# Livrable 5

Amberny Peran, Barnouin Clement, Burellier Loucas,
Krainik-Saul Vladimir, Schicke Samuel,

2025-03-21

## 1 Introduction

MiniCoffee est un groupe français spécialiste de l'univers du café, connu notamment pour ses machines à café en libre service. Pour l'année 2025, l'entreprise souhaite mettre à jour son infrastructure réseau interne en ajoutant :

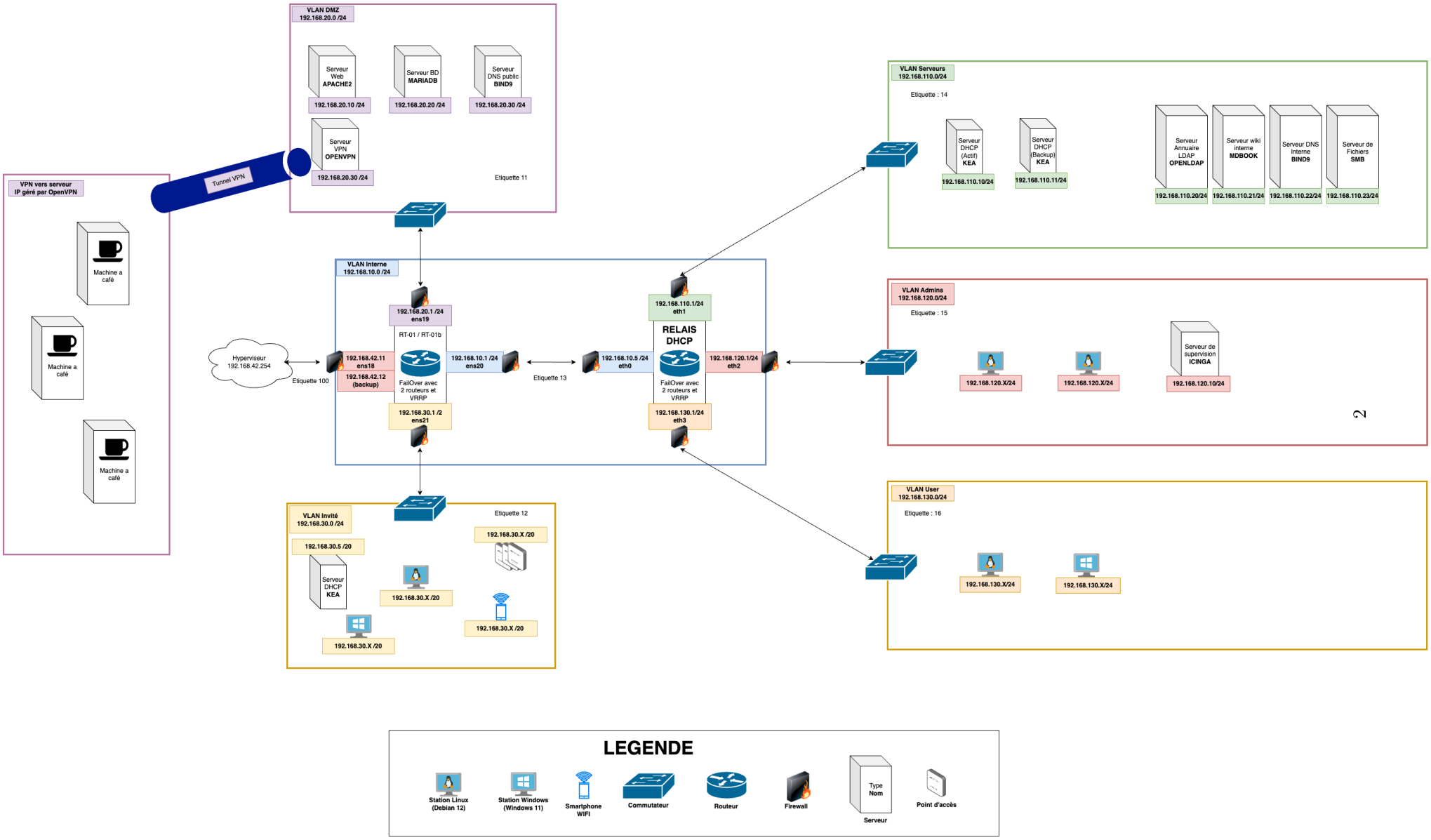
- Divers serveurs d'utilité interne pour les employés et l'équipe informatique;
- Un réseau invité pour permettre à ses fournisseurs d'utiliser du matériel informatique sur place;
- Divers serveurs accessibles en ligne (site Web, serveur DNS public);
- Une meilleure communication entre ses machines à café et son infrastructure, qui a été un des points faibles de l'entreprise ces dernières années.

Pour cette tâche, MiniCoffee a fait appel à BAV4, notre équipe d'étudiants de l'IUT2 Informatique de Grenoble.

## 2 Architecture

L'architecture de notre réseau n'a pas énormément changé. Les seules modifications apportées au réseau sont :

- Passage d'un LAN à un VLAN pour une meilleure segmentation du réseau.
- Les adresses IP internes se terminent par 1XX.
- Les adresses IP externes se terminent par XX.


VLAN DMZ
192.168.20.0 /24
Serveur Web APACHE2
192.168.20.10 /24
Serveur BD MARIADB
192.168.20.20 /24
Serveur DNS public BIND9
192.168.20.30 /24
Serveur VPN OPENVPN
192.168.20.30 /24
Etiquette 11
Tunnel VPN
VPN vers serveur IP géré par OpenVPN
Machine a café
Machine a café
Machine a café
VLAN Interne
192.168.10.0 /24
192.168.20.1 /24 ens19
RT-01 / RT-01b
Hyperviseur 192.168.42.254
Etiquette 100
192.168.42.11 ens18
192.168.42.12 (backup)
FailOver avec 2 routeurs et VRRP
192.168.10.1 /24 ens20
Etiquette 13
192.168.10.5 /24 eth0
RELAIS DHCP
192.168.110.1/24 eth1
192.168.120.1/24 eth2
FailOver avec 2 routeurs et VRRP
192.168.130.1/24 eth3
192.168.30.1 /2 ens21
VLAN Invité
192.168.30.0 /24
Etiquette 12
192.168.30.5 /20
Serveur DHCP KEA
192.168.30.X /20
192.168.30.X /20
192.168.30.X /20
192.168.30.X /20
VLAN Serveurs
192.168.110.0/24
Etiquette : 14
Serveur DHCP (Actif) KEA
192.168.110.10/24
Serveur DHCP (Backup) KEA
192.168.110.11/24
Serveur Annuaire LDAP OPENLDAP
192.168.110.20/24
Serveur wiki interne MDBOOK
192.168.110.21/24
Serveur DNS Interne BIND9
192.168.110.22/24
Serveur de Fichiers SMB
192.168.110.23/24
VLAN Admins
192.168.120.0/24
Etiquette : 15
192.168.120.X/24
192.168.120.X/24
Serveur de supervision ICINGA
192.168.120.10/24
VLAN User
192.168.130.0/24
Etiquette : 16
192.168.130.X/24
192.168.130.X/24
LEGENDE
Station Linux (Debian 12)
Station Windows (Windows 11)
Smartphone WIFI
Commutateur
Routeur
Firewall
Type Nom
Serveur
Point d'accès

# 3 Ressources Matérielles utilisées

Actuellement, notre infrastructure réseau comprend un total de 9 machines actives, chacune jouant un rôle spécifique dans notre environnement.

En ce qui concerne le stockage, nous allouons entre 3 et 5 Go d'espace disque par machine, en fonction de leurs besoins en ressources et des tâches qu'elles doivent accomplir. Plus précisément :

- Les machines nécessitant le moins de ressources se voient attribuer 3 Go de stockage, ce qui est suffisant pour assurer leur bon fonctionnement sans surconsommation d'espace.
- Les machines les plus sollicitées, qui traitent des volumes de données plus importants ou exécutent des processus plus intensifs, bénéficient quant à elles de 5 Go de stockage afin de garantir des performances optimales.

En termes de mémoire vive (RAM), chaque machine de notre réseau dispose actuellement de 1 Go. Cette allocation permet de répondre aux besoins de nos applications tout en maintenant un bon équilibre entre performance et consommation de ressources.

Nous surveillons régulièrement l'utilisation de la RAM et du stockage afin d'optimiser notre infrastructure si nécessaire et d'anticiper toute montée en charge.

# 4 Installation et Configuration des éléments de l'infrastrucure

On va détailler dans cette partie comment nous avons configuré les éléments de notre infrastructure réseau.

## 4.1 Résaux Virtuel

Nous avons créé des VXLAN pour interconnecter les hyperviseurs au sein du cluster, permettant ainsi une communication entre eux. De plus, nous avons mis en place des VNET spécifiques pour chaque hyperviseur afin de segmenter et d'optimiser la gestion du réseau virtuel, garantissant une meilleure performance et une isolation accrue des ressources.

## 4.2 Routeurs

Tout d'abord, il faut configurer les interfaces de la machine qui va servir de routeur.

Il faut configurer le fichier **/etc/network/interfaces** pour attribuer les bonnes ip et CIDR pour chaque interface. Voici un explique de chaque paramètre que l'on peut mettre :

- Une première ligne avec la façon d'on l'interface s'active.
  - **auto [nomInterface]** : Pour activer automatiquement au démarrage. Idéal pour les interfaces fixes, comme celles des serveurs.
  - **allow-hotplug [nomInterface]** : Pour activer uniquement quand elle est détectée. Idéal pour les interfaces amovibles.
- Une seconde qui définit sa configuration.
  - **iface eth0 inet manual** : Interface n'a pas de configuration IP et doit être activé à la main
  - **iface eth0 inet none** : Interface active mais sans configuration IP
  - **iface [nomInterface] inet dhcp** : Utilise le DHCP pour l'attribution d'IP, ...
  - **iface eth0 inet static** : Pour faire une configuration avec une IP statique
- Ajout de paramètres pour la configuration de IP si configuration **static** :
  - **address 192.168.100.14/24** : L'adresse IP static
  - **gateway 192.168.100.1** : Le gateway
  - **dns-nameservers 192.168.100.11 1.1.1.1** : Les DNS
  - **dns-domain it-connect.local** : Le domaine DNS
  - **metric 10** : La priorité de l'utilisation de cette interface. Plus la valeur est basse, plus la priorité est haute.
  - **up ip addr add 192.168.100.15/24 dev [nomInterface]** : Si tu veux plusieurs adresses IP sur la même interface
  - **mtu 9000** : MTU maximum
  - **post-up ip route add 192.168.200.0/24 via 192.168.100.10** : Si la machine doit accéder à un réseau via une passerelle spécifique. Tout le trafic vers 192.168.200.0/24 passera via 192.168.100.10.

Voici un fichier de configuration type pour un routeur avec plusieurs interface :

**Fichier :** /etc/network/interfaces

```
        # Interface WAN (connectee a Internet)
        auto eth0
        iface eth0 inet dhcp
        mtu 1500  # Taille MTU standard
        # Interface LAN 1
        (reseau interne 192.168.1.0/24)
        auto eth1
        iface eth1 inet static
            address 192.168.1.1/24
            gateway 192.168.1.254
            # Facultatif, utilise uniquement si ce reseau doit
            sortir par une autre route
            dns-nameservers 192.168.1.1 8.8.8.8
            dns-domain lan1.local
            mtu 9000  # Optimise pour les reseaux
            locaux rapides

        # Interface LAN 2
        (reseau interne 10.10.0.0/24)
        auto eth2
        iface eth2 inet static
            address 10.10.0.1/24
            mtu 9000  # Optimise pour un second
            reseau local

        # Activation du routage entre les reseaux
    post-up echo 1 > /proc/sys/net/ipv4/ip_forward

        # Ajout de routes pour permettre aux
        deux reseaux de communiquer entre eux
        # Ajout de routes pour permettre aux
        deux reseaux de communiquer entre eux
    post-up ip route add 192.168.1.0/24
    via 192.168.1.1 dev eth1
    post-up ip route add 10.10.0.0/24
    via 10.10.0.1 dev eth2
```

Une fois le fichier configurer, il redémarrer le service avec

```
$ sudo systemctl restart networking.service
```

Enfin, il faut activer l'interface avec

```
$ ifup [nomInterface]
```

Ensuite il faut configurer les routes entre les réseaux

Par défaut, une machine Linux ne fait pas passer n'importe quel paquet comme doit le faire un routeur. On doit donc activer cette fonctionnalité qui est sous la forme d'un option dans le fichier /**etc**/**sysctl.conf**, on devra y chercher la ligne suivante afin de la dé-commenter :

**Fichier :** /etc/sysctl.conf

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

Cette option active donc le **forwarding** (le "**relayage**" des paquets) d'une interface à une autre ou plus précisément d'un réseau à un autre. On pourra ensuite reloader notre sysctl :

```
$ sysctl -p /etc/sysctl.conf
```

Mettre en place le NAT (fichier nft a executer)

```
echo "1" > /proc/sys/net/ipv4/ip_forward
nft add table filtrage_nat
nft 'add chain filtrage_nat prerouting { type nat hook prerouting priority 0 ; }'
nft 'add chain filtrage_nat postrouting { type nat hook postrouting priority 0 ; }'
nft add rule filtrage_nat postrouting masquerade
```

## 4.3 Serveur DHCP

- Creer une VM avec une @IP statique car c'est lui qui donne les IPs
- Installer kea :

  ```
  # apt install kea-dhcp4-server
  ```

- Reset la config :

  ```
  # mv /etc/kea/kea-dhcp4.conf /etc/kea/kea-dhcp4.conf.bkp
  ```

- Ajouter les configuration du serveur dans le fichier **etc/kea/kea/kea-dhcp4.conf**
- Fichier de configuration type

**Fichier :** `/etc/kea/kea-dhcp4.conf.bkp`

```
{
    "Dhcp4": {
        "interfaces-config": {
            "interfaces": [
                "ens33"
            ]
        },
        "valid-lifetime": 691200,
        "renew-timer": 345600,
        "rebind-timer": 604800,
        "authoritative": true,
        "lease-database": {
            "type": "memfile",
            "persist": true,
            "name": "/var/lib/kea/kea-leases4.csv",
            "lfc-interval": 3600
        },
        "subnet4": [
            {
                "subnet": "192.168.14.0/24",
                "pools": [
                    {
                        "pool": "192.168.14.100
                        - 192.168.14.120"
                    }
                ],
                "option-data": [
                    {
                        "name": "domain-name-servers",
                        "data": "192.168.14.201"
                    },
                    {
                        "name": "domain-search",
                        "data": "it-connect.local"
                    },
                    {
                        "name": "routers",
                        "data": "192.168.14.2"
                    }
                ]
            }
        ]
    }
}
```

## 4.4 Serveur DNS

Marche a suivre :

1. Nous allons créer 2 serveur DNS, un serveur Externe, qui va être accessible depuis l'exterieur, et un serveur DNS Interne, qui va être accessible uniquement depuis l'interieur, ca sera utile pour le wiki.

2. On va commencer par créer le DNS privé puis on le clonera pour en faire un DNS public, et il faudra juste supprimer les alias créés pour le wiki et supprimer l'ACL "lan" et l'option lan dans allow-query (voir suite)

### 4.4.1 Installation de BIND9

On installe Bind9 avec la commande suivante :

```
# apt install bind9 dnsutils
```

### 4.4.2 Configuration des options DNS

nous allons d'abord copier la config pour pouvoir la rétablir facilement en cas d'erreur :

```
# cd /etc/bind
```

```
# cp named.conf.options named.conf.options.bkp
```

```
# cp named.conf.local named.conf.local.bkp
```

On va ensuite modifier le fichier de config /etc/bind/named.conf.options et changer le 0.0.0.0 dans le champ forwarders par le DNS de l'UGA : **IP:** 152.77.1.22
Etant donné qu'on créée actuelleùent le DNS privé, on va déclarer une ACL Lan, pour que seul le reseau local puisse y avoir accès (config a mettre avant le champ "option") :

**Fichier :** /etc/bind/named.conf

```
acl "lan" {
        192.168.110.0/24;
        192.168.110.0/24;
        192.168.110.0/24;
        localhost;
        localnets;
};
options{
[...]
        allow-query { lan; };
};
```

### 4.4.3 Configuration de zone

Nous allons desormais créer la zone bav4. Dans /etc/bind/named.conf.local ,on ajoute donc la zone suivante:

**Fichier :** /etc/bind/named.conf.local

```
zone "bav4.local" {
    type master;
    file "/etc/bind/db.bav4.local";
    allow-update { none; };
};
```

Puis nous allons dupliquer la db.local en l'appellant db.bav4.local pour pouvoir configurer la zone :

```
# cp /etc/bind/db.local /etc/bind/db.bav4.local
```

Puis, dans cdb.bav4.local, nous allons mettre en place la config suivante :

**Fichier :** /etc/bind/named.conf.local

```
$TTL    604800
@       IN      SOA     srv-dns.bav4.local.  root.bav4.local. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@                IN       NS       srv-dns.bav4.local.
srv-dns          IN       A        192.168.110.22
```

Une fois cela fait, on restart bind9 et notre serveur DNS est opérationnel :

```
# systemctl restart bind9
```

```
# systemctl enable named.service
```