



Build a Realtime Event Driven API with Kafka & Cassandra

Cassandra in the Cloud

DATASTAX



ANANT

Rahul Singh

- CEO of Anant Corporation
- Creator of [Cassandra.link](#) + Maintainer of [awesome-cassandra](#)
- Contributor of [awesome-kafka](#), [awesome-spark](#),[awesome-solr...](#)
- Manage Meetups groups in the Washington region



menti.com

21 09 02 9



Available on the iPhone
App Store

GET IT ON
Google play





Series

Event Driven Toolkit

- Simple REST API on Cassandra
- Event Sourced API on Kafka
- Connecting Kafka to Cassandra
- Kafka / Spark / Cassandra

Session - Kafka Topics for Event Driven API

1. Welcome
2. Basics of Event Driven API
3. Basics of Kafka Topic Modeling
4. Fork Code and test Connection
5. Create Topics in Kafka
6. Test Message Creation
7. Kafka Connect, Kafka Streams, Kafka REST Proxy

Prerequisites

1. Cloud Account on Astra
2. Cassandra.API Repo / Gitpod

Takeaways

3. Understand basics of Kafka Topic/Schema/ Partitioning
4. Understand Kafka API Basics
5. Create / Consume Topics in a few different ways

Sponsors



Gitpod - Online serverless IDE & free pro accounts for dev team.

DataStax - Online serverless Cassandra as a Service, Event Hosting

Anant - Source code, documentation, workshop content.



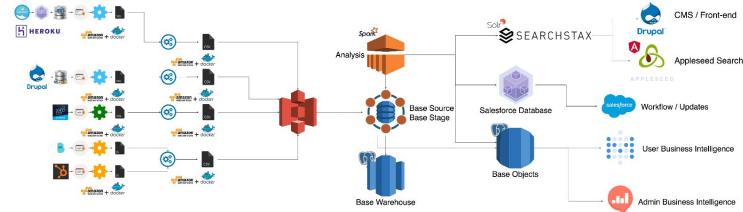
ARCHITECT

noun: architect; chief builder

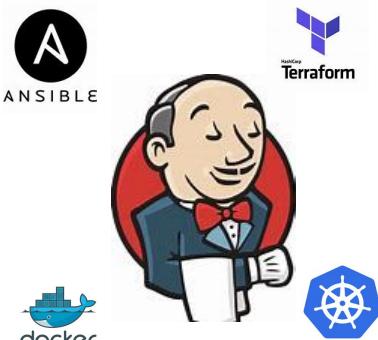
verb: architect; design or make (**COMPUTING**)

“We at Anant architect platforms on Cassandra and related technologies.”

Things We Love : Scalable Fast Data

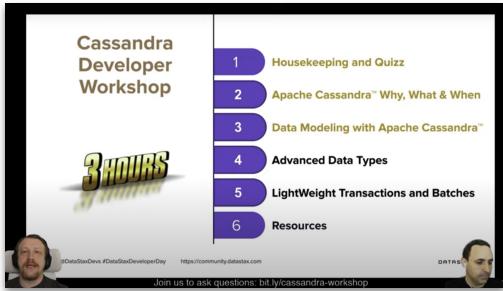


Without Datastax



With Datastax

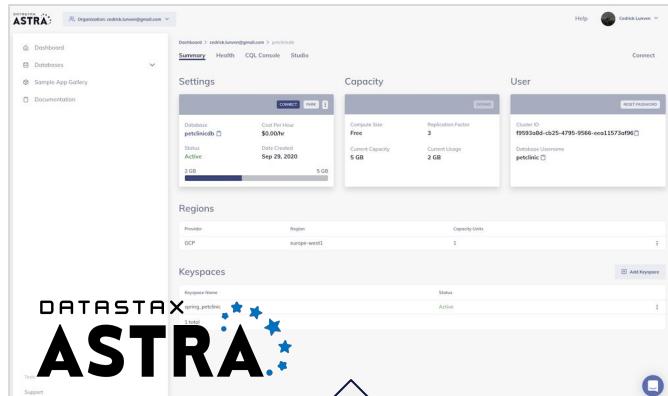
LIVE: youtube.com/DataStaxDevs



Quizz: menti.com



Runtime: dtsx.io/workshop



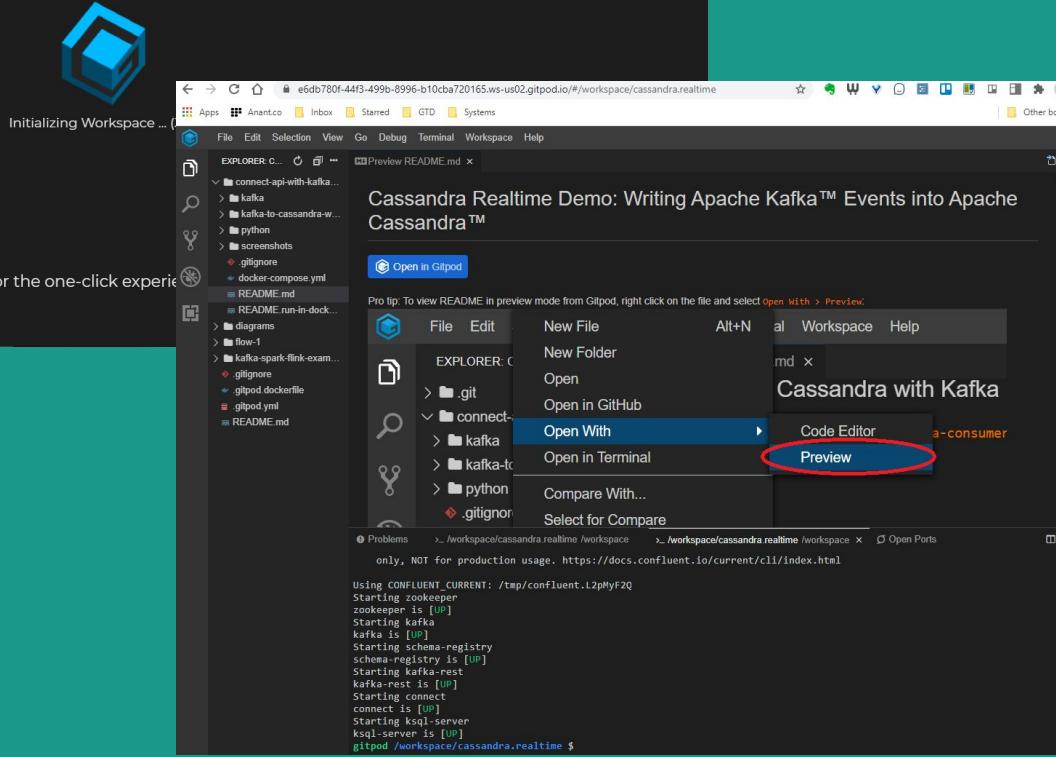
Let's Launch Gitpod

<https://github.com/anant/cassandra.realtime>

For Cloud Deployment



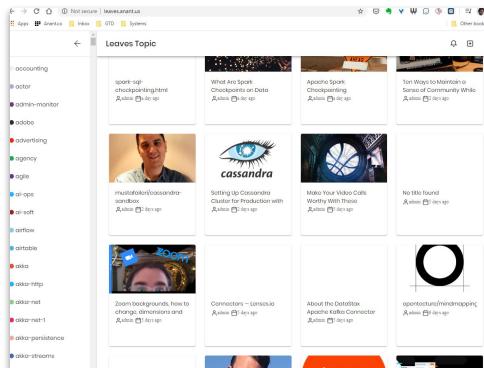
Open in Gitpod



Why are we building this app?

What is Leaves? Our Knowledge Base UI & API

1. Leaf / URL / Bookmark
2. Topic /Categories/Tag
3. Collections / Branch of Leaves
4. Users / Curators
5. Teams*



Not secure | leaves.anant.us/leaves/#!/leaf/15026.14991.14881?tag=cassandra

cassandra [709]
cassandra.lwt [3]
cassandra.stress [1]
cassandra.thrift [8]
cassandra [2]
relational->cassandra [2]

Leaves Reader

(3/3)- kdnan/PythonCassandraTutorial

(1)- Designing Microservices With Cassandra - Dzone Refcardz
(2)- Table Store Time Series Data Storage Architecture
(3)- kdnan/PythonCassandraTutorial

Join GitHub today

GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.

Sign up

This code is part of the tutorial related to Cassandra and Python

▶ Python
▶ Branch: master Find file ▶ Clone or download

Type Name Latest commit message Commit time

Failed to load latest commit information.

README.md

This repo is the part of the tutorial Getting started with Cassandra and Python

The screenshot shows a web browser window with a dark theme. The address bar says "Not secure | leaves.anant.us/leaves/#!/leaf/15026.14991.14881?tag=cassandra". On the left, there's a sidebar with a list of tags: accounting, actor, actor-monitor, apache, advertising, agency, log4j, ops, soft, airflow, ambari, kafka, net, netty, netty-http, netty-net, netty-nett, netty-persistence, and netty-streams. The main content area has a title "Leaves Reader" and a sidebar with a list of "cassandra" entries. Below that is a card for "How to use Apache Cassandra's Stress Tool: A step by step guide" from medium.com, posted 18 days ago by admin. To the right is a GitHub integration section for "kdnan/PythonCassandraTutorial" with three commits. At the bottom is a large "assandra" logo with the text "Getting started with Apache Cassandra and Python".

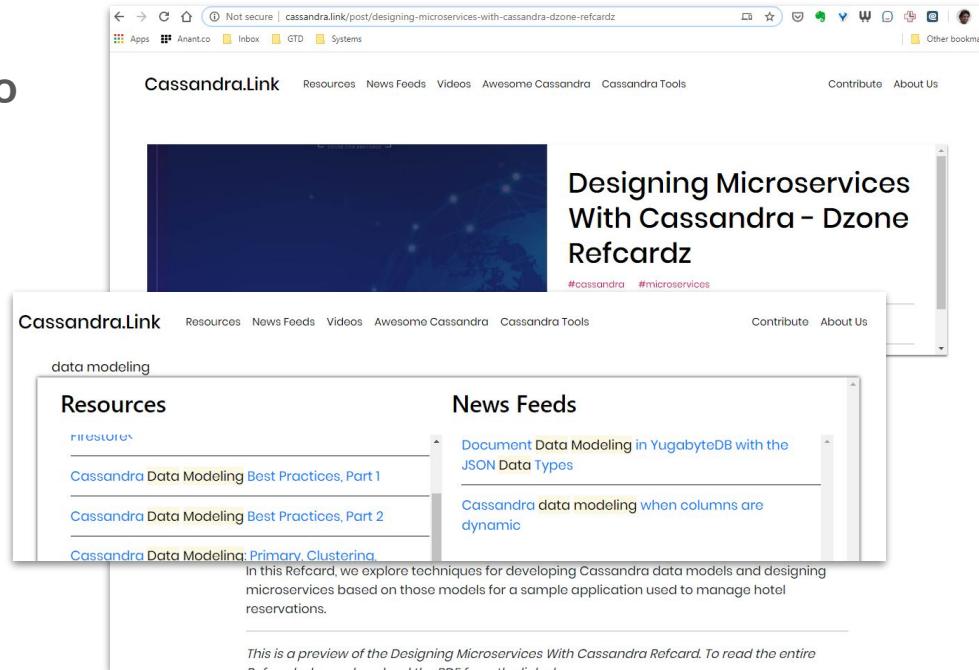
How else do we use our Knowledge Base API?

1. Consumers

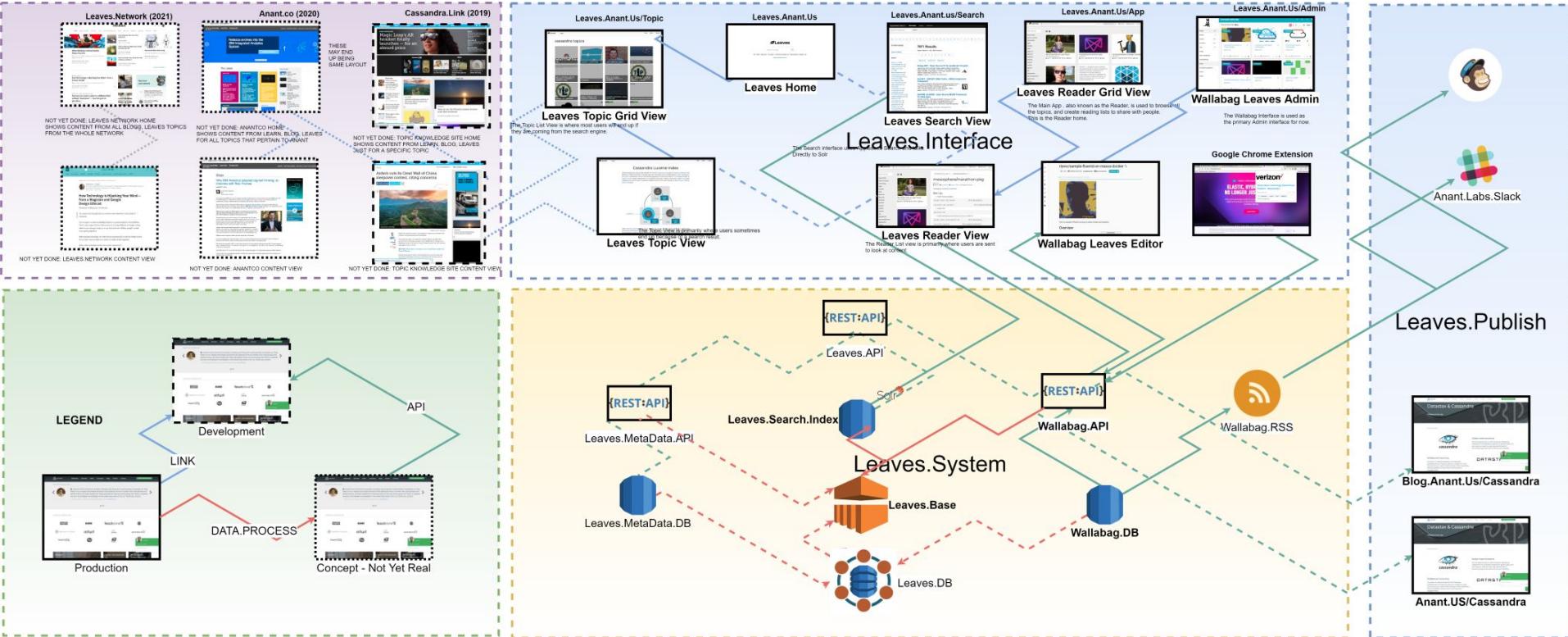
- a. Static Sites Rendered to CDNs
- b. Users can search links/contents via full-text search.
- c. Users find links with topics.

2. Curators

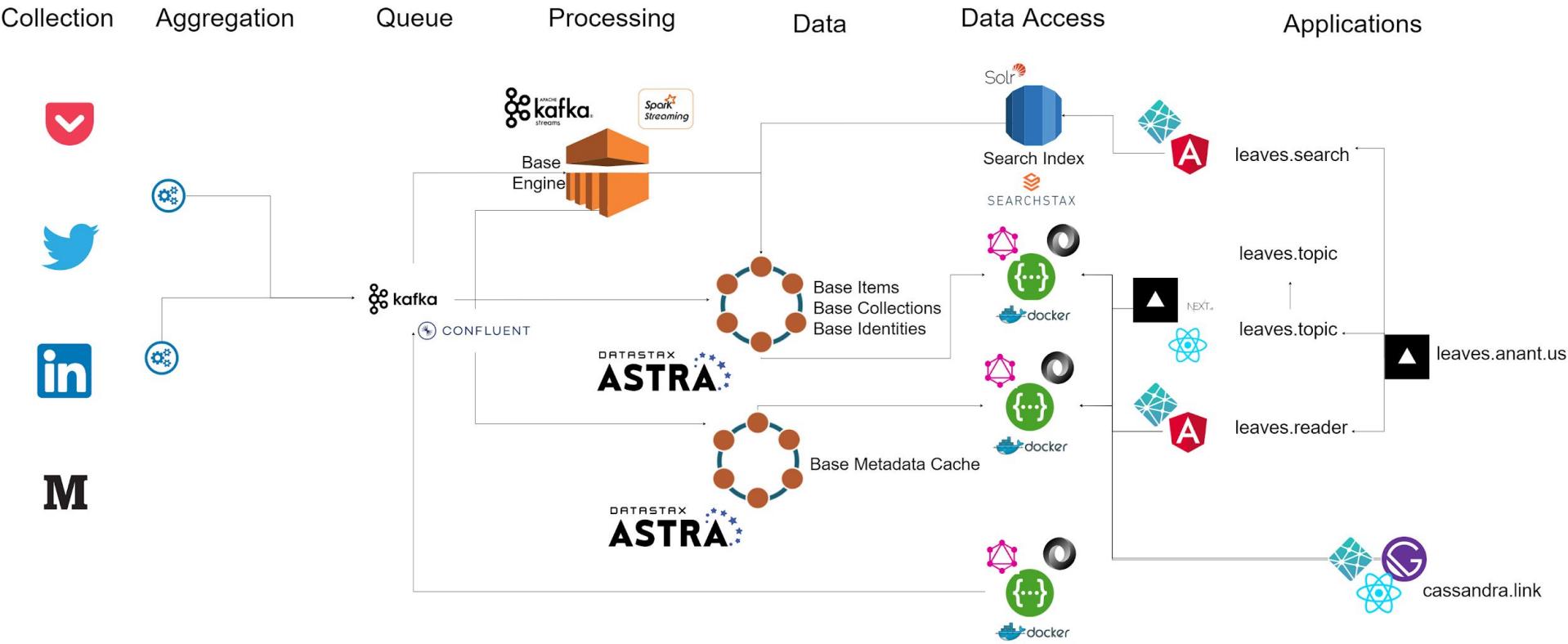
- a. Users can share collections



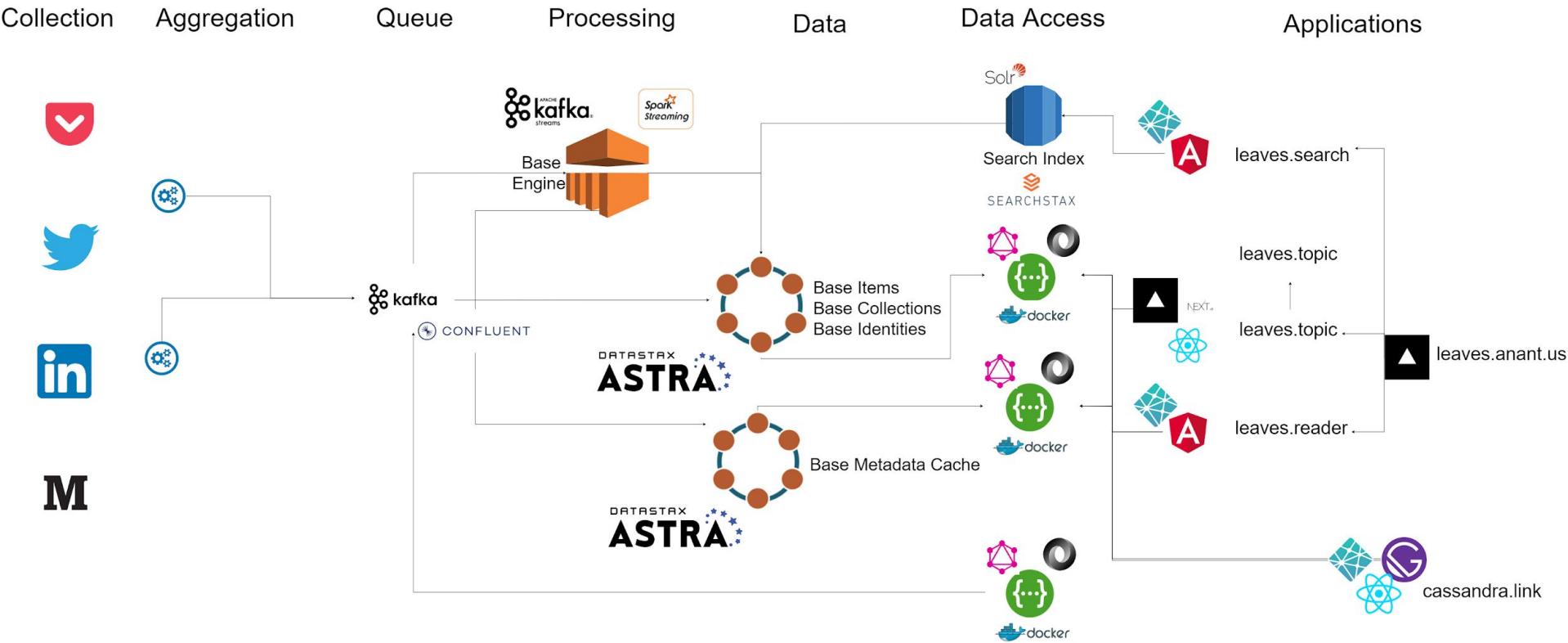
Where we are now:



Where we are going:



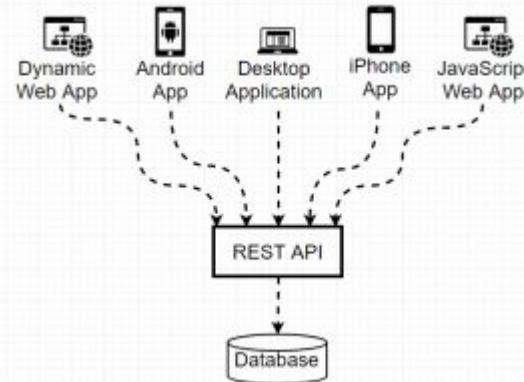
Where we are going:



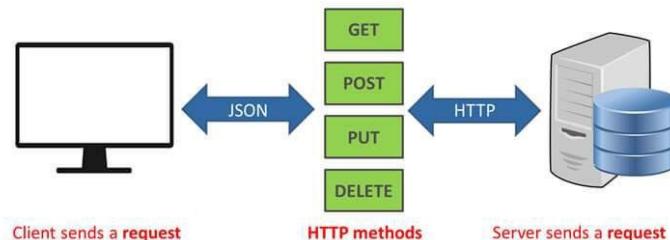
Overview of REST vs. Microservice vs. Event Driven Architecture

REST: Quick Introduction

- REST = REpresentational State Transfer
- Transfer the “State” of the Database at any Given time.
- Used to wrap other APIs as well.
- REST is Stateless so nothing is assumed.
- Infinitely scalable layer if done right.
- All over HTTP using GET, PUT, etc.
- Can be extended e.g. HATEOAS, GraphQL



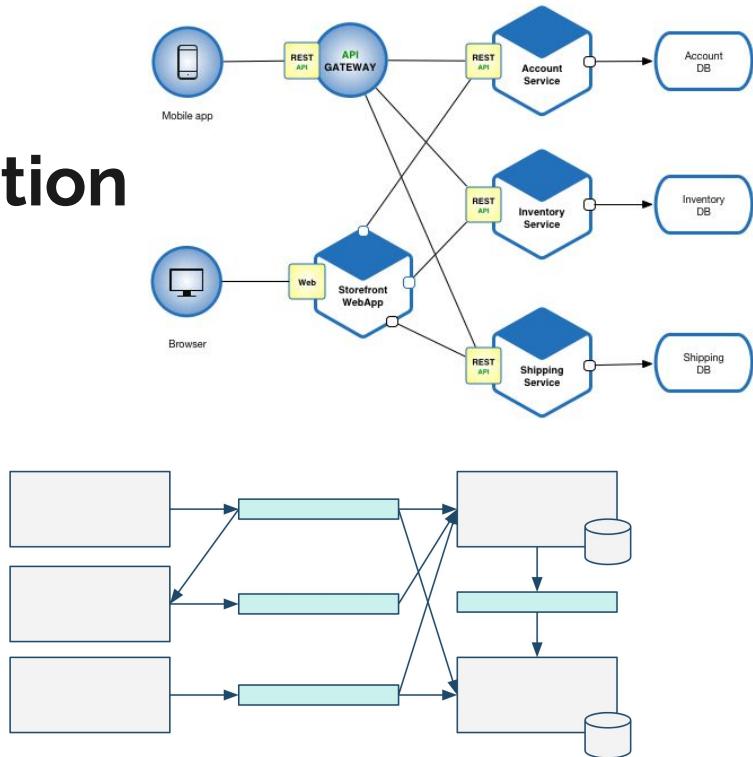
<https://www.parallels.com/blogs/ras/rest-api/>



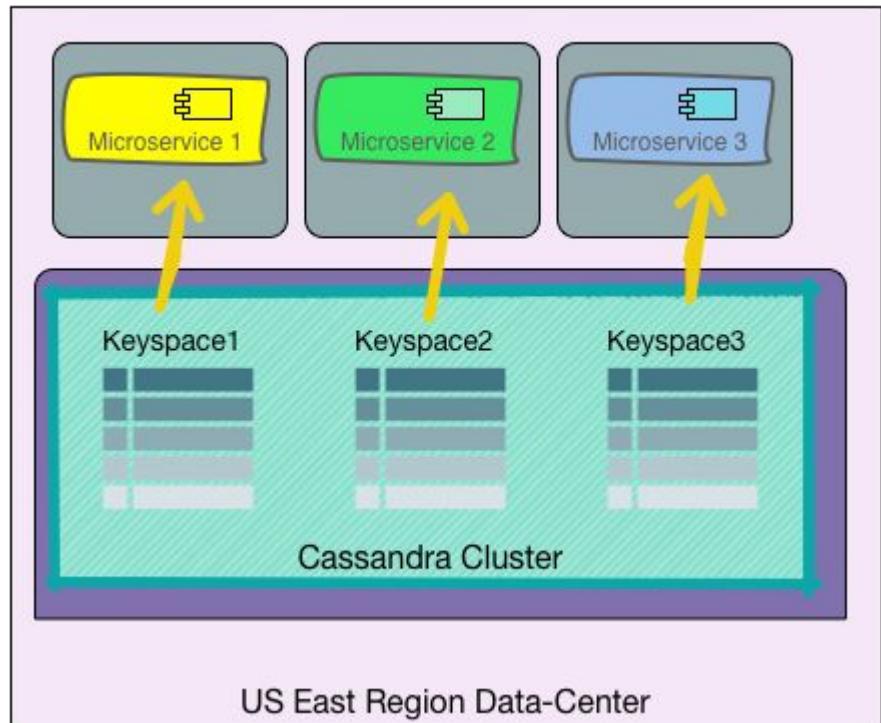
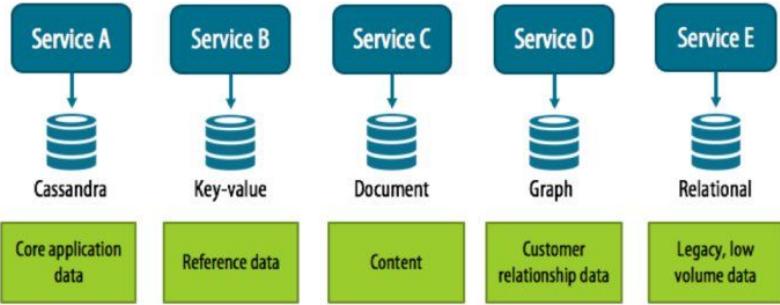
<https://medium.com/faun/consuming-rest-api-s-with-python-eb86c6b724c5>

Microservice: Quick Introduction

- Loosely Coupled with other services
- Independently Deployable
- Can be implemented Sync or Async
 - E.g. HTTP/REST = Sync
 - E.g. AMQP, Kafka = Async
- Maintained by small teams
- Each service has its own DB
- Consistency Achieved via Saga Pattern
- Automated deployments



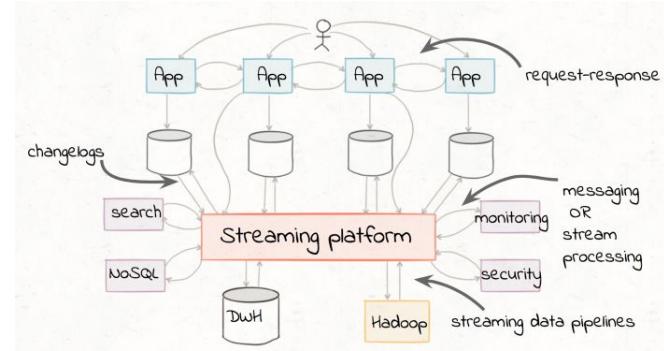
Microservices on Cassandra



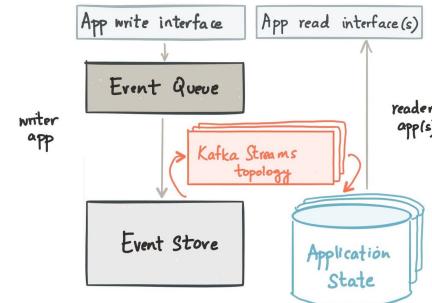
<https://www.infoworld.com/article/3236291/how-to-choose-a-database-for-your-microservices.html>

Event Driven Architecture

- Pub / Sub Model for Communication
- Event Sourcing - Everything is sourced from Events in a Queue or a Database
 - Events are created by API into a Channel
 - Channel being a Queue, DB, etc.
 - Events are processed by a Processing Engine
- Implemented sometimes on top of Message Driven Architecture

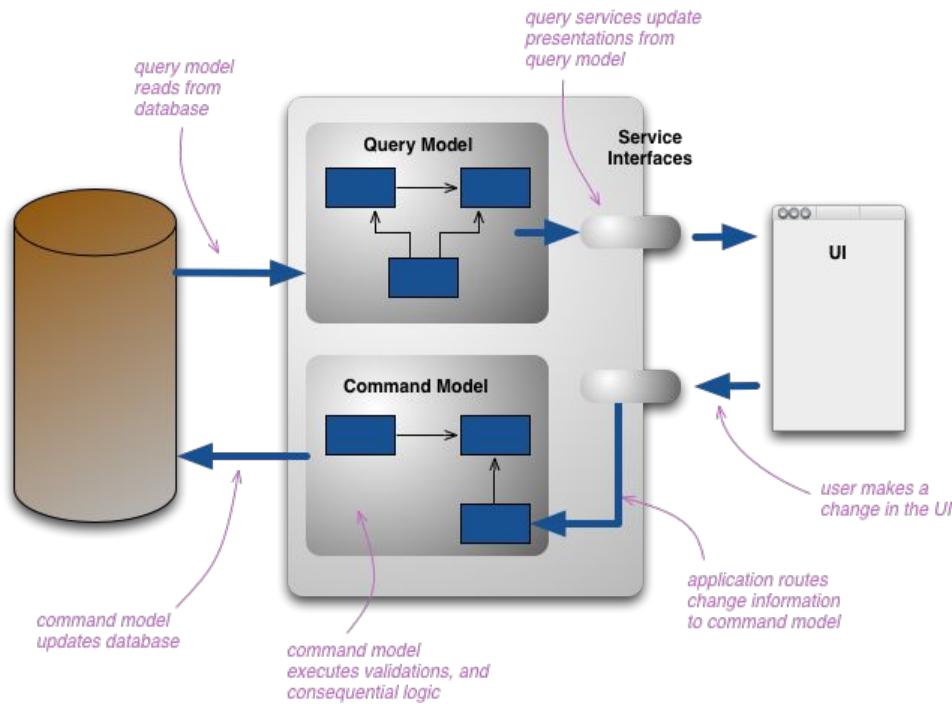
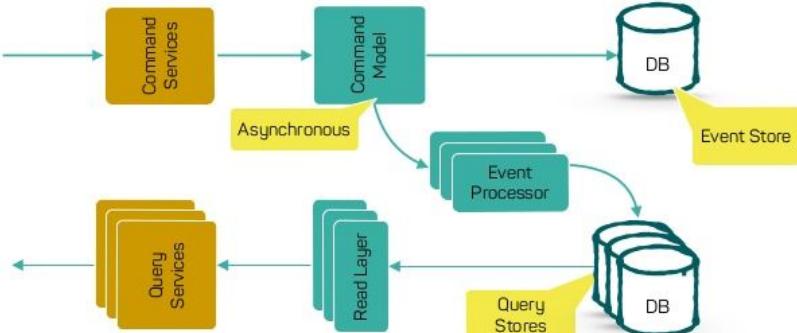


<https://medium.com/@javier.ramos1/devops-microservices-part-5-streaming-platforms-e9220dc06ed8>



Command Query Separation (aka CQRS)

Event Sourcing & CQRS



Apache Kafka™ Basic Concepts



What, When and Why.

Apache Kafka

= Distributed Message Log/Stream

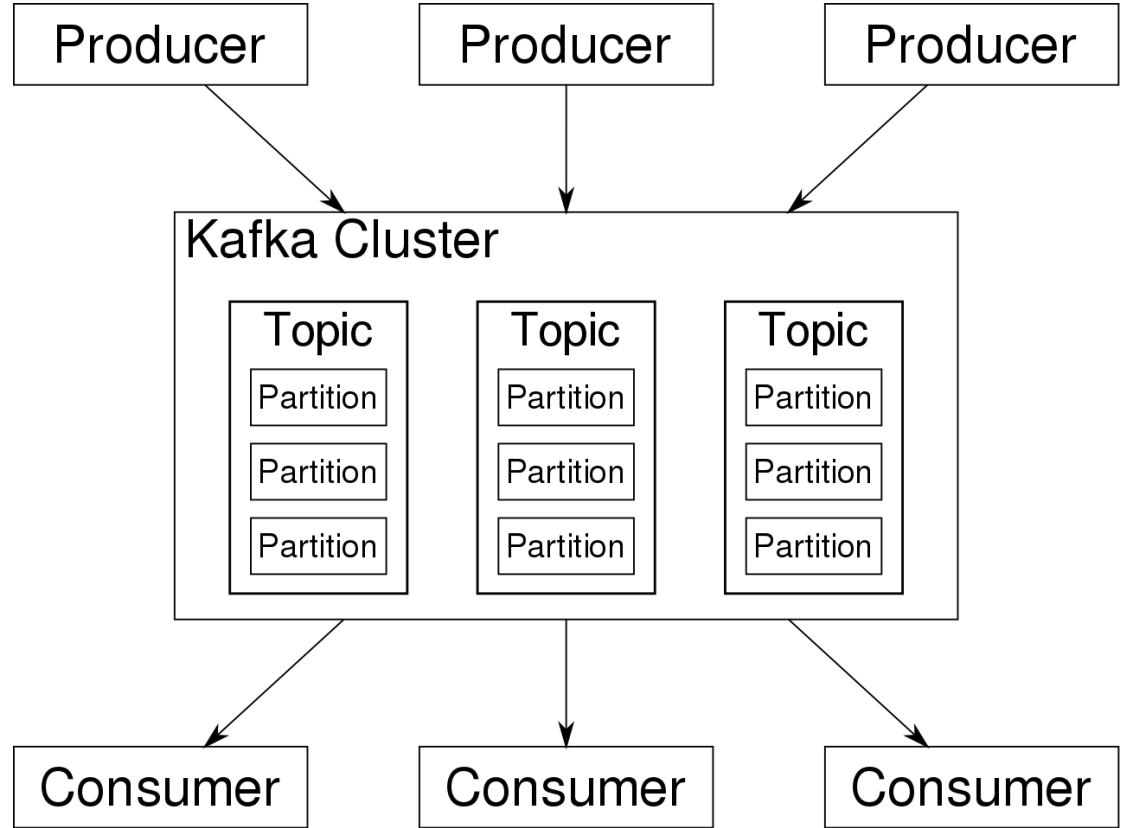
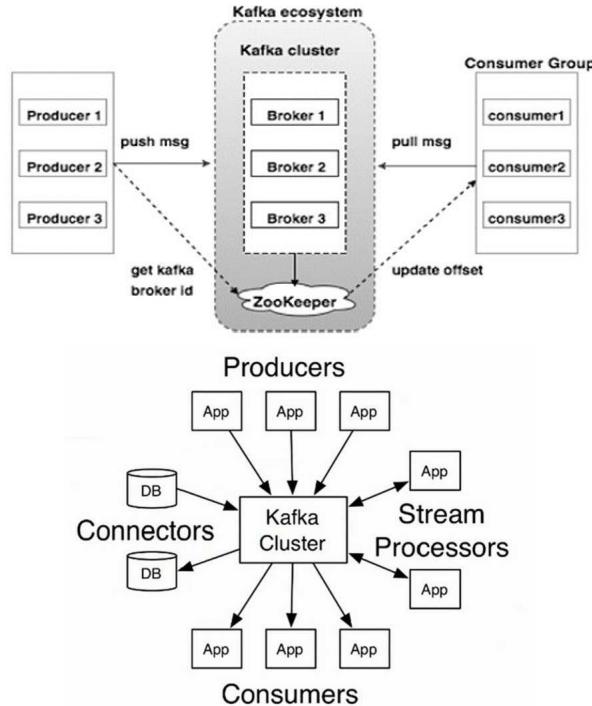


Image Credit: [Tutorials Point](#), [Wikipedia](#)

Kafka Cluster Architecture

- **Kafka == Kafka Broker Service**
 - Clusters / Brokers
 - Topics / Partitions / Messages
- **Kafka != Kafka Connectivity**
 - Kafka Client API -> Consumers/Producers
 - Kafka Connect -> Pluggable Connectors
 - Kafka Streams -> Stream Processors
- **Kafka != Kafka Helpers**
 - Kafka Rest Proxy
 - Kafka Schema Registry

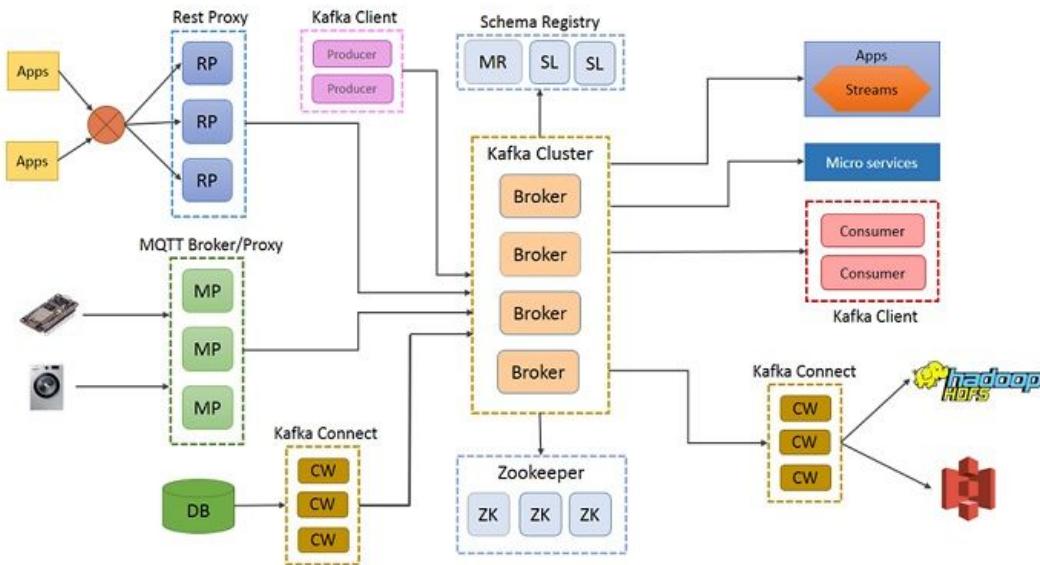
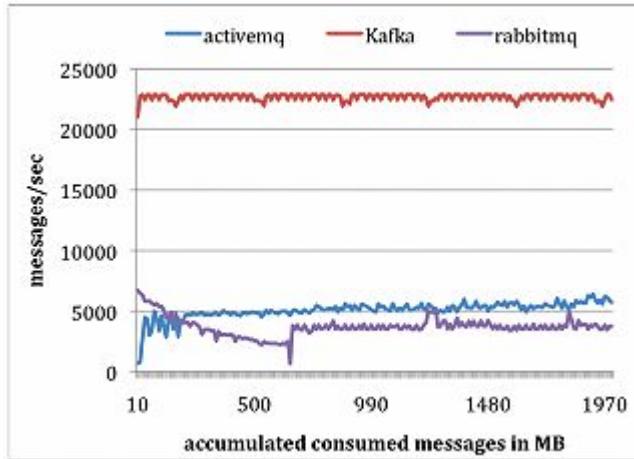
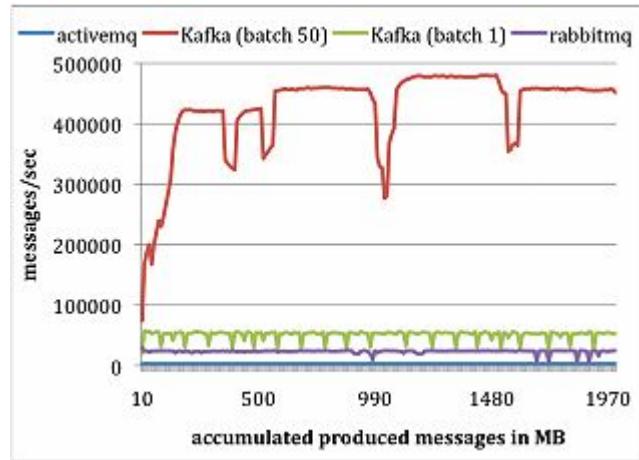


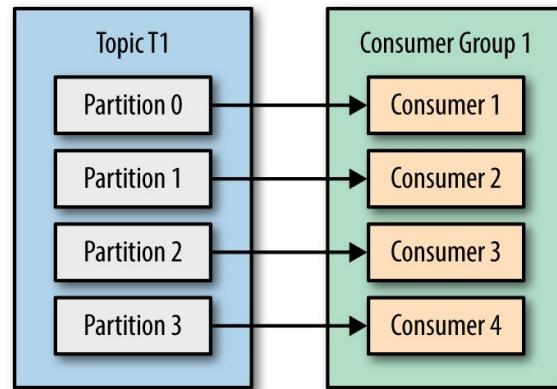
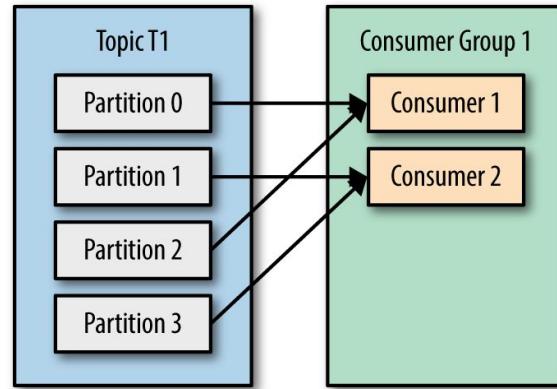
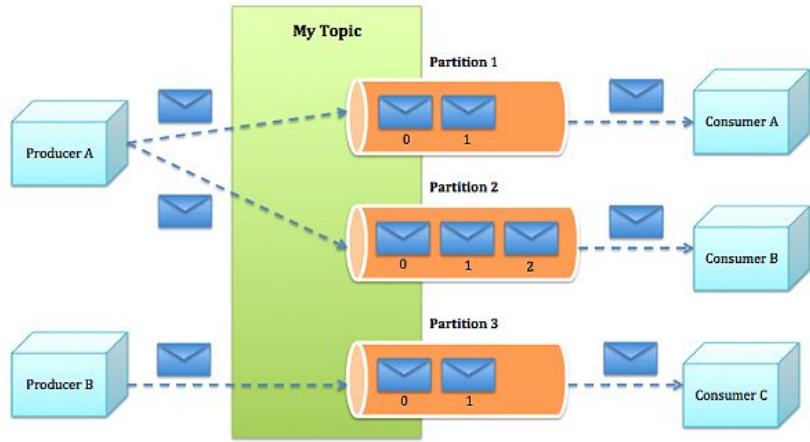
Image Credit: [Data Flair](#)

Scales Like a Champ

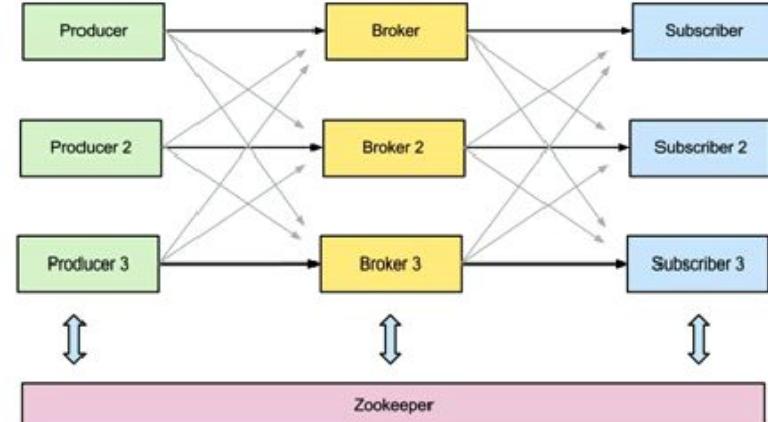
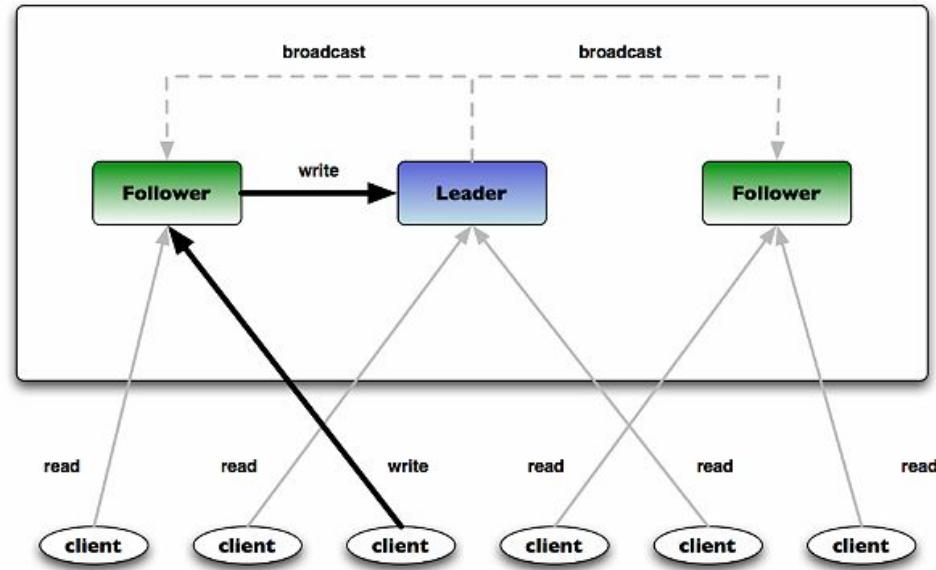
- Outperforms other messaging systems in write/read speeds.
- Can be Chained, Aggregated, Mirrored
- Works well with Cassandra, Spark, Flink, etc.
- Street credibility : LinkedIn, Uber, Netflix, Comcast...
- Competing technologies also now Kafka compliant: e.g. Apache Pulsar



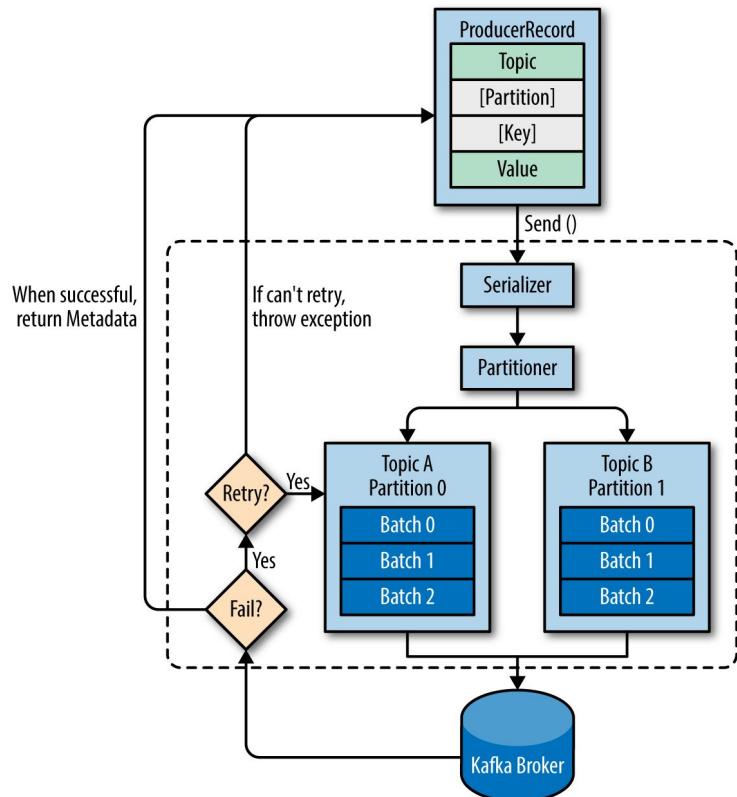
Messages are Partitioned



Messages are Distributed & Replicated



Messages are Written / Read by Partitioning Strategy



1. **Message Partition ID**
if provided in Message
2. **Message Key % # of Partitions**
if no Message Partition ID
3. **Round Robin**
If no Message Partition ID,
no Message Key

You can use what's given or you can make your own partitioning algorithm ... to create strategies.

Messages are Written / Read by Partitioning Strategy



Partition 0



Partition 1



Partition 2

Random Partitioning



Partition 0



Partition 1



Partition 2

Partition by Aggregate



Partition 0



Partition 1



Partition 2

Partition by Guarantee

Kafka is Web Scale with Mirror Maker

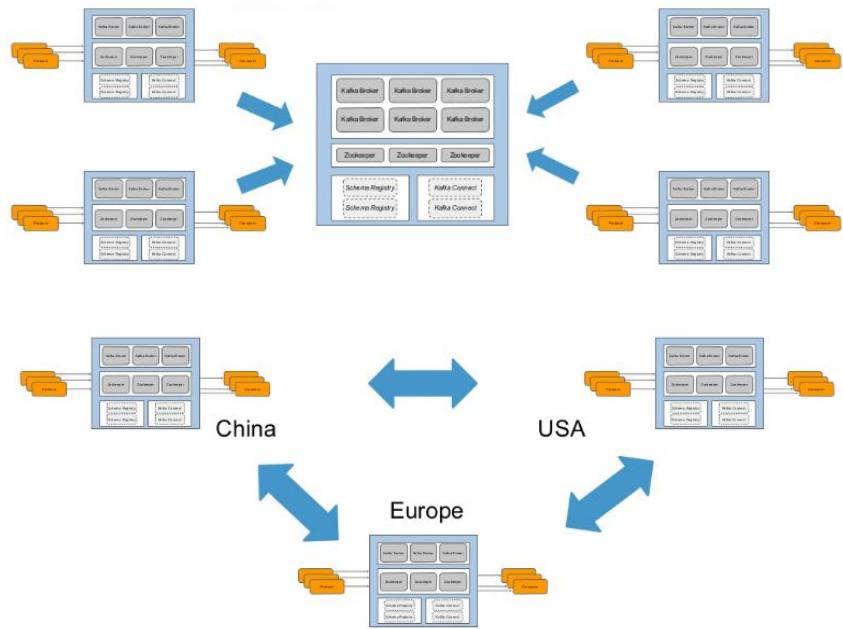
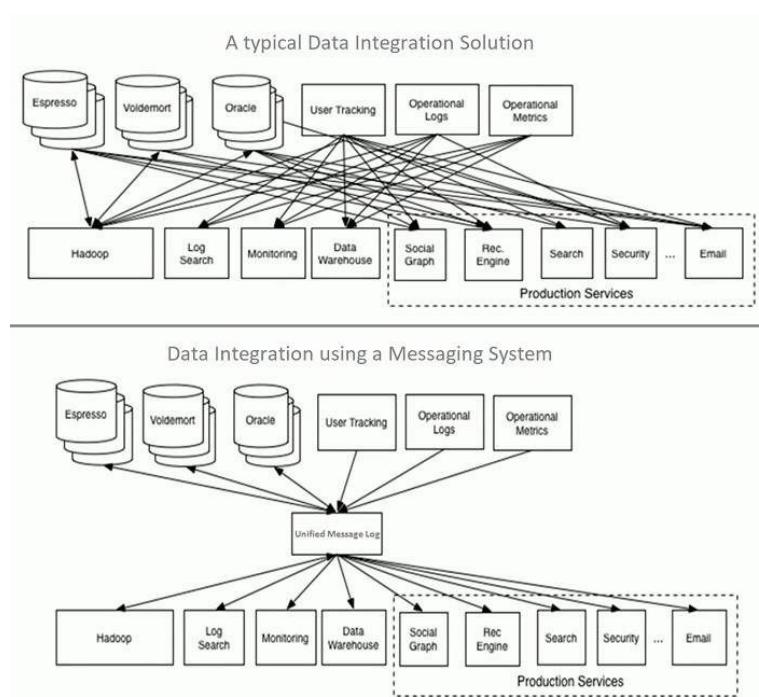


Image Credit: YouTube, [Kai Wehner](#)

Understanding Kafka Use Cases



- **Technical**
 - **Messaging (RMQ,AMQ)**
 - **Web Activity Tracking (original)**
 - **Metrics**
 - **Log Aggregation**
 - **Stream Processing**
 - **Event Sourcing (@here)**
 - **Commit Log**
- **Industry**
 - **E-Commerce**
 - **IoT / Edge**
 - **Retail**
 - **Banking**
 - **Logistics**

Image Credit: [Jay Kreps Blog on LinkedIn Engineering](#)

Astra

=

Cassandra As a Service



Introducing Astra



Eliminate Operations

everything from provisioning to backups is fully automated



Secure Your Data

with the most advanced security available for Cassandra



Simplify App Development

with auto-configured developer tools that deploy with a click

Astra Development Features

Familiar Language

```
INSERT INTO mytable  
(id,name,address) VALUES  
(1,'Bob Smith','1 Main  
Street')  
SELECT * FROM mytable  
WHERE id=1  
UPDATE mytable SET  
name='Tom Smith' WHERE  
id=1  
DELETE FROM mytable WHERE  
id=1
```

Easy Dev Tools

index	video_id	username	event	timestamp	event	video_timestamp
0	"99991ch-5ch-4612-349f-38af78785000"	tosdd	"99991ch-5ch-4612-349f-38af78785000"	1646110310431170748079193"	stop	3000
1	"99991ch-5ch-4612-349f-38af78785000"	tosdd	"99991ch-5ch-4612-349f-38af78785000"	"99991ch-5ch-4612-349f-38af78785000"	start	3000
2	"99991ch-5ch-4612-349f-38af78785000"	tosdd	"99991ch-5ch-4612-349f-38af78785000"	"99991ch-5ch-4612-349f-38af78785000"	stop	3000
3	"99991ch-5ch-4612-349f-38af78785000"	tosdd	"99991ch-5ch-4612-349f-38af78785000"	"99991ch-5ch-4612-349f-38af78785000"	start	3000
4	"99991ch-5ch-4612-349f-38af78785000"	tosdd	"99991ch-5ch-4612-349f-38af78785000"	"99991ch-5ch-4612-349f-38af78785000"	stop	3000
5	"99991ch-5ch-4612-349f-38af78785000"	tosdd	"99991ch-5ch-4612-349f-38af78785000"	"99991ch-5ch-4612-349f-38af78785000"	start	3000

Great Drivers



Why use Astra?

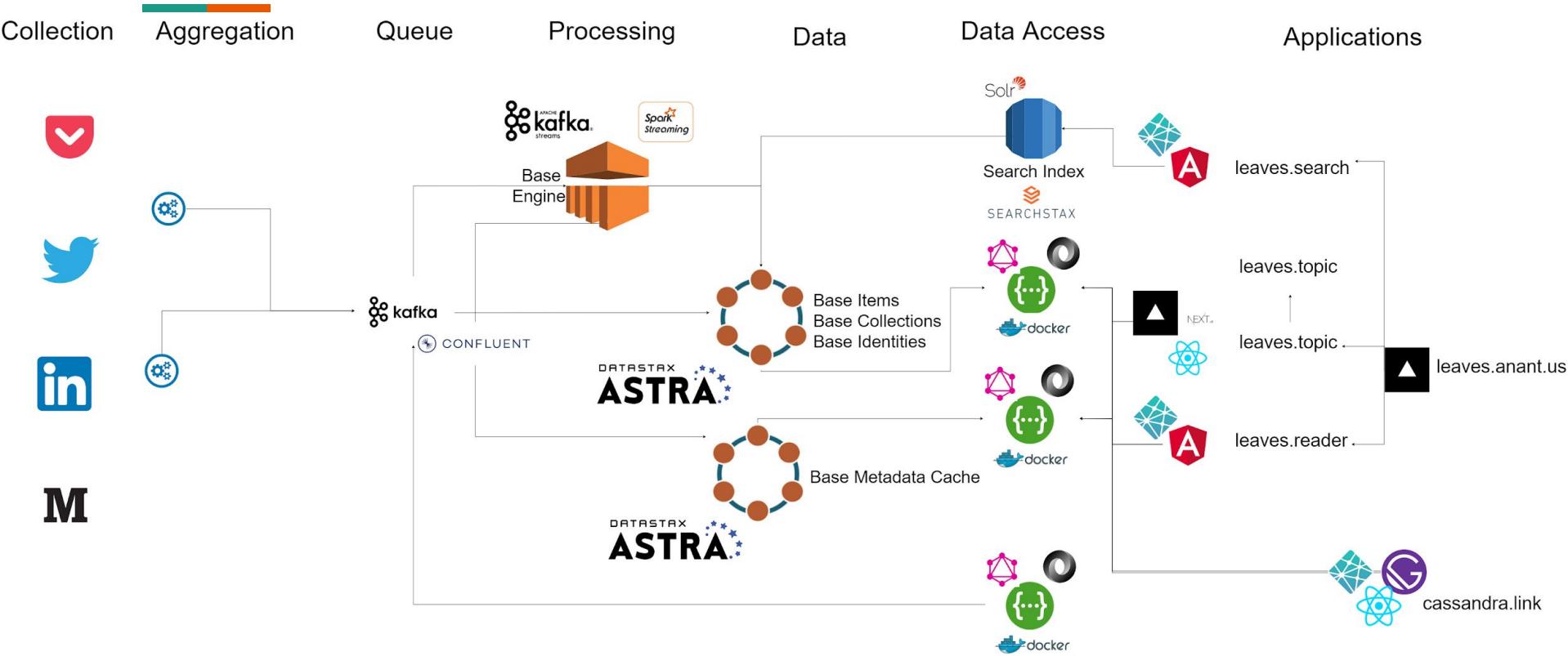
1. Focus on the feature
2. Focus on what matters to us.
3. Trend towards serverless.
4. Our time is valuable.
5. We can build faster.
6. We can scale faster.

The screenshot displays the Astra platform's user interface. At the top, there is a search bar with the placeholder "datastax" and filter options for "Match Any Words" and "Match All Words". Below the search bar are buttons for "Clear All Filters", "All User", "All Time", and date range selection ("From: mm-dd-yyyy" and "To: mm-dd-yyyy"). On the right, there are buttons for "Items Per Page" and sorting by "Date" (with "A-Z" and "Z-A" options). The main area shows a grid of article cards. Each card includes a thumbnail image, the article title, and a brief description. Some cards also show the author and publication time. The bottom left corner of the grid features a sidebar with a list of tags and their counts: dropbox [2], oskar [1], airflow [1], ericidle [2], akka [58], akka-net [1], akka-http [1], akka.net [2], akka.persistence [1], akka.stream [2], algorithm [1], apapka [1], amazon [5], anaconda [1], analytics [34], android [4]. The bottom right corner shows a snippet of a search result titled "How to Configure a Droplet as a VPC Gateway - DigitalOcean Product Documentation".

Let's Launch Astra

<http://astra.datastax.com/>

Where we are going:





Don't have an account? [Sign up](#)

Sign In

Email

Password

Remember Me

[Forgot Password?](#)

Sign In

OR

```
database name: leavesnetwork
keyspace name: killrvideo
username: KVUser
password: KVPassword
```

DataStax Astra Database-as-a-Service

Cassandra Made Easy in the Cloud

Choose Plan & Provider

Select a Plan

Free ⓘ

\$0

For anyone who wants to test Astra in a sandbox with no obligation and 5 GB of storage.

Production Workloads ⓘ

Starts at \$2.25/hour ▾

RF=3 Database for Production Workloads with Dedicated Resources

High Density Production Workloads ⓘ

Starts at \$5.95/hour ▾

RF=3 Database for High Density Production Workloads with Dedicated Resources

Select a Cloud Provider

Select a plan before choosing a cloud provider.

Astra Tools

Astra Cluster

Cluster Overview

Nodes Up: 1

Coordinator Metrics

(10 rows)
anton_admin@cqlsh> [REDACTED]

Code Editor

Language: CQL Keyspace: knowledge

SELECT * FROM urls;

index	id	all	content
			<article class="markdown-body entry-content p-5" itemprop="text"><h2><a id="user-content-open-sou...
			[REDACTED]
			<!DOCTYPE html> <html lang="en-gb"> <head> <meta charset="utf-8"/> <meta content="width=device-w...

Metrics

Clients Connected: 16

Client Timeouts: 0

Total Hints: 0

Live Data Size: 240 MB

Memtable Space All/All / Cluster

2.5 MB
2.0 MB
1.5 MB
1.0 MB
0.5 MB
0.0 MB

2.5 MB
2.0 MB
1.5 MB
1.0 MB
0.5 MB
0.0 MB

10:35 10:40 10:45 10:50

cluster((cluster)) ((.90)) (requestType) cluster((cluster)) ((.90)) (requestType)

Resources

Cassandra

- [Cassandra.link](https://cassandra.link)
- [Cassandra.tools](https://cassandra.tools)
- anant.github.io/awesome-cassandra

Datastax

- [Datastax.com](https://datastax.com)
- Astra.datastax.com

Kafka

- <https://kafka.apache.org/>

Confluent

- <https://www.confluent.io/>
- <https://www.confluent.io/confluent-cloud/>

Knowledge Base

Discover the best resources related to Apache Cassandra

Project by anant.us

News Feeds

CosmosDB Query For
Searching in a DB
43 minutes ago

Adding an Entity to an
Azure Cosmos Table with
Javascript Azure Functi
an hour ago

Scylla University: Take the
Challenge and Save the
Planet!
8 hours ago

Cassandra timeout cqlsh
query large(ish) amount
of data
8 hours ago

cassandra

Understanding How
Cassandra Stores
Data



cassandra

Apache Drill
Contribution Ideas –
Apache Drill



cassandra

Apache Cassandra
Multi-Datacenter
Essentials (Julien
Anguenot, iLand ...)



solt

Subscribe

Share your email with us
and get the latest and
greatest Cassandra
resources

Enter email

FEED ME GOOD
RESOURCES

OSS Projects

intuit wasabi
intuit

cassandra-
stable-tools
instaclustr



Hands On

-
- Gitpod / Docker for Dev & DevOps
- Postman
1. Verify Cassandra.API /
 2. Verify Kafka / Start Kafka
 3. Create Topic / Schema / Verify
 4. Create / Verify Messages with Python
 5. Kafka to Cassandra with Kafka Streams
 6. Kafka to Cassandra with Kafka Connect

Gitpod IDE

The screenshot shows the Gitpod IDE interface. The top navigation bar includes File, Edit, Selection, View, Go, Debug, Terminal, and Help. The left sidebar (Explorer) lists project files: astra.api, leaves.api.node, node_modules, src, Routes (with router.js selected), .env, .eslintrc.json, .gitignore, .gitignore.txt, example.env, package-lock.json, package.json, Procfile, README.md, leaves.api.python, leaves.api.tests, astra.credentials, and astra.import. The bottom terminal pane shows a nodemon log and a message indicating the server is listening at http://localhost:8000 and astra connected.

```
[nodemon] 1.19.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): ***!
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/server.js`
Server listening at http://localhost:8000
astra connected
```

The screenshot shows the Gitpod IDE interface. The top navigation bar includes Help, Preview README.md, SolrToAstra.py (selected), and a preview of README.md. The main editor pane displays Python code for connecting to an Astra cluster and executing a query. The bottom terminal pane shows the same nodemon log as the previous screenshot.

```
from cassandra.cluster import Cluster
from cassandra.auth import PlainTextAuthProvider
import json
import requests
from datetime import datetime
import time

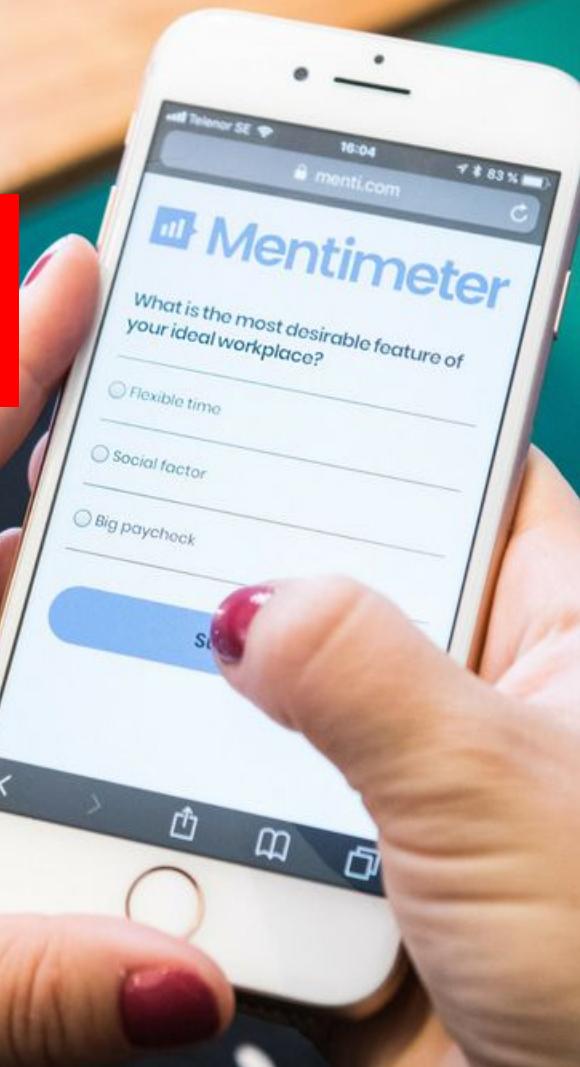
#Connect to Astra Cluster
#Redo this after git project restructuring
with open('astra.credentials/UserCred.json') as f:
    cred = json.load(f)
cloud_config= {
    'secure_connect_bundle': 'astra.credentials/secure-connect-'+cred['cluster']+'.zip'
}
auth_provider = PlainTextAuthProvider(cred['username'], cred['password'])
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.")

#Create Table leaves if it does not exist
session.set_keyspace('killrvideo')
f = open('astra.import/schema/AstraTableDef')
session.execute('CREATE TABLE IF NOT EXISTS '+str(f.read()))
```

menti.com

21 09 02 9



Available on the iPhone
App Store

GET IT ON
Google play





Strategy: Scalable Fast Data

Architecture: Cassandra, Spark, Kafka

Engineering: Node, Python, JVM,CLR

Operations: Cloud, Container

Rescue: Downtime!! I need help.



www.anant.us | solutions@anant.us | (855) 262-6826
3 Washington Circle, NW | Suite 301 | Washington, DC 20037



Any Questions?