✅ 1. Generate SSH Keys for Both Accounts

🧑 Personal Account

```
ssh-keygen -t ed25519 -C "your_personal_email@example.com"
# Save as: C:\Users\YourName\.ssh\id_ed25519_personal
```

💼 Work Account

```
ssh-keygen -t ed25519 -C "your_work_email@company.com"
# Save as: C:\Users\YourName\.ssh\id_ed25519_work
```

✅ 2. Add SSH Keys to `ssh-agent`

Start the agent(Git Bash) and add both keys:

```
eval `ssh-agent -s`

ssh-add ~/.ssh/id_ed25519_personal
ssh-add ~/.ssh/id_ed25519_work
```

On Powershell(require administrator priviledge)

```
ssh-add C:/Users/YourName/.ssh/id_ed25519_personal
ssh-add C:/Users/YourName/.ssh/id_ed25519_work
```

🧪 Verify It Works

```
ssh-add -l
```

💡 Optional: Automate Agent Start in Git Bash

If you always use Git Bash, add this to the end of your `~/.bashrc` (or `~/.bash_profile` if you're using that instead):

```
# Start ssh-agent if not already running
eval "$(ssh-agent -s)" > /dev/null
```

✅ 3. Add Public Keys to GitHub Accounts

Go to each GitHub account:

- Profile → Settings → **SSH and GPG Keys**
- Click **"New SSH Key"** and paste contents of:

```
cat ~/.ssh/id_ed25519_personal.pub
cat ~/.ssh/id_ed25519_work.pub
```

✅ 4. Configure SSH `config` File

Edit (or create) `~/.ssh/config`:

```
# Personal Account
Host github-personal
  HostName github.com
  User git
  IdentityFile ~/.ssh/id_ed25519_personal

# Work Account
Host github-work
  HostName github.com
  User git
  IdentityFile ~/.ssh/id_ed25519_work
```

✅ 5. Clone Repos with Custom Hosts

If repository already exists

- **Personal:**

```
git clone git@github-personal:yourusername/repo.git
```

- **Work:**

```
git clone git@github-work:yourworkusername/repo.git
```

If creating new repository

```
git init

git add .
git commit -m "Initial commit"


git remote add origin git@github-personal:yourusername/my-new-project.git

git push -u origin main  # or master, depending on your branch
```

Same for Work Account Just change the remote to:

```
git remote add origin git@github-work:yourworkusername/work-repo.git
```

✅ 6. Set Git User Per Repository

Navigate into your cloned repo:

```
cd repo/
git config user.name "Your Personal Name"
git config user.email "your_personal_email@example.com"
```

For work repo:

```
cd work-repo/
git config user.name "Your Work Name"
git config user.email "your_work_email@company.com"
```

✅ 7. Verify

Inside a repo:

```
git remote -v
git config user.name
git config user.email
```

Push test commits to confirm correct identity is used.