

Ansible Collection zu Github hinzufügen

1. Erstmal die Repo mit der Namenskonvention anlegen: Siehe Abschnitt:

Eine Repo in Github anlegen

2. Im Terminal in den Ordner der Collection gehen dort folgenden Befehle eingeben und das Git erstmalig einzurichten.

Falls mehre Git Accounts auf einem Rechner benutzt werden sollen siehe Abschnitt :

Git mit mehreren Schlüsseln auf einem Client

```
#damit wird das Verzeichnis für git vorbereitet
git init
#das komplette verzeichnis samt unterverzeichnisse zum syncen markieren
git add .
#Nun einen Verwendungszweck für das Comitten also änderung hochschieben vergeben. Beim
ersten mal Initial Commit, ansonsten die Änderungen
git commit -m 'Initial Commit'
#den branch auswählen
git branch -M main
#den Remotepfad angeben. Syntax besteht aus benutzernamen/collectionname
git remote add origin git@github.com:immohacker/ansible-collection-meinname
#Nur wenn mehre Github Accounts verwendet werden hier noch Benutzername und Email
übergeben
Gleichzeitig den richtigen Benutzername und Email setzten
git config user.email "meinepersonalemail"
git config user.name "immohacker"
#Nur wenn mehre Github Accounts verwendet werden hier noch Benutzername und Email
übergeben ende
git push -u origin main
```

3. Collection aktualisieren. Wenn Änderungen ins Git geschoben werden sollen, braucht man nicht nochmals alles angeben.

Ein paar Befehle reichen:

```
#wieder das ganze Verzeichnis samt Unterverzeichnisse
git add .
#Einen Namen für das Commit vergeben
git commit -m 'update irgendwas'
#Den Branch festlegen
git branch -M main
#Hochschieben
git push -u origin main
#fertig.
```

Man kann sich das auch in eine sh Datei packen und muss dann nur noch den Namen fürs commit übergeben

Beispiel

```
nano update.sh
```

```
#!/bin/bash
#wieder das ganze Verzeichnis samt Unterverzeichnisse
git add .
#Einen Namen für das Commit vergeben
git commit -m '$1'
#Den Branch festlegen
git branch -M main
#Hochschieben
git push -u origin main
#fertig.
```

```
chmod +x update.sh
```

```
./update.sh "Mein Commit"
```

4. möchte man eine remote Repo komplett überschreiben, weil zum Beispiel was gelöscht hat im Remote aber dann doch das Repo vom lokalen hinzufügen möchte gibts den Parameter force.

Somit wird die Remote Repo komplett überschrieben

```
git push --force -u origin main
```

Damit wird der lokale stand vom Remote überschrieben.

```
git pull --force -u origin main
```

Alle Commits löschen und die vom lokalen aufs Remote Repo schieben.
Damit wird das .git Verzeichnis gelöscht und neuerstellt.

```
#Git Verzeichnis löschen
rm -rf .git

#Git Neu init
git init

#Repo Verbinden in unserem beispiel die ansible collection, sollte man mehrere Benutzer haben dann
mit dem namen davor
#original mit nur einem Benutzernamern bei git, Hier wählen, beides geht nicht
git remote add origin https://github.com/immohacker/ansible-collection-general
#oder mit einem anderen account bei git
git remote add origin git@git_immohacker:immohacker/ansible-collection-general

#Nun noch die Credentials neu setzten, wenn mehrere Benutzer benutzt weden
#ansonsten überspringen, weil diese dann ja schon global eingetragen sind.
git config user.email "meinepersonalemail"
git config user.name "immohacker"

#Remotestand anschauen
git remote -v

#Output:
#origin git@git_immohacker:https://github.com/immohacker/ansible-collection-general.git (fetch)
#origin git@git_immohacker:https://github.com/immohacker/ansible-collection-general.git (push)

#branch festlegen
git checkout -b main

#Alle Dateien hinzufügen
git add .

#Commit namen vergeben und alle dateien zum change hinzufügen (-a)
git commit -m "Initial commit"
git push -f origin main
```

Fertig. Nur ein Commit und alles neu erstellt.

Diese Collection Repo Ansible Galaxy verbinden

Version #13

Erstellt: 10 Juli 2022 07:25:48 von Admin

Zuletzt aktualisiert: 13 August 2022 16:52:46 von Admin