

TIC-NUX4

Documentation des modules :

- **import paramiko** : importe la bibliothèque Paramiko qui permet d'implémenter les protocoles ssh.
- **import yaml** : importe la bibliothèque Yaml qui permet de lire et d'écrire des fichiers Yaml
- **import click** : importe la bibliothèque Click qui permet de créer des interfaces en ligne de commande.
- **import os** : importe la bibliothèque Os qui permet d'interagir avec le système d'exploitation sur lequel le python va être exécuté.
- **import logging** : permet d'enregistrer les log des différentes actions effectuées.
- **from jinja2 import Environment, FileSystemLoader** : importe les objets Environnement et FileSystemLoader de la bibliothèque Jinja2.
- **from getpass import getpass** : permet de récupérer la getpass de la bibliothèque getpass afin de demander à l'utilisateur de saisir un mot de passe de manière sécurisée.

Documentation des classes :

- **class CmdResult:** Stock les résultats d'exécution d'une commande.
 - stdout: Affiche le résultat de la commande.
 - stderr : Affiche un message en cas d'erreur de la commande.
 - exit_code : Stocke le code d'erreur de la commande.
- **def command_module(command, shell, ssh_client) :**
 - Prends en argument une commande à exécuter, ssh_client un objet de la classe SSHClient, si aucun shell n'est spécifié, /bin/bash sera utilisé.
 - Cette fonction exécute la commande demandée sur un client distant en ssh. Les résultats d'exécutions sont enregistrés dans les journaux de logs.
- **def sysctl_module(attribute, value, permanent, ssh_client) :**
 - Prends en argument ssh_client un objet de la classe SSHClient, l'argument et la valeur de la commande à effectuer et un attribut "permanent" booléen.
 - Cette fonction modifie un attribut de configuration du système du client hôte en utilisant la commande sysctl via Ssh. Les résultats d'exécutions sont enregistrés dans les journaux de logs.
- **def copy_module(src, dest, backup, ssh_client) :**
 - Prends en arguments des chaînes de caractère src et dest qui représente les fichiers à copier le dossier de destination, ssh_client un objet de la classe SSHClient et un booléen backup.
 - Cette fonction copie un dossier vers un client ssh distant à l'aide d'une connexion sftp. Avant l'envoi, elle vérifie que la ressource existe localement. Elle vérifie ensuite après l'envoi que la ressource a bien été envoyée. Les résultats d'exécutions sont enregistrés dans les journaux de logs.
- **def render(playbook):**
 - Prends en argument un objet playbook qui contient les informations relatives au modèle à générer.
 - Cette fonction génère le rendu d'un modèle à partir de paramètres spécifiés dans un playbook. Le rendu est ensuite écrit dans un fichier spécifié.
- **def apt_package_management(package_name, desired_state, ssh_client):**
 - Prends en argument trois chaînes de caractères package_name, desired_state et ssh_client un objet de la classe SSHClient.
 - Cette fonction gère l'installation et la désinstallation de package (package_name) selon la valeur de "desired-state" en utilisant des commandes "apt-get" sur un client ssh distant.

- **def service_management(service_name, desired_state, ssh_client):**
 - Prends en argument trois chaînes de service_name, qui représente le service à gérer, desired_state, qui représente l'état souhaité pour le service et ssh_client un objet de la classe SSHClient.
 - Cette fonction permet de gérer les différents services du système client ssh avec des commandes "systemctl".

- **def run_remote_cmd(command: str, ssh_client: SSHClient) -> CmdResult:**
 - Prends en argument une chaîne de caractère "command" et ssh_client un objet de la classe SSHClient
 - Cette fonction exécute l'instruction donnée en commande sur le client ssh, le résultat de cette exécution est stocké grâce à la fonction CmdResult et est renvoyé.

- **def execute_playbook(playbook_file, inventory_file):**
 - Prends en argument deux fichiers Yaml, le premier contient les instructions a effectué, le second contient les clients ssh sur lesquels les instructions vont être exécutées.
 - La fonction charge et lis les deux fichiers d'instructions Yaml. Elle se connecte ensuite aux hôtes spécifiés.
Selon les exécutions demandées par le playbook, la fonction appellera les différentes fonctions explicitées précédemment et les exécutera.