Rendu de Projet

Bases de Données

Paul VEROT, 20212888
Master 1 CNS-SR, 2025

Semestre 7

UNIVERSITÉ ÉVRY
PARIS-SACLAY

Département d'informatique
univ-evry.fr

Titre:
Rendu de Projet

Thème:
Bases de Données

Groupe:
Master 1 CNS-SR

Participants:
Paul VEROT, 20212888

Email:
pauljeanlouisverot@protonmail.com
paul.verot@etud.univ-evry.fr

Superviseur:
Damien PLOIX

Nombre de pages:
6

Dernier changement:
28-12-2025

Résumé:

Mise en place d'une machine virtuelle Oracle Linux 10, installation et configuration de bases de données Oracle SQL et MongoDB, puis interaction avec ces données.

Mise en Place

J'ai commencé par suivre les instructions du document fourni : J'ai installé Oracle Linux 10 sur VMware avec les partitions suivantes :

```
pverot@E20212888:~$ lsblk
NAME         MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda            8:0    0  50G  0 disk
├─sda1         8:1    0   1M  0 part
├─sda2         8:2    0   1G  0 part /boot
└─sda3         8:3    0  49G  0 part
  ├─ol-root 252:0    0  44G  0 lvm  /var/lib/containers/storage/overlay
  │                                 /
  └─ol-swap 252:1    0   5G  0 lvm  [SWAP]
sr0           11:0    1   8G  0 rom  /run/media/pverot/OL-10-0-0-BaseOS-x86_64
```

Figure 0.1: lsblk

Avec les ressources système suivantes :

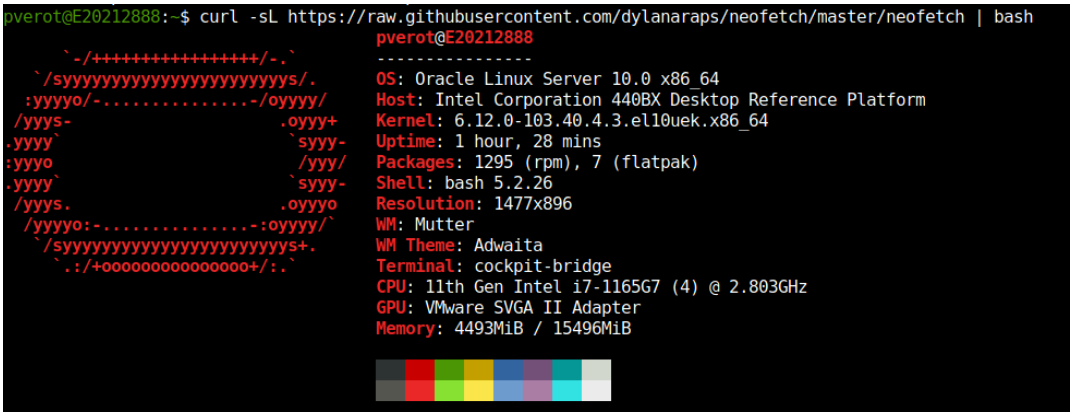


Figure 0.2: neofetch

Overview de Cockpit:

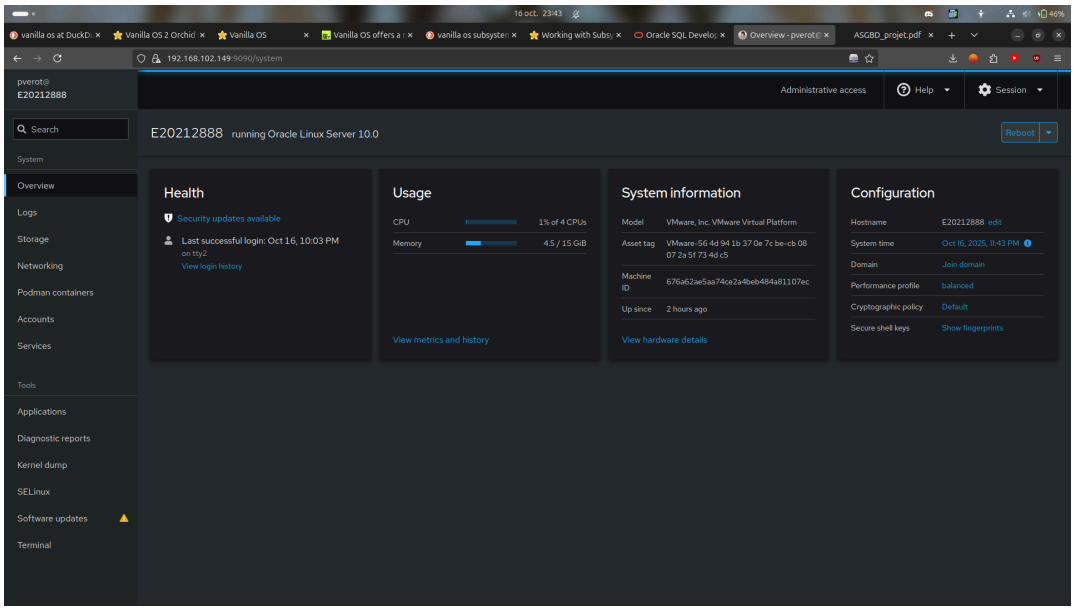


Figure 0.3:

J'ai ensuite installé les conteneurs en suivant les commandes indiquées.
Comme mentioné en cours, une ligne du fichier de configuration devait être modifiée :

```
podman run -d --network dbnet --name E20212888 \
  -e ORACLE_PWD=motdepasse -p 1521:1521 \
  -v ~/partage:/tmp/partage \
  container-registry.oracle.com/database/free:latest
```

Devient :

```
podman run -d --network dbnet --name E20212888 \
  -e ORACLE_PWD=motdepasse -p 1521:1521 \
  -v ~/partage:/tmp/partage:z \
  container-registry.oracle.com/database/free:latest
```

Le flag :z indique à SELinux que le contenu du dossier sera partagé entre plusieurs conteneurs.[1]

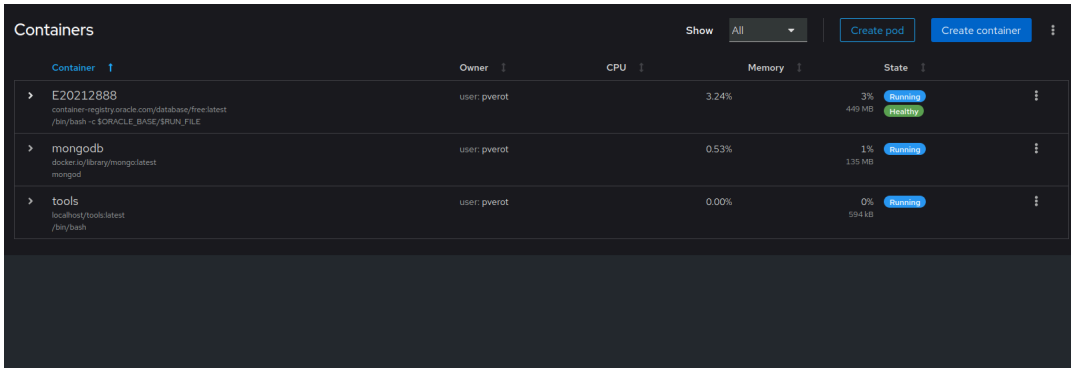
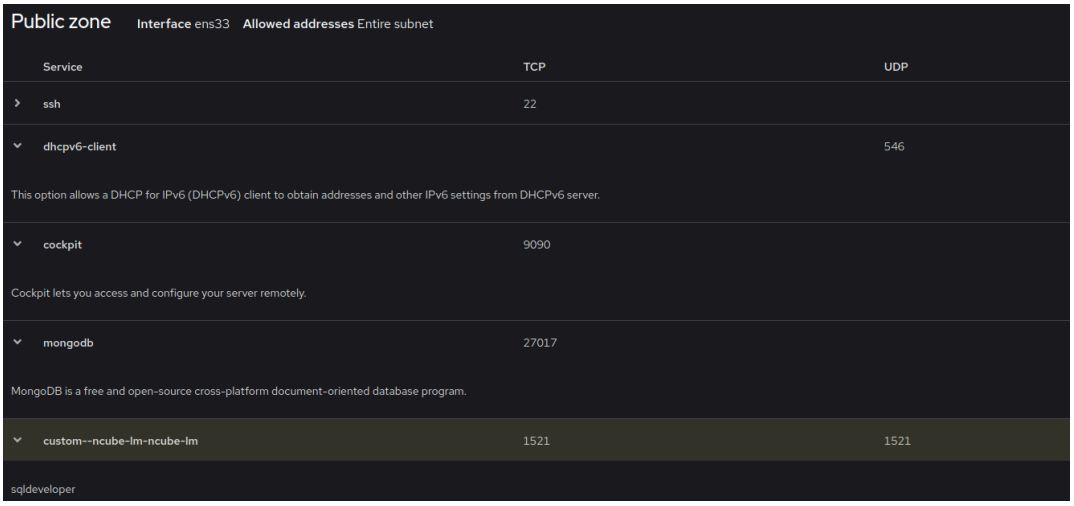


Figure 0.4: podman overview

J'ai ensuite configuré le pare-feu pour permettre l'accès aux services depuis l'interface web de Cockpit.

[1] https://docs.docker.com/engine/storage/bind-mounts/#configure-bind-propagation

- MongoDB ⟶ port 20017
- OracleSQL ⟶ port 1521



J'ai ensuite installé sur ma machine hôte SQL Developer et MongoDB Compass. MongoDB Compass n'étant pas disponible au format .deb, j'ai dû convertir le .rpm avec alien : `alien --to-deb mongodb-compass*.rpm`, que j'ai installé avec `dpkg -i mongodb-compass*.deb` Je n'ai pas pu convertir le fichier .rpm de SQL Developer. Je suis donc passé par DistroBox, qui permet de créer un environnement Oracle Linux virtualisé, permettant, entre autres, d'installer simplement des applications qui ne sont pas nativement disponibles pour ma distribution (Ubuntu 25.04)

```
distrobox create --name oracle-dev --image oraclelinux:9
distrobox enter oracle-dev
sudo dnf install -y java-17-openjdk
sudo dnf install -y /home/paul/Downloads/sqldeveloper-*.rpm
```

J'ai ensuite pu me connecter à la base de données.

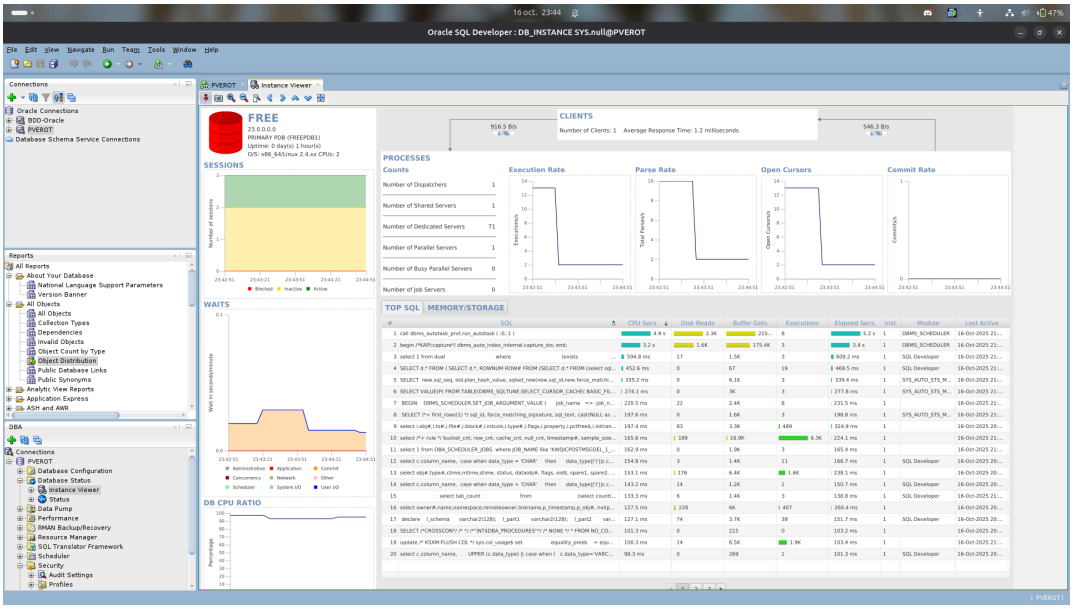


Figure 0.6: Visualisation d'une instance SQL Developer

Export des données vers MongoDB

Pour exporter des données vers MongoDB, on a besoin d'un script. Par exemple, pour exporter une liste des adresses, on peut utiliser le script bash suivant :

```
cat << 'END_OF_SCRIPT' > import2.sh
#!/bin/bash
(/usr/lib/sqlcl/bin/sql -s HR/paul@//E20212888:1521/freepdb1 <<EOF
set pagesize 0
set trimspool on
set headsep off
set null '0'
set echo off
set feedback off
set linesize 1000

SELECT JSON_OBJECT( 'adresses' value JSON_ARRAYAGG( JSON_OBJECT(
    'address'     VALUE l.STREET_ADDRESS,
    'ville'       VALUE l.CITY,
    'code postal' VALUE l.POSTAL_CODE
)))
FROM LOCATIONS l;

exit;
EOF
) | mongoimport --host=mongodb --port=27017 --db HR --collection HR2 --mode
upsert
END_OF_SCRIPT
```

[2]

Ce script permet de réaliser les actions suivantes :

- se connecter à la base de données
- réduire la quantité d'informations retournées
  - set pagesize 0 désactive les sauts de ligne et les en-têtes de colonne
  - set trimspool on supprime les espaces en fin de ligne
  - set null '0' spécifie la valeur à inscrire si le champ est nul
  - set echo off désactive l'affichage de la commande exécutée
  - set feedback off masque les informations de retour à la fin des sélections
  - set linesize 1000 définit la longueur maximale des lignes affichées
- lancer une requête SQL pour obtenir les informations souhaitées (retournées au format JSON)

[2] Ce script doit être exécuté dans le conteneur *Tools* et disposer des permissions appropriées

    - adresse
    - ville
    - code postal
- importer les données dans la base Mongo en définissant une nouvelle collection

JSON_ARRAYAGG sert à agréger les données en une seule ligne JSON. Sans cela, le script renverrait autant de lignes que d'entrées dans la table. L'emploi de JSON_ARRAYAGG définit la structure des données dans MongoDB. Si cette fonction n'est pas utilisée, on obtient une structure plate avec un document par adresse. Si elle est utilisée, on obtient une structure imbriquée dans laquelle un document contient plusieurs adresses.

Sans :
Document 1 : `{ "_id": 1, "address": "123 Grand Rue", "ville": "Paris" }`
Document 2 : `{ "_id": 2, "address": "456 Rue du Four", "ville": "Nice" }`
Document 3 : `{ "_id": 3, "address": "789 Avenue de Villerbane", "ville": "Lyon" }`

Avec :
Document 1:

```
{
  "_id": 1,
  "adresses": [
     { "address": "123 Grand Rue", "ville": "Paris" },
     { "address": "456 Rue du Four", "ville": "Nice" },
     { "address": "789 Avenue de Villerbane", "ville": "Lyon" }
  ]
}
```

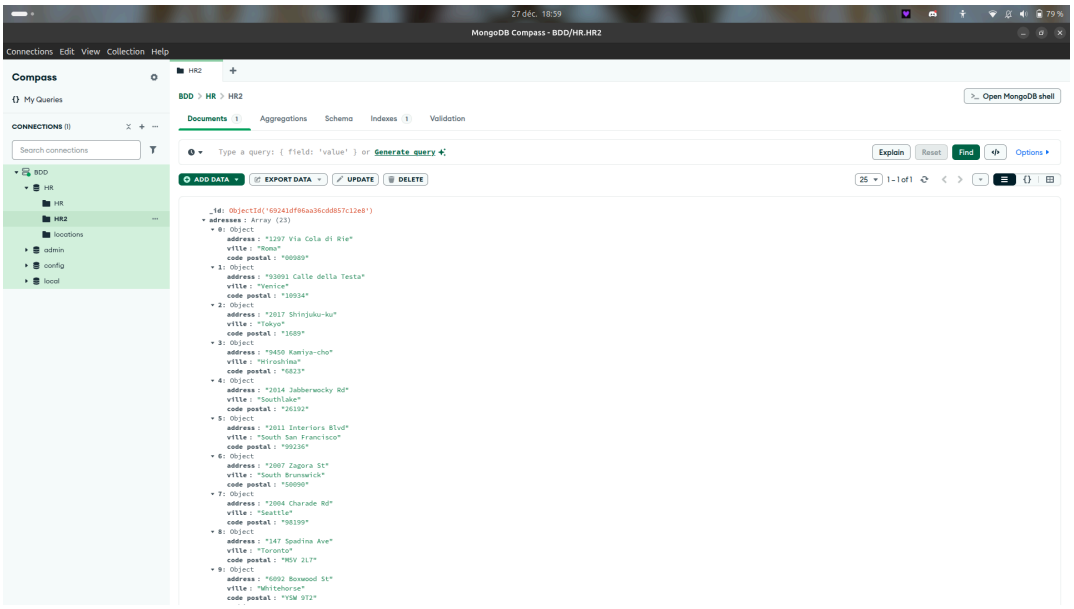


Figure 0.7: Capture dans Compass après l'import des données avec le script