Fairest Feline

Voting System

Welcome to the world’s greatest cat competition, where you will get the chance to cast your vote to choose who you think is

THE FAIREST FELINE

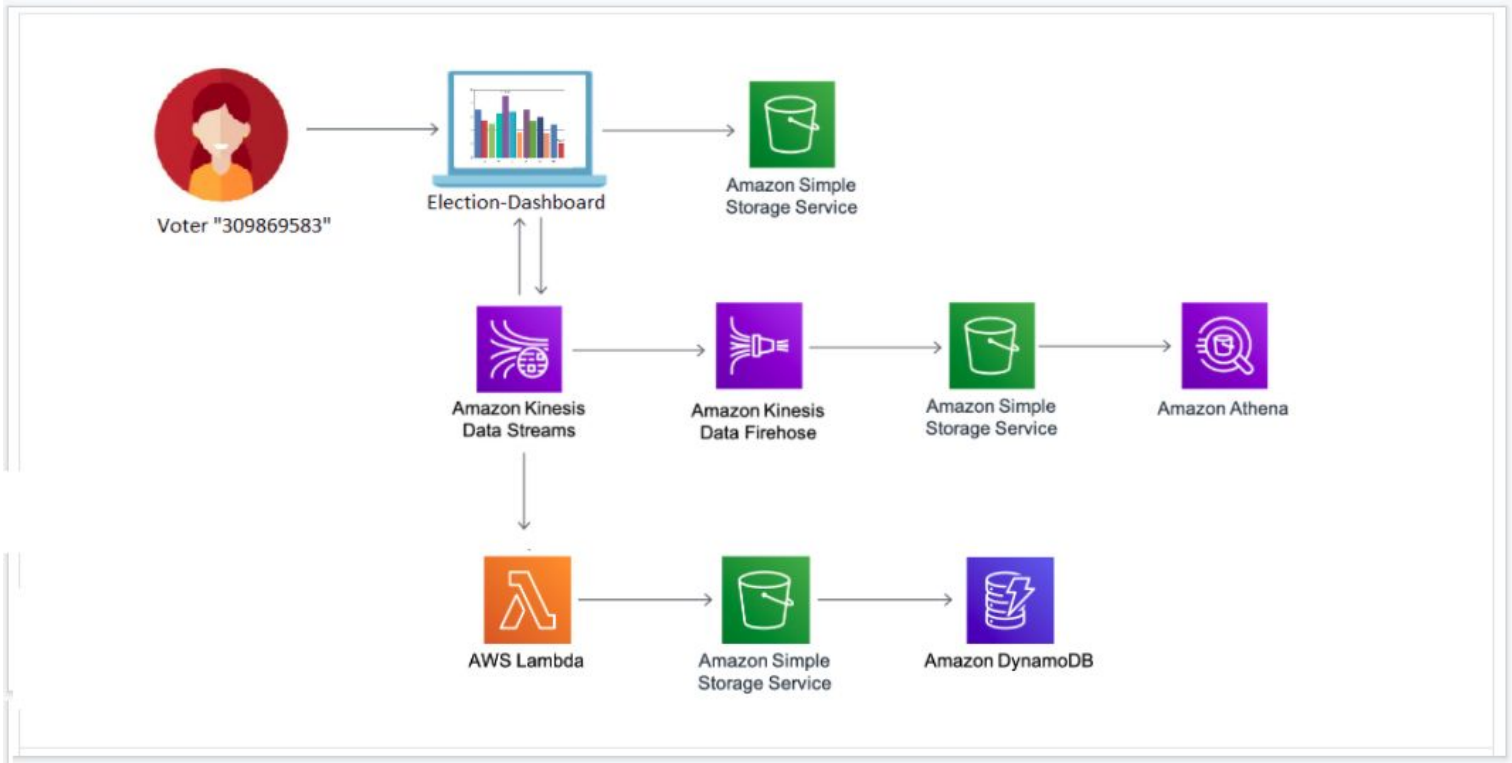
Intro

For our project we chose to create a voting system for a cat election.

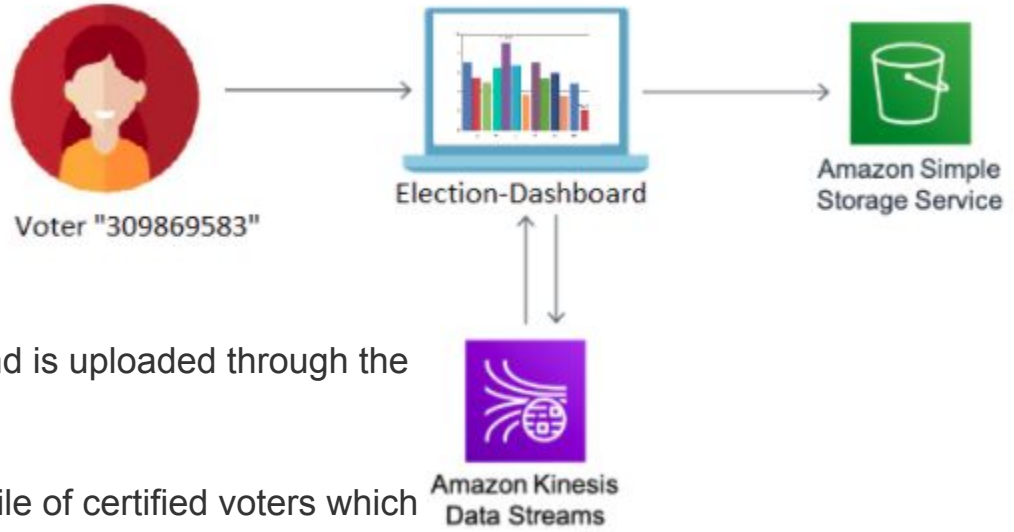
You can cast your vote and view the current status of the election at any time by accessing the Election Dashboard.

The votes are sent through a Kinesis Data Stream, stored in S3 and queried through Athena.

Application Architecture



Voting



Voting is done through the Election Dashboard and is uploaded through the Kinesis stream “FFDataStream”.

The ID with which you vote is checked against a file of certified voters which is stored in S3. If the ID isn’t in the file, you are not certified to vote and thus your vote is invalid and will not be counted.

Your ID is also checked against previous votes. If the ID was already used to vote, your duplicate vote will not be counted.

All votes are uploaded to the stream, including invalid and duplicate votes. However, only valid votes will be defined as “valid”.

Storage

We used Kinesis Data Firehose to flush the raw data to an S3 bucket for archival purposes. Using Athena, we can run SQL queries against the raw data for ad-hoc analyses. In addition, we used a lambda function to send the data to a backend database stored in DynamoDB and run queries against those data.

We chose to use both Athena and DynamoDB due to their different query abilities. We can use Athena for more complicated SQL queries while using DynamoDB for simpler queries that need high performance and low latency.

Election Status

The Election Dashboard receives all the votes from the stream “FFDataStream”.

However, it only processes those that are “valid” and counts them in the graph.