

**BTS Services informatiques aux organisations - SISR****Session 2022****E4 – Support et mise à disposition de services informatiques****Coefficient 4****DESCRIPTION DE LA REALISATION PROFESSIONNELLE****NOM et prénom du candidat : LEDUC Quentin****N° candidat : 02145639104****Contexte de la réalisation professionnelle**

Au sein de Scani les mails sont gérés uniquement par Bruno. Malheureusement personne d'autre ne s'y connaît pour gérer le serveur mail. Par conséquent et pour que plus de gens touche au serveur mail nous voulons changer notre serveur pour qu'il soit plus facile à prendre en main avec une interfaces Web pour tous les coopérateurs de Scani.

**Intitulé de la réalisation professionnelle**

Installation et configuration d'un serveur Mailcow

**Période de réalisation : DU 21/01/22 AU 22/01/22****Lieu : Joigny****Modalité : Individuelle****Principale(s) activité(s) concernée(s) :**

- *Mettre en place et vérifier les niveaux d'habilitation associés à un service*
- *Réaliser les tests d'intégration et d'acceptation d'un service*
- *Déployer un service*
- *Développer son projet professionnel*
- *Mettre en place son environnement d'apprentissage personnel*

**Conditions de réalisation**

- **Ressources présentes** Nous avons un serveur mail avec Dovecot et postfix sur un FreeBSD
- **Résultats attendus** Installer un serveur mail avec une interfaces Web : Mailcow
- **Durée de réalisation** 1H

**Modalités d'accès à cette réalisation professionnelle.**Site internet : [www.netwaze.fr](http://www.netwaze.fr)

Aller dans « Réalisations Professionnelles » Mot de passe : Mr.Robot

**Partie 1 – Procédure de mise en œuvre.****PRÉREQUIS MATÉRIEL**

- RockyLinux - Debian
- Docker

# PRÉREQUIS

Avant de commencer nous devons mettre à jour notre système.

```
$ dnf update --refresh
```

Ainsi notre système à jour nous pouvons le redémarrer avant de commencer l'installation.

## INSTALLATION DE DOCKER ET DOCKER-COMPOSE

Pour faire fonctionner Mailcow il doit avoir Docker car son créateur a décidé de ne plus maintenir officiellement Mailcow en barre métal. On va aussi avoir besoin de Docker-Compose. Pour celui-ci il sert à faire dialoguer et a créer des volumes persistants entre tout les Dockers. En effet Mailcow est constitué de 19 conteneurs pour fonctionner.

**Disclaimer** : Pour des raisons de sécurité et de confidentialité nous allons reproduire mon installation et une configuration minimal de Mailcow comparer à celle en production. Aussi pour reproduire cette installation nous allons la faire dans une machine virtuelle. Des configurations peuvent être modifiées.

Pour installer Docker il faut installer le dépôt officiel de Docker.

```
$ dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo
```

Et on installe la dernière version de Docker : `dnf install docker-ce --nobest -y`

On peut maintenant le démarrer et l'activer a chaque démarrage du système.

```
$ systemctl start docker
```

```
$ systemctl enable docker
```

```
[root@Rocky-Mailcow kiu]# systemctl start docker
[root@Rocky-Mailcow kiu]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@Rocky-Mailcow kiu]#
```

Pour vérifier la version de notre Docker : `$ docker --version`

```
[root@Rocky-Mailcow kiu]# docker --version
Docker version 20.10.9, build c2ea9bc
[root@Rocky-Mailcow kiu]#
```

On passe maintenant à l'installation de notre Docker-Compose. A l'heure actuelle la version 2.0.1 de Docker-Compose est sortie mais non compatible avec Mailcow. Pour ce faire j'ai ouvert un nouveau Git sur GitHub avec la version 1.29.2.

Pour récupérer mon fichier on va avoir besoin de Git

```
$ dnf install git
```

On télécharge ensuite notre fichier en le mettant dans `usr/local/bin`

```
$ git clone https://github.com/netwaze/Docker-compose-V1.29.2.git
```

On renomme et on met notre Docker-Compose dans `/usr/local/bin/`

```
$ mv Docker-compose-V1.29.2/docker-compose-Linux-x86_64 Docker-compose-V1.29.2/docker-compose && mv Docker-compose-V1.29.2/docker-compose /usr/local/bin/
```

On lui donne ensuite les droits d'exécution

```
$ chmod +x /usr/local/bin/docker-compose
```

Nous pouvons maintenant essayer si docker compose fonctionne bien en lui demandant sa version.

```
$ docker-compose --version
```

```
[root@Rocky-Mailcow bin]# docker-compose --version
docker-compose version 1.29.2, build 5becea4c
[root@Rocky-Mailcow bin]#
```

## INSTALLATION DE MAILCOW

Pour installer Mailcow on va devoir ce rendre dans le dossier `/opt` et de Git Mailcow

```
$ cd /opt
```

```
$ sudo git clone https://github.com/mailcow/mailcow-dockerized
```

```
[root@Rocky-Mailcow opt]# sudo git clone https://github.com/mailcow/mailcow-dockerized
Clonage dans 'mailcow-dockerized'...
remote: Enumerating objects: 38104, done.
remote: Counting objects: 100% (5532/5532), done.
remote: Compressing objects: 100% (2822/2822), done.
remote: Total 38104 (delta 2891), reused 5206 (delta 2624), pack-reused 32572
Réception d'objets: 100% (38104/38104), 27.82 Mio | 20.14 Mio/s, fait.
Résolution des deltas: 100% (23834/23834), fait.
[root@Rocky-Mailcow opt]#
```

Une fois Mailcow télécharger nous devons générer le fichier de configuration de Mailcow

```
$ ./generate_config.sh
```

Nous devons répondre au question poser par le script. Il nous faut un FQDN :

[mailcow.netwaze.fr](mailto:mailcow.netwaze.fr) et une tmezone qui doit récupérer automatiquement. Puis il crée un certificat RSA pour le SSL.

```
[root@Rocky-Mailcow mailcow-dockerized]# ./generate_config.sh
Press enter to confirm the detected value '[value]' where applicable or enter a custom value.
Mail server hostname (FQDN) - this is not your mail domain, but your mail servers hostname: mailcow.netwaze.fr
Timezone [Europe/Paris]:
Generating snake-oil certificate...
Generating a RSA private key
.....++++
.....++++
writing new private key to 'data/assets/ssl-example/key.pem'
-----
Copying snake-oil certificate...
[root@Rocky-Mailcow mailcow-dockerized]#
```

Nous pouvons maintenant modifier dans notre Mailcow.conf comment va fonctionner notre serveur mail.

Dans SKIP\_LETS\_ENCRYPT mettre y a la place de n et il faut décommenter WATCHDOG\_NOTIFY\_EMAIL et mettre son mail.

On peut sauvegarder et quitter et lancer le téléchargement des images pour que Mailcow fonctionne.

```
$ docker-compose pull
```

```
[root@Rocky-Mailcow mailcow-dockerized]# docker-compose pull
Pulling unbound-mailcow ... done
Pulling mysql-mailcow ... done
Pulling redis-mailcow ... done
Pulling clamd-mailcow ... done
Pulling php-fpm-mailcow ... done
Pulling sogo-mailcow ... done
Pulling dovecot-mailcow ... done
Pulling rspamd-mailcow ... done
Pulling postfix-mailcow ... done
Pulling memcached-mailcow ... done
Pulling nginx-mailcow ... done
Pulling acme-mailcow ... done
Pulling netfilter-mailcow ... done
Pulling watchdog-mailcow ... done
Pulling dockerapi-mailcow ... done
Pulling solr-mailcow ... done
Pulling olefy-mailcow ... done
Pulling ofelia-mailcow ... done
Pulling ipv6nat-mailcow ... done
[root@Rocky-Mailcow mailcow-dockerized]#
```

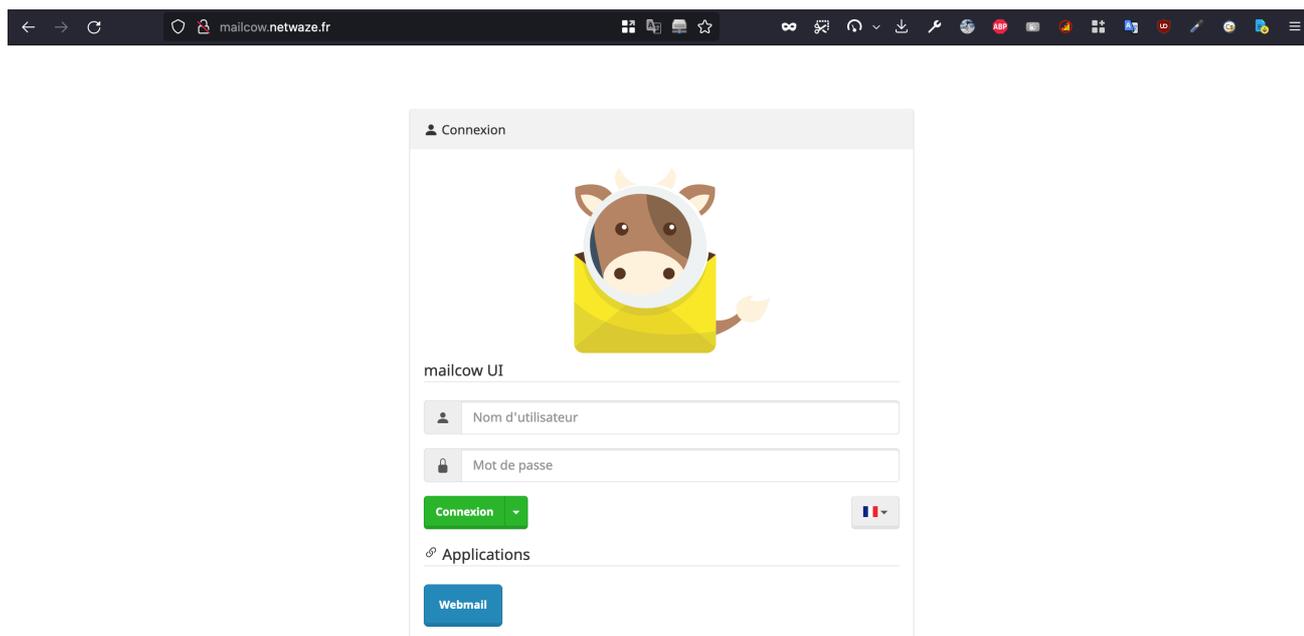
On peut maintenant lancer Mailcow

```
$ docker-compose up -d
```

(l'option -d sert a se détacher des codes de Mailcow. Sans le -d nous pouvons avoir accès au debug de la machine)

```
[root@Rocky-Mailcow mailcow-dockerized]# docker-compose up -d
Creating network "mailcowdockerized_mailcow-network" with driver "bridge"
Creating volume "mailcowdockerized_vmail-vol-1" with default driver
Creating volume "mailcowdockerized_vmail-index-vol-1" with default driver
Creating volume "mailcowdockerized_mysql-vol-1" with default driver
Creating volume "mailcowdockerized_mysql-socket-vol-1" with default driver
Creating volume "mailcowdockerized_redis-vol-1" with default driver
Creating volume "mailcowdockerized_rspamd-vol-1" with default driver
Creating volume "mailcowdockerized_solr-vol-1" with default driver
Creating volume "mailcowdockerized_postfix-vol-1" with default driver
Creating volume "mailcowdockerized_crypt-vol-1" with default driver
Creating volume "mailcowdockerized_sogo-web-vol-1" with default driver
Creating volume "mailcowdockerized_sogo-userdata-backup-vol-1" with default driver
Creating mailcowdockerized_olefy-mailcow_1 ... done
Creating mailcowdockerized_solr-mailcow_1 ... done
Creating mailcowdockerized_watchdog-mailcow_1 ... done
Creating mailcowdockerized_dockerapi-mailcow_1 ... done
Creating mailcowdockerized_memcached-mailcow_1 ... done
Creating mailcowdockerized_redis-mailcow_1 ... done
Creating mailcowdockerized_unbound-mailcow_1 ... done
Creating mailcowdockerized_clamd-mailcow_1 ... done
Creating mailcowdockerized_sogo-mailcow_1 ... done
Creating mailcowdockerized_php-fpm-mailcow_1 ... done
Creating mailcowdockerized_mysql-mailcow_1 ... done
Creating mailcowdockerized_nginx-mailcow_1 ... done
Creating mailcowdockerized_dovecot-mailcow_1 ... done
Creating mailcowdockerized_postfix-mailcow_1 ... done
Creating mailcowdockerized_acme-mailcow_1 ... done
Creating mailcowdockerized_rspamd-mailcow_1 ... done
Creating mailcowdockerized_netfilter-mailcow_1 ... done
Creating mailcowdockerized_ofelia-mailcow_1 ... done
Creating mailcowdockerized_ipv6nat-mailcow_1 ... done
[root@Rocky-Mailcow mailcow-dockerized]#
```

On peut voir que toutes nos machines sont démarrées nous pouvons maintenant nous rendre sur notre navigateur internet et mettre l'URL de notre Mailcow.



L'utilisateur et le mot de passe de Mailcow par défaut est : admin / moohoo

## CONFIGURATION DE MAILCOW ET CRÉATION DE BOITE MAIL

Nous devons mettre un nom de domaine pour pouvoir créer des boîtes mails.

Pour ce faire nous devons aller dans configuration puis configuration du courrier et ajouter un domaine.

Pour nos tests nous allons mettre local.lan nous pouvons laisser le reste de la configuration par défaut mais à changer pour la production. Pour enregistrer notre domaine nous pouvons cliquer sur « Ajouter le domaine et redémarrer SOGo »

### Ajouter un domaine

domaine	<input type="text" value="local.lan"/>
Description	<input type="text"/>
Nombre maximal d'alias	<input type="text" value="400"/>
Nombre maximal de boîtes	<input type="text" value="10"/>
Quota boîte mail par défaut	<input type="text" value="3072"/>
Quota max par boîte (Mo)	<input type="text" value="10240"/>
Quota total du domaine (Mo)	<input type="text" value="10240"/>

Liste d'adresses globale (GAL)  
La liste d'adresses globale (GAL) contient tous les objets d'un domaine et ne peut pas être éditée par un utilisateur. Les informations libres/occupées dans SOGo sont manquantes si elles sont désactivées! **Redémarrer SOGo pour appliquer les modifications.**

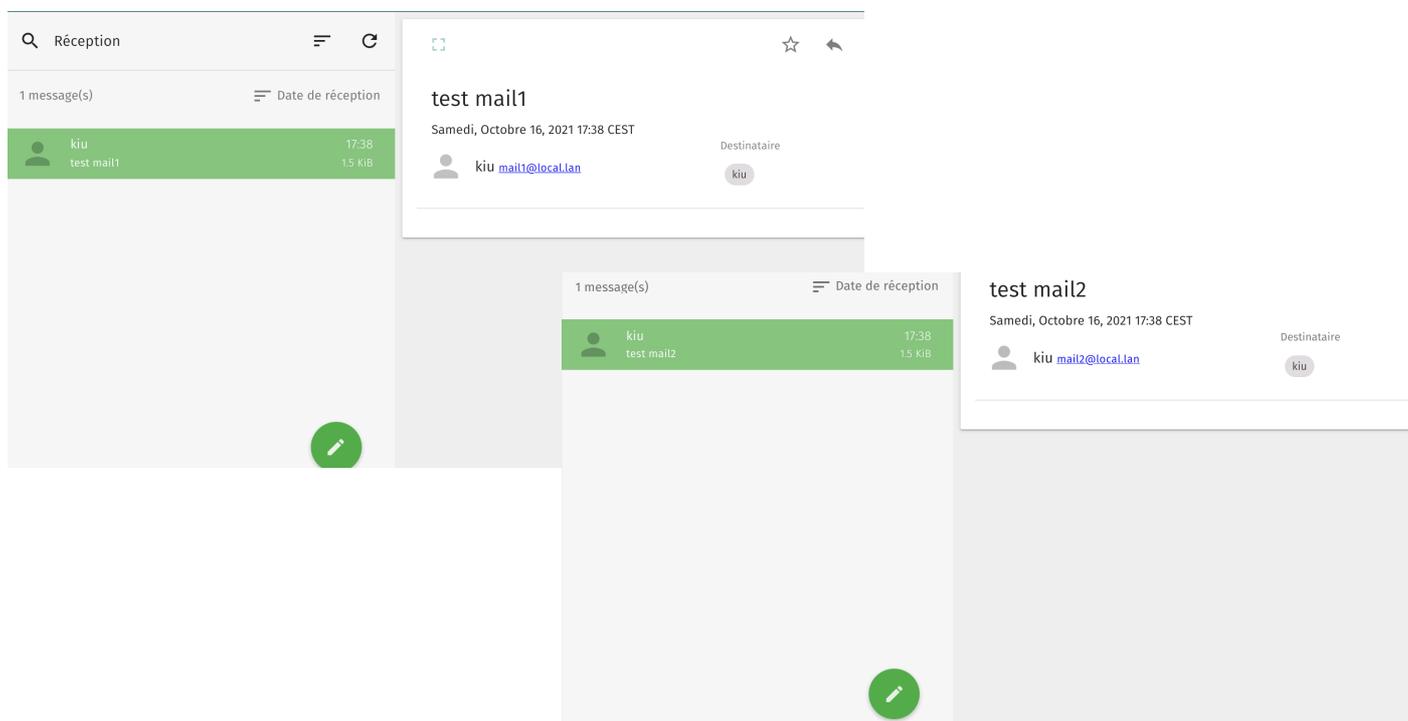
Actif

Dans le deuxième onglet Boîte mail nous allons créer 2 boîtes mail : mail1.local.lan et mail2.local.lan

Nom d'utilisateur	Quota	Dernière connexion mail	Utilisé (%)	Message #
mail1@local.lan	0 B / 100.0 MIB	IMAP @ x POP3 @ x SMTP @ x	0%	0
mail2@local.lan	0 B / 100.0 MIB	IMAP @ x POP3 @ x SMTP @ x	0%	0

## Partie 2 – Validation.

On va avec 2 navigateurs se connecter au webmail SOGo avec nos identifiants et envoyer des mails de test entre nos 2 utilisateurs.



On voit que nos 2 mails sont bien partis et on bien étaient reçus par nos 2 utilisateurs. Notre serveur de mail fonctionne.

Pour envoyer un mail vers l'extérieur, nous devons utiliser un certificat SSL.

### **Partie 3 – Veille technologique.**

Nous aurions pu utiliser Exchange avec un Windows Server ou hMail et Mail-in-a-box