

# jetson에서 git 사용

## ✓ 하나의 PC에서 GitHub 계정 2개(개인/회사) 분리 사용하기 요약

### 1. 🔑 각 계정별 SSH 키 생성

# 개인 계정용

```
ssh-keygen -t ed25519 -C "personal@example.com" -f ~/.ssh/id_ed25519_personal
```

# 회사 계정용

```
ssh-keygen -t ed25519 -C "work@example.com" -f ~/.ssh/id_ed25519_work
```

GitHub 웹에서 각각의 계정에 해당 공개키 ( `.pub` ) 등록하기:

- <https://github.com/settings/keys>

### 2. ⚙️ `~/.ssh/config` 설정

```
# ~/.ssh/config
```

```
# 개인용
```

```
Host github-personal
```

```
  HostName github.com
```

```
  User git
```

```
  IdentityFile ~/.ssh/id_ed25519_personal
```

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

| Host는 내가 붙인 별명이고, 이후 git clone할 때 사용됨

### 3. 🌀 클론할 때 계정에 맞는 Host 사용

```
# 개인용
git clone git@github-personal:username/repo.git

# 회사용
git clone git@github-work:orgname/repo.git
```

| username/repo.git은 GitHub 상의 실제 사용자/레포 이름

### 4. 👤 커밋할 사람 계정 설정 ( **-local** 만 사용!)

```
git config user.name "Alice"
git config user.email "alice@example.com"

# 또는 다른 사람
git config user.name "Bob"
git config user.email "bob@company.com"
```

전역(--global) 설정은 제거하거나 사용하지 않기!

## 5. 🚀 푸시 시 계정 자동 분리됨

- SSH `Host` 별로 연결되므로, `push` 시 해당 계정의 SSH 키로 GitHub에 접속
- `origin` remote가 올바르게 설정되어 있어야 함

```
git remote -v
```

필요 시:

```
git remote set-url origin git@github-work:orgname/repo.git
```

## 🧪 연결 테스트 팁

```
ssh -T git@github-personal  
ssh -T git@github-work
```

정상 연결되면 해당 GitHub 계정의 환영 메시지가 출력됨