



# **Ansible pour professionnels**

## **Linux / Unix**

Le support du cours «Ansible pour professionnel Linux/Unix » est non contractuel ; il ne doit pas être redistribué et/ou reproduit en partie ou en totalité sans permission explicite et écrite de la société Adlere.

Red Hat, le logo Red Hat, OpenShift et Ansible sont des marques déposées ou commerciales de Red Hat, Inc ou ses filiales aux États-Unis et dans d'autre pays. Linux® est une marque déposée de Linus Torvalds aux États-Unis et dans d'autre pays.

UNIX® est une marque déposée par « The Open Group » aux Etats-Unis et dans d'autres pays. Wiindows® est une marque déposée de Microsoft Corporation aux États-Unis et dans d'autre pays.

Les autres marques citées sont déposées par leurs propriétaires respectifs.



**commandes ad-hoc**

- on peut invoquer directement des modules depuis la ligne de commande
- avec la commande `ansible`, pas `ansible-playbook`
- pour des tâches simples en une ligne, ne nécessitant pas un playbook
- Syntaxe générale :  
`ansible [-i inventory] <host-pattern> -m <module> [-a "param1=value1 'param2=value2 avec espaces'"]`

## Exemples

```
ansible -m ping <TARGET>
```

test complet d'une connexion + version de python

```
ansible -m setup <TARGET>
```

gather facts

```
ansible -m setup teacher -a "filter=ansible_distribution"
```

```
ansible -m command <TARGET> -a "whoami"
```

à remplacer par la commande de son choix

```
ansible <TARGET> -m command -a /usr/bin/hostname
```

```
ansible <TARGET> -m command -a /usr/bin/hostname -o
```

sur une seule ligne

```
ansible <TARGET> -a /usr/bin/hostname
```

`-m command` présumé par défaut





Catégorie	Module	Commentaires / exemples
modules de fichiers	copy	copie d'un fichier local vers la cible (ansible -m copy all -a "src=file dest=/tmp/file")
	fetch	récupère un fichier distant en local ansible target01 -m fetch -u automation -a "src=/etc/passwd dest=passwd flat=yes"
	file	positionne des permissions et propriétés sur des fichiers
	lineinfile	s'assure (ou pas) de la présence d'une ligne en particulier dans un fichier ansible all -m lineinfile -a "dest=/etc/group regexp='^(users:x:100:)(.*)' line='\1ldapusername,\2' state=present backrefs=yes"
	synchronize	synchronisation de contenu avec rsync
gestion logicielle	package	gestion des logiciels en fonction du gestionnaire natif détecté
	yum / apt / dnf / gem / pip	Gestionnaires spécifiques à un OS ou format donné.



Catégorie	Module	Exemples / commentaires
système	service (systemd disponible)	<code>ansible webservers -m service -a "name=httpd state=enabled state=started"</code>
	reboot	redémarrage du système distant
	user / group	ajout, effacement, modifications sur des utilisateurs / groupes <code>ansible all -m user -a "name=newbie uid=4000 state=present"</code>
	setup	informations système <code>ansible target01 -m setup -u automation -a 'filter=ansible_distribution*'</code>
	authorized_key	Dépôt d'une clef ssh pour un utilisateur donné <code>ansible TARGET -m authorized_key -a "user=root key='ssh-rsa AAAA ... XXX == <u>root@hostname</u>'"</code>
réseau	get_url	téléchargement de fichiers par HTTP, HTTPS, ou FTP
	nmcli	gestion du réseau
	uri	interactions avec les services web
	firewalld	gestion du firewall



Module	Caractéristiques
<code>command</code>	Python obligatoire sur la cible; comme c'est lui qui gère la commande, les variables d'environnement et les fonctions de stream (redirections, enchaînements) ne fonctionnent pas.
<code>shell</code>	Fait appel à l'environnement utilisateur (donc si problème avec celui-ci, le module peut ne pas fonctionner); enchaînements et redirections disponibles.  <code>/bin/sh</code> est le shell par défaut.
<code>raw</code>	Similaire au module shell, utilise le shell défini pour l'utilisateur; ne nécessite pas python du tout.



# Commandes ad-hoc

option `-t` pour faire appel au plugin `ansible.builtin.tree` callback

- option `-t <répertoire>` fait gérer la sortie d'une commande ad-hoc par le plugin de sortie (callback plugin) `tree`
- la sortie de chaque host est stockée dans un fichier à son nom, dans le répertoire donné en argument de `-t` :

```
$ ansible -m ping all -t ./output
proxmox | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
docker01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
[ ... ]
$ ls -R output
output:
aac01 aah01 ansible01 awx01 docker01 eda01 pods01 proxmox psql01 psql02 ref0 ref1 rsyslog01.intra.ks2i.net
]$ cat ./output/proxmox | jq .
{
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```





**Merci**

