Se connecter à un VPS avec SSH

Table des matières

I - SSH : Secure Shell 3

II - Exercice : Appliquer la notion 6

III - Exercice : Agir sur un serveur comme à la maison 7

IV - Exercice : Échanger des fichiers avec un serveur 8

V - Exercice : [exercice optionnel] Se connecter à un serveur avec une clé 9

VI - Exercice : [exercice optionnel] Donner accès au serveur à des tiers 11

Solutions des exercices 12

Glossaire 15

Crédits des ressources 16

Contenus annexes 17

I SSH : Secure Shell

Objectif

- Se familiariser avec le protocole SSH (Secure SHell).

Mise en situation

SSH est une application d'Internet qui n'est pas aussi connue que le Web ou le mail.

Mais elle est très utile dans le monde du développement informatique car elle permet de se connecter à un serveur à distance, puis d'exécuter des commandes sur ce serveur. C'est aujourd'hui le mode de contrôle privilégié des informaticiens sur les serveurs qui font fonctionner Internet.

Terminal et shell

Rappel

Le **terminal** (aussi appelé invite de commande ou console) est le moyen le plus naturel pour interagir avec un ordinateur. Ce programme offre un **shell** (une interface système) donnant accès aux programmes de l'ordinateur. Il suffit d'un écran et d'un clavier pour utiliser ce shell.

- *Terminal et shell* (cf. p.17)

Le programme SSH (Secure SHell)

Az Définition

Le programme SSH (*Secure Shell*) permet d'interagir de manière sécurisée avec un ordinateur distant via un shell. C'est le programme de référence pour effectuer des opérations à distance.

- Toutes les commandes tapées depuis un clavier d'ordinateur à un emplacement A sont exécutées dans le shell d'un ordinateur à un emplacement B.
- Les informations qui transitent sur Internet via SSH sont les chaînes de caractères représentant les commandes à exécuter et les chaînes de caractères représentant les résultats de ces exécutions.

Le protocole SSH

Az Définition

SSH est aussi le nom du protocole de communication utilisé par le programme SSH. Ce protocole fonctionne sur une **architecture client-serveur** ; l'ordinateur qui fournit les commandes est le **client** et l'ordinateur qui exécute les commandes sur son système est le **serveur**.

SSH repose généralement sur TCP pour le transport et est par défaut associé au port **22**.

L'établissement d'une connexion SSH se fait en deux étapes :

- l'établissement d'une communication sécurisée,
- l'authentification du client.

Établir une connexion SSH

Méthode

On utilise la commande `ssh` depuis un terminal :

```
ssh alice@adresse.ip
```

La machine se trouvant à l'adresse `serveur.exemple.com` doit avoir un utilisateur `alice`.

Sur Windows, Putty est un client SSH qui permet d'ouvrir une console sous SSH. Il est téléchargeable sur le site dédié : https://www.putty.org/

Attention

Si `ssh` se connecte pour la première fois au serveur, l'utilisateur devra accepter d'utiliser la clé publique du serveur pour des raisons de sécurité.

Remarque

Si un serveur est associé à un nom de domaine, on peut utiliser ce domaine à la place de l'IP.

```
ssh alice@serveur.exemple.com
```

Chiffrement symétrique et chiffrement asymétrique

Complément

Le **chiffrement symétrique** repose sur le chiffrement et le déchiffrement de messages par une seule clé appelée **clé symétrique**. Ainsi, si deux ordinateurs veulent protéger leurs communications, ils peuvent se mettre d'accord sur une clé symétrique à utiliser et veiller à ce qu'ils soient les seuls à la détenir. Le chiffrement symétrique est rapide mais nécessite un canal sécurisé pour échanger la clé.

Le **chiffrement asymétrique** repose sur le chiffrement et le déchiffrement de messages par deux clés différentes (l'une chiffre les messages et l'autre déchiffre). Dans ce système, une clé est connue de tous : on parle de **clé publique** ; l'autre clé est gardée secrètement : on parle de **clé privée**. Le chiffrement asymétrique est plus lent mais ne nécessite par de canal sécurisé préalable.

Étape 1 : Établissement d'une communication sécurisée

Complément

Au début de la session et avant de commencer toute interaction entre le client et le serveur, c'est-à-dire avant même que le serveur vérifie que le client est légitime, les deux ordinateurs doivent se mettre d'accord sur un moyen de sécuriser leur communication.

Pour cela, le client et le serveur se mettent d'accord via le protocole de transport (TCP) sur la version du protocole SSH à utiliser et sur une méthode de **chiffrement symétrique** de leur communication.

Les deux ordinateurs vont utiliser des méthodes de chiffrement asymétrique pour échanger une clé symétrique. Cette clé leur permettra donc de protéger toutes les communications suivantes de la session.

À la fin de cette étape, une connexion sécurisée est établie entre le client et le serveur.

Étape 2 : Authentification de l'utilisateur côté client

Complément

Une fois une connexion sécurisée établie, le serveur doit **authentifier** l'utilisateur qui cherche à se connecter. Il existe deux méthodes d'authentification :

- **Méthode 1 : Par mot de passe**

  Puisque le but est ici d'ouvrir un shell à distance, le client cherche à se connecter à un utilisateur de la machine distante. Le serveur peut ainsi demander au client le mot de passe de l'utilisateur via lequel il veut se connecter.

- **Méthode 2 : Par clé**

  Chaque utilisateur présent sur le serveur possède une liste de clés publiques de clients de confiance, présentes dans un fichier `authorized_keys`.

  1. Le client envoie une des clés publiques de l'utilisateur avec lequel il veut se connecter.
  2. Le serveur vérifie le fichier `authorized_keys` de l'utilisateur pour s'assurer que cette clé publique existe.
  3. Si oui, le serveur chiffre un nombre aléatoire avec ladite clé publique et envoie le résultat au client.
  4. Le client utilise sa clé privée pour déchiffrer le nombre et y appliquer un certain traitement avant de le renvoyer au serveur.
  5. Le serveur applique le même traitement au nombre qu'il a envoyé et vérifie que le résultat du client est le même que le sien.
  6. Si le résultat est correct, et comme l'utilisateur est le seul à posséder la clé privée et déchiffrer le nombre envoyé par le serveur, il est authentifié.

Complément

La méthode d'authentification par clé nécessite qu'une paire de clés publique/privée soit générée au préalable, et que le serveur ait stocké la clé publique du client dans le fichier `authorized_keys` de l'utilisateur.

Installer un serveur SSH sur un serveur

Complément

Pour rendre un serveur accessible à distance, il faudra installer et lancer un serveur SSH. L'implémentation de référence est OpenSSH que l'on peut installer et activer ainsi sur des serveurs GNU/Linux de la famille Debian.

```
sudo apt update
sudo apt install openssh-server
sudo systemctl start ssh
```

À retenir

- Le protocole SSH permet à un client d'ouvrir un shell sur un serveur distant.
- Ce protocole sécurise les communications en employant plusieurs méthodes de chiffrement, de l'authentification au transfert de données.

II Exercice : Appliquer la notion

Pré-requis

Disposer d'un VPS chez un hébergeur qui fournit un accès SSH à un serveur.

VPS : serveur dédié virtuel *(cf. p.20)*

Question 1

[solution n°1 p. 12]

Se connecter à ce serveur en utilisant une authentification par mot de passe.

Indice :

Il faut utiliser la commande `ssh` depuis un shell local.

Indice :

Une connexion `ssh` nécessite :

- L'adresse IP du serveur,
- Le nom d'un utilisateur,
- Le mot de passe de l'utilisateur.

Question 2

[solution n°2 p. 12]

Lors de la première connexion, pourquoi le terminal a-t-il demandé si vous faisiez confiance au serveur ?

Question 3

[solution n°3 p. 12]

Vérifier que la clé SSH du serveur a bien été ajoutée aux hôtes connus (`known_hosts`).

La commande suivante permet de vérifier qu'un hôte est connu sur votre client SSH :

```
1 ssh-keygen -F ip_serveur
```

La commande renvoie des informations sur le serveur, la méthode de chiffrement utilisée ainsi que la clé publique.

III Exercice : Agir sur un serveur comme à la maison

Pour réaliser cet exercice, connectez-vous à un VPS en SSH en tant que *root*.

Testez ensuite chacun des commande suivantes.

- Regarder le nom de votre machine : `hostname`
- Regarder qui vous êtes : `whoami`
- Regarder où on est : `pwd`
- Regarder quel est le système installé : `lsb_release -a`
- Aller dans le dossier */tmp* : `cd /tmp`
- Créer un fichier avec votre lieu de naissance : `nano doujeviens`
- Regarder ses processus : `top`
- Regarder l'état de son disque : `df -h`
- Regarder qui est connecté au serveur : `who`
- Regarder l'état du service (*daemon*) SSH : `systemctl status ssh`
- Regarder les logs du service SSH :

  ```
  journalctl -u ssh
  journalctl -n 20 -u ssh
  ```
- Trouver votre IP : `curl https://ifconfig.me && echo`

IV Exercice : Échanger des fichiers avec un serveur

Pour réaliser cet exercice vous devez être connecté à votre VPS.

On appellera `sara` le *user* connecté.

Question 1

[solution n°4 p. 13]

Créez un dossier `shared` dans le répertoire `/home/sara`.

Indice :

`sara` doit être un *user* autorisé à se connecté en SSH au serveur.

Question 2

[solution n°5 p. 13]

Copiez un fichier contenant le nom de votre artiste préféré depuis votre PC vers votre VPS (dans le dossier `shared`) : `rsync -v tmp.txt sara@51.15.235.148:/sara/shared`

Question 3

[solution n°6 p. 13]

Créer un dossier partagé avec SFTP et l'explorateur de fichier : `sftp://sara@51.15.235.148/home/sara/shared`.

Ajoutez une liste d'œuvres de votre artiste préféré.

Question 4

[solution n°7 p. 13]

Créer un espace d'échange avec SFTP et Filezilla

- installer Filezilla
- configurer une connexion SFTP vers votre VPS
- déposer une image de votre artiste préféré dans `shared`

Question 5

[solution n°8 p. 13]

Copiez le contenu de votre dossier `shared` dans un dossier accessible à un serveur web sur votre VPS.

V Exercice : [exercice optionnel] Se connecter à un serveur avec une clé

L'objectif est :

1. de générer une paire de clé (publique et privée)
2. puis de transférer la clé publique au serveur
3. et de se connecter en utilisant l'authentification par clé

Question 1

[solution n°9 p. 13]

Questions préliminaires :

- pourquoi transfère-t-on la clé publique ?
- pourquoi ne transfère-t-on pas la clé privée ?

Indice :

La clé publique sert à chiffrer les donner et la clé privée sert à déchiffrer les données.

Indice :

Comprendre le chiffrement par clé : *Gépégix (cf. Gépégix)*

Pour générer une paire de clé il suffit de taper la commande suivante :

```
1 ssh-keygen
```

Il vous sera demandé une *passphrase* (mot de passe) qui chiffrera la clé sur votre machine. Ce n'est pas obligatoire mais il s'agit d'une sécurité supplémentaire au cas où quelqu'un d'autre que vous utiliserait votre machine.

Votre paire de clé sera enregistrée dans le dossier *caché* p.15 `.ssh` accessible depuis votre *home*.

Question 2

[solution n°10 p. 13]

Utilisez le terminal pour vous rendre dans le répertoire `ssh` de votre machine et affichez la paire de clés qui s'y trouve.

Question 3

[solution n°11 p. 13]

En utilisant la commande `cat ~/.ssh/id_rsa.pub`affichez le contenu de votre clé publique.

Question 4

[solution n°12 p. 13]

L'objectif est à présent de transférez votre clé publique SSH sur votre serveur afin de s'y connecter sans mot de passe.

On suppose ici le *user* concerné est `admin`.

Configurer l'accès SSH :

- se connecter en tant que admin : `su admin`
- créer le dossier caché `/home/admin/.ssh`
- créer le fichier `.ssh/authorized_keys`

- copier la clé publique dans le fichier `.ssh/authorized_keys`
- ajuster les droits :
    - 700 pour le dossier `.ssh`
    - 644 pour la clé publique et le fichier `authorized_keys`

Indice :

Vous pouvez désormais vous connecter sans mot de passe. La *passphrase* de votre clé vous sera demandée à la place.

Question 5

[solution n°13 p. 13]

NB : le dossier .ssh et le fichier authorized_keys doivent appartenir à jack (si ce n'est pas le cas on utilise chown)

VI Exercice : [exercice optionnel] Donner accès au serveur à des tiers

L'objectif est de donner accès à votre serveur à une autre personne en ajoutant sa clé sur votre serveur.

On appellera cet utilisateur *jack* dans le cadre de cet exercice.

Question 1

[solution n°14 p. 13]

Créer un nouvel utilisateur sur le serveur : `adduser jack`

Question 2

[solution n°15 p. 14]

Configurer l'accès SSH pour Jack

- se connecter en tant que jack : `su jack`
- créer le dossier `/home/jack/.ssh`
- créer le fichier `.ssh/authorized_keys`
- ajouter la clé à `.ssh/authorized_keys`
- ajuster les droits :
    - 700 pour le dossier `.ssh`
    - 644 pour la clé publique et le fichier `authorized_keys`

NB : le dossier .ssh et le fichier authorized_keys doivent appartenir à jack (si ce n'est pas le cas on utilise chown)

Solutions des exercices

Solution n°1

[exercice p. 6]

Exemple

L'hébergeur a fourni une adresse IPv6 et a créé un utilisateur `exemple`.

```
1 ssh exemple@2001:4b99:1:1:216:3eff:fe90:63f
```

```
└─▪ ssh exemple@2001:4b99:1:1:216:3eff:fe90:63f
exemple@2001:4b99:1:1:216:3eff:fe90:63f's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-22-generic x86_64)

[----------------------------------------]
  Gandi - Welcome to your new OS image.

  Documentation :
   [EN] http://wiki.gandi.net/en/iaas
   [FR] http://wiki.gandi.net/fr/iaas


  Configuration file for Gandi :
   /etc/default/gandi or
   /etc/sysconfig/gandi
[----------------------------------------]


Last login: Mon Apr 27 14:48:51 2020 from 2a01:e35:2427:ced0:2c33:f293:fdf5:b1a1
exemple@ssh-exemple:~$ mkdir test
exemple@ssh-exemple:~$ ls
test
exemple@ssh-exemple:~$ _
```

Solution n°2

[exercice p. 6]

La clé SSH d'un serveur est unique et sert à la fois à l'identifier et à chiffrer les communications.

Le client `ssh` stocke les clés publiques de tous les serveurs auxquels il s'est déjà connecté dans le fichier `known_hosts`.

1. Lors la première connexion à un serveur qui n'est pas encore connu, `ssh` s'assure que vous savez qu'il s'agit d'un nouveau serveur.
2. Si ce message apparaît et que ce n'est pas la première connexion, alors :
   - soit le serveur a changé de clé publique,
   - soit il s'est fait usurper son identité.

Solution n°3

[exercice p. 6]

Exemple

```
1 ssh-keygen -F 194.187.168.100
```

Solution n°4

[exercice p. 8]

Solution n°5

[exercice p. 8]

Solution n°6

[exercice p. 8]

Solution n°7

[exercice p. 8]

Solution n°8

[exercice p. 8]

Solution n°9

[exercice p. 9]

- On communique notre clé publique au serveur afin que le serveur chiffre les données qu'il va nous envoyer avec notre clé publique. Les données seront déchiffrables uniquement grâce à la clé privée allant de paire avec la clé publique.
- Si la clé privée est communiquée sur le réseau ou stockée sur le serveur elle risque d'être récupérée. Les données envoyées par le serveur pourront donc être déchiffrées par des tiers. Il est donc primordial de ne jamais communiquer sa clé privée. Elle ne sert qu'à déchiffrer les données que l'on vous envoie (chiffrées avec votre clé publique).

Solution n°10

[exercice p. 9]

Il vous faut taper la commande suivante :

```
#commandes
ls ~/.ssh
```

```
#résultat
id_rsa  id_rsa.pub  known_hosts
```

id_rsa est votre clé privée, *id_rsa.pub* votre clé publique et *known_hosts* contient la listes serveurs auquel vous vous êtes déjà connecté.

Solution n°11

[exercice p. 9]

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDU6aCD3LJM7KtMPT3xyn8vpuT3X8JrCWCnhD3jaTRRQmN4ZH8ofmqRh3/Czcp9c
stc@hal9017

```

Solution n°12

[exercice p. 9]

Solution n°13

[exercice p. 10]

Solution n°14

[exercice p. 11]

Solution n°15

[exercice p. 11]

Glossaire

Fichier caché

Sous Linux les fichiers et dossiers cachés sont matérialisés par un point devant leur nom.

Exemple : *.ssh*

Les fichiers et dossiers cachés sont visibles via la commande *ls -a* (a pour *all*) et sont accessibles par leur chemins comme les dossiers normaux. Exemple de chemin : *~/.ssh*

Crédits des ressources

p. 12

Licence : Domaine Public

Contenus annexes

1. À la découverte du terminal

Objectif

- Savoir ouvrir un terminal et exécuter une commande.

Mise en situation

On a l'habitude d'interagir avec les ordinateurs en mode graphique, c'est-à-dire en utilisant essentiellement la souris pour cliquer sur des éléments d'interface pour effectuer des actions. Mais il existe une autre manière d'interagir avec un ordinateur, très utilisée en informatique : le mode texte ou **interface en ligne de commande** (CLI en anglais).

Ce mode d'interaction est très utile pour pouvoir **utiliser un ordinateur à distance**, ce qui est généralement le cas lorsque l'on souhaite administrer un serveur web.

À travers ce module vous allez découvrir le **shell** qui permet de dialoguer avec le système d'exploitation d'une machine. Il existe beaucoup de shells, le plus connu vient du monde Linux et se nomme bash. Vous allez découvrir les commandes de base pour parcourir des répertoires ou éditer des fichiers.

Terminal ou CLI — Az Définition

Un **terminal**, ou **interface en ligne de commande** (*CLI* en anglais), est une interface homme-machine dans laquelle l'utilisateur interagit avec la machine en mode **texte**. L'utilisateur écrit des **lignes de commande**, la machine les exécute et affiche le résultat des commandes.

Le terme terminal est très général : un terminal peut servir à dialoguer avec un programme informatique, à donner des ordres à un ordinateur, ou à exécuter d'autres programmes.

Shell — Az Définition

Un **shell** est une interface en ligne de commande permettant de dialoguer avec le système d'exploitation de la machine. On dit qu'il **interprète** les commandes.

Langages CLI — Exemple

Il existe au moins autant de shells que de systèmes d'exploitation. Chaque shell propose des commandes spécifiques. Le choix d'un shell se fait surtout par rapport à des critères pratiques (de quoi ai-je besoin ?) et des critères subjectifs (quel shell me semble le plus ergonomique ?).

- Sur Linux, le shell le plus connu et installé par défaut sur la plupart des systèmes s'appelle `bash` (pour "Bourne Again Shell").
- Sur Windows, il existe trois shells :
  - `cmd`, le shell historique,
  - `powershell`, une version plus moderne,
  - et il est possible d'utiliser `bash` avec le sous-système Linux.
- Sur macOS, le shell installé par défaut s'appelle `zsh`. Il partage de grandes similarités avec `bash`.

Shells et Windows

Remarque

Le shell historique `cmd` de Windows n'est presque plus utilisé. `powershell` utilise une syntaxe assez différente de la plupart des autres shells. Il est possible d'installer `bash` sous Windows depuis peu, mais il n'est pas inclus par défaut dans les installations.

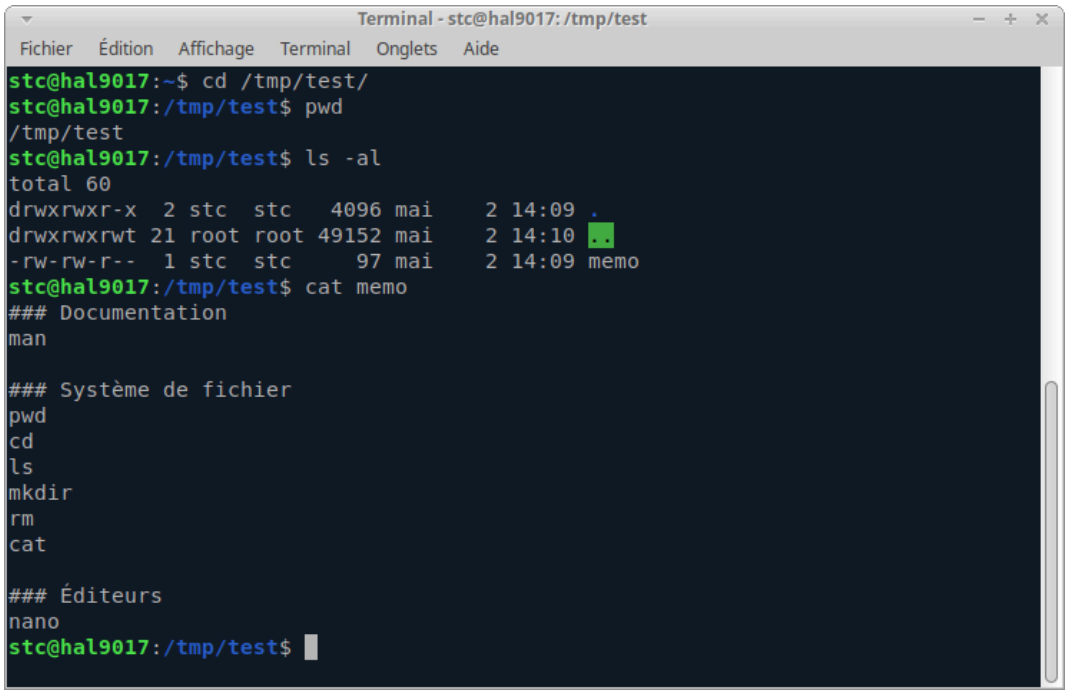
Bash et Windows

Conseil

La plupart des shells adhèrent à des standards communs, mais ce n'est pas le cas des shells disponibles sur Windows. Pour suivre les exemples et exercices qui suivent, il est recommandé **d'activer le sous-système Linux pour Windows 10** et d'utiliser `bash`, en suivant la partie "Installer le sous-système Windows pour Linux" de cette documentation : docs.microsoft.com/fr-fr/windows/wsl/install-win10

Commandes et Bash

Exemple



Cette image montre quelques commandes de base, exécutées par un shell `bash` sur un système Linux.

- `cd /tmp/test` se rend dans le dossier `test` qui se trouve dans le dossier racine `/tmp`.
- pwd affiche le dossier où on est situé dans le système de fichier.
- `ls -al` affiche la liste des fichiers dans le dossier courant.
- `cat memo` affiche le contenu du fichier `memo`.

Terminal et environnement graphique

Remarque

- Un ordinateur personnel moderne PC dispose d'une interface graphique et d'une interface terminal : les deux permettent d'effectuer à peu près les mêmes opérations : visualiser des fichiers, les supprimer, ouvrir des applications, etc.
- Un serveur n'offre en général qu'une possibilité d'accès à distance via un terminal. C'est une des raisons pour lesquelles savoir utiliser le terminal est utile.

Ouvrir un terminal sous Linux

Méthode

En général (et en particulier sur les systèmes Ubuntu), le raccourci `Ctrl+Alt+T` ouvre un terminal. Une alternative consiste à chercher `Terminal` dans la liste des applications.

Ouvrir un terminal sous Windows 10

Méthode

Ouvrir la fenêtre `Exécuter` à l'aide du raccourci `Super`+R (la touche `Super` est en général représentée par un logo Windows sur le clavier). Entrer :

- `cmd` ou `powershell` dans la fenêtre qui s'est ouverte pour démarrer un terminal Windows
- `bash` pour ouvrir un shell Bash s'il a été installé

Ouvrir un terminal sous macOS

Méthode

Depuis le Launchpad, chercher `Terminal` et cliquer sur l'icône qui s'affiche.

Quelques commandes de base

Syntaxe

Ces commandes sont des commandes Bash qui fonctionnent également sous macOS et avec la plupart des autres shells Linux.

- `ls -al` : lister les fichiers dans le répertoire courant
- pwd : afficher le répertoire courant
- `cat` *`fichier`* : afficher le contenu d'un fichier
- `cd` *`dossier`* : se rendre dans un dossier fils du dossier courant
- `cd ..` : se rendre dans le dossier parent
- `echo` *`message`* : afficher un message

- `man` *`commande`* : afficher la documentation détaillée d'une commande

Copier/coller dans un terminal Méthode

Les actions « copier » et « coller » sont accessibles en effectuant un clic droit sur la fenêtre d'un terminal. Il est plus rapide d'utiliser des raccourcis clavier pour copier et coller, respectivement :

- Sur Bash et la plupart des shells Linux :`Ctrl+Shift+C` et `Ctrl+Shift+V`
- Sur macOS : `Command+C` et `Command+V`
- Sur Windows (`powershell`) : `Ctrl+C` et `Ctrl+V`

À retenir

- Un shell, ou par abus de langage un terminal, permet de dialoguer avec le système d'exploitation d'une machine.
- Il existe beaucoup de shells. Le plus connu vient du monde Linux et se nomme `bash`.
- Un shell permet de se passer du mode graphique, ce qui est souvent indispensable pour travailler sur une machine à distance.

2. VPS : serveur dédié virtuel

Objectifs

- Savoir ce qu'est un VPS
- Savoir créer un VPS chez un hébergeur
- Savoir se connecter à distance sur un VPS avec SSH

Serveur Rappel

Un **serveur** est un ordinateur accessible depuis Internet, qui rend des **services** aux utilisateurs.

Il se distingue des **ordinateurs personnels** que l'on ne peut pas contacter directement aussi simplement depuis Internet.

Utilisation quotidienne des serveurs Exemple

- Lorsque je me rends sur le site wikipedia.org[1], je demande en réalité aux **serveurs** de Wikipédia de m'envoyer le contenu de la page que je veux afficher.
- Un ami ne peut pas accéder aux fichiers de mon ordinateur personnel : pour les partager, je dois les téléverser sur un **serveur** (envoyer un mail, utiliser un service partage de fichiers, etc.).

1. Wikipédia - https://fr.wikipedia.org

Remarque

Tout ordinateur personnel peut être transformé temporairement en serveur, mais on ne traite pas ce cas ici.

VPS

Az Définition

Un VPS (serveur dédié virtuel, ou *Virtual Private Server*) peut s'envisager comme un serveur réservé à son usage personnel. En réalité, il s'agit d'une partie d'un serveur physique isolée du reste du système : un serveur **virtuel**.

À quoi sert un VPS ?

Exemple

Un VPS peut servir :

- à mettre en ligne un site web : serveur Apache, Nginx.
- à travailler à plusieurs sur une même machine : partage de fichier avec SFTP, serveur NextCloud, serveur GitLab.
- à tester et installer d'autres applications web : Etherpad, Mattermost.
- à tester et installer des applications d'Internet : mail.

Créer un VPS chez un hébergeur

Méthode

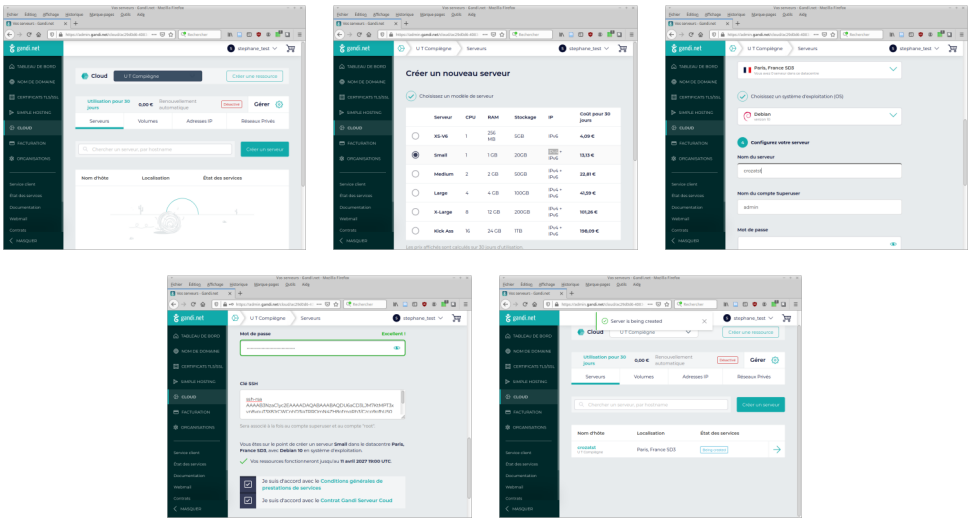
Il existe plusieurs **hébergeurs** professionnels qui proposent la location de VPS, on retrouvera en général les étapes suivantes :

1. Se rendre sur le site de l'hébergeur (exemple : gandi.net[2])
2. Choisir une offre (à noter que pour disposer d'un serveur réellement accessible sur Internet par tout le monde, il faut que le VPS soit doté d'une adresse IPv4)
3. Choisir le système d'exploitation souhaité, sa version (par exemple : Debian 10)
4. Choisir un nom pour identifier le VPS, créer un mot de passe **robuste** pour le compte administrateur et éventuellement associer une clé SSH

2. https://www.gandi.net/fr/cloud

Créer un VPS chez Gandi

Exemple



Créer un VPS chez Gandi

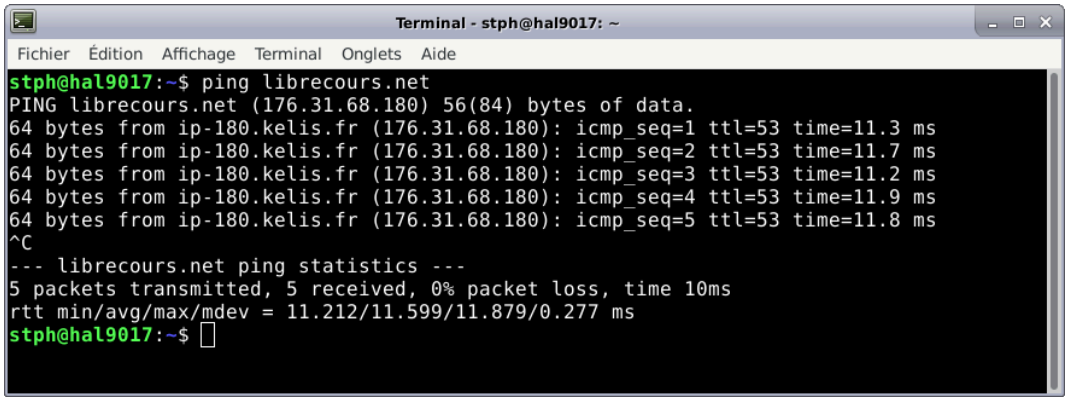
Remarque

Le VPS est contactable par son **adresse IP**, qui est unique sur Internet et est l'équivalent d'une adresse postale.

On peut utiliser la commande `ping` pour vérifier qu'un serveur répond bien.

Ping

Exemple



Accéder à un VPS avec SSH

Méthode

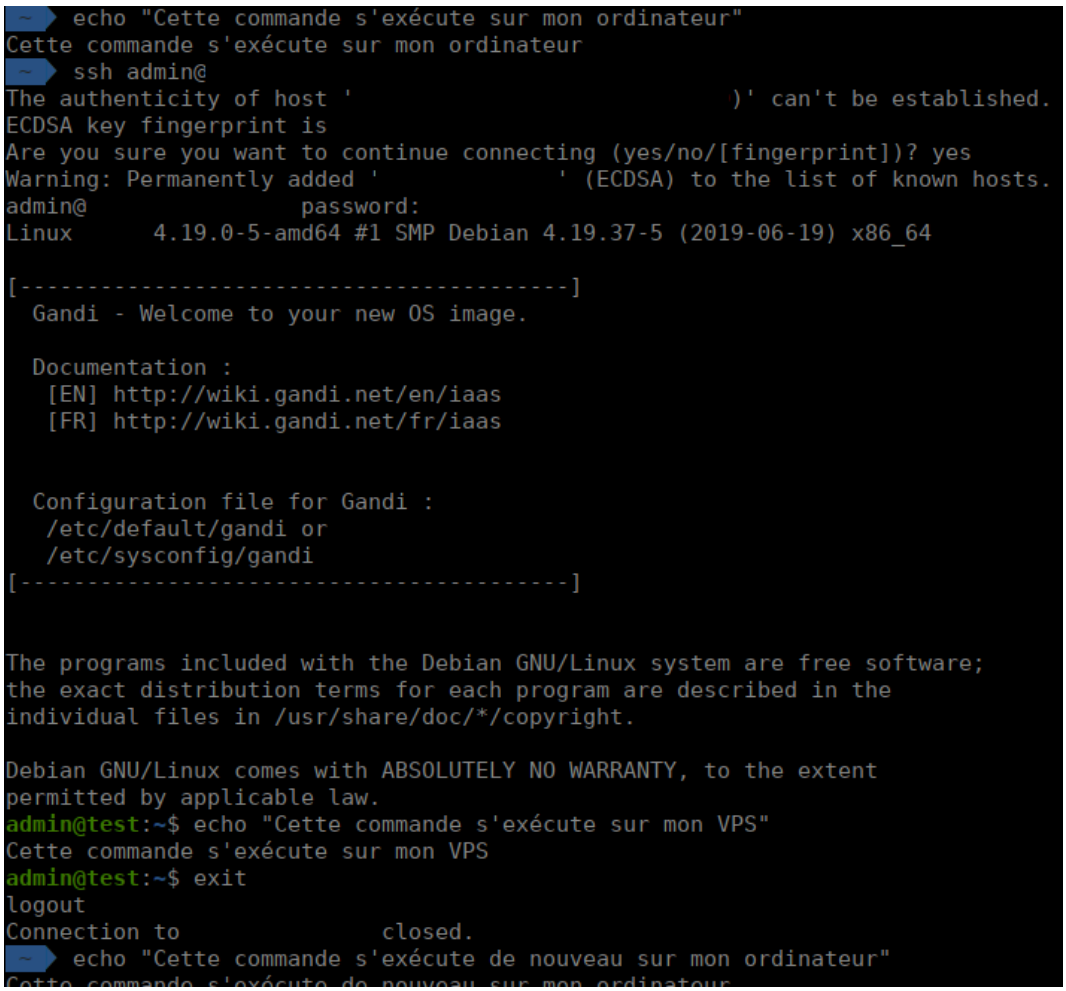
Pour travailler sur un VPS, il faut un moyen de s'y connecter et d'y ouvrir un shell. SSH (*Secure SHell*) est un outil standard qui remplit cette fonction : une fois la connexion établie, on travaille sur un VPS comme on travaille sur un shell local.

Dans un shell local, copier la commande reçue par mail pour ouvrir un shell distant sur le VPS.

```
ssh <super-utilisateur>@<adresse-IP>
```

Accéder à un VPS avec SSH

Exemple



Cette image montre une session SSH classique :

- La première commande s'exécute sur l'ordinateur local.
- Après la connexion SSH, les commandes s'exécutent automatiquement sur le VPS distant.
- La commande `exit` ferme la connexion SSH, les commandes s'exécutent de nouveau sur l'ordinateur local.

Autres fournisseurs de VPS français

Complément

Il existe d'autres fournisseurs de VPS français, comme OVH[3] et Scaleway[4].

3. OVH - https://www.ovh.com/fr/
4. Scaleway - https://www.scaleway.com/fr/

SSH et Windows 10

Complément

Windows n'intègre pas SSH par défaut. Il y plusieurs possibilités pour l'installer :

- Suivre le tutoriel de Microsoft[5] pour activer l'utilisation de SSH dans `powershell`.
- Installer un logiciel tiers, comme PuTTY[6].
- Utiliser SSH dans un shell Bash, en installant le sous système Linux[7].

Pourquoi louer un VPS et pas un serveur physique ?

Complément

Les VPS répondent à un problème classique : louer un serveur physique impose de choisir des composants adaptés à la puissance voulue. Si les besoins augmentent, il faut changer de machine, ce qui peut être très coûteux.

Les fournisseurs de serveurs ont trouvé une astuce : séparer un serveur physique en plusieurs serveurs **virtuels**, dont la puissance peut être adaptée en fonction des besoins. Pour les utilisateurs, le coût est moindre, et pour les fournisseurs, l'utilisation d'un serveur physique est optimisée.

À retenir

- Un VPS est l'équivalent d'un serveur que l'on peut louer pour son usage personnel. Il est accessible depuis Internet.
- SSH permet de se connecter à distance sur son VPS, et d'y exécuter des commandes.
- Il existe plusieurs fournisseurs de VPS français, comme Gandi, OVH ou Scaleway.

5. Installation d'OpenSSH sous Windows 10 - https://docs.microsoft.com/fr-fr/windows-server/administration/openssh/openssh_install_firstuse
6. PuTTY - https://putty.org/
7. https://docs.microsoft.com/fr-fr/windows/wsl/install-win10