GitHub Setup for DATA 3402 (WSL & macOS)

This document walks me step by step through setting up GitHub **correctly** for this course using **WSL (Ubuntu on Windows)** or **macOS**. I will do this **once**, and then use the same workflow for the entire semester.

Big Picture (What I Am Setting Up)

- I will **fork** the course repository (my own public copy).
- I will **clone my fork** to my computer.
- I will configure Git so that:
- `git pull` pulls updates from the **class repository**
- `git push` pushes my work to **my fork**
- I will submit labs by committing and pushing notebooks.

Step 0: Prerequisites

WSL (Windows)

- Ubuntu installed via WSL
- Terminal opens into Ubuntu

macOS

- Terminal.app
- Xcode Command Line Tools installed:

```
xcode-select --install
```

Both

- A GitHub account

Step 1: Fork the Course Repository (Browser)

1. Open the course GitHub repository
2. Click **Fork** (top right)
3. Fork it into **my own GitHub account**

⚠️ **Important**: Because the course repo is public, my fork must also be public.

Step 2: Choose How I Will Authenticate (HTTPS or SSH)

I must choose **one** of the following methods. Both are acceptable for this course.

Option A (Recommended for most students): HTTPS

- Easier to set up
- No SSH keys required
- Uses browser-based authentication (GitHub login)

What I copy: 1. On *my fork*, click the green **Code** button 2. Select **HTTPS** 3. Copy the URL (looks like `https://github.com/username/DATA3402.Spring.2026.git`)

Option B: SSH (Advanced / Already Familiar)

- More secure
- Requires one-time SSH key setup

What I copy: 1. On *my fork*, click the green **Code** button 2. Select **SSH** 3. Copy the URL (looks like `git@github.com:username/DATA3402.Spring.2026.git`)

Step 3: Create a Course Directory (Local)

WSL (Recommended location)

```
cd ~
mkdir Data-3402
cd Data-3402
```

⚠️ **Do NOT work inside ``**

- Causes Git and Jupyter issues
- Slower file access

macOS

```
mkdir ~/Data-3402
cd ~/Data-3402
```

Step 4: Clone My Fork

If I chose HTTPS

```
git clone https://github.com/MY_USERNAME/DATA3402.Spring.2026.git
cd DATA3402.Spring.2026
```

- A browser window may open asking me to log into GitHub
- This is normal

If I chose SSH

```
git clone git@github.com:MY_USERNAME/DATA3402.Spring.2026.git
cd DATA3402.Spring.2026
```

Possible prompt (normal)

```
Are you sure you want to continue connecting (yes/no)?
```

Type `yes`

Step 5: SSH Setup (Only If I Chose SSH)

⚠️ Skip this entire step if I chose HTTPS.

Symptom

```
Please make sure you have the correct access rights
```

Fix

5.1 Generate SSH key

```
ssh-keygen -t ed25519 -C "my_email@uta.edu"
```

Press Enter for all prompts.

5.2 Start agent and add key

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519
```

5.3 Add key to GitHub

```
cat ~/.ssh/id_ed25519.pub
```

- Copy output
- GitHub → Settings → SSH & GPG keys → New SSH key → Paste → Save

5.4 Test

```
ssh -T git@github.com
```

✔️ Should say I am authenticated

Then **re-run clone**.

Step 6: Configure Git Identity (Required)

If I see:

```
Author identity unknown
```

I must run:

```
git config --global user.name "My Full Name"
git config --global user.email "my_netid@uta.edu"
```

Verify:

```
git config --global --list
```

Step 7: Fix Remotes (CRITICAL STEP)

Check current remotes

```
git remote -v
```

Remove default origin

```
git remote remove origin
```

Add class repo as fetch origin

```
git remote add origin https://github.com/UTA-DataScience/DATA3402.Spring.
2026.git
```

Set push to my fork

If I cloned with HTTPS

```
git remote set-url --push origin https://github.com/MY_USERNAME/DATA3402.Spring.
2026.git
```

If I cloned with SSH

```
git remote set-url --push origin git@github.com:MY_USERNAME/DATA3402.Spring.
2026.git
```

Verify:

```
git remote -v
```

✔️ Fetch → class repo ✔️ Push → my fork

Step 8: First Commit and Push

```
git add .
git commit -m "Initial setup"
```

Possible Error B: No upstream branch

```
The current branch main has no upstream branch
```

Fix:

```
git push --set-upstream origin main
```

Step 9: Non-Fast-Forward Error (Very Common)

Symptom

```
rejected (non-fast-forward)
```

Meaning

- GitHub already has commits
- I must pull once before pushing

Fix

```
git config pull.rebase false
git pull origin main --allow-unrelated-histories
git push
```

✔️ This is **safe and expected**

Weekly Submission Workflow (After Setup)

Every lab:

```
git pull
# work on notebook
git add .
git commit -m "Lab X solution"
git push
```

Common Errors & Quick Fix Table

Error Message	Meaning	Fix
not a git repository	Wrong directory	`cd DATA3402.Spring.2026`
author identity unknown	Name/email missing	`git config --global`
permission denied (publickey)	SSH not set	Add SSH key
non-fast-forward	Remote ahead	`git pull --allow-unrelated-histories`
divergent branches	Git needs merge strategy	`git config pull.rebase false`

One-Line Rule to Remember

> **Pull from class. Push to my fork. Never force push.**

Final Sanity Check

```
git status
git branch -vv
git remote -v
```

If all look correct, I am fully set for the semester.