## AWS Avista Blog Post

**How Avista uses AWS Glue and Apache Hudi to derive insights from utility meter data.**

*Introduction*

Utilities produce vast quantities of data, from the hydro and thermal plants that generate the power, to the transmission and distribution systems that transport the power, down to the meters that record customer consumption of that power. This information, combined with customer records, billing information, equipment condition, and more must be combined into a comprehensive Enterprise Data Warehouse (EDW) capable of providing insights and recommended actions in today’s ever-changing utility landscape.

Avista Utilities – an investor-owned vertically-integrated electric and natural gas utility headquartered in Spokane, WA – sought a cost-effective way to create an AWS cloud-based data lake capable of near-real-time upserts that could be leveraged across multiple AWS query, processing, and machine learning toolsets. In this blog post, Avista and AWS showcase how Avista uses AWS services and Apache Hudi to build a data pipeline with CDC capabilities. This post describes and demonstrates how to take advantage of the serverless architecture of AWS Glue and S3, while still maintaining the capability of upserting data into a data lake.

*The Problem*

As previously stated, utilities generate large volumes of data, including consumption and voltage data from smart meters. These datasets are utilized across multiple business use cases. Traditional data warehousing approaches weren’t scalable or performant enough to meet Avista’s requirements for ingestion, joining, and processing of data across multiple disparate systems. The existing on-prem architecture was taking one-to-two days to provide curated data to business analytics, data scientists, and other downstream applications.

\*\*Add cost context.

Avista was aware that AWS customers use AWS Data Migration Services (DMS) to replicate data from on-prem source systems into Amazon Simple Storage Service (S3). Using S3 as a DMS replication target decouples storage and compute, but applying Change Data Capture (CDC) processes to data in S3 can be challenging given the immutable nature of S3 data. One solution is to utilize an always-on RDBMS

*Solution Overview*

Using DMS, Glue, S3, Athena, RedShift Spectrum, and other managed serverless services, Avista is able to ingest, join, and transform the data from multiple source systems at scale. Curated datasets are now available within 30 minutes of change, with some data available as quickly as 5 minutes following an on-prem source system update.

Avista uses Apache Hudi with AWS Glue to apply record-level inserts, updates, and deletes from their source systems into their S3 datalake without always-on RDBMS targets. By combining the capabilities of AWS Glue, Apache Hudi, and a Glue custom connector, Avista’s solution framework provides a high-throughput data pipeline that delivers DML capabilities for curated data residing in an S3-based datalake.

\*\*In this blog post we will provide an overview of the architecture, and code that can be deployed to demonstrate the solution.

**(Include high level architecture diagram)**

\*\*Explain the flow in the diagram

1. On-prem DB changes are already being replicated using DMS to this raw bucket.
   1. DMS tasks settings, data formats
   2. DMS considerations
   3. S3 considerations
2. Glue job functionality
   1. Initial load/incremental
   2. Hudi CoW/MoR
   3. Transaction Ids.
   4. Framework - control file/DynamoDB
      1. Different settings purpose
      2. What problem is Framework solving(repeatable, automated, re-usable, on-boarding)
   5. Catalog
3. Processed output
   1. Athena Queries
   2. Redshift Spectrum
4. Observability
   1. Metrics
   2. Monitoring

**(Code deployment steps and walk through)**

\*\*Explain IaaC briefly.

Why CDK. Avista uses CF/Service Catalog, but exploring CDK in this blogpost. Costs with deployment.

1. Stack overview
2. Pre-Req stack deployment
   1. Walk through
   2. What it would look like with DMS and on-prem DB(not in code scope)
3. Glue stack deployment
   1. Walk through

**(Solution in action)**

\*\*Explain what we are trying to show.

1. Full load – Data in raw S3->Run job->Query using Athena.
2. Incremental load – Load new data in S3->Run job->Show updates
3. Touch on how the framework can be leveraged to add additional sources.

Clean-up stack

**(Conclusion, next steps)**

What has this solution achieved for your business. Any challenges? Journey. Some data points which show processing volumes, productivity.etc

----