Working with GitHub Multiple Users and Commits | SSH keys [22nd Feb]

Detailed step-by-step guide to set up Git inside a Docker container, generate SSH keys, configure GitHub authentication, and commit changes. This serves as complete documentation, covering everything from **Docker setup to Git commits**.

🚀 **Git Setup & Commit Guide in Docker**

This guide will help you:

✅ Set up **Git inside a Docker container**

✅ Generate **SSH keys** and configure GitHub authentication

✅ Make **Git commits and push changes** from Docker

1️⃣ **Build & Run the Docker Container**

First, build your Docker image (**if not already built**):

```
docker build --build-arg UID=$(id -u) --build-arg GID=$(id -g) -f dataset_creation.dockerfile -t hrithik_dataset_creation_pipeline .
```

Then, run a container with a mounted directory:

```
docker run -it --rm -v /home/venkat_kesav/hrithik/dataset_creation/:/home/hrithik_sagar/ hrithik_dataset_creation_pipeline bash
```

This mounts your **local project directory** (/home/venkat_kesav/hrithik/dataset_creation/) to /home/hrithik_sagar/ inside the container.

2️⃣ **Install & Configure Git (If Not Installed)**

Inside the container, check if Git is installed:

```
git --version
```

If not installed, install it:

```
sudo apt update && sudo apt install git -y
```

Set your **Git username & email**:

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

Verify the configuration:

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

3️⃣ Generate SSH Keys (If Not Available)

Check if an SSH key already exists:

```
ls -la ~/.ssh
```

If **no key exists**, generate a new SSH key:

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

- **File path**: /home/hrithik_sagar/.ssh/id_rsa
- Press **Enter** for defaults
- **Passphrase?** Optional (press Enter to skip)

This will generate:

- **Private key** → /home/hrithik_sagar/.ssh/id_rsa
- **Public key** → /home/hrithik_sagar/.ssh/id_rsa.pub

4️⃣ Add SSH Key to GitHub

Extract the **public key**:

```
cat ~/.ssh/id_rsa.pub
```

Copy the key, then:

1. Go to **GitHub → Settings → SSH and GPG Keys**
2. Click **New SSH Key**
3. **Paste** the public key
4. Save it

5️⃣ Start SSH Agent & Add Key

Since Docker containers are stateless, you need to **start the SSH agent** and add the key **every time you restart the container**:

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

Verify:

```
ssh-add -l
```

Test SSH connection with GitHub:

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

Expected output:

```
Hi <your_github_username>! You've successfully authenticated, but GitHub
does not provide shell access.
```

6️⃣ Configure SSH for GitHub (If Needed)

Ensure your SSH config file is set up:

```
nano ~/.ssh/config
```

Add the following lines:

```
Host github.com
   IdentityFile ~/.ssh/id_rsa
   StrictHostKeyChecking no
```

I have made the config file something like this:

```
Host tih
   Hostname ssh.github.com
   PreferredAuthentications publickey
   Identityfile /home/hrithik_sagar/my_keys/tih_keys
```

Save and exit (**Ctrl + X → Y → Enter**).

Apply correct permissions:

```
chmod 600 ~/.ssh/config
```

7️⃣ Clone an Existing Repository

To clone a repo using SSH:

```
git clone git@github.com:<your_github_username>/<repo_name>.git
```

Move into the project directory:

```
cd <repo_name>
```

8️⃣ Make Changes & Track Them

Check the current branch:

```
git branch
```

Check the current status:

```
git status
```

To **stage all changes**:

```
git add .
```

To **stage a specific file**:

```
git add <filename>
```

9️⃣ Commit & Push Changes

Commit with a meaningful message:

```
git commit -m "Your commit message"
```

Push changes to GitHub:

```
git push origin main
```

or for other branches:

```
git push origin <branch_name>
```

🔟 Automate SSH Key Addition (Optional)

To avoid manually adding your SSH key each time, add this to your .bashrc:

```
echo 'eval "$(ssh-agent -s)" && ssh-add ~/.ssh/id_rsa' >> ~/.bashrc
```

Then apply changes:

```
source ~/.bashrc
```

Now the SSH key will be loaded automatically in future sessions.

✅ Quick Reference

Action	Command
Build Docker Image	docker build -t hrithik_dataset_creation_pipeline .
Run Docker Container	docker run -it --rm -v /home/venkat_kesav/hrithik/dataset_creation/:/home/hrithik_sagar/ hrithik_dataset_creation_pipeline bash

Install Git	sudo apt update && sudo apt install git -y
Set Git Config	git config --global user.name "Your Name" git config --global user.email "your_email@example.com"
Generate SSH Key	ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
Add SSH Key to Agent	eval "$(ssh-agent -s)" ssh-add ~/.ssh/id_rsa
Test GitHub SSH Connection	ssh -T git@github.com
Clone Repo	git clone git@github.com:<your_github_username>/<repo_name>.git
Commit & Push	git add . git commit -m "Your commit message" git push origin main

🎯 Final Notes

Now, you're all set to **commit & push changes to GitHub from Docker**! 🚀

If you restart your Docker container, make sure to **restart ssh-agent and add your key** before pushing.

▼ Unwanted stuff:

We have created a Public and private key in the server (not in GitHub). Now, get that public key, and then in the server, we used ssh-agent commands to add the private key to the config.

▼ Commit history

```
hrithik_sagar@e18c6c97d318:~$ history
1  clear
2  which python
3  clear
4  conda
5  python --version
6  conda create --name .dataset_creation python=3.11
7  conda activate .dataset_creation
8  conda activate .dataset_creation
9  conda init bash
10  exec bash
11  clear
12  conda activate .dataset_creation
13  clear
```

```
14  clear
15  ls
16  ls -a
17  cd Indic-GR-Dataset-Creation-Pipeline/
18  ls
19  git status
20  git rm data/testset
21  git rm -r data/testset
22  git status
23  git commit -m "mast delete chesinai"
24  git config user.email "
hrithik.sagar@tihiitb.org"
25  git config
user.name "Hrithik sagar"
26  clear
27  git commit -m "chala delete chesinai"
28  clear
29  git push
30  ls
31  ls o-a
32  ls -a
33  cd ..
34  ls -a
35  cd .ssh/
36  ls
37  cat known_hosts
38  clear
39  ls
40  cd ..
41  ls
42  ls -a
43  clear
44  ls
45  clear
46  clear
47  ls
48  clear
49  clear
```

```
50  mkdir my_keys
51  clear
52  cd my_keys/
53  ls
54  ssh-keygen --help
55  clear
56  ssh-keygen -t ed25519 -f tih_keys
57  ls
58  cat tih_keys.pub
59  cd ..
60  ls -a
61  clear
62  ls -a | grep ssh
63  cd .ssh/
64  ls
65  vim config
66  cd ..
67  cd my_keys/
68  ls
69  pwd
70  cd ..
71  cd .ssh
72  vim config
73  ssh -T git@tih
74  clear
75  ssh-agent eval
76  cd
77  clear
78  eval "$(ssh-agent -s)"
79  ssh-agent --help
80  ssh-add my_keys/tih_keys
81  clear
82  cd Indic-GR-Dataset-Creation-Pipeline/
83  git pus
84  git push
85  ls
86  cd ..
87  cd my_keys/
```

```
88  ls
89  clear
90  ls
91  clear
92  clear
93  ls
94  clear
95  cd ..
96  mkdir delete_later
97  cd delete_later/
98  clear
99  git clone
git@github.com:githubtraining/hellogitworld.git
100  git clone git@tih:githubtraining/hellogitworld.git
101  ls
102  rm -r hellogitworld/
103  clear
104  rm -rf hellogitworld/
105  ls
106  clear
107  git clone
git@github.com:hrithiksagar-tih/delete_later.git
108  rm -rf delete_later/
109  clear
110  ls
111  cd
112  cd .ssh/
113  ls
114  rm known_hosts
115  ls
116  clear
117  cd
118  cd delete_later/
119  git clone
git@github.com:hrithiksagar-tih/delete_later.git
120  clear
121  git clone git@tih:hrithiksagar-tih/delete_later.git
122  rm -rf delete_later/
```

```
123  git clone git@tih:hrithiksagar-tih/delete_later.git
124  clear
125  exit
126  history
(base) hrithik_sagar@e18c6c97d318:~$
```