DATASTAX

DATASTAX ASTRA DB | IA Generativa

O que você precisa saber sobre o Astra DB

Vector Search as-a-service

Neste documento, resumimos os principais pontos a considerar durante a avaliação do Astra para aplicações profissionais voltadas a IA Generativa com as referências para a documentação pública e códigos de exemplo.

Aspectos Técnicos

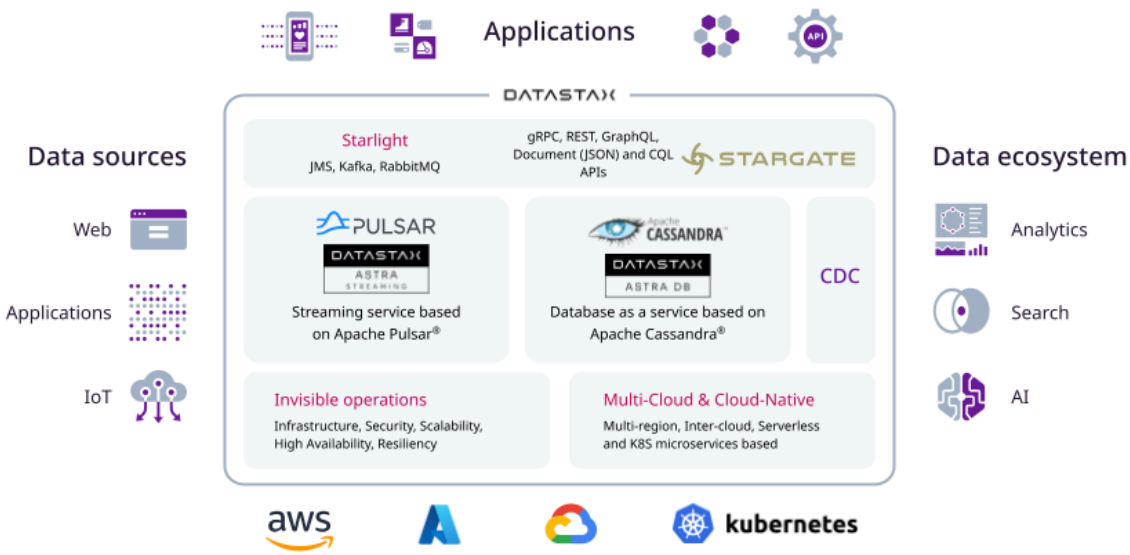
O que são o Astra DB Serverless, Astra DB Vector e Astra Streaming?

Astra DB Serverless é um banco de dados NoSQL que permite aos clientes armazenar dados em um formato compatível com Mongo ou Apache Cassandra.

Astra DB Vector é um plataforma de dados que oferece a capacidade de armazenar elementos de dados multimodais como representações numéricas chamadas vetores. Vetores são matrizes de pontos flutuantes que quando gerados por modelos de linguagem são chamados de embeddings, pois carregam uma representação semântica do conteúdo original. Assim, eles podem ser usados para pesquisas semânticas e enriquecem prompts submetidos aos LLMs.

Astra Streaming é um serviço de streaming de dados compatível com Apache Pulsar, Kafka, RabbitMQ e JMS.

Juntas, essas tecnologias fornecem uma plataforma para desenvolvedores de aplicativos e operação em produção.



Ref:

https://cdn.sanity.io/files/bbnkhnhl/production/325d36f40acc590b146ea9d77442cf4ed1da9eac.pdf

https://www.datastax.com/guides/what-is-vector-search

https://www.datastax.com/products/datastax-astra

https://docs.datastax.com/en/streaming/astra-streaming/index.html

Como desenvolver aplicações com o AstraDB

De maneira geral, todas as linguagens que possuam driver Cassandra podem usar o Astra como banco de dados. Então, Java, Python, NodeJS, C++, C# podem ter o acesso a partir do driver de modo persistente.

https://awesome-astra.github.io/docs/pages/develop/

No entanto, estas e outras linguagens podem utilizar também a camada APIs para interagir com o Astra. Estas APIs REST, GraphQL e gRPC não geram custo adicional e podem ser usadas livremente.

https://awesome-astra.github.io/docs/pages/develop/api/rest/

Aplicações focadas em IA Generativa desenvolvidas com Python podem usar a library Astrapy para acelerar o desenvolvimento:

https://github.com/datastax/astrapy

Há ainda a compatibilidade entre modelos de dados JSON através da Data API. Esta API é compatível com o Mongoose, package NodeJS para acesso ao MongoDB.

https://docs.datastax.com/en/astra/astra-db-vector/api-reference/data-api.html

https://docs.datastax.com/en/astra/astra-db-vector/api-reference/data-api-with-mongoosejs.html

Conexão com AstraDB

Há diferentes maneiras de se conectar ao Astra que dependem fundamentalmente do caso de uso e linguagem utilizada.

Clients: Criados para linguagens específicas e abstraem operações comuns, acelerando o desenvolvimento de aplicações. É recomendado para projetos novos

Driver: Conexão através de drivers recomendada para migração de projetos legados ou requisitos de esquema de dados específico.

A autenticação entre aplicações e o Astra será autenticada com um par de Client ID e Secret, junto à utilização do Secure Connection Bundle, um arquivo zip que contém certificados que garantem a comunicação end-to-end.

Há ainda a opção de acesso via APIs é realizado por meio de tokens de segurança. Este método não demanda utilização de driver e tampouco gera custos adicionais.

https://docs.datastax.com/en/astra/astra-db-vector/databases/connection-methods-comparison.html

Uso de framework para o desenvolvimento (ex Spring / Node.JS / Outros)

Sim, a utilização com frameworks é bastante comum e há integração com diversos frameworks.

No Awesome Astra há diversos exemplos:

Spring: https://awesome-astra.github.io/docs/pages/develop/frameworks/spring/

Javascript: https://awesome-astra.github.io/docs/pages/develop/languages/javascript/

Avaliação de "IDE" para acesso ao Cassandra

No console do Astra há uma aba de "Console" que permite a execução de comandos CQL sobre o banco de dados.

Como ferramentas, além do CQLSH (Client para linha de comando) e Astra CLI, há outras opções, sendo as mais comuns:

DataStax Data Studio: https://www.datastax.com/dev/datastax-studio

DBeaver: https://awesome-astra.github.io/docs/pages/data/explore/dbeaver/

IA Generativa

O que é o RAGStack?

O RAGStack é uma biblioteca para aplicações orientadas ao Retrieval Augmented Generation , ou RAG, pronta para produção que aproveita LangChain e LlamaIndex. Nosso objetivo é fornecer aos desenvolvedores uma biblioteca consistente para aplicativos RAG que os coloque no controle da evolução para novas funcionalidades. Em vez de ter que acompanhar as inúmeras mudanças em técnicas e bibliotecas, você tem um único fluxo, para poder se concentrar na construção de seu aplicativo. Você pode usar o RAGStack hoje para incorporar as melhores práticas para LangChain e LlamaIndex prontas para uso; avanços como o ColBERT chegarão ao RAGstack nos próximos lançamentos.

https://www.datastax.com/products/ragstack

Quais as medidas de similaridade disponíveis no Astra?

As medidas disponíveis são coseno, dot-product e euclidean. É importante considerar a normalização dos vetores ao definir a métrica a ser utilizada. Há ainda diferenças de performance a serem consideradas.

Ref: https://docs.datastax.com/en/astra/astra-db-vector/get-started/concepts.html#metrics

Qual o algoritmo utilizado para o Vector Search?

O índice utilizado no Astra é o JVector, uma implementação Java do DiskANN, considerado o state-of-the-art para este caso.

A primeira versão de Vector Search no Astra foi criada a partir do HNSW, mas encontramos benefícios ao migrar para o JVector/DiskANN, como performance e eficiência computacional.

Referências:

JVector: https://github.com/jbellis/jvector

DiskANN: https://suhasjs.github.io/files/diskann_neurips19.pdf

Como é feita a indexação dos embeddings?

A indexação no Astra é feita de modo síncrono. Ou seja, a indexação ocorre no momento da gravação, com os dados disponíveis para consulta imediatamente após a sua gravação. Isto é importante para aplicações que necessitam de acesso ao documento em tempo real.

O Astra utiliza ainda Storage Attached Index, ou SAI, um mecanismo de índice otimizado para grandes volumes de dados.

Ref:

https://docs.datastax.com/en/astra/astra-db-vector/get-started/concepts.html#indexing

É possível realizar consultas baseadas em coordenadas de geolocalização com o Vector Search?

Sim, este é um caso que o Vector Search atende com alta performance e escala. Basta criar um campo do tipo vector<float,2>, o respectivo índice com métrica "euclidean" e utilizar a função GEO_DISTANCE nas consultas.

Ref:

https://www.datastax.com/blog/hornet-metadata-vector-search

É possível realizar buscas híbridas?

Sim. Há diferentes possibilidades de combinar o vector search com outros tipos de filtros::

- Busca baseada em metadados: Sim, é possível armazenar pares de chave/valor em um campo e aplicar filtros nestes valores antes da busca por similaridade. Esta é, inclusive, a melhor opção para otimização dos custos e performance.

  https://cassio.org/frameworks/langchain/qa-vector-metadata/#metadata-filtering-in-question-answering

  https://github.com/datastax/astrapy?tab=readme-ov-file#53-find-one-and-update-with-vector-search

- Busca baseada em texto: Sim. Este é um tipo de operação que tem um operador específico ":" e permite a busca por textos. Há opções para configuração de tokenização e analyzers específicos por idioma (incluindo o Português Brasileiro)

  https://docs.datastax.com/en/astra/astra-db-vector/cql/use-analyzers-with-cql.html

- Busca baseada em chaves de partições, agrupamento ou colunas indexadas: Estas opções derivam da modelagem clássica do Cassandra e portanto seguem disponíveis. Ao modelar os dados, deve-se levar em conta as queries que a aplicação fará e com isso conseguimos direcionar ao melhor modelo de dados.

Todas estas opções podem ser combinadas com Vector Search.

É possível aplicar re-ranking nos resultados?

Atualmente, diretamente no Astra, não. O que temos visto é a combinação dos resultados do banco de dados e pós ordenação. Frameworks como LangChain e LlamaIndex (que fazem parte do Ragstack) possuem métodos e integrações para o re-ranking.

Outra técnica que pode ser considerada na melhoria dos resultados é o MMR (Maximal Marginal Relevance) que está incluída no CassIO.

https://cassio.org/frameworks/langchain/qa-maximal-marginal-relevance/

O Cohere é uma opção que, de acordo com benchmarks, tem tido melhor performance ao classificar os documentos após a busca semântica. O ColBERT é outra opção que gera ótimos resultados

Aqui um exemplo da utilização do ColBERT no Astra:
https://thenewstack.io/overcoming-the-limits-of-rag-with-colbert/

Como os dados são modelados e como isso afeta a performance ?

O Astra deriva do Apache Cassandra, um banco de dados NoSQL voltado à performance de escrita e leitura com alto volume e replicação em múltiplos data-centers, o que resulta em velocidade e resiliência. O modelo de dados é também orientado aos mesmos objetivos.

Por exemplo, o Cassandra/Astra tem por conceito direcionar à desnormalização dos dados em detrimento de relacionamento de entidades e integridade referencial. Isto porque é mais eficiente, sob o ponto de vista de custo e performance, consumir mais armazenamento e reduzir o consumo de processamento para combinação de um grande volume de dados (ou seja, replicando dados e evitando o processamento de joins).

Nestes casos, a recomendação é: Grave junto o que é retornado junto. Isto economiza recursos e acelera a aplicação.

Como o Astra suporta casos de uso relacionados à busca de imagens e vídeos?

A abordagem do Astra nestes casos é utilizar modelos para extração de textos ou imagens e, a partir da utilização de modelos multimodais, encontrar conteúdos por similaridade. Com estes conteúdos particionados e convertidos para embeddings, a busca por similaridade resolve grande parte dos casos.

Alguns exemplos:

- Extração de conteúdos de vídeos do Youtube, conversão em embedding e busca híbrida com CQL:

  https://github.com/smatiolids/astra-vector-notebooks/blob/main/YouTube_Knowledge_Base_Text_and_Similarity_Search.ipynb

- Busca multimodal (texto + imagem) voltada a ecommerce com Astra e Gemini Pro.

  https://github.com/smatiolids/ecommerce-astra-genai

  IA Generativa para ecommerce com Gemini PRO Vision e Vector Search

Qual a performance do Astra?

Ao avaliar a performance de bancos de dados de vetores precisamos ir além das métricas tradicionais: latência e throughput. Os bancos de dados com Vector Search devem também oferecer resultados adequados para o vetor de entrada. É o que chamamos de relevância. Quanto mais o BD retornar resultados relevantes, melhor a qualidade dos textos gerados pelos modelos.

Um estudo conduzidos pela GigaOM comparando Astra e o Pinecone (p2x8) indica que:

- Throughput

    - O Astra indexa os dados até 6x mais rápido
    - A ingestão e indexação de dados é até 9x mais rápida no Astra DB.
- Latência
    - O Astra retorna dados até 6x mais rápido durante a indexação
    - O Astra retorna dados até 74x mais mais rápido, considerando P99, durante a ingestão e indexação
- TCO
    - O Astra tem custo total de operação (TCO) 55% a 80% menor
- Relevância
    - A relevância dos documentos retornados pelo Astra é até 20% mais alta (F1-Recall)

O relatório completo pode ser visto aqui: https://bit.ly/48soBnP

Excelência Operacional

O Astra pode ser executado em quais provedores de Cloud?

Bancos de dados Astra podem ser criados na AWS, Azure e GCP. Diversas regiões estão disponíveis.

Os dados gravados no Astra possuem réplicas?

Sim. Os bancos de dados criados no Astra por padrão são replicados em 3 zonas de disponibilidade na região selecionada. Isto garante a disponibilidade de 99.99%

Também é possível replicar os dados em múltiplas regiões do mesmo provedor de nuvem. Então os mesmos dados ficam disponíveis, por exemplo, na AWS São Paulo e AWS US-EAST-1. Este é mais um dos benefícios nativos do Apache Cassandra que pode ser utilizado para aumentar a disponibilidade do ambiente para 99.999%

O Astra pode ser executado On-premise?

Não, o Astra é uma plataforma de dados para utilização em nuvem. No entanto, o DSE (DataStax Enterprise) é nossa versão para execução on-premise, podendo inclusive fazer deployment de ambientes baseados em Kubernetes..

https://www.datastax.com/products/datastax-enterprise

Como é feito o sizing do Astra?

Não é necessário calcular o sizing do ambiente ou provisionar recursos.

Como o Astra é serverless e sua operação é toda automatizada com Kubernetes, o provisionamento de recursos é feito pela DataStax conforme a demanda da aplicação.

Todo este processo é transparente para o usuário.

Isto ainda nos permite ter um modelo de cobrança orientado a utilização da plataforma e não ao provisionamento de recursos e garante ao usuário a melhor utilização da solução sob o ponto de vista financeiro e de performance.

Quais conhecimentos o time de SRE precisa ter para operar o Astra?

Não é necessário ter conhecimento do Astra, Cassandra, Kubernetes ou outras tecnologias internas para utilização do Astra. Todas as operações de escalonamento, atualizações, backups, etc... são realizadas pelo time da DataStax sem impacto para o usuário.

O time de SRE poderá monitorar as métricas de performance da aplicação através de suas ferramentas. Estas métricas podem ser exportadas para o Grafana, Kafka, AWS CloudWatch, Splunk e Datadog.

https://docs.datastax.com/en/astra/astra-db-vector/databases/metrics.html

Como o deployment do Astra é executado?

Há diferentes opções.

- Dashboard: A maneira mais simples de provisionar uma instância do Astra é via o Astra Dashboard, onde são definidos o nome do BD, Cloud e região de execução. Todas as operações podem ser feitas diretamente no dashboard

- DevOps API: Para automatizar a criação de ambientes através de código, a DevOps API pode ser utilizada.

  https://docs.datastax.com/en/astra-serverless/docs/_attachments/devopsv2.html

- Terraform: Empresas que já utilizam Terraform para realizar o provisionamento podem utilizá-lo com o Astra também.

  https://registry.terraform.io/providers/datastax/astra/latest/docs

Qual a estrutura de suporte do Astra?

Os usuários do Astra possuem acesso à nossa equipe de suporte 24x7 distribuída no esquema "follow the sun". Há diferentes opções de SLA para atender aplicações com qualquer tipo de criticidade.

Governança

Validação do custo do serviço gerenciado (forma de cobrança, por uso, por banco, etc)

O método de cobrança inicial do Astra é baseado na utilização on-demand, sendo as métricas consideradas:

- Requisições de escrita
- Requisições de leitura
- Armazenamento
- Transferência de dados.

Para workloads com consumo elevado, há a opção de ter uma capacidade provisionada.

No início, estimamos um volume de créditos para atender o workload e, conforme acontece o consumo, debitamos o correspondente do valor de créditos adquiridos

A contratação pode ser feita via marketplace da nuvem que irá hospedar o banco de dados.

https://docs.datastax.com/en/astra/astra-db-vector/administration/pricing-and-billing.html

Operações

Como adminstrar a solução?

Há diferentes opções.

- Dashboard: A maneira mais simples de administrar um banco de dados no Astra é via o Astra Dashboard, onde são definidos o nome do BD, Cloud e região de execução. Todas as operações podem ser feitas diretamente no dashboard

- DevOps API: Para automatizar a criação de ambientes através de código, a DevOps API pode ser utilizada.

  https://docs.datastax.com/en/astra-serverless/docs/_attachments/devopsv2.html

- Terraform: Empresas que já utilizam Terraform para realizar o provisionamento podem utilizá-lo com o Astra também.

  https://registry.terraform.io/providers/datastax/astra/latest/docs

Como são feitos o Backup/Restore do dados?

Os backups são executados de hora em hora e armazenados por 20 dias. A restauração de um backup deve ser feita via chamado.

https://docs.datastax.com/en/astra/astra-db-vector/databases/database-overview.html#backup-and-restore

Não há custo adicional para a execução dos backups e armazenamento das informações.

Os dados são replicados para oferecer mais resiliência para aplicações críticas?

Os bancos de dados criados no Astra por padrão são replicados em 3 zonas de disponibilidade na região selecionada. Isto garante a disponibilidade de 99.99%

Também é possível replicar os dados em múltiplas regiões do mesmo provedor de nuvem. Então os mesmos dados ficam disponíveis, por exemplo, na AWS São Paulo e AWS US-EAST-1. Este é mais um dos benefícios nativos do Apache Cassandra que pode ser utilizado para aumentar a disponibilidade do ambiente para 99.999%

https://docs.datastax.com/en/astra/astra-db-vector/databases/manage-regions.html

Como é possível monitorar a performance do ambiente?

As métricas de performance podem ser importadas para diversas ferramentas, incluindo Cloudwatch.

No entanto, não exportamos dados como logs ou métricas operacionais, afinal o objetivo é livrar os usuários de terem que se preocupar com detalhes técnicos de operação da solução.

https://docs.datastax.com/en/astra/astra-db-vector/databases/metrics.html

É possível ter ambientes ativo-ativo em múltiplas regiões?

Dada a natureza multi-datacenter do Astra, é possível ativar múltiplas regiões dentro do mesmo provedor de nuvem. A replicação é praticamente instantânea.

https://www.datastax.com/blog/enhanced-multi-region-database-consistency-astra-db

É possível migrar do Astra para outros bancos, evitando o Lock-In? Ou de outros bancos para o Astra?

O modelo de dados do Astra é o mesmo utilizado no Apache Cassandra, então a migração dos dados para outra solução, ou mesmo a versão open source, é possível.

Já para a migração de outros bancos para o Astra temos diversas ferramentas para auxiliar este processo, que passam pela migração de dados sem downtime, restauração de backups ou arquivos de dados.

Experiência do desenvolvedor

Como os desenvolvedores criam aplicações com o AstraDB

O AstraDB oferece diferentes maneiras para o desenvolvedor interagir com os dados no Astra.

CQL: Como é baseado no Apache Cassandra, então possui total compatibilidade com o driver nativo deste banco de dados. Isto demanda conhecimento de modelagem e do CQL (Cassandra Query Language) que apesar de serem simples, exigem atenção do desenvolvedor e time de arquitetura de dados.

CassIO: No entanto, aplicações de IA Generativa possuem determinadas características, como embeddings, textos, metadados que estão abstraídos no CassIO, uma library python que possui modelos de dados otimizados e métodos predefinidos para as necessidades mais comuns nestas aplicações.

JSON: Caso o desenvolvedor deseje um modelo de dados mais flexível, a Data API permite a criação de aplicações a partir de modelos de dados JSON. As libraries Astrapy e @astra-db-ts, para Python e Javascript respectivamente, podem ser usadas para agilizar o desenvolvimento.

Referências:

Data API: https://docs.datastax.com/en/astra/astra-db-vector/api-reference/data-api.html

Astrapy: https://docs.datastax.com/en/astra/astra-db-vector/clients/python.html / https://github.com/datastax/astrapy

Astra-db-ts: https://docs.datastax.com/en/astra/astra-db-vector/clients/typescript.html / https://github.com/datastax/astra-db-ts

CassIO: https://cassio.org/start_here/

O Astra possui integração com quais frameworks?

Langchain: https://python.langchain.com/docs/integrations/vectorstores/astradb
https://python.langchain.com/docs/integrations/vectorstores/cassandra

LLamaIndex: https://docs.llamaindex.ai/en/latest/examples/vector_stores/AstraDBIndexDemo.html
https://docs.llamaindex.ai/en/latest/examples/vector_stores/CassandraIndexDemo.html

RAGStack: https://www.datastax.com/products/ragstack
https://github.com/datastax/ragstack-ai

Há ainda outros, disponíveis aqui:
https://docs.datastax.com/en/astra/astra-db-vector/integrations/integrations-overview.html

Segurança

Como funciona a segurança no Astra?

Todas as informações relacionadas a segurança do ambiente podem ser encontradas em detalhe neste whitepaper:

https://www.datastax.com/resources/whitepaper/astra-security

O Astra possui certificações de segurança?

Sim, o Astra possui as certificações:

- PCI
- HIPAA
- SOC-2
- ISO 27001

Mais detalhes em: https://trust.datastax.com/

O Astra possui integração com IDPs (Identity Providers)?

Sim, o Astra possui integração com provedores de identidade baseados em SAML, tais como Azure AD, Okta, Google, entre outros.

https://docs.datastax.com/en/astra/astra-db-vector/administration/configure-sso.html

Para criação de Tokens dinâmicos e com gestão adicional, é possível também utilizar o HashiCorp Vault:

https://docs.datastax.com/en/astra/astra-db-vector/administration/hashicorp-vault.html

O Astra possui perfis de acesso baseada em Roles (RBAC)?

Sim, é possível criar roles com perfis bem específicos e com isso limitar o acesso aos dados de acordo com as necessidades de segurança.

Aqui, um notebook onde demonstramos como criar roles através da DevOps e como eles garantem o acesso somente aos dados concedidos:
https://github.com/smatiolids/astra-vector-notebooks/blob/main/Astra_DB_Vector_Security_for_data_access.ipynb

https://docs.datastax.com/en/astra/astra-db-vector/administration/manage-database-access.html

Há segurança na comunicação entre aplicação e o Astra?

A comunicação entre Astra e aplicação pode ser feita via Internet, mas para ambientes produtivos, recomendamos fortemente a utilização de Private Endpoints entre a VPC da aplicação e o Astra. Assim, todo o tráfego acontece dentro da nuvem, com mais segurança e performance.

A conexão via Internet é possível e mais aplicável em momentos de prototipação e testes.

https://docs.datastax.com/en/astra/astra-db-vector/administration/manage-private-endpoints.html

Como os dados são armazenados? Os dados são criptografados?

Os dados são criptografados e armazenados em um object store dedicado à sua organização no Astra, com isso garantimos o isolamento dos dados. A arquitetura do Astra é voltada à segurança e ao longo da sua evolução seguimos adicionando funcionalidades e camadas para aumentar a privacidade dos dados e proteção do ambiente.

Além disso, é possível que o gerenciamento das chaves de criptografia sejam feitos pelo usuário através do Bring-Your-Own-Key (BYOK), adicionando mais uma camada de segurança aos dados:

https://docs.datastax.com/en/astra-serverless/docs/manage/org/byok.html

Os dados são armazenados na minha conta no meu provedor de nuvem??

Não, os dados são armazenados na conta da DataStax, em um object storage dedicado à sua organização.

Sobre a DataStax

A DataStax foi fundada em 2011 no Vale do Silício e é uma das maiores contribuidoras do Apache Cassandra. Desde então, atendemos diversos clientes com suporte e também uma versão on-premise, o DataStax Enterprise, que tem recursos adicionais focados em segurança e gestão de grandes clusters.

Em 2020, foi lançado o DataStax Astra, uma plataforma de dados baseada no Apache Cassandra, mas operada pela equipe técnica da DataStax, tirando dos usuários toda a carga de trabalho que uma solução de dados distribuídos, escalável e de alta criticidade demanda.

Além dos benefícios em delegar a operação de um ambiente complexo e crítico a um time altamente especializado, o Astra integra outras funcionalidades que visam acelerar o desenvolvimento de aplicações, tais como:

- Camada de APIs REST, GraphQL, gRPC e Data API (JSON)
- Streaming compatível com Kafka, RabbitMQ, KMS e Apache Pulsar

- Auto scaling
- Modelo de consumo sob demanda
- Monitoramento de performance
- Replicação dos dados em múltiplas zonas de disponibilidade e regiões
- Sem lock-in com fornecedor

Faltou algo? Vamos conversar!

https://www.datastax.com/