

## Lab 3 – Les commandes Ad-hoc

### Introduction

Il existe plusieurs tâches dans Ansible pour lesquelles vous n'avez pas besoin d'écrire un Playbook Ansible; vous pouvez simplement exécuter une commande ad-hoc pour cette tâche. Il s'agit d'une commande à une seule ligne pour effectuer une seule tâche sur l'hôte cible.

### Commandes Ad-hoc

#### Commandes de base

La commande ad-hoc ci-dessous exécute un module **ping** sur toutes les hôtes du fichier inventaire. Ici **-m** est l'option pour un module.

```
ansible all -m ping
```

#### Example of output:

```
app1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
app2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

La commande mentionnée ci-dessous exécute le module setup (Gather facts) sur un groupe d'hôtes, l'option **-a** ou **--args** sert à définir les arguments du module exécuté.

```
ansible all -m setup -a "filter=ansible_distribution*"
```

#### Exemple d'output:

```
app1 | SUCCESS => {
  "ansible_facts": {
    "ansible_distribution": "CentOS",
    "ansible_distribution_file_parsed": true,
    "ansible_distribution_file_path": "/etc/redhat-release",
    "ansible_distribution_file_variety": "RedHat",
```

```
    "ansible_distribution_major_version": "7",
    "ansible_distribution_release": "Core",
    "ansible_distribution_version": "7.9",
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false
}
...
```

## Autres exemples pour la commande de configuration “setup”

# Display facts from all hosts and store them indexed by hostname at /tmp/facts.

```
ansible all -m setup --tree /tmp/facts
```

# Display only facts regarding memory found by ansible on all hosts and output them.

```
ansible all -m setup -a 'filter=ansible_*_mb'
```

# Display only facts returned by facter.

```
ansible all -m setup -a 'filter=facter_*'
```

# Display only facts about certain interfaces.

```
ansible all -m setup -a 'filter=ansible_eth[0-2]'
```

# Restrict additional gathered facts to network and virtual.

```
ansible all -m setup -a 'gather_subset=network,virtual'
```

# Do not call puppet facter or ohai even if present.

```
ansible all -m setup -a 'gather_subset=!facter,!ohai'
```

# Only collect the minimum amount of facts:

```
ansible all -m setup -a 'gather_subset=!all'
```

```
# Display facts from Windows hosts with custom facts stored in C:\\custom_facts.
```

```
ansible windows -m setup -a "fact_path='c:\\custom_facts'"
```

## Besoin de privilèges

La commande ci-dessous vous permet d'exécuter des commandes ad-hoc en tant qu'utilisateur non root avec des privilèges root.

```
ansible dbbservers -m shell -a 'fdisk -l'
```

L'option **--become** / **-b** donne les privilèges root et l'option **--ask-become-pass** / **-K** demande le mot de passe du sudo.

```
ansible dbbservers -m shell -a 'fdisk -l' --become --ask-become-pass
```

## Exemple d'Output:

```
db | CHANGED | rc=0 >>
Disk /dev/sdb: 10 MiB, 10485760 bytes, 20480 sectors
Units: sectors of 1 \* 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk /dev/sda: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 \* 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x291a2629
Device Boot Start End Sectors Size Id Type
/dev/sda1 \* 2048 20971486 20969439 10G 83 Linux
```

## Module command

Cette commande ad-hoc est utilisée pour redémarrer le système avec l'option **-f** pour définir le nombre de forks (exécutions concurrentes).

```
ansible all -m command -a "/sbin/reboot" -f 1
```

OU

```
ansible all -a "/sbin/reboot" -f 1
```

## Exemple d'Output:

```
db | CHANGED | rc=0 >>
Disk /dev/sdb: 10 MiB, 10485760 bytes, 20480 sectors
Units: sectors of 1 \* 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk /dev/sda: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 \* 512 = 5
```

## Module shell vs command

Dans la plupart des cas d'utilisation, les deux modules mènent au même objectif. Voici les principales différences entre ces modules.

- Avec le module Command, la commande sera exécutée sans passer par un shell. Par conséquent, certaines variables comme *\$HOME* ne sont pas disponibles. Et aussi des opérations de flux comme *<, >, /* et *&* ne fonctionnera pas.
- Le module Shell exécute une commande via un shell, par défaut */bin/sh*. Le tube et la redirection sont disponibles.
- Le module de commande est plus sécurisé, car il ne sera pas affecté par l'environnement de l'utilisateur.

## Transfert de fichiers

La commande ad-hoc ansible ci-dessous est utilisée pour copier un fichier d'une source vers une destination pour un groupe d'hôtes défini dans le fichier d'inventaire. Si la sortie avec le paramètre «changed» est «true», le fichier a été copié vers la destination.

```
touch file1.txt

ansible all -m copy -a 'src=file1.txt dest=/home/vagrant/Desktop/ owner=root
mode=0644' --become
```

Exécutez la commande ci-dessous pour vérifier si le module copy a fonctionné correctement ou non. Le fichier copié doit arriver à la destination mentionnée dans la commande précédente.

```
ls -l /home/vagrant/Desktop
```

## le module fetch

La commande ad-hoc ci-dessous est utilisée pour télécharger un fichier à partir d'une hôte définie dans la commande (l'inverse de copy).

```
ansible all -m fetch -a "src=/etc/passwd dest=/tmp/"
```

### Exemple d'Output:

```
app1 | CHANGED => {
  "changed": true,
  "checksum": "8098e40f089b4c49d96313ee65f100b43f8b526a",
  "dest": "/tmp/app1/etc/passwd",
  "md5sum": "5de7dcc4e4e3e74219a876ad2252a1c",
  "remote_checksum": "8098e40f089b4c49d96313ee65f100b43f8b526a",
  "remote_md5sum": null
}
```

Vérifiez si le fichier a été téléchargé ou non à la destination mentionnée dans la commande.

```
ls /tmp
```

essayez avec l'argument flat:

```
ansible all -m fetch -a "src=/etc/passwd dest=/tmp/ flat=yes"
```

```
ls /tmp
```