



Débuter avec CrateDB

CrateDB est une base de données SQL distribuée qui simplifie le stockage et l'analyse en temps réel d'énormes quantités de données provenant de machine. Voici quelques étapes pour bien commencer avec CrateDB.

Installation

- Linux

Configuration en une étape

Vous pouvez télécharger et exécuter CrateDB sur l'un de ces systèmes d'exploitation à l'aide d'une simple commande dans votre application de terminal:

```
sh$ bash -c "$(curl -L https://try.crate.io/)"
```

- macOS

Configuration en une étape

Vous pouvez télécharger et exécuter CrateDB sur l'un de ces systèmes d'exploitation à l'aide d'une simple commande dans votre application de terminal:

```
sh$ bash -c "$(curl -L https://try.crate.io/)"
```

Si Java 11 n'est pas déjà installé, la commande ci-dessus tentera de s'occuper de cela pour vous, ainsi que de quelques tâches de maintenance.

Windows

<https://crate.io/docs/crate/getting-started/en/latest/install-run/special/windows.html>

- sur Docker

CrateDB et Docker s'accordent à merveille grâce à l'architecture horizontale évolutive sans partage de CrateDB qui se prête bien à la conteneurisation.

Configuration en une étape

Faites pivoter l'image officielle de Docker CrateDB, comme suit:

```
sh$ docker run -p "4200:4200" crate
```

Par défaut, l'image CrateDB Docker stocke les données dans le conteneur. Si vous supprimez le conteneur, les données seront également supprimées. Lorsque vous êtes prêt à utiliser CrateDB pour des données qui vous intéressent, consultez le guide complet de CrateDB et de Docker.

Première utilisation

Afin de se familiariser avec les bases de CrateDB, nous allons:

- importer des données de test pour pouvoir commencer à expérimenter
- apprendre les bases des requêtes

IMPORTATION DE DONNÉES DE TEST

Avant de pouvoir réellement expérimenter CrateDB, vous avez besoin de quelques données. Dans cette partie, nous vous montrons comment importer des données de test.

- Présentation: l'interface utilisateur d'administration de CrateDB

CrateDB est fourni avec une interface utilisateur d'administration Web, ou une interface utilisateur d'administration, en abrégé. L'interface utilisateur d'administration de CrateDB s'exécute sur chaque nœud CrateDB. Vous pouvez l'utiliser pour inspecter et interagir avec l'ensemble du cluster CrateDB de plusieurs manières.

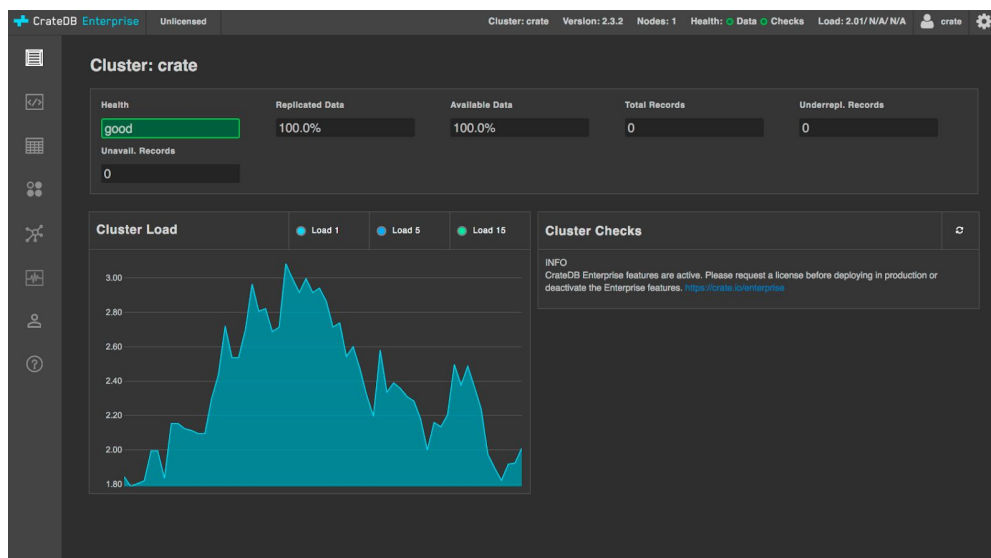
Nous allons utiliser l'interface utilisateur d'administration tout au long de cette section.

Accédez à l'interface utilisateur d'administration dans votre navigateur à l'aide d'une URL comme celle-ci:

```
http://localhost:4200/
```

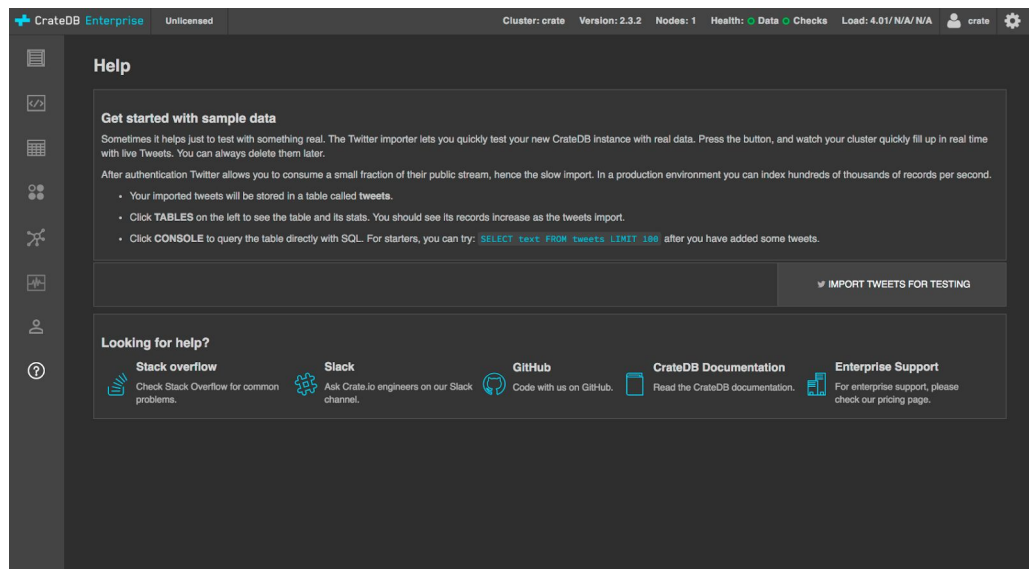
NB. Si CrateDB ne s'exécute pas localement, remplacez localhost par le nom d'hôte sur lequel CrateDB est en cours d'exécution.

Vous devriez voir quelque chose comme ça:



- Importer des tweets

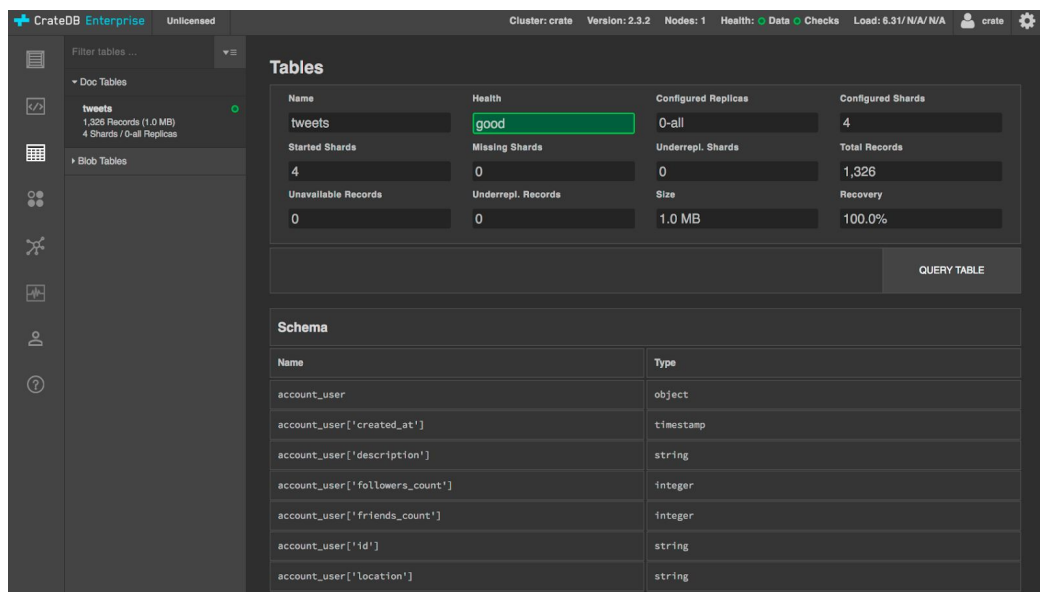
Dans l'interface utilisateur d'administration, sélectionnez l'onglet Aide (icône en forme de point d'interrogation) dans le menu de navigation de gauche. Vous devriez voir quelque chose comme ça:



Sélectionnez IMPORT TWEETS FOR TESTING et suivez les instructions pour authentifier votre compte Twitter.

Laissez cette tâche en cours et arrêtez-la lorsque vous avez suffisamment de tweets. Visez au moins quelques milliers. Lorsque vous avez terminé, sélectionnez l'icône Tables dans la navigation de gauche.

Vous devriez voir quelque chose comme ça:

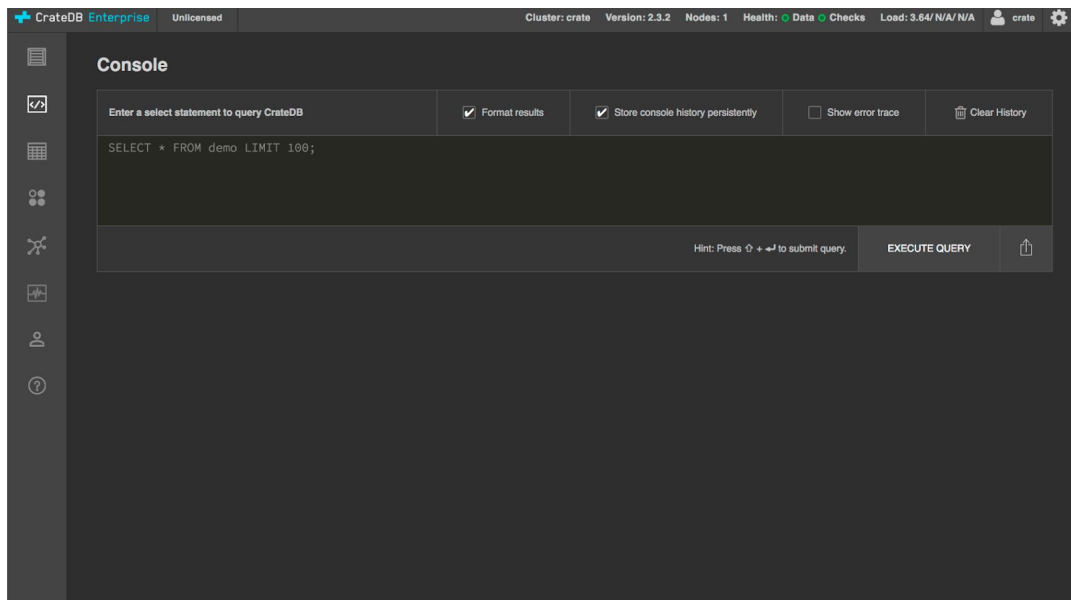


Effectuer des requêtes sur vos tweets

L'un des moyens les plus simples d'interroger CrateDB consiste à utiliser la console de requête de l'interface utilisateur d'administration CrateDB.

- Ouvrez l'interface utilisateur d'administration.
- Sélectionnez l'onglet Console dans le menu de navigation de gauche.

Vous devriez voir quelque chose comme ça:



Si vous souhaitez en savoir plus sur le tableau des tweets, sélectionnez l'onglet Tableaux dans le menu de navigation de gauche. Cet écran vous montrera le schéma de la table.

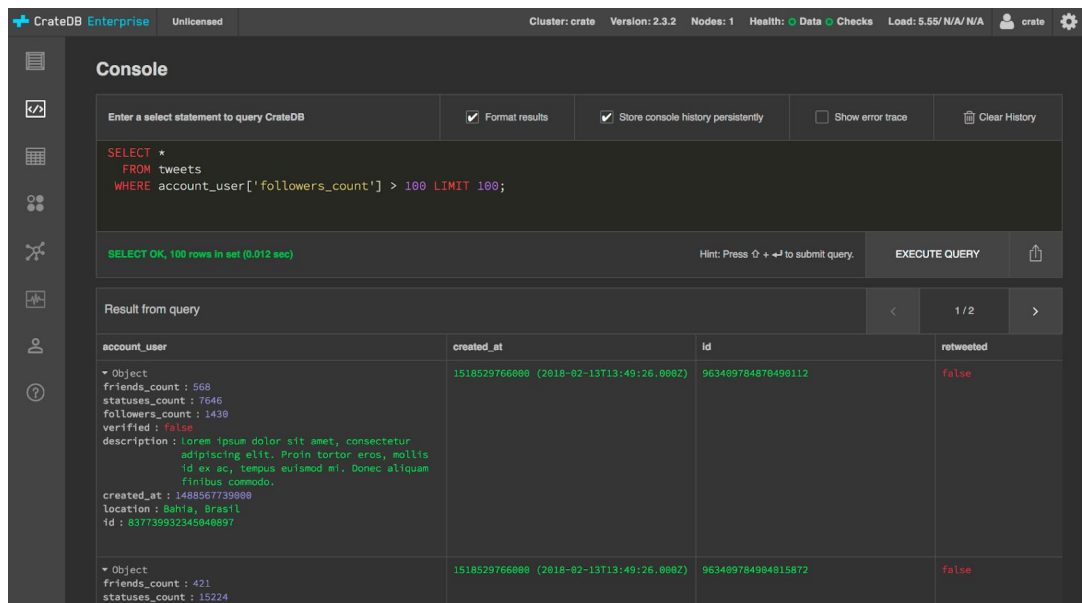
Pour un exemple de requête, on pourrait filtrer ces tweets de manière à ne consulter que les tweets de personnes de plus de 100 abonnés. Nous pouvons le faire avec une requête comme celle-ci:

```
SELECT *  
FROM tweets  
WHERE account_user['followers_count'] > 100  
LIMIT 100;
```

La table des tweets a une colonne nommée **account_user**. **Account_user** est un objet et peut être interrogé en spécifiant des attributs d'objet. Dans la requête ci-dessus, il s'agit des tuples dans lesquelles l'attribut `followers_count` de cet objet est supérieur à 100.

Tapez cette requête dans la console de requête de l'interface utilisateur admin. Puis sélectionnez EXECUTE QUERY.

Vous devriez voir quelque chose comme ça:



La prise en charge SQL de CrateDB inclut l'agrégation distribuée, les fonctions scalaires et la prise en charge géospatiale pour les requêtes plus complexes et l'analyse de données. CrateDB vient également avec la recherche en texte intégral.

NB. Voici un lien pour plus d'infos sur les requêtes possibles avec CrateDB
<https://crate.io/docs/crate/reference/en/latest/general/dql/index.html>

Allons plus loin!

Avant d'examiner les bibliothèques clientes de CrateDB, jetons un coup d'œil rapide à trois autres façons de consulter CrateDB.

- Le shell CrateDB

Le shell CrateDB (aka Crash) est un shell en ligne de commande.

Tout d'abord, vous devez installer Crash ⇒ [lien d'installation](#)

Ensuite, vous pouvez démarrer le shell comme suit:

```
sh$ crash
```

Cela se connectera automatiquement à CrateDB s'exécutant sur localhost. À partir de là, vous pouvez exécuter des requêtes sur CrateDB en les tapant et en appuyant sur Entrée.

Essayons quelque chose de nouveau.

Tapez la requête suivante:

```
SELECT account_user['followers_count']
FROM tweets
ORDER BY account_user['followers_count'] DESC
LIMIT 10;
```

Ici, nous ordonnons par nombre de suiveurs du plus élevé au plus bas, puis nous sélectionnons les 10 premières valeurs.

Appuyez sur Entrée.

Vous devriez voir quelque chose comme ça:

```
+-----+
| account_user['followers_count'] |
+-----+
|                               |
|                               |
|                               |
|                               |
|                               |
|                               |
|                               |
|                               |
|                               |
|                               |
|                               |
+-----+
SELECT 10 rows in set (0.003 sec)
```

- Le point final HTTP CrateDB

CrateDB fournit un point de terminaison HTTP.

Exécutons une autre requête.

```
SELECT COUNT(*) FROM tweets
```

Cette fois, tout ce que nous voulons savoir, c'est combien de tweets nous avons.

En utilisant HTTPie(<https://httpie.org/>), vous pouvez exécuter cette requête contre CrateDB en écoutant sur `localhost:4200`, comme suit:

```
sh$ http localhost:4200/_sql stmt="SELECT COUNT(*) FROM tweets"
```

CrateDB répondra par un fichier JSON. Vous devriez voir quelque chose comme ça:

```
HTTP/1.1 200 OK
content-length: 71
content-type: application/json; charset=UTF-8

{
  "cols": [
    "count(*)"
  ],
  "duration": 11.847271,
  "rowcount": 1,
  "rows": [
    [
      3879
    ]
  ]
}
```

Dans cet exemple, nous avons une seule ligne avec une seule colonne: 3879.

- Outils clients tiers¶

L'interface utilisateur d'administration et Crash sont les deux outils client fournis avec CrateDB. Mais ce ne sont pas les seuls outils clients que vous pouvez utiliser avec CrateDB. Du fait que CrateDB utilise le protocole filaire PostgreSQL, de nombreux outils client compatibles avec PostgreSQL fonctionnent avec CrateDB.

Learn more [here](#)

Commencer à construire votre système

CrateDB s'intègre au cœur de toute pile d'applications, avec un certain nombre de bibliothèques pour prendre en charge le développement.

Vous trouverez ci-dessous une sélection de bibliothèques de développement connues pour fonctionner avec CrateDB. Cela est divisé en deux catégories: les logiciels gérés par Crate.io et les logiciels gérés par la communauté CrateDB.

Choisissez [votre bibliothèque](#) et commencez à construire!

Language	Maintainers	Drivers	Plugins and ORMs
Java	Crate.io	crate-jdbc	
Python	Crate.io	crate	SQLAlchemy
PHP	Crate.io	CrateDB PDO	DBAL
C# (.NET)	Crate.io	Npgsql	
Python	Community	asyncpg	
Ruby	Community	crate_ruby	ActiveRecord
Scala	Community	crate-scala	
Node.js	Community	crate-connect , cratejs , node-crate	Loopback
Go	Community	pgx	
Perl	Community	DBD::Crate	
Erlang	None (EOL)	craterl	

If you would like to see something added to this page, please [get in touch](#).

