Event driven streams for External Integration for Sorted Services

About Client

Our client Sorted offers turnkey solutions in the real estate state sector to manage rentals for property agencies. They also provide an app that helps end-users manage their bill payments right from the time of moving into a rented property. This one-stop app helps users pay all their bills including house rent, electricity, broadband, water, gas, and other utilities.

Problem Statement

Sorted depends on external property management systems (PMS) to pull property, the landlord, and tenant data. The ability to integrate with multiple PMS available in the market was key to business, each having its own data structure and provider APIs to fetch data. Each integration would take a lot of time in development, with extensive quality checks and release cycles to get it right.

HashedIn's Proposal

The solution proposed was to move to an event-based Kinesis data streams to process property, landlord, and tenant data. Microservices were written for each integration that publish data events as a generic payload into the streams. Consumer service on the other end will process the payload and store the information in RDS. To ensure a fault-tolerant system, Kinesis stream retention policy is leveraged to replay data from the last processed record, in case of failures. Buildkite is used to automate the CI/CD process and terraform was utilized to implement infrastructure-as-code to speed up the infra setup-up process.

Technical Highlights

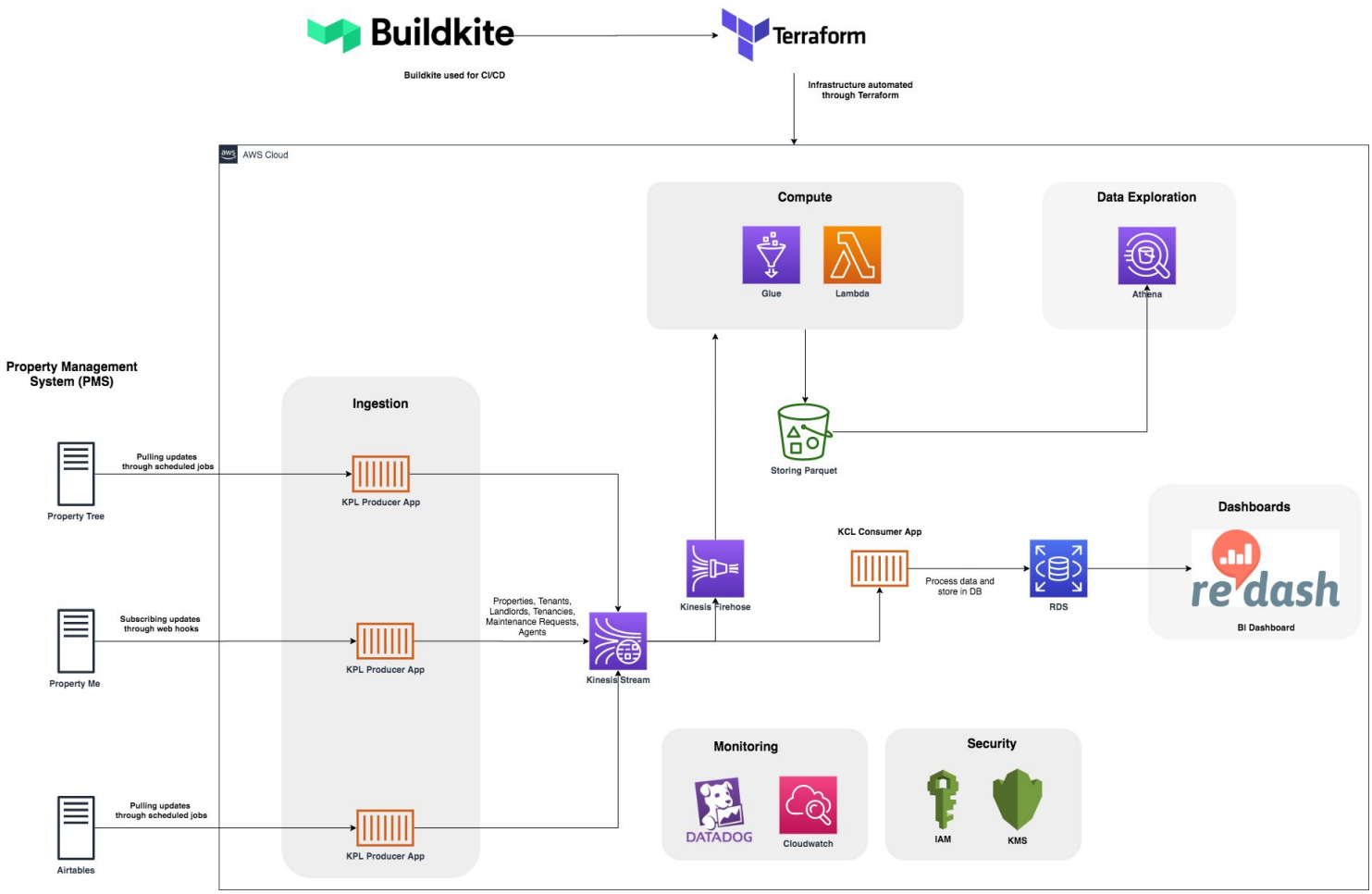
- One microservice is responsible for each PMS integration, calling provider APIs to pull data, convert into a generic payload, and ingest data into streams.
- AWS Firehose for delivering data to the S3 bucket that can be used to run ad-hoc debugging queries through AWS Athena.
- Checkpointing is implemented to replay from last successful record in case of failures or while publishing new codes. Retention policy is set to 24 hours in case of failures.
- Data ingestion success, latency, and throughput are monitored through AWS Cloudwatch metrics and alerts.
- Deployment of application in Amazon ECS with container registry and autoscaling services has helped in auto-scaling the application according to the traffic spike.

Technology Stack

Cloud Services

amazon S3
Amazon ECS
Amazon Athena
Amazon RDS
amazon KINESIS
Amazon CloudWatch
Terraform
Buildkite

Detailed Architecture



Outcomes

- Time taken to onboard new integrations was reduced from 6 weeks to 2 weeks.
- Constant monitoring of critical data of business was performed through Cloudwatch to ensure that everything is in order.
- Stream based architecture can be leveraged to provide real-time analysis through graphical representation.

HashedIn has helped many promising firms across the globe by building customized solutions to give the users a completely hassle-free experience. Kindly let us know if you have any specific problem/use case, where we can provide more information or consult you.

https://hashedin.com/contact-us/