# Cassandra Connect

Blending Cassandra with Python

## What is Cassandra?

Apache Cassandra™ is a distributed NoSQL database that delivers continuous availability, high performance, and linear scalability that successful applications require.

Apache Cassandra is an open source, distributed NoSQL database that began internally at Facebook and was released as an open-source project in July 2008. Cassandra delivers continuous availability (zero downtime), high performance, and linear scalability that modern applications require, while also offering operational simplicity and effortless replication across data centers and geographies. Cassandra can handle petabytes of information and thousands of concurrent operations per second, enabling organizations to manage large amounts of data across hybrid cloud and multi cloud environments.

DataStax have provided a fantastic outlook on the “What is” and “Why” of Cassandra @ https://www.datastax.com/cassandra

However, it is indeed a subtly convoluted process to get started with Cassandra considering its installation itself takes some good time and efforts. In this article, I take you through this process step by step to install and ensure Cassandra is up and running in the local system. In addition, this article also gives a glimpse of using accessing the cloud version of Cassandra through DataStax. By the end of this article, I finally discuss configuring Cassandra-Python integration to perform the usual Database operations with Cassandra through Python.

So here we start.

## Cassandra Installation – Local System

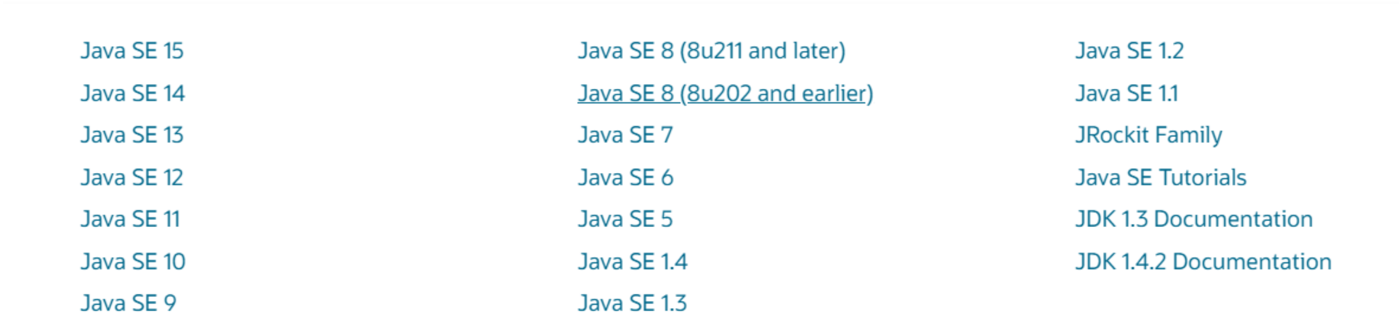
Cassandra Installation follows the sequence as:

- Java Installation (Version 8u202 or earlier)
- Cassandra Installation
- Python2 Installation (Cassandra-Python Integration demands Python2 to work seamlessly as of now)
- Apache Thrift Installation

## Java Installation

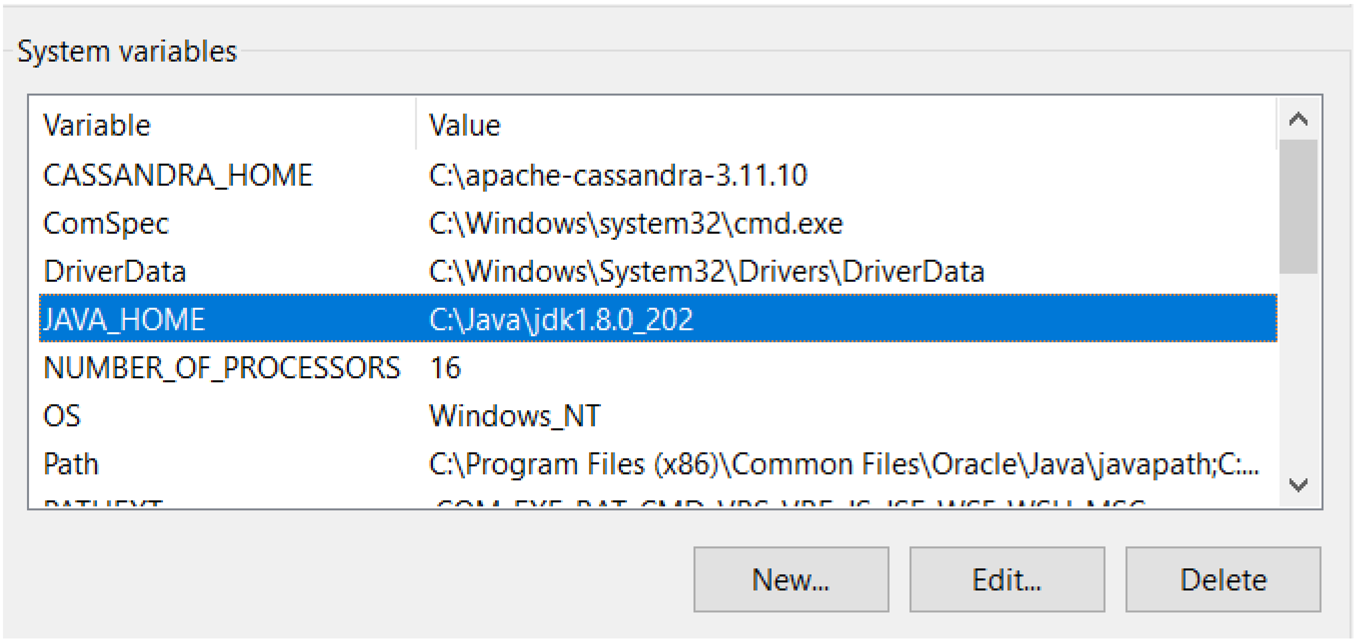
Follow the link : https://www.oracle.com/in/java/technologies/oracle-java-archive-downloads.html

Signing into Oracle is mandatory to progress. Please ensure, the version earlier than Java SE 8u202 is chosen.



- Java SE 15
- Java SE 14
- Java SE 13
- Java SE 12
- Java SE 11
- Java SE 10
- Java SE 9
- Java SE 8 (8u211 and later)
- Java SE 8 (8u202 and earlier)
- Java SE 7
- Java SE 6
- Java SE 5
- Java SE 1.4
- Java SE 1.3
- Java SE 1.2
- Java SE 1.1
- JRockit Family
- Java SE Tutorials
- JDK 1.3 Documentation
- JDK 1.4.2 Documentation

Once, JAVA is downloaded, set up a new System Environment Variable JAVA_HOME. This should have the link to the Java folder downloaded.



System variables

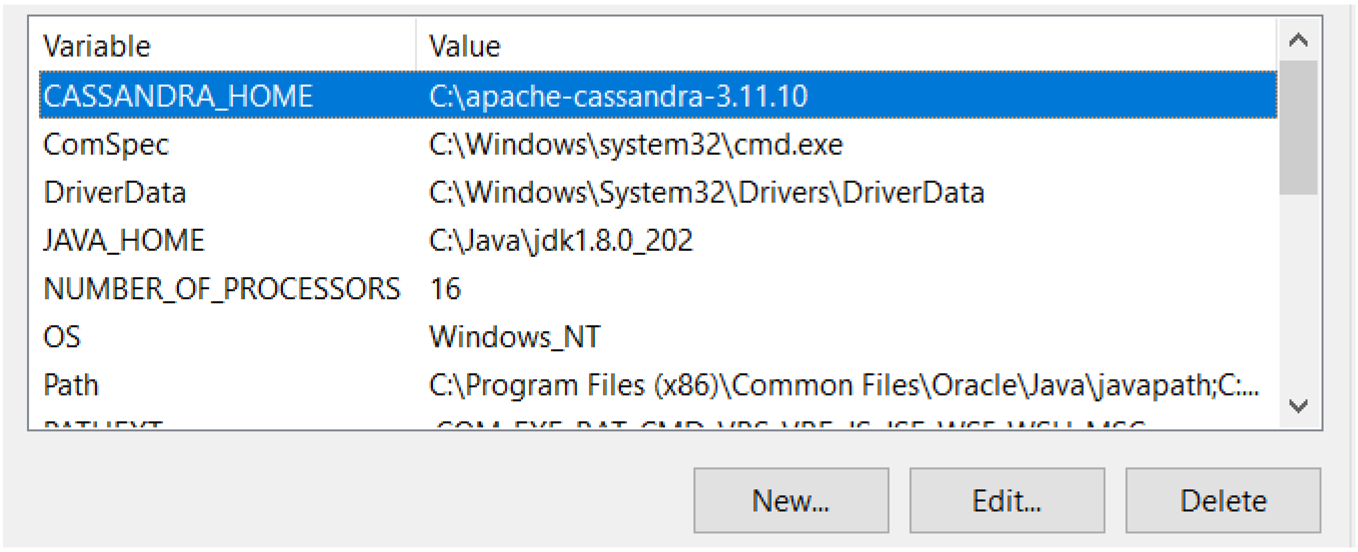
| Variable | Value |
|---|---|
| CASSANDRA_HOME | C:\apache-cassandra-3.11.10 |
| ComSpec | C:\Windows\system32\cmd.exe |
| DriverData | C:\Windows\System32\Drivers\DriverData |
| JAVA_HOME | C:\Java\jdk1.8.0_202 |
| NUMBER_OF_PROCESSORS | 16 |
| OS | Windows_NT |
| Path | C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:... |
| PATHEXT | .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC |

## Cassandra Installation

Follow the link: https://apachemirror.wuchna.com/cassandra/3.11.10/apache-cassandra-3.11.10-bin.tar.gz

Ensure, 7zip is installed in your system to handle the “. tar.gz” files.

Extract the folder from the download the “. tar.gz” file. Create another system variable, CASSANDRA_HOME with the link to the extracted folder.



| Variable | Value |
|---|---|
| CASSANDRA_HOME | C:\apache-cassandra-3.11.10 |
| ComSpec | C:\Windows\system32\cmd.exe |
| DriverData | C:\Windows\System32\Drivers\DriverData |
| JAVA_HOME | C:\Java\jdk1.8.0_202 |
| NUMBER_OF_PROCESSORS | 16 |
| OS | Windows_NT |
| Path | C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:... |

## Python 2 Environment

Well, this is not mandatory, but I have observed this a few times myself and checked with some friends of mine, Cassandra encounters issues when we try to launch the CQL (Cassandra Query Language) through a Python 3 environment. So, it is advisable to create a Python 2 environment and try launching Cassandra from this Python 2 environment.

I have used Anaconda Prompt to get this done. I have created a Python 2 environment named Python2.

## Apache Thrift Installation

Well, one the steps which is swift to perform.

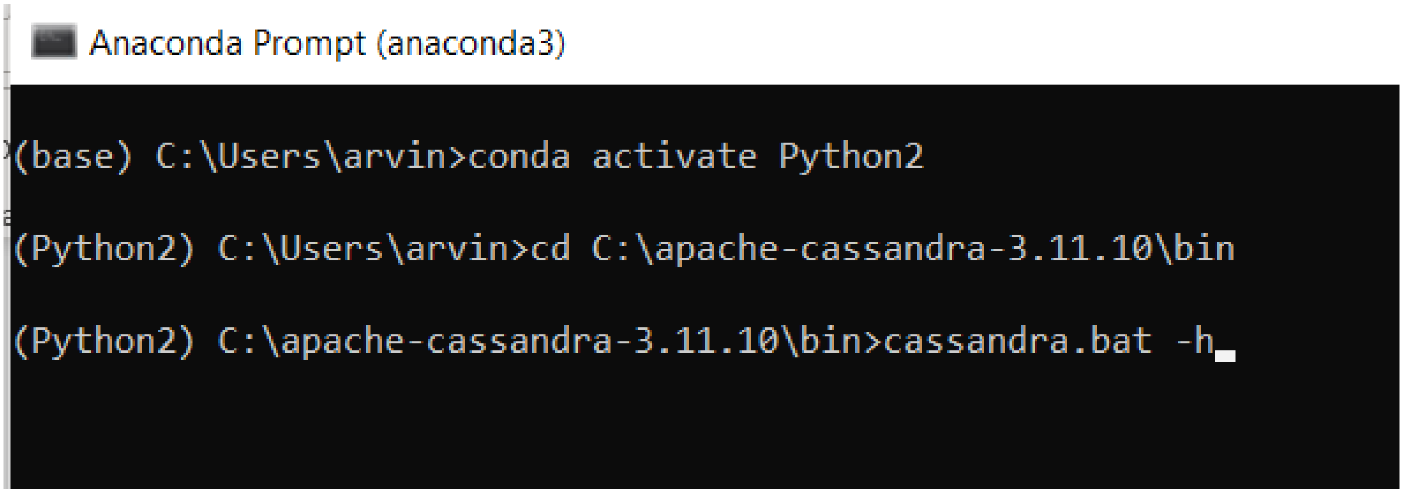
Follow the link https://apachemirror.wuchna.com/thrift/0.14.2/thrift-0.14.2.exe and execute the '.exe' file. This will flash within a second. No installation steps are involved, just execute the file.

With these 4 steps done, we are good to get started with Cassandra.

## Cassandra Query Language Execution

Launch Command Prompt or Anaconda Prompt. Activate the Python 2 environment. It is advisable to launch the prompt in the admin mode. Change the directory to the bin folder within the Cassandra folder extracted.

I have placed the extracted Cassandra in my C drive, so the change directory command looks like:

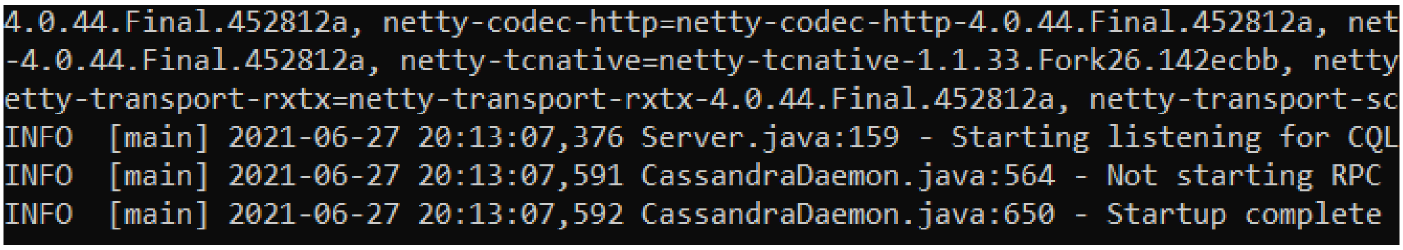


```
(base) C:\Users\arvin>conda activate Python2

(Python2) C:\Users\arvin>cd C:\apache-cassandra-3.11.10\bin

(Python2) C:\apache-cassandra-3.11.10\bin>cassandra.bat -h
```

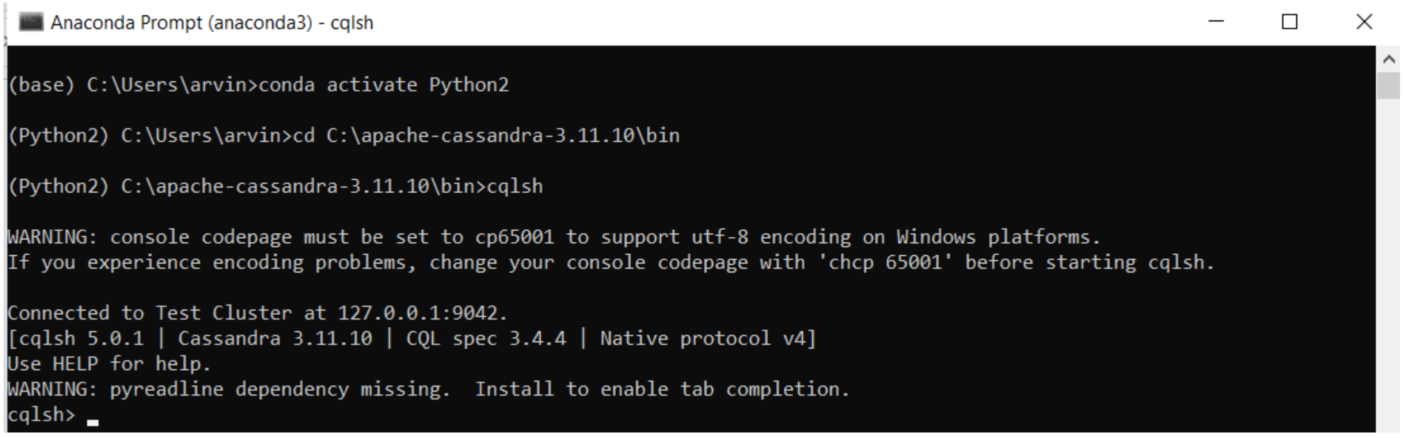
On executing, "cassandra.bat -h", we should get the message, Startup Complete as shown below.



```
4.0.44.Final.452812a, netty-codec-http=netty-codec-http-4.0.44.Final.452812a, net
-4.0.44.Final.452812a, netty-tcnative=netty-tcnative-1.1.33.Fork26.142ecbb, netty
etty-transport-rxtx=netty-transport-rxtx-4.0.44.Final.452812a, netty-transport-sc
INFO  [main] 2021-06-27 20:13:07,376 Server.java:159 - Starting listening for CQL
INFO  [main] 2021-06-27 20:13:07,591 CassandraDaemon.java:564 - Not starting RPC
INFO  [main] 2021-06-27 20:13:07,592 CassandraDaemon.java:650 - Startup complete
```

Once, this is done, keep this prompt open and launch a new prompt, repeat the steps to change the directory to bin. This time, execute 'cqlsh'.

This is exactly how we jump into CQL, the CASSANDRA QUERY LANGUAGE. With successful execution, this is how the prompt should look like.



```
(base) C:\Users\arvin>conda activate Python2

(Python2) C:\Users\arvin>cd C:\apache-cassandra-3.11.10\bin

(Python2) C:\apache-cassandra-3.11.10\bin>cqlsh

WARNING: console codepage must be set to cp65001 to support utf-8 encoding on Windows platforms.
If you experience encoding problems, change your console codepage with 'chcp 65001' before starting cqlsh.

Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
WARNING: pyreadline dependency missing.  Install to enable tab completion.
cqlsh>
```

Congratulations, Cassandra is up and running successfully in your local system as soon as you can notice the cqlsh>

IT WAS LONG, WASN'T IT, REALLY TIME CONSUMING AND CONVOLUTED?

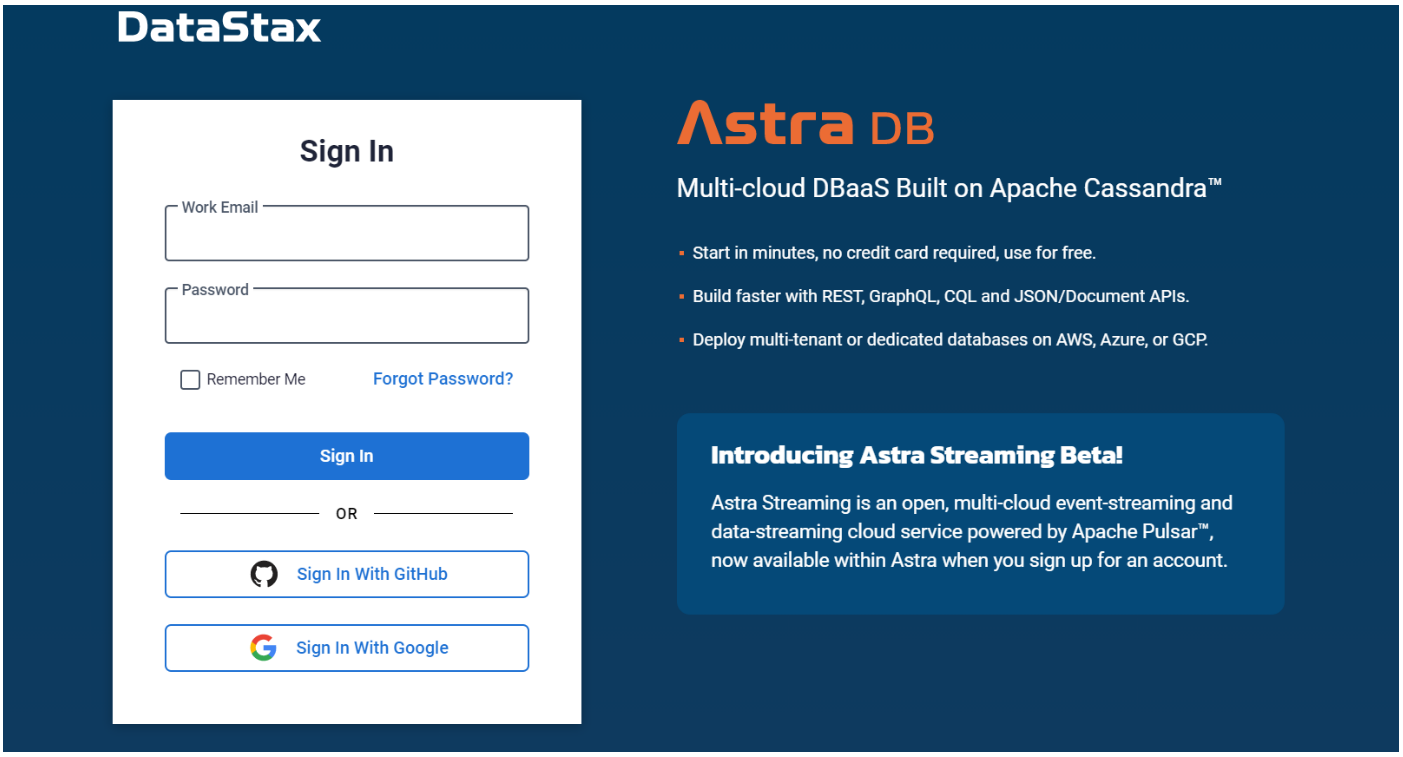
Let us take the cloud path to reach this in no time.

## Cassandra Through DataStax

DataStax, Inc. is a data management company based in Santa Clara, California. Its product provides commercial support, software, and cloud database-as-a-service based on Apache Cassandra. DataStax also provides event streaming support and a cloud service based on Apache Pulsar.

We can create a free account and use their services free of cost worth 25 dollars every month. For the purposes of within the scope of this article and understand Cassandra-Python integration, free version of DataStax is more than enough.

All we need to do is, create an account through the link : https://auth.cloud.datastax.com/auth/realms/CloudUsers/protocol/openid-connect/auth?client_id=auth-proxy&redirect_uri=https%3A%2F%2Fgatekeeper.auth.cloud.datastax.com%2Fcallback&response_type=code&scope=openid+profile+email&state=DCjl128Yg5gwCOgOIgkuNEyfoR8%3D



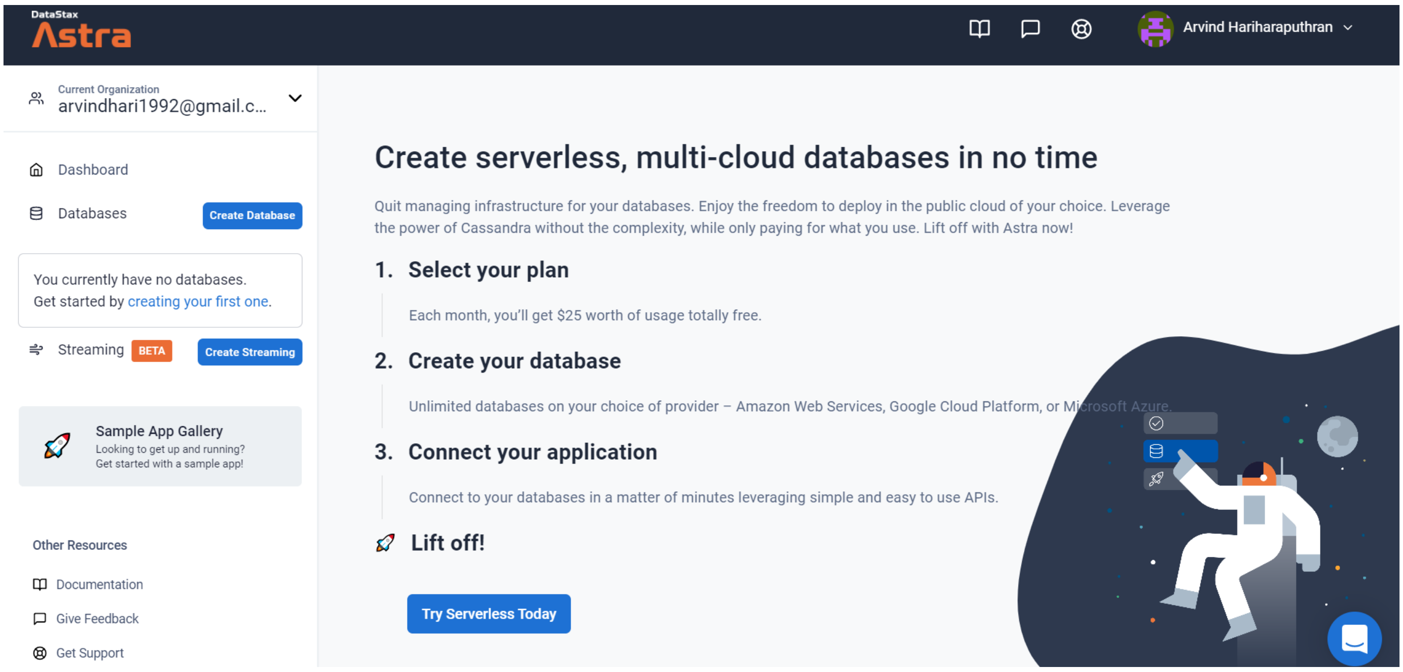
Multi-cloud DBaaS Built on Apache Cassandra™

- Start in minutes, no credit card required, use for free.
- Build faster with REST, GraphQL, CQL and JSON/Document APIs.
- Deploy multi-tenant or dedicated databases on AWS, Azure, or GCP.

**Introducing Astra Streaming Beta!**

Astra Streaming is an open, multi-cloud event-streaming and data-streaming cloud service powered by Apache Pulsar™, now available within Astra when you sign up for an account.

Once logged in, this is what we will be seeing.



You currently have no databases.
Get started by creating your first one.

Create serverless, multi-cloud databases in no time

Quit managing infrastructure for your databases. Enjoy the freedom to deploy in the public cloud of your choice. Leverage the power of Cassandra without the complexity, while only paying for what you use. Lift off with Astra now!

1. Select your plan

   Each month, you'll get $25 worth of usage totally free.

2. Create your database

   Unlimited databases on your choice of provider – Amazon Web Services, Google Cloud Platform, or Microsoft Azure.

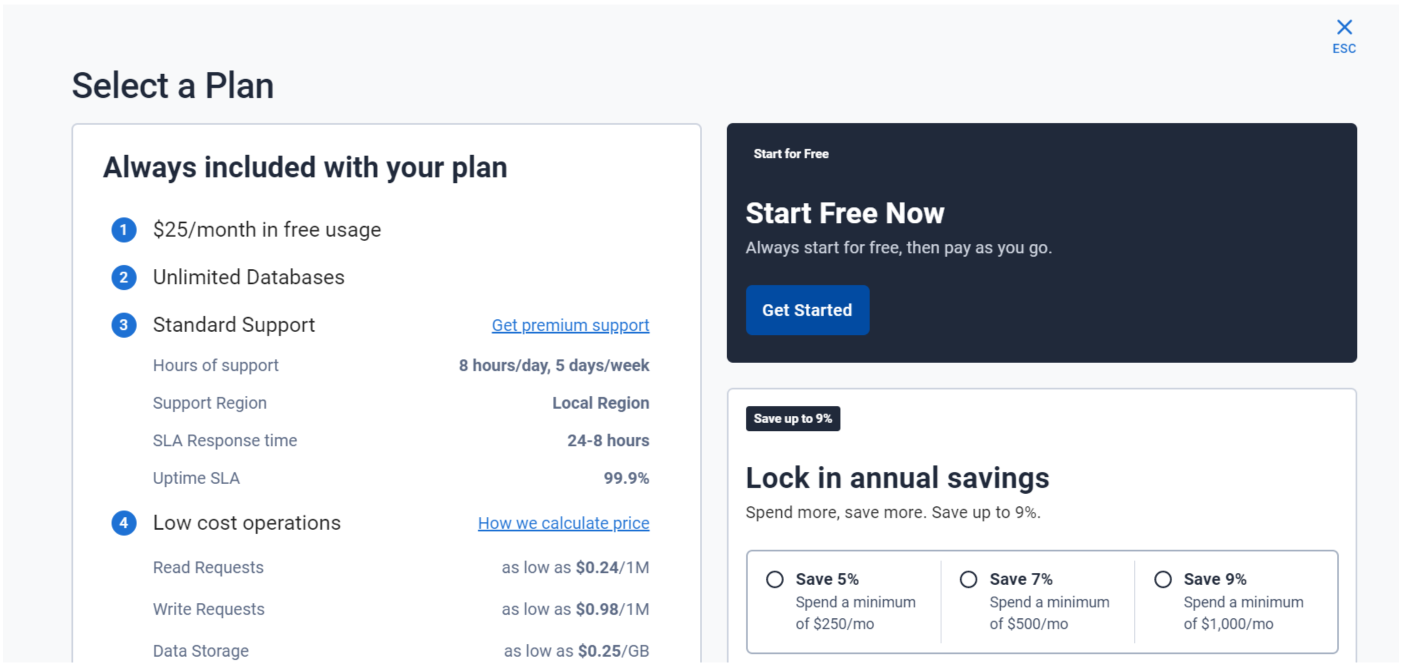
3. Connect your application

   Connect to your databases in a matter of minutes leveraging simple and easy to use APIs.

Lift off!

Click on the Create Database option to proceed with it. Be wise and use the Free Version. I am emphasizing again, the free version is more than enough to get the job done for us.

Click on the Get Started button under "Start Free Now"


Select a Plan

**Always included with your plan**

1. $25/month in free usage
2. Unlimited Databases
3. Standard Support — Get premium support

|  |  |
|---|---|
| Hours of support | 8 hours/day, 5 days/week |
| Support Region | Local Region |
| SLA Response time | 24-8 hours |
| Uptime SLA | 99.9% |

4. Low cost operations — How we calculate price

|  |  |
|---|---|
| Read Requests | as low as **$0.24**/1M |
| Write Requests | as low as **$0.98**/1M |
| Data Storage | as low as **$0.25**/GB |

Start for Free

**Start Free Now**

Always start for free, then pay as you go.

Save up to 9%

**Lock in annual savings**

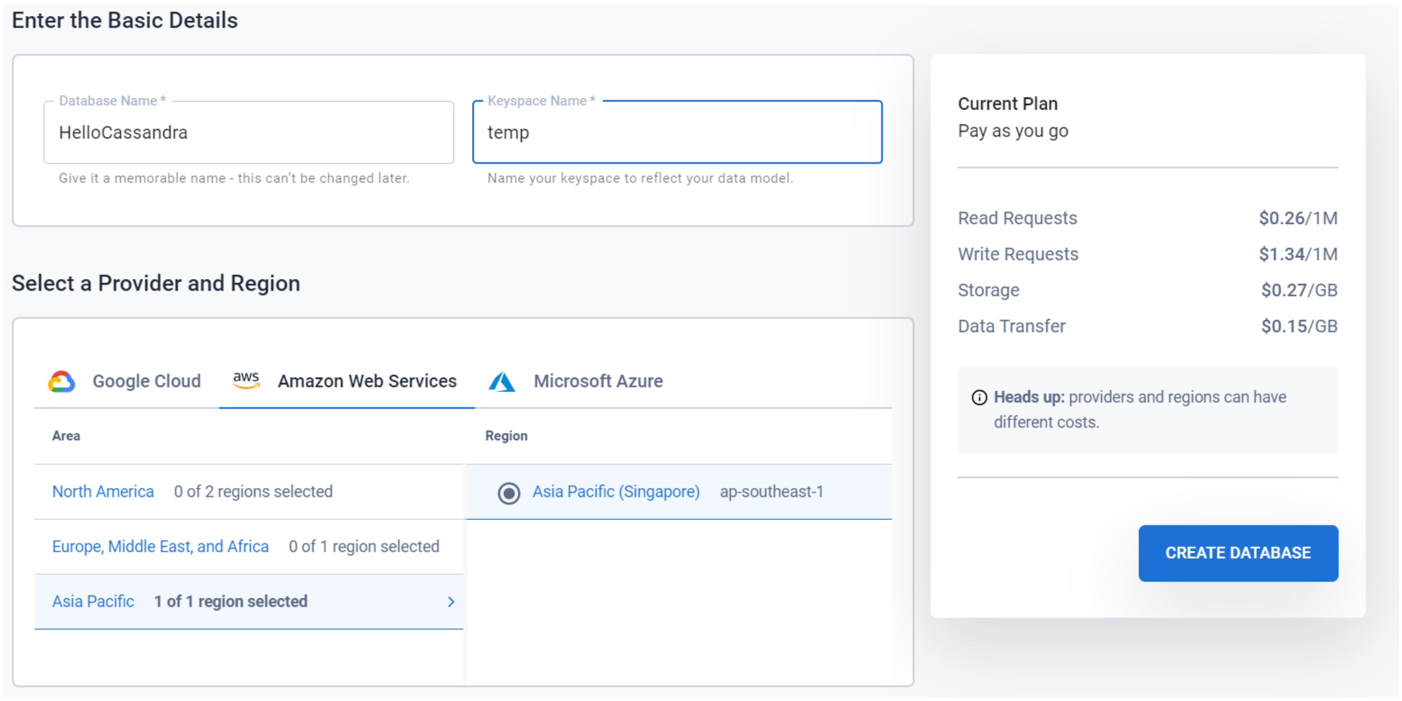
Spend more, save more. Save up to 9%.

- **Save 5%** Spend a minimum of $250/mo
- **Save 7%** Spend a minimum of $500/mo
- **Save 9%** Spend a minimum of $1,000/mo

Now comes the fun part, Cassandra has a slightly different architecture than other databases. Tables are not stored directly inside a database, rather they are stored in keyspaces. In other terms, a table is stored in a keyspace which is stored in a database. To understand more on what a keyspace is, how it differs from a table, follow the link : https://docs.datastax.com/en/astra/docs/db-glossary.html#_keyspace

Choose the cloud and region of your interest, it really does not matter, it is just a matter of luck, how quickly we can get our database created. As we are using shared services through free account, it takes some significant time (a few minutes) to get the Database created, initialised and provide the access.

IMPORTANT, Ensure the KEYSPACE names are completely **lowercase**.


Enter the Basic Details

Database Name *: HelloCassandra
Give it a memorable name - this can't be changed later.

Keyspace Name *: temp
Name your keyspace to reflect your data model.

Select a Provider and Region

Google Cloud | Amazon Web Services | Microsoft Azure

| Area |  |
|---|---|
| North America | 0 of 2 regions selected |
| Europe, Middle East, and Africa | 0 of 1 region selected |
| Asia Pacific | 1 of 1 region selected |

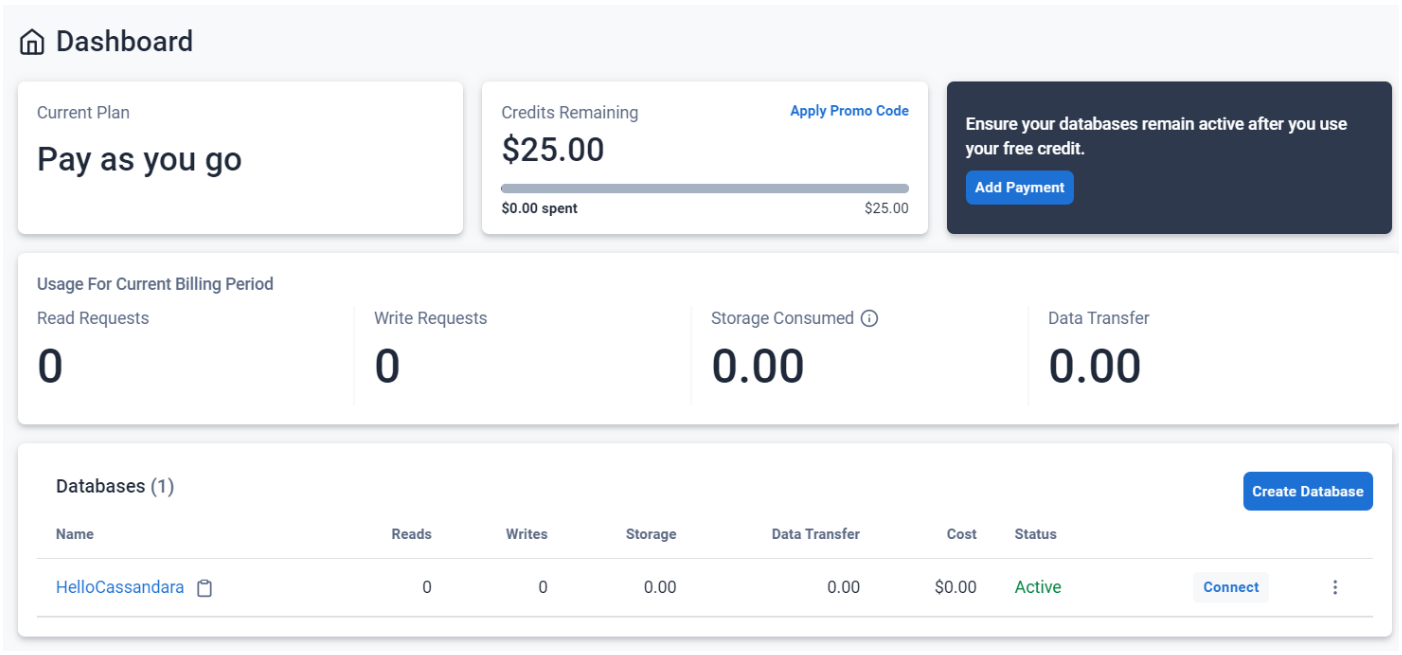
| Region |  |
|---|---|
| ● Asia Pacific (Singapore) | ap-southeast-1 |

Current Plan
Pay as you go

|  |  |
|---|---|
| Read Requests | $0.26/1M |
| Write Requests | $1.34/1M |
| Storage | $0.27/GB |
| Data Transfer | $0.15/GB |

**Heads up:** providers and regions can have different costs.

And, hurray, our new Cassandra Database is now ready. Wait until the Connect button gets activated. It takes a few minutes as we are working through shared servers.


Dashboard

Current Plan
Pay as you go

Credits Remaining
Apply Promo Code
$25.00
$0.00 spent
$25.00

Ensure your databases remain active after you use your free credit.
Add Payment

Usage For Current Billing Period

| Read Requests | Write Requests | Storage Consumed | Data Transfer |
|---|---|---|---|
| 0 | 0 | 0.00 | 0.00 |

Databases (1)
Create Database

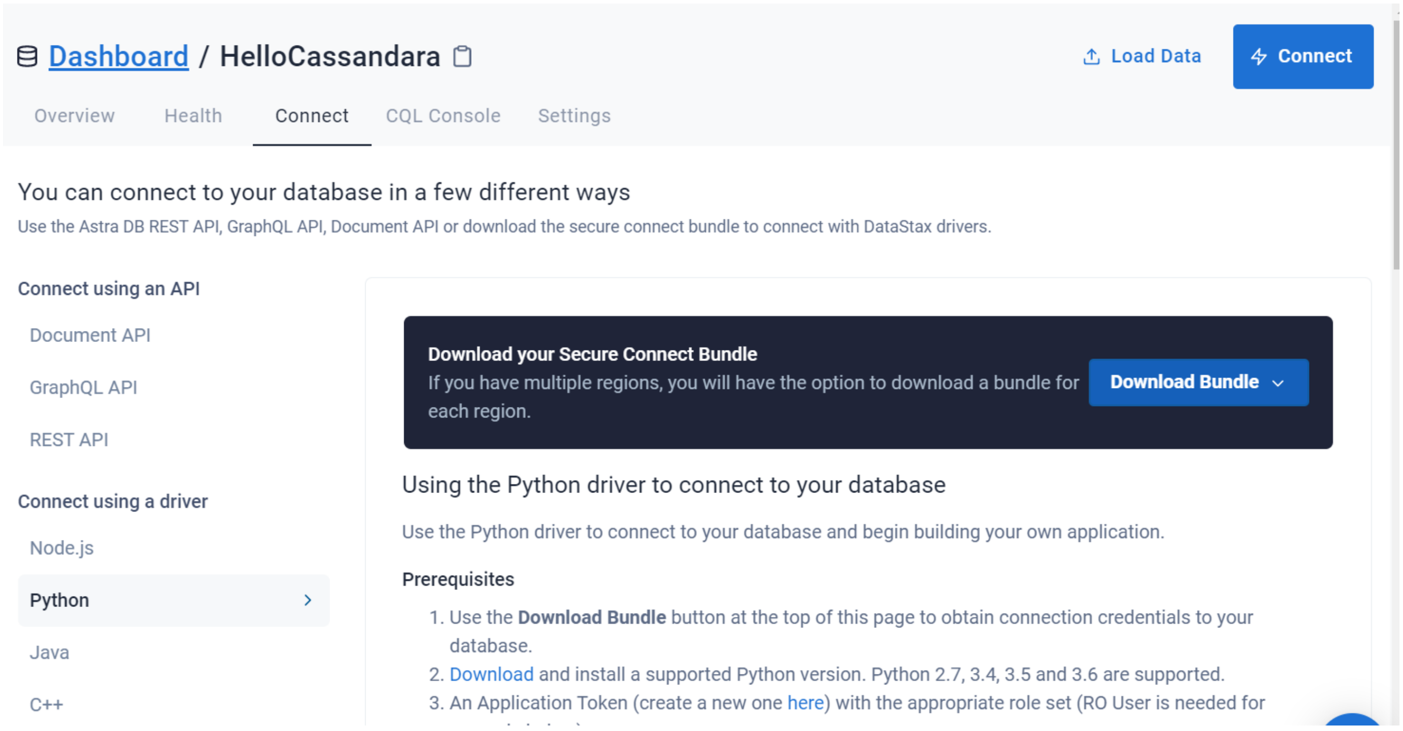
| Name | Reads | Writes | Storage | Data Transfer | Cost | Status | | |
|---|---|---|---|---|---|---|---|---|
| HelloCassandara | 0 | 0 | 0.00 | 0.00 | $0.00 | Active | Connect | ⋮ |

Finally, it is time to access and play with Cassandra Databases through Python.

## CASSANDRA-PYTHON Integration

We will start with discussing integrating the cloud version with Python, hold tight, a few steps process coming up. Cheers as it is a one-time activity per database.

As soon as we click on the new database and got to the Connect tab, we get to see the window as shown below. As we are trying to connect Cassandra with Python, select Python from the list of labels to the left. Then download the bundle from the Download Bundle option. Just download this and do nothing else yet.


Dashboard / HelloCassandara

Load Data
Connect

Overview
Health
Connect
CQL Console
Settings

You can connect to your database in a few different ways

Use the Astra DB REST API, GraphQL API, Document API or download the secure connect bundle to connect with DataStax drivers.

Connect using an API
Document API
GraphQL API
REST API

Connect using a driver
Node.js
Python
Java
C++

Download your Secure Connect Bundle

If you have multiple regions, you will have the option to download a bundle for each region.

Download Bundle

Using the Python driver to connect to your database

Use the Python driver to connect to your database and begin building your own application.

Prerequisites

1. Use the **Download Bundle** button at the top of this page to obtain connection credentials to your database.
2. Download and install a supported Python version. Python 2.7, 3.4, 3.5 and 3.6 are supported.
3. An Application Token (create a new one here) with the appropriate role set (RO User is needed for

Scroll down the page, and reach the bullet 4, which has a code snippet as shown below. This bunch of code is the key to connecting Cassandra with Python. Every single program through which, we are trying to reach the cloud version of Cassandra needs this piece of code. For every new database we create, a similar piece of code will be available.

All we need to do are:

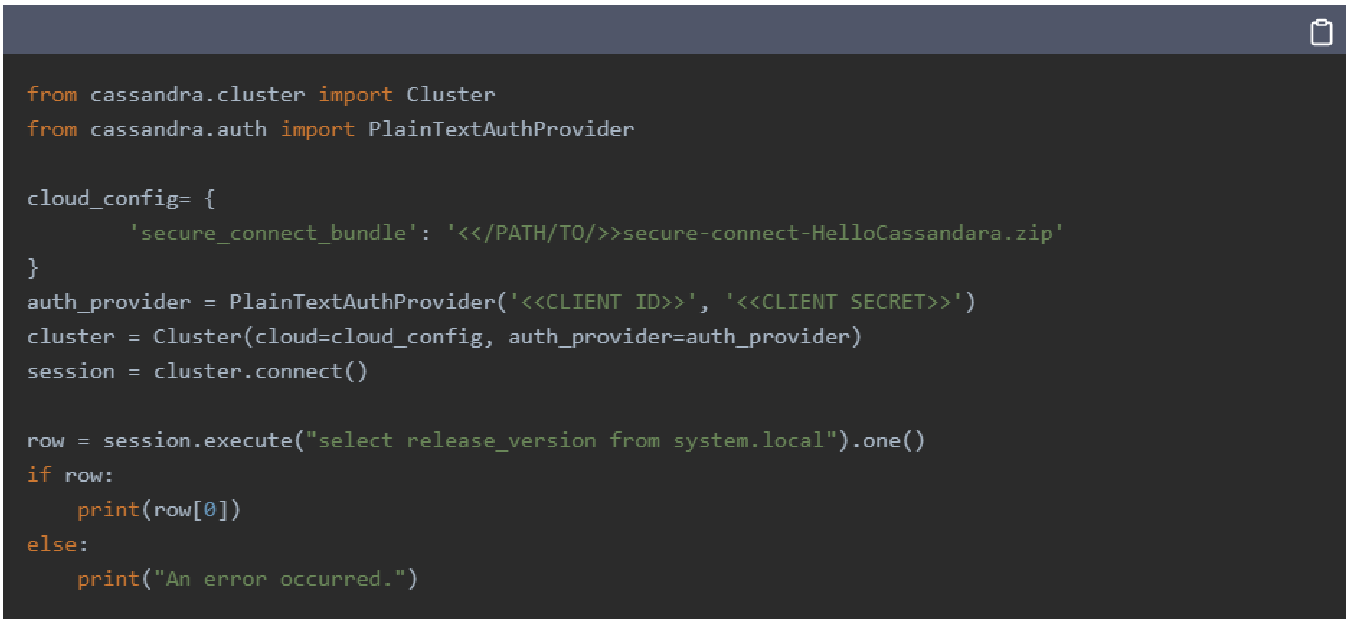
- **Import the CASSANDRA driver using !pip install cassandra-driver**
- Replace the <</PATH/TO/>> with the folder location where the zipped bundle has been downloaded. Do not forget to place a backward slash (/) at the end of folder path.
- Replace the CLIENT ID and CLIENT SECRET with the corresponding tokens from the Token Management option. Just copy sand paste the ID and SECRET into the code.

On executing the code, no errors should be encountered, especially, it should not return "An error occurred." If so congratulations, we have successfully established the connection between cloud version of Cassandra and Python.

All we need is, place the Cassandra Query within the execute().

To integrate the local version of Cassandra with Python, we do not need any authorisation. Hence, just ignore the lines related to cloud_config and auth_provider. Just call the Cluster(<no args>) and create a session.

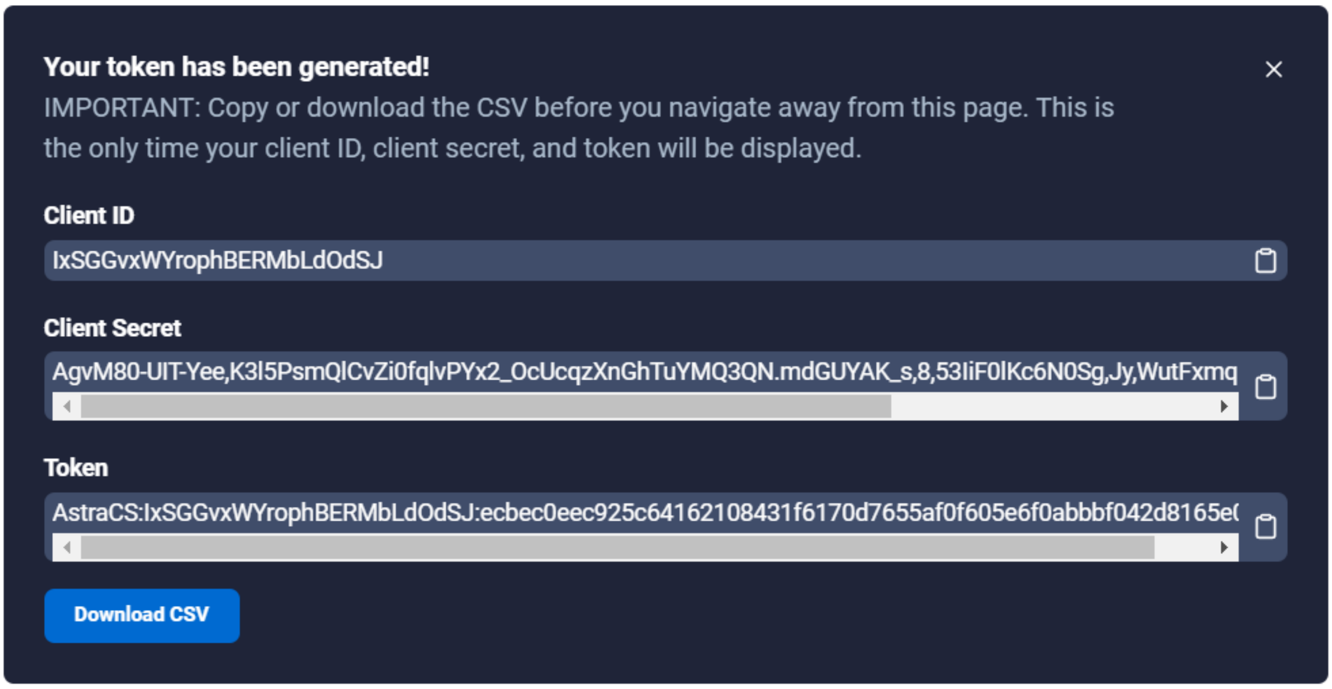
4. Copy the following connection code into the `connect_database.py` file:



```
from cassandra.cluster import Cluster
from cassandra.auth import PlainTextAuthProvider

cloud_config= {
        'secure_connect_bundle': '<</PATH/TO/>>secure-connect-HelloCassandara.zip'
}
auth_provider = PlainTextAuthProvider('<<CLIENT ID>>', '<<CLIENT SECRET>>')
cluster = Cluster(cloud=cloud_config, auth_provider=auth_provider)
session = cluster.connect()

row = session.execute("select release_version from system.local").one()
if row:
    print(row[0])
else:
    print("An error occurred.")
```



**Your token has been generated!**

IMPORTANT: Copy or download the CSV before you navigate away from this page. This is the only time your client ID, client secret, and token will be displayed.

**Client ID**

```
IxSGGvxWYrophBERMbLdOdSJ
```

**Client Secret**

```
AgvM80-UIT-Yee,K3l5PsmQlCvZi0fqlvPYx2_OcUcqzXnGhTuYMQ3QN.mdGUYAK_s,8,53IiF0lKc6N0Sg,Jy,WutFxmq
```

**Token**

```
AstraCS:IxSGGvxWYrophBERMbLdOdSJ:ecbec0eec925c64162108431f6170d7655af0f605e6f0abbbf042d8165e(
```