



# FIRE: THE FIRST-YEAR INNOVATION & RESEARCH EXPERIENCE

## **SUSTAINABILITY ANALYTICS**

How to update changes in Posit Cloud to the same  
GitHub repository

# Storing new changes back into GitHub

There are three steps:

1. **Stage**
2. **Commit** - add a message to record what you just did.
3. **Push** - replace/add the updated files into the GitHub repository.

Caveat:

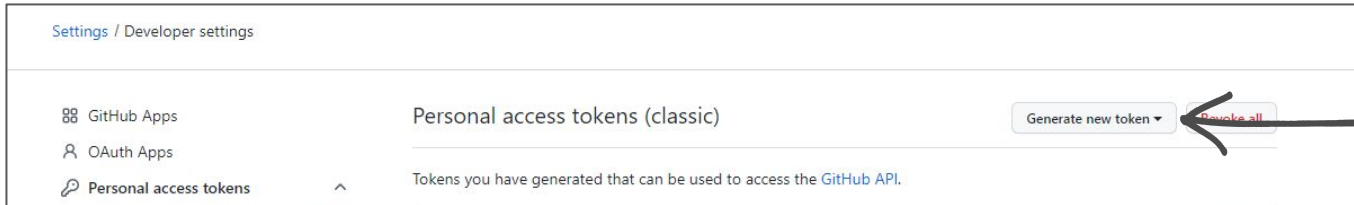
You need a PAT (Personal Access Tokens, or a GitHub-generated password) to verify your identity and **push** the files.

# Storing new changes back into GitHub

## Getting a PAT\*

Step 1: Go to <https://github.com/settings/tokens>

Step 2: Choose “Generate new token” and “Generate new token (classic)”



\*If you have previously generated a PAT, you can skip this step.

# Storing new changes back into GitHub

## Getting a PAT\*

Step 3: Write a Note

Step 4: Change **Expiration to 90 days**

Step 5: Check **repo, user, and project** and Click **Generate Token**

Step 6: Store the token in a secure location, such as lastpass.com

\*If you have previously generated a PAT, you can skip this step.

The screenshot shows the GitHub 'New personal access token (classic)' page. It includes a 'Note' field with the text 'connect github to my posit cloud account' (callout 3), an 'Expiration' dropdown set to '90 days' (callout 4), and a 'Select scopes' section with several checked options: 'repo', 'repo:status', 'repo:deployment', 'public\_repo', 'repo:invite', and 'security\_events' (callout 5). Other unchecked options include 'workflow', 'write:packages', 'read:packages', 'delete:packages', 'admin:org', 'write:org', 'read:org', 'manage\_runners:org', 'admin:public\_key', 'write:public\_key', and 'read:public\_key'.

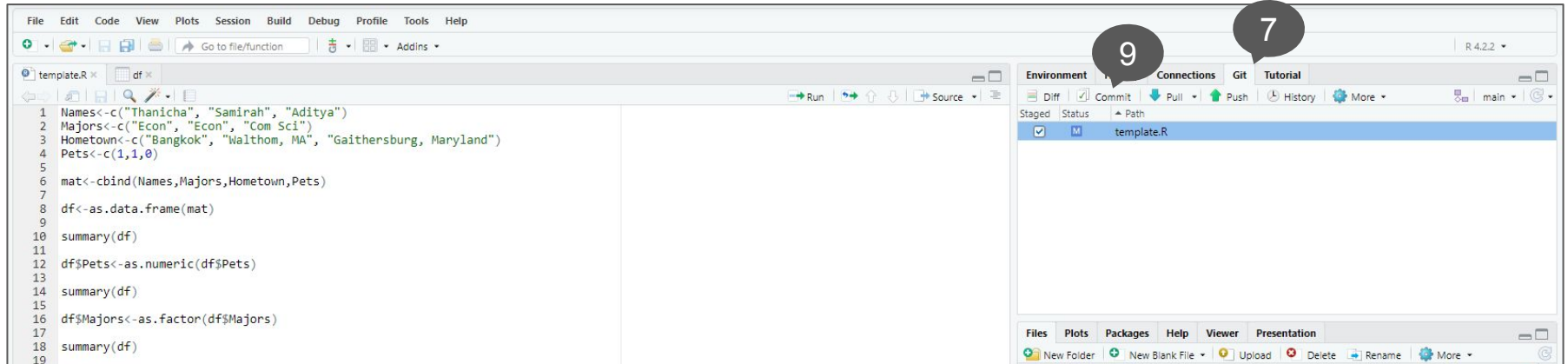
Scope	Description
<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo:deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> manage_runners:org	Manage org runners and runner groups
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys

# Storing new changes back into GitHub

Step 7: Go to the Git tab.

Step 8: Check all files that have been saved. This is the stage process.

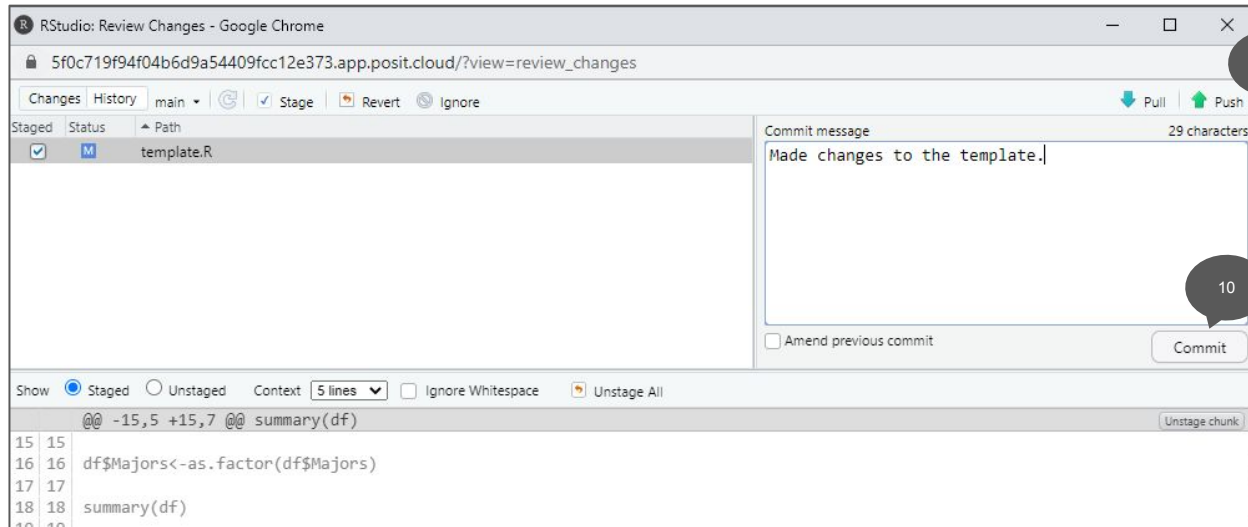
Step 9: Click Commit.



# Storing new changes back into GitHub

Step 10: Write a Commit message to briefly describe what you did. Click Commit.

Step 11: Click Push.

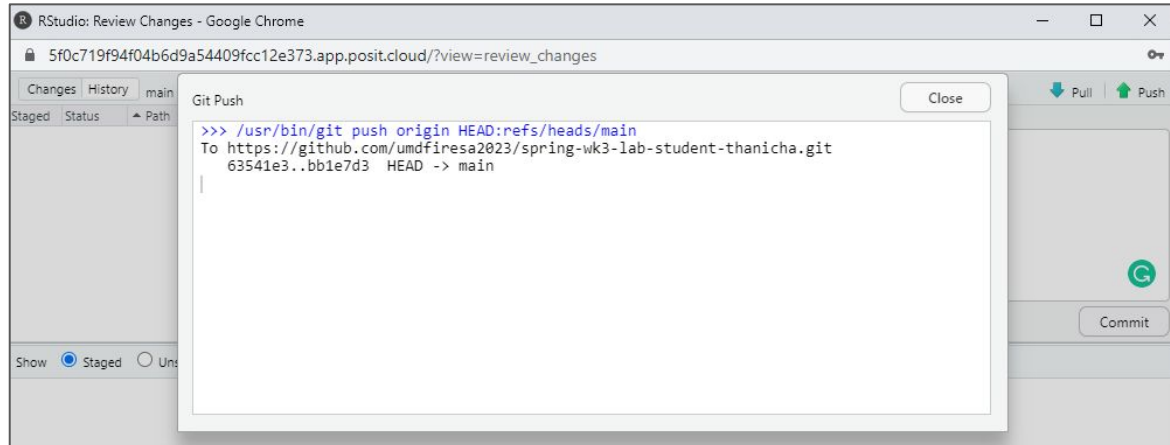


# Storing new changes back into GitHub

Step 12: Type your GitHub username

Step 13: Insert the newly generated PAT as your password.

A message similar to the one shown below means that you have successfully push your file in GitHub.



The screenshot shows a web browser window titled "RStudio: Review Changes - Google Chrome" with the URL "5f0c719f94f04b6d9a54409fcc12e373.app.posit.cloud/?view=review\_changes". The browser displays the RStudio Review Changes interface. A "Git Push" dialog box is open, showing the following output:

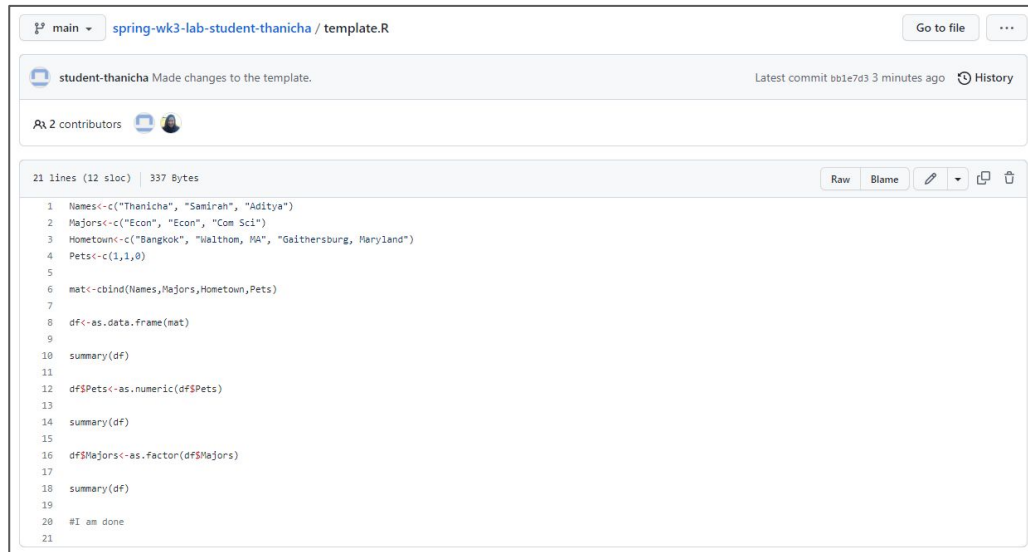
```
>>> /usr/bin/git push origin HEAD:refs/heads/main
To https://github.com/umdfiresa2023/spring-wk3-lab-student-thanicha.git
63541e3..bb1e7d3 HEAD -> main
```

The dialog box has a "Close" button in the top right corner. In the background, the Review Changes interface shows a "Commit" button and a green circular icon.

# Storing new changes back into GitHub

Step 14: Remember to close your Posit Cloud tab.

Step 15: Check if your GitHub repository has been updated.



The screenshot shows a GitHub repository page for a file named `template.R` in the `main` branch of the repository `spring-wk3-lab-student-thanicha`. The page indicates that `student-thanicha` made changes to the template, with the latest commit being `bb1e7d3` from 3 minutes ago. There are 2 contributors listed. The file content is displayed as follows:

```
1 Names<-c("Thanicha", "Samirah", "Aditya")
2 Majors<-c("Econ", "Econ", "Com Sci")
3 Hometown<-c("Bangkok", "Waltham, MA", "Gaithersburg, Maryland")
4 Pets<-c(1,1,0)
5
6 mat<-cbind(Names,Majors,Hometown,Pets)
7
8 df<-as.data.frame(mat)
9
10 summary(df)
11
12 df$Pets<-as.numeric(df$Pets)
13
14 summary(df)
15
16 df$Majors<-as.factor(df$Majors)
17
18 summary(df)
19
20 #I am done
21
```