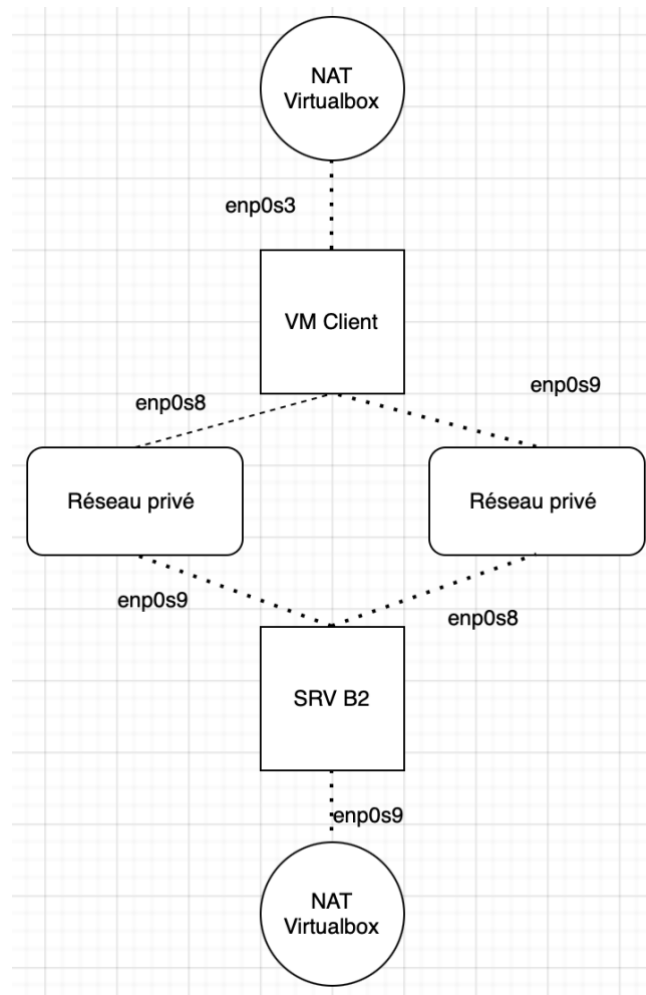


Mise en place d'un serveur DHCP, DNS, HTTP(S) et SSH avec Cockpit sous Linux



Voici un schéma de l'infrastructure une fois terminer

1. Installation du serveur DHCP avec systemd-networkd

Sur la VM SRV, assurez-vous que **systemd-networkd** et **systemd-resolved** sont installés et activés :

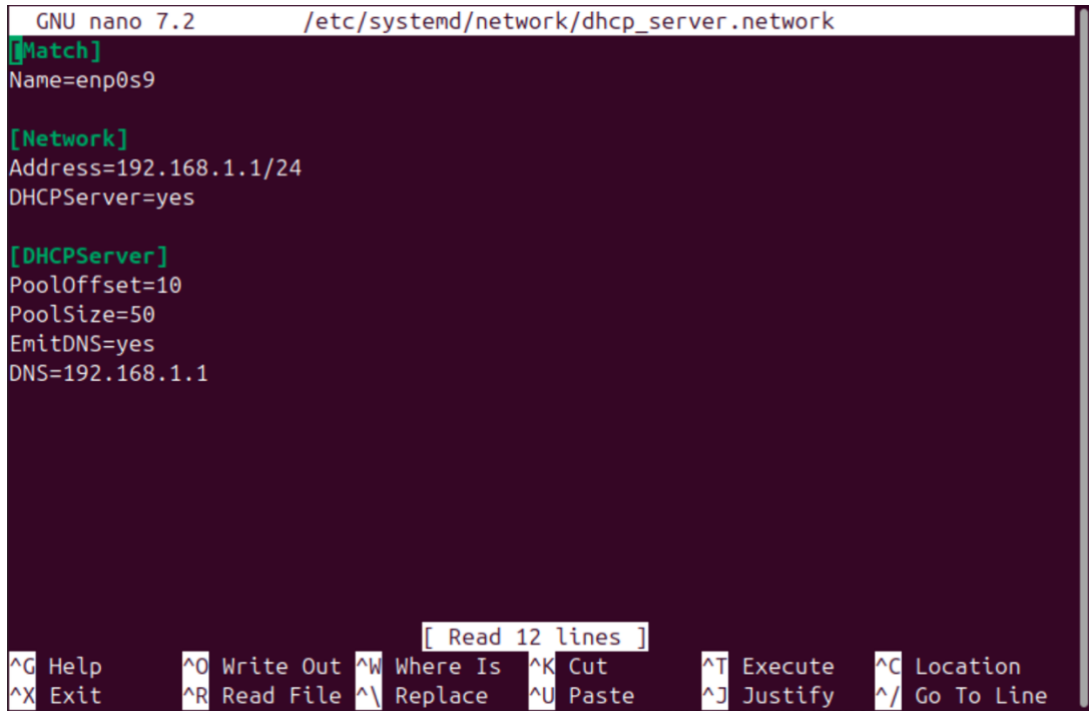
```
sudo systemctl enable systemd-networkd  
sudo systemctl enable systemd-resolved
```

Configuration du DHCP

Créez un fichier de configuration pour **systemd-networkd** :

```
sudo nano /etc/systemd/network/dhcp_server.network
```

Ajoutez le contenu suivant :



```
GNU nano 7.2 /etc/systemd/network/dhcp_server.network
[Match]
Name=enp0s9

[Network]
Address=192.168.1.1/24
DHCPServer=yes

[DHCPServer]
PoolOffset=10
PoolSize=50
EmitDNS=yes
DNS=192.168.1.1

[ Read 12 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

A la place de enp0s9 mettez votre interface réseau

Activez la configuration :

```
sudo systemctl restart systemd-networkd
```

Sur la VM Client créez un fichier de configuration :

```
sudo nano /etc/systemd/network/dhcp_client.network
```

Ajoutez :

```
pat2@pat2: /etc/systemd/network
GNU nano 7.2 dhcp_client.network
[Match]
Name=enp0s9

[Network]
DHCP=yes
```

Activez la configuration :

```
sudo systemctl restart systemd-networkd
```

Avec IP à vérifier que votre client a la bonne IP :

```
pat2@pat2: ~
valid_lft forever preferred_lft forever
2: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:eb:a6:64 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.41/24 metric 1024 brd 192.168.1.255 scope global dynamic enp0
s9
        valid_lft 3490sec preferred_lft 3490sec
        inet 192.168.1.14/24 brd 192.168.1.255 scope global secondary dynamic nopref
ixroute enp0s9
            valid_lft 3493sec preferred_lft 3493sec
            inet6 fe80::a00:27ff:feeb:a664/64 scope link
            valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:16:03:de brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s8
        valid_lft 86293sec preferred_lft 86293sec
        inet6 fd00::7ef1:3fb5:9fc0:c947/64 scope global temporary dynamic
        valid_lft 86294sec preferred_lft 14294sec
        inet6 fd00::a00:27ff:fe16:3de/64 scope global dynamic mngtmpaddr
        valid_lft 86294sec preferred_lft 14294sec
        inet6 fe80::a00:27ff:fe16:3de/64 scope link
        valid_lft forever preferred_lft forever
pat2@pat2:~$
```

2. Installation et configuration de BIND

Installation de Bind :

```
sudo apt update && sudo apt install bind9 -y
```

Configuration de la zone DNS

Éditez le fichier de configuration principal :

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

Ajoutez :

```
GNU nano 7.2 /etc/bind/named.conf.local
zone "ynov.b2" {
    type master;
    file "/etc/bind/db.ynov.b2";
};
```

Créez le fichier de zone :

```
GNU nano 7.2 /etc/bind/db.ynov.b2
$TTL 86400
@      IN      SOA  ns.ynov.b2. admin.ynov.b2. (
        2024011001    ; Serial
        3600          ; Refresh
        1800          ; Retry
        604800        ; Expire
        86400 )       ; Minimum TTL

@      IN      NS   ns.ynov.b2.
ns     IN      A    192.168.1.1
client IN      A    192.168.1.50
site3  IN      A    192.168.1.1
site4  IN      A    192.168.1.1
```

Vérifiez la configuration :

```
sudo named-checkconf
```

```
sudo named-checkzone ynov.b2 /etc/bind/db.ynov.b2
```

Redémarrez le service :

```
sudo systemctl restart bind9
```

3. Installation et configuration d'Apache2 avec HTTPS

Installation d'Apache2 :

```
sudo apt install apache2 -y
```

Activez le support SSL et HTTP/2 :

```
sudo a2enmod ssl http2
```

```
sudo systemctl restart apache2
```

Créations des VirtualHosts

Créez deux fichiers de configuration :

```
sudo nano /etc/apache2/sites-available/site3.conf
```

Ajoutez :

```
GNU nano 7.2 /etc/apache2/sites-available/site3.conf
<VirtualHost *:443>
  ServerName site3.ynov.b2
  DocumentRoot /var/www/site3

  SSLEngine on
  SSLCertificateFile /etc/ssl/certs/site3.crt
  SSLCertificateKeyFile /etc/ssl/private/site3.key

  # Activer HTTP/2
  Protocols h2 http/1.1
</VirtualHost>
```

Faites de même pour site4.conf en remplaçant site3 par site4.

Activez les sites :

```
sudo a2ensite site3.conf site4.conf
```

```
sudo systemctl restart apache2
```

Sur le client essayer d'accéder au site :

```
curl -k https://site3.ynov.b2
```

```
pat2@pat2:~$ curl -k https://site3.ynov.b2
Welcome to Site 3
pat2@pat2:~$
```

4. Création d'une Autorité de Certification (CA) et signature des certificats

Générer une clé privée pour le CA :

```
sudo openssl genrsa -out /etc/ssl/private/ca.key 4096
```

Créer un certificat auto-signé pour le CA :

```
sudo openssl req -x509 -new -nodes -key /etc/ssl/private/ca.key -out  
/etc/ssl/certs/ca.crt -days 3650 -subj "/CN=MyCA"
```

Générer une clé privée pour le site :

```
sudo openssl genrsa -out /etc/ssl/private/site3.key 4096
```

Créer une demande de signature de certificat (CSR) :

```
sudo openssl req -new -key /etc/ssl/private/site3.key -out /etc/ssl/private/site3.csr -subj  
"/CN=site1.ynov.b2"
```

Signer le certificat avec le CA :

```
sudo openssl x509 -req -in /etc/ssl/private/site3.csr -CA /etc/ssl/certs/ca.crt -CAkey  
/etc/ssl/private/ca.key -CAcreateserial -out /etc/ssl/certs/site3.crt -days 365
```

Vérifier que le certificat a bien été généré :

```
openssl x509 -in /etc/ssl/certs/site3.crt -text -noout  
pat2@ubuntu:~$ openssl x509 -in /etc/ssl/certs/site3.crt -text -noout  
Certificate:  
  Data:  
    Version: 1 (0x0)  
    Serial Number:  
      04:76:e3:57:d2:93:97:df:7b:30:a5:bc:60:42:51:a0:95:eb:c0:78  
    Signature Algorithm: sha256WithRSAEncryption  
    Issuer: CN = MyCA  
    Validity  
      Not Before: Jan 11 12:17:40 2025 GMT  
      Not After : Jan 11 12:17:40 2026 GMT  
    Subject: CN = site3.ynov.b2  
    Subject Public Key Info:  
      Public Key Algorithm: rsaEncryption  
      Public-Key: (2048 bit)  
      Modulus:  
        00:b5:01:6e:86:e3:56:b2:57:55:3d:e4:90:bd:59:  
        9a:ee:bd:4d:00:ce:a4:d6:13:4b:82:78:cc:1e:87:  
        fe:58:3e:0c:8c:c3:9e:39:ad:dc:66:e4:6a:75:2b:  
        0f:3c:bd:f1:24:76:c1:5c:6d:ef:d4:f2:c6:f5:9e:  
        39:eb:d5:2c:31:4a:ab:39:7e:09:18:97:c5:9c:f8:  
        8d:67:1a:bf:65:07:8e:4d:86:ae:b9:60:ad:c4:cc:  
        b2:da:53:07:a0:ef:66:97:31:66:74:2c:d4:02:74:  
        9a:56:7c:86:b0:fa:6f:83:11:95:33:d6:79:02:b6:  
        a0:1d:44:7d:74:a6:56:fd:06:39:90:82:56:3a:d3:  
        fa:cd:59:10:57:a9:ed:33:fb:f4:f0:82:37:65:99:  
        13:f2:d3:e9:3c:65:a3:1c:2a:3c:77:a9:8d:05:d1:  
        48:b4:aa:64:9d:9e:37:da:78:a7:95:ca:36:c7:3d:
```

Ensuite reconfigurer les VirtualHost pour HTTPS :

```
sudo nano /etc/apache2/sites-available/site3.conf
```

Ajouter ces lignes :

```
SSLCertificateFile /etc/ssl/certs/site3.crt
```

```
SSLCertificateKeyFile /etc/ssl/private/site3.key
```

Activer le site et recharger Apache :

```
sudo a2ensite site3.conf
```

```
sudo systemctl restart apache2
```

Faire la même avec les autres sites

Copier le certificat sur la VM client :

Sur la VM Serveur :

```
scp /etc/ssl/certs/site3.crt user@192.168.1.X:/home/user/ (en remplaçant user et l'IP par la vôtre)
```

Sur la VM Client :

```
sudo mv /home/user/site3.crt /etc/ssl/certs/
```

Ajouter le certificat CA dans Firefox :

1. **Ouvrir Firefox**
2. Aller dans **Paramètres** → **Confidentialité et Sécurité**
3. Descendre jusqu'à **Certificats** et cliquer sur **Afficher les certificats**
4. Aller dans l'onglet **Autorités** → **Importer**
5. Sélectionner ca.crt
6. **Cocher** : ☒ *Faire confiance à cette autorité de certification pour identifier des sites web*
7. Valider avec **OK**

Vérification et tests

Tester l'accès HTTPS depuis le client :

```
curl https://site3.ynov.b2
```

```
pat2@pat2:~$ curl https://site3.ynov.b2
Welcome to Site 3
```


Ici on voit aucune erreur SSL ce qui montre que le protocole fonctionne bien

5. Mise en place d'un service SSH avec authentification par clé

Sur la VM serveur, installer OpenSSH Server :

```
sudo apt install openssh-server -y
```

Génération et configuration des clés SSH :

Sur la VM Client :

```
ssh-keygen -t rsa -b 4096
```

Copiez la clé publique sur le serveur :

```
ssh-copy-id pat2@192.168.1.1 (à la place de pat2 mettre l'utilisateur de la VM Serveur)
```

Configuration de SSH pour n'autoriser que l'authentification par clé :

```
sudo nano /etc/ssh/sshd_config
```

Modifiez ou ajoutez les lignes suivantes :

```
GNU nano 7.2 /etc/ssh/sshd_config

# no default banner path
#Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#    X11Forwarding no
#    AllowTcpForwarding no
#    PermitTTY no
#    ForceCommand cvs server
PasswordAuthentication no
PubkeyAuthentication yes
ChallengeResponseAuthentication no
```

Redémarrer SSH :

```
sudo systemctl restart ssh
```


Tester la connexion SSH :

`ssh pat2@192.168.1.1` (remplacer pat2 par l'utilisateur de votre serveur)

```
pat2@pat2:~$ ssh pat2@192.168.1.1
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-51-generic aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of din. 12 janv. 2025 18:59:22 UTC

System load:            0.12
Usage of /:              81.3% of 10.70GB
Memory usage:           64%
Swap usage:             40%
Processes:              204
Users logged in:        1
IPv4 address for enp0s8: 10.0.2.15
IPv6 address for enp0s8: fd00::4a89:1d51:931:2482
IPv6 address for enp0s8: fd00::a00:27ff:fe23:10a6

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

La maintenance de sécurité étendue pour Applications n'est pas activée.
18 mises à jour peuvent être appliquées immédiatement.
```

6. Installation et configuration de Cockpit

Installation de Cockpit :

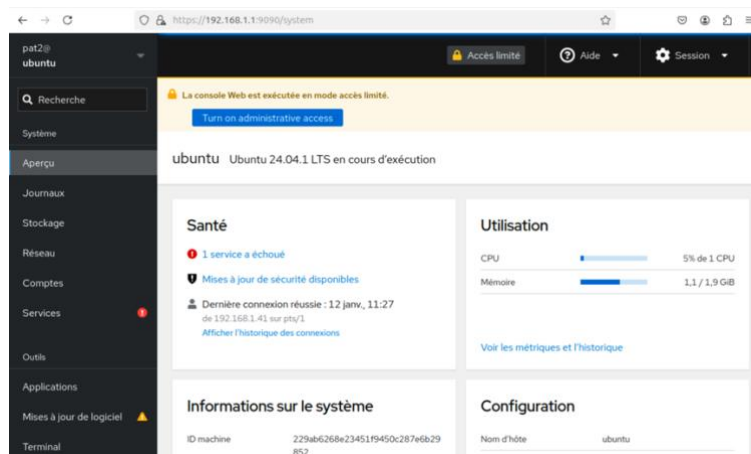
`sudo apt install cockpit -y`

Activez et démarrez Cockpit :

`sudo systemctl start cockpit.socket`

Accédez à Cockpit via un navigateur à l'adresse :

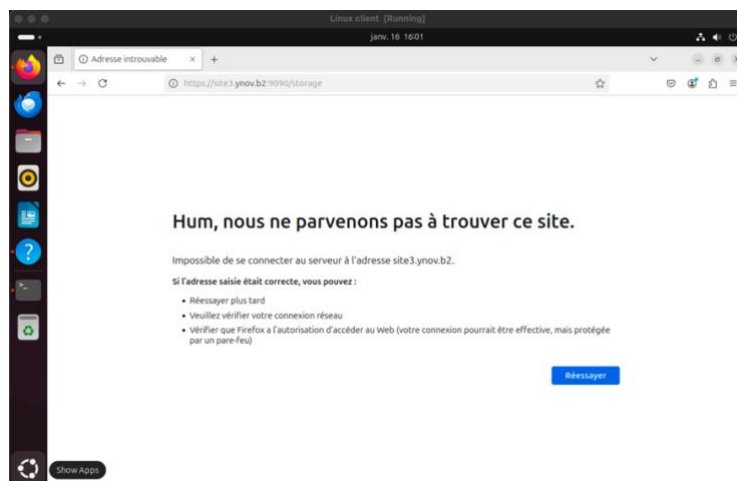
<https://site3.ynov.b2:9090/> ou <https://192.168.1.1:9090/> (mettre l'adresse IP du serveur) connecter vous avec vos identifiants



Installer firewalld sur la VM Serveur :

`sudo apt install firewalld`

En faisant cela on ne peut plus accéder à cockpit avec la VM Client



Dans l'interface graphique, ajouter le service nécessaire pour que le pare-feu autorise le port 9090 :



Redémarrer cockpit :

`sudo restart systemctl cockpit`

Accéder à cockpit à partir de la VM Client :

