KBS POV on Data Storage for Data Warehousing in AWS

Last Updated: May 24th, 2018

This document intends to cover high-level recommendations for cloud data warehousing storage technologies and techniques for AWS under best practices within Koch Industries.

Background

Koch has been on a path since late 2016 to move its data centers to the cloud (AWS and soon to be Azure). Data warehouses do not go away as we engage in digital transformation. They remain an important part of our data assets, providing high business value with consistent, curated and trusted data that supplements the reporting and analytics landscape of the future. The storage (or database) behind the data warehouse solution is a key component to unlocking the business value of company data at scale.

That said, KBS provides here a standard recommendation for a cloud-based data warehousing storage platform.

Evaluation

KBS completed a proof of concept and experimentation with the following top recommended data warehousing specialty database platforms in AWS:

- Redshift
- Snowflake

Though there is always the option of raw and/or big data analytics, the focus of our findings is for more traditional data warehouse methodology as we do not believe that the need for such is going away any time soon.

Recommendation: *Snowflake*

Snowflake has proven to KBS to be a disruptive service to the cloud data warehouse scene for the following reasons:

- True cloud-first data warehouse solution
- Only SaaS based AWS data warehouse platform
- Only truly auto-scaling AWS database platform
- Only AWS data warehouse platform that spins down to zero cost when not in use
- Requires no index/key management, no backup management, and no updates, removing need for traditional DBA activities

KBS will support Redshift as an alternative, but firmly believes most use cases will support leveraging Snowflake based on the combination of feature comparison (below) and TCO to the customer.

Key Decision Drivers

The key factors in KBS' position on Snowflake as the preferred data warehousing storage platform in AWS were:

- Lower TCO
  - Snowflake is purely SaaS and has lower FTE requirements
  - Snowflake provides HA/DR at no additional cost (assuming Snowflake Enterprise Edition)
- Accessibility
  - Snowflake will be multi-cloud (AWS/Azure at least, GCP unknown); Redshift will never be in any cloud provider other than AWS
  - Snowflake is SaaS and therefore can be accessed from anywhere via URL without regards to network layer controls (Redshift must live in a VPC and the company is responsible for network access controls – today, this makes Redshift difficult to access from on-prem unless deployed to a Legacy IAM account)
- Performance
  - There was no clear performance winner between the two when clusters were sized appropriately, however, Snowflake holds a distinct advantage in allowing you to run smaller cluster sizes (lower cost) until you need the larger cluster temporarily, scaling automatically to handle the load.

Direct feature Comparison:

Category	Factor	Redshift	Snowflake	Notes
Data Support	Structured	●	●	
	Semi-Structured	●	●	Snowflake has native support for XML/JSON/AVRO/PARQUET with persisted auto-generated schemas and stores objects in their entirety for performance and scalability. Redshift supports the same file types, requires manually defining schemas in AWS Data Catalog for Spectrum to query as external tables, and the data stays in S3. See External factor below.
	Unstructured	■	■	No native support for either product. Requires ELT.
	Streaming	●	●	
Querying	Performance	●	●	https://blog.fivetran.com/warehouse-benchmark-dce9f4c529c1
	Ad-Hoc	●	●	
	External	▲	▲	All data must be loaded into Snowflake to query, but Snowflake stores entire objects natively w/o flattening so performance will scale; Redshift has Spectrum to read data from S3 directly, and though data is not loaded into Redshift, there is concern about performance at scale.
	SQL	●	●	

	Concurrency	●	●	Though neither are poor, Snowflake has a distinct advantage here with the virtual data warehouses and decoupled storage/compute architecture
DevOps	Infra as Code support	●	▲	Snowflake doesn't support CloudFormation or Terraform to automate deployments, but is a SaaS solution, so this is not considered a benefit, but a feature for IaaS/PaaS
	SDK	●	▲	No AWS SDK support for Snowflake
	Monitoring	●	■	Snowflake doesn't support AWS Monitoring, it has a separate in-house monitoring tool
	CI/CD	●	●	Both will support CI/CD in their own ways. Snowflake allows for a little easier schema CI/CD as no indexing is required for performance, whereas you must make good choices early with sort/distribution keys with Redshift
	AWS Tagging	●	■	No AWS Tagging support for Snowflake
	Alerts/Thresholds/Events	●	▲	Snowflake alerting is not integrated with AWS or as inherent as Redshift
	Auto Scaling	▲	●	No native autoscaling for Redshift today but is on the roadmap. Will be done with "burst computing". Can also be achieved by IaaC, but coordination can be complicated and delivery time is slow
	Cloning	▲	●	DW can be cloned endlessly for dev/test/QA in Snowflake; Redshift requires complete cluster duplication
Availability	Across AZ	▲	●	No manual configuration needed for Snowflake. Redshift requires duplication of infrastructure and processing or replication
	Across Regions	▲	●	No manual configuration needed for Snowflake. Redshift requires duplication of infrastructure and processing or replication
	Backups/Snapshots	●	●	Though Snowflake is automatic
	Data Lineage (RPO)	■	●	Snowflake provides Time-Travel feature, so it's automated and data can be restored to any point-in-time with little effort or time constraint. Redshift requires traditional snapshot/restore and potentially complicated data re-integration/alignment
	Disaster Recovery (RTO)	▲	●	Redshift requires cluster snapshots to be restored
	Multi-Cloud	■	●	Snowflake recently announced an Azure offering and has mentioned

				Google. Redshift will only ever be in AWS
Security	IAM/LDAP	▲	●	LDAP and Federated Identity are supported by both, however SAML implementation for Redshift is “Active” and therefore not supported by Koch. Snowflake does not support IAM roles from AWS today, but is on the roadmap
	Encryption	●	●	
	RBAC	▲	●	Both support RBAC, but controls and management are more robust in Snowflake
Knowledge	Talent	●	▲	Talent is readily available for Redshift in the market
	Ease of Use	▲	●	Snowflake’s browser experience is much easier to understand and use than the AWS console. Snowflake also includes a query tool in the browser GUI, so users need not maintain a separate query application
	FTE Requirements	▲	●	Traditional DBA resources required to support Redshift, not for Snowflake. Redshift roadmap has some automated features coming that will supposedly reduce this dependency
	Product Maturity	●	▲	Redshift has been around since 2013 but based on PostgreSQL 8.1 which was released in 2005; Snowflake was first offered in 2014
	Training Resources	●	●	
	Documentation	●	●	
Integration	With AWS services	●	▲	Redshift works seamlessly with other AWS services; Snowflake does not
	ETL/ELT	●	●	
	BI/Dashboards	●	●	
	Data Science	●	●	
	JDBC/ODBC	●	●	
Vendor	Maturity	●	▲	AWS is Amazon, around since 1996; Snowflake is strong, around since 2012, but does not have the same stature as the big 3
	Financial Strength	●	▲	AWS is worth $691B; Snowflake is worth $1.5B

Total Cost of Ownership (TCO)

As with all major hardware/software investments, you must weigh more than the infrastructure and licensing costs alone. This is an area where Redshift and Snowflake, today, differ by a reasonable margin. Because of the unique benefits of SaaS (Software as a Service) over PaaS (Platform as a Service), the needle, though clearly pointing to Redshift as the winner for pure platform cost, the TCO for the products tells a different story.

To illustrate this POV, we chose several common use cases for data warehousing in place today, and what we see as future state.

TCO comparisons (Annual)

			Platform Cost		Labor Cost		TCO	
Use Case	**Definition**		**Redshift**	**Snowflake**	**Redshift**	**Snowflake**	**Redshift**	**Snowflake**
Small Data Warehouse – Batch, no load/use overlap	Database Size: Load Duration: Avg Daily Use: Analysts: Report Consumers: Data Scientists:	100GB 1 Hr 8 Hrs 2 50 0	2 Node dc2.large **$3k* or $6k**	1-2x Extra Small Multi-Cluster **$8k**	1/20 FTE **$9k**	1/100 FTE **$1.8k**	**$12k* or $15k**	**$9.8k**
Medium Data Warehouse – Batch, no load/use overlap	Database Size: Load Duration: Avg Daily Use: Analysts: Report Consumers: Data Scientists:	1TB 4 Hrs 16 Hrs 10 100 1	8 Node dc2.large **$13k* or $26k**	1-3x Extra Small Multi-Cluster **$26.5k**	1/20 FTE **$9k**	1/100 FTE **$1.8k**	**$22k* or $35k**	**$28.3k**
Medium Data Warehouse – Real-Time/ Streaming	Database Size: Load Duration: Avg Daily Use: Analysts: Report Consumers: Data Scientists:	1TB Constant 20 Hrs 10 100 1	8 Node dc2.large **$13k* or $26k**	1-3x Extra Small Multi-Cluster **$29.3k**	1/15 FTE **$13.5k**	1/75 FTE **$2.4k**	**$27k* or $39.5k**	**$31.7k**
Large Data Warehouse – Batch, no load/use overlap	Database Size: Load Duration: Avg Daily Use: Analysts: Report Consumers: Data Scientists:	20TB 3 Hrs 12 Hrs 30 500 2	12 Node ds2.xlarge **$65k* or $130k**	1-5x Small Multi-Cluster **$60k**	1/10 FTE **$18k**	1/75 FTE **$2.4k**	**$83k* or $148k**	**$62.4k**
Large Data Warehouse – Real-Time/ Streaming	Database Size: Load Duration: Avg Daily Use: Analysts: Report Consumers: Data Scientists:	20TB Constant 16 Hrs 50 1000 2	12 Node ds2.xlarge **$65k* or $130k**	1-3x Medium Multi-Cluster **$84k**	1/8 FTE **$22k**	1/50 FTE **$3.6k**	**$87k* or $152k**	**$87.6k**

* Cost **without** Disaster Recovery

Assumptions:

- Labor calculated @ $180k fully encumbered, annually (Redshift requiring DBA & developer labor, Snowflake only developer)
- Redshift cost estimates based on reserved node pricing + 18% Koch discount
- Snowflake cost estimates based on Enterprise Edition + 20% discount