

Tutorial 7 Utiliser les exemples de Github.

Table des matières

- [Objectifs](#)
- [Ressources](#)
- [Pré-Requis](#)

Objectifs

Dans ce TP ,nous allons utiliser les exemples fournis avec la collection Ibmi ansible.Ansible sur notre IBMi. Pour rappel :

- [site Github](#)
- [doc reference](#) Ainsi nous allons :
- utiliser l'exemple [ibmi-sysval-sample.yml](#)
- utiliser l'exemple [ibmi-cl-command-sample.yml](#) en utilisant une variable d'inventaire pour le nom de la bibliothèque
- Analyser les résultats

⚠ ⚠ attention Nous allons utiliser des ressources que nous n'avons pas écrites ! Il est très utile de lire ces ressources avant de les lancer sur une machine mais si celle_çi se trouve être une machine de test.

Ressources

- Environnement
- Temps : 60 mn.

Pré-Requis

- avoir configurer Ansible pour travailler avec notre l'environnement sur l'IBMI.
[TP05 - Hello IBMi](#)
- avoir installer la collectionIBMi.
[TP06 - Du vrai IBMi](#)

Énoncé

Etape 1 créer le projet TP07 en dupliquant le dossier TP06

Etape 2 [ibmi-sysval-sample.yml](#).

1. Trouver l'exemple.

- sur github. site Github

IBM/ansible-for-i: the tool is to p x Power IBM i collection for Ansible x +

github.com/IBM/ansible-for-i/tree/devel

Applications gmail SG ABA English - The b... 630lou accord : Piano blues Bui

Search or jump to...

Pull requests

Issues

Co

IBM / ansible-for-i Public

<> Code

Issues 30

Pull requests 2

Actions

Projects

devel 34 branches 0 tags

LiJunBJZhu Merge pull request #147 from IBM/git-sync ...

docs	Travis build: 6597
playbooks	Travis build: 6597
plugins	Travis build: 6599
roles	Travis build: 6568
tests	Travis build: 6226

VOLUBIS 2 / 10

github.com/IBM/ansible-for-i/tree/devel/playbooks

Applications gmail SG ABA English - The b... catilou accord : Piano blues Building and Testin... UsingGenera

Code

devel + 🔍

Go to file t

> docs

▼ playbooks

- > enable-ansible-for-i
- > enable_offline_ibmi
- > ibmi-install-nodejs
 - hosts_ibmi.ini
 - ibmi-check-default-passwords.y...
 - ibmi-cl-command-sample.yml
 - ibmi-fix-check.yml
 - ibmi-fix-group-check.yml
 - ibmi-fix-repo-cum-package.yml
 - ibmi-fix-repo-download-status....
 - ibmi-fix-repo-lv1.yml
 - ibmi-fix-repo-ptf-group.yml
 - ibmi-fix-repo-single-ptf.yml
 - ibmi-sql-sample.yml
 - ibmi-sqlite3-sample.yml
 - ibmi-sysval-sample.yml**
 - install_fix_repo_dependencies.yml
 - query-iasp-sample.yml

ansible-for-i / playbooks /

- ibmi-cl-command-sample.yml Tra
- ibmi-fix-check.yml Tra
- ibmi-fix-group-check.yml Tra
- ibmi-fix-repo-cum-package.yml Tra
- ibmi-fix-repo-download-status.yml Tra
- ibmi-fix-repo-lv1.yml Tra
- ibmi-fix-repo-ptf-group.yml Tra
- ibmi-fix-repo-single-ptf.yml Tra
- ibmi-sql-sample.yml Tra
- ibmi-sqlite3-sample.yml Tra
- ibmi-sysval-sample.yml** Tra
- install_fix_repo_dependencies.yml Tra
- query-iasp-sample.yml Tra

LiJunBJZhu Travis build: 6597

Code Blame 127 lines (110 loc) · 4.81 KB

```
1 # Copyright (c) IBM Corporation 2019, 2020
2 # Apache License, Version 2.0 (see https://opensource.org/licenses/Apache-2.0)
3
4 # Description:
5 # The sample file is to provide an example about how to query and check system values
6
7 # Restriction:
8 # You must have either all object (*ALLOBJ) or audit (*AUDIT) special authority to retrieve
9 # the values for QAUDCTL, QAUDENDACN, QAUDFRCLVL, QAUDLVL, QAUDLVL2, and QCRTOBJAUD
10
11 # Parameters:
12 # sysvalue -- The list containing the system values to be checked
13 ---
14 - hosts: all
15   gather_facts: false
16   collections:
17     - ibm.power_ibmi
18
19   tasks:
20     - block:
21         - name: Display a system value
22           ibmi_sysval:
23             sysvalue:
24               - {'name': 'qccsid'}
```

- sur la machine dans la collection. Utiliser le lien défini sur la collection dans le browser IFS de C4I.

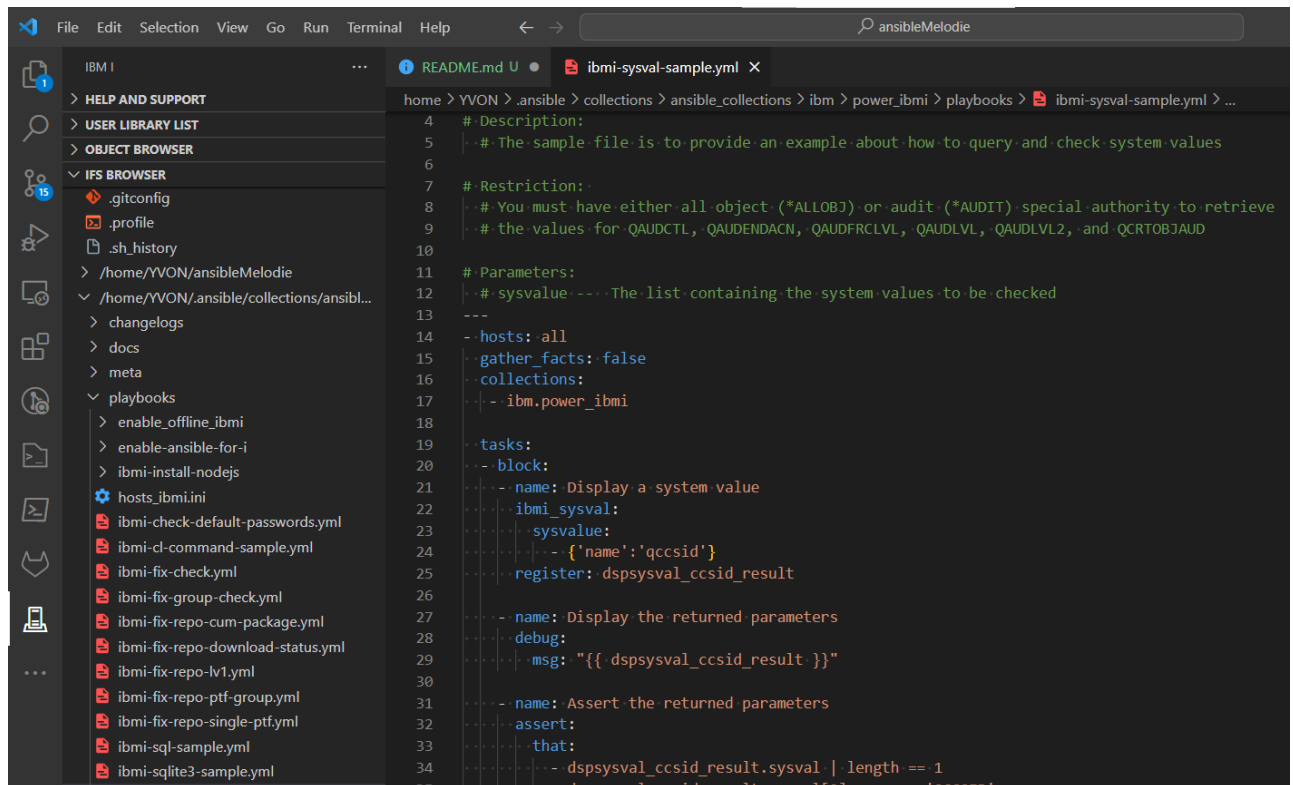
The screenshot shows the IBM IFS Browser interface. On the left, the 'IFS BROWSER' pane displays a file tree. Handwritten yellow arrows and numbers highlight the following elements:

- 1**: Points to the `.gitconfig` file.
- 2**: Points to the `changelogs` directory.
- 3**: Points to the `playbooks` directory.
- 4**: Points to the `ibmi-sysval-sample.yml` file.

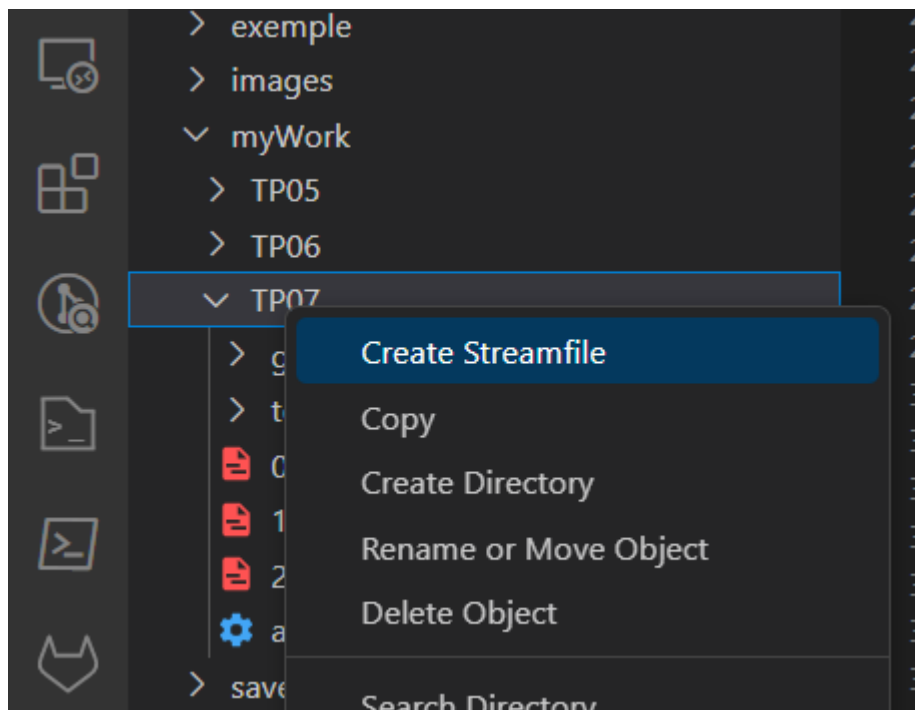
The right pane displays the content of the `README.md` file. The text includes sections for Objectifs, Ressources, Pré-Requis, and Énoncé. The `ibmi-sysval-sample.yml` file is mentioned in the 'Étape 1' section.

At the bottom, a tooltip shows the file path and metadata for `ibmi-sysval-sample.yml`:

```
/home/YVON/.ansible/collections/ansible_collections/ibm/power_ibmi/playbooks/ibmi-
-sysval-sample.yml
Size: 4927
Modified: 2023-06-16 13:14:38
Owner: YVON
```



1. dupliquer le playbook dans votre projet si il vous convient. plusieurs façons de faire copie du fichier via le browser IFS (clic droit sur le fichier) ou création d'un nouveau fichier dans le projet et CTRL-a CTRL-c sur la source et ctrl-a CTRL-V sur la cible



```

/home/YVON/ansibleMelodie/myWork/TP07/30_controleValeursSystemes.yml
Name of new streamfile (Press 'Enter' to confirm or 'Escape' to cancel)
GitHub > README.md > # Tutorial 7 Utiliser les exemples de Github. > ## Enoncé > ### Etape 2

home > YVON > ansibleMelodie > myWork > TP07 > 30_controleValeursSystemes.yml > {} 0 > [ ] tasks > {} 0 > [ ] block > {} 13 > {} debug
93     ....- { 'name': 'qmaxsgnacn', 'expect': '3' }
94     ....- { 'name': 'qccsid' }
95     ....register: chksysval_result
96
97     ....- name: Display the many returned parameters
98     ....debug:
99     ....msg: "{{ chksysval_result }}"
100
101     ....- name: Assert the many returned parameters
102     ....assert:
103     ....that:
104     ....- chksysval_result.sysval | length == 3
105     ....- chksysval_result.sysval[0].name == 'QMAXSIGN'
106     ....- chksysval_result.sysval[0].type == '8A'
107     ....- chksysval_result.sysval[0].compliant | bool
108     ....- chksysval_result.sysval[1].name == 'QMAXSGNACN'
109     ....- chksysval_result.sysval[1].type == '4A'
110     ....- chksysval_result.sysval[1].value == chksysval_result.sysval[1].expect
111     ....- chksysval_result.sysval[1].compliant | bool
112     ....- chksysval_result.sysval[2].name == 'QCCSID'
113     ....- chksysval_result.sysval[2].type == '10i0'
114     ....- chksysval_result.sysval[2].value | int >= 0 and chksysval_result.sysval[2].value | int <= 655
115
116     ....- name: Check with equal_as_list option
117     ....ibmi_sysval:
118     ....sysvalue:
119     ....- { 'name': 'QATNPGM', 'expect': 'QEZMAIN..QSYS' }
120     ....- { 'name': 'QATNPGM', 'expect': 'QSYS..QEZMAIN' }

```

2. Contrôler sa possible execution

- Lancer en check ,diff ...
 - diff `ansible-playbook 30_controleValeursSystemes.yml -i 00_inventory.yml --diff -check` diff `ansible-playbook 30_controleValeursSystemes.yml -i 00_inventory.yml -check` modifier les valeurs testées dans le playbook pour qu'ils passent les controles. ⚠ NE PAS MODIFIER LES VALEURS SYSTEMES DE LA MACHINE !!!
- Lancer pas à pas `ansible-playbook 30_controleValeursSystemes.yml -i 00_inventory.yml -step`
 1. ajouter la clause `ignore_errors: true` pour laisser passer les controles ko
 2. Lancer en global `ansible-playbook 30_controleValeursSystemes.yml -i 00_inventory.yml correction`

Etape 2 ibmi-cl-command-sample.yml.

objectif créer une bibliothèque avec un nom de la variable d'inventaire `nomLIBTravail` .

1. Trouver l'exemple.

- sur github. site Github

IBM/ansible-for-i: the tool is to p

Power IBM i collection for Ansible

github.com/IBM/ansible-for-i/tree/devel

Applications gmail SG ABA English - The b... 630lou accord : Piano blues Bui

Search or jump to...

Pull requests Issues Co

IBM / ansible-for-i Public

<> Code Issues 30 Pull requests 2 Actions Projects

devel 34 branches 0 tags

LiJunBJZhu Merge pull request #147 from IBM/git-sync

docs	Travis build: 6597
playbooks	Travis build: 6597
plugins	Travis build: 6599
roles	Travis build: 6568
tests	Travis build: 6226

github.com/IBM/ansible-for-i/tree/devel/playbooks

Applications gmail SG ABA English - The b... catilou accord : Piano blues Building and Testin... UsingGenera

Code

devel + 🔍

Go to file t

> docs

▼ playbooks

- > enable-ansible-for-i
- > enable_offline_ibmi
- > ibmi-install-nodejs
 - hosts_ibmi.ini
 - ibmi-check-default-passwords.y...
 - ibmi-cl-command-sample.yml
 - ibmi-fix-check.yml
 - ibmi-fix-group-check.yml
 - ibmi-fix-repo-cum-package.yml
 - ibmi-fix-repo-download-status....
 - ibmi-fix-repo-lv1.yml
 - ibmi-fix-repo-ptf-group.yml
 - ibmi-fix-repo-single-ptf.yml
 - ibmi-sql-sample.yml
 - ibmi-sqlite3-sample.yml
 - ibmi-sysval-sample.yml**
 - install_fix_repo_dependencies.yml
 - query-iasp-sample.yml

ansible-for-i / playbooks /

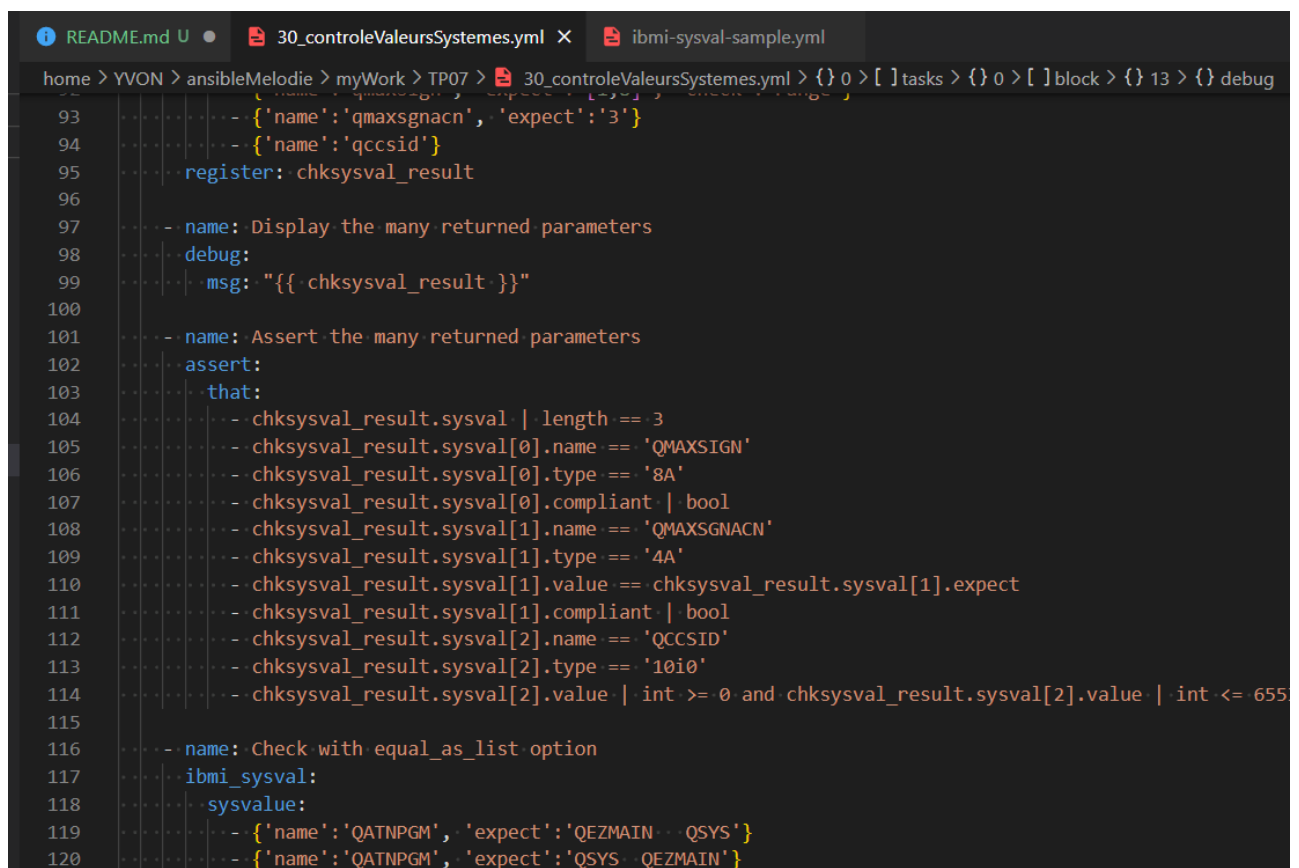
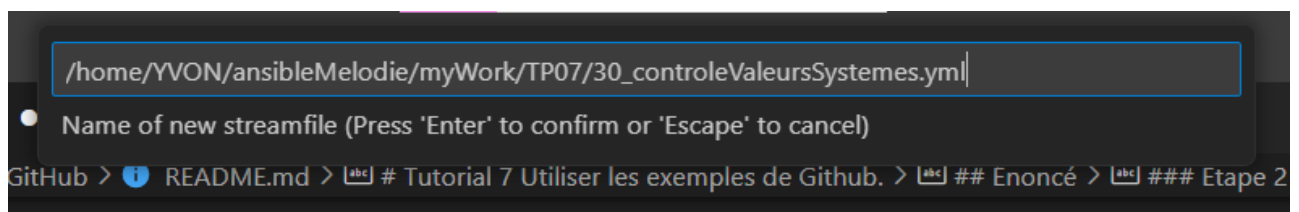
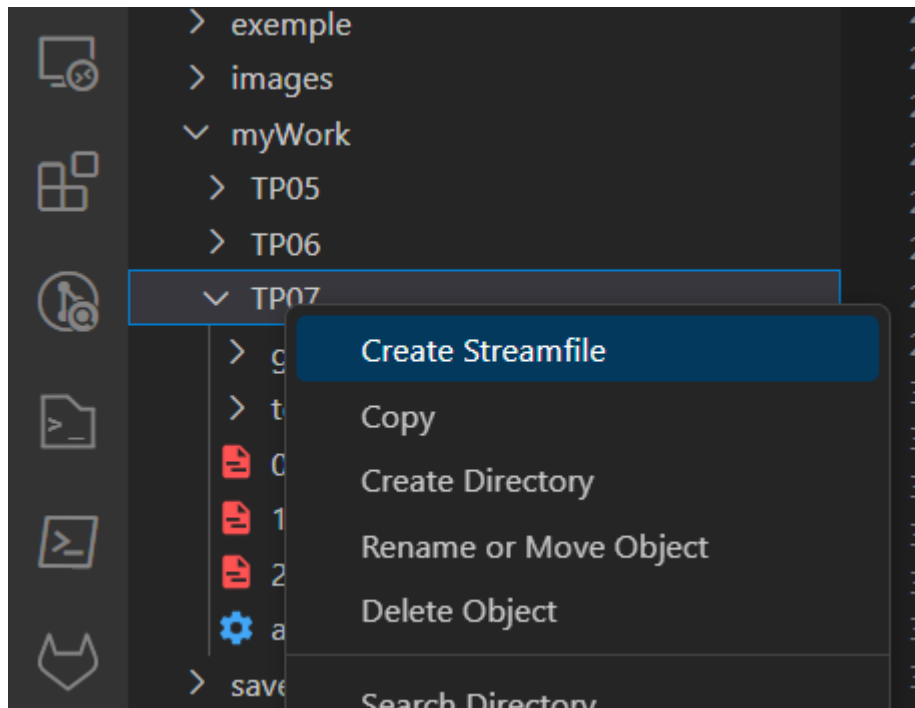
- ibmi-cl-command-sample.yml Tra
- ibmi-fix-check.yml Tra
- ibmi-fix-group-check.yml Tra
- ibmi-fix-repo-cum-package.yml Tra
- ibmi-fix-repo-download-status.yml Tra
- ibmi-fix-repo-lv1.yml Tra
- ibmi-fix-repo-ptf-group.yml Tra
- ibmi-fix-repo-single-ptf.yml Tra
- ibmi-sql-sample.yml Tra
- ibmi-sqlite3-sample.yml Tra
- ibmi-sysval-sample.yml** Tra
- install_fix_repo_dependencies.yml Tra
- query-iasp-sample.yml Tra

LiJunBJZhu Travis build: 6597

Code Blame 127 lines (110 loc) · 4.81 KB

```
1 # Copyright (c) IBM Corporation 2019, 2020
2 # Apache License, Version 2.0 (see https://opensource.org/licenses/Apache-2.0)
3
4 # Description:
5 # The sample file is to provide an example about how to query and check system values
6
7 # Restriction:
8 # You must have either all object (*ALLOBJ) or audit (*AUDIT) special authority to retrieve
9 # the values for QAUDCTL, QAUDENDACN, QAUDFRLVL, QAUDLVL, QAUDLVL2, and QCRTOBJAUD
10
11 # Parameters:
12 # sysvalue -- The list containing the system values to be checked
13 ---
14 - hosts: all
15   gather_facts: false
16   collections:
17     - ibm.power_ibmi
18
19   tasks:
20     - block:
21         - name: Display a system value
22           ibmi_sysval:
23             sysvalue:
24               - {'name': 'qccsid'}
```


- sur la machine dans la collection. Utiliser le lien défini sur la collection dans le browser IFS de C4I.
1. dupliquer le playbook dans votre projet si il vous convient. plusieurs façons de faire copie du fichier via le browser IFS (clic droit su le fichier) ou création d'un nouveau fichier dans le projet et CTRL-a CTRL-c sur la source et ctrl-a CTRL-V sur la cible



2. ajouter la variable d'inventaire et modifier le playbook pour qu'il en tienne compte.
 3. Contrôler sa possible execution
- Lancer en check ,diff ...
 - `diff ansible-playbook 40_creationLibTravail.yml -i 00_inventory.yml --diff -check`
`diff ansible-playbook 40_creationLibTravail.yml -i 00_inventory.yml --check`
modifier les valeurs testées dans le playbook pour qu'ils passent les controles. ⚠ NE PAS
MODIFIER LES VALEURS SYSTEMES DE LA MACHINE !!!
 - Lancer pas à pas `ansible-playbook 40_creationLibTravail.yml -i 00_inventory.yml --step`
 - 1. Lancer en global `ansible-playbook 40_creationLibTravail.yml -i 00_inventory.yml` relancer
[correction](#)

Conclusion et feed-back

Correction

Idées

-