

About your Astra database

Welcome! Let's cover some basics and review how you can get connected.

[DataStax Astra](#), built on the best distribution of Apache Cassandra™, provides the ability to develop and deploy data-driven applications with a cloud-native service, without the hassles of [database](#) and infrastructure administration.

By automating tuning and configuration, DataStax Astra radically simplifies database operations.

[Limits](#) help enforce best practices by defining parameters for operations, such as [table](#) number and [column](#) size limits, ensuring the database operates with consistent performance and maintains cluster health.

Each Astra database on a paid tier employs a single [capacity unit](#) (CU) by default, which represents three database [instances](#) that are grouped together. A single CU represents three replicas. CUs ensure high availability by implementing a consistent [replication factor](#) that is acceptable for development and production workloads.

To ensure that an Astra database doesn't exceed capacity, you can [expand the database](#) by adding CUs.

Developers can easily [resize resources](#) as their needs grow and change and [park databases](#) when not in use to save costs and pay for only storage.

For more information about DataStax Astra, see [DataStax Astra FAQ](#).

Your paid [database](#) starts the following specifications:

- A single [region](#)
- A single [keyspace](#)
- Storage based on your [service tier](#)
- Capacity for up to 200 tables
- Replication factor of three to provide optimal uptime and data integrity
- One [capacity unit](#) (CU), which represents three database [instances](#) grouped together

Note

If you choose the Free tier, then your database has a 5 GB capacity so you can test Astra and start developing today. Free tier databases also have a [replication factor](#) of one.

You can later [resize the database](#) to increase the number of CUs.

To better understand your database capabilities, review the [DataStax Astra database limits](#).

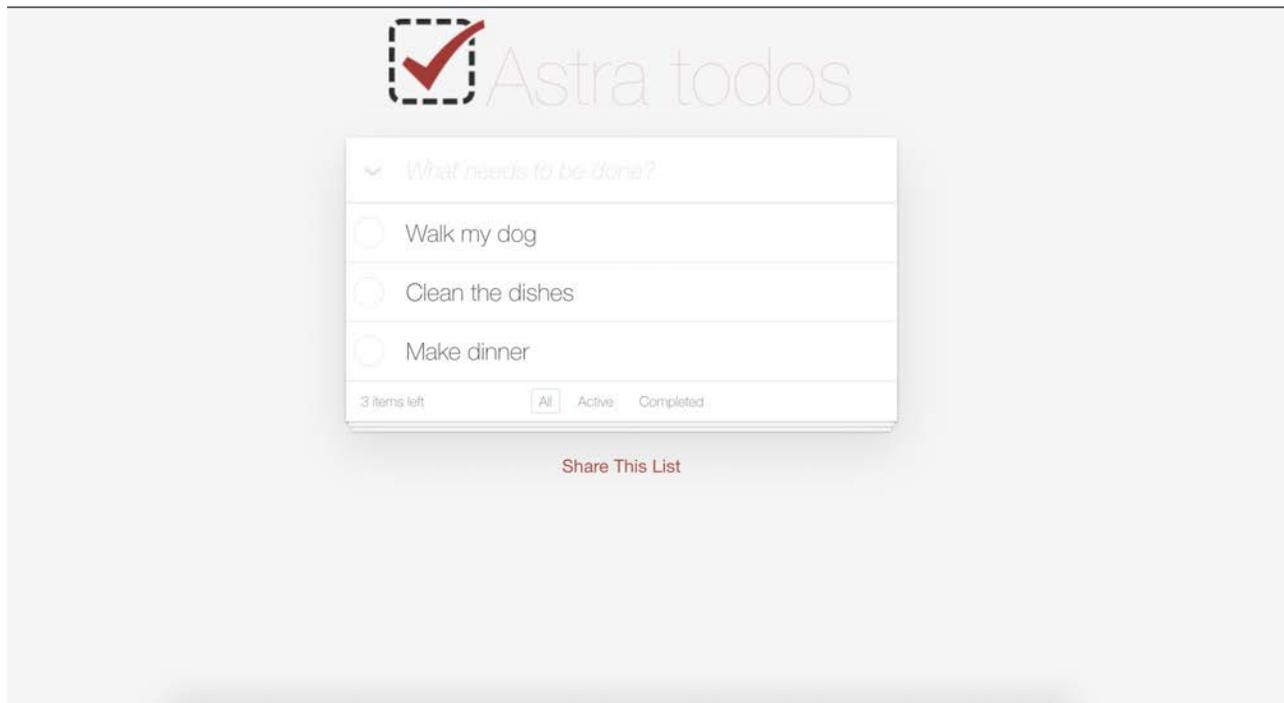
No configuration necessary

Use the integrated DataStax Studio instance, which is preconfigured to work with your Astra database. No configuration required. Even better, we added a DataStax Studio notebook that you can use to create tables in your keyspace, generate data, and learn the Cassandra Query Language (CQL) as you go.

CQL is the primary language for communicating with your Astra database. Use CQL to insert, update, and query data in tables, drop tables, and much more. Like SQL, CQL statements change data, look up data, store data, or change the way data is stored. If you're new to CQL, see the DataStax documentation for understanding the [CQL command syntax](#).

Building sample apps

Simple, sample applications to get started with Astra.

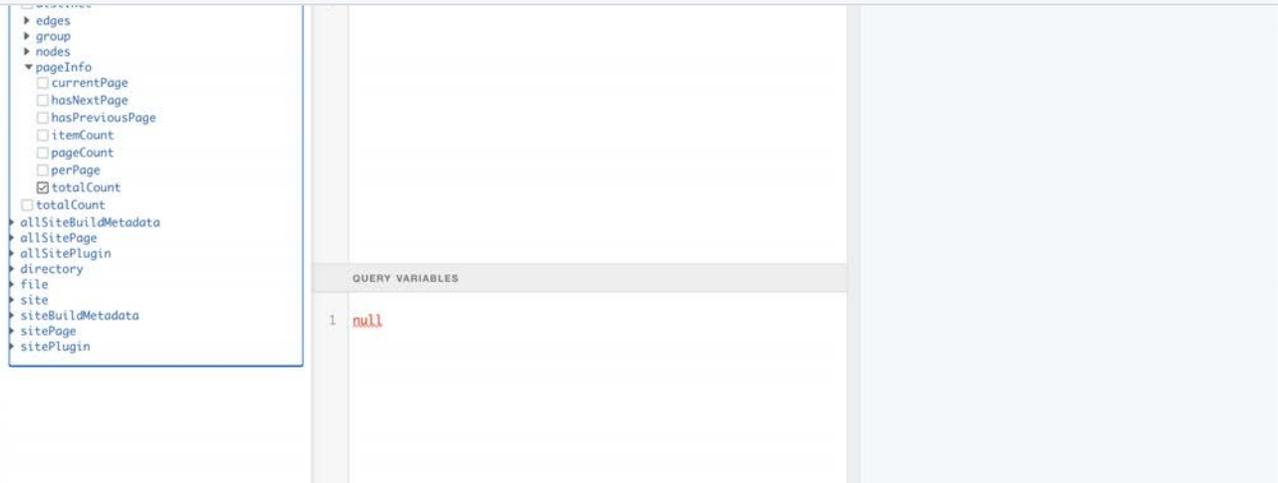


[React.js, JAMStack, REST Todo List](#)

Create a simple React Todo List using just front-end Javascript talking directly to Astra REST API.

🕒 10 minutes, Beginner

Updated on July 8th, 2020

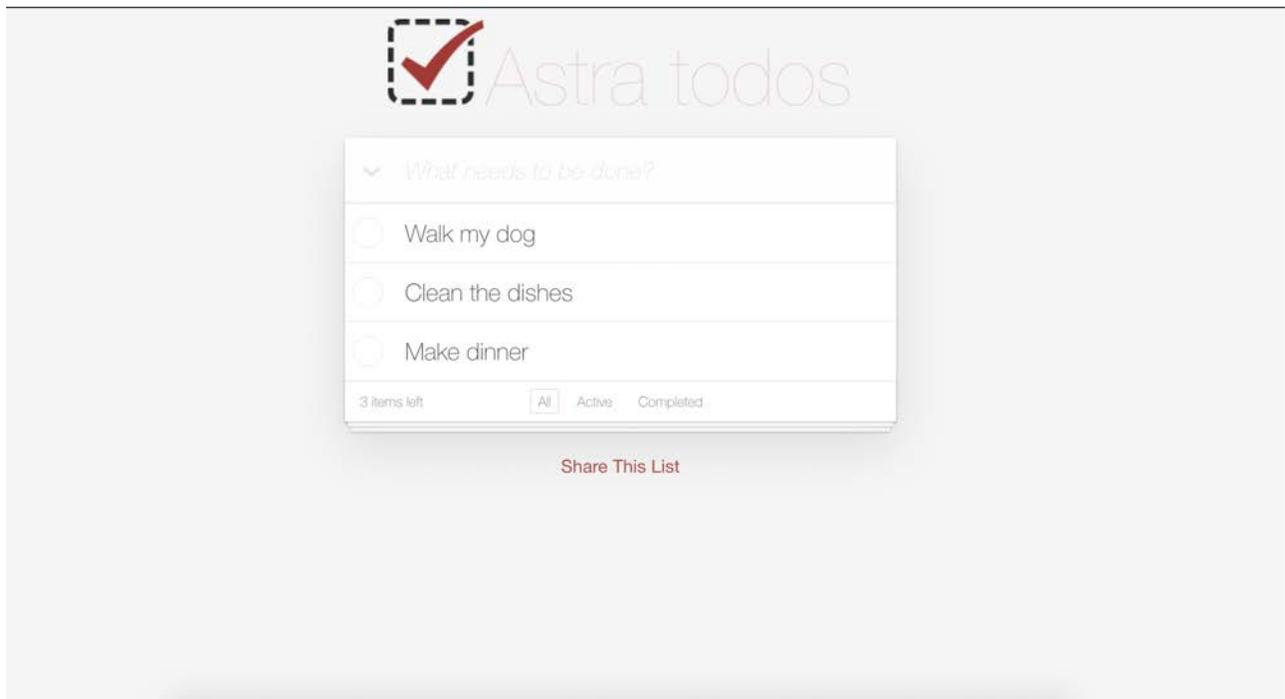


React.js, Gatsby.js, GraphQL Starter

Create a React.js, Gatsby application that reads data from Astra using GraphQL.

🕒 30 minutes, Beginner

Updated on July 5th, 2020



Node.js, React.js, Serverless Todo List

Create a Serverless React Todo list following AWS best practices including Lambda.

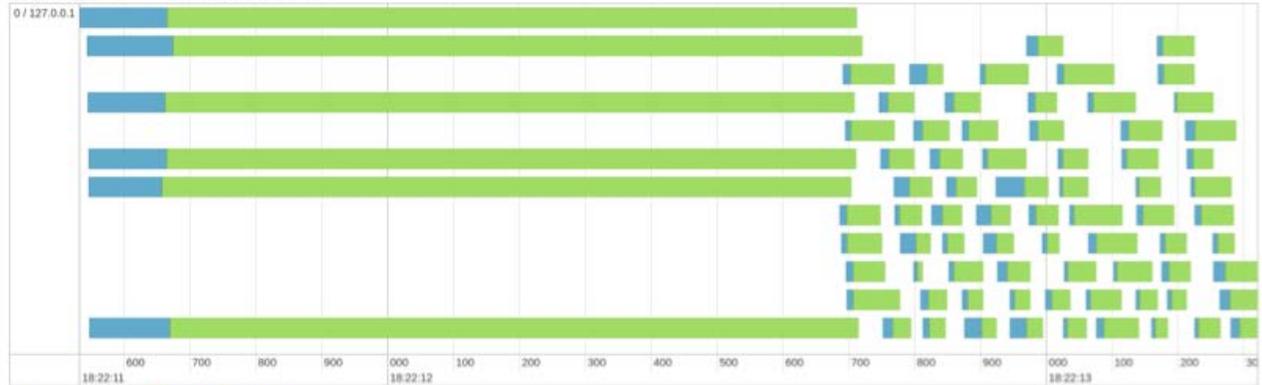
🕒 30 minutes, Intermediate

Details for Stage 0 (Attempt 0)

Total Time Across All Tasks: 9 s
Locality Level Summary: Node local: 78

- ▶ DAG Visualization
- ▶ Show Additional Metrics
- ▶ Event Timeline
- ☐ Enable zooming

- Scheduler Delay
- Executor Computing Time
- Getting Result Time
- Task Deserialization Time
- Shuffle Write Time
- Result Serialization Time
- Shuffle Read Time



Summary Metrics for 0 Completed Tasks

No tasks have reported metrics yet.

▶ Aggregated Metrics by Executor

Astra/Spark Migration Tool

Migrate data from an existing Cassandra cluster to Astra using a Spark application.

🕒 40 minutes, Expert

Updated on July 20th, 2020

Spring Data Cassandra REST API ^{1.0.0}

[Base URL: localhost:8081/]
<http://localhost:8081/v2/api-docs?group=orders>

Sample REST API for Orders: powered by DataStax Astra

Terms of service
DataStax Examples - Website
Send email to DataStax Examples
Apache v2.0

Order Entity Simple Cassandra Repository

GET /orders findAllOrder

POST /orders saveOrder

GET /orders/search/name-and-price-only findProductNamesAndPricesFromOrderOrder

order-controller Order Controller

GET /orders/{id} getOrder

PUT /orders/{id} updateOrder

DELETE /orders/{id} deleteOrder

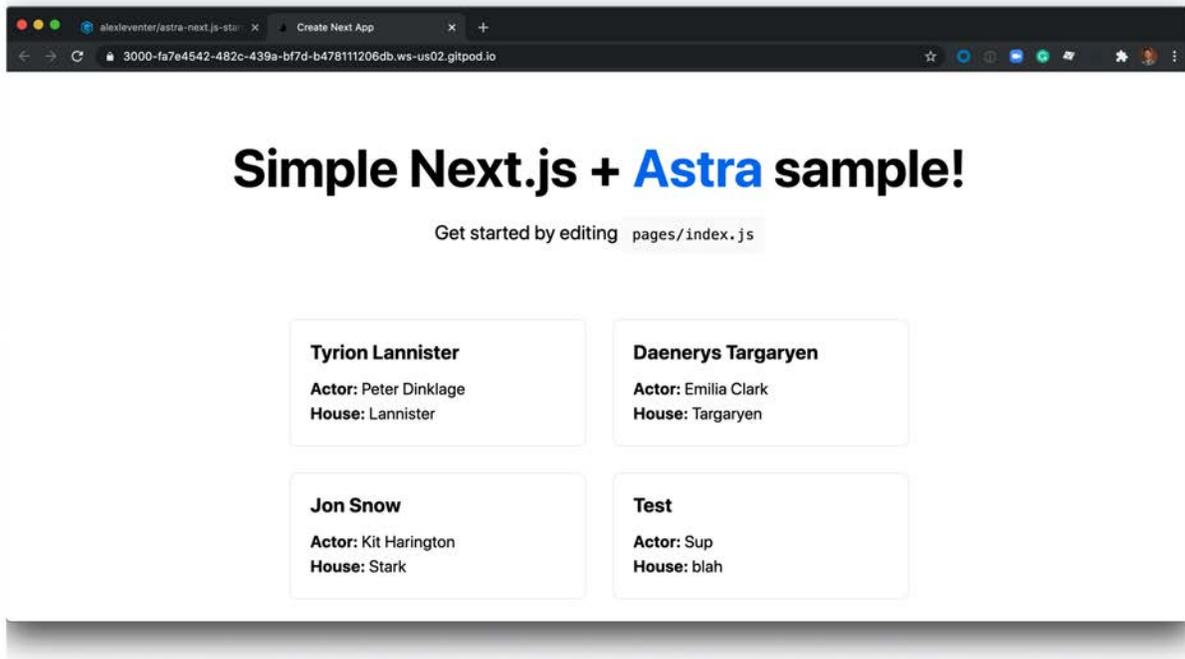
PATCH /orders/{id} patchOrder

GET /orders/{oid}/{pid} getProductInOrder

DELETE /orders/{oid}/{pid} deleteProductFromOrder

🕒 10 minutes, Beginner

Updated on Aug 7th, 2020



Next.js/Astra Starter

Create a React.js, Next.js application that reads data from Astra using GraphQL.

🕒 10 minutes, Beginner

Updated on Aug 9th, 2020



DataStax Astra FAQ

Frequently asked questions about DataStax Astra.

Opening the box

What tools does DataStax Astra include?

- **Astra REST API:** Use the [DataStax Astra REST API](#) to create an [authorization token](#) and begin interacting with your [database](#).
- **CQLSH:** Each Astra database includes an integrated [CQL shell](#) for interactive Cassandra Query Language (CQL) commands. Use CQLSH to create tables, insert data, and modify data in your [keyspace](#).
- **DataStax Developer Studio:** Each Astra database embeds a [DataStax Developer Studio](#) instance that is automatically configured for interactive CQL commands. Kickstart development with the included Developer Studio notebooks.
- **DataStax Bulk Loader:** Use DataStax Bulk Loader (dsbulk) to [load and unload](#) data into your Astra database. You can load data from a variety of sources and unload data from DSE or DDAC databases for transfer, use, or storage of data.
- **Metrics dashboards:** Use the integrated metrics dashboards to [view health metrics](#) that include information regarding latency and throughput to the database. These metrics provide insights into the performance of the database and how workloads are distributed.

Making comparisons

Use this FAQ to find answers to common questions and get help for DataStax Astra.

How is DataStax Astra different from DSE?

While you can run DataStax Enterprise (DSE) both on premises and in the cloud, DataStax Astra allows you to create Astra databases in minutes, with reduced complexity, fewer operational concerns, and an approachable entry point for developing your applications. DataStax Astra is a true database-as-a-service (DBaaS) offering that provides simplified cluster management with flexible pricing. Also, DSE includes Advanced Workloads (DSE Search, Analytics, and Graph), which are not available in DataStax Astra.

Assessing compatibility

Which version of Apache Cassandra™ is compatible with DataStax Astra?

Apache Cassandra version 3.11 is compatible with Astra databases. Additionally, each of the supported DSE and Apache Cassandra drivers is compatible. See [Which drivers work with DataStax Astra](#).

Which cloud vendors does DataStax Astra support?

DataStax Astra supports creating databases on Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure.

Which cloud regions are currently available?

DataStax Astra supports single-region deployments only. These regions vary based on the selected cloud provider and more are added over time. The regions that are currently available can be seen on the [create database screen](#). Within a [region](#), data is replicated across three availability zones to ensure high availability.

You can create up to 5 databases per region, and add a total of 20 capacity units (CUs) per database (10TBs). To adjust the [capacity unit limits](#) or request new regions contact [DataStax Support](#).

Is VPC peering possible?

[VPC peering](#) is available on C- and D-tier Astra databases. For more, see [Connect with AWS VPC peering to your Astra database](#), [Connect with GCP VPC peering to your Astra database](#), and [Connect with Azure VPC peering to your Astra database](#).

Getting connected

How do I connect to DataStax Astra databases?

Use the the Cassandra Query Language SHell (CQLSH), DataStax Studio, or the DataStax drivers to [connect to your Astra database](#).

Which DataStax drivers work with DataStax Astra?

Choose the DataStax driver for your preferred language and configure it to connect to your Astra database. See [Connecting to Astra databases using DataStax drivers](#)

Paying for services

How do I manage billing for DataStax Astra?

DataStax Astra handles billing through an integration with Stripe, and displays all related billing information on the **Billing Summary** page. Enter a credit card number and associated billing information to begin creating Astra databases. See [Billing](#) for more information.

How is pricing calculated for each Astra database?

Pricing for DataStax Astra is based on [capacity unit \(CU\)](#) and Service Tier. There's also a Free tier, which includes 5 GB of storage and is free.

See [Pricing](#) for more information.

Getting help

Where can I find help?

Use this documentation to get answers about what your database is capable of, or for guided information like connecting to your databases.

Practice using the DataStax Astra database with the [Better Botz blog series](#) or explore our [Katacode scenarios](#), which will walk you through setting up your database, creating keyspaces, and managing orders and products using CQL and GraphQL.

To send us feedback or get additional help, use the chat feature in the DataStax Astra console.