

# Cisco Packet Tracer

Li` œ de` I Ne ièN` c" a de` à effecq` eN" N de la  
 i` e e N} œ d'} ` £ iqph " } d'} N} œ} N

> : Enable	Permet de passer en mode privilège
(config) : hostname "nomdésiré"	Permet de changer le nom du <b>routeur</b> , switch
(config) : username "nomdésiré" password "motdepasse"	Permet de créer un utilisateur avec un mot de passe (Utile pour la configuration du <b>ssh</b> )
(config) : service password encryption	Permet de crypter tout les mots de passe qui de base apparaissent en clair avec la commande ( show run )
(config) : ip domain-name "nomdésiré"	Permet de choisir un nom de domaine ( Utile pour la configuration du <b>ssh</b> )
(config) : interface "fa "ou "gi" 0/...	Permet de se rendre dans l'interface souhaité
(config) : interface range "fa "ou "gi" 0/... - ...	Permet de se rendre dans <b>LES</b> interfaces souhaités
(config-if) ip address 192.168..... 255.255.....	Permet d'attribuer un IP à l'interface dans laquelle on se situe ( ne pas oublier le masque )

## Comme` Vla ( ` £ iqph )

Les **VLAN** permettent de séparer logiquement des départements ou des groupes de travail sans pour autant qu'ils soient séparés physiquement. Ainsi on pourra avoir le département comptabilité sur un **Vlan**, le département commercial sur un autre et de même pour le département production et le département direction. Elles optimisent aussi la sécurité d'un réseaux et permettent la création de sous réseau dans un réseau.

Conf t	Pour se rendre en mode configuration
(config) : vlan "numéroduvlan"	Création d'un vlan avec le numéro souhaité
(config-vlan) : name "nom"	Ici nous allons nommer notre vlan par exemple "Administration)
(config-vlan) : exit	Pour sortir de la configuration du vlan
(config) : interface "fa "ou "gi" 0/...	Permet de se rendre dans l'interface souhaité ( Pour aller dans plusieurs interface d'un coup ne pas oublier "range")
(config-if) : switchport mode access	Permet d'activer le mode access sur le ou les ports dans lesquelles on se trouve
(config-if) : switchport access vlan "..."	Permet d'attribuer le ou les ports au vlan souhaiter
(config-if) : no shutdown	Permet d'allumer les ports

## Acq•eNe ``h•eNi" 2

Activer le ssh version 2 va permettre une administration du switch ou du routeur à distance ce qui peut s'avérer très pratique dans certain cas

<pre>router&gt; enable router# configure terminal router(config)# username julien password pa\$\$word</pre>	<p>Ici on définit un utilisateur nommé "julien" dont le mot de passe associé est "pa\$\$word"</p>
-------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------

```
router(config)# hostname R1
R1(config)#
```

Cette commande permet de mettre un hostname particulier à votre équipement, ici "R1". Ce hostname est par ailleurs utilisé pour générer la clé de chiffrement RSA, créée plus bas

```
R1(config)# ip domain-name cisco.com
```

Cette commande permet de définir un nom de domaine (DNS – Domain Name Server) à votre **routeur**. Il s'appellera désormais "R1.cisco.com", qui est tout simplement la concaténation du hostname "R1" avec le nom de domaine "cisco.com".

Ce nom de domaine est aussi utilisé pour générer la clé de chiffrement RSA, créée juste à l'étape suivante.

```
R1(config)# crypto key generate rsa modulus 1024
```

Cette commande génère une clé de chiffrement RSA utilisé par le processus SSH pour générer la clé de session. La variable "**modulus 1024**" définit la taille de votre clé. A titre d'information, pour une clé asymétrique, 1024 est une taille correcte.

```
R1(config)# line vty 0 4
R1(config-line)# transport input ssh
R1(config-line)# login local
R1(config-line)# exit
```

Pour accéder en telnet ou ssh à un équipement Cisco, il faut configurer les lignes "virtuelles" de l'équipement. En effet, quand je fais un telnet vers un **routeur**, je peux **arriver** de n'importe quelles interfaces actives du **routeur**.

Pour cela, on définit les lignes virtuelles appelées "vty". Par défaut, il y a 5 lignes vty actives (de 0 à 4). D'où la commande "*line vty 0 4*". Pourquoi 5? Parceque

Plus sérieusement, ça permet à 5 administrateurs d'être

en SSH en même temps sur l'équipement.

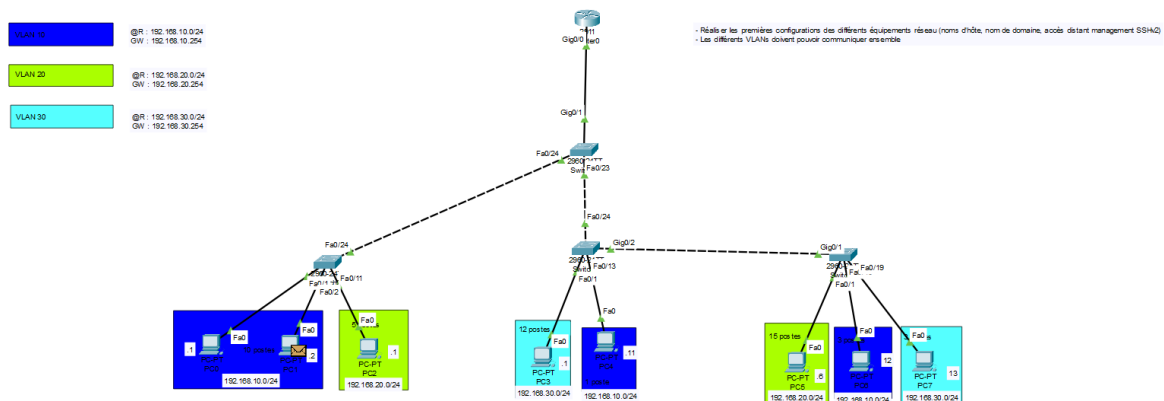
La commande `"transport input ssh"` définit quel protocole a le droit d'utiliser ces lignes vty. Par défaut, tous les protocoles ont le droit dont telnet et ssh. Cette commande permet de restreindre en précisant que seul ssh a le droit d'utiliser les lignes vty.

La commande `"login local"` permet de préciser où se trouve la base des comptes utilisateur. Une fois connecté au routeur en ssh, le routeur va vous demander un login/password, celui qu'on a défini plus haut (julin/pa\$\$word). `"login local"` indique au routeur que la base des comptes utilisateur se trouve dans sa configuration ("local").

Pour information, on peut définir une base de comptes utilisateur vers un serveur Radius par exemple.

Eq de d' } e-ence NegN } I a qlé " q" ` de TN } k,  
e cal ` } laq" ` , c" } icaq" i qNMa .

( parler du fait qu'il faille bien créer les vlan sur tout les switch sinon la connection ne se fera pas )



Dans cet exercice nous avons 4 switch 1 routeur et 9 pc à configurer.

1. Pour commencer je configure manuellement l'ip, le masque ainsi que la passerelle de chaque pc.

Indiquer une passerelle sur un pc, c'est lui dire " Si je te demande de communiquer avec une machine qui ne fait pas partie du réseau, tu dois aller parler à cette personne ( la passerelle ) pour pouvoir communiquer avec la machine voulue.

C'est pour cela que généralement la passerelle sera l'adresse de l'interface du **routeur** car c'est lui qui fait la connexion entre les réseaux. 😊

## 2. J'effectue ensuite les configurations basiques sur mes **switch**

```
> : Enable
```

```
(config) : hostname "nomdésiré"
```

```
(config) : enable password "password"
```

```
(config) : username "nomdésiré"  
password "motdepasse"
```

```
(config) : service password encryption
```

```
(config) : ip domain-name "nomdésiré"
```

## 3. J'active ensuite le **ssh** sur mes **switch** afin d'avoir un accès à distance sur eux

```
router> enable  
router# configure terminal  
router(config)# username julien password pa$$word
```

```
router(config)# hostname R1  
R1(config)#
```

```
R1(config)# ip domain-name cisco.com
```

```
R1(config)# crypto key generate rsa modulus 1024
```

```
R1(config)# line vty 0 4  
R1(config-line)# transport input ssh  
R1(config-line)# login local  
R1(config-line)# exit
```

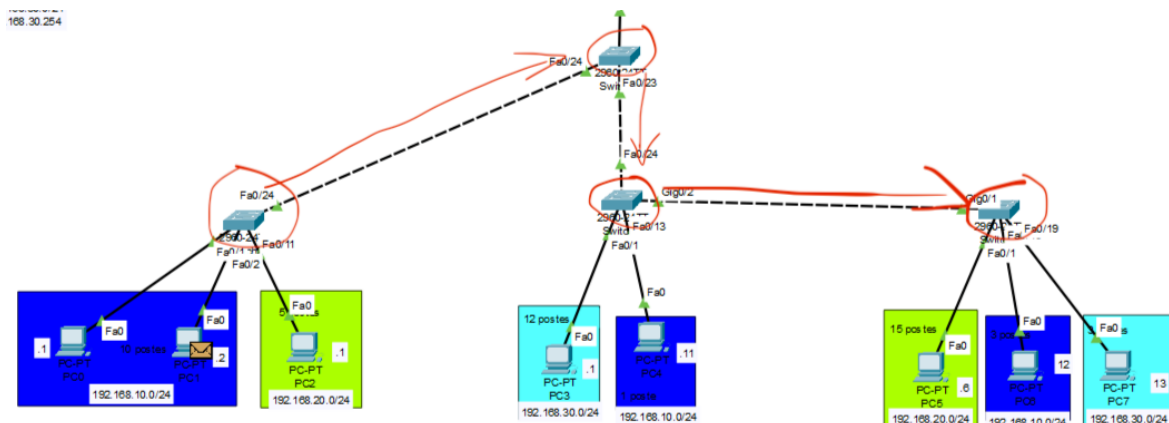
```
R1(config)# ip ssh version 2
```

4. Je crée ensuite mes différents **VLAN** sans oublier d'y mettre les interfaces voulus.

Conf t
(config) : vlan "numéroduvlan"
(config-vlan) : name "nom"
(config-vlan) : exit
(config) : interface "fa "ou "gi" 0/...
(config-if) : switchport mode access
(config-if) : switchport access vlan "..."
(config-if) : no shutdown

**Attentions les VLANS doivent être créés sur TOUT les switch. En effet si le ping doit passer par un switch où la vlan n'a pas été créée (Vraiment juste créé histoire de dire qu'elle existe) le message ne pourra pas arriver à destination car le switch ne saura pas où l'envoyer.**

Voici un exemple de chemin qu'un ping devrait emprunter pour aller d'un pc à un autre.



On voit bien ici que les **vlan**s doivent être absolument toutes ( 10, 20 et 30 ) configurer sur les **switch** pour permettre au ping de traverser le réseau, sinon le ping sera stopper sur un des **switch** où la **vlan** n'a pas été configurée.

## 5. Maintenant que mes VLANS ont été créés et annoncer sur tout les switch, je peux passer au trunk

Da base toutes les interfaces se situe dans le vlan1, le vlan natif.

Ce qui veut dire que si j'ai d'autre vlan , ici la 10,20,30, je devrait brancher un cable sur chaque interface voulus, ce qui est une perte de temps et d'argents.

Intervient alors le mode trunk.

Le trunk va agir comme une sorte de tunnel ou toutes mes vlans pourrons passer dans un seul cable.

Ainsi je dois configurer pour mon switch 1 et 2 je dois entrer dans les 2 interfaces 24 de ceux-ci et y mettre la commande

```
(config-interface) : swichport mode trunk
```

A ce stade mon réseau bleu peut communiquer avec les autres bleu même s'ils se trouvent sur un autre switch.

Par contre la communication bleu vers vert est encore impossible car c'est le routeur qui se charge de la connection ENTRE les réseaux

Voici un exemple de communication inter-vlan du même réseau, on voit que le message ne passe pas par le routeur

 Enregistrement de l'écran 2022-02... 8 MB

Par contre si l'on veut ping un autre réseau cette fois-ci le message va obligatoirement passer par le routeur !

## 6. Passons au routage

Nous voulons maintenant pouvoir effectuer une communication inter-vlan.

C'est a dire que la VLAN 10 puisse communiquer avec la VLAN 20 ou 30.

Le ping que l'on va envoyer va donc devoir changer de réseau, et c'est le routeur qui se charge de cela.

Comme l'adresse IP que l'on veut ping se situe dans un autre réseau, notre paquet va devoir passer par le routeur.

Mais comment faire cela ? Et bien c'est grâce à la passerelle en effet lorsque que l'on veut ping une machine qui n'est pas dans notre réseau et qui est donc



inconnu de notre ordinateur, le paquet est directement envoyé à l'adresse ip rentré dans "passerelle".

Seulement la passerelle doit se trouver dans le même réseau que le pc duquel on fait le ping, et comme nous avons 3 vlans nous avons donc 3 passerelles.

192.168.10.254 192.168.20.254 192.168.30.254

Le problème étant que notre routeur ne possède qu'une interface physique, la question est donc, comment pouvons nous mettre 3 ip ( qui vont servir de passerelle à chaque pc ) sur une seule interface ?

La réponse est simple nous allons créer des sous interfaces qui seront uniquement logiques et non physiques.



Pour ce faire nous allons utiliser les commandes ci dessous :

```
R1 #: conf t
```

```
R1 (config): interface g0/0. (numéro du vlan  
généralement)
```

```
R1(config-interface): encapsulation dot1q ( numéro du  
vlan )
```

```
R1(config-interface): ip address "192.168...."
```

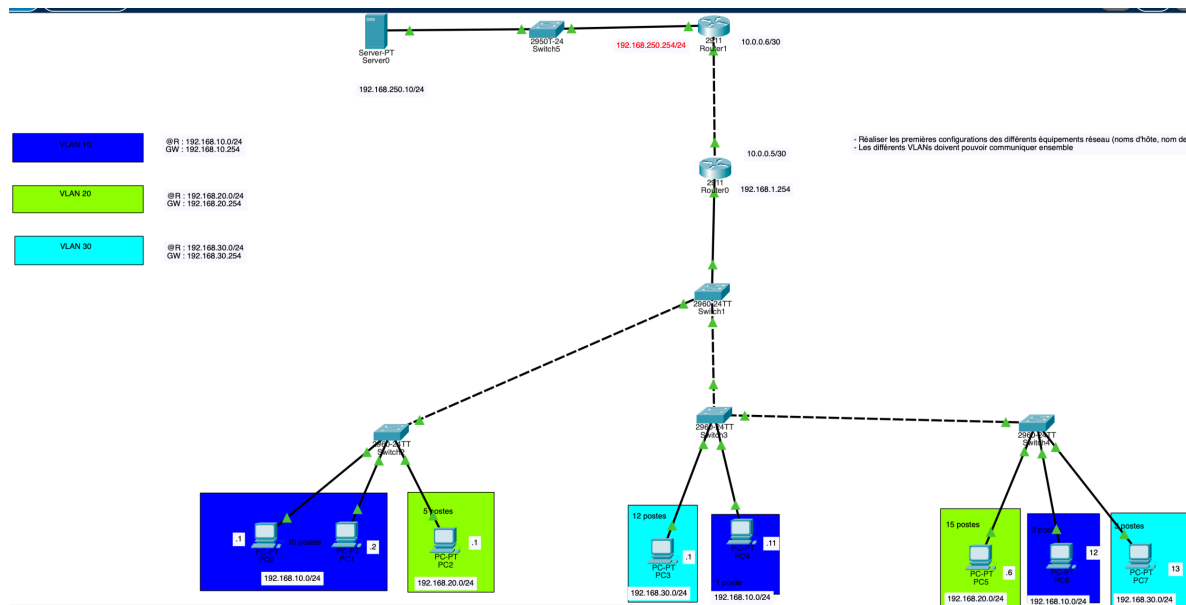
```
R1(config-interface): no shutdown
```

Nous allons donc faire cela pour chacune de nos vlans afin de les connecter au routeur.

Les vlans pourront ensuite communiquer entre elles.

## Exercice 2 :

Voici la suite de cette exercice en image



Le but maintenant va être de faire communiquer les **vlan 10** et **20** avec le serveur. Pour commencer, il faut que notre **routeur 0** puisse communiquer avec le serveur, pour cela il doit connaître la route à emprunter.

C'est le routage.

Pour ce faire nous nous rendons dans le **routeur 0** et nous entrons les commandes ci-dessous :

```
RTR-0 (config) : ip route 192.168.250.0 255.255.255.0 10.0.0.6
```

**Le réseau que je souhaite découvrir**

**Son masque**

**L'adresse de l'interface dans laquelle je vais entrer**

Et voila mon message peut donc parvenir jusqu'au serveur, cependant il doit aussi pouvoir faire le chemin inverse.

Pour cela j'effectue les configurations mais cette fois ci depuis l'autre **routeur**

```
RTR-1 (config) : ip route 192.168.1.0 255.255.255.0 10.0.0.5
```