Serverless Vote Data Edison Research Modernization of Election Coverage

Edison Research utilizes AWS Platform Services to Crush the Competition for Media Coverage of US Elections

Tabulated Vote Coverage

Edison Research is the sole provider of exit poll coverage for every major midterm, presidential primaries and general elections in the United States. They were tasked with now taking in the tabulated votes published by the counties, precincts and states throughout the country for live analysis and distribution for the major broadcast networks in the US.

The Challenge

Each district and county independently start publishing vote results throughout the vote counting process on election night using various media formats and end points. Some sources publish the format before the election while others release vote data only on election night. There are also several sources for the same data that need to have automated quality control algorithms filter data to pick the latest results in order to ensure the data gets reported as quickly as possible to the news organizations for analysis. The technology for vote sources varies from web APIs to having a link to an excel file, FTP sites or results on a web page.

For some sources, the methods and formats for the published data are known. In other cases, this is not known until election day. There are also many cases where the data formats and method of distribution are different on election day than what was communicated beforehand.

Since the data formats and distribution methods may change on election day, this poses a significant challenge.

The software to pull data, extract, transform and store it, in some cases, has to be developed on the fly, tested and deployed into production in near real-time in order to be competitive.

aws PARTNER Advanced Tier Services

edison research

About Edison Research

Businesses, governments, news operations – they all need to know the trends affecting their businesses, the opinions of the electorate, and the attitudes of their customers. From Fortune 500 companies to start-ups, from the United States Government to nonprofits, Edison's team of researchers and thought leaders is trusted in America and around the world to gather exactly the right information and to use it in exactly the right way.

With expertise in both quantitative and qualitative research, Edison utilizes telephone, Internet, and inperson research. And our network of more than 19,000 experienced interviewers allows us to conduct research in almost any location.

The Results

Edison was consistently ahead of their competition covering Elections. This gave them and their clients, the National Election Pool, an advantage not only in having the latest and most accurate data but also in being able to perform complex analytics to predict outcomes before anyone else.

This new solution also provided flexibility to be able to take data in from many different sources and implement and deploy processing layers as they were discovered on Election Night.

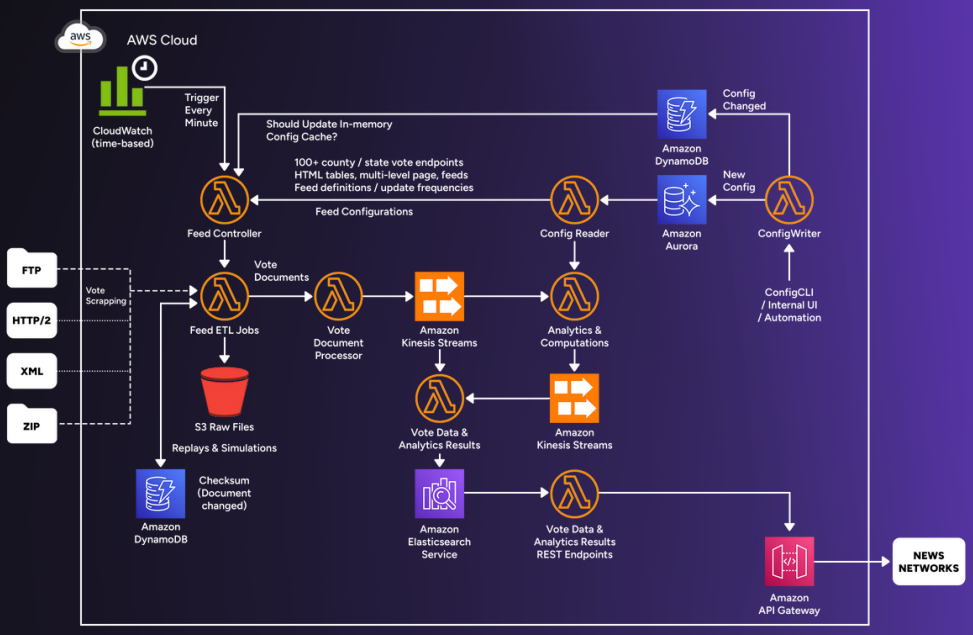
This competitive advantage opened the doors for even deeper, more insightful analytics that was out of reach before.

National Election Pool

Conducted by

Serverless Solution Workflow

Vote Document Feeds Processing



Staying Ahead of the Competition

To determine how to approach this daunting challenge, Edison called in the experts from ThorTech Solutions, an Amazon Select Tier Partner with over 10 year's experience with building solutions on AWS. ThorTech identified the "Feed and Fetch" aspect of this problem and broke down the data workflow space into several individual layers:

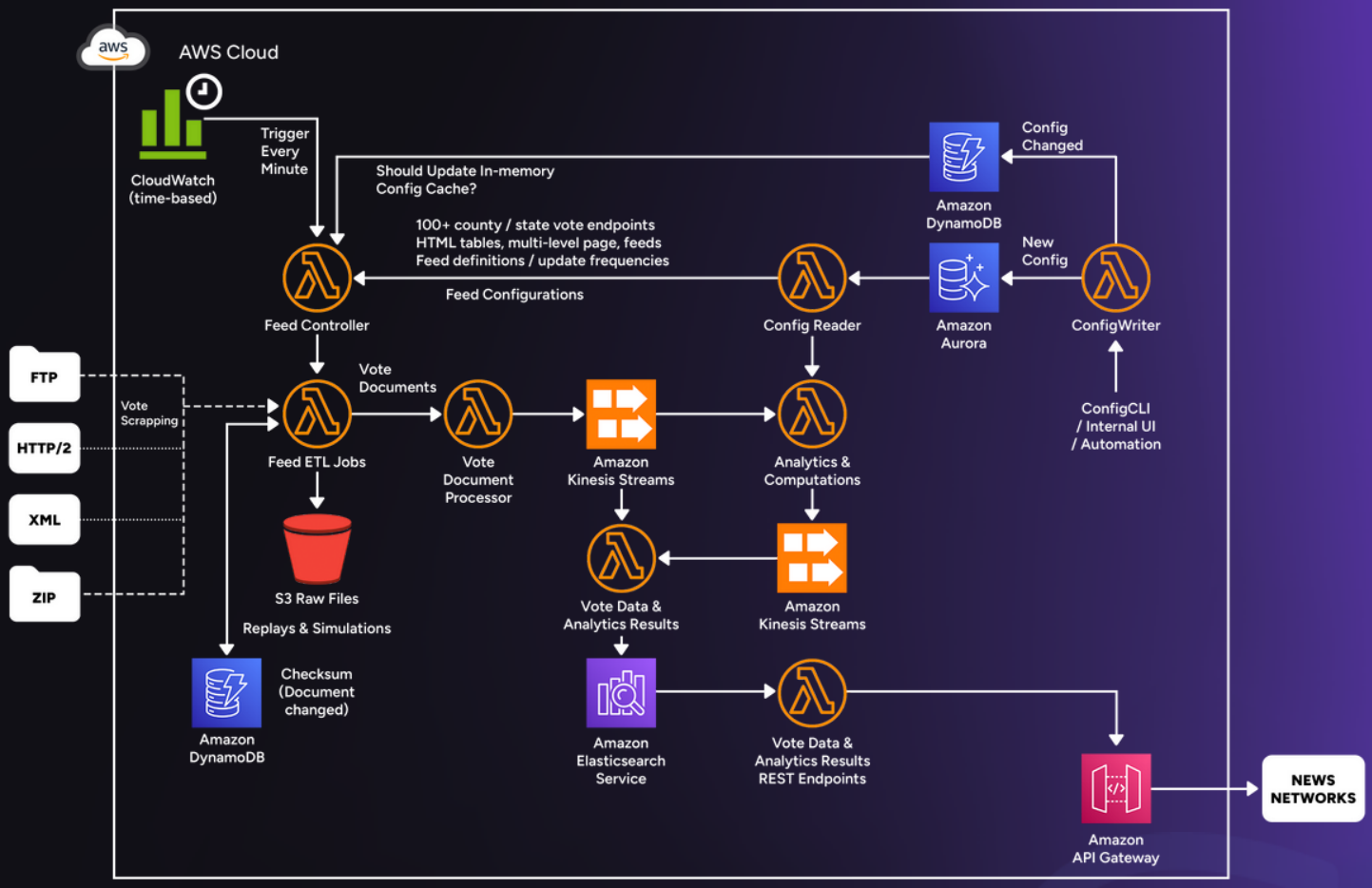
Data Acquisition

There are many different methods for collecting the data. This step in the workflow deals solely with how to fetch or supply push endpoints for storing the raw data from the source into S3.

Data Extraction

Once the data is in S3, the extraction is about taking the data from its origin format and converting it into a common format to be passed on to the transformation layer.

Serverless Solution Workflow
Vote Document Feeds Processing



Staying Ahead of the Competition

To determine how to approach this daunting challenge, Edison called in the experts from ThorTech Solutions, an Amazon Select Tier Partner with over 10 year's experience with building solutions on AWS. ThorTech identified the "Feed and Fetch" aspect of this problem and broke down the data workflow space into several individual layers:

Data Acquisition

There are many different methods for collecting the data. This step in the workflow deals solely with how to fetch or supply push endpoints for storing the raw data from the source into S3.

Data Extraction

Once the data is in S3, the extraction is about taking the data from its origin format and converting it into a common format to be passed on to the transformation layer.

Data Mapping

The mapping layer defines how the source data items get mapped into the target system. The data mapping is a combination of automated fuzzy logic and manual review to attempt to identify the physical location (county/precinct/ state, etc) of the source data.

Data Transformation

The transformation layer uses the common format from the Data Extraction Layer and the information from the Data Mapping layer to create a payload for direct ingestion into the rest of the system.

Each of these layers had to be able to satisfy several constraints:

- Must be simple to create or modify an instance of these for any given source type such that these layers could be authored within minutes.
- Each of the layers must be able to be deployed within minutes of source code being committed to version control.
- The monitoring of data processing failures must provide functionality to enable proactive instead of reactive response.
- No failure within any layer of the hundreds of sources running could interfere with the system performance or stability in any way.

ThorTech designed and implemented an event driven Serverless solution. Using AWS lambda, kinesis streams and other platform services, we were able to achieve a near real-time ingestion of data, scrub it, match it to the internal identities, perform quality control, use it in computations and publish the results to the national media organizations ahead of their competition.

Since the data also comes in very disparate times, the Serverless architectures 'pay for only what you use' is a powerful way to be able to scale up for the volume of a general election and save money in between events and for handling smaller events. There is also zero tolerance for downtime.

Being able to rely on AWS platform services eliminated having to worry about servers and greatly simplified the high availability requirements of the application.

Serverless Solution Workflow

A serverless architecture was designed and built using AWS lambdas to thread through the raw vote data into the statistical results used for projecting the election winner. A cloud watch event kicks off the feed controller lambda. It loads 100+ county endpoints definitions and starts ETL lambdas for all endpoints.

This lambda detects changes in the document and transforms into a compatible vote document and published to Amazon Kinesis Streams. From there, downstream vote data analytics lambdas and Computation lambdas are subscribed to Kinesis streams to continue the processing and trigger publishing lambdas to eventually store the results to Elastic search service.

Rest endpoints within API Gateway are exposed which allow customers to import data into their systems for further processing to make it newsworthy and reporting on-air to their viewers.

We follow recommended best practices for our security. All resources are contained in one of our two VPCs, which being peered guarantees secure communication. All IAM policies being utilized by the services are done following the principle of least privilege. Network access to resources, such as databases, is controlled using Security Groups following a provider/consumer paradigm. Finally, our S3 buckets are encrypted using custom KMS keys.

Monitoring of Alarms and Metrics of AWS Lambda functions

Events and log statements are written into Kinesis. A Splunk reader for AWS ingests this data into Splunk Cloud where dashboards provide alarms and metrics for monitoring all the activity throughout the Serverless processing. Utilizing meta data in the log statements we are able to monitor the health of all aspects of the system. When a Lambda invocation fails, the exception is logged and there is retry logic with Exponential backoff logic to facilitate retires without saturating the system.

To provide quick turn around on errors we required that all lambda payloads be logged to S3 and all lambdas have the ability to be run locally against data in S3 as if it were fed it in real time.

Concurrency

There are hundreds of concurrent Lambdas running. To manage this high level of parallelism and avoid unnecessary Lambda executions, and cost, we employed a process to guarantee we only trigger processing when new data is presented. Each new payload is first hashed in its entirety and then compared using DynamoDB's Conditional Write operation, which coordinates whether or not the processing should continue This atomic operation guarantees we only trigger once invocation per payload that we have not seen yet and avoids processing the same data more than once.

Deployment - Speed is Critical

Deployments are automated utilizing Jenkins. Jenkins builds and packages the code and uploads it to S3. A CloudFormation is updated to point to the new URI which contains the code for the Lambda. This process occurs within seconds of the code being pushed. Each of the layers defined above can be worked on and deployed individually or as a group. This enabled us to react to changes on Election Night; expeditiously patching live production code to stay ahead of the competition.

Performance/load testing

There is a significant difference in the amount of data that needs to be processed for a General Election vs a Presidential Primary or a Mayoral race. General Election data can generate hundreds of millions of data points where there is a fraction of that for single state elections or presidential primaries. The testing needed to ensure it would scale up and down for these different scenarios. Static test data sets were generated using historical data. Tooling was built to feed data into the system at different speeds over a virtual time to simulate the data patterns that would occur on election day.

Data was compressed in some cases, up to 20x expected load to ensure that spike in processing would not interfere with the processing time of getting vote reports into the system.