D'accord, voici une explication détaillée de chaque script :

**1. ip\_management/data.py**

**Contenu :**

python

Copy code

# Exemple de contenu du fichier data.py

network\_data = {

'available\_ips': ['192.168.1.1', '192.168.1.2', '192.168.1.3'],

'reserved\_ips': [],

'devices': {

'router1': '192.168.1.1',

'switch1': '192.168.1.2',

'host1': '192.168.1.3'

}

}

**Explication :**

* network\_data: Un dictionnaire contenant les données de configuration du réseau.
  + available\_ips: Une liste des adresses IP disponibles pour la réservation.
  + reserved\_ips: Une liste des adresses IP déjà réservées.
  + devices: Un dictionnaire des appareils réseau et leurs adresses IP associées.

**2. ip\_management/ip\_functions.py**

**Contenu :**

python

Copy code

# Exemple de contenu du fichier ip\_functions.py

from .data import network\_data

def reserve\_ip(ip\_address):

if ip\_address in network\_data['available\_ips']:

network\_data['available\_ips'].remove(ip\_address)

network\_data['reserved\_ips'].append(ip\_address)

return True

return False

def release\_ip(ip\_address):

if ip\_address in network\_data['reserved\_ips']:

network\_data['reserved\_ips'].remove(ip\_address)

network\_data['available\_ips'].append(ip\_address)

return True

return False

def validate\_ip(ip\_address):

return ip\_address in network\_data['available\_ips'] or ip\_address in network\_data['reserved\_ips']

**Explication :**

* reserve\_ip(ip\_address): Fonction pour réserver une adresse IP.
  + Vérifie si l'adresse IP est disponible.
  + Déplace l'adresse IP de available\_ips à reserved\_ips si elle est disponible.
* release\_ip(ip\_address): Fonction pour libérer une adresse IP.
  + Vérifie si l'adresse IP est réservée.
  + Déplace l'adresse IP de reserved\_ips à available\_ips si elle est réservée.
* validate\_ip(ip\_address): Fonction pour valider une adresse IP.
  + Vérifie si l'adresse IP est dans available\_ips ou reserved\_ips.

**3. packet\_tracer/pt\_functions.py**

**Contenu :**

python

Copy code

# Exemple de contenu du fichier pt\_functions.py

def configure\_packet\_tracer\_device(device\_name):

# Cette fonction configure un appareil dans Packet Tracer

# Remplacez par le code réel de configuration Packet Tracer

print(f"Configuration de l'appareil {device\_name} dans Packet Tracer.")

return True

**Explication :**

* configure\_packet\_tracer\_device(device\_name): Fonction pour configurer un appareil dans Cisco Packet Tracer.
  + Simule la configuration d'un appareil dans Packet Tracer.
  + Dans un véritable environnement, cette fonction interagirait avec l'API de Packet Tracer pour configurer l'appareil.

**4. ansible/ansible\_functions.py**

**Contenu :**

python

Copy code

# Exemple de contenu du fichier ansible\_functions.py

import subprocess

def run\_ansible\_playbook(playbook\_path):

try:

result = subprocess.run(['ansible-playbook', playbook\_path], check=True, capture\_output=True, text=True)

print(result.stdout)

return True

except subprocess.CalledProcessError as e:

print(f"Erreur lors de l'exécution du playbook Ansible: {e}")

print(e.output)

return False

**Explication :**

* run\_ansible\_playbook(playbook\_path): Fonction pour exécuter un playbook Ansible.
  + Utilise le module subprocess pour appeler la commande ansible-playbook.
  + Capture et affiche la sortie de la commande.
  + Retourne True si le playbook s'exécute correctement, sinon retourne False.

**5. main.py**

**Contenu :**

python

Copy code

# Exemple de contenu du fichier main.py

import sys

from ip\_management.ip\_functions import reserve\_ip, release\_ip, validate\_ip

from packet\_tracer.pt\_functions import configure\_packet\_tracer\_device

from ansible.ansible\_functions import run\_ansible\_playbook

def main():

print("Bienvenue dans le script d'automatisation réseau")

while True:

print("\nOptions:")

print("1. Réserver une adresse IP")

print("2. Libérer une adresse IP")

print("3. Valider une adresse IP")

print("4. Configurer un appareil dans Packet Tracer")

print("5. Exécuter un playbook Ansible")

print("6. Quitter")

choice = input("Choisissez une option: ")

if choice == '1':

ip\_address = input("Entrez l'adresse IP à réserver: ")

result = reserve\_ip(ip\_address)

if result:

print(f"Adresse IP {ip\_address} réservée avec succès.")

else:

print(f"Échec de la réservation de l'adresse IP {ip\_address}.")

elif choice == '2':

ip\_address = input("Entrez l'adresse IP à libérer: ")

result = release\_ip(ip\_address)

if result:

print(f"Adresse IP {ip\_address} libérée avec succès.")

else:

print(f"Échec de la libération de l'adresse IP {ip\_address}.")

elif choice == '3':

ip\_address = input("Entrez l'adresse IP à valider: ")

result = validate\_ip(ip\_address)

if result:

print(f"L'adresse IP {ip\_address} est valide.")

else:

print(f"L'adresse IP {ip\_address} est invalide.")

elif choice == '4':

device\_name = input("Entrez le nom de l'appareil à configurer: ")

config\_result = configure\_packet\_tracer\_device(device\_name)

if config\_result:

print(f"Appareil {device\_name} configuré avec succès dans Packet Tracer.")

else:

print(f"Échec de la configuration de l'appareil {device\_name} dans Packet Tracer.")

elif choice == '5':

playbook\_path = input("Entrez le chemin du playbook Ansible: ")

ansible\_result = run\_ansible\_playbook(playbook\_path)

if ansible\_result:

print(f"Playbook Ansible {playbook\_path} exécuté avec succès.")

else:

print(f"Échec de l'exécution du playbook Ansible {playbook\_path}.")

elif choice == '6':

print("Quitter le script.")

sys.exit()

else:

print("Choix invalide. Veuillez réessayer.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Explication :**

* Le script principal fournit une interface utilisateur pour interagir avec les différentes fonctions des modules ip\_management, packet\_tracer, et ansible.
* Utilise des entrées de l'utilisateur pour appeler les fonctions appropriées et affiche les résultats des opérations effectuées.

Ce script coordonne les différents modules pour réaliser les tâches d'automatisation réseau de manière intégrée et conviviale.