**CREATION DE MACHINE VIRTUELLES ET MISE EN RESEAU**

Apres installation de l’OS il faut installer les additionneuses : drivers pour améliorer la communication entre la VM et la Machine Hôte

* Présenter les interfaces Graphiques : Gnome (défaut) et KDE

**Désactiver le mode graphique:**  sudo systemctl set-default multi-user.target

**Activer le mode graphique :**  sudo systemctl set-default graphical.target

Conseiller de travailler en mode console car il consomme moins de ressource que le GUI

* 3 possibilités d’accéder au mode console :
* désactiver le mode GUI
* Accéder à l’application console via GUI
* Accès à distance Via un émulateur de terminal : permettant de se connecter à distance

Putty ou MobileXtern

Info à renseigner :

L’adresse du serveur distant (son numéro de phone)

Préciser le port par lequel l’on souhaite accéder à l’application sur le serveur ( Par défaut c’est le port 22 pour SSH)

Déterminer l’adresse IP du serveur avec : ip a

Par défaut virtualBox protège les machines virtuelles contre l’accès à distance

(<https://www.it-connect.fr/configurer-le-port-forwarding-sur-une-vm-virtualbox%EF%BB%BF/> )

VirtualBox fait de la NAT (Faire correspondre des @ip en d’autres adresses).

**ACCES A DISTANCE SUR UN SERVER VIA EMULATEUR DE CONSOLE : PUTTY**

1. Créer la VM
2. Configurer la VM pour permettre l’accès distant

* Pour accéder à une machine Virtuelle à distance se trouvant sur une machine Hôte, il faut ajouter une règle permettant d’accéder à cette VM à partir de l’adresse IP de la machine Hôte : => Ajouter une règle **de redirection de port** :
* Aller sur les paramètres de la machine => Interface Réseau => choisir l’interface => puis =>avancé => redirection=> ajouter une nouvelle Règle
* Installer openssh-server : ***sudo apt install openssh-server***
* Activer le service ssh : ***sudo systemctl start ssh***

1. Installer PUTTY et démarrer le !

Renseigner le **host Name** ou **@Ip** de l’hote de la VM et le **Port : 22**

Choisir le type de connexion : **ssh** (plus sécurisé que **Telnet** où les messages sont en clair) et valider Open/

NB : si la connexion échoue, il faut vérifier que :

La machine sur laquelle l’on désire se connecter est bien **en marche**

le service ssh est bien actif sur la VM **: sudo systemctl status ssh**

le port d’écoute est bien le Port 22**: grep port /etc/ssh/sshd\_config**

**etc.**

**ACCES A DISTANCE ET ECHANGE DE CLE ENTRE 2 MACHINES**

1. Créer les VMs
2. Configurer les VMs pour permettre l’accès distant et la communication entre les machines

* Arrêter les VMs
* Attribuer à chaque VM une nouvelle interface réseau : Aller sur les paramètres de la machine => Interface Réseau => choisir l’interface (Adaptater )

Mode d’accès => **réseau interne**/pont/….

* Configurer l’interface ajoutée :
* Veriufier les inerface réseau***: ip a***

Exemple config static : réseau interner **192.168.0.0/24**

VM1 : @Ip 192.168.0.**4**/24 et @GW : 192.168.0.1

VM2 : @Ip 192.168.0.**5**/24 et @GW : 192.168.0.1

* Les configurations se font dans le ***/etc/netplan/01-network-manager-all.yaml*** pour les versions récentes de DEBIAN Ou dans le ***/etc/network/interfaces*** pour les anciennes versions.
* Pour la VM1

***Sudo nano*** ***/etc/netplan/01-network-manager-all.yaml***

NB : ne pas utiliser la touche Tab pour les espaces

network:

    Version: 2

    Renderer: NetworkManager

    ethernets:

       enp0s8:

          Dhcp4: yes/no

          Addresses: [192.168.0.4/24]

          Gateway4: 192.168.0.1

* Valider la configuration faite

*sudo netplan apply*

* Modifier le nom de la machine. Par exemple : Passer de kamyPC à Madara

***sudo*** *hostnamectl set-hostname* ***Madara***

* Faire du DNS interne en éditant le fichier /etc/hosts

*sudo nano /etc/hosts*

27.0.0.1 localhost127.0.1.1 kamyPC192.168.0.4 Madara192.168.0.5 Hashirama

* Tester la connectivité ( on ping La machine Hashirama sur l’interface enp0s8)

*ping -I enp0s8 Hashirama*

SCENARIO :

Un utilisateur se connecte sur un serveur distant pour effectuer des actions.

NB : le couple de clé Pub/Privée est relatif à chaque utilisateur

* Créer le nouvel utilisateur à configurer (sur Madara et Hashirama)
  + adduser naruto
* Escalader vers le nouveau user
  + su naruto
* Créer le couple de clés : depuis la machine cliente
  + ssh-keygen -t rsa
  + indiquer le repertoire/fichier où les clés seront stuquées. Par défaut il le stock dans /home/USER/.ssh de chaque USER
  + entrer une passephrase pour protéger la clé privée. Même si une personne arrive à se connecter mon compte (exemple azerty)
* Se déplacer ou se trouve le couple de clés
  + cd .ssh/
  + vérifier que les 2 clés pub/privée ont été créées et que seul le propriétaire à les droits sur la clé privée et les autres ont le droit de lecture de la clé pub.
* Partager la clé publique à tous les Utilisateurs que je souhaite. Copier la clé publique sur le server distant (ici Hashirama)
  + ssh-copy-id -i id\_rsa.pub naruto@hashirama
  + verifier que le fichier know\_host qui stoque les empruntes des différentes machine sur lesquelles je me connecte a été créé dans .ssh
  + se connecter directement sur le server avec le nom d’USER naruto et verifier la clé public a bien été ajouter au fichier /home/Hashirama/.ssh/authorized\_keys

nano /home/Hashirama/.ssh/authorized\_keys

* test la connexon
  + ssh naruto@hashirama
  + renseigner la phrase de passe