

# Git mit mehreren Schlüsseln auf einem Client

1. Es besteht die ja die Möglichkeit das man einen privaten Github Account und einen Firmen Github Account hat.

Wie unterteilt man dieses? Es gibt Gott sei dank im .ssh verzeichnis eine Datei die nennt sich config.

Diese öffnen wir im Terminal mit nano.

Die Variable Host ist ein alias der nachher für den Gitservernamen gewählt wird.

```
nano ~/.ssh/config

# Company account
Host company
HostName github.com
PreferredAuthentications publickey
IdentityFile ~/.ssh/id_rsa_company

# Personal account
Host personal
HostName github.com
PreferredAuthentications publickey
IdentityFile ~/.ssh/id_rsa_personal
```

2. Sollte ein neuer SSH Schlüssel auf System kopiert werden, also am System nicht selbst erstellt dann müssen der Private und public key kopiert werden. Ansonsten diesen Teil überspringen

Beispiel:

```
#Private key
cp mein_privat_schlüssel_woher_auch_immer ~/.ssh/id_rsa_immohacker

#Public Key
cp mein_public_schlüssel_woher_auch_immer ~/.ssh/id_rsa_immohacker.pub
```

Nun die rechte setzten

```
chmod 600 ~/.ssh/id_rsa_immohacker  
chmod 600 ~/.ssh/id_rsa_immohacker.pub
```

und den Schlüssel in den ssh agent einspielen

```
ssh-add ~/.ssh/id_rsa_immohacker
```

3. Nun die Repo neu im Git Verzeichnis festlegen.

Damit das allerdings geht müssen falls vorhanden der globale Name aus dem Git entfernt werden. Dazu

```
git config --global --unset user.name  
git config --global --unset user.email
```

nun die Neue Repo mit allem was dazu gehört festlegen.

```
git remote add origin git@github.com:immohacker/ansible-role-meinerolle  
in  
git remote add origin git@personal:immohacker/ansible-role-meinerolle  
ändern  
Gleichzeitig den richtigen Benutzernamen und Email setzen  
git config user.email "meinepersonalemail"  
git config user.name "immohacker"
```

Und in der Firmen Git Verzeichnis

```
git remote add origin git@github.com:meincompanyname/ansible-role-meinerolle  
in  
git remote add origin git@company:meincompanyname/ansible-role-meinerolle  
ändern  
Gleichzeitig den richtigen Benutzernamen und Email setzen  
git config user.email "meinefirmenemail"  
git config user.name "meinfirmenbenutzername"
```

Ob Alles geklappt hat, kann man sich mit anschauen

```
git config -l --show-origin
```

Ausgabe:

```
file:.git/config core.repositoryformatversion=0
file:.git/config core.filemode=true
file:.git/config core.bare=false
file:.git/config core.logallrefupdates=true
file:.git/config user.email=xxxxxx
file:.git/config user.name=immohacker
file:.git/config remote.origin.url=git@git_immohacker:immohacker/ansible-collection-general
file:.git/config remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
```

--

---

Version #4

Erstellt: 10 Juli 2022 07:58:07 von Admin

Zuletzt aktualisiert: 12 August 2022 18:21:28 von Admin