

Collections

Collections Benutzen, erstellen und bei Github hochladen

- [Collection Grundgerüst erstellen](#)
- [Eine Standalone Rolle in eine Collection migrieren](#)
- [Collection bauen](#)

Collection Grundgerüst erstellen

Um eine Collection zu erstellen nimmt man ein Verzeichnis für alle Seine Collections. zum Beispiel, diese Verzeichnis. Wenn nicht besteht dann anlegen.

```
mkdir ~/mein-ansible/project1/collections/  
#in das Verzeichnis rein gehen.  
cd ~/mein-ansible/project1/collections/
```

Nun kann mittels ansible Galaxy Befehl eine collection angelegt werden
Der Aufbau einer Collection ist immer der benutzername.collectionname
Wir nennen unsere Collection mal general
also der Befehl lautet:

```
ansible-galaxy collection init immohacker.general
```

Hier mit wird dann folgende Verzeichnisstruktur erstellt:

```
~/mein-ansible/project1/collections/immohacker/general/
```

Die Verzeichnisstruktur sieht dann so aus:

```
.  
├─ docs  
├─ galaxy.yml  
├─ plugins  
| └─ README.md  
├─ README.md  
└─ roles
```

in der galaxy.yml kommen die Daten und die Versionsnummern rein.
bei jeder Änderung an der Collection muss die Versionsnummer in der Datei erhöht werden.
Diese Datei dient

Die wichtigsten Punkte in der Datei sind:

```
namespace: name des darunter liegenden spaces. Ich nehme immer den namen vom github Konto in diesem
Fall immohacker
name : name der collection z.b general
version:
authors : Unser Name
Description : Beschreibung der Collection, also was macht diese.
```

dependencies: die Collections wovon diese Collection abhängig sein könnte.

Allerdings sollte man es so bauen, das eine Colelction autark und nicht von anderen Collections abhängig ist.

ansonsten wenn es geht die Abhängigkeiten mit in die Collection einbauen, Nachteil die Abhängigkeiten werden dann halt nur nicht mehr automatisch gepflegt, diese muss man dann jedes mal selbst aktualisieren.

Aber falls nicht anders geht, hier ein Beispiel für Abhängigkeiten

Beispiel

```
dependencies:
  #größer oder gleich 1.0.0
  "other_namespace.collection1": ">=1.0.0"
  #größer oder gleich 2.0.0 und kleiner als 3.0.0
  "other_namespace.collection2": ">=2.0.0,<3.0.0"
  "anderson55.my_collection": "*" # note: "*" selects the highest version available
```

repository: Git url wenn vorhanden, meistens ja.

Zumindest wenn man veröffentlichen will, sonst natürlich nicht. Dann leer lassen.

documentation: url zur Docu

homepage: wenn nicht vorhanden leer lassen

issues: github issue tracker, wenn nicht vorhanden auch hier leer lassen.

Im rootverzeichnis kann optional noch eine

requirements.txt

erstellt werden.

In dieser werden alle Modul und Pluginsabhängigkeiten eintragen, falls die Collection welche hat.

Beispiel Aufbau:

```
# Module deps, useful for editor intellisense, etc.
ansible-core==2.13.1
requests==2.28.1
proxmoxer==1.3.1

# Linting & Formatting
```

```
ansible-lint==6.3.0
pylint==2.14.4
autopep8==1.6.0
pre-commit==2.19.0
```

```
# Testing
```

```
tox==3.25.1
```

```
# Utility packages needed for scripts
```

```
pyyaml==6.0.0
```

```
packaging==21.3
```

Das wars auch schon. Schon kann darin gearbeitet werden.

Möchte man das ganze noch in ein Git Repo syncen dann siehe Abschnitt :

[Ansible Collection zu Github hinzufügen](#)

Eine Standalone Rolle in eine Collection migrieren

Dazu einfach die den Rollen Inhalt in eine Neue Rolle im Ornder Roles kopieren.

Beispiel:

```
mv -r /locales_ansible_arbeitsverzeichnis/roles/meinerolle_1  
/meine/collections/immohacker/general/roles/
```

nun noch in der galaxy.xml, die abhängigkeiten eintragen die die Rolle hat.

in diesem Beispiel eine andere einzelne Rolle. Die Abhängigkeiten können in der Datei meta/main.yml im Rollenverzeichnis nachgeschaut werden und übertragen werden.

Beispiel:

```
....  
dependencies:  
- role: stefanhacker.ansible_extract_iso  
tags: [never]
```

Das Tag never bedeutet lade die Rolle für die abhängigkeit runter, führe diese aber nicht aus
Rollen die als abbhänngigkeit eingetragen werden, werden standard mäßig auch gleich nach dem download ausgeführt.

unter Umständen ist dies aber nicht gewünscht, weil custom parameter übergeben werden müssen.

Das wars auch schon.

Schon hat unsere collection die Erste Rolle. Nun kann sie gebaut werden. Siehe -> [Collection bauen](#)

Collection bauen

Eine Collection wird damit sie Veröffentlicht wird / installierbar ist als tarball erstellt.

Dazu in das Verzeichnis gehen

In unserem Beispiel:

```
cd ~/mein-ansible/project1/collections/immohacker/general/
```

Dort dann den befehl zum bauen ausführen

```
ansible-galaxy collection build
```

Ausgabe:

```
Created collection for immohacker.general at /home/duffy/ansible-strange-  
hosting/collections/immohacker/general/immohacker-general-1.0.0.tar.gz
```

Nun haben wir eine tar Datei die wir verteilen können. In der tar Datei steht im Namen auch immer die Versionsnummer die aus der galaxy.yml gelesen wird.