Réseau UBUNTU

Tutoriel de configuration réseau Linux/Ubuntu

Étapes

1. Commandes de base
2. Désactivation IPv6
3. Connecter deux PC par adrese IP fixe
4. Connecter deux PC par nom via le fichier hosts
5. Configurer la passerelle Internet => les trois PC accèdent à Internet
6. Multicast DNS : le fichier hosts devient inutile
7. Ajout d'un serveur DHCP, la fin des adresses IP fixe
8. Ajout d'un serveur DNS, la fin des fichiers hosts et du Multicast DNS

Commandes de base

Nom de la machine

```
hostname                # voir le nom actuel de la machine
sudo hostname pc1       # modifier le nom
sudo nano /etc/hostname # modifier le nom dans le fichier hostname
sudo nano /etc/hosts    # modifier le nom dans le fichier hosts
hostname                # vérifier
```

Commande `ip`

```
ip link show # voir les interfaces réseau et les adresses MAC
ip a # voir les informations sur la configuration Ethernet et IP
ip link set enp0s3 down # désactiver une interface réseau
ip link set enp0s3 up # activer une interface réseau
ip neig # voir la table arp (liste des appareils réseau connus avec
adresse MAC et IP)
ip route # voir les routes, la passerelle
# autres
ping ip # envoyer un paquet ICMP à un appareil réseau
```

Netplan

Netplan est un outil de configuration facile des paramètres réseau sous Linux.

```
cd /etc/netplan  # dossier des fichiers de configuration
ls /etc/netplan # liste des fichiers
sudo nano /etc/netplan/<fichier>.yaml # modification du fichier de configuration
# pour créer un nouveau fichier de configuration qui sera également pris en compte
# sudo nano /etc/netplan/99-config.yaml
cat /etc/netplan/01-fichier.yaml # voir le contenu du fichier de configuration
sudo netplan apply # appliquer les changements
sudo netplan try # vérifier les changements
ip a # vérifier les changements
netplan ip leases <interface> # voir les baux DHCP
```

Autres commandes utiles

```
dig
nslookup
dhclient
dhcp-lease-list
rndc
systemd-resolve --status
resolvectl dns
resolvectl domain
resolvectl status
```

Désactivation ipv6 pour une interface réseau

Fichier .yaml dans le dossier /etc/netplan ajouter `link-local: []` :

```
network:
  ethernets:
    enp0s3:
      link-local: []
  version: 2
```

Appliquer et voir les changements

```
sudo netplan apply && ip a
```

Configuration adresse IP statique

Fichier .yaml dans le dossier /etc/netplan :

```
network:
  ethernets:
    enp0s3:
      link-local: []
      dhcp4: false
      addresses: [172.16.0.51/16]
  version: 2
```

Appliquer et voir les changements

```
sudo netplan apply
ip a
```

Note **VALIDATION**

Vous devez avoir deux PC pc1 et pc2 qui se voient mutuellement par adresse IP.

Configurez le fichiers hosts de pc1 avec l'adresse IP de pc2 et le fichier hosts de pc2 avec l'adresse IP de pc1.

Les PC se connaissent mutuellement : exécutez `ping pc1` et `ping pc2` sur chaque PC.

Configuration de la passerelle Internet

```
La passerelle a deux interfaces réseau :
- une en Bridged connectée à Internet
- une connectée au LAN Segment de pc1 et pc2
```

1. Nommez la passerelle par exemple `networkserver`
2. Donnez une adresse IP fixe à la passerelle
3. La passerelle est bien connectée à Internet : `ping 8.8.8.8` et `ping google.fr`

La passerelle a deux interfaces réseau. Récupérer l'adresse MAC de l'interface réseau connectée au LAN Segment. Voici le fichier yaml :

```
network:
  ethernets:
    enp0s3:
      dhcp: true
    enp0s8:
      dhcp4: false
      link-local: []
      match:
        macaddress: 00:0c:29:13:cb:69
      set-name: ens38
      addresses: [172.16.0.254/16]
  version: 2
```

Une passerelle est un routeur. Donc nous allons configurer le routeur. Décommencer la ligne `net.ipv4.ip_forward=1` dans le fichier `/etc/sysctl.conf` pour autoriser le fait de faire passer les paquets IP.

```
sudo nano /etc/sysctl.conf
```

Redémarrer la machine

```
sysctl net.ipv4.ip_forward # vérifier
```

NAT (Network Address Translation) est une méthode de routage des adresses IP. On peut l'utiliser pour communiquer entre deux réseaux ou encore partager une connexion Internet (avec une adresse IP publique) depuis son réseau LAN. Iptables permet de configurer cela à travers le masquerade. Lorsque vous avez besoin de faire transiter tout le trafic d'une interface spécifique par votre ordinateur sans rien changer à l'intérieur des paquets, vous pouvez utiliser masquerade.

Sur le serveur :

- Création de la règle MASQUERADE sur l'interface réseau locale.
- POSTROUTING n'affecte que le trafic sortant
- Blocage du traffic du réseau isolé. On bloque le FORWARD pour les trames qui ne concernent que le réseau interne.

```
iptables -t nat -A POSTROUTING -o <interface sortante> -j MASQUERADE
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -s <réseau_interne> -d <réseau_interne> -j DROP
```

```
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -s 172.16.0.0/16 -d  172.16.0.0/16 -j DROP
```

Vérifier les règles iptables :

```
iptables -t   nat -L
iptables -S
iptables -L
iptables -L -v
```

Supprimer les règles :

```
iptables -F # supprimer toutes les règles
iptables -D FORWARD -s 172.16.0.0/16 -d  172.16.0.0/16 -j DROP # supprimer
une règle précise
```

Sur le serveur, rendre les règles persistantes :

```
sudo apt install iptables-persistent -y
#sudo dpkg-reconfigure iptables-persistent # en cas de modification des
règles
```

Sur pc1 et pc2 : ajouter la passerelle (route par défaut) dans le fichier yaml de netplan ainsi qu'un DNS public :

```
network:
  ethernets:
    ens33:
      nameservers:
        # DNS public de google on changera plus tard
        addresses: [8.8.8.8, 8.8.4.4]
      routes:
        - to: default
          via: 172.16.0.254
```

```
sudo netplan apply
```

Note **VALIDATION**

Redémarrer le serveur

pc1 et pc2 ont accès à Internet : `ping 8.8.8.8` `ping google.fr`

Nous allons maintenant activer le multicast DNS, que nous désactiverons plus tard. Le Multicast DNS permet aux machines de diffuser leur nom sur le réseau local sans avoir besoin de modifier le fichies hosts.

1. Dans le fichier hosts de pc1 supprimer l'entrée de pc2
2. Dans le fichier hosts de pc2 supprimer l'entrée de pc1
3. Les machines ne peuvent plus se reconnaître par leur nom
4. Installer avahi sur pc1, pc2 et sur le serveur

Solutions alternatives à AVAHI : SAMBA (service de partage de fichiers sous Windows) / NetBios (non recommandé).
D'autres solutions existent ; le DNS reste la meilleure solution aujourd'hui.

Pour pouvoir installer avahi sur les trois machines :

```
sudo apt update && sudo apt install avahi-daemon -y
```

Comme l'APIPIA, NetBios et AVAHI utilisent les adresses réseau de type Broadcast pour diffuser les noms des machines. Cela augmente le nombre de paquets qui circulent sur le réseau, ce qui rend cette configuration impopulaire auprès des administarteurs réseau. Possible pour un particulier ou une TPE ; sinon préférer un serveur DNS.

Le Multicast DNS utilise l'adresse IP 224.0.0.251 et le port UDP 5353.

Le Multicast DNS utilise le nom de domaine `.local`. Ne pas utiliser dans un autre cadre.

Note **VALIDATION**

Le fichiers hosts des machines ne contient pas ni le nom ni l'adresse IP des autres machines.

Les trois machines se connaissent : `ping pc1.local` `ping pc2.local` `ping server.local`

Installation du serveur DHCP et configuration

```
sudo apt install isc-dhcp-server
sudo nano /etc/dhcp/dhcpd.conf
```

```
# /etc/dhcp/dhcpd.conf
option domain-name "reseau1.lan"
option domain-name-servers 8.8.8.8, 8.8.8.4;

default-lease-time 600;
max-lease-time 7200;

subnet 172.16.0.0 netmask 255.255.0.0 {
  range 172.16.1.51 172.16.1.100;
  option routers 172.16.0.254;
  option subnet-mask 255.255.0.0;
}
```

Pour attribuer une adresse IP fixe :

```
  host client1 {
      hardware ethernet DD:GH:DF:E5:F7:D7;
      fixed-address 192.168.1.20;
  }
```

```
# /etc/default/isc-dhcp-server
# interface du réseau local ; vérifier avec ip a
INTERFACESv4="enp0s8"
INTERFACESv6=""
```

Redémarrer le serveur dhcp

```
sudo /etc/init.d/isc-dhcp-server restart
```

En cas de souci de configuration, pour voir les erreurs :

```
sudo systemctl start isc-dhcp-server
sudo systemctl status isc-dhcp-server
# récupérer la ligne Main PID non pas 1777 mais votre PID
journalctl _PID=1777 | more
```

Configurer pc1 et pc2 en dhcp

```
network:
  ethernets:
    enp0s3:
      link-local: []
      dhcp: true
  version: 2
```

Note **VALIDATION**

pc1 et pc2 ont une adresse IP attribuée par le serveur DHCP

Il se connaissent par leurs noms grâce au MulticastDNS. Les serveurs DNS publics sont également correctement configurés.

Installation de `bind9` en tant que serveur DNS, sur le même serveur que le DHCP

```
sudo apt install bind9 -y
# pour tout recommencer
# sudo rm -fR /etc/bind && sudo apt -o DPkg::Options::="--force-confmiss"
--reinstall install bind9
```

Attention **Vérifiez que vous êtes bien sur le serveur**

Pour le DNS voir https://www.rfc-editor.org/rfc/rfc1918.txt, https://www.rfc-editor.org/rfc/rfc1912.txt, https://www.rfc-editor.org/rfc/rfc6303.txt, https://www.rfc-editor.org/rfc/rfc8375.html, https://www.rfc-editor.org/rfc/rfc6762#appendix-G https://www.rfc-editor.org/rfc/rfc6761

Trouver les machines sur Internet (DNS de Google)

```
sudo nano /etc/bind/named.conf.options
```

Remarque : normalement, les forwarders devraient être les serveurs fournis par DHCP sur l'interface Internet. Bind n'est pas prévu pour faire cela, on doit configurer les forwarders à la main.

```
# /etc/bind/named.conf.options
acl reseau1 {
  172.16.0.0/16;
};

options {
  directory "/var/cache/bind";

  forwarders {
    8.8.8.8;
    8.8.4.4;
  };
  dnssec-validation auto;

  #listen-on-v6 { any; }; # supprimer cette ligne

  listen-on {
    172.16.0.0/16;
  };
  allow-query { reseau1; };
};
```

Configuration de nos zones locales

```
sudo nano /etc/bind/named.conf.local
```

```
# /etc/bind/named.conf.local

zone "reseau1.lan" {
  type master;
  file "/etc/bind/db.reseau1.lan";
};

# 16.172 adresse IP du réseau à l'envers
# in-addr.arpa nom de domaine réseau local
zone "16.172.in-addr.arpa" {
  type master;
  file "/etc/bind/db.16.172";
};
```

Configuration de la zone directe (conversion de noms en adresses IP)

```
sudo cp /etc/bind/db.local  /etc/bind/db.reseau1.lan
sudo nano /etc/bind/db.reseau1.lan
```

> Dans les fichiers DNS il faut augmenter le SERIAL de une unité à chaque modification du fichier

```
# /etc/bind/db.reseau1.lan
$TTL    604800
@       IN      SOA     ns.reseau1.lan. root.reseau1.lan. (
                              4         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.reseau1.lan.
ns      IN      A       172.16.0.254
www     IN      A       172.16.0.254
```

Configuration de la zone indirecte (conversion d'adresses IP en nom)

```
sudo cp /etc/bind/db.127 /etc/bind/db.16.172
sudo nano /etc/bind/db.16.172
```

```
# /etc/bind/db.16.172

$TTL    604800
@       IN      SOA     ns.reseau1.lan. root.reseau1.lan. (
                              3         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.reseau1.lan.
254.0   IN      PTR     ns.reseau1.lan.
```

> Pour chaque enregistrement A ou AAAA dans `db.reseau1.lan` in faut un enregistrement `PTR` dans `db.16.172`

Vérifier la configuration

```
sudo named-checkconf
sudo named-checkzone reseau1.lan /etc/bind/db.reseau1.lan
sudo named-checkzone 16.172.in-addr.arpa /etc/bind/db.16.172
```

Désactiver le DNS obtenu par DHCP dans le fichier `.yaml` du dossier `/etc/netplan`

```
# /etc/netplan/config.yaml
network:
  ethernets:
    ens33:
      dhcp4: true
      dhcp4-overrides:
        use-dns: no
      nameservers:
        addresses: [172.16.0.254]
    ens38:
      dhcp4: false
      link-local: []
      match:
        macaddress: <adresse>
      set-name: ens38
      nameservers:
        addresses: [172.16.0.254]
          # chercher les machines dans reseau1.lan
          # ping pc1 => ping pc1.reseau1.lan
          search: [reseau1.lan]
      addresses: [172.16.0.254/16]
  version: 2
```

Désactiver ou supprimer ahahi sur les trois PC

```
# désactiver avahi
sudo systemctl disable avahi-daemon
sudo systemctl stop avahi-daemon
sudo systemctl mask avahi-daemon
# ou bien désinstaller avahi
# sudo apt remove avahi-daemon -y
```

Mettre à jour le serveur dhcp

```
# /etc/dhcp/dhcpd.conf
option domain-name "reseau1.lan";
option domain-name-servers 172.16.0.254;
```

Redémarrer si nécessaire

```
sudo reboot # redémarrer la machine
```

Voir la correspondance ns.reseau1.lan <-> 172.16.0.254

```
ping -4 ns.reseau1.lan
host -4 172.16.0.254
dig -4 -x 172.16.0.254
dig -4 +noall +answer -x 172.16.0.254
nslookup 172.16.0.254
```

Activer le DNS Dynamique

Le DNS Dymamique (DDNS) permet au serveur DHCP de récupérer le nom de la machine qui demande une adresse IP et d'actualiser automatiquement le serveur DNS.

Fichier `/etc/default/named` :

```
RESOLVCONF=no
OPTIONS="-4 -u bind"
```

Clé commune

Créer une clé commune entre le serveur DHCP et le serveur DNS.

```
ddns-confgen
```

Copier / coller les 4 lignes concernant la clé dans le fichier `/etc/bind/ddns.key`

```
# /etc/bind/ddns.key
key "ddns-key" {
        algorithm hmac-sha256;
        secret "w2BhyMvsZue+Aa3tlFxoASknGBfLjPQgMvet0UDAfvU=";
};
```

```
# recopie du fichier pour éviter les problèmes de droits
sudo cp /etc/bind/ddns.key /etc/dhcp/ddns-keys/ddns.key
sudo chown root:dhcpd /etc/dhcp/ddns-keys/ddns.key
sudo chmod 640 /etc/dhcp/ddns-keys/ddns.key /etc/bind/ddns.key
```

DDNS et bind9

Déplacer les fichiers de zone dans `/var/lib/bind` au lieu de `/etc/bind` :

```
sudo mv /etc/bind/db.reseau1.lan /var/lib/bind/db.reseau1.lan
sudo mv /etc/bind/db.16.172 /var/lib/bind/db.16.172
```

Nous allons :

1. Mettre à jour le fichier de configuration avec le chemin `/var/lib/bind`
2. Ajouter la clé `ddns-key`

```
# /etc/bind/named.conf.local
include "/etc/bind/ddns.key";

zone "reseau1.lan" {
  type master;
  file "/var/lib/bind/db.reseau1.lan";
  update-policy {
    grant ddns-key zonesub ANY;
  };
};

zone "16.172.in-addr.arpa" {
  type master;
  file "/var/lib/bind/db.16.172";
  update-policy {
    grant ddns-key zonesub ANY;
  };
};
```

DDNS et DHCP

```
# /etc/dhcp/dhcpd.conf
include "/etc/dhcp/ddns-keys/ddns.key";

ddns-updates on;
ddns-update-style standard;
authoritative;
allow unknown-clients;
update-optimization off;
option domain-name "reseau1.lan";
option domain-search "reseau1.lan";

default-lease-time 600;
max-lease-time 7200;

zone reseau1.lan. {
  primary 172.16.0.254;
  key ddns-key;
}


zone 16.172.in-addr.arpa. {
  primary 172.16.0.254;
  key ddns-key;
}

subnet ...

host ...
```

> Assurez-vous bien que la ligne `ddns-update-style none;` est supprimée ou bien en commentaire avec un `#` devant.

Sur pc1 et pc2, arrêter les interfaces réseau :

```
sudo ip link set enp0s3 down
```

Sur le serveur dhcp/dns, redémarrer les services dns et dhcp :

```
sudo named-checkconf
sudo named-checkzone reseau1.lan /var/lib/bind/db.reseau1.lan
sudo named-checkzone 16.172.in-addr.arpa /var/lib/bind/db.16.172
```

```
sudo service bind9 stop
sudo service isc-dhcp-server stop
sudo service bind9 start
sudo service isc-dhcp-server start
```

Vérifier le bon démarrage des services. Corriger les erreurs éventuelles. Redémarrer la machine si nécessaire et revérifer les statuts.

```
sudo service bind9 status               # voir l'état
sudo service isc-dhcp-server status     # voir l'état
# journalctl _PID=1777 | more           # voir les détails en cas d'erreur
# sudo reboot
```

En cas de problème, vous pouvez supprimer les fichier jnl dans `/var/lib/bind` avec `sudo rm /var/lib/bind/*.jnl` et redémarrer `bind9` ou sinon :

```
rndc freeze reseau1.lan
# modifier /var/lib/bind/*
#  ajouter des enregistrements A ou CNAME
rndc reload reseau1.lan
rndc thaw reseau1.lan
```

Observer les requêtes dhcp sur le serveur. Exécuter la commande sur le serveur, garder la fenêtre ouverte en exécutant les autres commandes sur les autres machines :

```
journalctl -xef -u named -u isc-dhcp-server
```

Une fois que le serveur DHCP/DNS/DDNS fonctionne, sur pc1 et pc2, redémarrer les interfaces réseau :

```
sudo ip link set enp0s3 up
sudo dhclient # renouveler le bail DHCP
```

Vérfier sur le serveur :

```
ls /var/lib/bind # normalement il y a un fichier .jnl
dhcp-lease-list # liste des baux dhcp accordés par le serveur
rndc dumpdb -zones
cat /var/cache/bind/named_dump.db
```

Si tout va bien, les trois machines arrivent à se reconnaître mutuellement :

```
ping ns
ping pc1
ping pc2

host ns.reseau1.lan
host adresse_ip_dhcp_pc1
host adresse_ip_dhcp_pc2
```

Note **VALIDATION**

Les trois machines arrivent à se reconnaître mutuellement. Si on ajoute une quatrième machine appelée pc3 en dhcp, toutes les autres machines la reconnaissent automatiquement.

Si on ajoute une cinquième machine appelée imprimante1 avec une adresse ip fixe au niveau du serveur dhcp, elle est également reconnue.
