

Modules Ansible

Objectifs de certification

RHCE EX294 (RHEL8)

Si vous poursuivez des objectifs de certification voici ceux qui sont suggérés ici :

- **2. Maîtrise des composants de base d'Ansible**
 - 2.2. Modules
 - 2.3. Variables
 - 2.4. Facts
 - 2.8. Utiliser la documentation fournie pour trouver des informations spécifiques aux modules et commandes Ansible
- **5. Écriture de scripts pour les tâches d'administration**
 - 5.2. Créer des scripts shell simples qui exécutent les commandes Ansible ad hoc
- **7. Utilisation des modules Ansible**

1. Introduction

Les modules Ansible sont des “bouts de codes” écrits principalement en Python (mais tout langage supportant les retours JSON est autorisé) pour modifier l'état d'une propriété d'un hôte. Les modules sont invoqués par **l'exécution de tâches** soit directement dans la ligne de commande `ansible` ou dans des livres de jeu avec la commande `ansible-playbook`.

2. Documentation des modules

Certains d'entre nous trouveront la documentation des modules plus agréable à lire en ligne :

https://docs.ansible.com/ansible/latest/modules/modules_by_category.html. Mais on obtient exactement le même résultat hors-ligne, sur la machine de contrôle avec la commande `ansible-doc` :

Pour lister les modules :

```
ansible-doc -l
```

Pour appeler directement la documentation d'un module.

```
ansible-doc <module>
```

```
ansible-doc -s <module>
```

Veillez vous informer sur les modules suivants :

- `ansible-doc setup`
- `ansible-doc ping`
- `ansible-doc lineinfile`
- `ansible-doc uri`

3. Exécutions de tâches Ad-Hoc

La commande `ansible` offre la possibilité d'exécuter des modules ansible de manière "ad-hoc", c'est à dire tâche par tâche sur un groupe d'hôtes, une sorte de livre de jeux à tâche unique. L'intérêt est de pouvoir exécuter la même tâche en parallèle sur un inventaire (constitué d'un certain nombre de cibles) en tout ou en partie.

```
ansible --help
```

La commande offre ce résultat sur la sortie :

```
Usage: ansible <host-pattern> [options]
```

```
Define and run a single task 'playbook' against a set of hosts
```

En général, la commande s'écrit `ansible` suivie d'un hôte ou d'un groupe d'inventaire suivie `-m` `nom_de_module` suivie l'option d'argument `-a "clé=valeur clé=valeur clé=valeur"`, par exemple de manière formelle :

```
ansible <group_inventaire> -m nom_de_module -a "clé=valeur clé=valeur clé=valeur"
```

Trois autres exemples fonctionnels :

```
ansible localhost -m debug -a "msg='Hello World!'"
```

```
ansible localhost -m file -a "path=/tmp/test state=touch"
```

```
ansible localhost -m file -a "path=/tmp/test state=absent"
```

⚠ Attention, l'inventaire doit exister et doit être nourri des hôtes à gérer !

On trouvera aussi des options utiles de la ligne de commande comme celles-ci :

- `-b`, `--become` active l'élévation de privilèges (`sudo`)
- `-e` qui permet de placer des variables ponctuelles ou un fichier de variables,
- `-o` qui offre la sortie standard en une seule ligne,
- ou encore `-l`, `--limit` qui permet de limiter la tâche à certains hôtes seulement de l'inventaire,
- ou encore `-v` ou `--vvv` pour la verbosité,
- `-i` précise le chemin du fichier d'inventaire.

Dans les exemples qui précèdent les tâches portent sur l'hôte spécial "localhost" qui n'a pas besoin d'être référencé dans un inventaire.

4. Valeurs de retour de l'exécution des tâches et des modules

Comme constaté, Ansible génère des valeurs de retour à la suite de l'exécution des tâches et des modules.

[Valeurs de retour, Return Values](#)

Valeurs de retour	Signification de la valeurs de retour	
<code>backup_file</code> (fichier_de_sauvegarde)	Pour les modules qui implémentent <code>backup=no</code>	yes lors de la manipulation de fichiers, un chemin vers le fichier de sauvegarde créé.
<code>changed</code> (modifié)	Un <i>booléen</i> indiquant si la tâche a dû effectuer des modifications.	
<code>failed</code> (échoué)	Un <i>booléen</i> qui indique si la tâche a échoué ou non.	

Valeurs de retour	Signication de la valeurs deretour	
<code>invocation</code>	Informations sur la manière dont le module a été invoqué.	
<code>msg</code>	Une chaîne avec un message générique relayé à l'utilisateur.	
<code>rc</code>	Certains modules exécutent des utilitaires en ligne de commande ou sont conçus pour exécuter des commandes directement (raw, shell, commande, etc), ce champ contient le 'code de retour' de ces utilitaires.	
<code>results</code>	Si cette clé existe, elle indique qu'une boucle était présente pour la tâche et qu'elle contient une liste du module normal'résultat' par élément.	
<code>skipped</code> (évit)	Un <i>booléen</i> qui indique si la tâche a été ignorée ou non.	
<code>stderr</code>	Certains modules exécutent des utilitaires en ligne de commande ou sont conçus pour exécuter des commandes directement (raw, shell, commande, etc), ce champ contient la sortie d'erreur de ces utilitaires.	
<code>stderr_lines</code>	Lorsque <code>stderr</code> est retourné, nous fournissons aussi toujours ce champ qui est une liste de chaînes de caractères, un élément par ligne de l'original.	
<code>stdout</code>	Certains modules exécutent des utilitaires en ligne de commande ou sont conçus pour exécuter directement des commandes (raw, shell, commande, etc). Cette zone contient l'édition normale de ces utilitaires.	
<code>stdout_lines</code>	Lorsque <code>stdout</code> est retourné, Ansible fournit toujours une liste de chaînes de caractères, chacune contenant un élément par ligne de la sortie originale.	

5. Modules Linux

Les modules Ansible sont donc des “morceaux de code” pour modifier l'état d'une propriété d'un hôte.

On recense ici 240 modules utiles pour l'administration d'un système Linux.

5.1. Paquets logiciels et repos

- [ansible.builtin.dnf](#) – Manages packages with the *dnf* package manager
- [ansible.builtin.package](#) – Generic OS package manager
- [ansible.builtin.package_facts](#) – package information as facts
- [ansible.builtin.pip](#) – Manages Python library dependencies
- [ansible.builtin.rpm_key](#) – Adds or removes a gpg key from the rpm db
- [ansible.builtin.ansible.builtin.yum](#) – Manages packages with the *yum* package manager
- [ansible.builtin.yum_repository](#) – Add or remove YUM repositories
- [community.general.alternatives](#) – Manages alternative programs for common commands
- [community.general.apt_rpm](#) – apt_rpm package manager
- [community.general.urpmi](#) – Urpmi manager

5.2. Services

- [ansible.builtin.service](#) – Manage services
- [ansible.builtin.service_facts](#) – Return service state information as fact data
- [ansible.builtin.systemd](#) – Manage services
- [ansible.builtin.sysvinit](#) – Manage SysV services.

5.3. Règles de pare-feu

- [ansible.builtin.iptables](#) – Modify iptables rules
- [ansible.posix.firewalld](#) – Manage arbitrary ports/services with firewalld
- [community.general.iptables_state](#) – Save iptables state into a file or restore it from a file

5.4. Systèmes de fichiers

- [ansible.builtin.fetch](#) – Fetch files from remote nodes
- [ansible.builtin.file](#) – Manage files and file properties
- [ansible.builtin.find](#) – Return a list of files based on specific criteria
- [ansible.builtin.stat](#) – Retrieve file or file system status
- [ansible.posix.acl](#) – Set and retrieve file ACL information.
- [community.general.xattr](#) – Manage user defined extended attributes
- [ansible.builtin.get_url](#) – Downloads files from HTTP, HTTPS, or FTP to node
- [ansible.builtin.git](#) from git checkouts
- [ansible.builtin.subversion](#) – Deploys a subversion repository
- [ansible.builtin.slurp](#) – Slurps a file from remote nodes

- [ansible.builtin.uri](#) – Interacts with webservices
- [ansible.posix.synchronize](#) – A wrapper around rsync to make common tasks in your playbooks quick and easy

5.5. Périphériques de stockage

- [ansible.posix.mount](#) – Control active and configured mount points
- [community.general.crypttab](#) – Encrypted Linux block devices
- [community.crypto.luks_device](#) – Manage encrypted (LUKS) devices
- [community.general.filesystem](#) – Makes a filesystem
- [community.general.iso_create](#) – Generate ISO file with specified files or folders
- [community.general.iso_extract](#) – Extract files from an ISO image
- [community.general.lvg](#) – Configure LVM volume groups
- [community.general.lvol](#) – Configure LVM logical volumes
- [community.general.open_iscsi](#) – Manage iSCSI targets with Open-iSCSI
- [community.general.parted](#) – Configure block device partitions

5.6. Contenus de fichiers

- [ansible.builtin.assemble](#) – Assemble configuration files from fragments
- [ansible.builtin.blockinfile](#) – Insert/update/remove a text block surrounded by marker lines
- [ansible.builtin.lineinfile](#) – Manage lines in text files
- [ansible.builtin.replace](#) – Replace all instances of a particular string in a file using a back-referenced regular expression
- [ansible.builtin.copy](#) – Copy files to remote locations
- [ansible.builtin.tempfile](#) – Creates temporary files and directories
- [ansible.builtin.template](#) – Template a file out to a remote server
- [ansible.posix.patch](#) – Apply patch files using the GNU patch tool

5.7. Archives

- [community.general.archive](#) – Creates a compressed archive of one or more files or trees
- [ansible.builtin.unarchive](#) copying it from the local machine.

5.8. Tâches planifiées

- [ansible.builtin.cron](#) – Manage cron.d and crontab entries
- [ansible.posix.at](#) – Schedule the execution of a command or script file via the at command
- [community.general.cronvar](#) – Manage variables in crontabs

5.9. SSH

- [ansible.builtin.known_hosts](#) – Add or remove a host from the `known_hosts` file
- [ansible.posix.authorized_key](#) – Adds or removes an SSH authorized key
- [community.crypto.openssh_cert](#) – Generate OpenSSH host or user certificates.
- [community.crypto.openssh_keypair](#) – Generate OpenSSH private and public keys.

5.10. SELINUX

- [ansible.posix.seboolean](#) – Toggles SELinux booleans
- [ansible.posix.selinux](#) – Change policy and state of SELinux
- [community.general.sefcontext](#) – Manages SELinux file context mapping definitions
- [community.general.selinux_permissive](#) – Change permissive domain in SELinux policy
- [community.general.selogin](#) – Manages linux user to SELinux user mapping
- [community.general.seport](#) – Manages SELinux network port type definitions

5.11. Utilisateurs et groupes

- [ansible.builtin.user](#) – Manage user accounts
- [ansible.builtin.group](#) – Add or remove groups
- [community.general.pam_limits](#) – Modify Linux PAM limits
- [community.general.pamd](#) – Manage PAM Modules

5.12. Administration divers

- [ansible.builtin.hostname](#) – Manage hostname
- [ansible.builtin.ping](#) – Try to connect to host, verify a usable python and return `pong` on success
- [ansible.builtin.reboot](#) – Reboot a machine
- [community.general.shutdown](#) – Shut down a machine
- [ansible.posix.sysctl](#) – Manage entries in `sysctl.conf`.
- [community.general.locale_gen](#) – Creates or removes locales

5.13. Commandes

- [ansible.builtin.script](#) – Runs a local script on a remote node after transferring it
- [ansible.builtin.command](#) – Execute commands on targets
- [ansible.builtin.expect](#) – Executes a command and responds to prompts.
- [ansible.builtin.raw](#) – Executes a low-down and dirty command

- [ansible.builtin.shell](#) - Execute shell commands on targets
- [ansible.builtin.pause](#) - Pause playbook execution
- [ansible.builtin.wait_for](#) - Waits for a condition before continuing
- [ansible.builtin.wait_for_connection](#) - Waits until remote system is reachable/usable

5.14. Réseau

- [community.general.nmcli](#) - Manage Networking
- [community.general.listen_ports_facts](#) - Gather facts on processes listening on TCP and UDP ports.
- [community.general.nsupdate](#) - Manage DNS records.
- [community.general.ipify_facts](#) - Retrieve the public IP of your internet gateway
- [community.general.ipinfo_facts](#) - Retrieve IP geolocation facts of a host's IP address

5.15. Informations

- [ansible.builtin.debug](#) - Print statements during execution
- [ansible.builtin.gather_facts](#) - Gathers facts about remote hosts
- [ansible.builtin.set_fact](#) - Set host facts from a task
- [ansible.builtin.setup](#) - Gathers facts about remote hosts
- [ansible.builtin.fail](#) - Fail with custom message

5.16. Modules internes

- [ansible.builtin.add_host](#) to the ansible-playbook in-memory inventory
- [ansible.builtin.assert](#) - Asserts given expressions are true
- [ansible.builtin.async_status](#) - Obtain status of asynchronous task
- [ansible.builtin.getent](#) - A wrapper to the unix getent utility
- [ansible.builtin.group_by](#) - Create Ansible groups based on facts
- [ansible.builtin.meta](#) - Execute Ansible 'actions'
- [ansible.builtin.set_stats](#) - Set stats for the current ansible run

5.17. Include/Import

- [ansible.builtin.import_playbook](#) - Import a playbook
- [ansible.builtin.import_role](#) - Import a role into a play
- [ansible.builtin.import_tasks](#) - Import a task list
- [ansible.builtin.include](#) - Include a play or task list
- [ansible.builtin.include_role](#) - Load and execute a role
- [ansible.builtin.include_tasks](#) - Dynamically include a task list

- [ansible.builtin.include_vars](#) – Load variables from files, dynamically within a task

5.18. Autres Packagers et gestionnaires de dépendances

- [community.general.bundler](#) – Manage Ruby Gem dependencies with Bundler
- [community.general.composer](#) – Dependency Manager for PHP
- [community.general.cpanm](#) – Manages Perl library dependencies.
- [community.general.django_manage](#) – Manages a Django application.
- [community.general.pear](#) – Manage pear/pecl packages
- [community.general.easy_install](#) – Installs Python libraries
- [community.general.gem](#) – Manage Ruby gems
- [community.general.snap](#) – Manages snaps
- [community.general.make](#) – Run targets in a Makefile
- [community.general.npm](#) – Manage node.js packages with npm
- [community.general.pip_package_info](#) – pip package information
- [community.general.python_requirements_facts](#) – Show python path and assert dependency versions
- [community.general.python_requirements_info](#) – Show python path and assert dependency versions
- [community.general.yarn](#) – Manage node.js packages with Yarn

5.19. MySQL

- [community.mysql.mysql_db](#) – Add or remove MySQL databases from a remote host
- [community.mysql.mysql_info](#) – Gather information about MySQL servers
- [community.mysql.mysql_query](#) – Run MySQL queries
- [community.mysql.mysql_replication](#) – Manage MySQL replication
- [community.mysql.mysql_user](#) – Adds or removes a user from a MySQL database
- [community.mysql.mysql_variables](#) – Manage MySQL global variables

5.20. Docker Containers

- [community.general.docker_compose](#) – Manage multi-container Docker applications with Docker Compose.
- [community.general.docker_config](#) – Manage docker configs.
- [community.general.docker_container](#) – manage docker containers
- [community.general.docker_container_info](#) – Retrieves facts about docker container
- [community.general.docker_host_info](#) – Retrieves facts about docker host and lists of objects of the services.

- [community.general.docker_image](#) – Manage docker images.
- [community.general.docker_image_facts](#) – Inspect docker images
- [community.general.docker_image_info](#) – Inspect docker images
- [community.general.docker_login](#) – Log into a Docker registry.
- [community.general.docker_network](#) – Manage Docker networks
- [community.general.docker_network_info](#) – Retrieves facts about docker network
- [community.general.docker_node](#) – Manage Docker Swarm node
- [community.general.docker_node_info](#) – Retrieves facts about docker swarm node from Swarm Manager
- [community.general.docker_prune](#) – Allows to prune various docker objects
- [community.general.docker_secret](#) – Manage docker secrets.
- [community.general.docker_service](#) – Manage multi-container Docker applications with Docker Compose.
- [community.general.docker_stack](#) – docker stack module
- [community.general.docker_stack_info](#) – Return information on a docker stack
- [community.general.docker_stack_task_info](#) – Return information of the tasks on a docker stack
- [community.general.docker_swarm](#) – Manage Swarm cluster
- [community.general.docker_swarm_info](#) – Retrieves facts about Docker Swarm cluster.
- [community.general.docker_swarm_service](#) – docker swarm service
- [community.general.docker_swarm_service_info](#) – Retrieves information about docker services from a Swarm Manager
- [community.general.docker_volume](#) – Manage Docker volumes
- [community.general.docker_volume_info](#) – Retrieve facts about Docker volumes

5.20. LXC/LXD

- [community.general.lxc_container](#) – Manage LXC Containers
- [community.general.lxca_cmms](#) – Custom module for lxca cmms inventory utility
- [community.general.lxca_nodes](#) – Custom module for lxca nodes inventory utility
- [community.general.lxd_container](#) – Manage LXD Containers
- [community.general.lxd_profile](#) – Manage LXD profiles

5.21. LDAP

- [community.general.ldap_attr](#) – Add or remove LDAP attribute values
- [community.general.ldap_attrs](#) – Add or remove multiple LDAP attribute values
- [community.general.ldap_entry](#) – Add or remove LDAP entries.
- [community.general.ldap_passwd](#) – Set passwords in LDAP.
- [community.general.ldap_search](#) – Search for entries in a LDAP server

5.22. Web Servers

- [community.general.htpasswd](#) – manage user files for basic authentication
- [community.general.apache2_mod_proxy](#) – Set and/or get members' attributes of an Apache httpd 2.4 mod_proxy balancer pool
- [community.general.apache2_module](#) – Enables/disables a module of the Apache2 webserver.
- [community.general.nginx_status_facts](#) – Retrieve nginx status facts.
- [community.general.nginx_status_info](#) – Retrieve information on nginx status.

5.23. PostgreSQL

- [community.general.postgresql_copy](#) – Copy data between a file/program and a table
- [community.general.postgresql_db](#) – Add or remove PostgreSQL databases from a remote host.
- [community.general.postgresql_ext](#) – Add or remove PostgreSQL extensions from a database
- [community.general.postgresql_idx](#) – Create or drop indexes from a PostgreSQL database
- [community.general.postgresql_info](#) – Gather information about PostgreSQL servers
- [community.general.postgresql_lang](#) – Adds, removes or changes procedural languages with a PostgreSQL database
- [community.general.postgresql_membership](#) – Add or remove PostgreSQL roles from groups
- [community.general.postgresql_owner](#) – Change an owner of PostgreSQL database object
- [community.general.postgresql_pg_hba](#) – Add, remove or modify a rule in a pg_hba file
- [community.general.postgresql_ping](#) – Check remote PostgreSQL server availability
- [community.general.postgresql_privs](#) – Grant or revoke privileges on PostgreSQL database objects
- [community.general.postgresql_publication](#) – Add, update, or remove PostgreSQL publication
- [community.general.postgresql_query](#) – Run PostgreSQL queries
- [community.general.postgresql_schema](#) – Add or remove PostgreSQL schema
- [community.general.postgresql_sequence](#) – Create, drop, or alter a PostgreSQL sequence
- [community.general.postgresql_set](#) – Change a PostgreSQL server configuration parameter
- [community.general.postgresql_slot](#) – Add or remove replication slots from a PostgreSQL database
- [community.general.postgresql_subscription](#) – Add, update, or remove PostgreSQL subscription
- [community.general.postgresql_table](#) – Create, drop, or modify a PostgreSQL table
- [community.general.postgresql_tablespace](#) – Add or remove PostgreSQL tablespaces from remote hosts
- [community.general.postgresql_user](#) – Add or remove a user from a PostgreSQL server instance
- [community.general.postgresql_user_obj_stat_info](#) – Gather statistics about PostgreSQL user objects

5.24. RHSM Red Hat software channels

- [community.general.redhat_subscription](#) – Manage registration and subscriptions to RHSM using the `subscription-manager` command
- [community.general.rhn_channel](#) – Adds or removes Red Hat software channels

- [community.general.rhn_register](#) – Manage Red Hat Network registration using the `rhnsreg` command
- [community.general.rhsm_release](#) – Set or Unset RHSM Release version
- [community.general.rhsm_repository](#) – Manage RHSM repositories using the subscription-manager command

5.25. Fichiers ini, csv, xml

- [community.general.read_csv](#) – Read a CSV file
- [community.general.git_config](#) – Read and write git configuration
- [community.general.ini_file](#) – Tweak settings in INI files
- [community.general.xml](#) – Manage bits and pieces of XML files or strings

5.26. Gestion des processus et du noyau

- [community.general.capabilities](#) – Manage Linux capabilities
- [community.general.modprobe](#) – Load or unload kernel modules
- [community.general.pids](#) – Retrieves process IDs list if the process is running otherwise return empty list

5.27. Surveillance

- [community.general.snmp_facts](#) – Retrieve facts for a device using SNMP
- [community.general.syslogger](#) – Log messages in the syslog
- [community.general.mail](#) – Send an email

5.28. XFS / ZFS

- [community.general.xfs_quota](#) – Manage quotas on XFS filesystems
- [community.general.zfs](#) – Manage zfs
- [community.general.zfs_delegate_admin](#)
- [community.general.zfs_facts](#) – Gather facts about ZFS datasets.
- [community.general.zpool_facts](#) – Gather facts about ZFS pools.

5.29. Debian/Ubuntu

- [ansible.builtin.apt](#) – Manages apt-packages
- [ansible.builtin.apt_key](#) – Add or remove an apt key
- [ansible.builtin.apt_repository](#) – Add and remove APT repositories

- [ansible.builtin.dpkg_selections](#) - Dpkg package selection selections
- [ansible.builtin.debconf](#) - Configure a .deb package
- [community.general.apt_repo](#) - Manage APT repositories via apt-repo
- [community.general.apt_rpm](#) - apt_rpm package manager
- [community.general.dpkg_divert](#) - Override a debian package's version of a file
- [community.general.interfaces_file](#) - Tweak settings in /etc/network/interfaces files
- [community.general.ufw](#) - Manage firewall with UFW

5.30. Crypto TLS

- [community.crypto.acme_account](#) - Create, modify or delete ACME accounts
- [community.crypto.acme_account_facts](#) - Retrieves information on ACME accounts
- [community.crypto.acme_account_info](#) - Retrieves information on ACME accounts
- [community.crypto.acme_certificate](#) - Create SSL/TLS certificates with the ACME protocol
- [community.crypto.acme_certificate_revoke](#) - Revoke certificates with the ACME protocol
- [community.crypto.acme_challenge_cert_helper](#) - Prepare certificates required for ACME challenges such as [tls-alpn-01](#)
- [community.crypto.acme_inspect](#) - Send direct requests to an ACME server
- [community.crypto.certificate_complete_chain](#) - Complete certificate chain given a set of untrusted and root certificates
- [community.crypto.get_certificate](#) - Get a certificate from a host:port
- [community.crypto.openssl_certificate](#) - Generate and/or check OpenSSL certificates
- [community.crypto.openssl_certificate_info](#) - Provide information of OpenSSL X.509 certificates
- [community.crypto.openssl_csr](#) - Generate OpenSSL Certificate Signing Request (CSR)
- [community.crypto.openssl_csr_info](#) - Provide information of OpenSSL Certificate Signing Requests (CSR)
- [community.crypto.openssl_dhparam](#) - Generate OpenSSL Diffie-Hellman Parameters
- [community.crypto.openssl_pkcs12](#) - Generate OpenSSL PKCS#12 archive
- [community.crypto.openssl_privatekey](#) - Generate OpenSSL private keys
- [community.crypto.openssl_privatekey_info](#) - Provide information for OpenSSL private keys
- [community.crypto.openssl_publickey](#) - Generate an OpenSSL public key from its private key.
- [community.crypto.openssl_signature](#) - Sign data with openssl
- [community.crypto.openssl_signature_info](#) - Verify signatures with openssl
- [community.crypto.x509_certificate](#) - Generate and/or check OpenSSL certificates
- [community.crypto.x509_certificate_info](#) - Provide information of OpenSSL X.509 certificates
- [community.crypto.x509_crl](#) - Generate Certificate Revocation Lists (CRLs)
- [community.crypto.x509_crl_info](#) - Retrieve information on Certificate Revocation Lists (CRLs)

6. Modules Réseau

Les modules Ansible pour l'automatisation du réseau sont désignés par le constructeur de l'OS suivi d'un nom de module :

- *_facts : récupère des facts
- *_command : exécute des commandes en mode privilège
- *_config : exécute des commandes en mode de configuration

Selon le nom des plateformes :

- Arista EOS = eos_*
- Cisco IOS/IOS-XE = ios_*
- Cisco NX-OS = nxos_*
- Cisco IOS-XR = iosxr_*
- Juniper Junos = junos_*
- VyOS = vyos_*

Mais on trouve aussi un grand nombre des modules spécialisés.

7. Modules Windows

...

8. Modules VMWare

...

9. Développement de modules

[Développement de modules](#)

Exemple de développement d'un module simple.

Revision #1

Created 3 October 2021 21:08:08 by garfi

Updated 3 October 2021 21:08:42 by garfi