

## EXOLAB : Découverte d'un contrôleur de réseau SDN sous Packet Tracer



Propriétés	Description
<b>Intitulé long</b>	<i>Découverte du contrôleur de réseau SDN sous Packet Tracer</i>
<b>Formation concernée</b>	BTS SIO SLAM et SISR
<b>Matière</b>	Bloc 2
<b>Présentation</b>	<p>Dans un premier temps, les étudiants sont amenés à étudier les différences entre la gestion d'un réseau à partir de l'interface en ligne de commande (CLI) et l'utilisation d'un contrôleur de réseau défini par logiciel (SDN).</p> <p>Dans un deuxième temps, ils utiliseront le contrôleur réseau de Packet Tracer et la documentation de l'API associée pour envoyer des requêtes REST à partir de Postman et de Visual Studio Code (VS Code).</p>
<b>Notions</b>	<p>La gestion d'un réseau se fait habituellement à partir de l'interface de lignes de commandes (CLI). Progressivement une nouvelle méthode prend de l'ampleur, il s'agit de l'utilisation d'un contrôleur de réseau défini par logiciel (SDN).</p> <p>Le paramétrage du SDN peut se faire grâce aux API par interface graphique, POSTMAN ou Python.</p>
<b>Transversalité</b>	SLAM et SISR
<b>Pré-requis</b>	<p>Connaissance des commandes de base pour le paramétrage des matériels de connexion.</p> <p>Notions de base en programmation (Python).</p>
<b>Outils</b>	<p>Cisco Packet Tracer Version 8 ou plus</p> <p>Microsoft Visual Studio ou VS Code</p> <p>POSTMAN</p>
<b>Mots-clés</b>	<p>Software Defined Networking ou SDN, contrôleur de réseau défini par logiciel (SDN).</p> <p>CLI Commande Line Interface, interface en ligne de commande</p> <p>Topologie réseau</p> <p>Contrôleur réseau</p> <p>Jeton d'authentification, Ticket API</p> <p>Scripts</p> <p>API REST</p> <p>POSTMAN</p> <p>REST avec Python</p> <p>HTTP : GET-POST-PUT-DELETE</p>
<b>Durée</b>	Entre 4h et 6h
<b>Auteur(es)</b>	<p>Zakari BERREMILI</p> <p>Relecture : Cécile Nivaggioni, Gilles Gouraud, Valéry Tschaen</p>
<b>Version</b>	v1.0
<b>Date de publication</b>	Février 2022
<b>Contenu du package</b>	<p>LAB_SDN_v1.0.docx</p> <p>LAB_SDN_v1.0.pdf</p> <p>SDN_Version_LAB_1.pkt</p> <p>SDN_Version_LAB_2.pkt</p> <p>01_get-ticket.py</p> <p>02_get-network-device.py</p> <p>03_get-host.py</p>

# Découverte du contrôleur de réseau SDN sous Packet Tracer

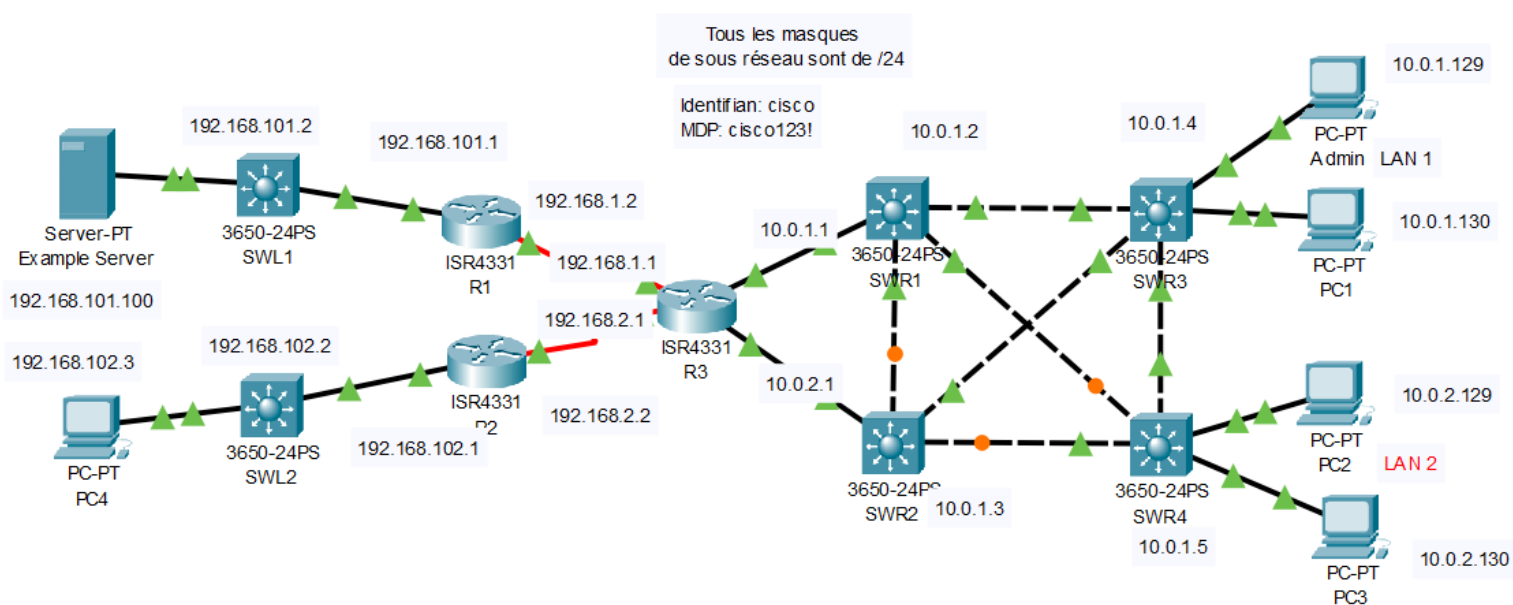
Un scénario largement inspiré de deux laboratoires DEVASC Netacad. Adapté par Zakari BERREMILI.

## Présentation rapide du contexte

Dans un premier temps, vous comparerez les différences entre la gestion d'un réseau à partir de l'interface en ligne de commande (CLI) et l'utilisation d'un contrôleur de réseau défini par logiciel (SDN).

Dans un deuxième temps, vous utiliserez le contrôleur réseau de Packet Tracer et la documentation de l'API associée pour envoyer des requêtes REST à partir de Postman et Microsoft Visual Studio ou de Visual Studio Code (VS Code). Packet Tracer prend également en charge un environnement de codage en langage Python. Par conséquent, dans la partie finale de cette activité, vous enverrez des requêtes REST soit à partir de Packet Tracer soit directement depuis Visual Studio installé sur votre station.

## Schéma de la maquette



## Table des matières

Première Partie : comparez la gestion d'un réseau à partir de l'interface de ligne de commande (CLI) et l'utilisation d'un contrôleur de réseau défini par logiciel (SDN).....	3
Deuxième Partie : Implémenter les API REST avec un contrôleur SDN.....	21
Troisième Partie : Requêtes REST en python .....	28
ANNEXE A : Apport théorique pour le LAB SDN .....	34

# Première Partie : comparez la gestion d'un réseau à partir de l'interface en ligne de commande (CLI) et l'utilisation d'un contrôleur de réseau défini par logiciel (SDN).

## Objectifs

Etape 1 : Explorer la topologie du réseau

Etape 2 : Utiliser l'interface en ligne de commande pour recueillir des informations

Etape 3 : Configurer un contrôleur SDN

Etape 4 : Utiliser un contrôleur SDN pour découvrir une topologie

Etape 5 : Utiliser un contrôleur SDN pour recueillir des informations

Etape 6 : Utiliser un contrôleur SDN pour configurer les paramètres réseau

## Étape 1 : Explorer la topologie du réseau

Dans cette étape, vous utiliserez la maquette LAB 1 (SDN Version LAB 1.pkt) et vous vous familiariserez avec la topologie que vous utiliserez pour les activités s'appuyant sur les interfaces de programmation réseau (API) et le SDN. Il vous faudra au minimum la version 8 de Cisco Packet Tracer.

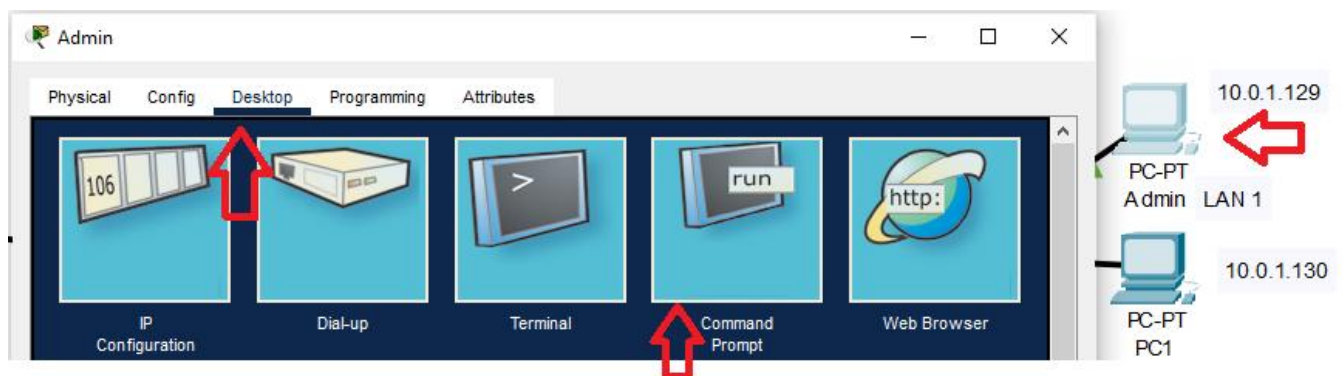
Le réseau est configuré comme suit :

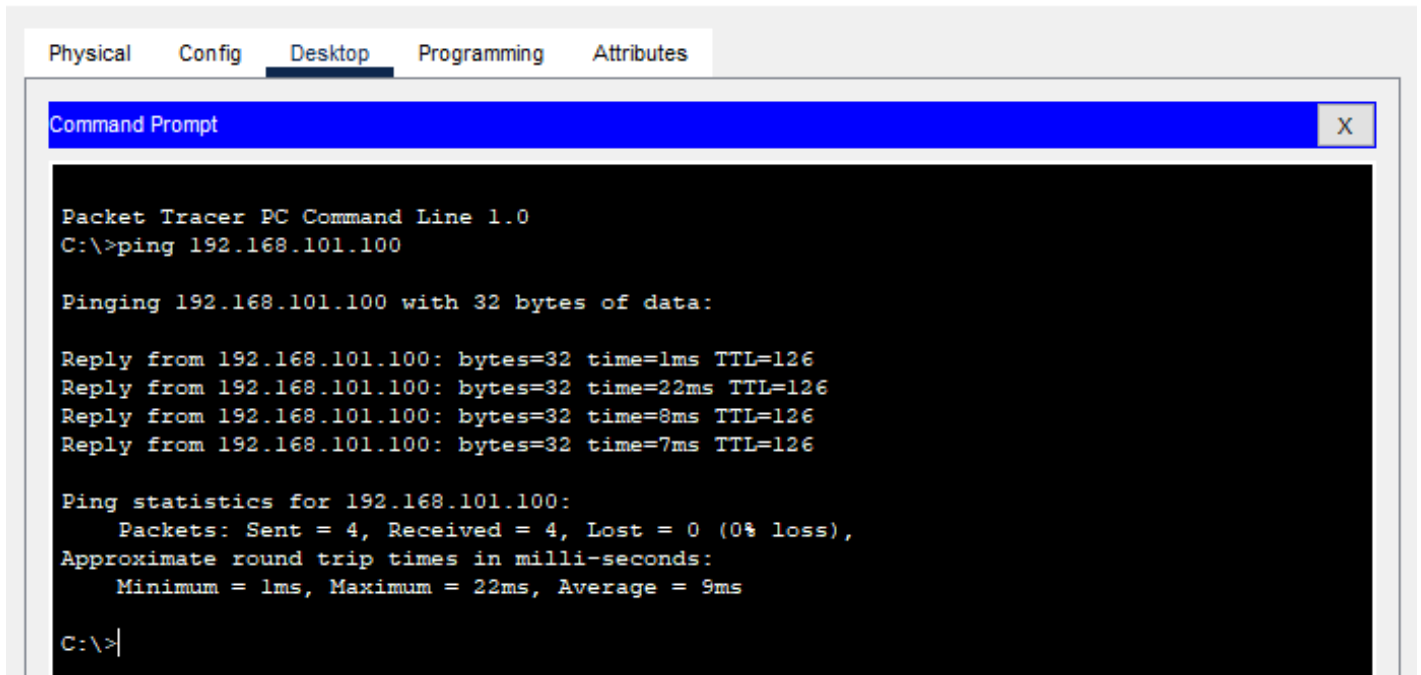
- Les routeurs exécutent OSPFv2
- SSH est activé sur tous les appareils avec utilisateur Cisco et mot de passe cisco123!
- Le sous réseau 192.168.101.0 /24 n'a pas d'hôte.
- L'hôte du sous réseau 192.168.102.0 /24 est configuré statiquement.
- Le routeur R3 joue le rôle de serveur DHCPv4 pour le sous réseau 10.0.1.0 /24 appelé LAN1 et le sous réseau 10.0.2.0 /24 LAN2
- Les commutateurs sont de couche 2 (pas de VLAN).
- Tous les commutateurs SWR# ont une adresse de management dans le réseau LAN1

Tâche 1 : A l'aide de la commande ping, vérifiez que tous les appareils sont interconnectés.

Utilisez la ligne de commande sur le PC Admin pour vérifier que tous les périphériques sont accessibles par un ping et que la connectivité de bout en bout est effective. D'autres pings à partir d'autres stations auraient pu être nécessaires pour vérifier la connectivité.

Ex. Ping depuis Admin jusqu'à 192.168.101.100 (Server-PT)





The screenshot shows the 'Admin' window in Packet Tracer. The 'Desktop' tab is selected. A 'Command Prompt' window is open, displaying the output of a ping command to 192.168.101.100. The output shows four successful replies with varying times and a TTL of 126. Ping statistics indicate 4 packets sent, 4 received, and 0% loss, with an average round trip time of 9ms.

```
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.101.100

Pinging 192.168.101.100 with 32 bytes of data:

Reply from 192.168.101.100: bytes=32 time=1ms TTL=126
Reply from 192.168.101.100: bytes=32 time=22ms TTL=126
Reply from 192.168.101.100: bytes=32 time=8ms TTL=126
Reply from 192.168.101.100: bytes=32 time=7ms TTL=126

Ping statistics for 192.168.101.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 22ms, Average = 9ms

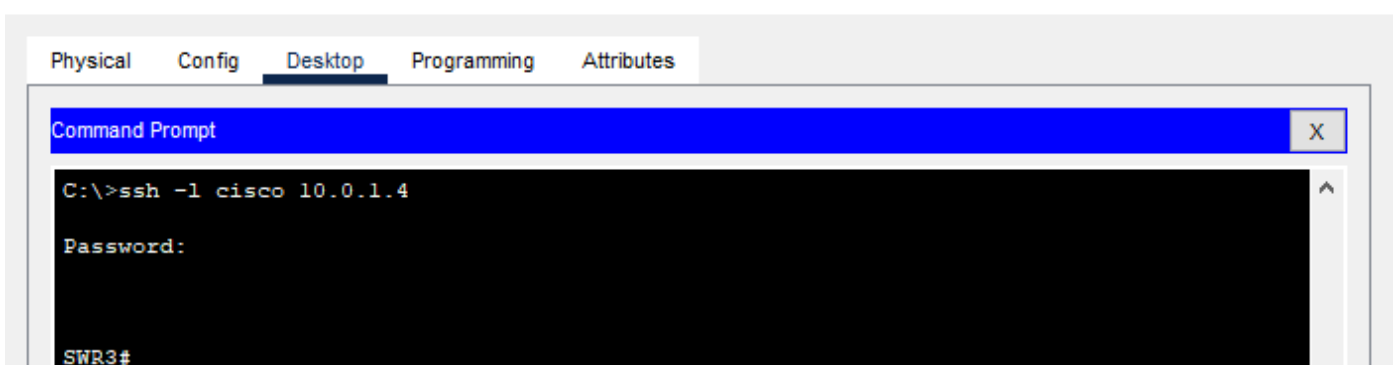
C:\>
```

## Étape 2 : Utiliser l'interface en ligne de commande pour recueillir des informations

Dans cette étape, vous accédez manuellement à chaque périphérique pour recueillir des informations sur la version du logiciel.

Tâche 1 : Depuis le PC d'administration, accédez en toute sécurité au commutateur SWR3.

- Cliquez sur Admin > Desktop > Command Prompt
- Entrez la commande `ssh -l cisco 10.0.1.4`. L'option `-l` est la lettre "L", et non le chiffre un.
- Lorsque vous y êtes invité, entrez `cisco123!` comme mot de passe. Vous êtes maintenant connecté à SWR3.



The screenshot shows the 'Admin' window in Packet Tracer. The 'Desktop' tab is selected. A 'Command Prompt' window is open, displaying the output of an SSH command to 10.0.1.4. The user is prompted for a password and successfully connects to the SWR3 switch, indicated by the prompt change from 'C:\>' to 'SWR3#'.

```
C:\>ssh -l cisco 10.0.1.4

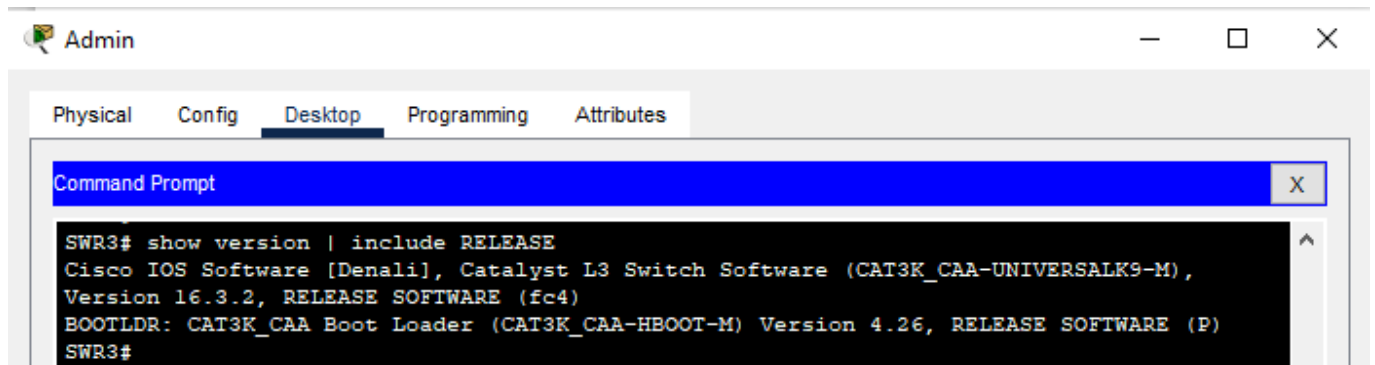
Password:

SWR3#
```

Tâche 2 : Rassemblez des informations sur le système d'exploitation Cisco IOS installé sur SWR3.

- Entrez la commande suivante pour filtrer la sortie de la commande show version afin de ne voir que les informations sur le logiciel installé sur l'appareil. Notez que SWR3 exécute IOS 16.3.2 et Boot Loader 4.2.6.

```
SWR3# show version | include RELEASE
```



Sur un réseau connu ou inconnu les commandes SHOW s'avèrent très utiles pour récolter des informations sur les matériels d'interconnexion et les stations. Toutefois, cela peut s'avérer rapidement fastidieux.

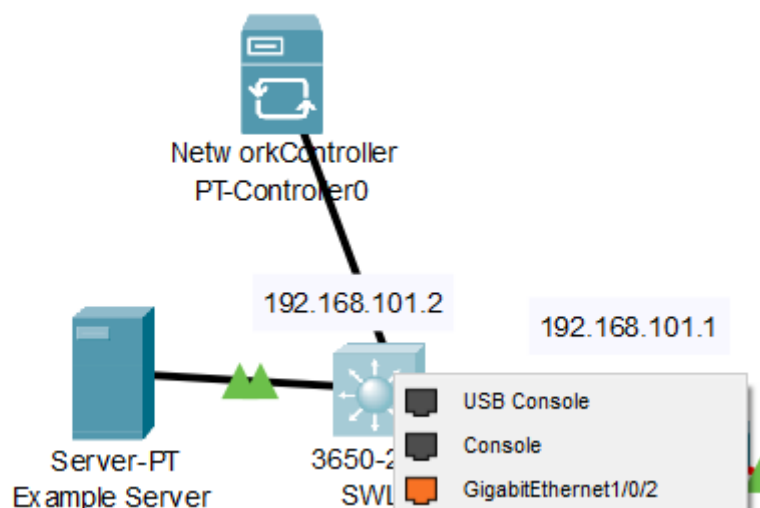
Depuis de nombreuses années, les administrateurs réseau ont utilisé des outils d'automatisation tels que des scripts bash ou des logiciels compatibles SNMP (Simple Network Management Protocol, protocole de communication qui permet aux administrateurs réseau de gérer les équipements du réseau, de superviser et de diagnostiquer des problèmes réseaux et matériels à distance) pour effectuer un processus similaire à celui que vous avez fait à l'étape précédente. Avec l'introduction du SDN, ce processus a été considérablement amélioré. Packet Tracer fournit un contrôleur SDN simple pour simuler un contrôleur SDN.

### Étape 3 : Explorer la topologie du réseau

Dans cette partie, vous allez connecter et configurer le contrôleur SDN simulé dans Packet Tracer.

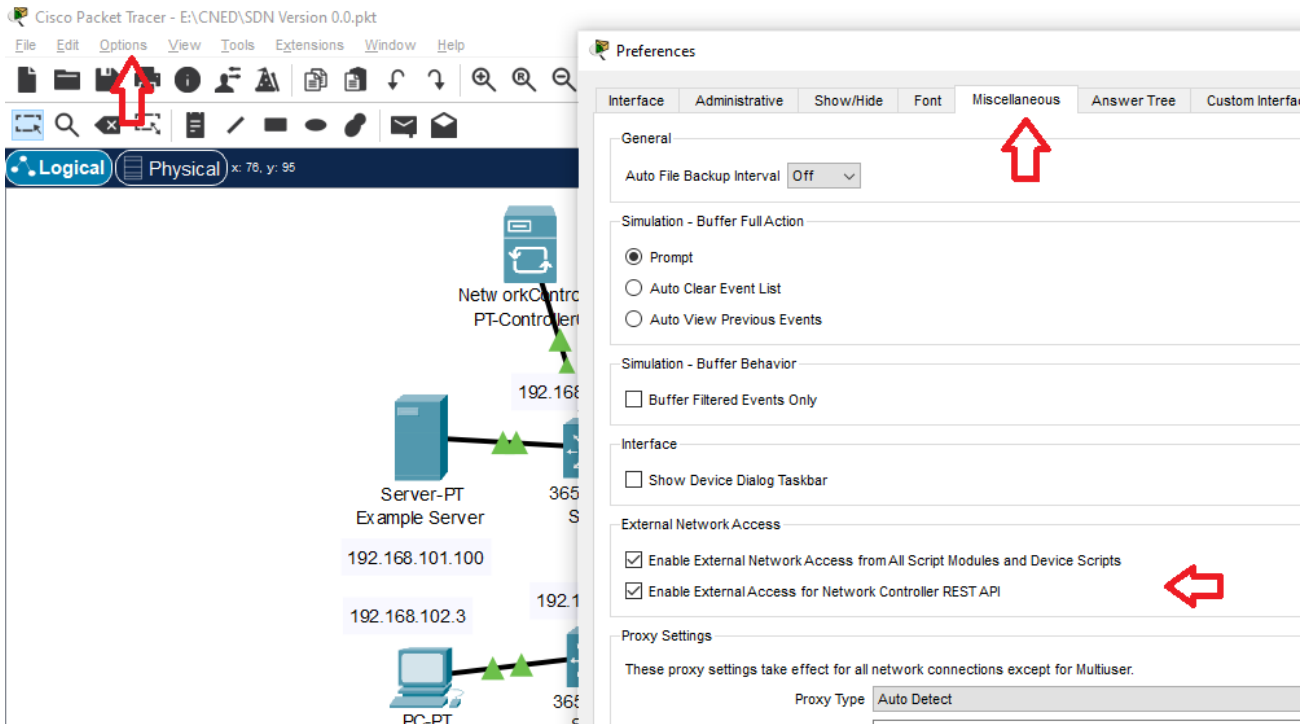
Tâche 1 : Ajoutez un contrôleur réseau à la topologie.

- Dans le coin inférieur gauche de l'interface de Packet Tracer, cliquez sur **End Devices > Network Controller**
- Ajoutez le contrôleur réseau dans la zone vide gauche du commutateur SWL1. Le nom doit être **PT-Controller0**. Si ce n'est pas le cas, cliquez sur le nom et modifiez-le.
- En bas, cliquez sur le boulon en forme d'éclair pour **Connections**. Cliquez sur le câble Copper Straight-Through noir.
- Cliquez sur PT-Controller0 et choisissez GigabitEthernet0. Cliquez ensuite sur SWL1 et choisissez la première interface Gigabit Ethernet disponible.

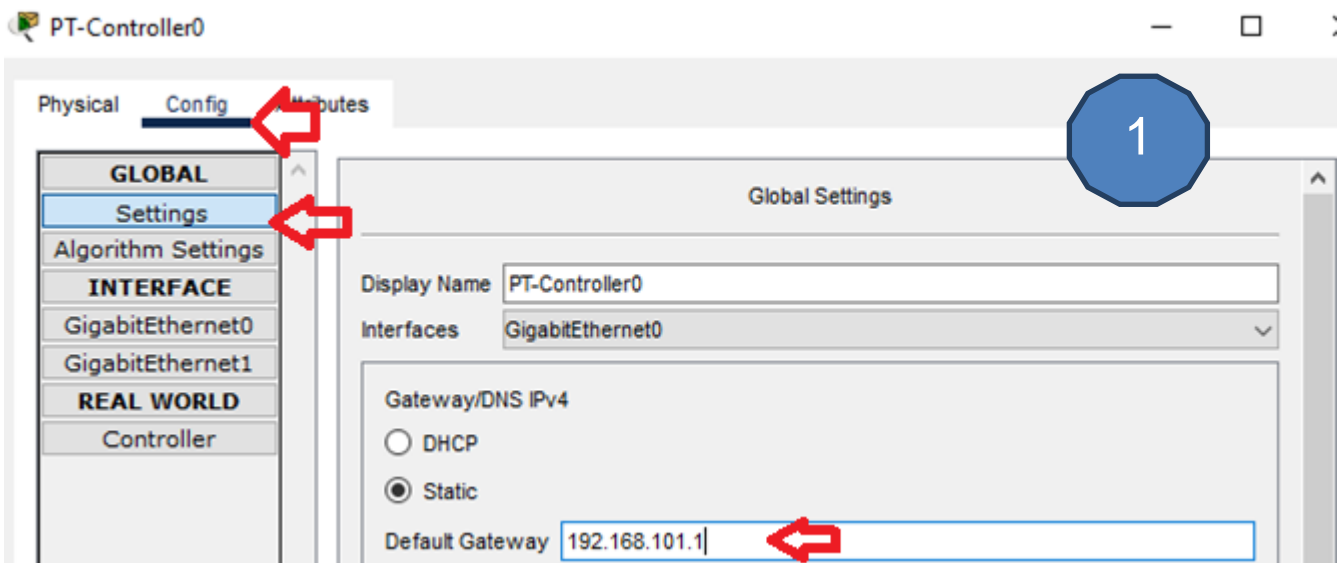


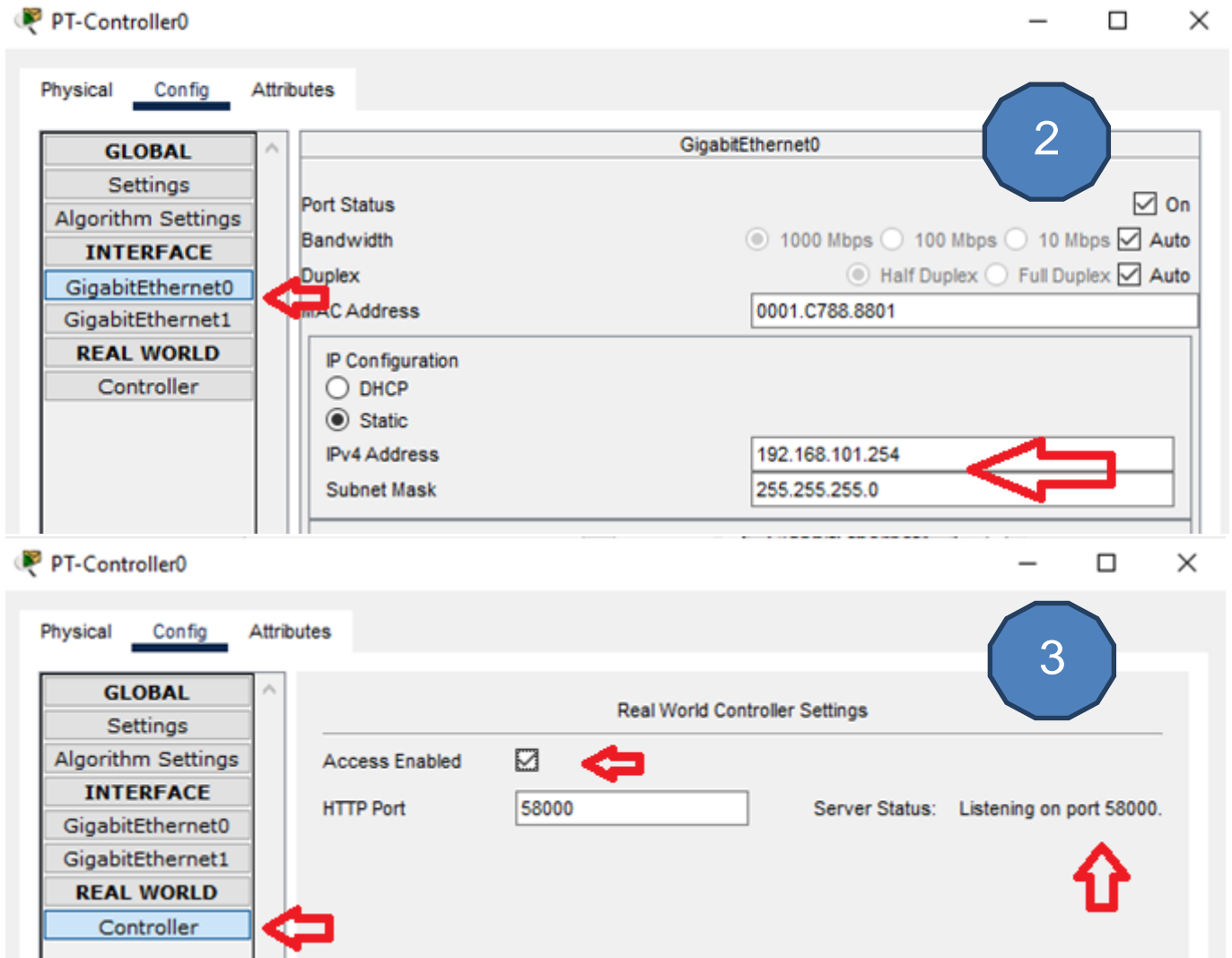
Tâche 2 : Configurez la connectivité pour le contrôleur PT-0 en paramétrant son adresse IPV4 et sa passerelle ainsi qu'en permettant un accès par navigateur extérieur à Packet Tracer sur le port 58000

- Sélectionnez Options > Preferences dans les menus Packet Tracer.
- Cliquez sur Miscellaneous (DIVERS).
- Sous External Network Access, cliquez sur Enable External Access for Network Controller REST API.
- Fermez Preferences et cliquez sur PT-Controller0



- Cliquez sur PT-Controller0 > Config.
- Pour Gateway/DNS IPv4, entrez 192.168.101.1 comme adresse de passerelle.
- Sur la gauche, sous INTERFACE, cliquez sur GigabitEthernet0.
- Pour la configuration IP, entrez l'adresse IP 192.168.101.254 et le masque de sous-réseau 255.255.255.0.
- Sur la gauche, sous REAL WORLD, cliquez sur Contrôleur.
- L'état du serveur doit être arrêté. Cliquez sur Access Enabled pour l'activer. L'état du serveur passe à l'écoute sur le port 58000. Si le port est une autre valeur, changez-la en 58000. Il s'agit du numéro de port dans les scripts Python.





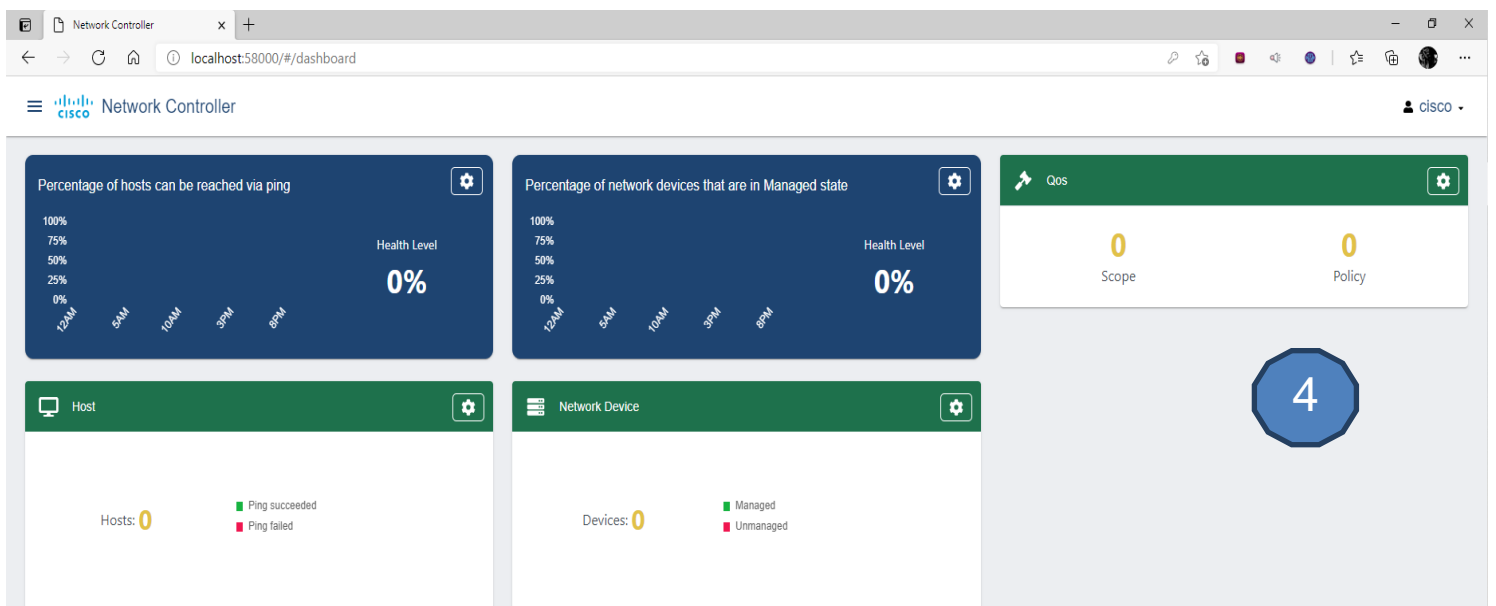
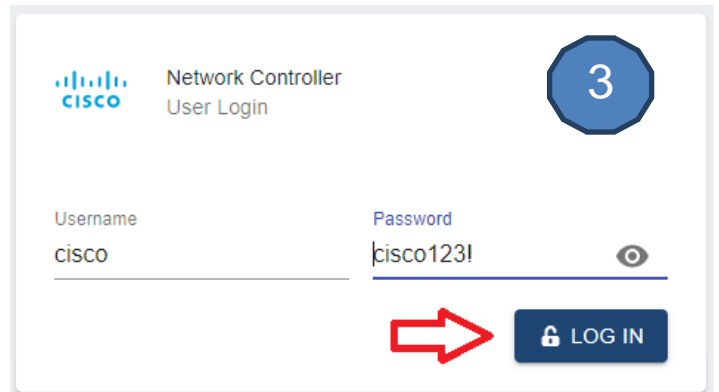
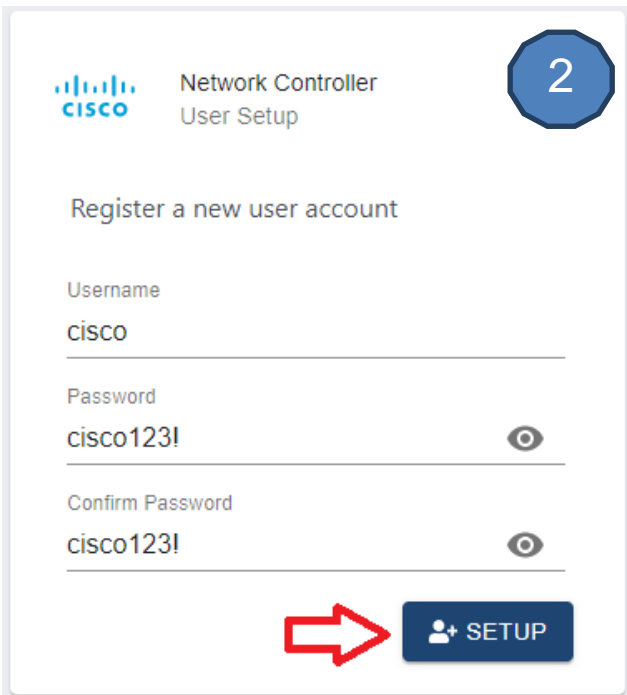
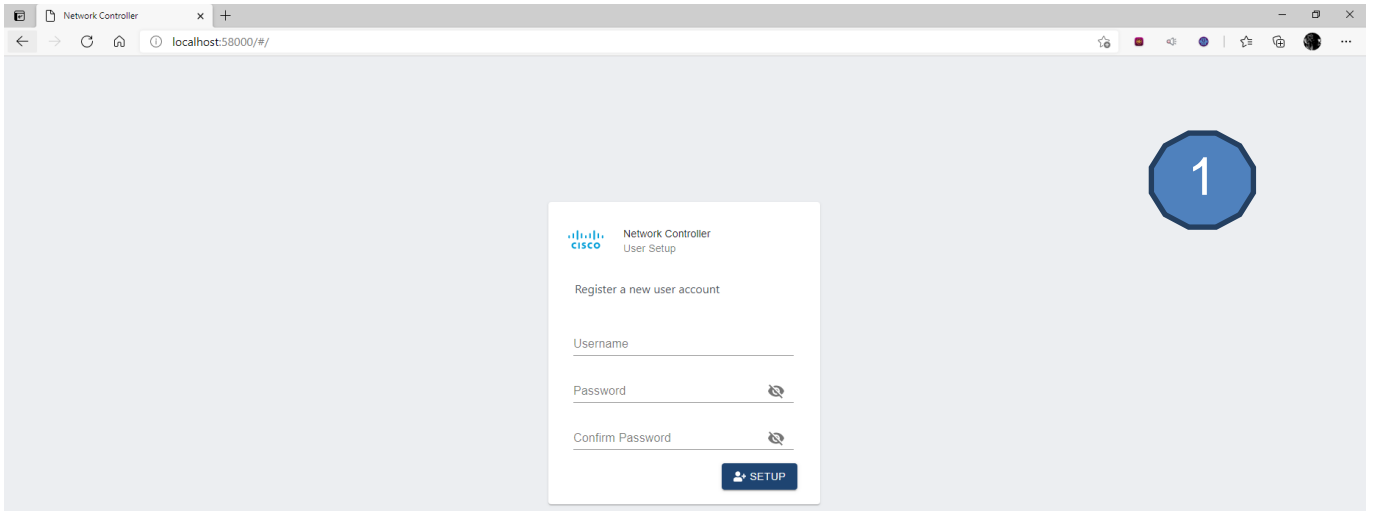
Tâche 3 : Configurez la connectivité pour le contrôleur PT-0.

- Vérifiez que l'administrateur peut pinguer PT-Controller0. (Depuis le PC Admin)
- Si vous n'êtes pas en mesure de pinguer, assurez-vous que votre configuration correspond aux spécifications de l'étape précédente.

Tâche 4 : Enregistrez un nouvel utilisateur et connectez-vous au PT-Controller0.

- Depuis votre propre PC, à l'extérieur de PT, tapez dans un navigateur <http://localhost:58000> pour accéder à la configuration utilisateur pour PT-Controller0.
- Entrez cisco dans le champ Nom d'utilisateur et cisco123! dans les champs Mot de passe et Confirmer le mot de passe, puis cliquez sur SETUP
  - **Remarque** : Vous pouvez utiliser le nom d'utilisateur et le mot de passe que vous voulez ici. Pour plus de simplicité, il est recommandé d'utiliser les informations d'identification courantes utilisées dans le reste de l'activité.
- Sur l'écran Connexion utilisateur, saisissez vos informations d'identification et cliquez sur LOGIN.
- Vous êtes maintenant connecté au tableau de bord pour PT-Controller0. À ce stade, il peut être utile d'agrandir la fenêtre afin que vous puissiez voir toute l'interface.

(Description par capture d'écran à la page suivante).





## Étape 4 : Utiliser un contrôleur SDN pour découvrir une topologie

Dans cette partie, vous allez configurer PT-Controller0 pour utiliser Cisco Discover Protocol (CDP) et découvrir automatiquement les neuf périphériques réseau de votre topologie. Le contrôleur PT-0 découvrira également les cinq périphériques hôtes connectés au réseau

Tâche 1 : Ajoutez des informations d'identification pour accéder à tous les périphériques réseau de la topologie.

- Dans l'interface web du contrôleur réseau, cliquez sur le bouton de menu situé à gauche du logo Cisco.
- Sélectionnez Provisioning. À partir de là, vous pouvez ajouter manuellement des périphériques réseau. Toutefois, vous utiliserez CDP pour découvrir automatiquement les appareils à votre place.
- Cliquez sur CREDENTIALS, puis cliquez sur+ CREDENTIAL pour ajouter New Credential.
- Pour Nom d'utilisateur, entrez cisco, et pour Mot de passe, entrez cisco123!. Laissez le champ Enable Password vide. Pour Description, entrez les informations d'identification d'administrateur (admin credentials), puis cliquez sur OK.
- Les nouvelles informations d'identification de la CLI sont maintenant stockées sur PT-Controller0 pour une utilisation dans les tâches d'automatisation.

The screenshots illustrate the process of adding a credential in the Network Controller web interface:

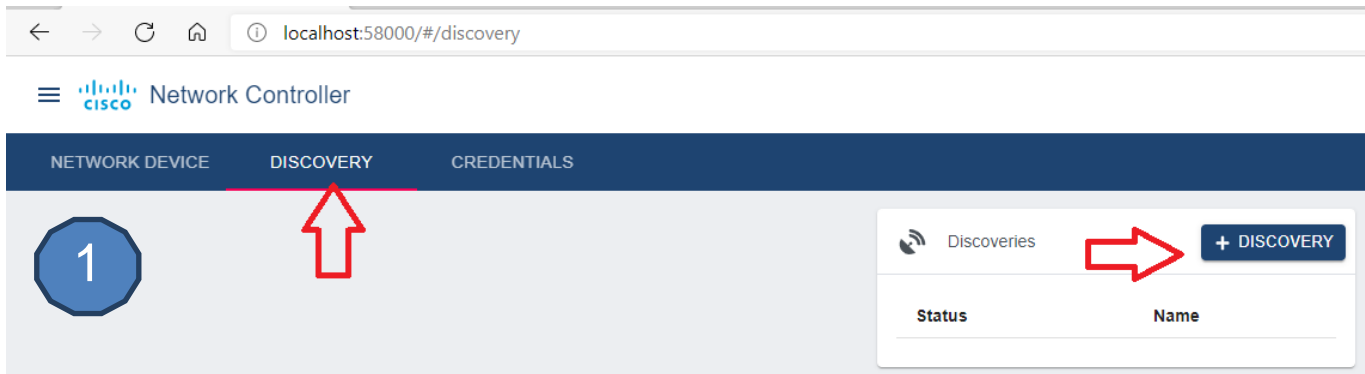
- Step 1:** The user clicks the menu icon (three horizontal lines) on the left side of the Network Controller header.
- Step 2:** The user navigates to the Provisioning section in the left sidebar.
- Step 3:** The user clicks the up arrow icon and then the + CREDENTIAL button in the CREDENTIALS section.
- Step 4:** The user fills out the 'New Credential' form with the following details:
  - Username: cisco
  - Password: cisco123!
  - Enable Password: (unchecked)
  - Description: Admin Credentials
- Step 5:** The credential is successfully added to the list of CLI Credentials.

ID	Username	Description	Action
bacc0a38-5205-4c85-a790-37f828cb0de3	cisco	Admin Credentials	

Tâche 2 : Utilisez CDP pour découvrir tous les périphériques du réseau.

- Cliquez sur DISCOVERY et cliquez sur + DISCOVERY pour ajouter une nouvelle découverte.
- Pour **Nom**, entrez SWL1. Pour **Adresse IP**, entrez 192.168.101.2. Pour **CLI Credential List**, déroulez la liste et choisissez cisco - Admin Credentials.
- Cliquez sur ADD.
- Vous devriez maintenant voir le Status comme in Progress.

**Remarque** : Vous pouvez attendre que Packet Tracer termine la simulation, également cliquer sur le bouton Fast Forward Time dans la fenêtre de topology processus.



New Discovery

Discovery Type: CDP

Name: SWL1

IP Address: 192.168.101.2

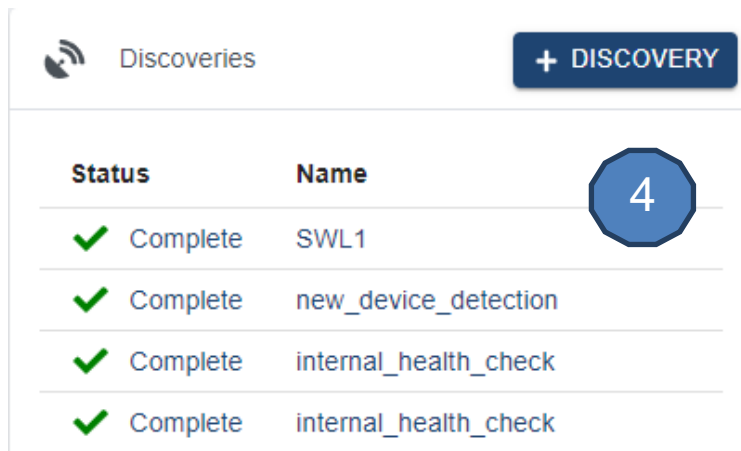
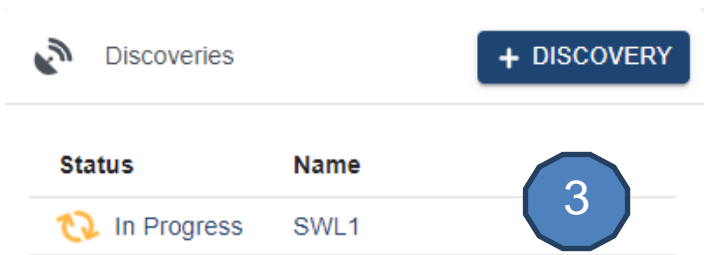
Timeout: 5

Retry: 3

CDP Level: 16

cisco - Admin Credentials

CANCEL ADD



## Étape 5 : Utiliser un contrôleur SDN pour collecter des informations

La découverte réalisée à l'étape précédente a permis de découvrir l'ensemble des autres périphériques connectés au réseau. Elle a été réalisée avec le protocole CDP (Cisco Discovery Protocol). Ce protocole de découverte de réseau permet, avec le protocole SNMP, de trouver d'autres périphériques voisins directement connectés et ceci de proche en proche avec ici une profondeur de 16 sauts.

Grâce à ce mécanisme, vous pouvez voir en cliquant sur SWL1, l'ensemble des périphérique connectés.

Status	Name
✔ Complete	SWL1

SWL1 EDIT START

**Condition:** Complete ✔  
**Status:** Inactive  
**Type:** CDP  
**ID:** 3

**Discovery Details**

CDP Level	Retry Count	TimeOut	IP Range
16	3	5	192.168.101.2

**CLI Credentials**

ID	Username	Description
f8fda37c-8523-492a-bc53-43545c44bd49	cisco	Admin credentials

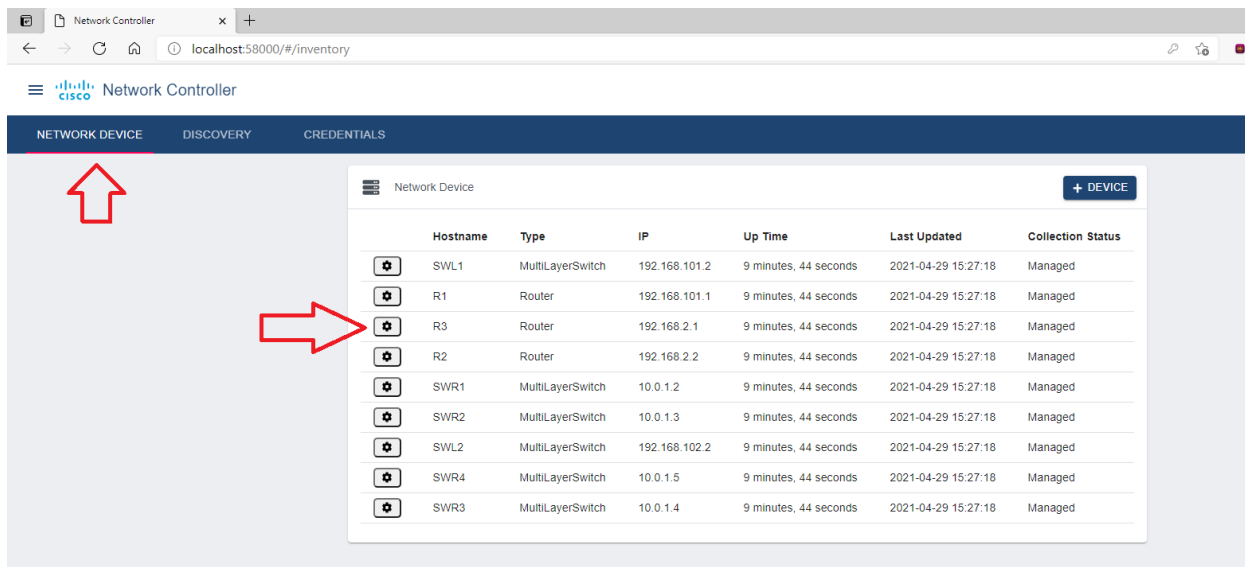
**Discovered Devices**

Hostname	Type	IP	Reachability Status
		0.0.0.0	Unreachable
R3	Router	10.0.1.1	Reachable
PC1	Pc	10.0.1.129	Reachable
Admin	Pc	10.0.1.130	Reachable
SWR1	MultiLayerSwitch	10.0.1.2	Reachable
SWR2	MultiLayerSwitch	10.0.1.3	Reachable
SWR3	MultiLayerSwitch	10.0.1.4	Reachable
SWR4	MultiLayerSwitch	10.0.1.5	Reachable
R3	Router	10.0.2.1	Reachable
PC2	Pc	10.0.2.129	Reachable
PC3	Pc	10.0.2.130	Reachable
R3	Router	192.168.1.1	Reachable
R1	Router	192.168.1.2	Reachable
R1	Router	192.168.101.1	Reachable
Example Server	Server	192.168.101.100	Reachable
SWL1	MultiLayerSwitch	192.168.101.2	Reachable
R2	Router	192.168.102.1	Reachable
SWL2	MultiLayerSwitch	192.168.102.2	Reachable
PC4	Pc	192.168.102.3	Reachable
R3	Router	192.168.2.1	Reachable
R2	Router	192.168.2.2	Reachable

Dans cette étape, vous utiliserez l'interface graphique PT-Controller0 pour afficher des informations sur les périphériques réseau et les périphériques hôtes découverts sur le réseau. Vous allez afficher la topologie créée par le contrôleur, puis effectuer une trace de chemin sur le réseau.

Tâche 1 : Affichez la liste des périphériques réseau découverts.

- Cliquez sur NETWORK DEVICE permet de visualiser les neuf périphériques réseaux répertoriés.
- Cliquez ensuite sur l'icône Engrenage en regard du nom d'hôte de n'importe quel appareil pour afficher les informations collectées par le processus de découverte. Notez que la version du logiciel est répertoriée ainsi qu'une variété d'autres informations détaillées sur l'appareil.



Update Network Device

**Device Detail**

Hostname	R3
ID	FDO13024QWP-uuid
Interface Count	6
Software Version	15.4
MAC Address	0009.7CB0.7876
Management IP Address	192.168.2.1
Platform ID	ISR4300
Product ID	ISR4331
Serial Number	FDO13024QWP-
Type	Router
UpTime	11 minutes, 59 seconds

---

**Collection Status** Managed

**Connected Interface Name** GigabitEthernet1/0/1  
GigabitEthernet1/0/1  
Serial0/1/0  
Serial0/1/1

**Connected Network Device Name** SWR1  
SWR2  
R1  
R2

**Connected Network Device Ip Address** 10.0.1.2  
-  
192.168.1.2  
192.168.2.2

**Error Description**

**Inventory Status Detail** Managed

**Last Update Time** 0 seconds

**Last Updated** 2021-04-29 15:29:33

**Reachability Failure Reason**

**Reachability Status** Reachable

**Device Configuration**

cisco - admin Credentials

DELETE
UPDATE CANCEL

Tâche 2 : Affichez la liste de tous les périphériques hôtes découverts.

- Retour au tableau de bord. Cliquez sur le menu en regard du logo Cisco, puis sur Tableau de bord. (Vous pouvez également cliquer simplement sur la bannière Network Controller pour revenir au tableau de bord de n'importe où.
- Sur le tableau de bord, vous verrez des graphiques indiquant le nombre d'hôtes pouvant être atteints via ping et le nombre de périphériques réseau gérés. Les deux devraient être à 100%.
- Vous devriez également voir les mosaïques pour QoS, Network Device et Host.
- Cliquez sur l'icône Engrenage de Host. Cela vous mènera à l'onglet HOSTS du menu ASSURANCE.
- Sur cette page, vous pouvez afficher toutes les informations de connectivité des couches 2 et 3 pour chaque hôte ainsi que le périphérique réseau auquel chacun est connecté.
- Cliquez sur l'icône "Engrenage" à côté de n'importe quel hôte pour obtenir des informations plus détaillées

The screenshot shows the Cisco Network Controller dashboard. At the top, there are four main widgets:

- Percentage of hosts can be reached via ping:** A bar chart showing 100% health level across five time periods (15:34:38 to 15:35:03).
- Percentage of network devices that are in Managed state:** A bar chart showing 100% health level across five time periods.
- QoS:** Two cards for 'Scope' and 'Policy', both showing 0.
- Hosts:** A card showing 6 hosts, with a 100% success rate for ping (green) and 0% failure rate (red).
- Network Device:** A card showing 9 devices, with 100% managed (green) and 0% unmanaged (red).

Below these is a navigation bar with tabs: ASSURANCE, HOSTS, TOPOLOGY, PATH TRACE. The 'HOSTS' tab is active, showing a table of host devices and their connections.

	Host Device				Connected Network Device		
	MAC	IP	Hostname	Type	IP	Hostname	Port
⚙️	000A.4113.C0B0	192.168.101.100	Example Server	Server	192.168.101.2	SWL1	GigabitEthernet1/0/3
⚙️	0001.435B.5044	192.168.102.3	PC4	Pc	192.168.102.2	SWL2	GigabitEthernet1/0/24
⚙️	0060.700B.2BC5	10.0.2.129	PC2	Pc	10.0.1.5	SWR4	GigabitEthernet1/0/23
⚙️	0050.0F6E.234D	10.0.2.130	PC3	Pc	10.0.1.5	SWR4	GigabitEthernet1/0/24
⚙️	000B.BE16.D6BA	10.0.1.130	Admin	Pc	10.0.1.4	SWR3	GigabitEthernet1/0/21
⚙️	0001.9747.D29B	10.0.1.129	PC1	Pc	10.0.1.4	SWR3	GigabitEthernet1/0/22

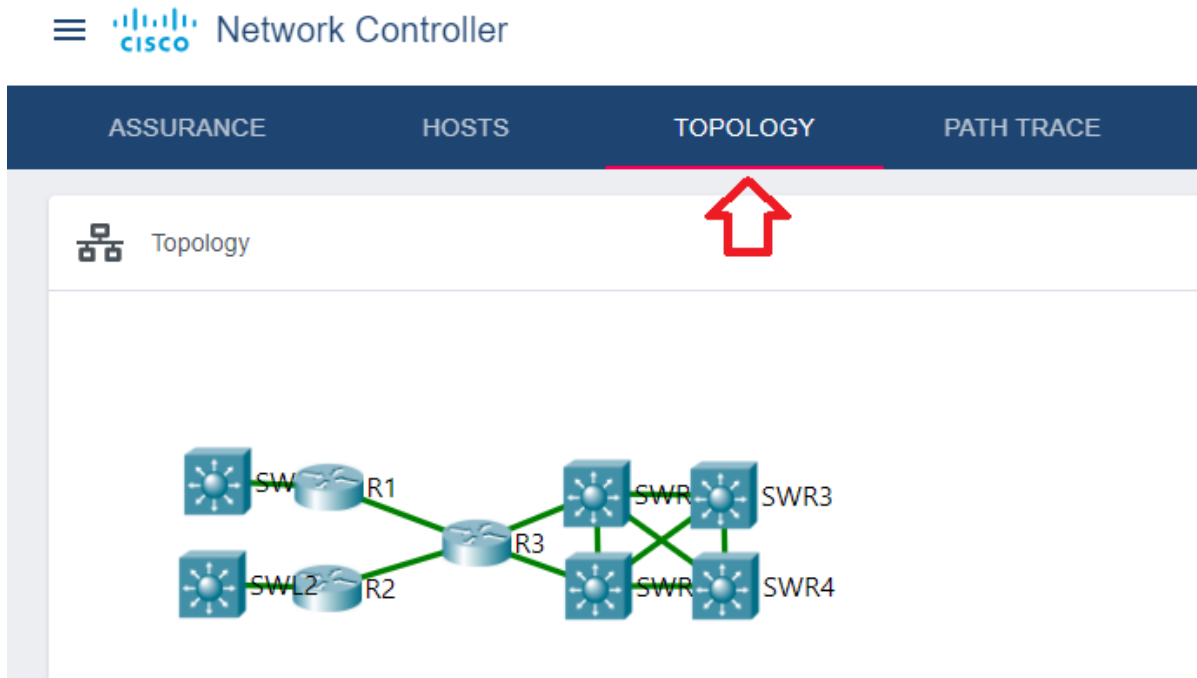
The screenshot shows a 'Host Detail' modal window overlaid on the Hosts table. It provides detailed information for the selected host (MAC: 000A.4113.C0B0).

Host Detail	
Connected AP MAC Address	
Connected AP Name	
Connected Network Interface Name	GigabitEthernet1/0/3
Connected Network Device IP Address	192.168.101.2
Connected Network Device Name	SWL1
Host IP	192.168.101.100
Host MAC	000A.4113.C0B0
Host Name	Example Server
Host Type	Server
ID	PTT08105D89-uuid
Last Updated	2021-04-29 15:36:19
Ping Status	SUCCESS
VLAN ID	

At the bottom of the modal, there are buttons for 'DELETE' and 'CANCEL'.

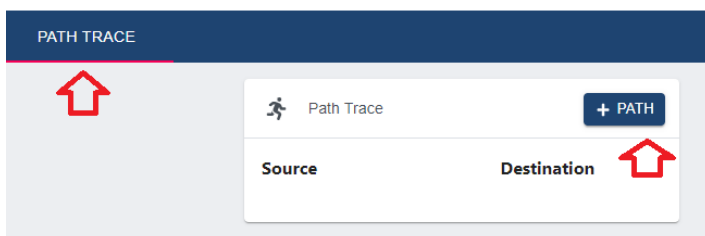
Tâche 3 : Affichez la topologie créée par PT-Controller0.

- Cliquez sur l'onglet TOPOLOGY. Notez que le contrôleur de réseau a créé dynamiquement la même topologie que celle que vous voyez dans la fenêtre principale de Packet Tracer.
- Dans cette vue, vous pouvez cliquer sur n'importe quel périphérique réseau pour afficher ses détails.
- Vous pouvez également cliquer et faire glisser les icônes de périphérique pour réorganiser la topologie.
- Toutefois, vos modifications ne seront pas enregistrées lorsque vous quittez l'espace de travail TOPOLOGY.



Tâche 4 : Tracez le chemin d'un périphérique à un autre périphérique.

- Cliquez sur l'onglet PATH TRACE.
- Cliquez sur + PATH pour ajouter un nouveau chemin.
- Tracez le chemin d'accès d'une extrémité du réseau à l'autre. Par exemple, vous pouvez entrer les adresses IP de PC1 à PC4. Cliquez ensuite sur OK.
- Cliquez sur le nouveau chemin qui a été ajouté pour lancer la trace du chemin.



New Path

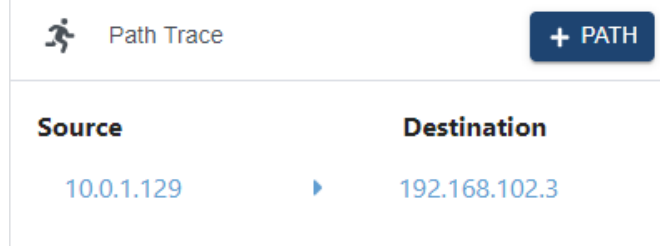
Source

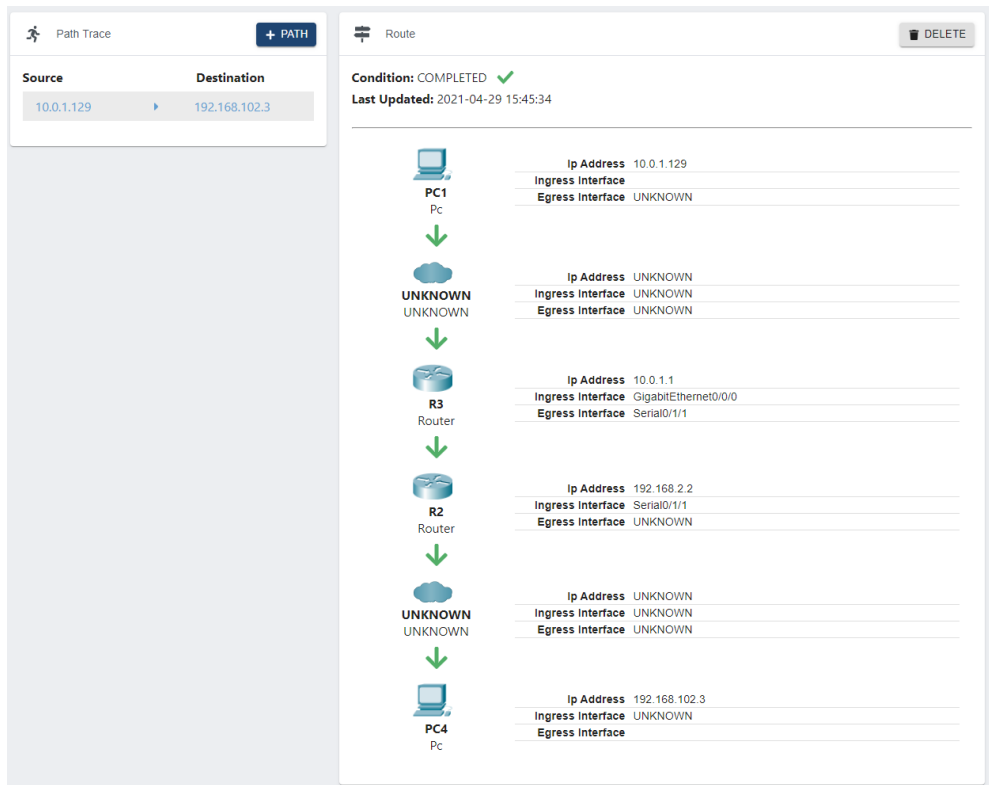
10.0.1.129

Destination

192.168.102.3

CANCEL OKAY





Vous obtiendrez un rapport d'itinéraire qui affiche tous les sauts de la source à la destination. Notez que seules les informations sur l'appareil de couche 3 sont répertoriées. Les commutateurs sont affichés comme un périphérique INCONNU. Cela est dû au fait qu'ils fonctionnent tous à la couche 2 uniquement.

## Étape 6 : Utiliser un contrôleur SDN pour configurer les paramètres réseau

L'un des principaux avantages de l'automatisation du réseau à l'aide d'un contrôleur est la possibilité de configurer les paramètres et les stratégies réseau globaux pour tous les appareils, puis d'appliquer (PUSH) cette configuration d'un simple clic sur un bouton. Dans cette étape, vous allez configurer PT-Controller0 avec les paramètres réseau pour DNS, NTP et Syslog. Vous allez ensuite pousser cette configuration vers les périphériques réseau pris en charge. Enfin, vous allez vérifier et tester la stratégie.

Tâche 1 : Examinez la configuration du serveur Example.

- Cliquez sur Example Server > Services.
- Sous SERVICES, cliquez sur DNS. Notez que le service DNS est activé et qu'il existe un enregistrement pour [www.example.com](http://www.example.com).
- Sous SERVICES, cliquez sur SYSLOG. Notez que le service Syslog est activé.
- Sous SERVICES, cliquez sur NTP. Notez que le service NTP est activé.

The screenshot shows the configuration for the 'Example Server' under the 'Services' tab. The 'DNS' service is enabled (On). A resource record is configured as follows:

No.	Name	Type	Detail
0	www.example.com	A Record	192.168.101.100

## Tâche 2 : Vérifiez l'absence d'une stratégie réseau global

Pour l'instant il n'y a pas de stratégie réseau global pour vos matériels d'interconnexion. Avant d'en créer une et de la déployer avec le contrôleur de réseau, vérifions que les matériels d'interconnexion ne possèdent aucun paramétrage pour le DNS, NTP et SYLOG.

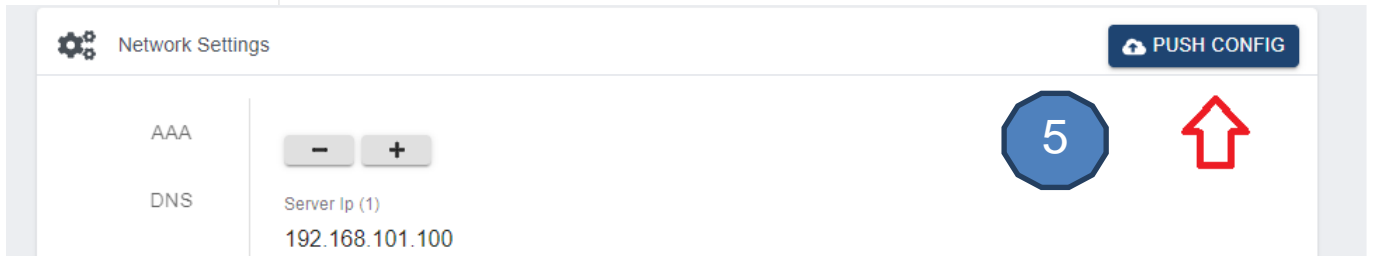
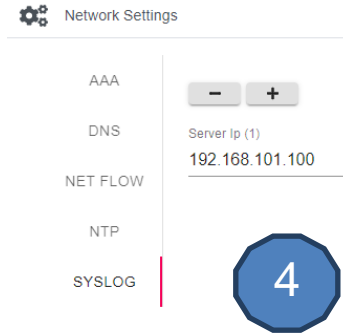
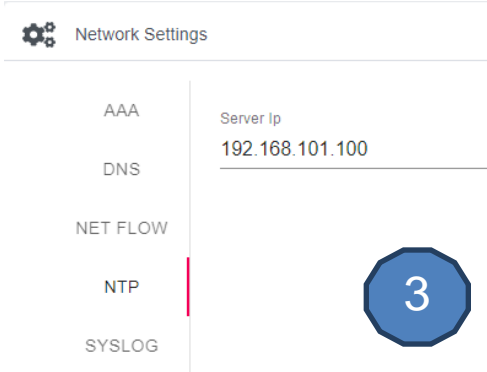
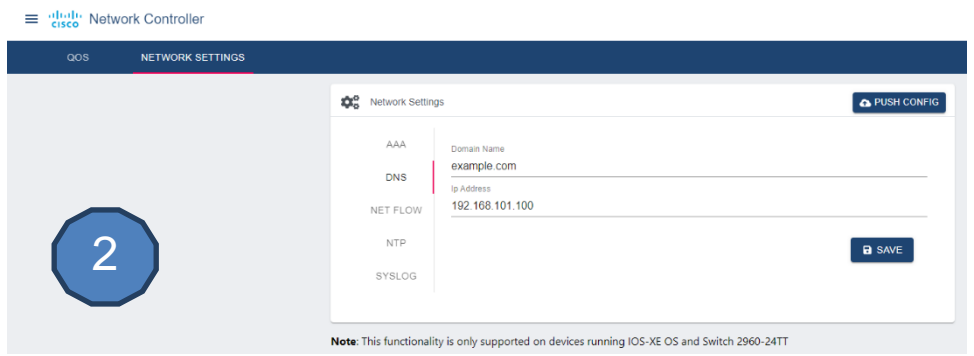
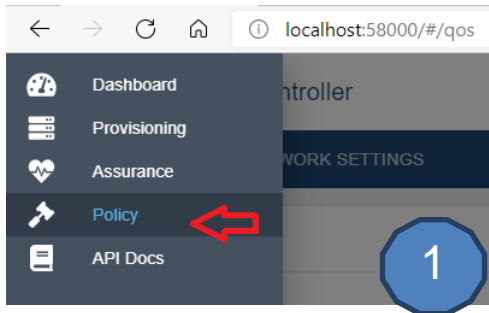
- Comment fonctionnent les commandes ci-dessous et que vont-elles vous apprendre sur les matériels d'interconnexion.
  - `show run | begin ip domain`  
\_\_\_\_\_  
\_\_\_\_\_
  - `show run | begin ip name-`  
\_\_\_\_\_  
\_\_\_\_\_
  - `show ntp associations`  
\_\_\_\_\_  
\_\_\_\_\_
  - `show run | include logging`  
\_\_\_\_\_  
\_\_\_\_\_
- Utilisez ces commandes à bon escient pour vérifier l'absence de stratégies réseau global.

## Tâche 3 : Configurez une stratégie globale pour DNS, SYSLOG et NTP.

- Rendez-vous à nouveau sur le navigateur web vous permettant d'administrer le SDN. Si vous avez fermé votre navigateur ouvrez le et authentifiez vous à nouveau.
- Cliquez sur le menu situé à gauche du logo Cisco.
- Cliquez sur Policy. Dans l'onglet QOS, notez qu'il existe des options pour configurer l'étendue et la stratégie. Dans cette activité, vous allez configurer NETWORK SETTINGS.
- Cliquez sur NETWORK SETTINGS.
- Cliquez sur DNS. Entrez `example.com` comme domaine Name et `192.168.101.100` comme adresse IP.
- Cliquez sur Save
- Cliquez sur NTP. Entrez `192.168.101.100` comme adresse IP.
- Cliquez sur Save.
- Cliquez sur SYSLOG. Entrez `192.168.101.100` comme adresse IP.
- Cliquez sur Save.
- Cliquez à nouveau sur DNS, NTP et SYSLOG pour vérifier que les informations sont correctes. Si ce n'est pas le cas, corrigez les informations enregistrées à chaque fois.
- Cliquez sur PUSH CONFIG.
- La boîte de dialogue Push All Network Settings s'ouvre. Vérifiez vos paramètres et cliquez sur OKAY. Un message "Saved Successfully" s'affiche brièvement.

*(Description par capture d'écran à la page suivante).*



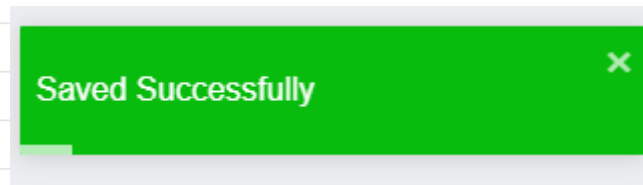


### Push All Network Settings

The following configurations will be saved and applied to the network wide settings.

Note: If you have not saved yet, please do so first.

AAA	
Ip Address	-
Key	-
DNS	
Domain Name	192.168.101.100
Ip Address	example.com
NETFLOW	
Reflection Ip	-
Port Number	-
NTP	
Server Ip	192.168.101.100
SYSLOG	
Server Ip (1)	192.168.101.100

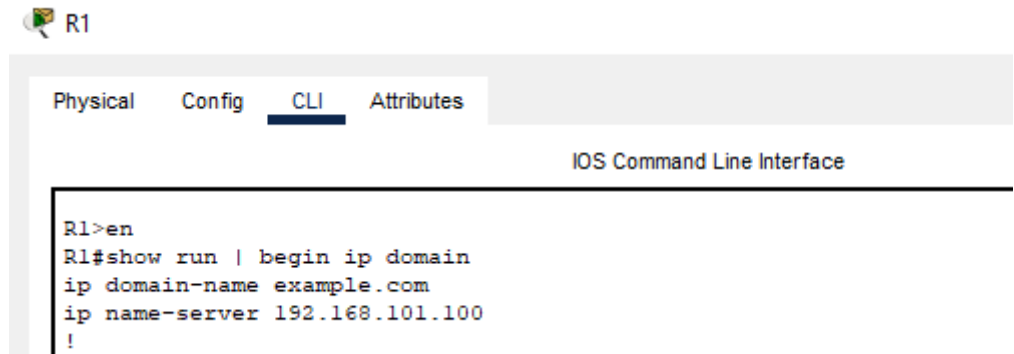


CANCEL OKAY

Tâche 4 : Vérifiez et testez les paramètres réseau qui ont été poussés sur les périphériques.

- Cliquez sur l'un des trois routeurs.
- Cliquez sur CLI.
- Cliquez à l'intérieur de la fenêtre et appuyez sur Enter pour obtenir une invite de commande.
- Entrez le mode EXEC privilégié et vérifiez les paramètres DNS

```
R1> enable
R1# show run | begin ip domain
ip domain-name example.com
ip name-server 192.168.101.100
!
<...>
```



- Entrez les commandes suivantes pour vérifier les paramètres du NTP. L'heure sur R1 doit correspondre à votre heure actuelle. Packet Tracer peut prendre un peu de temps pour propager les messages NTP. Vous pouvez cliquer sur le bouton Fast Forward Time pour accélérer le processus.

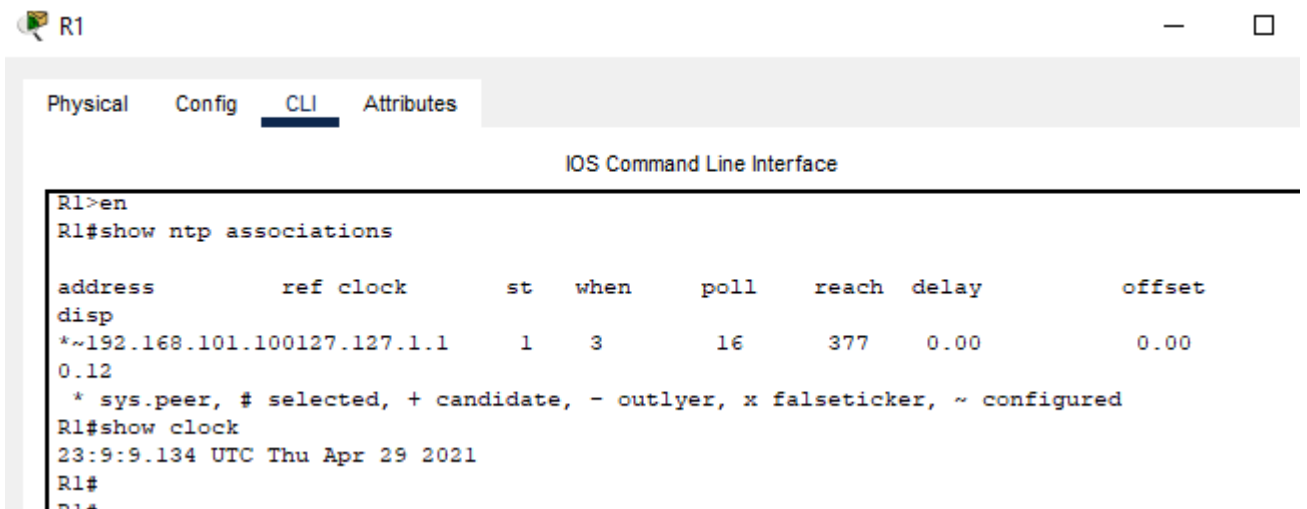
```
R1# show ntp associations
```

```
address ref clock st when poll reach delay offset
disp
*~192.168.101.100127.127.1.1 1 12 16 377 0.00 0.00 0.12 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~
configured
```

```
R1# show clock
```

```
23 :9 :9.134 UTC Thu. Apr 29 2021
```

```
R1#
```



- Entrez la commande suivante pour vérifier que la journalisation est configurée.

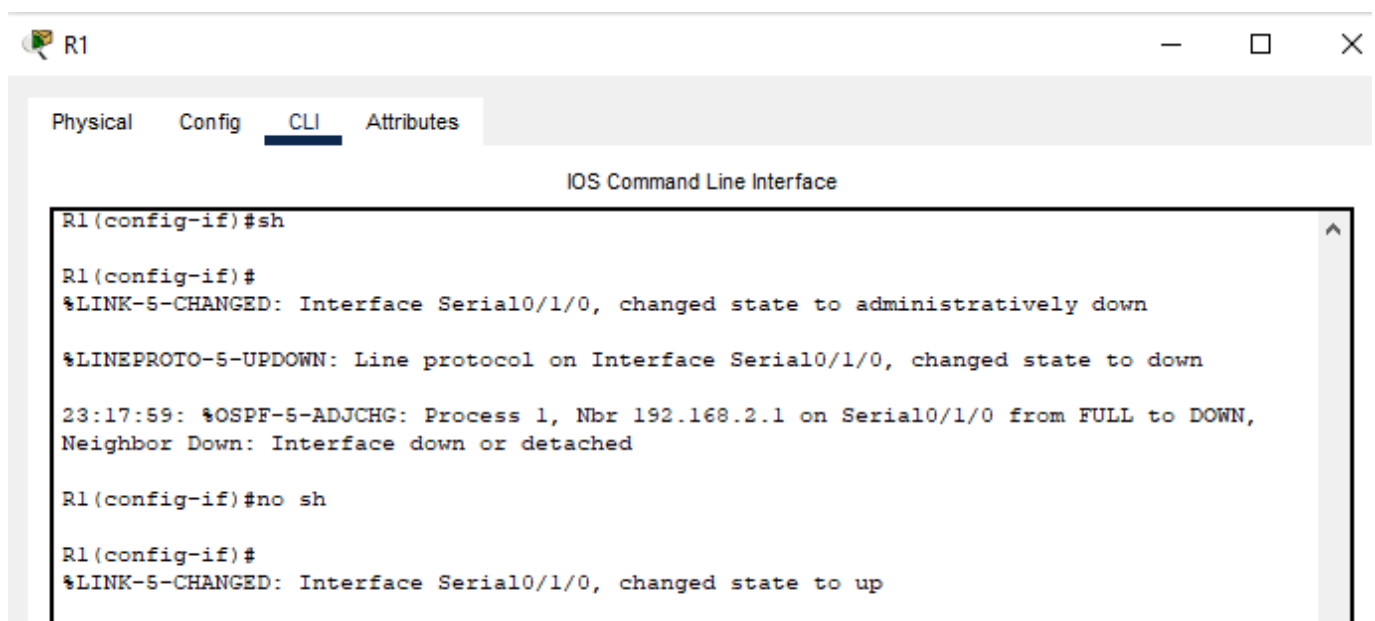
```
R1# show run | include logging
logging 192.168.101.100
R1#
```



- Pour tester la journalisation, arrêtez l'interface Serial0/1/0, puis réactivez-la.

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface s0/1/0
R1(config-if)# shutdown
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, changed state to down 15:36:37: %OSPF-5-ADJCHG:
Process 1, Nbr 192.168.2.1 on Serial0/1/0 from FULL to DOWN,
Neighbor Down: Interface down or detached

R1(config-if)# no shutdown
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, changed state to up
15:36:53: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.1 on Serial0/1/0 from LOADING to FULL, Loading Done
R1(config-if)# end
```



Cliquez sur Example Server > Services > SYSLOG. Vous devriez voir les mêmes messages syslog que vous avez vus dans l'interface de ligne de commande qui sont également enregistrés sur le serveur. Double-cliquez sur l'une des entrées pour consulter les message

The screenshot shows the 'Example Server' configuration interface. The 'Services' tab is active, and 'SYSLOG' is selected in the left-hand menu. The main area displays a 'Syslog' configuration window with a table of messages. The table has columns for 'Time', 'HostName', and 'Message'. There are 10 rows of data. Two red arrows point to rows 5 and 6. A tooltip is visible over row 4, showing the message text: '%LINK-5-CHANGED: Interface Serial0/1/0...'.

Service	Time	HostName	Message
1	-	192.168.101.2	%LINK-5-CHANGED: Interfa...
2	-	192.168.101.2	%LINEPROTO-5-UPDOWN: ...
3	-	192.168.101.1	%SYS-5-CONFIG_: ...
4	-	192.168.101.1	%SYS-6-LOGGINGHOST_...CHANGED: Interface Serial0/1/0...
5	-	192.168.101.1	%LINK-5-CHANGED: Interfa...
6	-	192.168.101.1	%LINEPROTO-5-UPDOWN: ...
7	-	192.168.101.1	23:17:59: %OSPF-5-ADJCH...
8	-	192.168.101.1	%LINK-5-CHANGED: Interfa...
9	-	192.168.101.1	%LINEPROTO-5-UPDOWN: ...
10	-	192.168.101.1	23:18:17: %OSPF-5-ADJCH...

Quels sont les inconvénients d'un paramétrage classique en ligne de commande CLI ?

---



---



---

Quels sont les avantages de l'utilisation du contrôleur SDN pour le paramétrage des matériels ?

---



---



---

Quel(s) type(s) d'erreur(s) ce déploiement évite-il ?

---



---



---

Existe-t-il d'autres solutions pour déployer des paramètres de matériels de façon centralisée ?

---



---



---

## Deuxième Partie : Implémenter les API REST avec un contrôleur SDN

Dans cette partie, vous utiliserez le contrôleur réseau de Packet Tracer et la documentation de l'API associée pour envoyer des requêtes REST à partir de Postman. Packet Tracer prend également en charge un environnement de codage Python.

### Objectifs

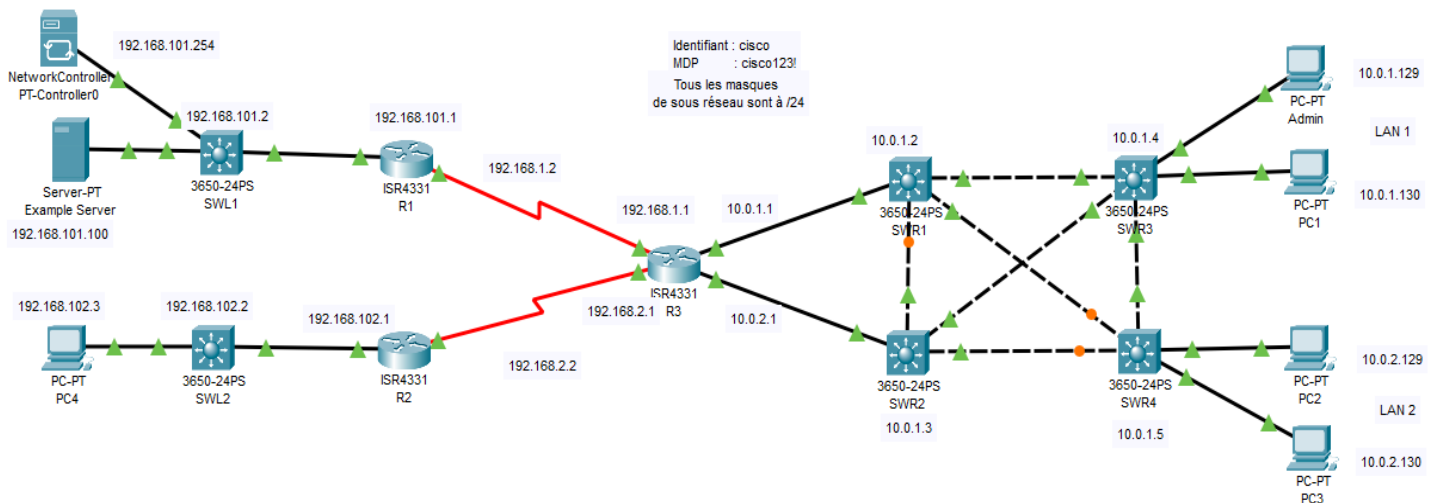
Étape 1 : Vérifier la connectivité externe avec Packet Tracer

Étape 2 : Demander un jeton d'authentification avec Postman

Étape 3 : Envoyer des demandes REST avec Postman

### Schéma de la maquette :

(Identique avec un contrôleur de réseau préinstallé, vous utiliserez SDN Version LAB 2.pkt mais vous pouvez également continuer sur votre premier fichier s'il est correctement configuré)



### Étape 1 : Vérifier la connectivité externe avec Packet Tracer

Dans cette étape, vous vérifierez que Packet Tracer est accessible par d'autres applications.

Tâche 1 : Vérifiez les paramètres de Packet Tracer pour l'accès externe.

- Cliquez sur **Options > Preferences > Miscellaneous**. Sous **External Network Access**, vérifiez que **Enable External Access for Network Controller REST API** est cochée.
- Fermez la fenêtre **Preferences**.
- Cliquez sur **PT-Controller0 > Config**.
- Sur la gauche, sous **REAL WORLD**, cliquez sur **Controller**.
- Cochez **Access Enabled** et notez le numéro de port, qui est très probablement 58000 à moins qu'il ne soit plus disponible ce sera alors le prochain disponible (ex 58001). Il s'agit du numéro de port dont vous aurez besoin lorsque vous accédez à l'activité Packet Tracer depuis un navigateur, VS Code et Postman.

(Capture d'écran Cf. Première Partie > Etape 3 > Tâche 2 page 6)

Tâche 2 : Vérifiez que vous pouvez accéder à Packet Tracer à partir d'un autre programme. (Pour la suite de la présentation nous considérerons que c'est le port 58000 qui est utilisé).

- Ouvrez votre navigateur et accédez à <http://localhost:58000/api/v1/host> .

Vous obtiendrez la réponse suivante. Cette étape vérifie que vous pouvez accéder en externe à Packet Tracer et PT-Controller0. Notez que l'autorisation nécessite un ticket. Vous obtiendrez un jeton d'autorisation dans la partie suivante.

```
{
  "response": {
    "detail": "Security Authentication Failure",
    "errorCode": "REST_API_EXTERNAL_ACCESS",
    "message": "Ticket-based authorization: empty ticket."
  },
  "version": "1.0"
}
```

## Étape 2 : Demander un jeton d'authentification avec Postman

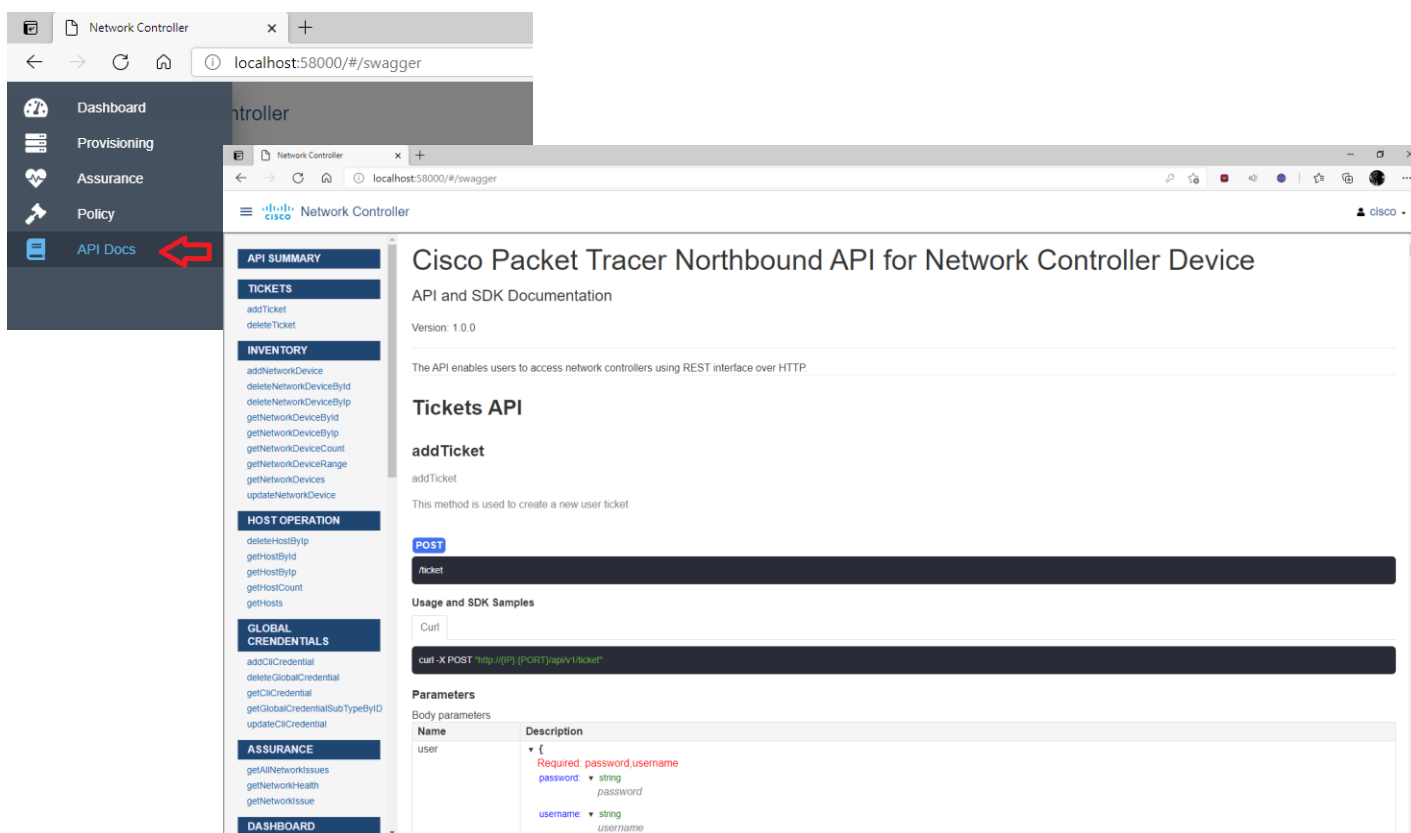
Dans cette étape, vous allez examiner la documentation de l'API REST dans Packet Tracer et utiliser Postman pour demander un jeton d'authentification à partir du PT-Controller0.

Lien de Téléchargement de Postman : [Téléchargement de Postman](#)

Vous pourrez également le faire plus tard dans VS Code avec un script Python.

Tâche 1 : Examinez la documentation de l'API REST pour le contrôleur réseau.

- Ouvrir votre navigateur web et saisir <http://localhost:58000>
- Connectez-vous avec l'utilisateur cisco et le mot de passe cisco123!.
- Cliquez sur le menu en regard du logo Cisco et choisissez API Docs.



Vous pouvez également accéder à cette même documentation à partir du menu Aide.

- Dans Packet Tracer Cliquez sur **Help > Contents**.
- Dans le volet de navigation à gauche, faites défiler vers le bas environ les deux tiers du chemin et cliquez sur **Network Controller API**. Cela fournit la même documentation que celle trouvée sur PT-Controller0.
- Dans la documentation de l'API, cliquez sur AddTicket. Vous utiliserez cette fonctionnalité par la suite.

**Remarque :** Certaines fonctionnalités de l'API REST peuvent ne pas être disponibles dans la version actuelle de Packet Tracer. Pour la version 8.0 celles précédées par l'icône 1 sont disponibles :

**TICKETS** 1

- addTicket
- deleteTicket

**INVENTORY**

- addNetworkDevice
- deleteNetworkDeviceById
- deleteNetworkDeviceByIp
- getNetworkDeviceById
- getNetworkDeviceByIp
- getNetworkDeviceCount
- getNetworkDeviceRange
- getNetworkDevices
- updateNetworkDevice

**HOST OPERATION**

- deleteHostByIp
- getHostById
- getHostByIp
- getHostCount
- getHosts

**GLOBAL CREDENTIALS**

- addCliCredential
- deleteGlobalCredential
- getCliCredential
- getGlobalCredentialSubTypeById
- updateCliCredential

**ASSURANCE** 1

- getAllNetworkIssues
- getNetworkHealth
- getNetworkIssue

**DASHBOARD**

- getSiteHealth

**DISCOVERY**

- deleteAllDiscovery
- deleteDiscoveryById
- deleteDiscoveryByRange
- getDiscovery
- getDiscoveryById
- getDiscoveryByRange
- getDiscoveryCount
- getNetworkDeviceByDiscoveryId
- getNetworkDeviceByDiscoveryIdBy
- getNetworkDeviceCountByDiscovery
- insertDiscovery
- updateDiscovery

**FLOW ANALYSIS** 1

## API and SDK Documentation

Version: 1.0.0

The API enables users to access network controllers using REST interface over HTTP.

### Tickets API

#### addTicket 2

addTicket

This method is used to create a new user ticket

**POST** 3

`/ticket` 4

#### Usage and SDK Samples

Curl

```
curl -X POST "http://(IP):(PORT)/ticket" 5
```

#### Parameters 6

Body parameters

Name	Description
user	<pre>{   Required: password,username   password: string     password   username: string     username }</pre>

#### Responses 7

Status: 200 - success

Schema

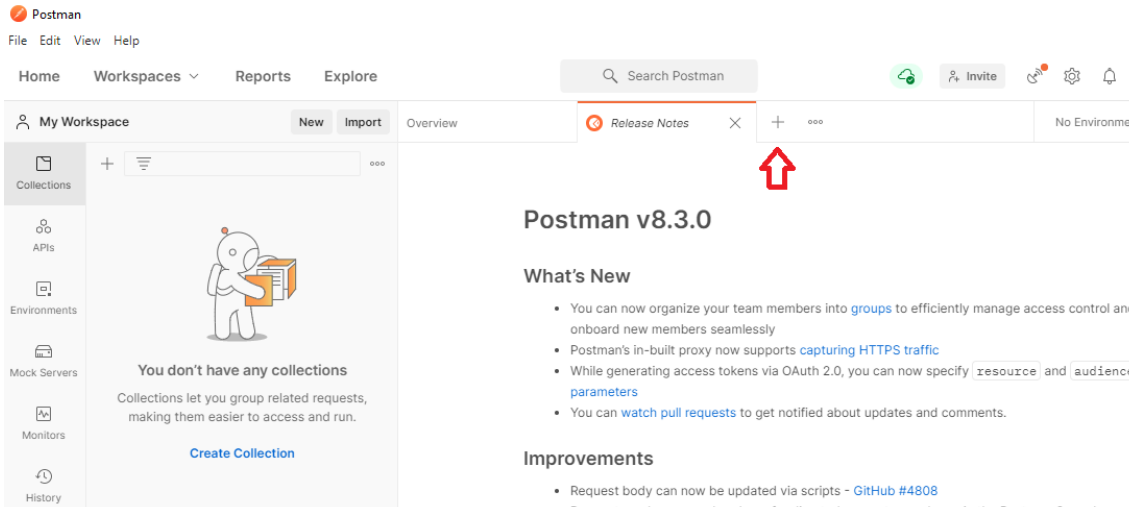
```
{
  version: string
  response: {
    Object used to retrieve the service ticket
    Required: serviceTicket
    idleTimeout: integer (int32)
    serviceTicket: string
      Service Ticket to be used as authentication Ticket
  }
}
```

## Tâche 2 : Créez une nouvelle demande POST

- Après avoir examiné la documentation AddTicket REST API Method, ouvrez Postman
  - Ci-dessous la version utilisée lors de la création de ce Labo :
- Dans la zone de droite, cliquez sur le signe + pour créer une demande sans titre.



Postman for Windows  
Version 8.3.0  
win32 10.0.19041 / x64

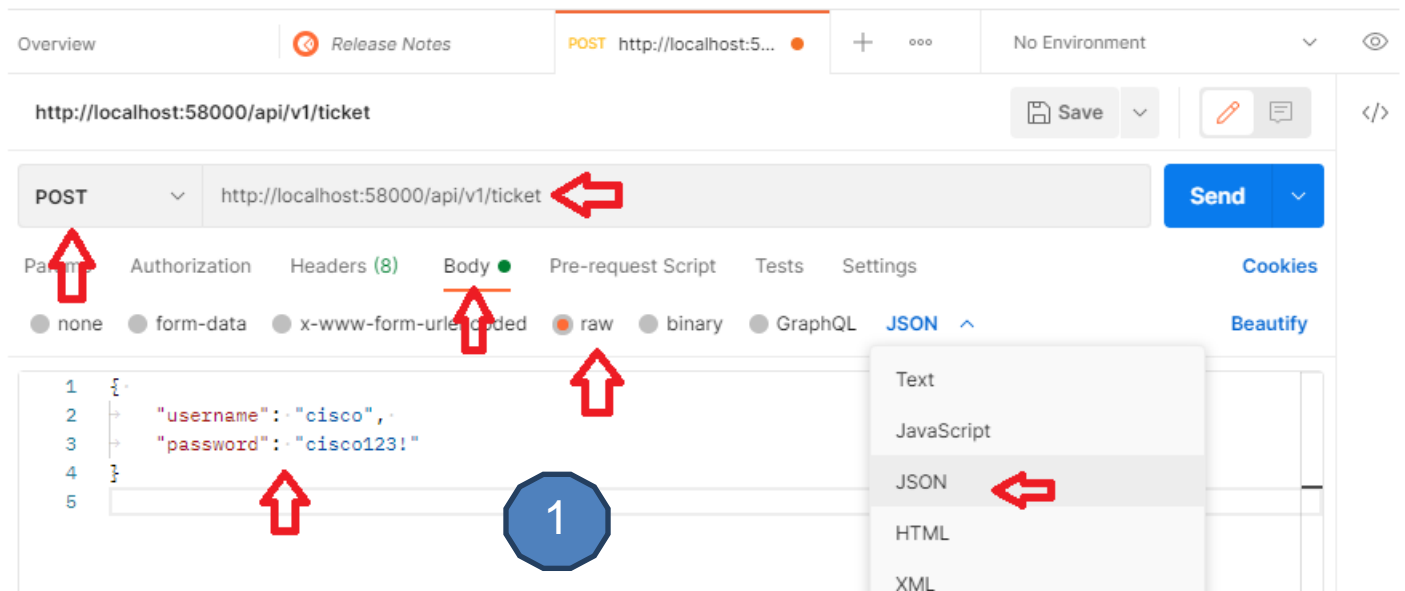


- Cliquez sur la flèche vers le bas et changez le type de **GET** à **POST**.
- Entrez l'URL `http://localhost:58000/api/v1/ticket`.
- Sous le champ **URL**, cliquez sur **Body**. Changez le type en **raw**.
- Cliquez sur la flèche vers le bas en regard de **Text** et changez-la en **JSON**. Cette modification définira également l'en-tête HTTP "Content-type" sur "application/json" qui est requis pour cet appel d'API.
- Collez l'objet **JSON** suivant dans le champ Body. Assurez-vous que votre code est correctement formaté

```
{  
  "username": "cisco",  
  "password": "cisco123!"  
}
```

- Cliquez sur **Send** pour envoyer la demande POST au PT-Controller0. Vous devriez obtenir une réponse similaire à la suivante. Cependant, votre *Numéro de Ticket* sera une valeur réelle.

```
{  
  "response": {  
    "idleTimeout": 900,  
    "serviceTicket": "your_serviceTicket",  
    "sessionTimeout": 3600  
  },  
  "version": "1.0"  
}
```





Overview | Release Notes | POST http://localhost:58000/api/v1/ticket | No Environment

http://localhost:58000/api/v1/ticket

POST http://localhost:58000/api/v1/ticket

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1
2 "username": "cisco",
3 "password": "cisco123!"
4
5

```

Body Cookies Headers (5) Test Results 201 Created 540 ms 346 B Save Response

Pretty Raw Preview Visualize Text

```

1
2 "response": {
3   "idleTimeout": 900,
4   "serviceTicket": "NC-31-32e9ee14690d49039988-nbi",
5   "sessionTimeout": 3600
6 },
7 "version": "1.0"

```

Copiez la valeur du Ticket sans les guillemets dans un fichier texte pour une utilisation ultérieure

Sachant que l'obtention d'un jeton permet l'accès par API au contrôleur et peut être conditionné à un mot de passe et limité dans le temps, quels en sont les avantages :

### Étape 3 : Envoyer des demandes REST avec Postman

Dans cette étape, vous utiliserez votre ticket de service pour envoyer trois demandes REST au contrôleur PT0.

Tâche 1 : Créez une nouvelle requête GET pour tous les périphériques réseau du réseau.

- Dans Postman, cliquez sur le signe + pour créer une Untitled Request.
- Saisissez l'URL `http://localhost:58000/api/v1/network-device`.
- Sous le champ URL, cliquez sur Headers.
- Sous la dernière KEY, cliquez sur le champ Key et entrez X-Auth-Token.
- Dans le champ Value, saisissez la valeur de votre ticket de service.

Overview | Release N... | POST http://loc... | GET http://local... | No Environment

http://localhost:58000/api/v1/network-device.

GET http://localhost:58000/api/v1/network-device.

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

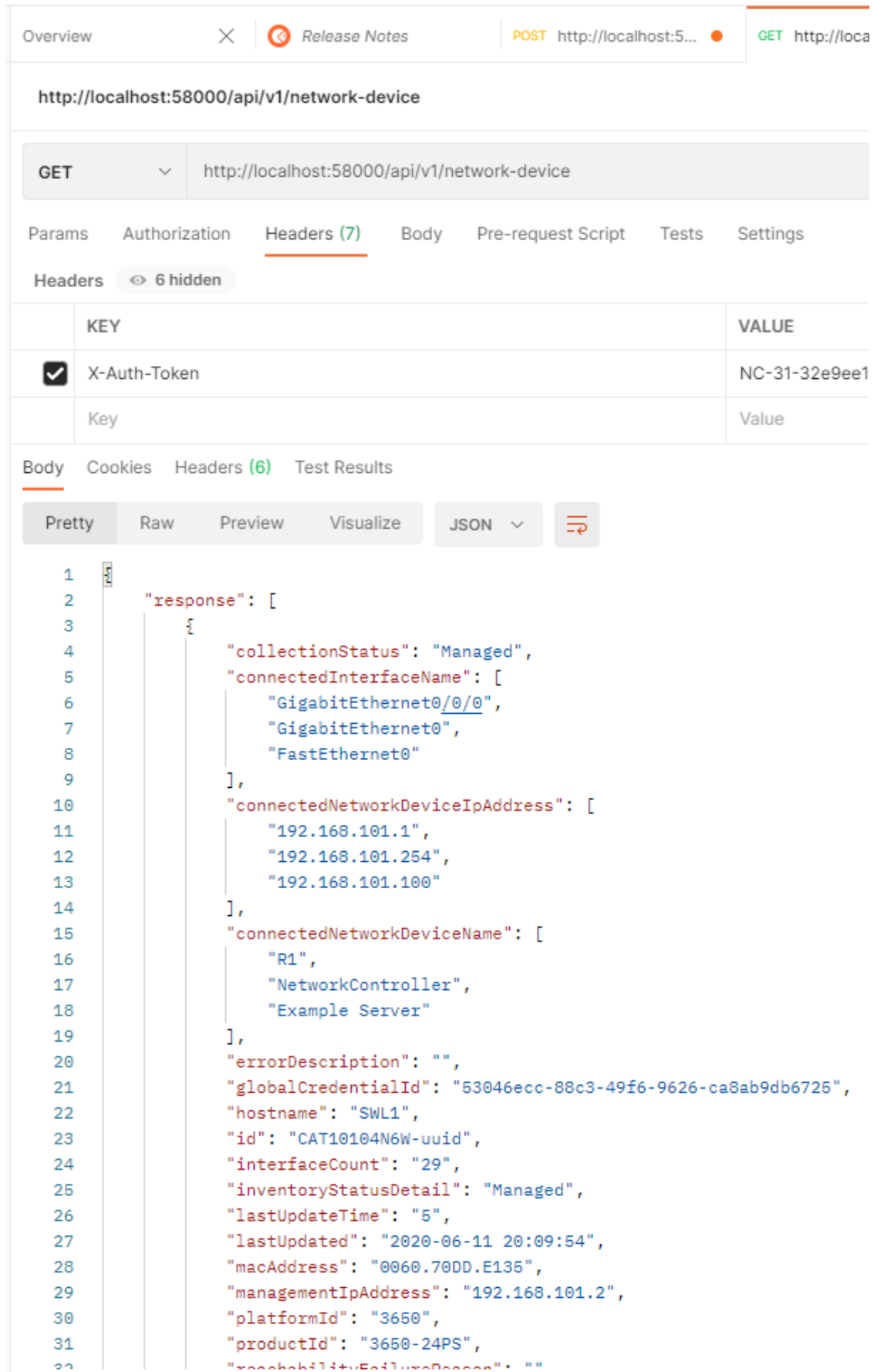
Headers 6 hidden

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> X-Auth-Token	NC-31-32e9ee14690d49039988-nbi	
Key	Value	Description

## Tâche 2 : Envoyez la demande GET.

- Cliquez sur Send pour envoyer la demande GET au PT-Controller0.

Vous devriez obtenir une réponse répertoriant les détails que le contrôleur possède pour les neuf périphériques réseau du réseau.



The screenshot shows a REST client interface with the following details:

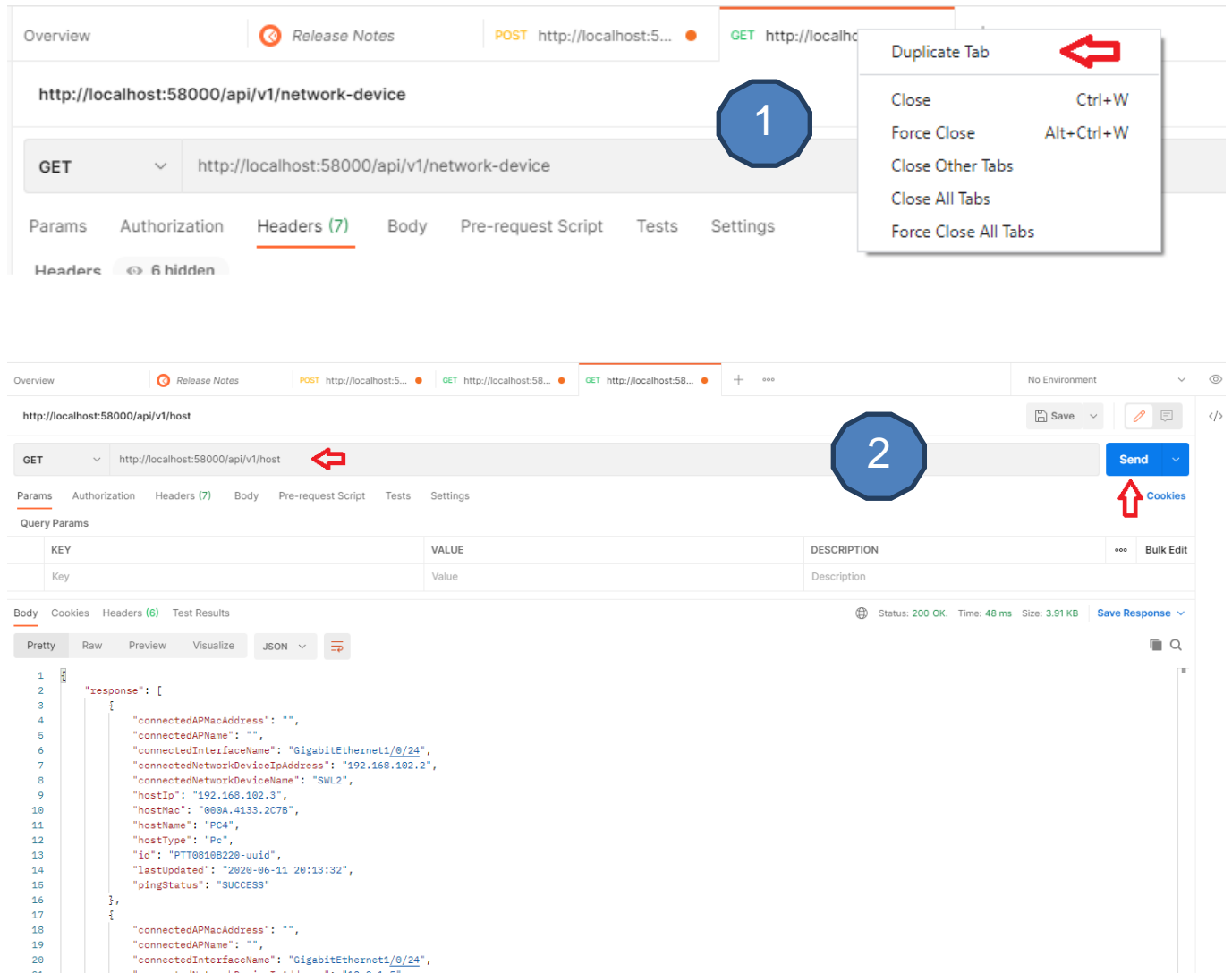
- Request:** GET `http://localhost:58000/api/v1/network-device`
- Headers:** X-Auth-Token: NC-31-32e9ee1
- Response (JSON):**

```
1  {
2    "response": [
3      {
4        "collectionStatus": "Managed",
5        "connectedInterfaceName": [
6          "GigabitEthernet0/0/0",
7          "GigabitEthernet0",
8          "FastEthernet0"
9        ],
10       "connectedNetworkDeviceIpAddress": [
11         "192.168.101.1",
12         "192.168.101.254",
13         "192.168.101.100"
14       ],
15       "connectedNetworkDeviceName": [
16         "R1",
17         "NetworkController",
18         "Example Server"
19       ],
20       "errorDescription": "",
21       "globalCredentialId": "53046ecc-88c3-49f6-9626-ca8ab9db6725",
22       "hostname": "SWL1",
23       "id": "CAT10104N6W-uuid",
24       "interfaceCount": "29",
25       "inventoryStatusDetail": "Managed",
26       "lastUpdateTime": "5",
27       "lastUpdated": "2020-06-11 20:09:54",
28       "macAddress": "0060.70DD.E135",
29       "managementIpAddress": "192.168.101.2",
30       "platformId": "3650",
31       "productId": "3650-24PS",
32       "capabilitiesEnforceAccess": ""
33     }
34   ]
35 }
```

### Tâche 3 : Dupliquez la requête GET et modifiez-la pour tous les hôtes du réseau.

- Dans Postman, cliquez avec le bouton droit sur l'onglet correspondant à votre demande GET hôte et choisissez Duplicate Tab.
- Toutes les informations contenues dans le ticket sont les mêmes, sauf pour l'URL.
- Il suffit de changer le network-device en host : <http://localhost:58000/api/v1/host>.
- Cliquez sur Send pour envoyer la demande GET au PT-Controller0.

2



Vous pouvez fermer PostMan pour vous libérer des ressources et passer à la prochaine étape.

Postman est utilisé pour tester les API.

Quelles informations sont demandées par la requête ? Le retour est-il cohérent ?

---

---

---

---

Recherchez dans les captures d'écran précédentes sur Postman les codes de retour 200 et 201 et donnez leurs significations.

---

---

---

---

# Troisième Partie : Requêtes REST en python

## Objectifs

Étape 1 : Envoyer des demandes REST en Python (avec Microsoft Visual Studio, VS code ou tout autre environnement de développement)

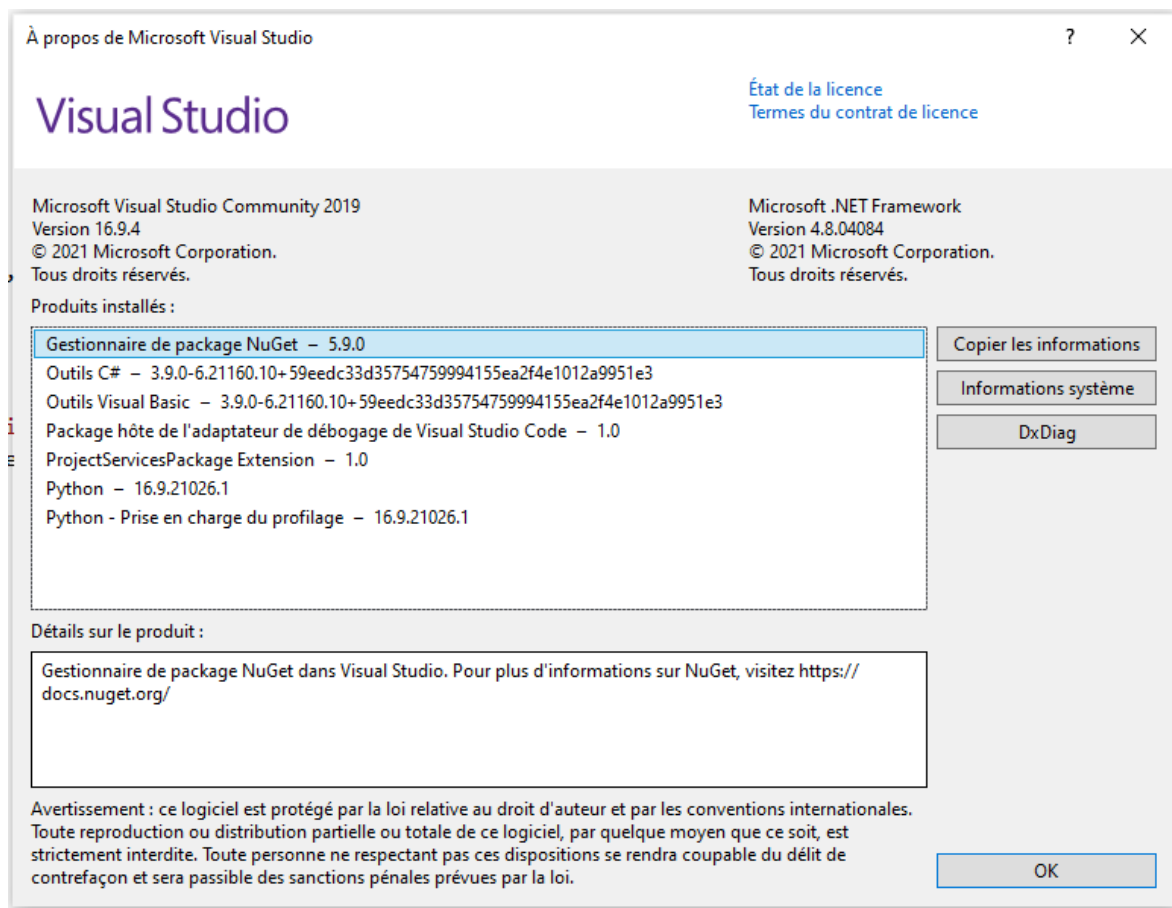
Étape 2 : Envoyer des requêtes REST à l'intérieur du Packet Tracer

## Étape 1 : Envoyer des requêtes REST en Python avec Visual Studio

Dans cette étape, vous utiliserez le script Python dans VS Code pour envoyer les mêmes requêtes d'API que vous avez envoyées dans Postman. Cependant, vous utiliserez également Python pour les boucles pour analyser le JSON et afficher uniquement des paires de valeurs clés spécifiques.

Sur Microsoft Visual Studio, l'installation de Python peut se faire pendant l'installation du logiciel. Sur d'autres environnements comme Visual Studio Code, l'installation de Python se fait séparément.

Version de Microsoft Visual Studio utilisée :



Les 3 programmes utilisés :

- 1) 01\_get-ticket.py
- 2) 02\_get-network-device.py
- 3) 03\_get-host.py

**REMARQUE** : Si vous utilisez Microsoft Visual Studio, rendez vous à l'annexe B pour installer la librairie requests.  
(pip install requests)

- 01\_get-ticket.py

```
import json
import requests

api_url = "http://localhost:58000/api/v1/ticket"

headers = {
    "content-type": "application/json"
}

body_json = {
    "username": "cisco",
    "password": "cisco123!"
}

resp = requests.post(api_url, json.dumps(body_json), headers=headers, verify=False)

print("Ticket request status: ", resp.status_code)
response_json = resp.json()

serviceTicket = response_json["response"]["serviceTicket"]
print("The service ticket number is: ", serviceTicket)
```

- 02\_get-network-device.py

```
import json
import requests

api_url = "http://localhost:58000/api/v1/network-device"

headers={"X-Auth-Token": "NC-99-3808c9f9875e41529ff0-nbi"}

resp = requests.get(api_url, headers=headers, verify=False)

print("Request status: ", resp.status_code)

response_json = resp.json()
networkDevices = response_json["response"]

for networkDevice in networkDevices:
    print(networkDevice["hostname"], "\t", networkDevice["platformId"], "\t",
networkDevice["managementIpAddress"])
```

Votre numéro de ticket sera différent

- 03\_get-host.py

```
import json
import requests

api_url = "http://localhost:58000/api/v1/host"

headers={"X-Auth-Token": "NC-93-18e6f5d304c04501ad1c-nbi"}

resp = requests.get(api_url, headers=headers, verify=False)

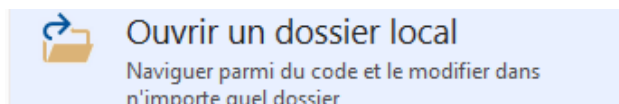
print("Request status: ", resp.status_code)

response_json = resp.json()
hosts = response_json["response"]

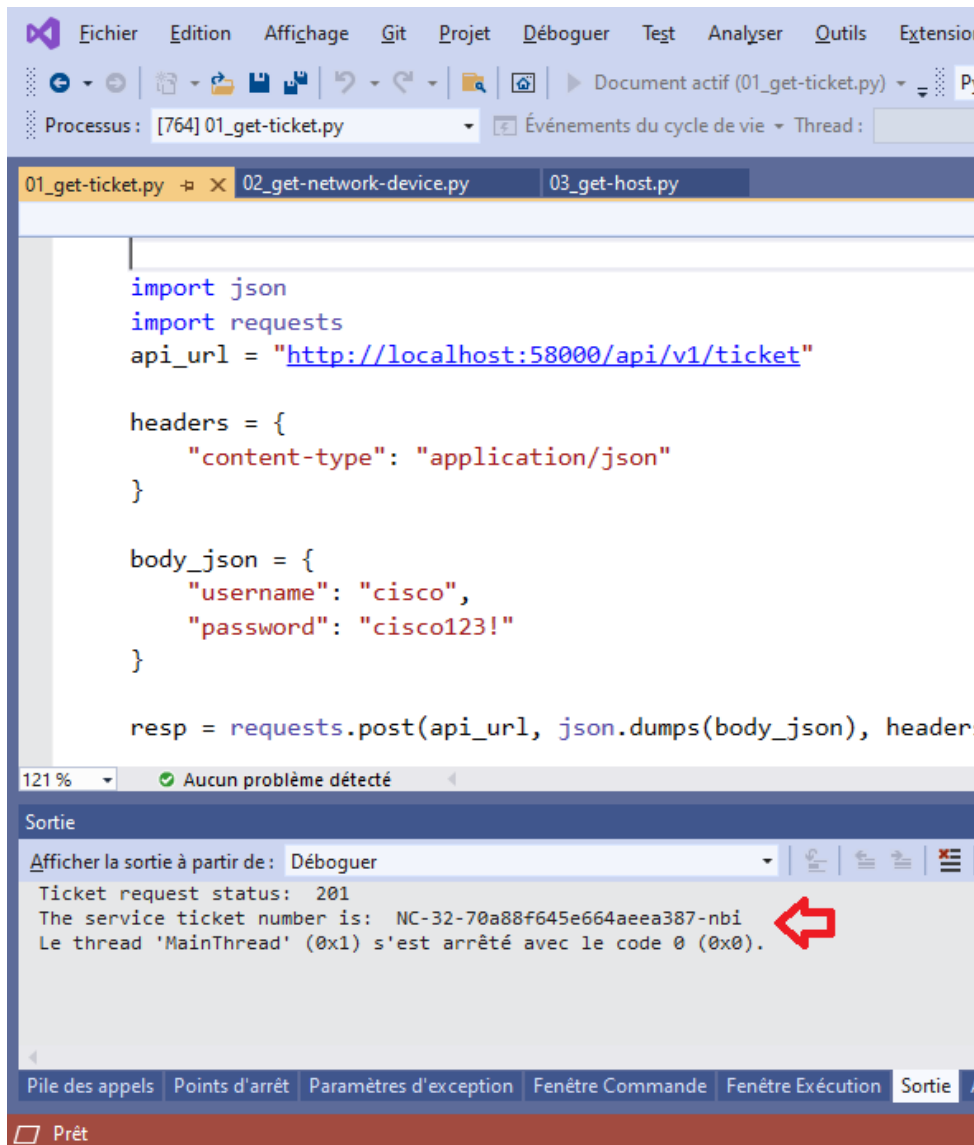
for host in hosts:
    print(host["hostName"], "\t", host["hostIp"], "\t", host["hostMac"], "\t",
host["connectedInterfaceName"])
```

Votre numéro de ticket sera différent

- Ouvrir un dossier local contenant les 3 fichiers



- Placez-vous sur 01\_get-ticket.py et cliquez sur « Démarrer Document actif »
- Récupérer votre numéro de ticket pour l'utiliser dans les deux autres programmes



Quelle est l'utilité du ticket (Token) et quand sera-t-il utilisé ?

---

---

---

---

- Arrêtez le programme précédent.
- Placez-vous sur 02\_get-network-device.py.
- Insérez le numéro de ticket que vous venez d'obtenir.
- Cliquez sur « Démarrer Document actif » pour obtenir le résultat attendu.

REMARQUE : Auparavant dans Postman, l'appel à l'API du périphérique réseau renvoyait une liste des neuf périphériques réseau et toutes les informations disponibles pour chaque périphérique. Cependant, le script 02\_get-network-device.py imprime uniquement les valeurs des clés qui intéressent le programmeur : hostname, PlatformId et ManagementIpAddress.

The screenshot shows a Python IDE with three tabs: 01\_get-ticket.py, 02\_get-network-device.py (active), and 03\_get-host.py. The active tab contains the following Python code:

```
import requests
api_url = "http://localhost:58000/api/v1/network-device"

headers={"X-Auth-Token": "NC-32-70a88f645e664aeea387-nbi"}

resp = requests.get(api_url, headers=headers, verify=False)

print("Request status: ", resp.status_code)

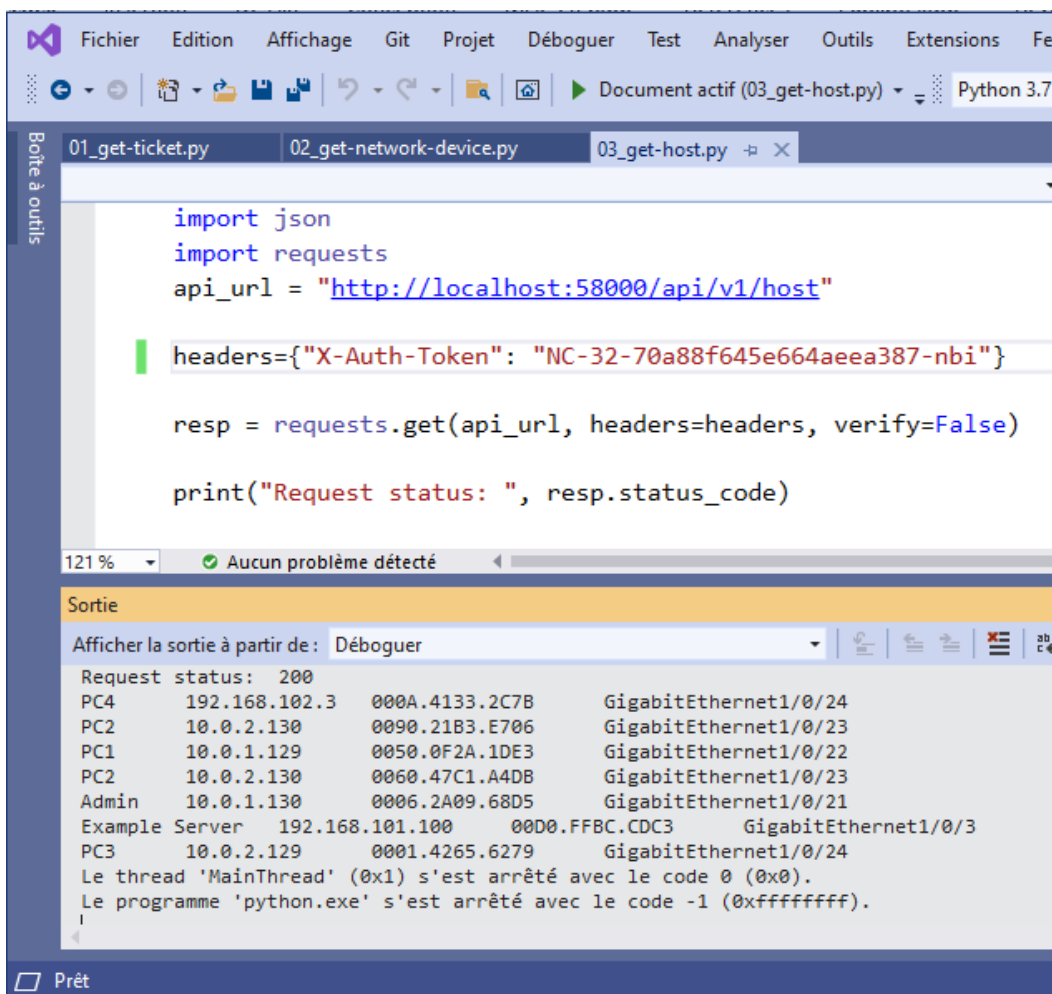
response_json = resp.json()
networkDevices = response_json["response"]

for networkDevice in networkDevices:
    print(networkDevice["hostname"], "\t", networkDevice["platfo
```

Below the code editor, the 'Sortie' (Output) window shows the execution results:

```
Request status: 200
SWL1      3650      192.168.101.2
R1   ISR4300      192.168.1.2
R3   ISR4300      192.168.2.1
SWR1      3650      10.0.1.2
SWR2      3650      10.0.1.3
R2   ISR4300      192.168.2.2
SWL2      3650      192.168.102.2
SWR4      3650      10.0.1.5
SWR3      3650      10.0.1.4
```

- Placez-vous sur 03\_get-host.py et le définir comme fichier de démarrage
- Insérez le numéro de ticket
- Cliquez sur « Démarrer » pour obtenir le résultat attendu.



Quel est l'utilité des deux scripts python précédents ? Le résultat est-il cohérent avec la demande ?

---



---



---

Est-il possible de récupérer le résultat et de le rendre exploitable ? Donnez un exemple d'exploitation de ces résultats ?

---



---

## Étape 2 : Pour aller plus loin : Envoyer des requêtes REST à l'intérieur de Packet Tracer

Dans cette étape, vous utiliserez les mêmes scripts avec une petite modification pour envoyer les mêmes requêtes d'API dans Packet Tracer que vous avez envoyées à partir de VS Code

Tâche 1 : Créez un projet dans Packet tracer

- Dans Packet tracer, cliquez sur le PC Admin.
- Cliquez sur l'onglet Programming.
- Il n'y a actuellement aucun projet. Cliquez sur New (Nouveau).
- Entrez les API REST comme Name et choisissez Empty - Python comme modèle.
- Cliquez sur Create. Le projet REST API (Python) est maintenant créé avec un script main.py vide



Tâche 2 : Modifiez les scripts à exécuter dans Packet Tracer.

L'accès d'une application à une autre sur la même machine hôte nécessite que le numéro de port soit spécifié dans l'URL. Ici Packet Tracer est en train de simuler un réseau réel. Dans le monde réel, vous utiliserez un nom de domaine ou une adresse IP dans l'URL lorsque vous effectuez des requêtes API.

- Dans VS Code, copiez le code pour 03\_get-host.py.
- Dans l'onglet Admin > Programming, double-cliquez sur le script main.py pour l'ouvrir.
- Collez le code dans le script main.py.
- Modifiez l'url api\_url. Remplacez localhost:58000/api/v1/host par **192.168.101.254/api/v1/host**.
- Les modifications sont automatiquement enregistrées. Cliquez sur **Run**. La sortie de Packet Tracer ne simule pas exactement ce que vous voyez dans la ligne de commande en réalité. Cependant, vous devriez voir une sortie similaire comme indiqué ci-dessous.

```
Starting REST APIs (Python)...
('Request status: ', 200)
('PC4', '\t', '192.168.102.3', '\t', '00E0.F96C.155B', '\t', 'GigabitEthernet1/0/24')
('PC3', '\t', '10.0.2.129', '\t', '0004.9A42.C245', '\t', 'GigabitEthernet1/0/24')
('PC1', '\t', '10.0.1.129', '\t', '00E0.A330.3359', '\t', 'GigabitEthernet1/0/22')
('PC2', '\t', '10.0.2.130', '\t', '0060.47C1.A4DB', '\t', 'GigabitEthernet1/0/23')
('Admin', '\t', '10.0.1.130', '\t', '0050.0FCE.B095', '\t', 'GigabitEthernet1/0/21')
('Example Server', '\t', '192.168.101.100', '\t', '000A.413D.D793', '\t',
'GigabitEthernet1/0/3')
REST APIs (Python) finished running.
```

- Copiez et collez 02\_get-network-device.py dans le fichier main.py. Modifiez l'URL et exécutez-la

```
REST APIs (Python) finished running.
Starting REST APIs (Python)...
('Request status: ', 200)
('SWL1', '\t', '3650', '\t', '192.168.101.2')
('R1', '\t', 'ISR4300', '\t', '192.168.1.2')
('R3', '\t', 'ISR4300', '\t', '192.168.2.1')
('SWR1', '\t', '3650', '\t', '10.0.1.2')
('SWR2', '\t', '3650', '\t', '10.0.1.3')
('R2', '\t', 'ISR4300', '\t', '192.168.2.2')
('SWL2', '\t', '3650', '\t', '192.168.102.2')
('SWR4', '\t', '3650', '\t', '10.0.1.5')
('SWR3', '\t', '3650', '\t', '10.0.1.4')
REST APIs (Python) finished running
```

### ANNEXE A : Apport théorique pour le LAB SDN

Software Defined Networking (ou SDN « réseau défini par l'application ») : tous les fournisseurs de services et de matériels réseau apportent cet élément nouveau et central aux infrastructures, le SDN.

SDN n'est pas seulement un mot qui fait le buzz. Tout comme l'arrivée du cloud il y a quelques années c'est un changement de paradigme. Progressivement, l'infrastructure réseau sera gérée bien différemment. L'interface en ligne de commande ne sera plus la méthode principale de paramétrage. Cela va également plus loin que les scripts que nous utilisons pour automatiser le déploiement. Le contrôleur SDN amène un objet central et puissant au cœur du réseau qui est accessible à distance et exploitable à l'infini avec des scripts.

Une connaissance des protocoles et des paramétrages sera toujours utile aux administrateurs réseau mais il y a tout de même fort à parier que les administrateurs de demain auront une connaissance poussée des langages de programmation pour le réseautage (comme Python) et des formats d'échanges (comme JSON ou XML).

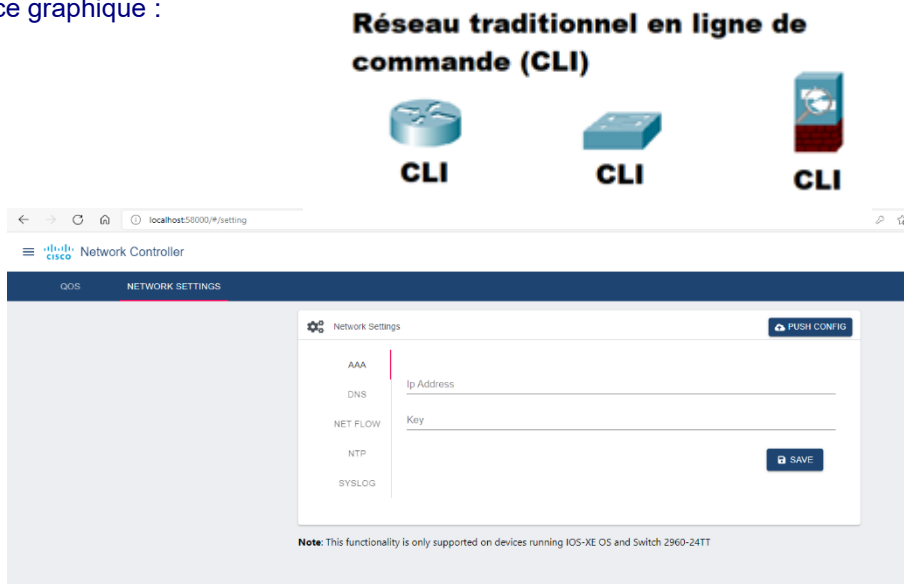
Traditionnellement, le paramétrage des matériels d'interconnexion et plus largement la mise en réseau n'était pas centralisée. Chaque périphérique réseau était paramétré individuellement en ligne de commande CLI et la communication se réalisait depuis chaque périphérique en utilisant des protocoles tels que ARP, STP, OSPF, EIGRP, etc. Les périphériques réseau communiquaient entre eux pour atteindre un état de convergence mais aucun équipement central ne disposait d'une vue d'ensemble ou ne contrôlait l'ensemble du réseau<sup>1</sup>.

Avec SDN on utilise un contrôleur central, il peut être un élément physique ou une machine virtuelle.

Les matériels d'interconnexion sont paramétrés de façon centralisée par le SDN qui possède un accès complet aux matériels et une vision globale centralisée de l'infrastructure réseau. Openflow est le protocole défini par l'ONF (Open Networking Foundation) pour transférer ces règles. Il permet par exemple à un contrôleur d'injecter des règles sur des commutateurs ou des routeurs.

Il existe deux possibilités pour paramétrer le SDN :

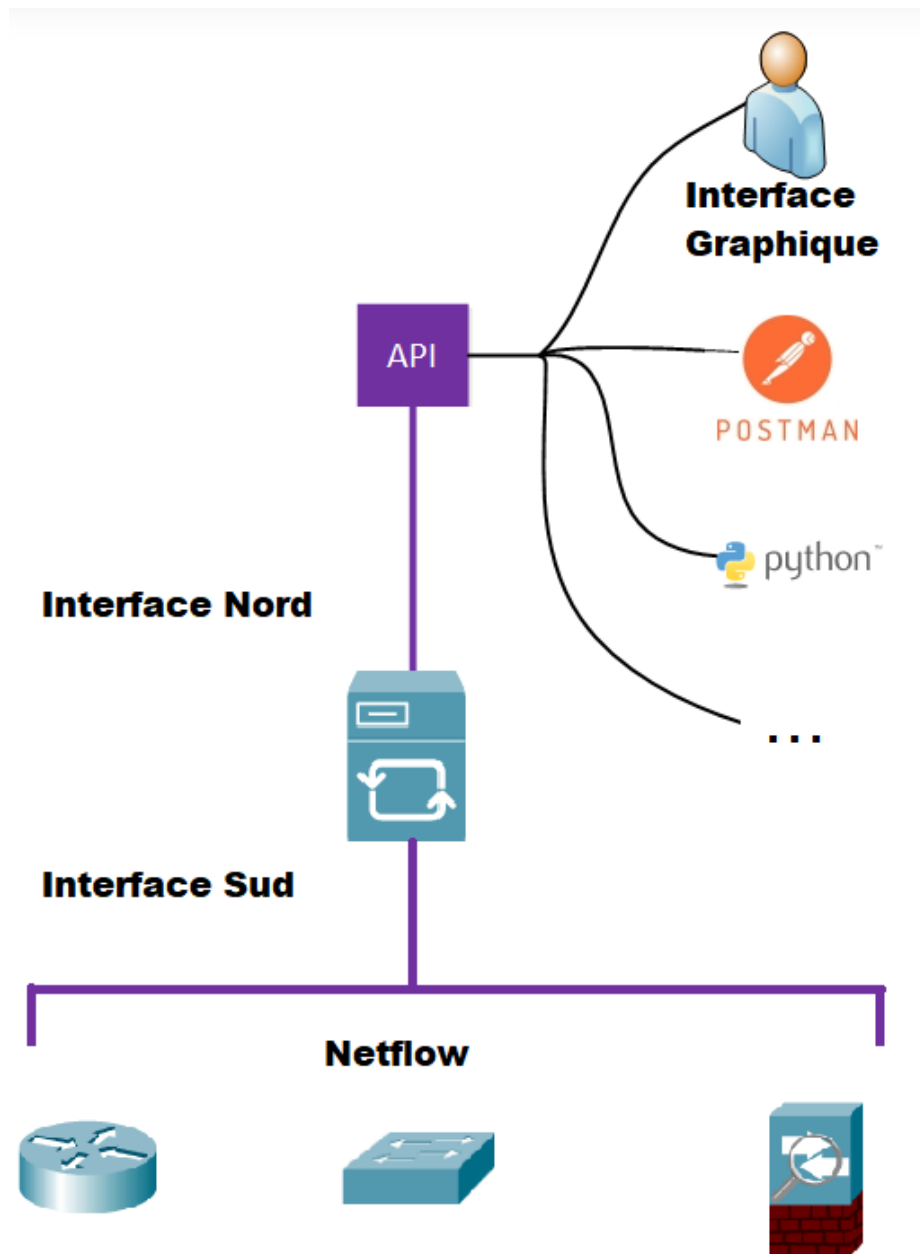
L'interface graphique :



- L'utilisation des API qui peut se faire, par exemple, par Postman<sup>2</sup> et Python :

<sup>1</sup> A l'exception des réseaux sans fil, avec les contrôleurs WLAN (WLC : Wireless Lan Controller)

<sup>2</sup> Postman est un logiciel permettant de tester les interfaces de programmation d'applications (API) directement et dans le cadre des tests d'intégration pour déterminer si elles répondent aux attentes en matière de fonctionnalités, de fiabilité, de performances et de sécurité.

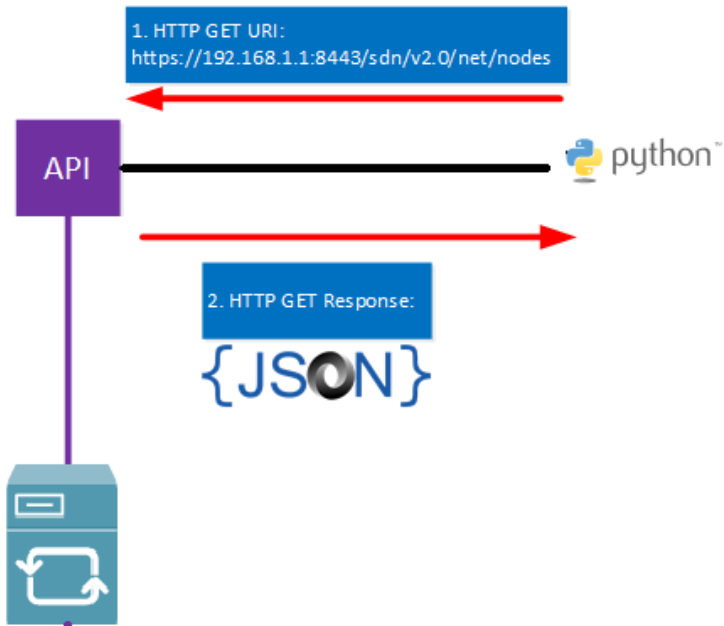


Les contrôleurs SDN utilisent généralement une **API REST (Representational State Transfer)**.

L'API REST utilise des messages HTTP pour envoyer et recevoir des informations entre le contrôleur SDN et une autre application. Il utilise les mêmes messages HTTP utilisés lorsqu'on accède à une page Web (HTTP) :

- HTTP GET : utilisé pour récupérer des informations.
- HTTP POST/PUT : utilisé pour télécharger ou mettre à jour des informations.

Avec les API REST on ne demande pas une page web mais un objet particulier du contrôleur SDN, par exemple une liste de tous les VLAN du réseau.



Lorsque le contrôleur SDN reçoit la requête HTTP GET, il répond avec une réponse HTTP GET contenant les informations demandées. Ces informations sont fournies dans un format de données commun. Les deux formats de données les plus utilisés sont :

- JSON (JavaScript Object Notation)
- XML (eXtensible Markup Language)

Voici un exemple de JSON qui est un format facile à comprendre et à exploiter tout comme XML.

```
{
  "nodes": [
    {
      "ip": "172.16.1.1",
      "mac": "fa16.3e5d.f1f4",
      "vid": 0,
      "dpid": "00:00:00:00:00:00:00:03",
      "port": 1
    }, {
      "ip": "172.16.1.2",
      "mac": "fa16.3e5d.f1f5",
      "vid": 0,
      "dpid": "00:00:00:00:00:00:00:03",
      "port": 2
    }
  ]
}
```

Les API REST ne se limitent pas à la consultation. Les suppressions et modifications peuvent également être gérées par les API.

HTTP	Equivalent SQL	Description
GET	select	Lecture d'une information
POST	insert	Écrire une information
PUT	update	Mettre à jour une information
DELETE	delete	Supprimer une information

Le travail avec les API se fait souvent à distance et l'on peut bien sûr avoir des dysfonctionnements. Dans la recherche des erreurs nous pouvons nous appuyer sur le protocole HTTP qui offre une grande variété de codes de retours. Voici les plus courants :

#### Code de retour

200 OK

201 CREATED

204 No Content

206 Partial Content

304 Not Modified

400 Bad Request

401 Unauthorized

403 Forbidden

404 Not Found

500 Internal Server Error

#### Description et utilisation

Le serveur a traité la requête avec succès.

Une nouvelle ressource a été créée.

Peut être utilisée en réponse à une requête DELETE effectuée avec succès.

En réponse à une requête demandant une réponse trop lourde pour être envoyée en une seule fois. De la pagination va être nécessaire pour récupérer l'ensemble des informations

Le client peut utiliser les données en cache car elles n'ont pas été modifiées depuis la date spécifiée.

La requête est invalide et ne peut pas être traitée par le serveur.

La requête nécessite que le client soit identifié.

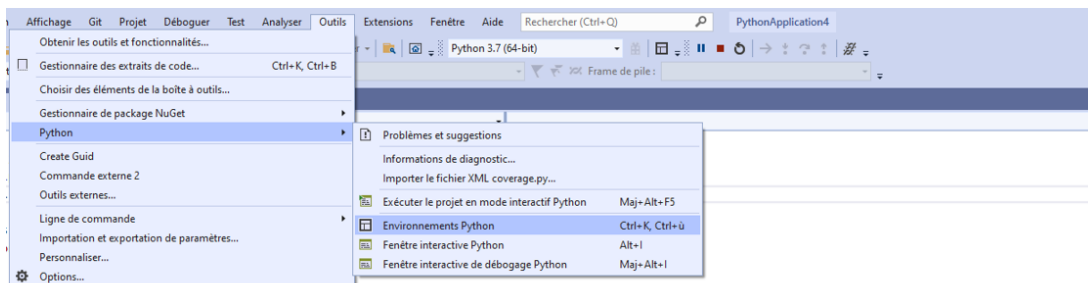
Le serveur a compris la requête mais l'utilisateur n'est pas autorisé à accéder à cette API.

La ressource demandée n'existe pas.

Votre code ne devrait jamais renvoyer cette erreur. Cette erreur devrait être récupérée par votre code et traitée, pour ensuite renvoyer une réponse adéquate au client.

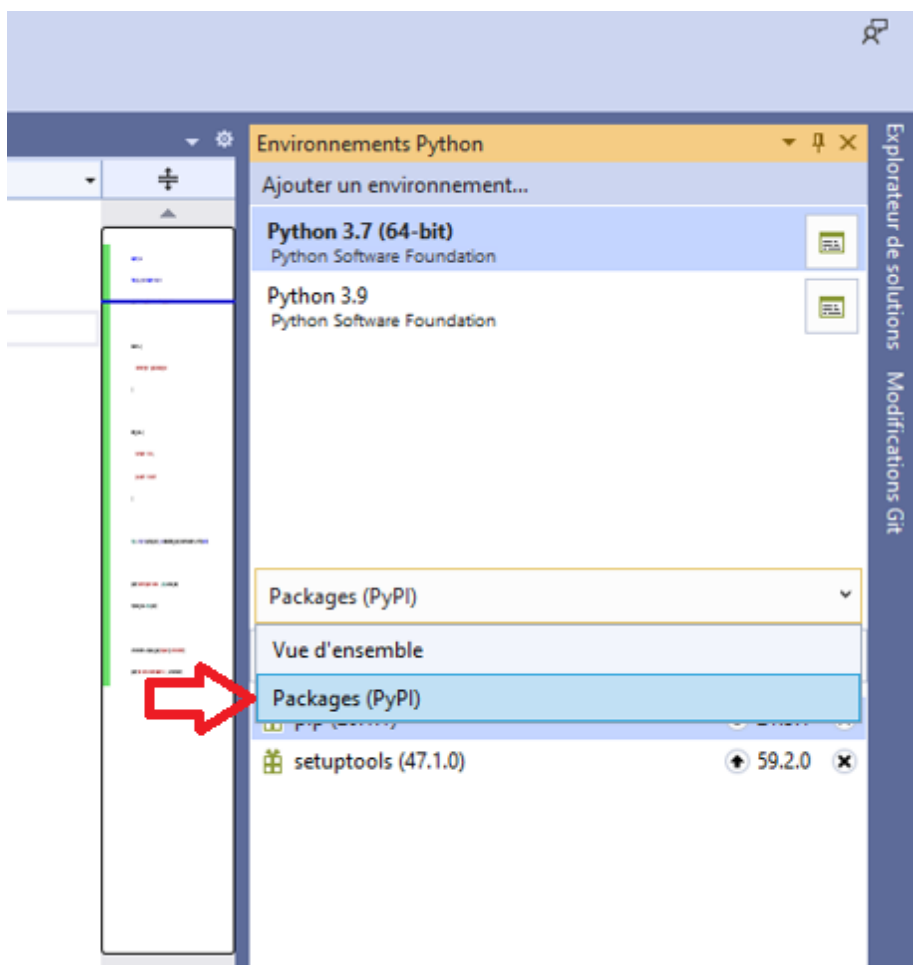
## ANNEXE B : Installation de la librairie requests sur Microsoft Visual Studio

- Rendez vous dans le menu : Outils → Environnements Python

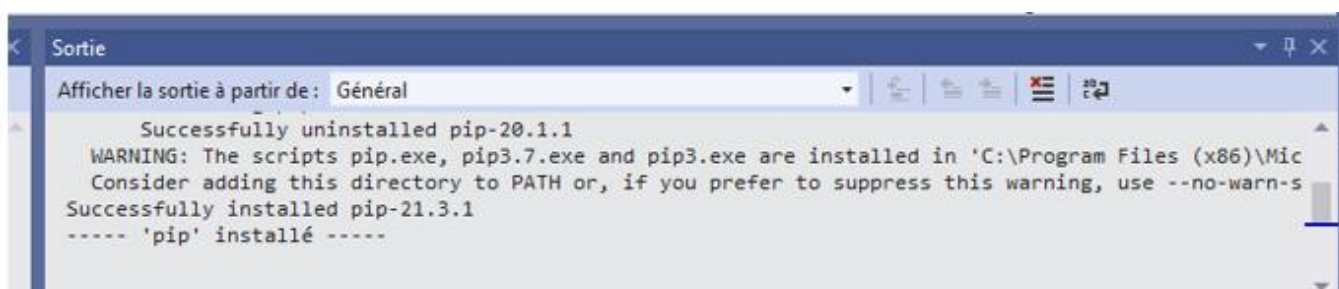
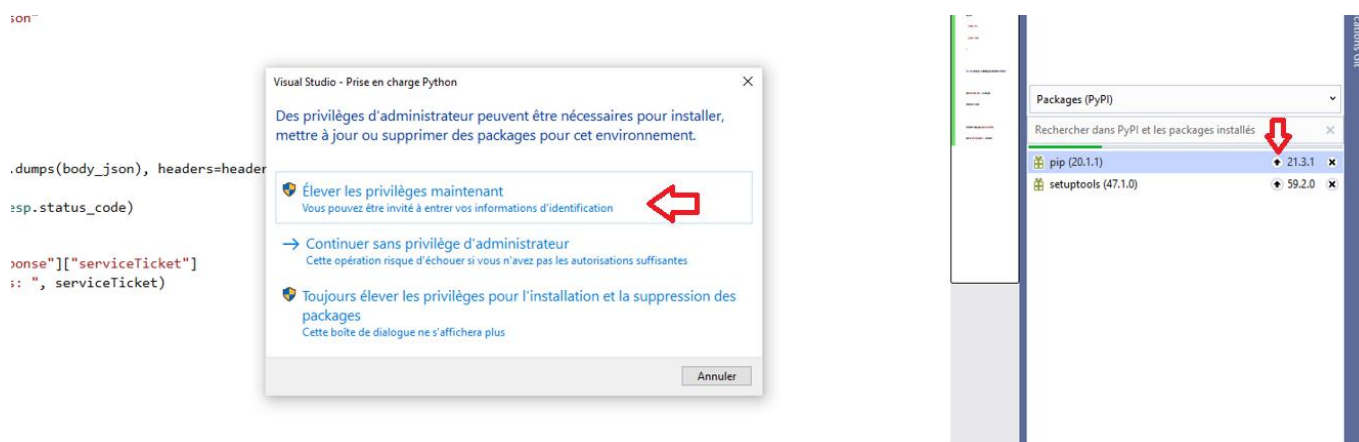


```
on = {  
    "username": "cisco",  
    "password": "cisco123!"  
}  
  
requests.post(api_url, json.dumps(body_json), headers=headers, verify=False)
```

- Sélectionnez Packages (PyPI)



- Après avoir sélectionné le petite icône de mise à niveau pip une élévation de droit vous sera certainement nécessaire.



- Recherchez la librairie requests et exécutez la commande pip install requests

