

TP 01

Automatisation Réseau et conteneurisation Docker

5ETI : Module Automatisation Réseau et
conteneurisation Docker

Amar KANTAS

Partie 1

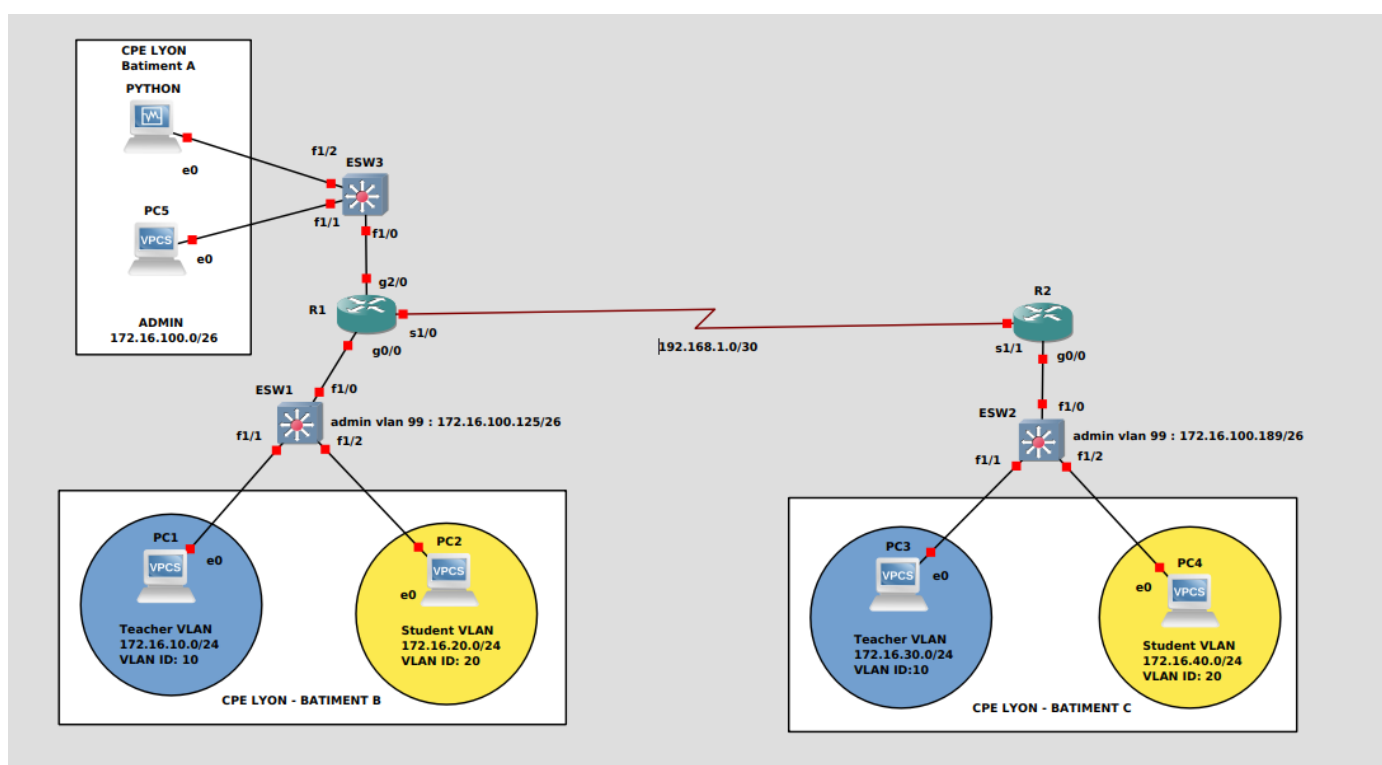
Générer des configurations réseau avec Python

1-Automatiser la création de configuration réseau pour switchs et routeurs Cisco

1.1- Objectif

Ce TP a pour objectif de développer un ensemble de scripts Python dans le but d'automatiser la création de configuration réseau pour des routeurs Cisco 7200 et des switchs de niv2/ niv3 3725 Cisco.

1.2 - Architecture réseau



1.3- Liste des équipements et de leurs adresses ip d'admin

Hostname	Ip address
PC5	172.16.100.1
PYTHON	172.16.100.2
R1	g2/0 : 172.16.100.62 g0/0.99 : 172.16.100.126
ESW1	172.16.100.125
R2	g0/0.99 : 172.16.100.190
ESW2	172.16.100.189

1.3- Prenez connaissance de l'architecture et de l'adressage réseau mis en place sur la figure 1

Vérifiez que vous pouvez vous connecter aux différents équipements de l'architecture et consultez la configuration mise en place sur les équipements réseau du bâtiment A du site CPE LYON.

Comparez avec la configuration réalisée sur le bâtiment B.

Les équipements du bâtiment B ne sont pas configurés (sauf l'admin - vlan 99) et vous allez devoir générer leur configuration depuis la machine virtuelle d'automatisation (Python).

1.4 - Connectez-vous sur la machine virtuelle d'automatisation

Utilisez le login / password suivant : cpe / 123

Vérifiez que la machine virtuelle peut joindre l'ensemble des équipements du réseau d'ADMIN. Si l'accès au réseau d'admin ne fonctionne pas, vérifiez l'état des interfaces de la machine depuis virtualbox .

Vérifiez également que la machine virtuelle PYTHON dispose d'une route vers le réseau d'admin : 172.16.100.0/24. Si ce n'est pas le cas , ajoutez la route:

```
sudo ip route add 172.16.100.0/ via 172.16.100.62 dev enp0s3
```

La machine virtuelle doit pouvoir joindre les équipements suivants :

- R1 via 172.16.100.126
- ESW1 via 172.16.100.125
- R2 via 172.16.100.190
- ESW2 via 172.16.100.189

Vérifiez que la machine virtuelle peut accéder à internet. Si l'accès à internet ne fonctionne pas, vérifiez l'état des interfaces de la machine depuis virtualbox:

- La première interface doit être connectée à GNS3 (GenericDriverI)
- La seconde interface doit être en accès par pont (Bridge)

1.5 - Environnement de développement

Prenez connaissance de l'environnement de développement en ouvrant l'éditeur de code VsCode installé dans la machine virtuelle. Ajoutez le dossier workspace/tp01 à votre éditeur si ce n'est pas déjà fait.

Activez l'environnement virtuel en vous plaçant à la racine du projet et en entrant : **pipenv shell**

Note: Vous pouvez directement utiliser le terminal depuis l'éditeur VsCode

Créez un fichier test.py et affichez simplement un helloworld sur la sortie standard. Exécutez le script pour vous assurer que tout fonctionne correctement en utilisant la commande: **python3 -m test**

En cas de problème, vous pouvez solliciter le responsable pédagogique.

1.6 -Préparation des variables pour le template JINJA

A ce stade vous avez pris connaissance de l'architecture et des éléments de configuration en place sur les équipements réseaux de l'architecture. Vous avez également vérifié le bon fonctionnement de votre environnement de développement.

- 1) Quels sont les éléments de configuration à réaliser pour que les équipements du vlan 10 et du vlan 20 du bâtiment C puissent communiquer ? (ne faites pas la configuration pour l'instant)
- 2) Parmi les éléments de configuration identifiés à la question 1, quelles sont les variables Jinja que vous avez pu identifier ?
- 3) Définissez une structure de données JSON pour recenser les variables identifiées . Créez les fichiers R2.json et ESW2.json dans le dossier data.

Faites vérifier par le responsable pédagogique

1.7 -Développer la fonction python permettant de lire les données définis dans les fichiers R2.json et ESW2.json

- 4) Créez une fonction dans le fichier scripts/ __main__.py permettant de retourner le contenu du fichier JSON passé en paramètre.

Exécutez le script en mode "module" : **python3 -m scripts.__main__**

Faites vérifier le résultat par le responsable pédagogique

- 5) Essayez d'exécuter votre script avec un chemin d'un fichier inexistant . Que se passe-t-il ? Quel est le type d'erreur qui est levé ?

- 6) Améliorez votre code pour gérer les exceptions (erreurs) dans le cas où le chemin du fichier n'est pas correct par exemple.

Faites vérifier le résultat par le responsable pédagogique

1.8 -Définition des templates Jinja pour la configuration du routeur R2 et du switch ESW2

- 7) Créez le template R2.j2 dans le dossier templates et définissez le contenu de la config et ses variables

Faites vérifier par le responsable pédagogique

- 8) Créez le template ESW2.j2 dans le dossier templates et définissez le contenu de la config et ses variables

Faites vérifier par le responsable pédagogique

1.9- Génération automatique de la configuration réseau

- 9) Initialisez les dépendances pour utiliser les fonctions du package Jinja2

pipenv install Jinja2

- a) importez les packages en début de fichier de votre script python (__main__.py)

```
from jinja2 import Template, Environment, FileSystemLoader
```

- b) Définissez le répertoire par défaut contenant vos templates Jinja

```
env = Environment(loader=FileSystemLoader("templates"))
```

- 10) Complétez la fonction `render_network_config` pour que celle-ci retourne le résultat de configuration générée pour chaque équipement (R2 / ESW2).

```
template = env.get_template(.....) #à compléter ....  
return template.render(.....) #à compléter.....
```

Faites vérifier par le responsable pédagogique

- 11) Développez une fonction `save_built_config` permettant de sauvegarder la configuration générée par la fonction de la question 10.
Les fichiers de conf doivent automatiquement être stockés dans le dossier `config` :

- `config/R1.conf`
- `config/ESW1.conf`

Faites vérifier par le responsable pédagogique

- 12) Implémentez la configuration sur les équipements et testez la communication entre les machines du vlan 10 et du vlan 20 du bâtiment C.

1.10 - Ajout d'un nouveau bâtiment D à l'architecture

- 13) Connectez un deuxième switch au routeur R2 avec deux VPCs connectés au switch.

Ne réalisez aucune configuration sur les équipements, hormis sur les VPCs .

Hostname	Ip address
ESW4	172.16.100.253 (/26) (vlan 99)
VPC-5	172.16.50.1 (/24) (vlan 10)
VPC-6	172.16.60.1 (/24) (vlan20)
R2	g2/0.10: 172.16.50.254 (/24) g2/0.20; 172.16.60.254 (/24) g2/0.99: 172.16.100.254 (/26)

- 14) Reprenez le jeux de données JSON défini dans vos fichiers R2.json et ESW2.json et créez les jeux de données en YAML dans les fichiers R2.yaml et ESW4.yaml avec les données du tableau précédent.

Faites vérifier par le responsable pédagogique

- 15) Installez les dépendances pour utiliser les fonctions du package yaml et importer le package en début de fichier de votre script

```
pipenv install PyYAML
```

```
import yaml
```

- 16) Créez une fonction permettant de retourner automatiquement le contenu des fichiers R2.yaml et ESW4.yaml

Note: Vous pouvez consulter la documentation pour savoir quelle méthode utiliser : <https://pyyaml.org/wiki/PyYAMLDocumentation>

- 17) Dans quel format sont les données de sortie de la fonction à la question 16 ?

- 18) D'après-vous est-il utile de modifier les templates Jinja2 pour que votre configuration depuis yaml soit prise en compte ? Pourquoi?

Faites vérifier par le responsable pédagogique

- 19) Sauvegardez automatiquement la configuration dans les fichiers suivant : configs/R2_from_yaml.conf et configs/ESW2_from_yaml.conf

Faites vérifier par le responsable pédagogique

- 20) Implémentez la configuration générée sur les équipements R2 et ESW4 (simple copier / coller) et testez que les communications fonctionnent entre les vlan 10 et 20 et vers les autres bâtiments.

Faites vérifier par le responsable pédagogique

Partie 2

Prise en main Docker

Le but de cette deuxième partie du TP est de se familiariser à l'utilisation de conteneurs pour le développement d'applications ou l'installation d'applications métier. L'objectif est de profiter de la puissance et la flexibilité de Docker pour déployer des applications dans les mêmes conditions d'exécution utilisées pendant la phase de développement.

2.1 - Création de l'environnement de travail Docker

- 1) Installez docker sur votre machine en suivant le guide d'installation

<https://docs.docker.com/engine/install/ubuntu/>

- 2) Par défaut seul l'utilisateur root a le droit d'exécuter docker, si vous voulez éviter d'utiliser la commande sudo à chaque appel de docker, vous devez donner les permissions pour l'exécution de conteneurs à votre utilisateur:

<https://docs.docker.com/engine/install/linux-postinstall/>

Pensez à redémarrer la VM pour que les modifications soient prises en compte

- 3) Testez que docker est bien installé en exécutant le conteneur de test "hello-world".

```
docker run hello-world
```

2.2 - Développement d'une stack PHP sous Docker

Dans cette section vous allez développer un script PHP qui se connecte à une base de données MySQL afin de récupérer les entrées de la table devices. La stack PHP, MySQL et PhpMyAdmin sera exécuté dans un environnement Docker.

- 4) Installez PHP et les librairies apache sur votre machine virtuelle

```
sudo apt update
sudo apt install php libapache2-mod-php
sudo apt-get install php-mysql
```

- 5) Vérifiez que php est correctement installé en exécutant la commande suivante : `php -version` (double tirets du 6)
- 6) Créez un dossier `web_app` dans votre workspace et ajoutez le dossier à votre environnement de développement VsCode.
- 7) Créez un fichier php nommé `index.php` et affichez un hello world dans la console de votre terminal en vous aidant des instructions suivantes:

```
<?php
    echo "hello world";
?>

php index.php
```

- 8) Maintenant que vous vous êtes assuré que votre environnement de développement PHP est fonctionnel, vous allez devoir exécuter un serveur de base de données MySQL au sein d'un conteneur docker avec un serveur web phpmyadmin hébergé dans une autre instance docker.

- a) Créez votre premier conteneur MySQL en vous aidant de la documentation du repos officielle docker MySQL https://hub.docker.com/_/mysql et des indications suivantes:

```
Nom du container: cpe_lyon_mysql
Port d'écoute: 3306
Mode: detached
Mot de passe: cpelyon123
```

Vérifiez à l'aide de la commande (à compléter)
docker container
que votre conteneur à bien été créé.

- b) Arrêtez votre container en vous appuyant sur la commande (à compléter) **docker stop**
- c) Relancer votre container à l'aide de la commande (à compléter) **docker container start**
- d) Créez votre deuxième container PhpMyAdmin en vous aidant de la documentation du repos officielle docker PhpMyAdmin <https://hub.docker.com/r/phpmyadmin/phpmyadmin/> et des indications suivantes:

Nom du container: cpe_lyon_phpmyadmin

Port d'écoute: 8080

Mode: detached

Link: linkez votre instance à votre container MySQL

- e) Si vous avez correctement exécuter vos commandes docker vous devriez pouvoir vous connecter à phpmyadmin à l'adresse suivante: <http://localhost:8080>. Connectez-vous à l'interface à l'aide du login root et du mot de passe configuré précédemment.
- f) Créez une base données portant le nom app_db et créez une table de 4 colonnes en respectant les indications suivantes:
- Nom de la table: devices
 - Nombre de colonne : 4
 - device_id de type int (activé l'auto-incrément et définissez ce champ en tant que PK)
 - device_name de type varchar (limite 255)
 - device_ip de type varchar (limite 255)
 - device_vendor de type varchar (limite 255)
- g) Insérez 4 enregistrements dans votre table devices. Enregistrez les routeurs et switches du bâtiment B (partie 1) et du bâtiment C de votre réseau définis à la partie 1.

```
INSERT INTO `devices` (`device_id`, `device_name`, `device_ip`, `device_vendor`) VALUES
(NULL, 'R1', 'g2/0 : 172.16.100.62\r\nng0/0.99 : 172.16.100.126\r\n', 'Cisco'), (NULL,
'ESW1', '172.16.100.125\r\n\r\n\r\n', 'Cisco');
```

- 9) Ajoutez le code suivant à votre script php index.php et complétez le afin que celui-ci soit en mesure de se connecter à votre serveur de base de données MySQL (Docker) et affichez la liste des devices enregistrés dans la table devices précédemment créée.

```
<?php
function initMysqlConnection(){
    $pdo = new PDO('mysql:host=XXXXX;dbname=XXXX', "USERNAME", "PASSWORD");
    return $pdo;
}

function getDevices($pdo){
    $query = $pdo->query("SQL QUERY TO BE COMPLETED.....");
    return $query;
}

$pdo = initMysqlConnection();
$devices = getDevices($pdo);

foreach ($devices as $row) {
    echo "-----\n";
    echo "device name : ".."<br/>\n";
    echo "device ip : ".."<br/>\n";
    echo "device vendor : ".."<br/>\n";
}
?>
```

Faites vérifier par le responsable pédagogique la bonne exécution du script

- 10) Nous allons à présent faire en sorte que notre script index.php soit exécuté dans un conteneur Docker et nous connecter au serveur web afin d'afficher la liste des devices directement depuis votre navigateur web.
- a) Nous allons créer notre propre image docker, pour cela il vous suffit de créer le fichier Dockerfile dans le dossier web_app.
 - b) Copiez / collez et complétez les commandes suivantes dans votre DockerFile

```

FROM php:7-apache

# Installer des logiciel à l'aide d'apt-get
RUN apt-get update && apt-get install -y libpng-dev curl libcurl4-openssl-dev

# Ajoute l'extension pdo_mysql avec les outils docker
RUN docker-php-ext-install pdo_mysql gd

# Activer le module rewrite dans apache
RUN a2enmod rewrite
RUN service apache2 restart

# Donner un nom au serveur
RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf

# Pour intégrer le dossier de votre site web directement dans l'image
COPY .....A COMPLÉTER..... /var/www/html

```

c) Buildez votre image à l'aide de la commande suivante :

`docker build -t cpe_lyon_webapp .`

- d) Vérifiez que votre image a bien été générée et enregistré sur votre machine
- e) Le serveur apache écoute par défaut le port 80, exécuter votre container en utilisant l'image générée à la question c et faites en sorte que votre container écoute sur le port 8081.
- f) Rendez-vous à l'adresse localhost:8081, que se passe-t-il ? Comment expliquer ce résultat ?

Faites vérifier votre réponse par le responsable pédagogique

- g) Mettez en application votre réponse à la question F dans votre environnement Docker et faites le nécessaire pour avoir le résultat attendu. (Il existe plusieurs solutions).