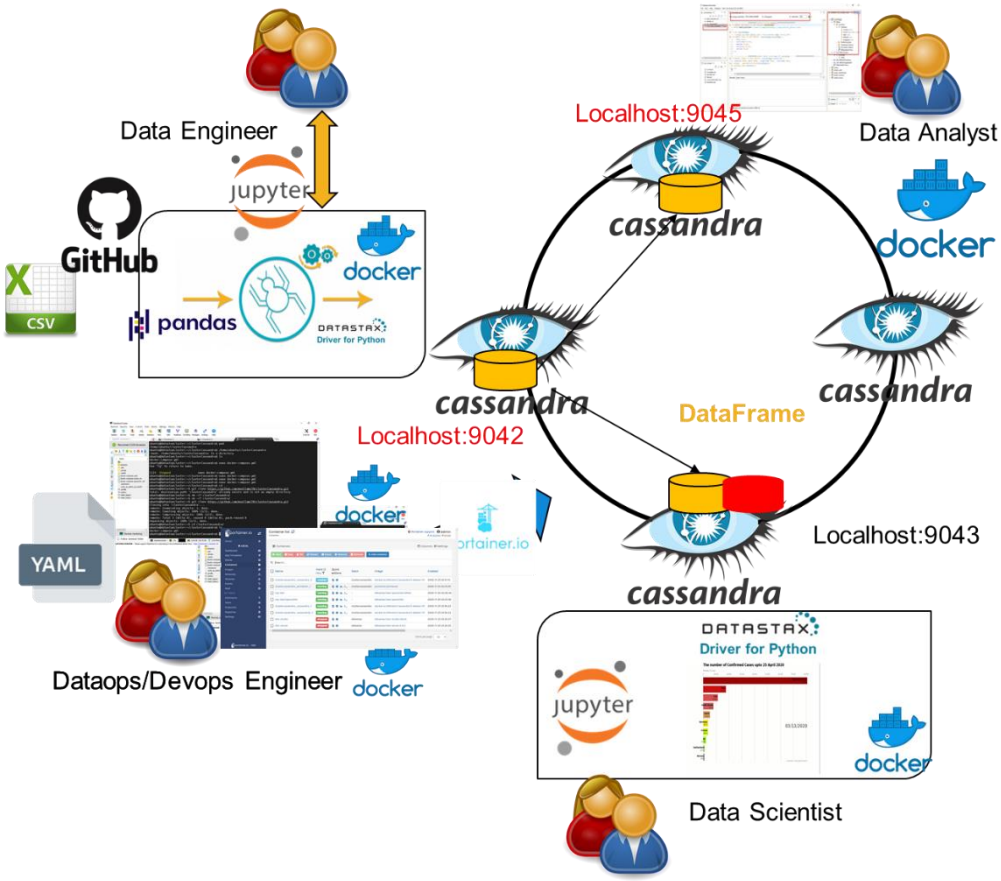
BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

Atelier 2 : Mise en place d'un Cluster Cassandra



Objectifs

Après avoir terminé cet atelier, vous serez en mesure de :

Installation de Docker et Docker Composer
Configurer le Cluster avec un fichier YAML
Lancer le Cluster
Vérifier que des nœuds du Cluster
Exécution des requêtes CQL Via DevCenter sur différents Nœuds
Ajouter et Supprimer des Nœuds de Cluster

Installation de Docker et Docker Compose

```
#Installation Docker Compose
```

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

```
sudo curl -L
"https://github.com/docker/compose/releases/download/1.27.4/docker-compose-
$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

```
#Installation Docker
```

```
sudo apt-get update

sudo apt-get install     apt-transport-https     ca-certificates     curl
gnupg-agent     software-properties-common

sudo add-apt-repository    "deb [arch=amd64]
https://download.docker.com/linux/ubuntu \

$(lsb_release -cs) \

stable"

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Récupérer et Editer le fichier de Configuration YAML via Git Clone

Lancer un Shell et Cloner le projet à partir du GitHub

```
$ cd  /home/fitec/

$ git clone https://github.com/msellamiTN/clusterCassandra.git

$ cd  clusterCassandra
```

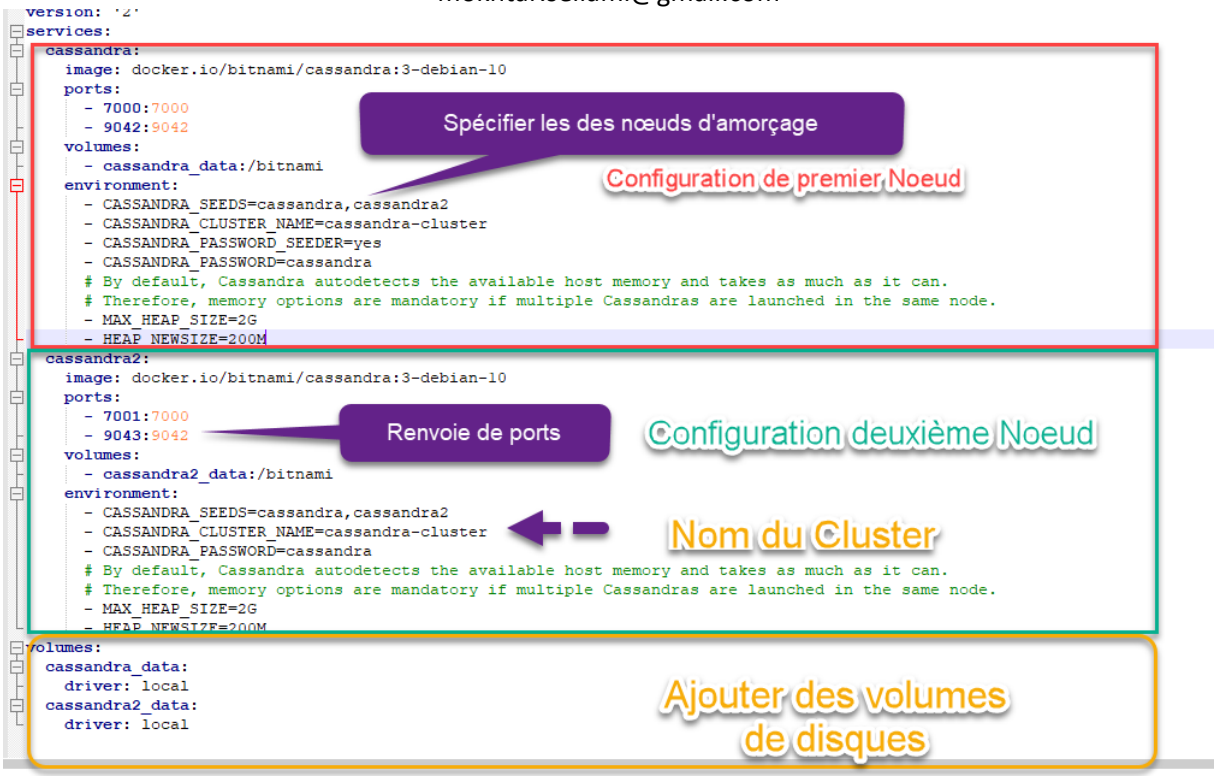
```
ubuntu@datastaxcluster:~$ git clone https://github.com/msellamiTN/clusterCassandra.git
Cloning into 'clusterCassandra'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
ubuntu@datastaxcluster:~$ cd clusterCassandra/
ubuntu@datastaxcluster:~/clusterCassandra$ ls
docker-compose.yml
ubuntu@datastaxcluster:~/clusterCassandra$
```

Editer le Fichier docker-compose.yml pour configurer votre nœuds de Cluster

```
$ gedit docker-compose.yml
```

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



Enregistrer vos fichiers et taper la commande suivante pour lancer le cluster

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

Il faut avoir un résultat similaire à l'écran suivant :

```
ubuntu@datastaxcluster:~/clusterCassandra$ sudo docker-compose up -d
Creating network "clustercassandra_default" with the default driver
Creating volume "clustercassandra_cassandra_data" with local driver
Creating volume "clustercassandra_cassandra2_data" with local driver
Pulling cassandra (docker.io/bitnami/cassandra:3-debian-10)...
3-debian-10: Pulling from bitnami/cassandra
58212c1109c5: Pull complete
96824dc62a8e: Pull complete
d1fc77bdbd0d: Pull complete
12ad5de9ad28: Pull complete
8155d7751d4a: Pull complete
8e60f03cafa5: Pull complete
e1c5595ff8dc: Pull complete
a53fce7fa23b: Pull complete
0e079e2c5e6e: Pull complete
bf028f87464b: Pull complete
ab687316fb33: Pull complete
be514576d31e: Pull complete
8aff291030a7: Pull complete
c9cdf3d6c4b5: Pull complete
Digest: sha256:8a5de434943387c0bd5ce859b6d20ac0868fdfa458a0ea29d36fab12a9902e18
Status: Downloaded newer image for bitnami/cassandra:3-debian-10
Creating clustercassandra_cassandra2_1 ... done
Creating clustercassandra_cassandra_1  ... done
ubuntu@datastaxcluster:~/clusterCassandra$ sudo docker ps
```

Vérifier que les nœuds sont correctement relancés

```
$ sudo docker ps
```

BASE DE DONNEES NO SQL Cassandra

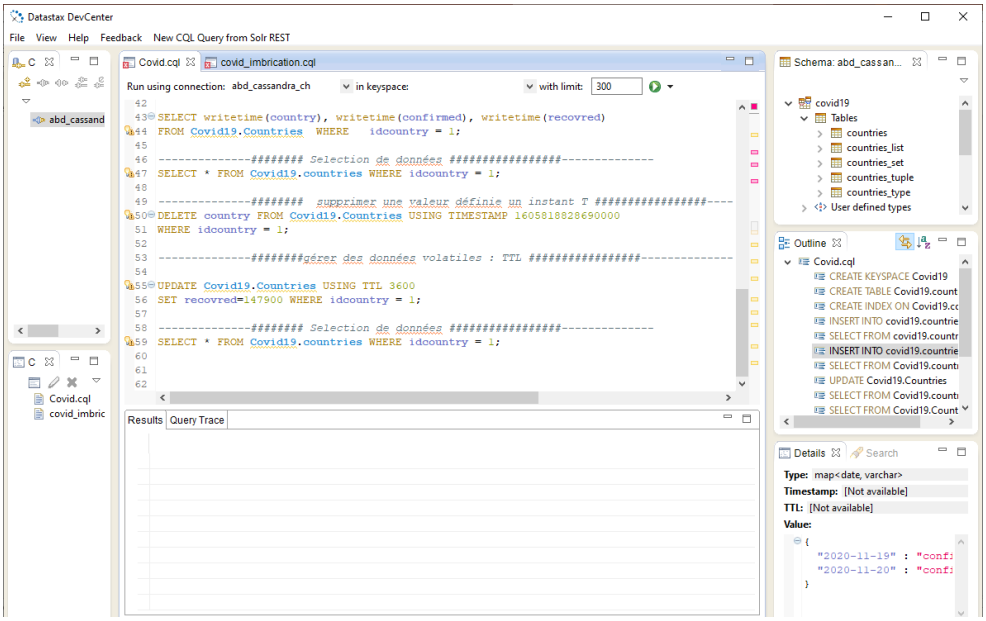
Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

```
CONTAINER ID        IMAGE                                COMMAND                  CREATED             STATUS              PORTS
                                    NAMES
46c22e7156a4        bitnami/cassandra:3-debian-10        "/opt/bitnami/script…"   54 seconds ago      Up 50 seconds       0.0.0.0:7000
->7000/tcp, 0.0.0.0:9042->9042/tcp    clustercassandra_cassandra_1
243638f04722        bitnami/cassandra:3-debian-10        "/opt/bitnami/script…"   54 seconds ago      Up 52 seconds       0.0.0.0:7001
->7000/tcp, 0.0.0.0:9043->9042/tcp    clustercassandra_cassandra2_1
```

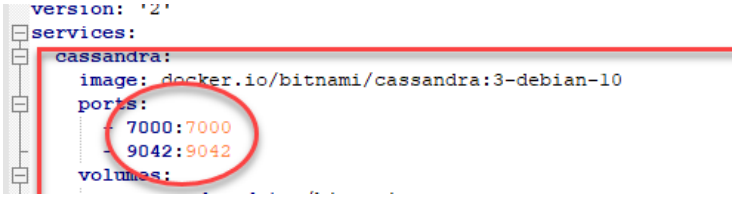
Access aux nœuds via DevCenter

Accéder au répertoire d'installation DevCenter et lancer le via la commande suivante

```
./DevCenter
```

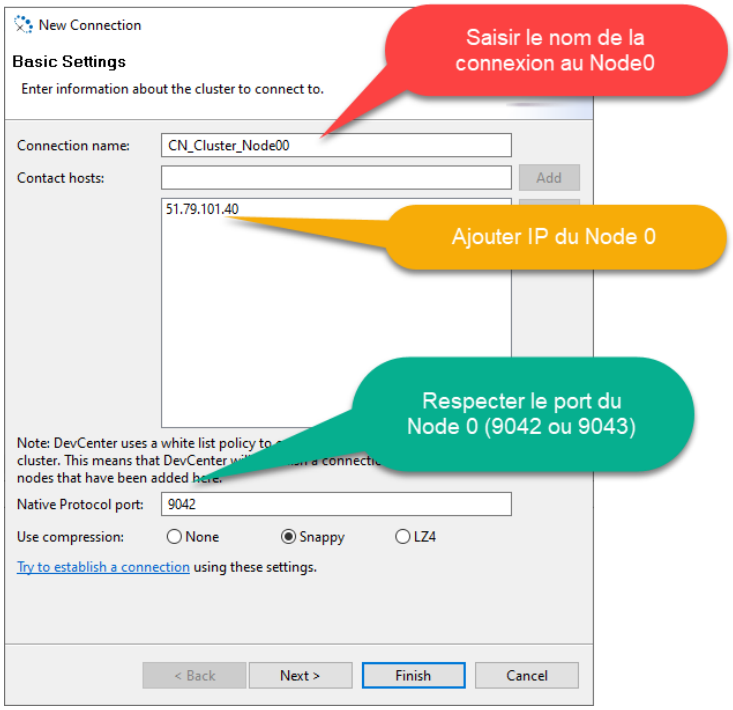


Création de la première connexion nomme « CN_Cluster_Node00 »

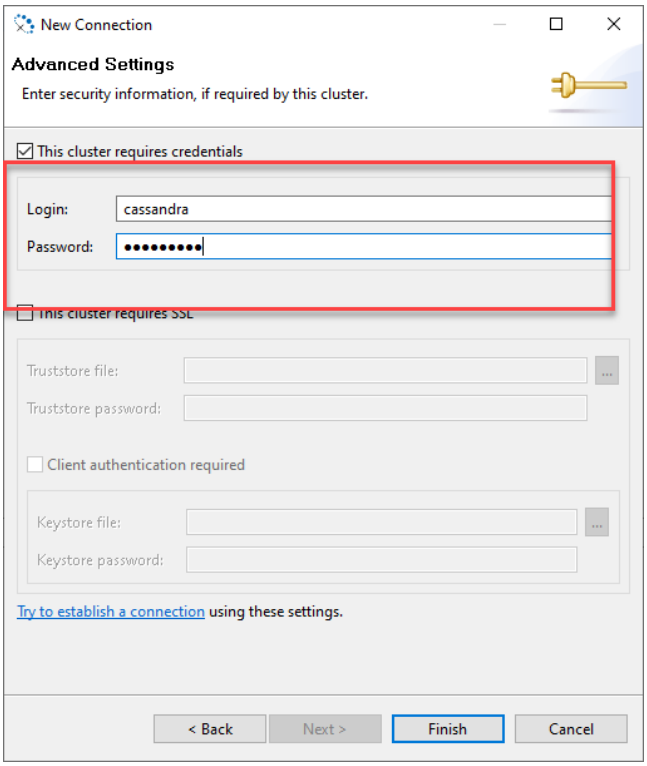


BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



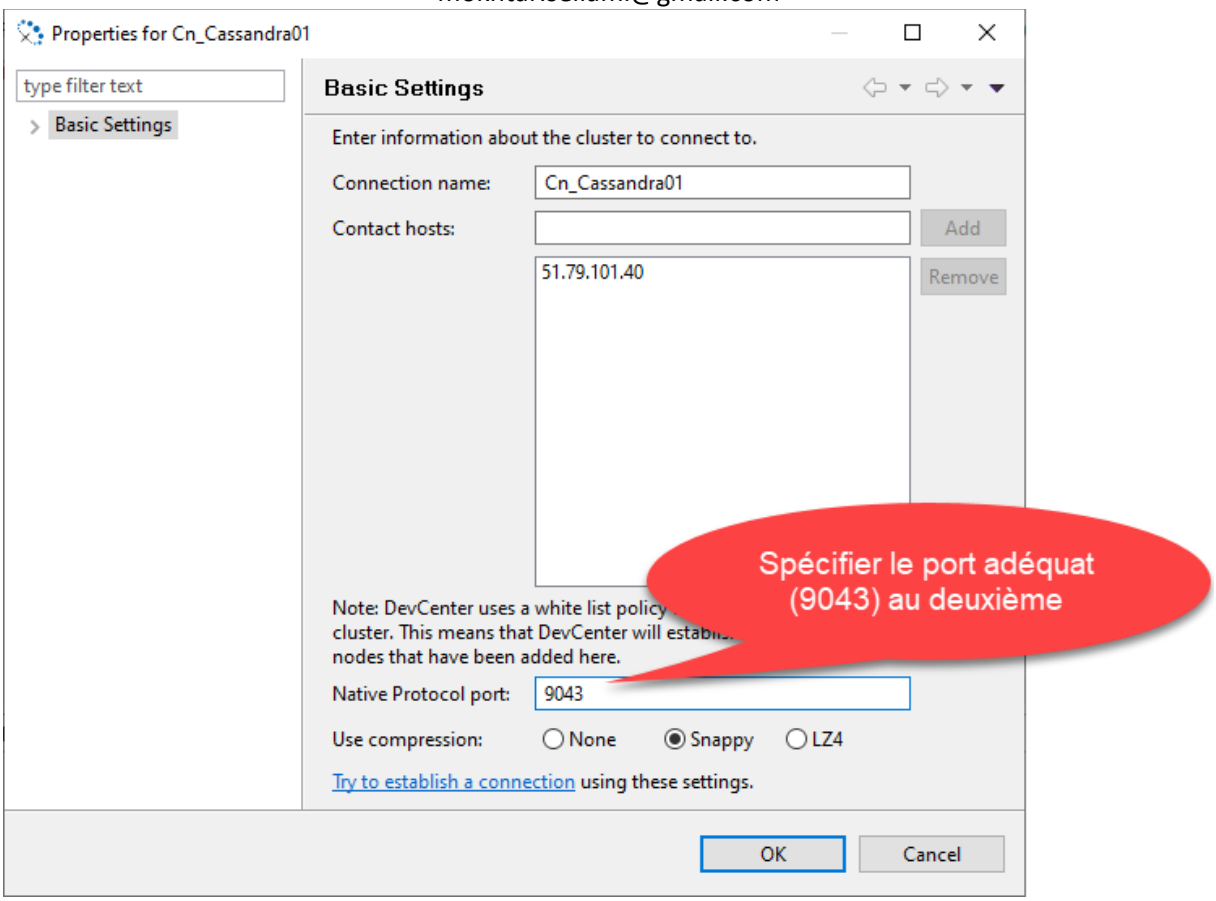
Saisir les paramètres de sécurité 'cassandra : cassandra'



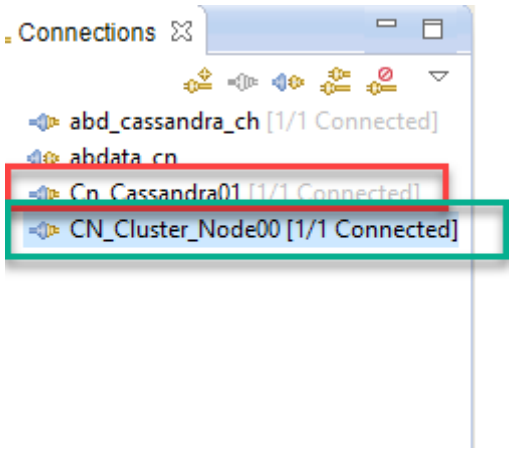
Passer à faire la même chose pour la création de deuxième connexion mais en modifiant cette fois ci le port d'accès en respectant le fichier de configuration docker-compose.yml

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



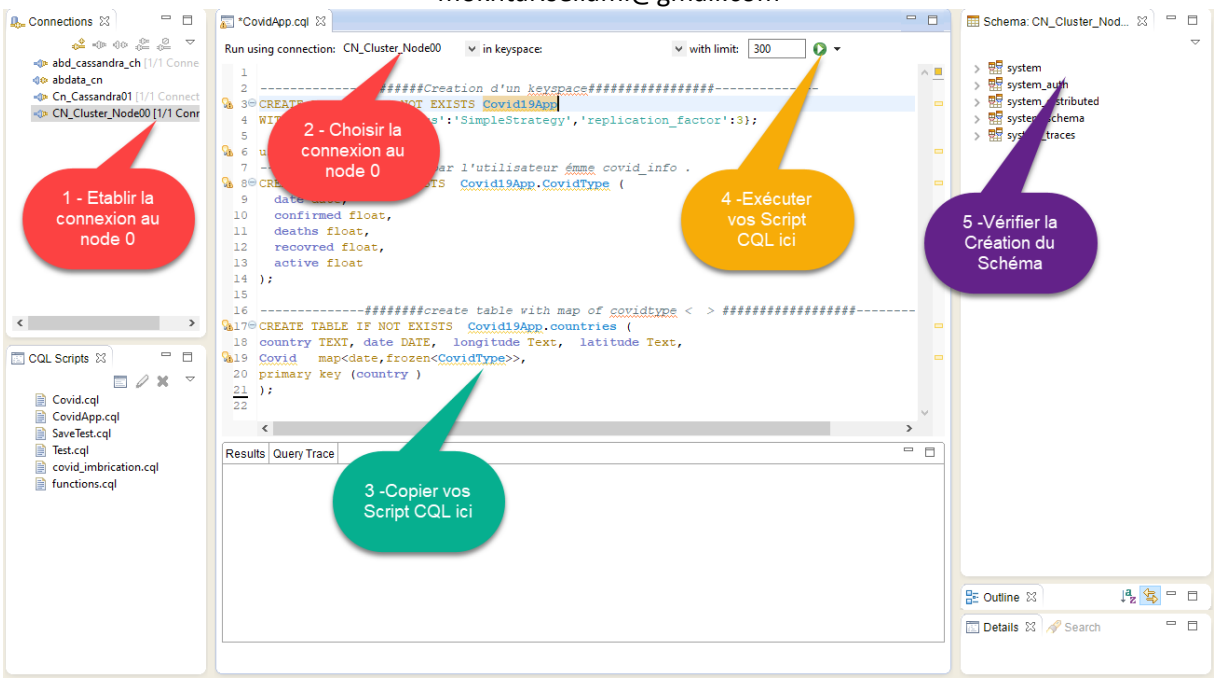
Vérifier que vois avez deux connexions actives au cluster



Passer Maintenant à créer un fichier CQL de création de keyspace et de spécifier la connexion CN_Cluster_Node00

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



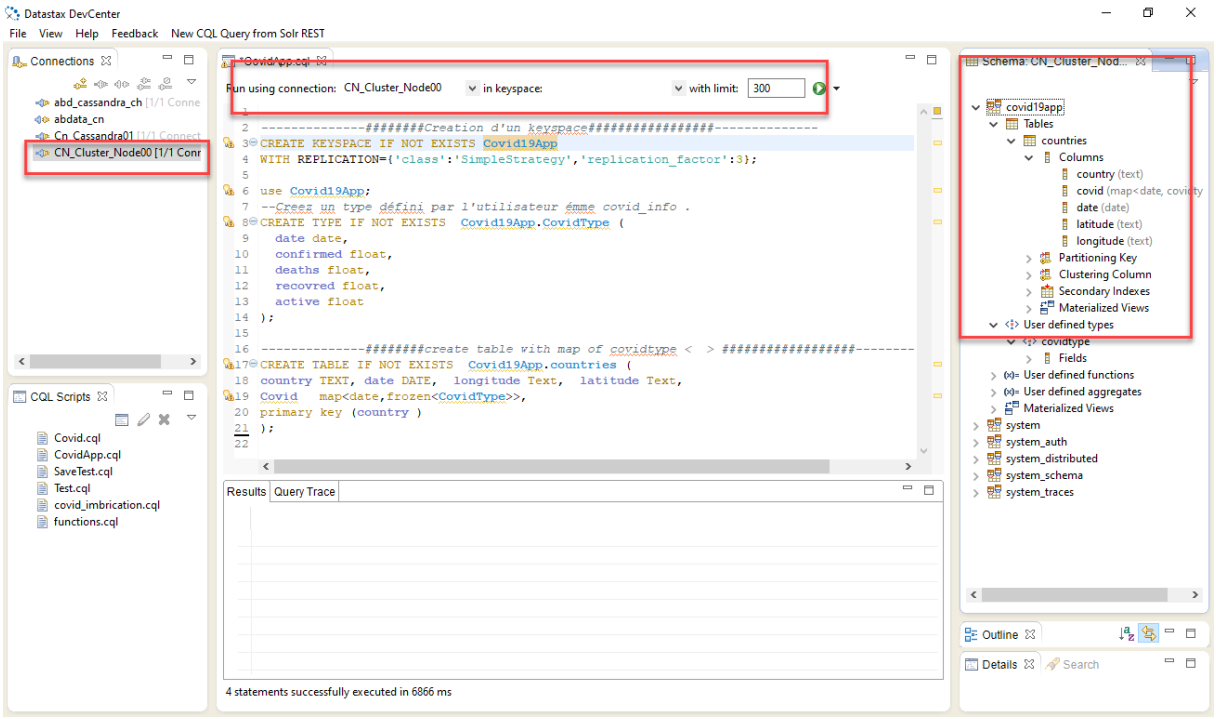
Ce Script permettant de créer un Keyspace avec une table countries utilisant une imbrication MAP et UDT – TYPE CovidType

```
--------------########Creation d'un keyspace################--------------
CREATE KEYSPACE IF NOT EXISTS Covid19App
WITH REPLICATION={'class':'SimpleStrategy','replication_factor':3};


use Covid19App;
--Creez un type défini par l'utilisateur émme covid_info .
CREATE TYPE IF NOT EXISTS  Covid19App.CovidType (
  date date,
  confirmed float,
  deaths float,
  recovred float,
  active float
);
 --------------########create table with map of covidtype <  >
##################-------------------
CREATE TABLE IF NOT EXISTS  Covid19App.countries (
country TEXT, date DATE,  longitude Text,  latitude Text,
Covid   map<date,frozen<CovidType>>,
primary key (country )
);
```

BASE DE DONNEES NO SQL Cassandra

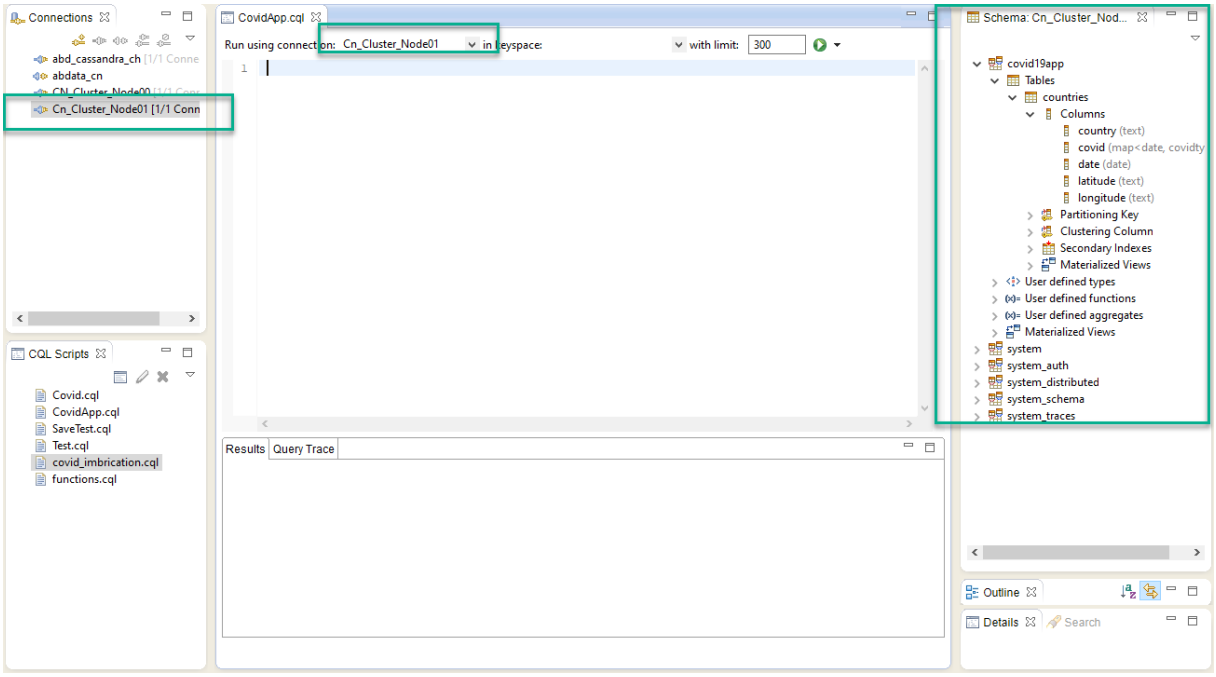
Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

Vérifier que la création est bien terminée avec succès au niveau de Cluster Node 0



Passer maintenant à vérifier dans le deuxième Nœud que se passe t'il

Comme vous pouvez noter que le mise à jour de la création du keyspace , table et type sont déjà présent dans le deuxième nœud Node01



Un Editeur web pour le développement Python

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

Jupyter est une application web utilisée pour programmer dans plus de 40 langages de programmation, dont Python, Julia, Ruby, R, ou encore Scala2. Jupyter est une évolution du projet IPython. Jupyter permet de réaliser des calepins ou notebooks, c'est-à-dire des programmes contenant à la fois du texte en markdown et du code en Julia, Python, R... Ces calepins sont utilisés en science des données pour explorer et analyser des données.

Exécuter un conteneur Jupyter

L'utilisation de l'une des Jupyter Docker Stacks nécessite deux choix:

Quelle image Docker vous souhaitez utiliser

Comment vous souhaitez démarrer les conteneurs Docker à partir de cette image

Cette section fournit des détails sur le second.

Utilisation de la CLI Docker

Vous pouvez lancer un conteneur Docker local à partir des Jupyter Docker Stacks à l'aide de l' interface de ligne de commande Docker . Il existe de nombreuses façons de configurer les conteneurs à l'aide de l'interface de ligne de commande. Voici quelques modèles courants.

Lancement du conteneur jupyter

Cette commande extrait l' image `jupyter/scipy-notebook` balisée `2c80cf3537ca` de Docker Hub si elle n'est pas déjà présente sur l'hôte local. Il démarre ensuite un conteneur exécutant un serveur Jupyter Notebook et expose le serveur sur le port hôte 8888. Les journaux du serveur apparaissent dans le terminal et incluent une URL vers le serveur Notebook.

```
$ sudo docker run --rm -p 8888:8888 -e JUPYTER_ENABLE_LAB=yes -v
"$PWD":/home/ubuntu/cassandra_abdata/ jupyter/scipy-
notebook:2c80cf3537ca
```

Le fait d'appuyer sur `Ctrl-C` arrête le serveur notebook mais laisse le conteneur intact sur le disque pour un redémarrage ultérieur ou une suppression permanente à l'aide de commandes telles que les suivantes :

```
# list containers

docker ps -a

CONTAINER ID        IMAGE                     COMMAND
CREATED     STATUS                      PORTS               NAMES
```

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

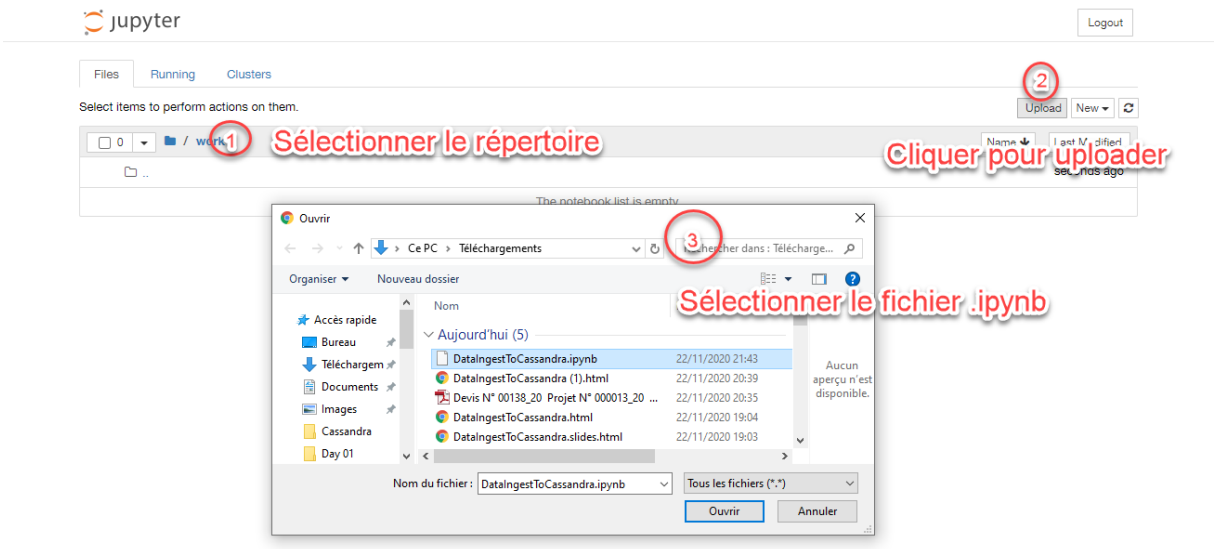
```
d67fe77f1a84        jupyter/base-notebook   "tini -- start-noteb…"   44
seconds ago     Exited (0) 39 seconds ago
cocky_mirzakhani
```

```
# start the stopped container

docker start -a d67fe77f1a84

Executing the command: jupyter notebook

[W 16:45:02.020 NotebookApp] WARNING: The notebook server is listening
on all IP addresses and not using encryption. This is not recommended.

...

# remove the stopped container

docker rm d67fe77f1a84

d67fe77f1a84
```

Lancer Jupyter en cliquant sur le lien dans le Shell

```
Digest: sha256:1457325a4df1803427042686b7b8c99261ec0ae75c6af1d4acbf2df9279c3668
Status: Downloaded newer image for jupyter/scipy-notebook:2c80cf3537ca
Executing the command: jupyter notebook
[I 20:18:59.153 NotebookApp] Writing notebook server cookie secret to /home/jovyan/.local/share/jupyter/runtime/notebook_cookie_s
ecret
[W 20:18:59.635 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not
recommended.
[I 20:18:59.676 NotebookApp] JupyterLab alpha preview extension loaded from /opt/conda/lib/python3.6/site-packages/jupyterlab
[I 20:18:59.676 NotebookApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 20:18:59.682 NotebookApp] Serving notebooks from local directory: /home/jovyan
[I 20:18:59.683 NotebookApp] 0 active kernels
[I 20:18:59.683 NotebookApp] The Jupyter Notebook is running at:
[I 20:18:59.683 NotebookApp] http://[all ip addresses on your system]:8888/?token=aa9cbf81a8cdaadc789a01f09decb64cb1091c7626720f5
7
[I 20:18:59.683 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 20:18:59.683 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=aa9cbf81a8cdaadc789a01f09decb64cb1091c7626720f57
```

Charger un notebook DataIngestion-ToCassandra-.ipynb existant sur le disque local

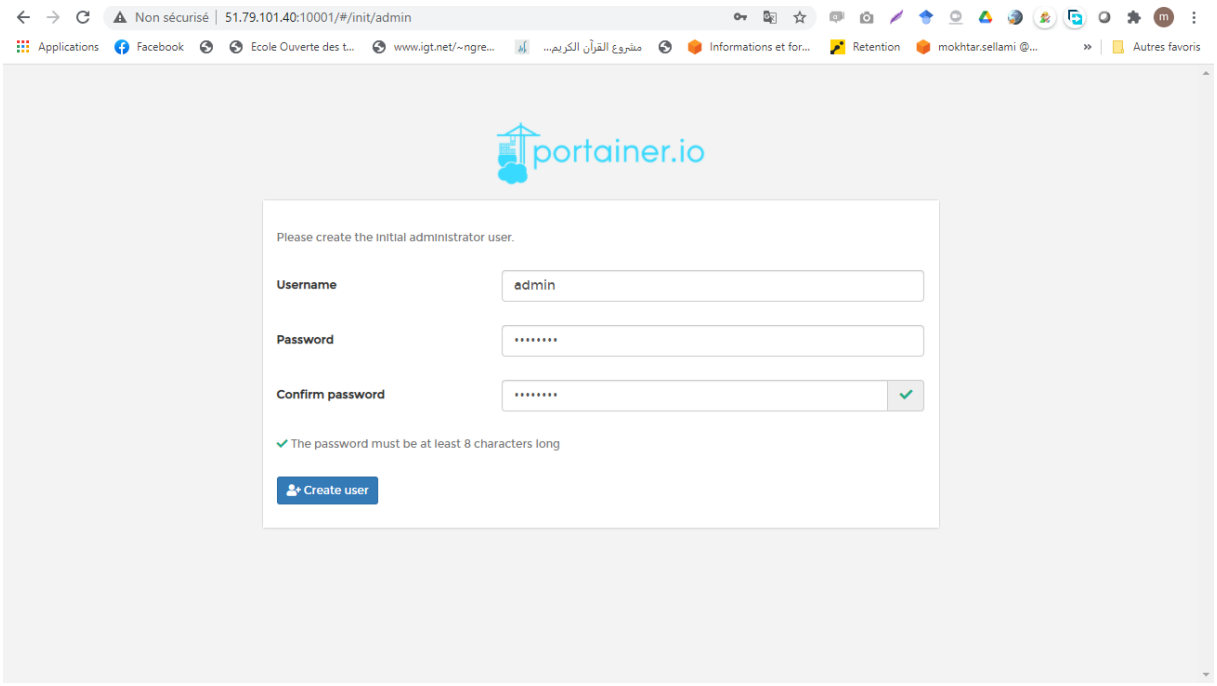


BASE DE DONNEES NO SQL Cassandra

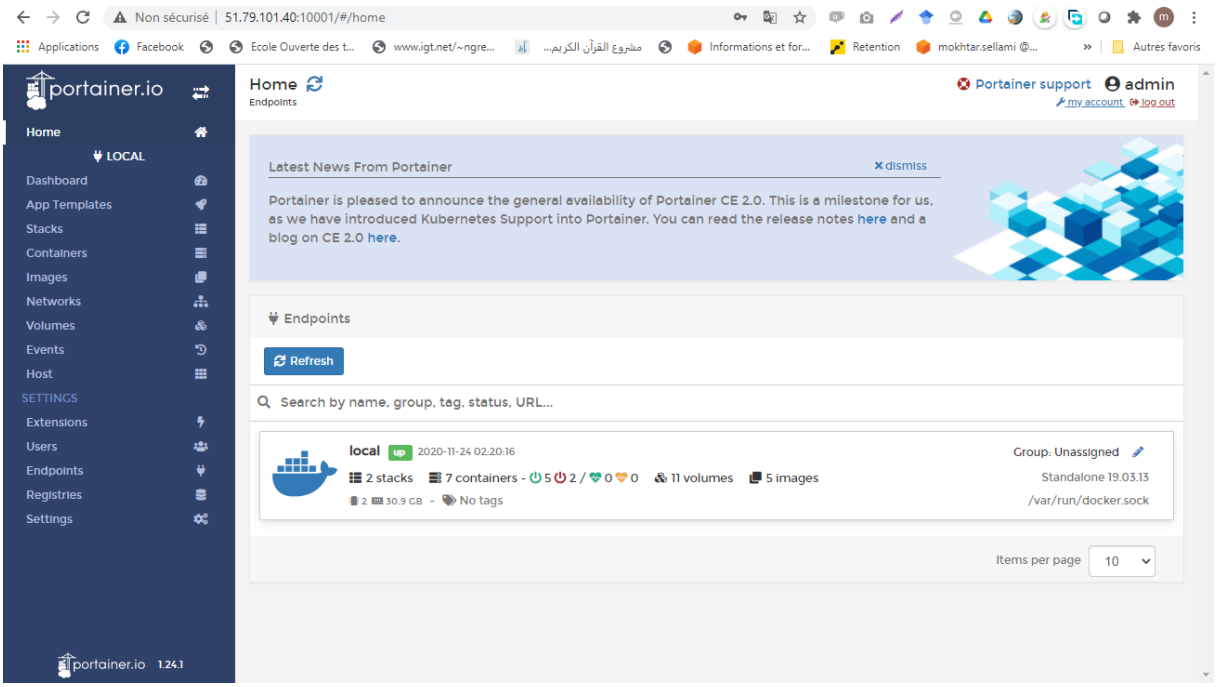
Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

Modifier les paramètres de l'adresse IP et le port pour choisir deux nœuds différents pour utiliser l'un pour ingester vos données et l'autres appliquer des analyses en lisant les données répliquées

Administration de vos conteneur dockers



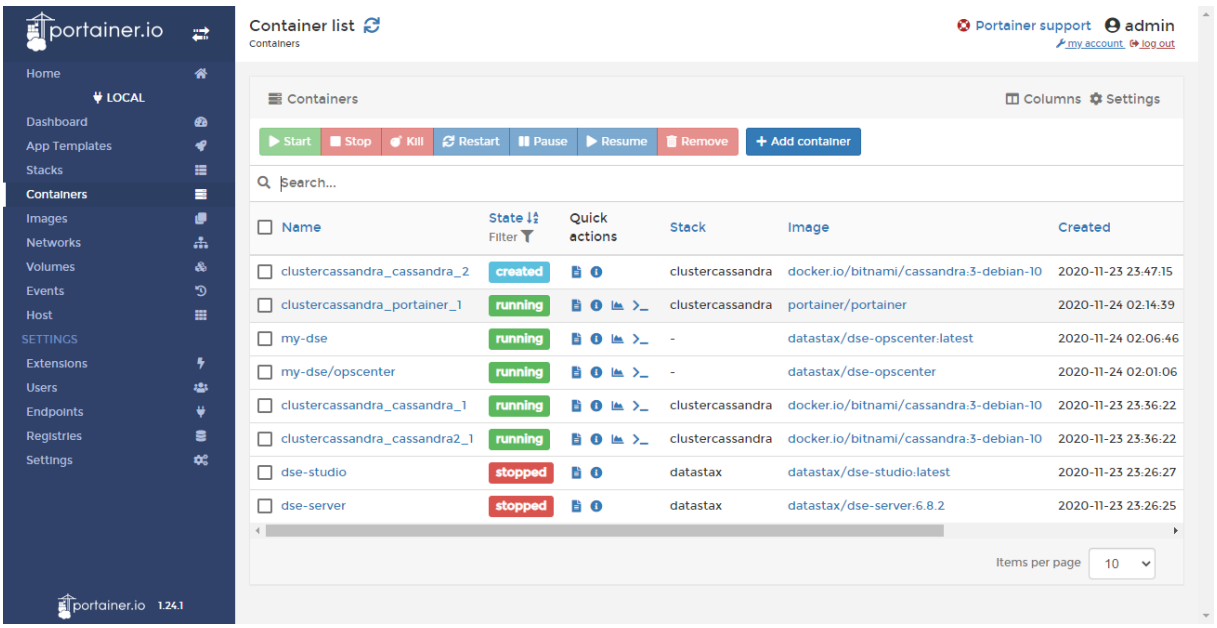
Choisir vos environnements locaux et valider pour allez vers la page d'accueil



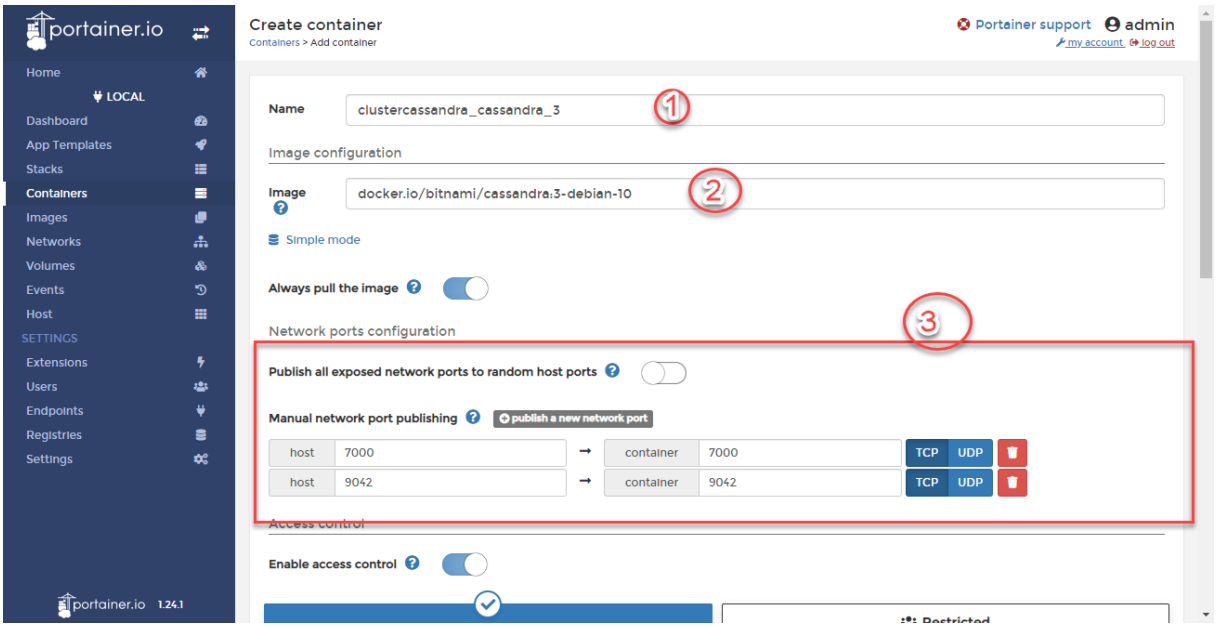
BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

Vérifier vos conteneurs et leur état



Créer un nœud Cassandra pour scale le cluster



1 Spécifier le nom du conteneur

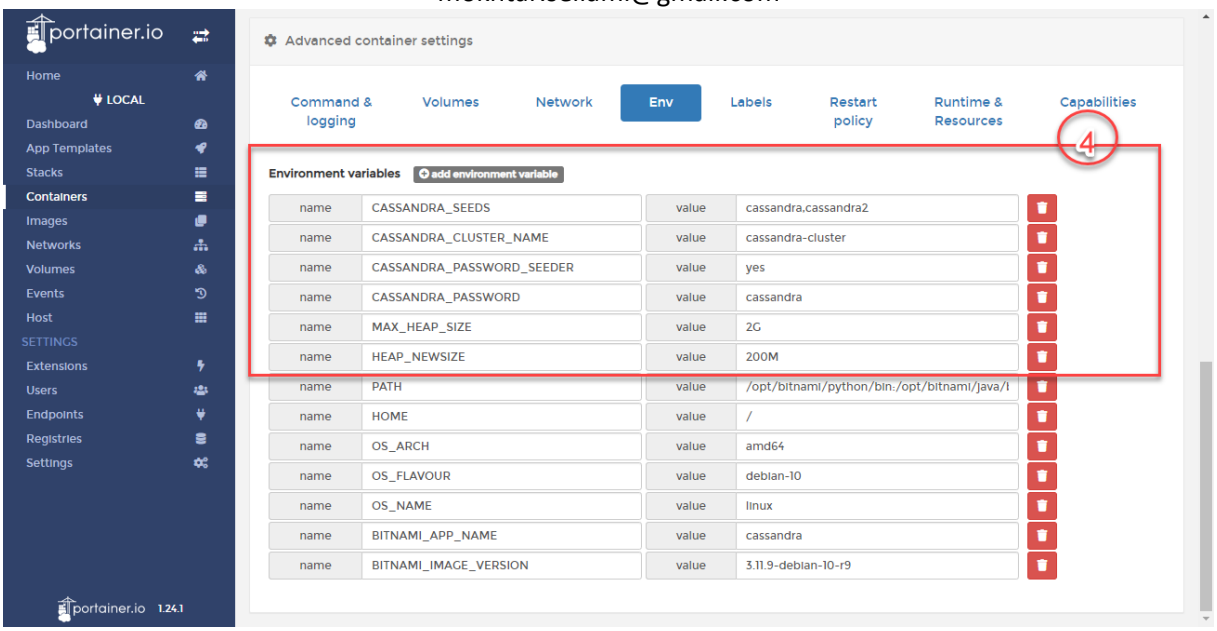
2 choisir la bonne image

3 configurations de ports et leur renvois ‘host-container’

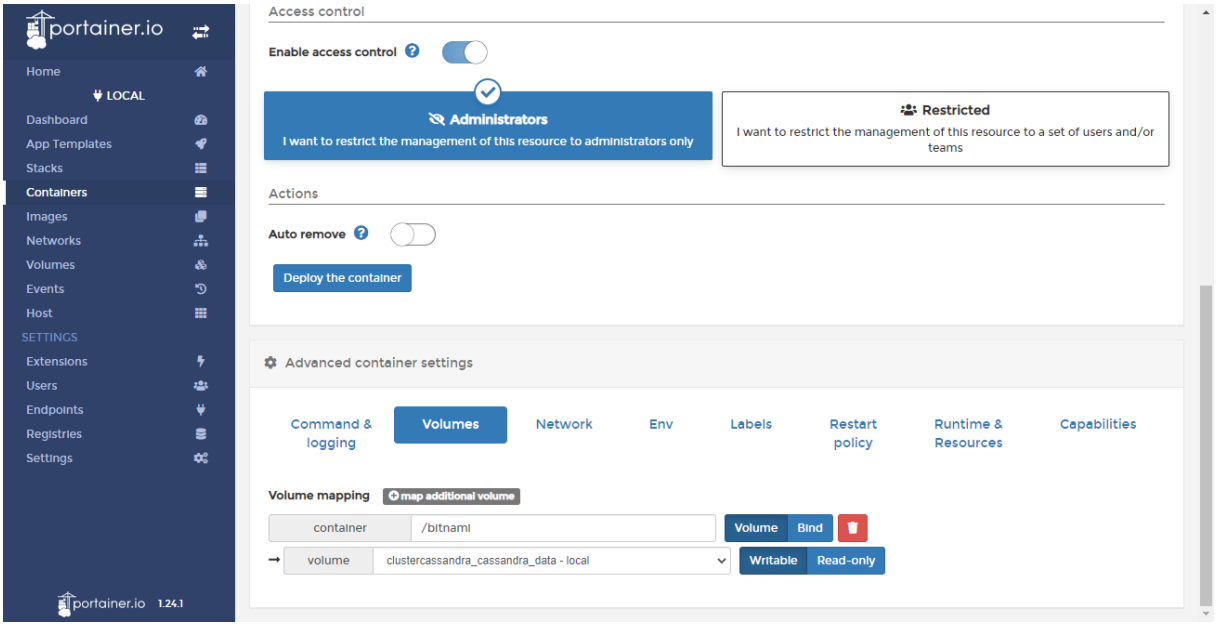
4 -configurer les variables d’environnement

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



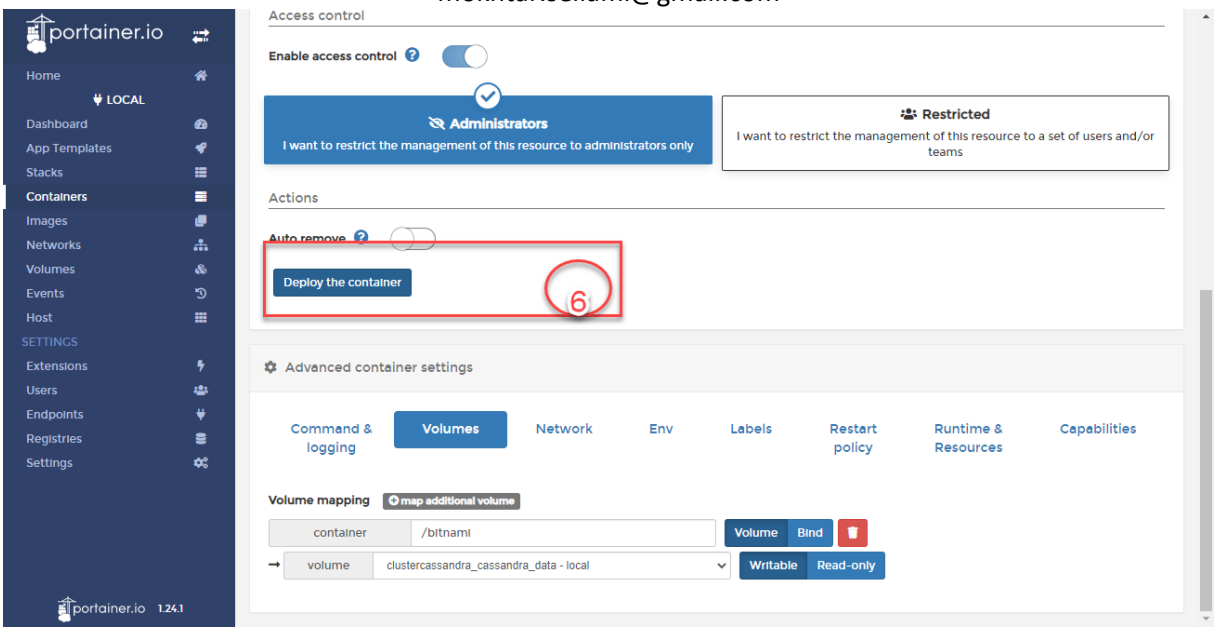
5 -Ajouter un volume a ton conteneur



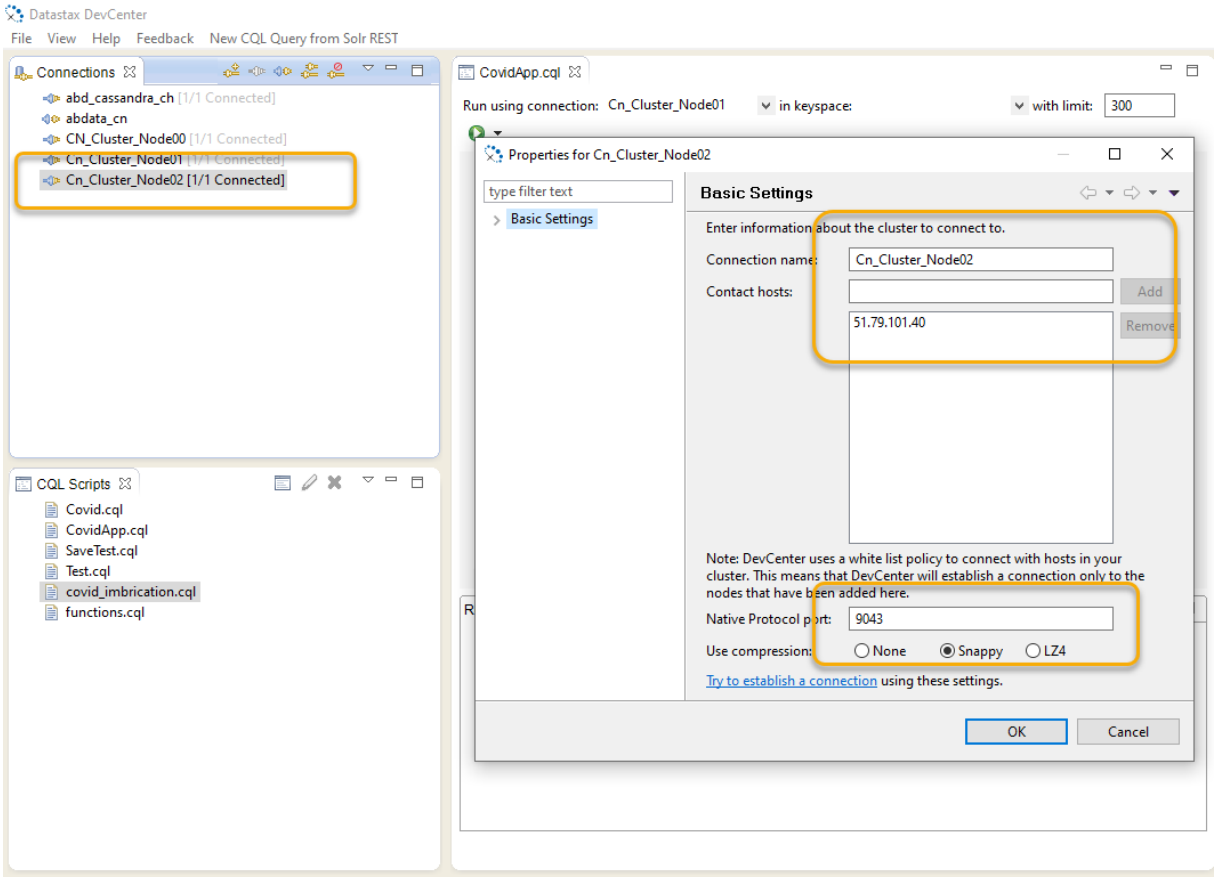
6- déployer ton conteneur

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



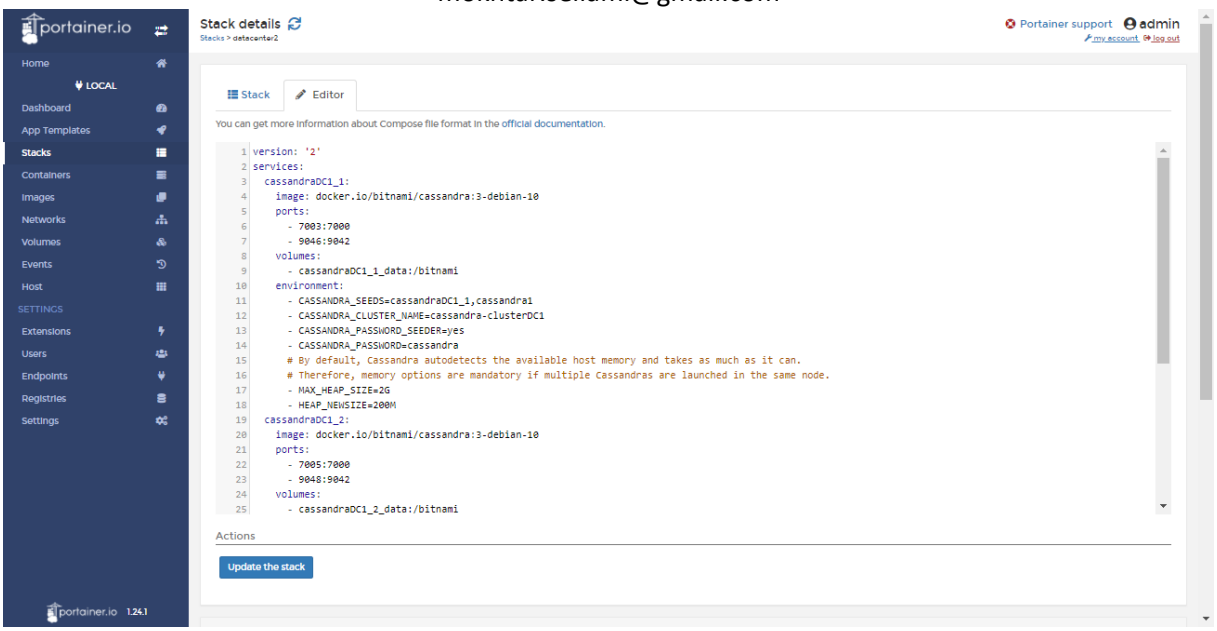
Vérification de la connexion au nouveau Node via DevCenter



Créer un DataCenter 2 via Stack

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



Configure ce DataCenter et le déployer