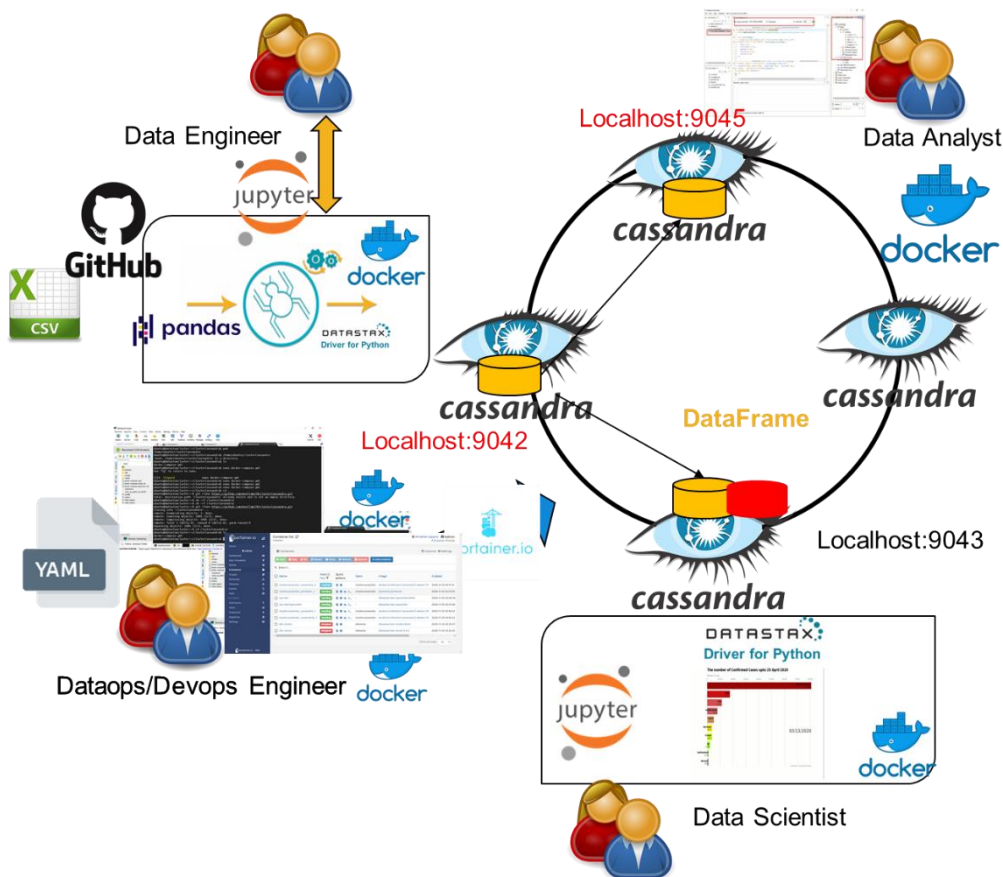


## BASE DE DONNEES NO SQL Cassandra

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 Compose
- 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

The image shows a Docker Compose file for a Cassandra cluster. It defines two services: `cassandra` and `cassandra2`. Both services use the `docker.io/bitnami/cassandra:3-debian-10` image. The `cassandra` service is configured with ports `7000` and `9042`, and a volume `cassandra_data` mapped to `/bitnami`. The `cassandra2` service is configured with ports `7001` and `9043`, and a volume `cassandra2_data` mapped to `/bitnami`. Both services have an environment section with `CASSANDRA_SEEDS=cassandra,cassandra2`, `CASSANDRA_CLUSTER_NAME=cassandra-cluster`, and `CASSANDRA_PASSWORD=cassandra`. Annotations in French highlight specific parts: 'Spécifier les des nœuds d'amorçage' points to the `SEEDS` environment variable; 'Configuration de premier Nœud' points to the `cassandra` service; 'Renvoi de ports' points to the `ports` section of `cassandra2`; 'Configuration deuxième Nœud' points to the `cassandra2` service; 'Nom du Cluster' points to the `CLUSTER_NAME` environment variable; and 'Ajouter des volumes de disques' points to the `volumes` section at the bottom.

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
8e60f03cfa5: 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

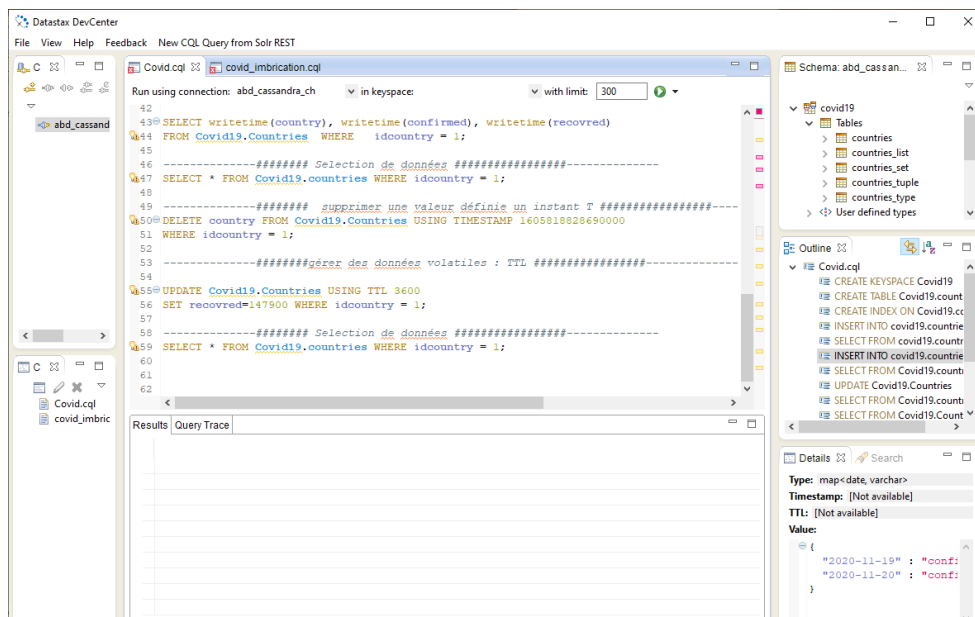
Formateur : Sellami Mokhtar  
mokhtar.sellami@gmail.com

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
46c22e7156a4	bitnami/cassandra:3-debian-10	cluster-cassandra_3	"/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	cluster-cassandra_2	"/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				

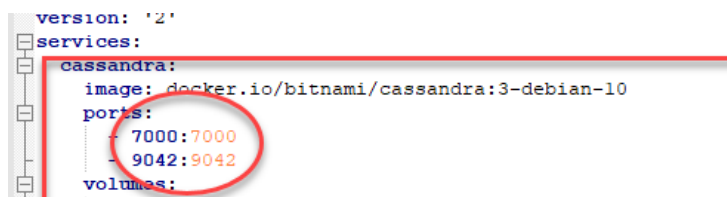
## Access aux nœuds via DevCenter

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

**New Connection**

**Basic Settings**  
Enter information about the cluster to connect to.

Connection name:

Contact hosts:

Native Protocol port:

Use compression: ☐ None ☒ Snappy ☐ LZ4

[Try to establish a connection](#) using these settings.

< Back Next > Finish Cancel

Saisir les paramètres de sécurité 'cassandra : cassandra'

**New Connection**

**Advanced Settings**  
Enter security information, if required by this cluster.

☒ This cluster requires credentials

Login:

Password:

☐ This cluster requires SSL

Truststore file:

Truststore password:

☐ Client authentication required

Keystore file:

Keystore password:

[Try to establish a connection](#) using these settings.

< Back Next > Finish Cancel

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

Properties for Cn\_Cassandra01

type filter text

> Basic Settings

### Basic Settings

Enter information about the cluster to connect to.

Connection name: Cn\_Cassandra01

Contact hosts:

51.79.101.40

Note: DevCenter uses a white list policy. This means that DevCenter will establish connections only to nodes that have been added here.

Native Protocol port: 9043

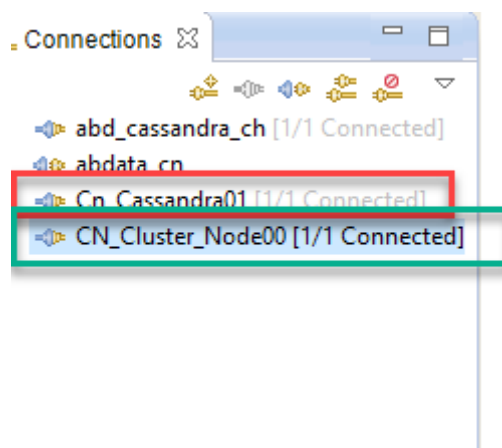
Use compression: ☐ None ☒ Snappy ☐ LZ4

[Try to establish a connection](#) using these settings.

OK Cancel

Spécifier le port adéquat (9043) au deuxième

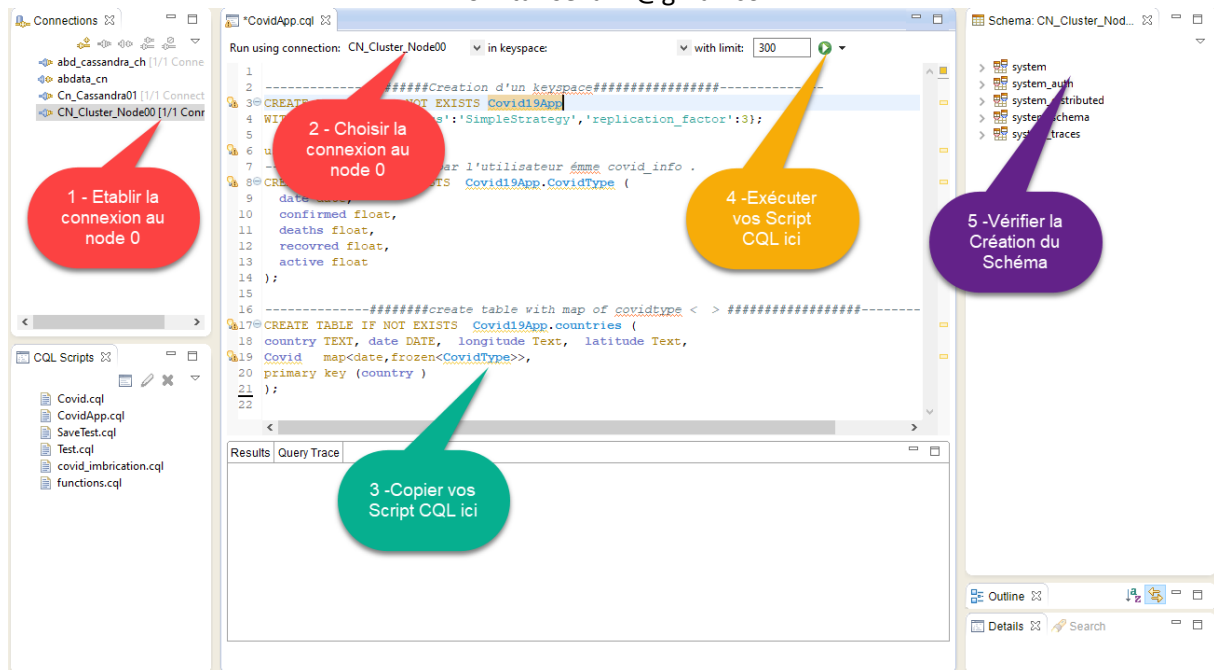
Vérifier que vous avez deux connexions actives au cluster



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

The screenshot shows the Datasax DevCenter interface. On the left, the 'Connections' panel lists 'CN\_Cluster\_Node00 [1/1 Conn]' which is highlighted with a red box. The main editor displays CQL scripts for creating a keyspace, a table, and a type. The 'Run' button is also highlighted with a red box. The 'Results' panel at the bottom shows '4 statements successfully executed in 6866 ms'. On the right, the 'Schema' panel for 'CN\_Cluster\_Node00' shows the 'covid19app' keyspace containing a 'countries' table with columns 'country', 'covid', 'date', 'latitude', and 'longitude', and a 'covidtype' type.

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

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

This screenshot shows the same Datasax DevCenter interface but with the connection switched to 'CN\_Cluster\_Node01 [1/1 Conn]', which is highlighted with a green box. The 'Run' button is also highlighted with a green box. The 'Schema' panel on the right shows that the 'covid19app' keyspace, 'countries' table, and 'covidtype' type have been replicated to this node as well.

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	PORTS	NAMES
CREATED	STATUS			

## BASE DE DONNEES NO SQL Cassandra

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_secret
[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=aa9cbf81a8cdaadc789a01f09dec64cb1091c7626720f57
[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=aa9cbf81a8cdaadc789a01f09dec64cb1091c7626720f57
```

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

The screenshot shows the Jupyter Notebook web interface. In the 'Files' tab, the 'work' directory is selected (annotated with a red circle 1 and the text 'Sélectionner le répertoire'). The 'Upload' button is visible (annotated with a red circle 2 and the text 'Cliquer pour uploader'). A file explorer window is open, showing the contents of the 'Téléchargements' folder. The file 'DatalIngestionToCassandra.ipynb' is selected (annotated with a red circle 3 and the text 'Sélectionner le fichier .ipynb').

# 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 ingérer vos données et l'autre pour appliquer des analyses en lisant les données répliquées

## Administration de vos conteneur dockers

Non sécurisé | 51.79.101.40:10001/#/init/admin

Applications Facebook Ecole Ouverte des t... www.igt.net/~ngre... مشروع القرآن الكريم... Informations et for... Retention mokhtar.sellami @... Autres favoris

**portainer.io**

Please create the initial administrator user.

Username: admin

Password: .....

Confirm password: ..... ✓

✓ The password must be at least 8 characters long

Create user

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

Non sécurisé | 51.79.101.40:10001/#/home

Applications Facebook Ecole Ouverte des t... www.igt.net/~ngre... مشروع القرآن الكريم... Informations et for... Retention mokhtar.sellami @... Autres favoris

**portainer.io**

Home Endpoints

Portainer support admin my account log out

Latest News From Portainer

Portainer is pleased to announce the general availability of Portainer CE 2.0. This is a milestone for us, as we have introduced Kubernetes Support into Portainer. You can read the release notes [here](#) and a blog on CE 2.0 [here](#).

Endpoints

Refresh

Search by name, group, tag, status, URL...

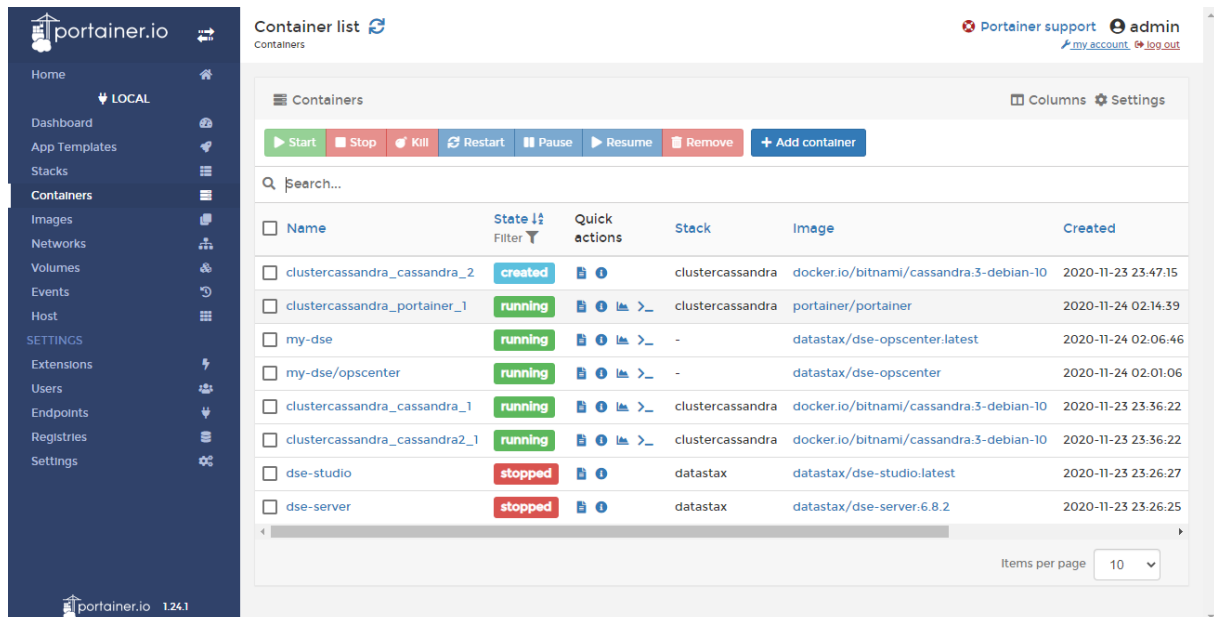
Endpoint	Status	Created	Group	Details
local	up	2020-11-24 02:20:16	Unassigned	2 stacks, 7 containers, 5 images, 11 volumes

Items per page: 10

# BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar  
mokhtar.sellami@gmail.com

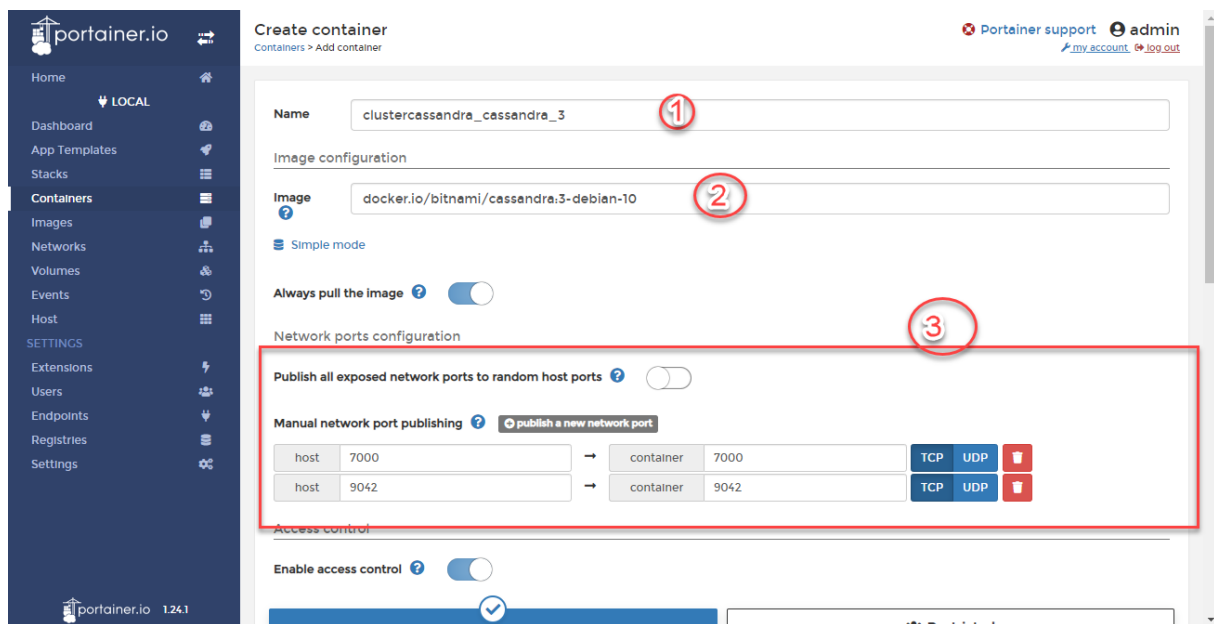
Vérifier vos conteneurs et leur état



The screenshot shows the Portainer.io web interface. On the left is a sidebar with navigation links: Home, LOCAL, Dashboard, App Templates, Stacks, Containers (selected), Images, Networks, Volumes, Events, Host, SETTINGS, Extensions, Users, Endpoints, Registries, and Settings. The main area is titled 'Container list' and shows a table of containers. At the top of the table are buttons for Start, Stop, Kill, Restart, Pause, Resume, Remove, and Add container. The table has columns for Name, State, Quick actions, Stack, Image, and Created. The containers listed are:

Name	State	Quick actions	Stack	Image	Created
clustercassandra_cassandra_2	created		clustercassandra	docker.io/bitnami/cassandra:3-debian-10	2020-11-23 23:47:15
clustercassandra_portainer_1	running		clustercassandra	portainer/portainer	2020-11-24 02:14:39
my-dse	running		-	datastax/dse-opscenter:latest	2020-11-24 02:06:46
my-dse/opscenter	running		-	datastax/dse-opscenter	2020-11-24 02:01:06
clustercassandra_cassandra_1	running		clustercassandra	docker.io/bitnami/cassandra:3-debian-10	2020-11-23 23:36:22
clustercassandra_cassandra_2_1	running		clustercassandra	docker.io/bitnami/cassandra:3-debian-10	2020-11-23 23:36:22
dse-studio	stopped		datastax	datastax/dse-studio:latest	2020-11-23 23:26:27
dse-server	stopped		datastax	datastax/dse-server:6.8.2	2020-11-23 23:26:25

Créer un nœud Cassandra pour scale le cluster



The screenshot shows the 'Create container' form in Portainer.io. The form has several fields and sections:

- Name:** clustercassandra\_cassandra\_3 (annotated with a red circle and the number 1).
- Image configuration:**
  - Image:** docker.io/bitnami/cassandra:3-debian-10 (annotated with a red circle and the number 2).
  - Simple mode:** A checkbox that is checked.
  - Always pull the image:** A toggle switch that is turned on.
- Network ports configuration:** (Annotated with a red circle and the number 3)
  - Publish all exposed network ports to random host ports:** A toggle switch that is turned off.
  - Manual network port publishing:** A section with a button 'publish a new network port' and a table:
 

host	container	TCP	UDP	
7000	7000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="X"/>
9042	9042	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="X"/>
- Access control:**
  - Enable access control:** A toggle switch that is turned off.

1 Spécifier le nom du conteneur

2 choisir la bonne image

3 configurations de ports et leur renvoi 'host-container'

4 -configurer les variables d'environnement

# BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar  
mokhtar.sellami@gmail.com

Advanced container settings

Command & logging Volumes Network **Env** Labels Restart policy Runtime & Resources Capabilities

Environment variables [add environment variable](#)

name	value	
CASSANDRA_SEEDS	cassandra.cassandra2	
CASSANDRA_CLUSTER_NAME	cassandra-cluster	
CASSANDRA_PASSWORD_SEEDER	yes	
CASSANDRA_PASSWORD	cassandra	
MAX_HEAP_SIZE	2G	
HEAP_NEWSIZE	200M	
PATH	/opt/bitnami/python/bin:/opt/bitnami/java/	
HOME	/	
OS_ARCH	amd64	
OS_FLAVOUR	debian-10	
OS_NAME	linux	
BITNAMI_APP_NAME	cassandra	
BITNAMI_IMAGE_VERSION	3.11.9-debian-10-r9	

## 5 -Ajouter un volume a ton conteneur

Access control

Enable access control ☒

**Administrators**  
I want to restrict the management of this resource to administrators only

**Restricted**  
I want to restrict the management of this resource to a set of users and/or teams

Actions

Auto remove ☐

[Deploy the container](#)

Advanced container settings

Command & logging **Volumes** Network Env Labels Restart policy Runtime & Resources Capabilities

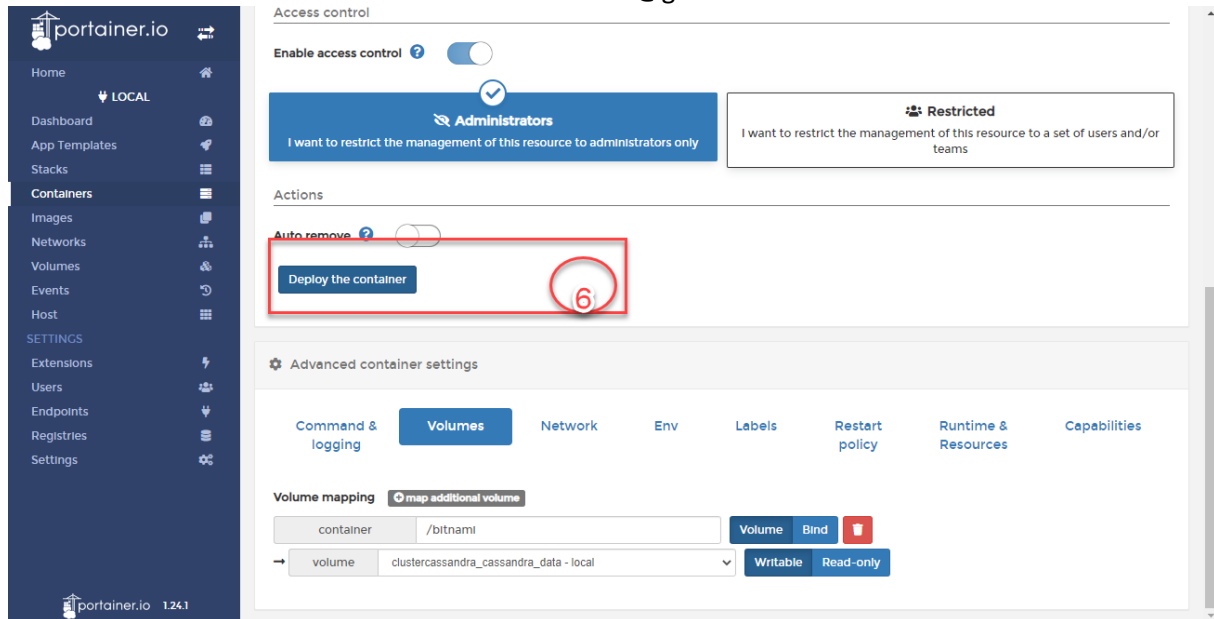
Volume mapping [map additional volume](#)

container	volume	Volume	Bind	
/bitnami	clustercassandra_cassandra_data - local	Writable	Read-only	

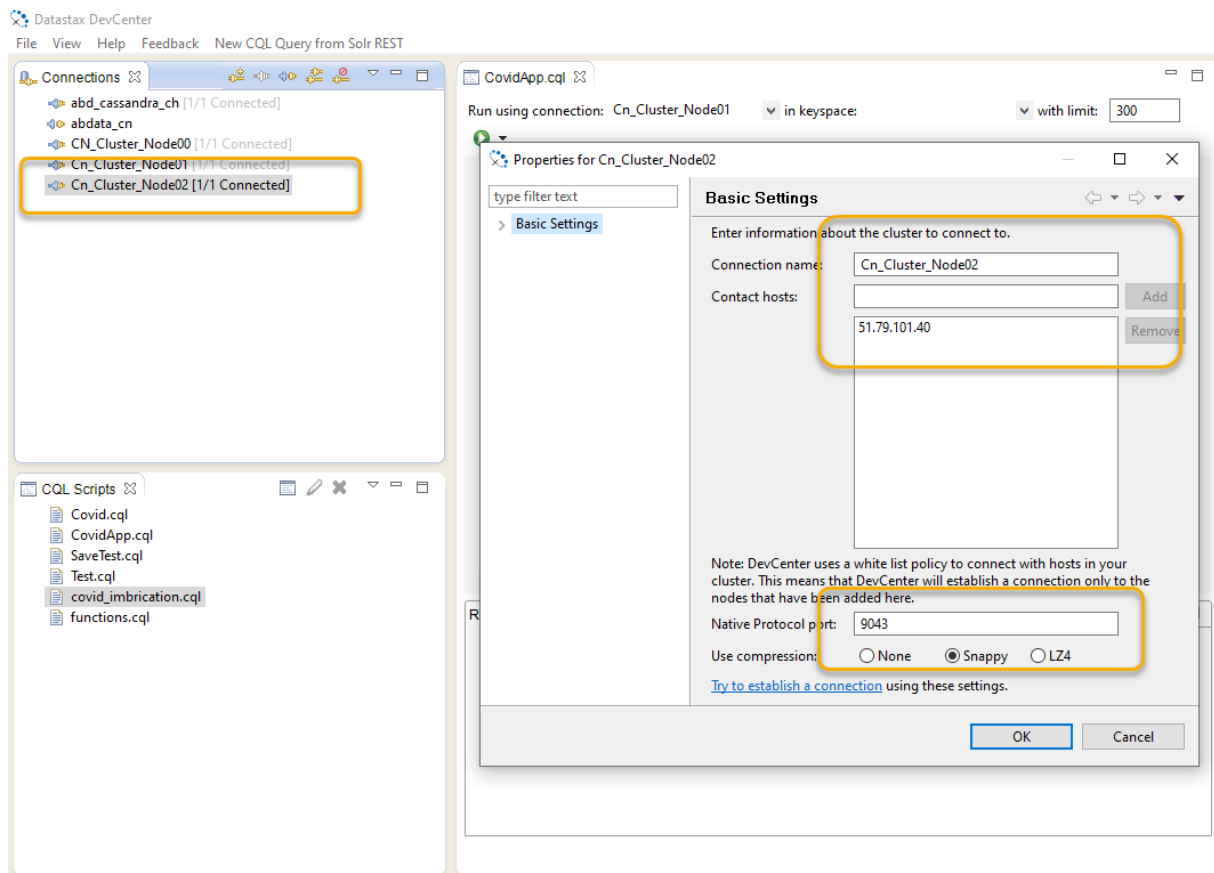
## 6- déployer ton conteneur

# 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

Stack details  
Stacks > defacenter2

Stack Editor

You can get more information about Compose file format in the [official documentation](#).

```

1 version: '2'
2 services:
3   cassandraDC1_1:
4     image: docker.io/bitnami/cassandra:3-debian-10
5     ports:
6       - 7003:7000
7       - 9046:9042
8     volumes:
9       - cassandraDC1_1_data:/bitnami
10    environment:
11      - CASSANDRA_SEEDS=cassandraDC1_1,cassandra1
12      - CASSANDRA_CLUSTER_NAME=cassandra-clusterDC1
13      - CASSANDRA_PASSWORD_SEEDER=yes
14      - CASSANDRA_PASSWORD=cassandra
15      # By default, Cassandra autodetects the available host memory and takes as much as it can.
16      # Therefore, memory options are mandatory if multiple Cassandras are launched in the same node.
17      - MAX_HEAP_SIZE=2G
18      - HEAP_NEWSIZE=200M
19    cassandraDC1_2:
20      image: docker.io/bitnami/cassandra:3-debian-10
21      ports:
22        - 7005:7000
23        - 9048:9042
24      volumes:
25        - cassandraDC1_2_data:/bitnami

```

Actions

[Update the stack](#)

Configure ce DataCenter et le déployer

Stack details  
Stacks > defacenter2

Stack Editor

You can get more information about Compose file format in the [official documentation](#).

```

1 version: '2'
2 services:
3   cassandraDC1_1:
4     image: docker.io/bitnami/cassandra:3-debian-10
5     ports:
6       - 7003:7000
7       - 9046:9042
8     volumes:
9       - cassandraDC1_1_data:/bitnami
10    environment:
11      - CASSANDRA_SEEDS=cassandraDC1_1,cassandra1
12      - CASSANDRA_CLUSTER_NAME=cassandra-clusterDC1
13      - CASSANDRA_PASSWORD_SEEDER=yes
14      - CASSANDRA_PASSWORD=cassandra
15      # By default, Cassandra autodetects the available host memory and takes as much as it can.
16      # Therefore, memory options are mandatory if multiple Cassandras are launched in the same node.
17      - MAX_HEAP_SIZE=2G
18      - HEAP_NEWSIZE=200M
19    cassandraDC1_2:
20      image: docker.io/bitnami/cassandra:3-debian-10
21      ports:
22        - 7005:7000
23        - 9048:9042
24      volumes:
25        - cassandraDC1_2_data:/bitnami

```

Actions

[Update the stack](#)