



TUNNEL SSH

Linux CentOS

Table des matières

1	Tunnel SSH.....	3
1.1	SSH Local	3
1.2	SSH Distant.....	5
1.3	Exemple pratique	7

1 Tunnel SSH

Un tunnel SSH implique les quatre nœuds suivants :

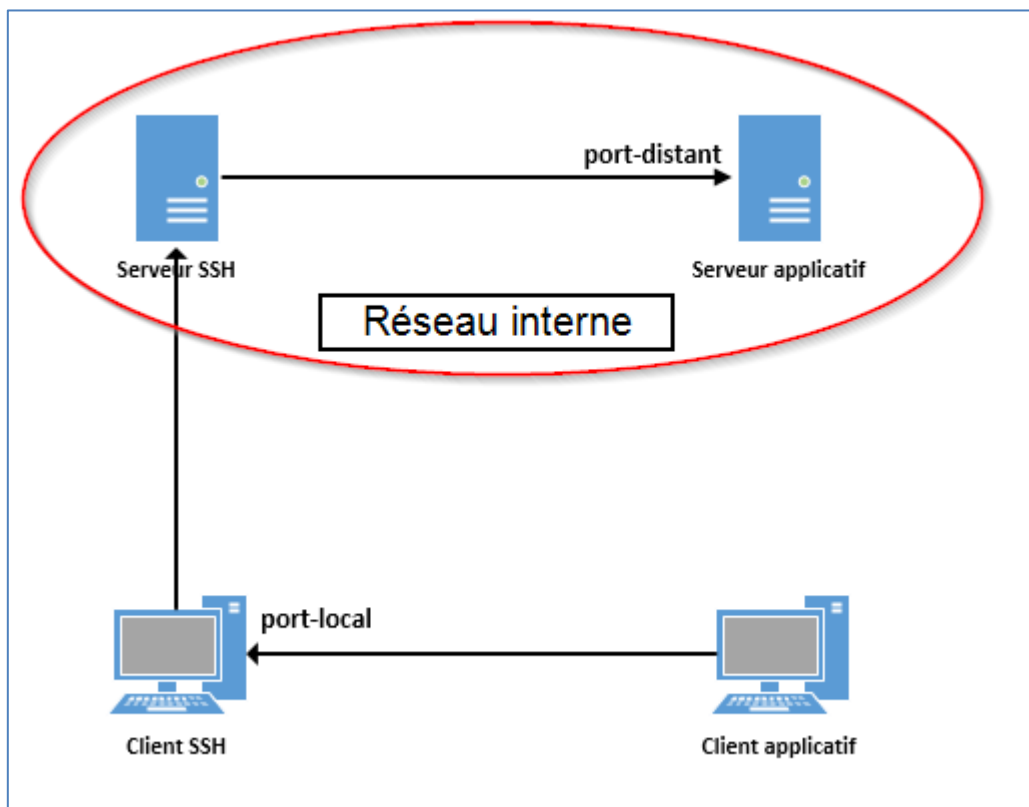
- Un client SSH
- Un serveur SSH
- Un client applicatif
- Un serveur applicatif

Configurer un tunnel **SSH** est un moyen simple de chiffrer n'importe quelle communication TCP entre un client applicatif et un serveur applicatif.

Un Tunnel peut être configuré en **mode local** ou en **mode distant**. On va décrire le fonctionnement de chaque mode.

1.1 SSH Local

Le mode local nous permet d'**accéder à un serveur sur un réseau interne (privé)**.



La syntaxe est la syntaxe:

```
client_ssh# ssh -L port-local:serveur_applicatif:port-distant serveur_ssh
```

Le SSH local permet de transmettre les demandes locales sur le port **port-local** de **client_ssh** vers le **serveur_ssh** qui les envoie sur le port **port-distant** de **serveur_applicatif**.

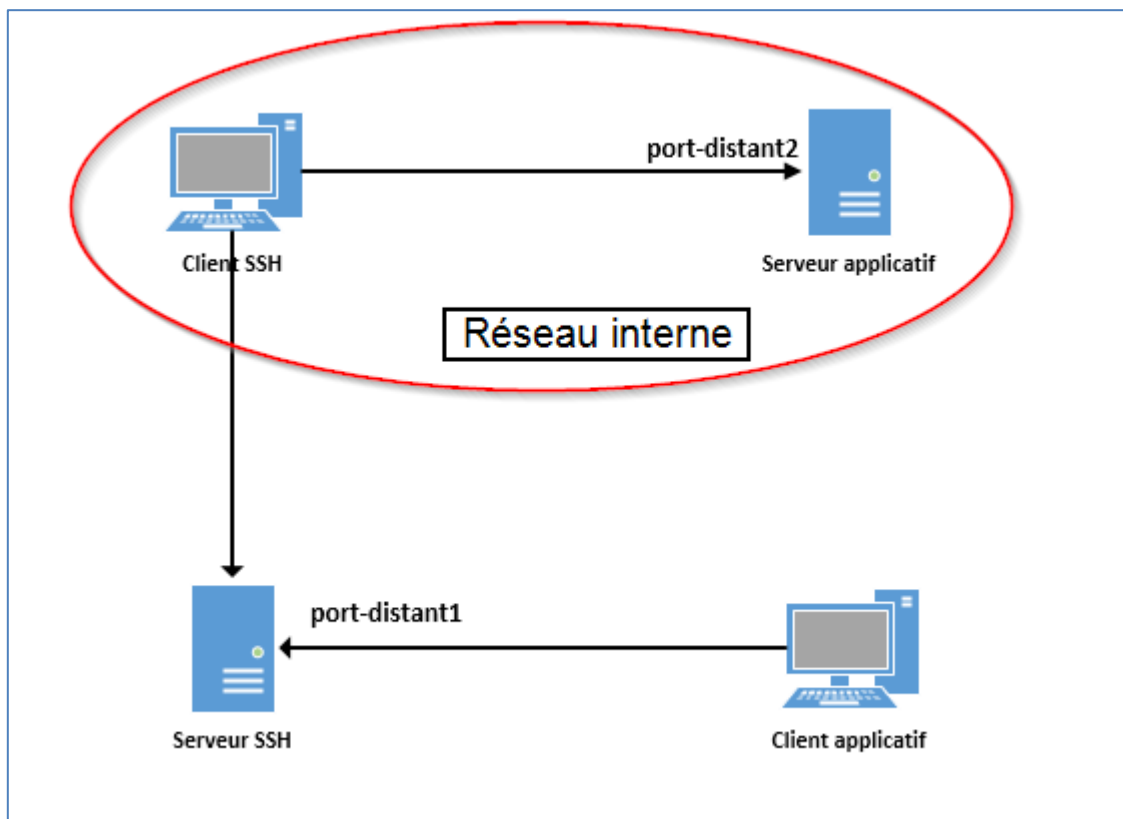
Remarque

Par défaut c'est le port **port-local** de l'interface local (127.0.0.1) qui est utilisé. Pour utiliser une autre interface il suffit de préciser l'adresse de l'interface avant le port **port-local** et les séparer par :

```
ssh -L adresse_ip:port-local:serveur_applicatif:port-distant serveur_ssh
```

1.2 SSH Distant

Le mode distant nous permet d'**ouvrir l'accès à un serveur sur un réseau privé**.



La syntaxe est la suivante:

```
client_ssh# ssh -R port-distant1:serveur_applicatif:port-distant2 serveur_ssh
```

Le SSH distant permet de transmettre les demandes distantes sur le port **port-distant1** du **serveur-ssh** vers port **port-distant2** du **serveur-applicatif**.

Remarque

Par défaut c'est le port **port-distant1** de l'interface local (127.0.0.1) qui est utilisé.
Pour pouvoir utiliser une autre interface il faut suivre les étapes suivantes :

- 1) Mettre le paramètre x à yes
- 2) Redémarrer le service SSH
- 3) Préciser l'adresse de l'interface avant le port **port-distant1** et les séparer par :

```
ssh -R adresse:port-distant1:serveur_applicatif:port-distant2 serveur_ssh
```

Pour utiliser toutes les interfaces, il faut préciser * :

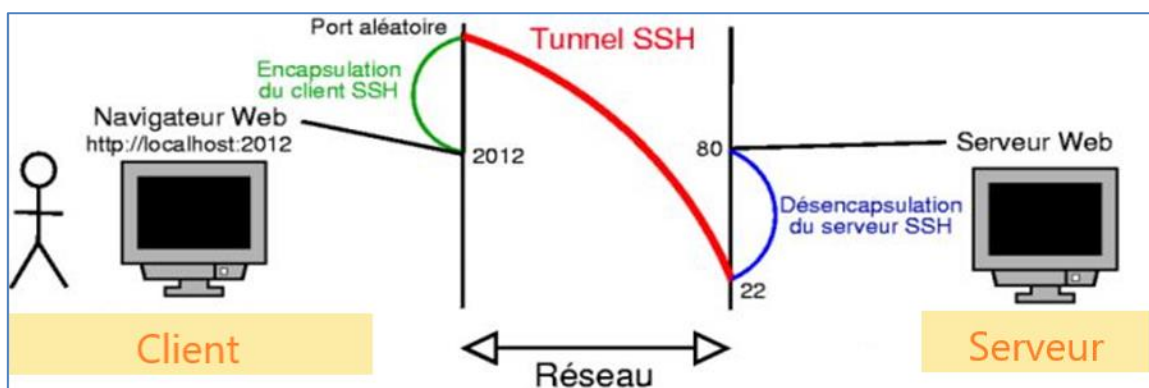
```
ssh -R *:port-distant1:serveur_applicatif:port-distant2 serveur_ssh
```

1.3 Exemple pratique

On va configurer un tunnel en mode local pour permettre à un client web d'accéder à un serveur web (faire transiter le flux HTTP dans le tunnel).

Adresse IP du client web : **192.168.1.102**

Adresse IP du serveur web : **192.168.1.103**



En premier, on va installer et démarrer le service **apache** sur le **serveur web**:

```
[root@serveurweb ~]# yum install httpd
```

```
[root@serveurweb ~]# systemctl start httpd
```

Ensuite, établir un tunnel SSH pour une connexion HTTP vers le serveur web à partir du client web:

```
[root@clientweb ~]#  
ssh -L 2012:192.168.1.103:80 root@192.168.1.102
```

2012 est le numéro de port sur la machine cliente à partir duquel la connexion entre dans le tunnel **SSH** (le port doit être supérieur à **1024** si on ne veut pas avoir à lancer le tunnel en tant que **root**).

Il suffit de lancer un navigateur Web en lui demandant de se connecter en local sur ce port : <http://localhost:2012>

```
ssh -L 2012:serveur:80 usager@serveur
```

Si on ferme la session SSH, alors le tunnel est fermé.

```
exit
```