

TP : Onyxia

L'objectif de cet exercice est de prendre en main la plateforme [Onyxia](#).

Exercice 1 : Configuration d'Onyxia

1. Se connecter à Onyxia et s'y inscrire avec son mail de l'Université (Paris-Saclay ou Evry)
2. Ajouter vos identifiants github aux configuration Git Onyxia dans l'onglet **Mon compte**.
 1. Se rendre sur Github (créer un compte si vous n'en n'avez pas encore)
 2. Aller dans Settings > Developer Settings > Personnel Access Token > Classic et générer un Token classic
 3. L'ajouter ensuite à Onyxia

La documentation associée est [ici](#)

Exercice 2 : Démarrer un service à partir d'un projet

Une documentation de la prise en main de la plateforme est disponible [ici](#).

1. Lancer un service **vscode-python** sans configuration et s'y connecter à partir des informations du service
2. Supprimer ensuite le service crée précédemment

Exercice 3 : Utiliser l'explorateur de données

Télécharger les fichiers **csv** suivant [usagers-2022.csv](#) et [vehicules-2022.csv](#) sur votre poste. Les importer dans l'explorateur de fichier S3 et les consulter via l'explorateur de données. L'explication du jeu de données est disponible [ici](#).

Exercice 4 : Ajouter des secrets dans vos services

La documentation associée est [ici](#)

1. Créer dossier **secrets** dans vault et y ajouter un secret **MON_SECRET=1**
2. Lancer un service **jupyter-python** et configurer ce dernier pour qu'il utilise le dossier **secrets**
3. Vérifier avec un script python que le secret existe dans votre instance. Il est disponible via une variable d'environnement.
4. Supprimer votre service ensuite

Exercice 5 : Exposer un site web via Onyxia

1. Lancer un service **jupyter-python** et changer les configurations suivantes :
 1. **role** en **admin** dans **Kubernetes**
 2. Activer **Enable custom service port** dans **Networking** avec le port **8000**

2. Créer un notebook et récupérer vos données depuis le S3 avec les commandes suivants dans une cellule jupyter :
 1. Lister les fichiers `!mc ls s3/<nom utilisateur>`
 2. Télécharger le fichier `!mc cp s3/<nom utilisateur>/<nom du fichier> ./`
3. Créer un terminal linux et exposer un service web. Se connecter ensuite au service web exposé en récupérant les informations depuis la page note du service.

Pour lancer un service web en python :

```
python3 -m http.server 8000
```

Exercice 6 : Importer des données dans PostgreSQL

1. Lancer un service **PostgreSQL**
2. Se connecter au service **jupyter-python** et importer les deux **csv** dans la base de données **postgresql**
 1. Dans un notebook, lire les **csv** avec **pandas**
 2. Installer la librairie python **psycopg2** et créer une connexion **sqlalchemy.engine.create_engine** avec les informations de connexions au format suivant **postgresql+psycopg2://user:password@hostname/defaultdb**
 3. Importer les tables avec la méthode **to_csv** du **dataframe** et la connexion **sqlalchemy**
 4. Lancer dans un terminal la commande **psql -U postgres -h <hostname> -d defaultdb** pour se connecter à la base. Vérifier avec **SELECT count(*) FROM usagers** que les données sont bien présentes

Exercice 7 : Sauvegarder votre travail

La documentation associée est [ici](#)

1. Sur Github, créer un dépôt de code vide avec le nom souhaité
2. Sauvegarder votre travail sur ce dépôt en utilisant les commandes donnée par Github

Exercice 8 : Connecter les services entre eux

1. Lancer un service **Superset**
2. Allez dans SQL Lab et vérifier que la connexion à la base Postgresql est bien présente
3. Lancer ensuite la requête suivante : **SELECT count(*) FROM usagers**