

Gérer en série des postes sous Debian-like depuis un poste sous Unix-like

Installation de Linux Mint

- Télécharger l'ISO de Linux Mint sur le [site officiel](#).
- Copier l'ISO sur une clés USB, un DVD ou une clés [Ventoy](#).
- Procéder à l'installation sur le poste (il faut probablement changer l'amorçage).

Une fois que l'installation est terminée, récupérer l'adresse IP locale (idéalement fixer l'adresse IP depuis le router) :

```
ip address
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp7s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group
default qlen 1000
    link/ether 40:61:86:1d:40:c6 brd ff:ff:ff:ff:ff:ff
3: wlp6s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
qlen 1000
    link/ether 1c:4b:d6:75:6d:52 brd ff:ff:ff:ff:ff:ff
    inet 192.168.188.46/24 brd 192.168.188.255 scope global dynamic noprefixroute wlp6s0
        valid_lft 863809sec preferred_lft 863809sec
    inet6 fe80::2803:cab0:e242:aaf5/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Ici, on note depuis l'interface `wlp6s0` que l'adresse IP locale est `192.168.188.46`.

Normalement le serveur SSH est installé et activé par défaut, on vérifie :

```
sudo systemctl status ssh
```

```
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: e>
   Active: active (running) since Mon 2021-09-20 11:32:56 CEST; 7min ago
   [...]

```

Si le processus n'est pas actif, ou si l'on obtient une erreur :

```
sudo apt install openssh-server -y
sudo systemctl enable ssh
sudo systemctl start ssh
```

On répète l'opération pour chaque PCs.

Préparations

Fichier DNS

Modifier le fichier `/etc/hosts` pour y entrer les adresses IP des PCs, exemple :

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.188.30 mintepn01
192.168.188.45 mintepn06
192.168.188.46 mintepn08
```

On peut tester la communication avec :

```
ping -c 4 mintepn08
```

```
PING mintepn08 (192.168.188.46) 56(84) octets de données.
64 octets de mintepn08 (192.168.188.46) : icmp_seq=1 ttl=64 temps=2.04 ms
64 octets de mintepn08 (192.168.188.46) : icmp_seq=2 ttl=64 temps=1.82 ms
64 octets de mintepn08 (192.168.188.46) : icmp_seq=3 ttl=64 temps=1.91 ms
64 octets de mintepn08 (192.168.188.46) : icmp_seq=4 ttl=64 temps=3.30 ms
```

```
--- statistiques ping mintepn08 ---  
4 paquets transmis, 4 reçus, 0% packet loss, time 3129ms  
rtt min/avg/max/mdev = 1.822/2.268/3.303/0.602 ms
```

Si on n'a pas de communication, on vérifie avant de continuer !

Clés SSH

On se connecte un première fois de façon classique en SSH :

```
ssh epnadm@epn08
```

Normalement il ajoute le PC dans le fichier `~/.ssh/known_host`.

Ensuite on transmet sa clés SSH publique :

```
ssh-keygen -t rsa # Pour créer la paire de clés SI ce n'est déjà fait  
ssh-copy-id epnadm@epn08 # Transfert de clés publique, à faire à chaque PC
```

Fichier permettant la remise à zéro des données

Créer un fichier `rc.local` :

```
#!/bin/sh -e  
#  
# rc.local  
mkdir /tmp/invite  
rsync -av /etc/skel/ /tmp/invite  
chown invite -R /tmp/invite  
[[ -d "/home/invite" ]] || ln -s /tmp/invite /home  
  
exit 0
```

Ansible sur le PC maître

Installation

Si [Ansible](#) n'est pas installé :

```
sudo dnf install ansible sshpass # Ici sous Fedora / sshpass dans le cas d'une installation un peu ancienne
```

Fichier `hosts` de Ansible

Éditer le fichier `/etc/ansible/hosts` pour y faire figurer les PCs avec les utilisateurs qui vont bien :

```
[workstations:vars]
ansible_user=epnadm
ansible_become_method=sudo

[workstations]
mintepn01
mintepn06
mintepn08      ansible_user=mickael[]ansible_become_pass='{{ another_passwd }}'
```

Ici, les lignes sont commentées étant donné que le nom d'utilisateur est mis en variable globale.

On peut faire un test de communication :

```
ansible -m ping workstations
```

```
[WARNING]: Platform linux on host mintepn08 is using the discovered Python
interpreter at /usr/bin/python3, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
mintepn08 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
mintepn06 | SUCCESS => {
  "ansible_facts": {
```

```
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

Fichier de secrets

Créer un fichier contenant le mot de passe de l'utilisateur ayant des droits `sudo` sur les PCs :

```
ansible-vault create secret
```

Pour séparer les clés des valeurs, il doit y avoir une espace ou une tabulation après `:` :

```
ansible_become_pass: <mot de passe utilisateur>
another_passwd: <un autre mot de passe>
```

Script Ansible

```
---
- name: Installs
  become: yes
  hosts: all
  vars:
    ansible_python_interpreter: /usr/bin/python3
  vars_files:
    - secret

  tasks:
    - name: Update & Upgrade
      apt:
        update_cache: yes
        upgrade: 'yes'
    - name: Add APT Key Ubuntu
      apt_key:
        keyserver: keyserver.ubuntu.com
        id: 83FBA1751378B444
    - name: Add LibreOffice repo
      apt_repository:
```

```
    validate_certs: no
    repo: 'deb http://ppa.launchpad.net/libreoffice/ppa/ubuntu focal main'
    state: present
register: repolo
- name: Update & Upgrade for LibreOffice
  apt:
    update_cache: yes
    upgrade: 'yes'
  when: repolo.changed
- name: Automatic upgrade
  ansible.builtin.shell:
    cmd: mintupdate-automation upgrade enable
- name: Install packages
  apt:
    pkg:
      - gimp
      - vim
      - bat
      - chromium
      - inkscape
      - gcompris
      - kmines
  state: latest
  update_cache: no
- name: Create user 'Invité'
  ansible.builtin.user:
    name: invite
    comment: Invité
    uid: 1001
    append: yes
    groups: adm,dialout,fax,cdrom,floppy,tape,audio,dip,video,plugdev,netdev,nopasswdlogin
    create_home: no
    home: /home/invite
- name: Remove invite's home directory
  file:
    path: /home/invite
    state: absent
- name: Copy rc.local
  ansible.builtin.copy:
    src: ./Data/rc.local
```

```
dest: /etc/rc.local
owner: root
group: root
mode: '0711'
register: cprclocal
- name: Unconditionally reboot the machine with all defaults
  reboot:
when: cprclocal.changed
```

Déplacer le fichier précédemment créé `rc.local` dans le dossier `Ansible/Data`.

Lancer le script Ansible :

```
ansible-playbook ~/Documents/Ansible/PB_install_mint.yml --ask-vault-pass
```

Patience le temps du déroulement.

Dorénavant les postes contiennent les paquets définis, mais aussi un utilisateur `Invité` dont le compte sera réinitialisé à chaque redémarrage de l'ordinateur.

Tips

Pour rajouter des applications sur tous les postes, il suffit d'entrer le nom du paquet à la suite de `pkg` dans le script Ansible, puis de relancer le script.

En cas de dysfonctionnement dans la réintialisation du compte `Invité`, jeter un œil [ici](#).

Révision #8

Créé 20 septembre 2021 09:24:52 par Mickaël G.

Mis à jour 9 août 2023 07:58:21 par Mickaël G.