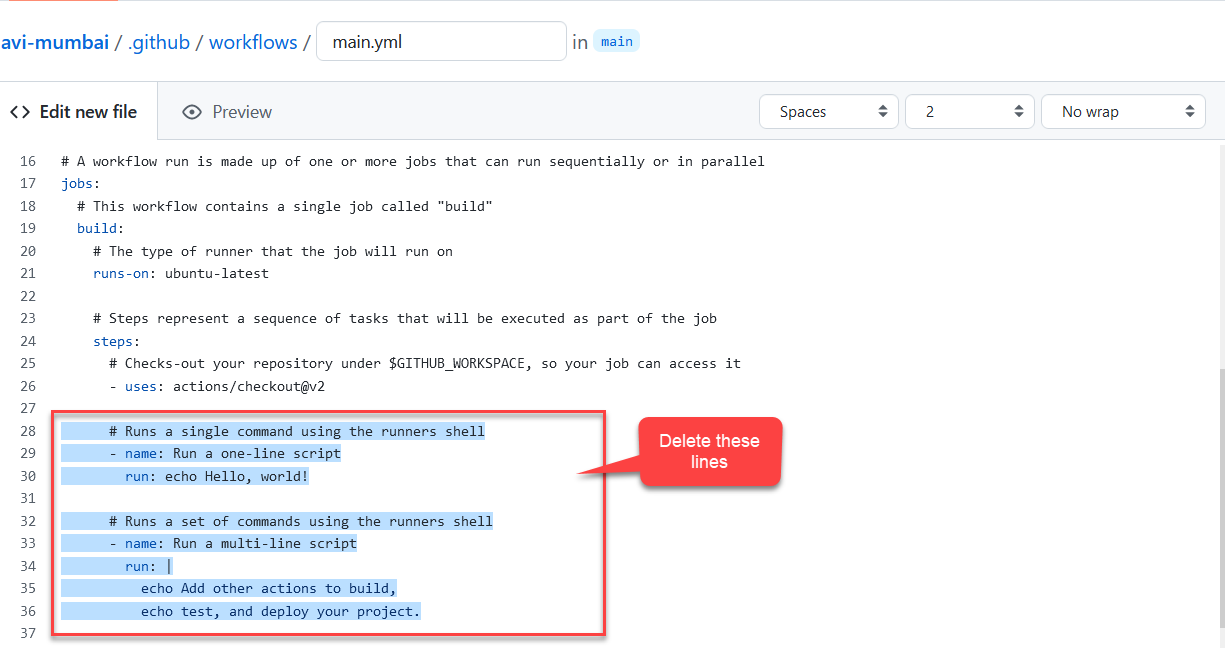
1. Now, create a new repository secret with name “PUBLISHPROFILE” and for value, copy the contents of “publishprofile” you have downloaded in step1.



1. Now, Create a new empty Workflow for current repository by using “Actions” tab



1. Once a workflow file (with default steps) is created, just delete the last two steps (highlighted)



1. Now, Add these lines (for azure app-service deployment)

      - name: Azure WebApp

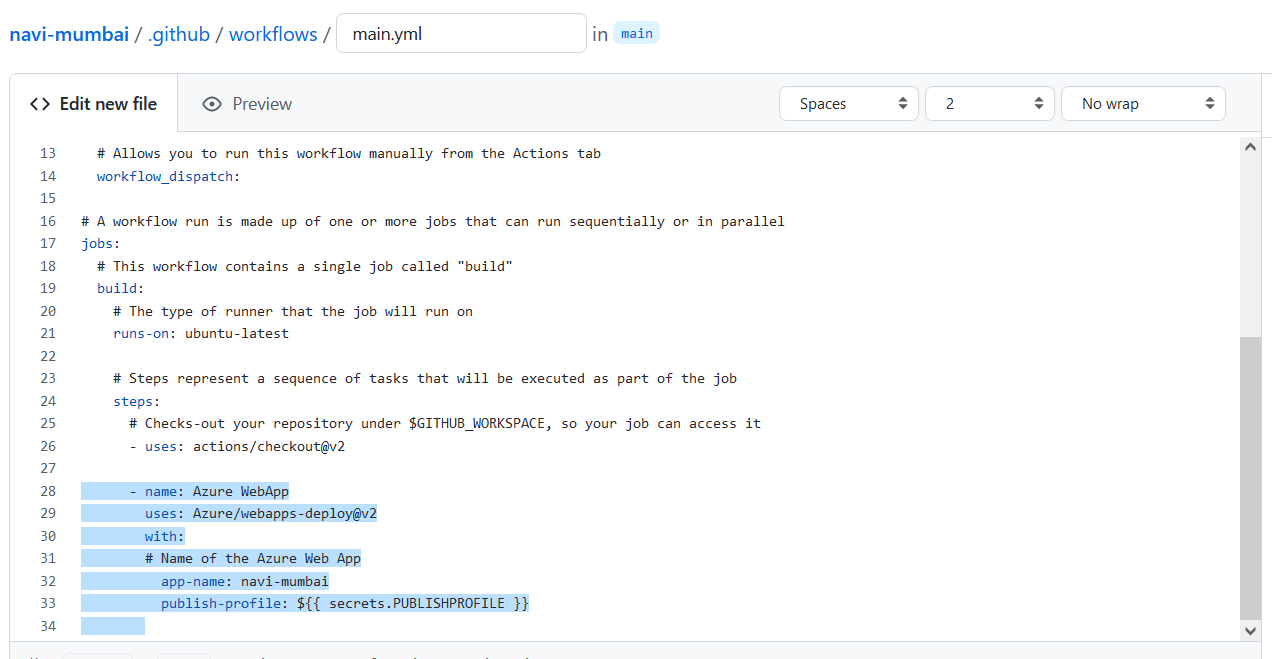
        uses: Azure/webapps-deploy@v2

        with:

        # Name of the Azure Web App

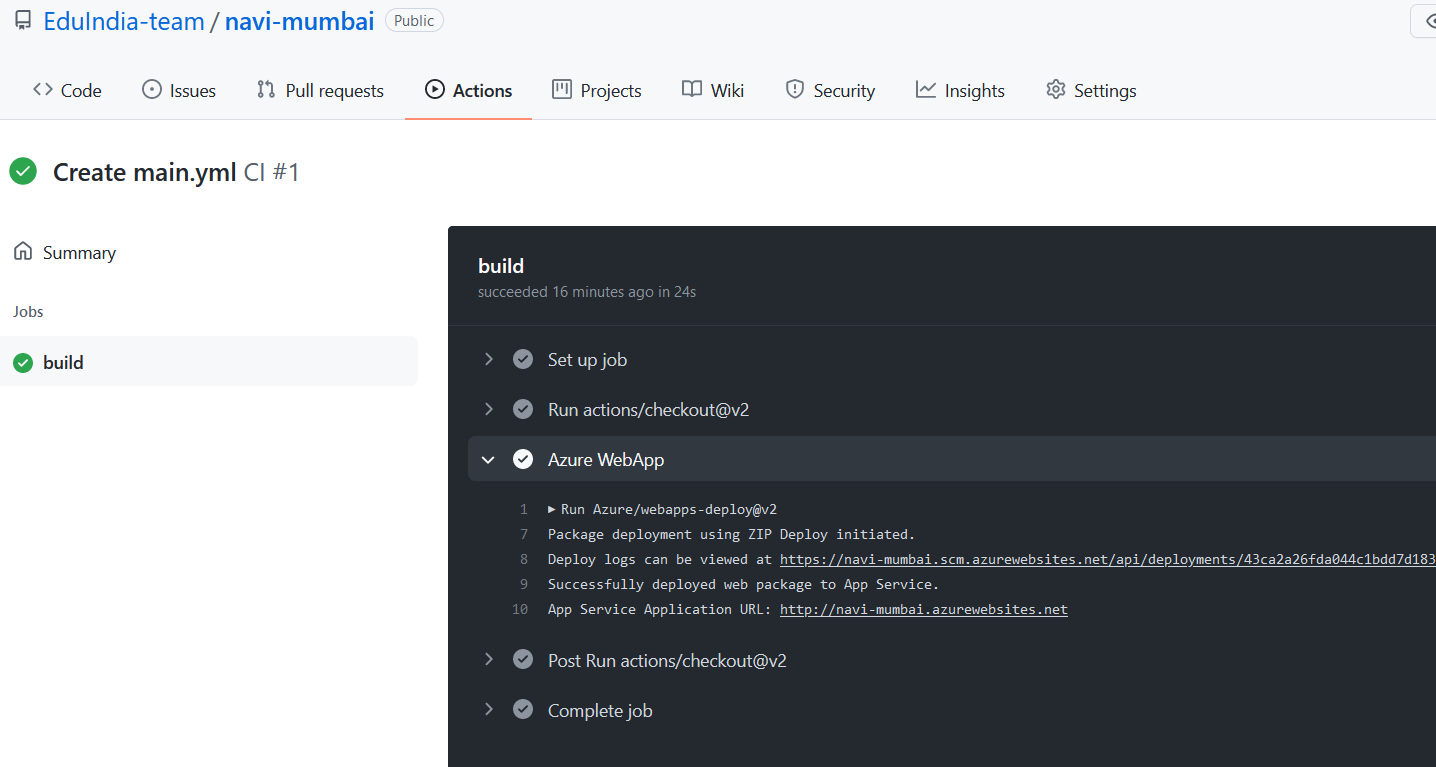
          app-name: navi-mumbai

          publish-profile: ${{ secrets.PUBLISHPROFILE }}



NOTE: Please check the vertical alignment of steps added.

1. Now, Commit the changes to Workflow and check the progress on next screen



1. To Verify, please visit the web-site.

PHASE 3: Continuous Deployment : Code to App Deployment

1. Open the local code repository in browser of your choice and then make some changes in “index.html” file . (I changed the title of page)
2. Use command prompt to update the local repository and then make and push new changes.

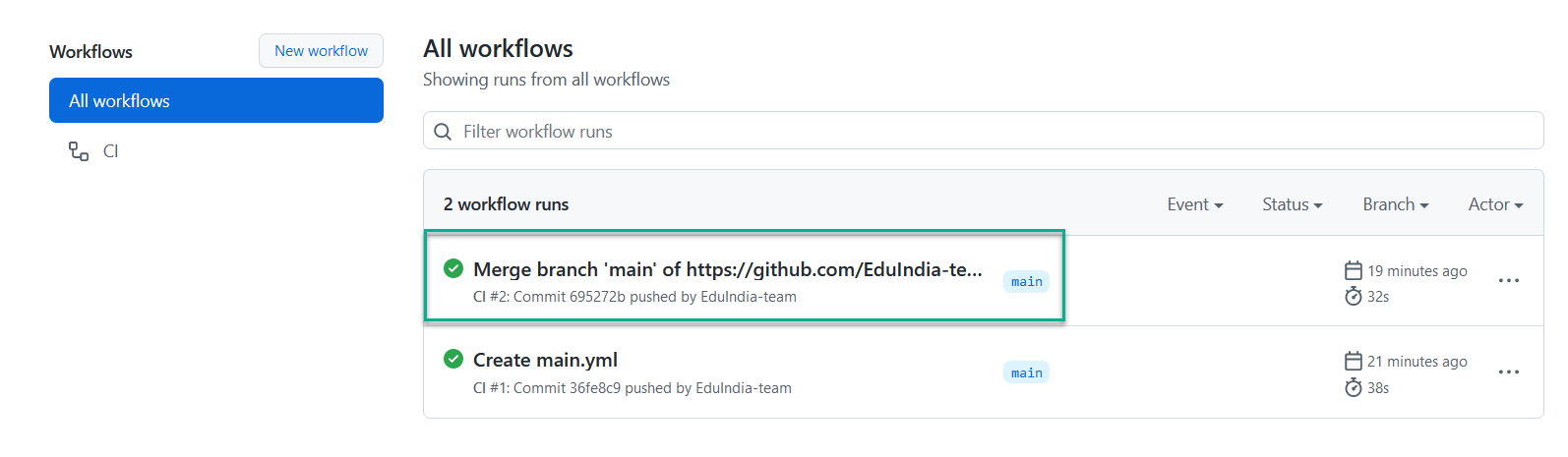
$ git add .

$ git commit -m “Updated title”

$ git pull

$ git push

1. Please note, you might have to provide the user-email and access token once again !
2. Check the github actions, you should get second run of workflow.



1. Check if the changes are reflected in web-site as well

