CC Week #2 HR-Muhammad.drawio

File Edit View Arrange Extras Help All changes saved

Share

50%

Search Shapes

Scratchpad

Drag elements here

General

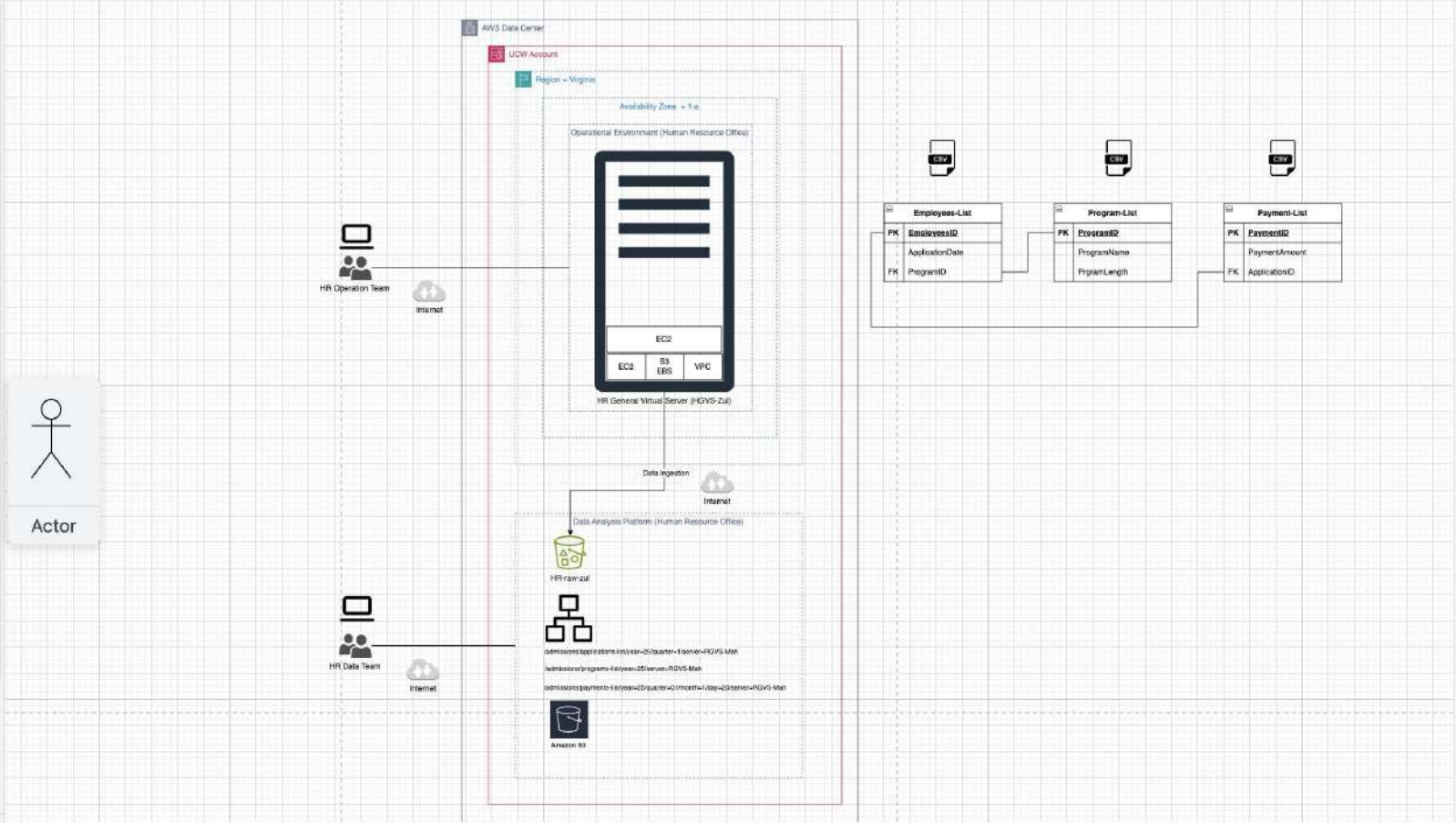
Misc

Advanced

Basic

Arrows

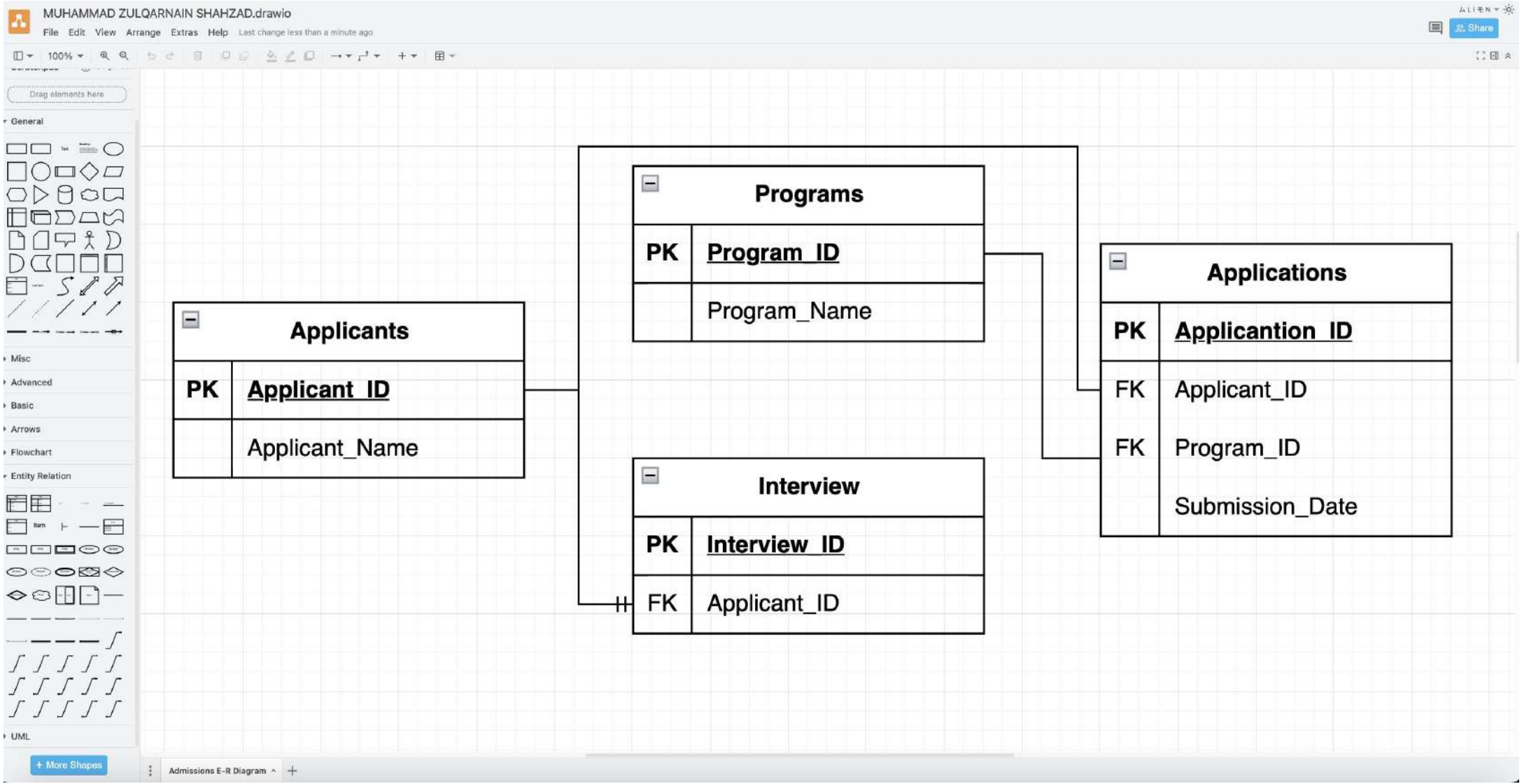
+ More Shapes

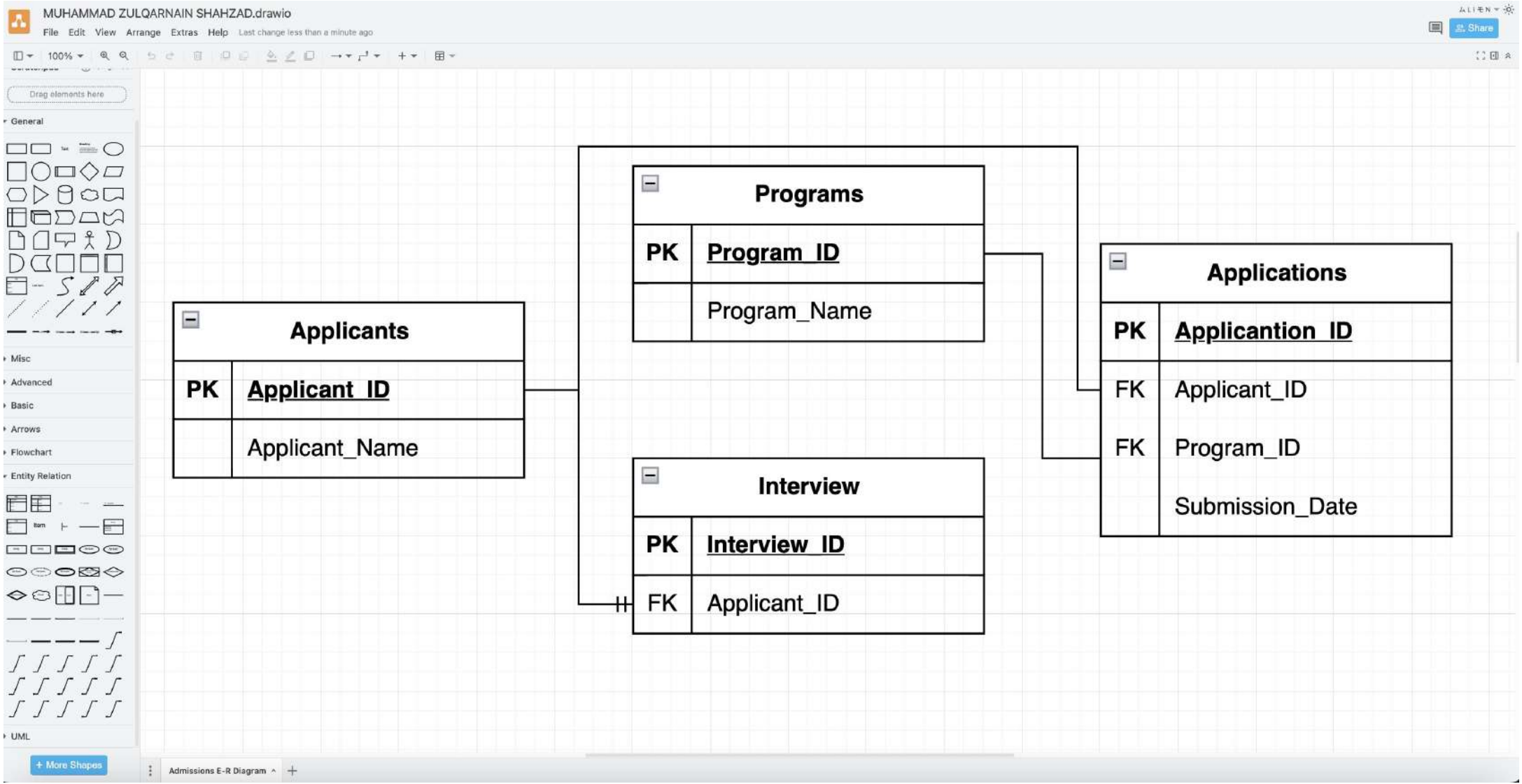


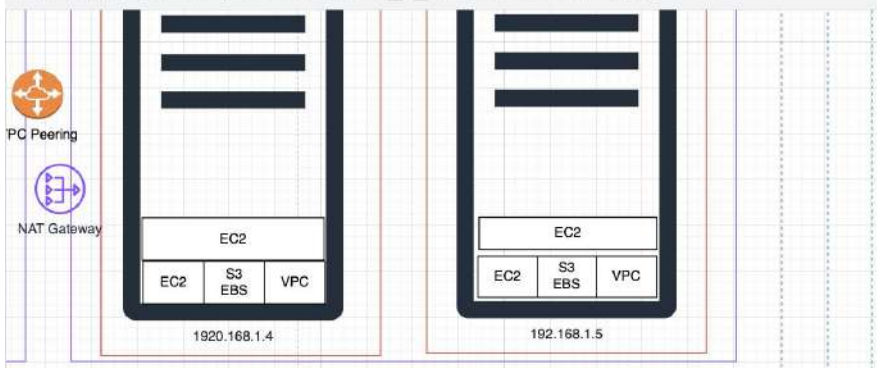
Actor

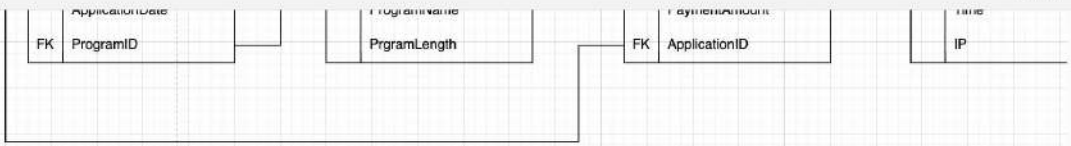
CC Solution #1

CC Solution #2

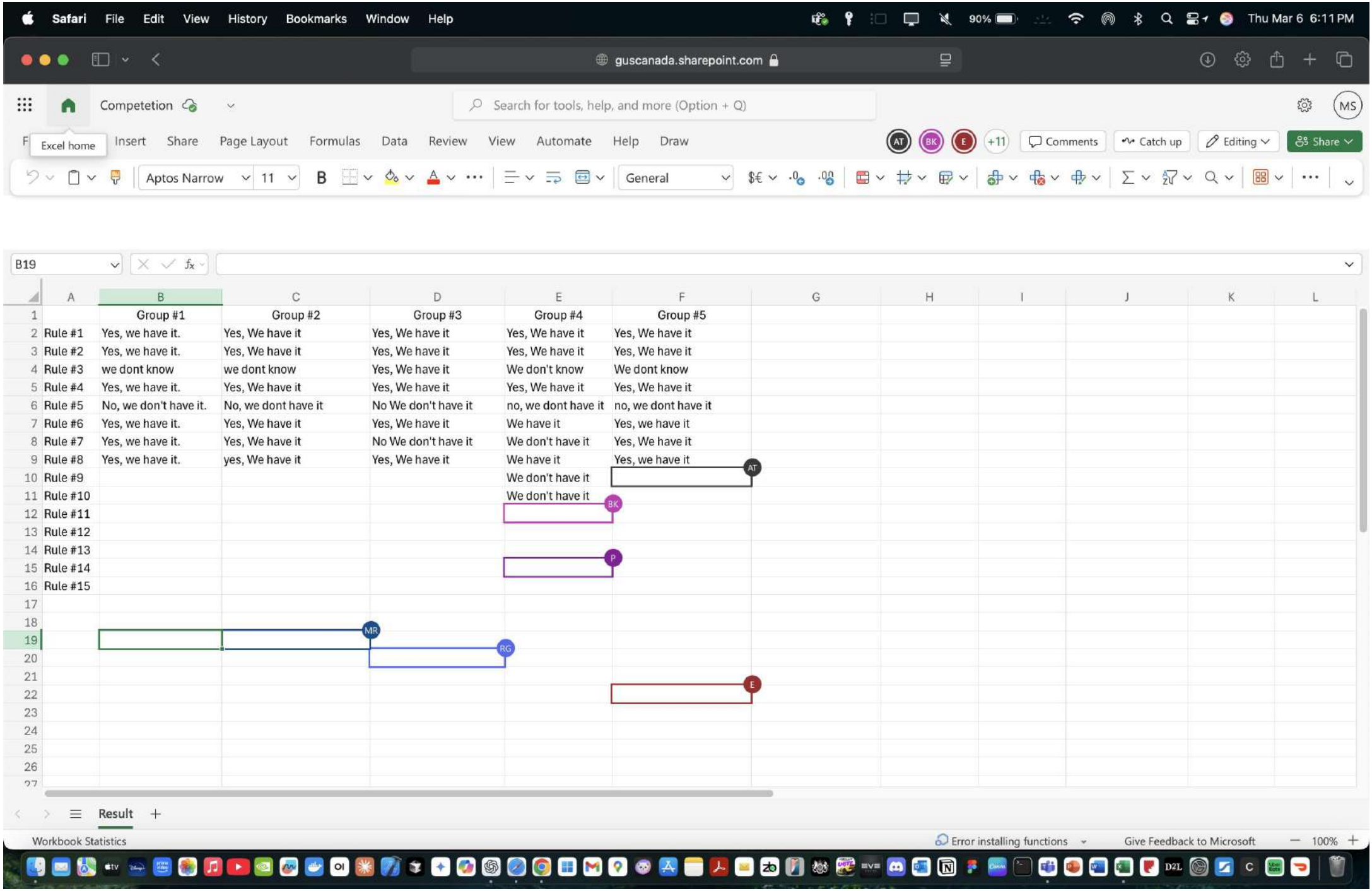

MUHAMMAD ZULQARNAIN SHAHZAD.drawio
File Edit View Arrange Extras Help Last change less than a minute ago
Share
Drag elements here
General
Misc
Advanced
Basic
Arrows
Flowchart
Entity Relation
UML
+ More Shapes
Admissions E-R Diagram
Applicants
PK Applicant_ID
Applicant_Name
Programs
PK Program_ID
Program_Name
Interview
PK Interview_ID
FK Applicant_ID
Applications
PK Applicantion_ID
FK Applicant_ID
FK Program_ID
Submission_Date


MUHAMMAD ZULQARNAIN SHAHZAD.drawio
File Edit View Arrange Extras Help Last change less than a minute ago
Share
Drag elements here
General
Misc
Advanced
Basic
Arrows
Flowchart
Entity Relation
UML
+ More Shapes
Admissions E-R Diagram
Applicants
PK Applicant_ID
Applicant_Name
Programs
PK Program_ID
Program_Name
Interview
PK Interview_ID
FK Applicant_ID
Applications
PK Applicantion_ID
FK Applicant_ID
FK Program_ID
Submission_Date

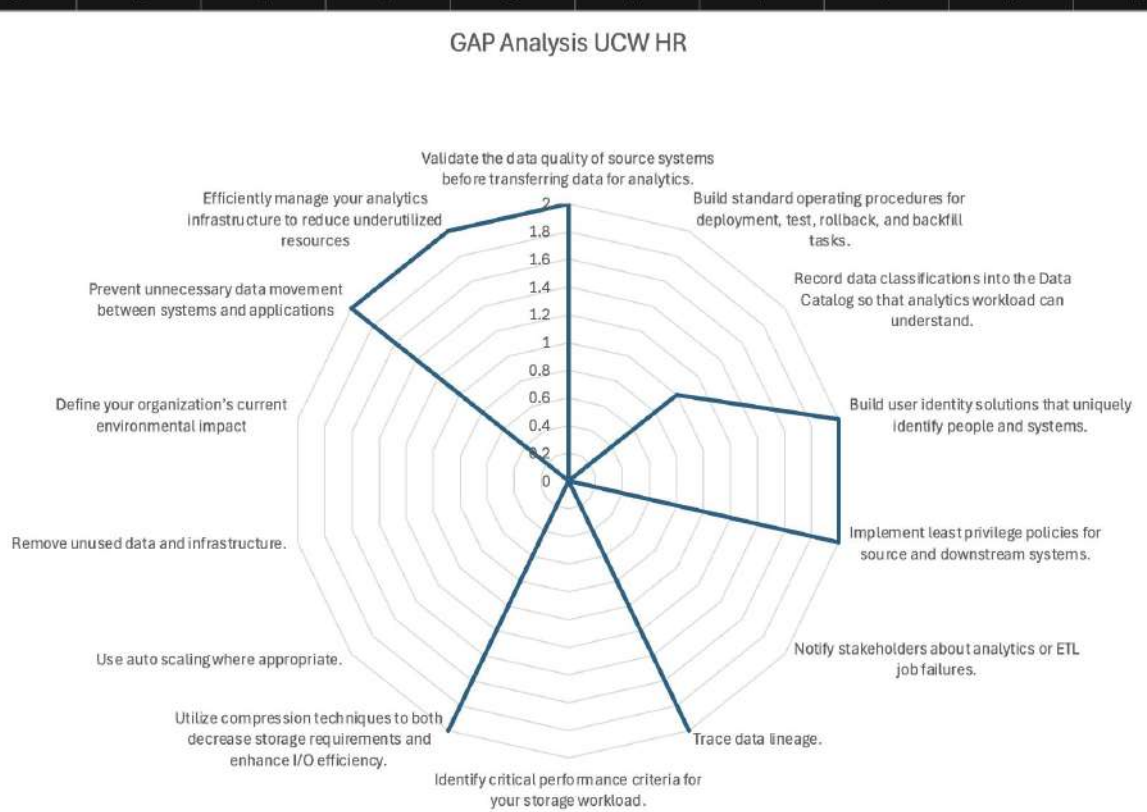




Data Protection Techniques	
Goal	Items
Identify Resources	All Datasets, All Buckets, All EC2 Instances. Data Catalog, All ETL Jobs (pipelines), All Cleaning Jobs,
Identify Containers	All empty S3 Buckets, All Servers, All EC2 Instances without , All Empty Cleaning jobs, Empty Data Catalog
Identify Contant	Data Sets, Data Catalog with Data, Contents of Cleaning jobs (Data Brew) Contents of ETL jobs
Identify Users	Registrar Operation Team, Registrar End User, Registrar Data Team User
Identification Phase	User Name is ID for Human, IP is ID for Computers
Authorazation Phase	Role C for group, Role B for individual, Role D for group, Role A for computer(individual)
C I A	Keys, in AWS key management service
Avalibility	All Buckets
Control Access	Isolation: Buckets, VPC, subnet IP range, Rules, Internet Gateway, other rules
Rules	Internet Gatwway, NAT Gateway


Safari File Edit View History Bookmarks Window Help
Thu Mar 6 6:11 PM
guscanada.sharepoint.com
Competetion
Search for tools, help, and more (Option + Q)
Excel home
Insert Share Page Layout Formulas Data Review View Automate Help Draw
Comments Catch up Editing Share
Aptos Narrow 11
General
B19
Group #1 Group #2 Group #3 Group #4 Group #5
Rule #1 Yes, we have it. Yes, We have it Yes, We have it Yes, We have it Yes, We have it
Rule #2 Yes, we have it. Yes, We have it Yes, We have it Yes, We have it Yes, We have it
Rule #3 we dont know we dont know Yes, We have it We don't know We dont know
Rule #4 Yes, we have it. Yes, We have it Yes, We have it Yes, We have it Yes, We have it
Rule #5 No, we don't have it. No, we dont have it No We don't have it no, we dont have it no, we dont have it
Rule #6 Yes, we have it. Yes, We have it Yes, We have it We have it Yes, we have it
Rule #7 Yes, we have it. Yes, We have it No We don't have it We don't have it Yes, We have it
Rule #8 Yes, we have it. yes, We have it Yes, We have it We have it Yes, we have it
Rule #9 We don't have it
Rule #10 We don't have it
Rule #11
Rule #12
Rule #13
Rule #14
Rule #15
AT
BK
P
MR
RG
E
Result
Workbook Statistics
Error installing functions
Give Feedback to Microsoft
100%

	A	B
1	Validate the data quality of source systems before transferring data for analytics.	2
2	Build standard operating procedures for deployment, test, rollback, and backfill task	0
3	Record data classifications into the Data Catalog so that analytics workload can unde	1
4	Build user identity solutions that uniquely identify people and systems.	2
5	Implement least privilege policies for source and downstream systems.	2
6	Notify stakeholders about analytics or ETL job failures.	0
7	Trace data lineage.	2
8	Identify critical performance criteria for your storage workload.	0
9	Utilize compression techniques to both decrease storage requirements and enhance	2
10	Use auto scaling where appropriate.	0
11	Remove unused data and infrastructure.	0
12	Define your organization's current environmental impact	0
13	Prevent unnecessary data movement between systems and applications	2
14	Efficiently manage your analytics infrastructure to reduce underutilized resources	2



CloudTrail

Dashboard
Event history
Insights
Lake
Dashboards
Query
Event data stores
Integrations
Trails

Settings

Pricing
Documentation
Forums
FAQs

hr-pd-tra-zul

Delete | Stop logging

General details

Edit

Trail logging
Logging

Trail name
hr-pd-tra-zul

Multi-region trail
Yes

Apply trail to my organization
Not enabled

Trail log location
aws-cloudtrail-logs-842597868275-69146130/AWSLogs/842597868275

Last log file delivered
March 19, 2025, 16:13:14 (UTC-07:00)

Log file SSE-KMS encryption
Not enabled

Log file validation
Disabled

Last file validation delivered
-

SNS notification delivery
Disabled

Last SNS notification
-

CloudWatch Logs

Edit

No CloudWatch Logs log groups

CloudWatch Logs is not configured for this trail

Tags

Manage tags

Key	Value

No tags

No tags associated with this trail

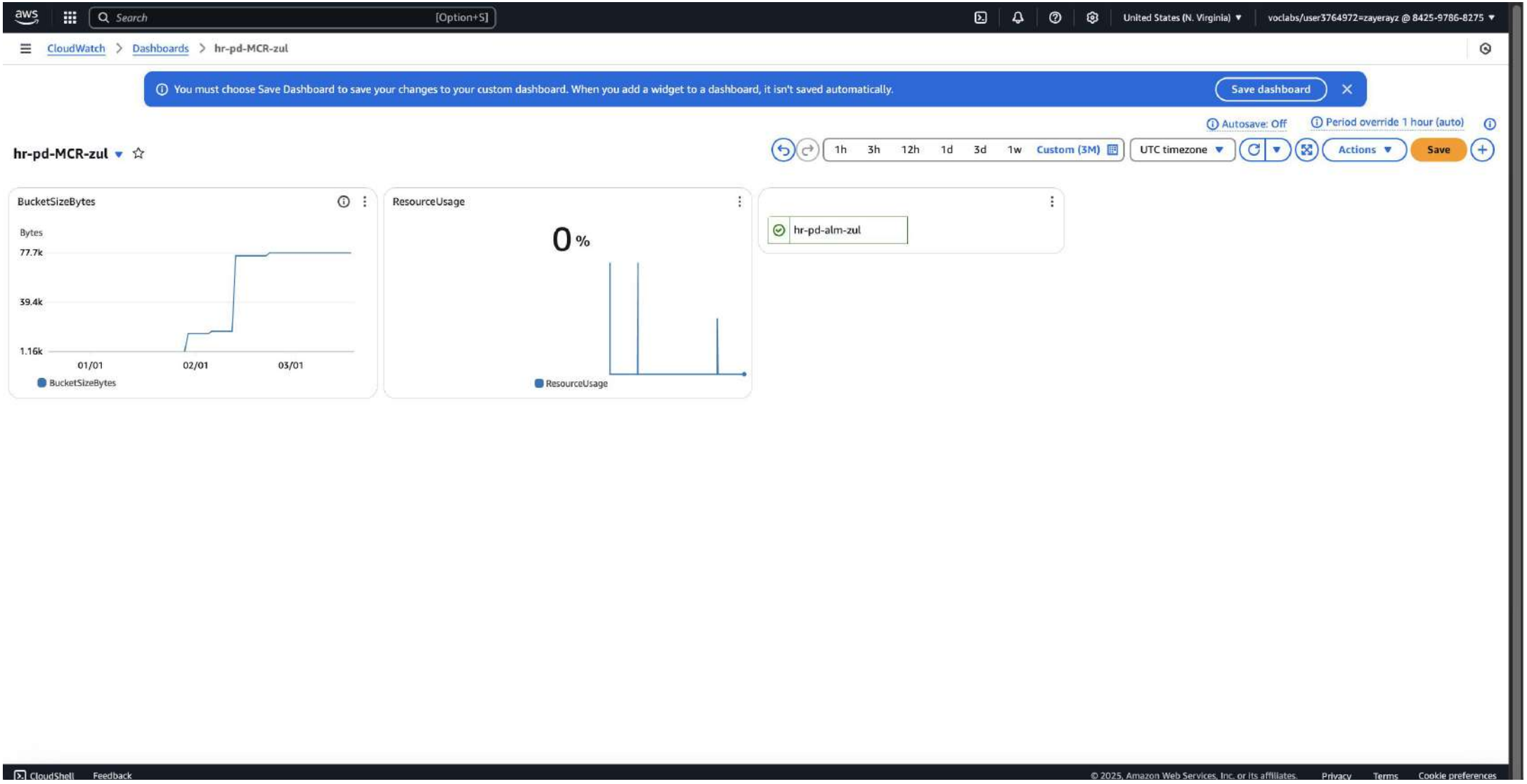
Management events

Edit

API activity
All

Exclude AWS KMS events
No

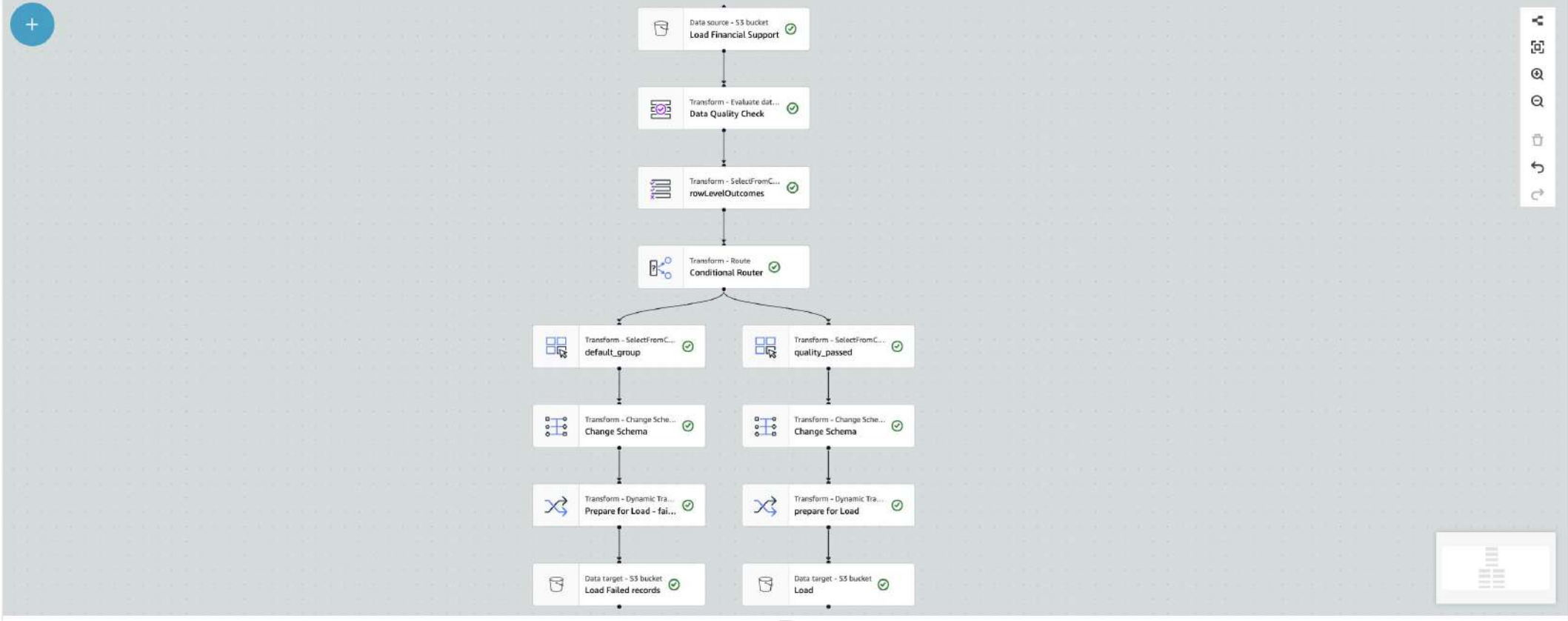
Exclude Amazon RDS Data API events
No

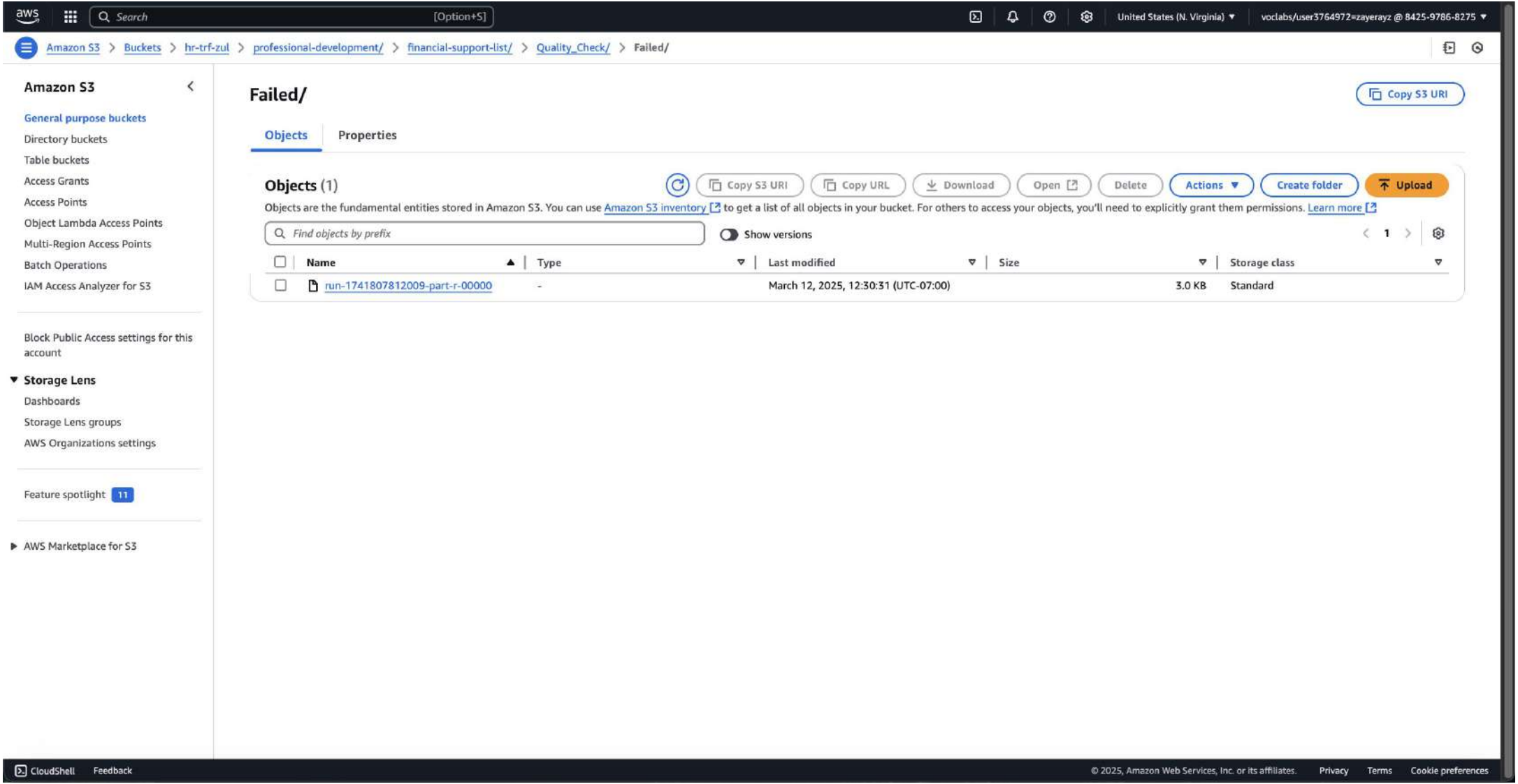

aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
CloudWatch > Dashboards > hr-pd-MCR-zul
You must choose Save Dashboard to save your changes to your custom dashboard. When you add a widget to a dashboard, it isn't saved automatically.
Save dashboard
Autosave: Off
Period override 1 hour (auto)
hr-pd-MCR-zul
1h 3h 12h 1d 3d 1w Custom (3M)
UTC timezone
Actions
Save
BucketSizeBytes
Bytes
77.7k
39.4k
1.16k
01/01
02/01
03/01
BucketSizeBytes
ResourceUsage
0 %
ResourceUsage
hr-pd-alm-zul
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

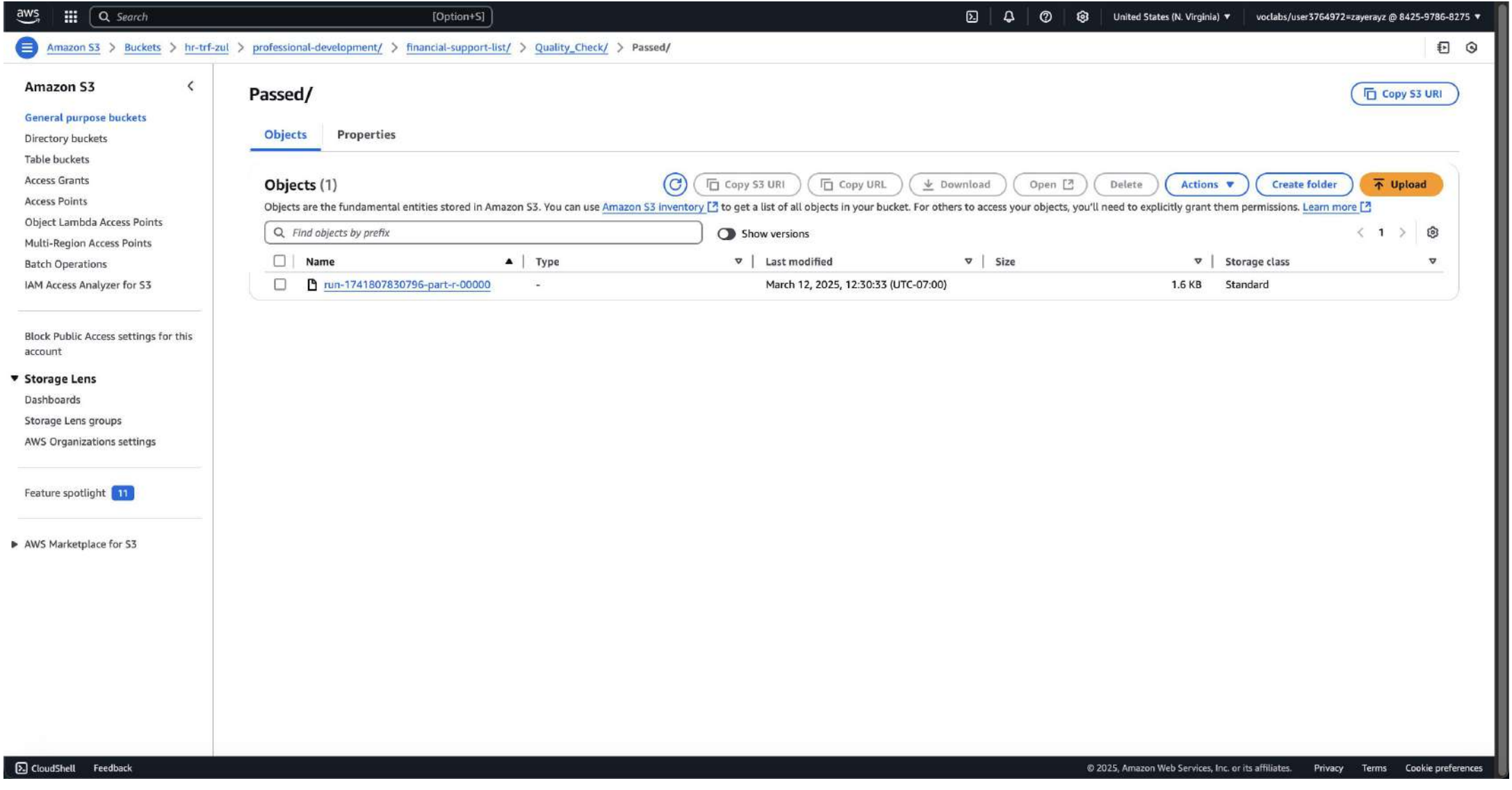
hr-pd-QC-Zul

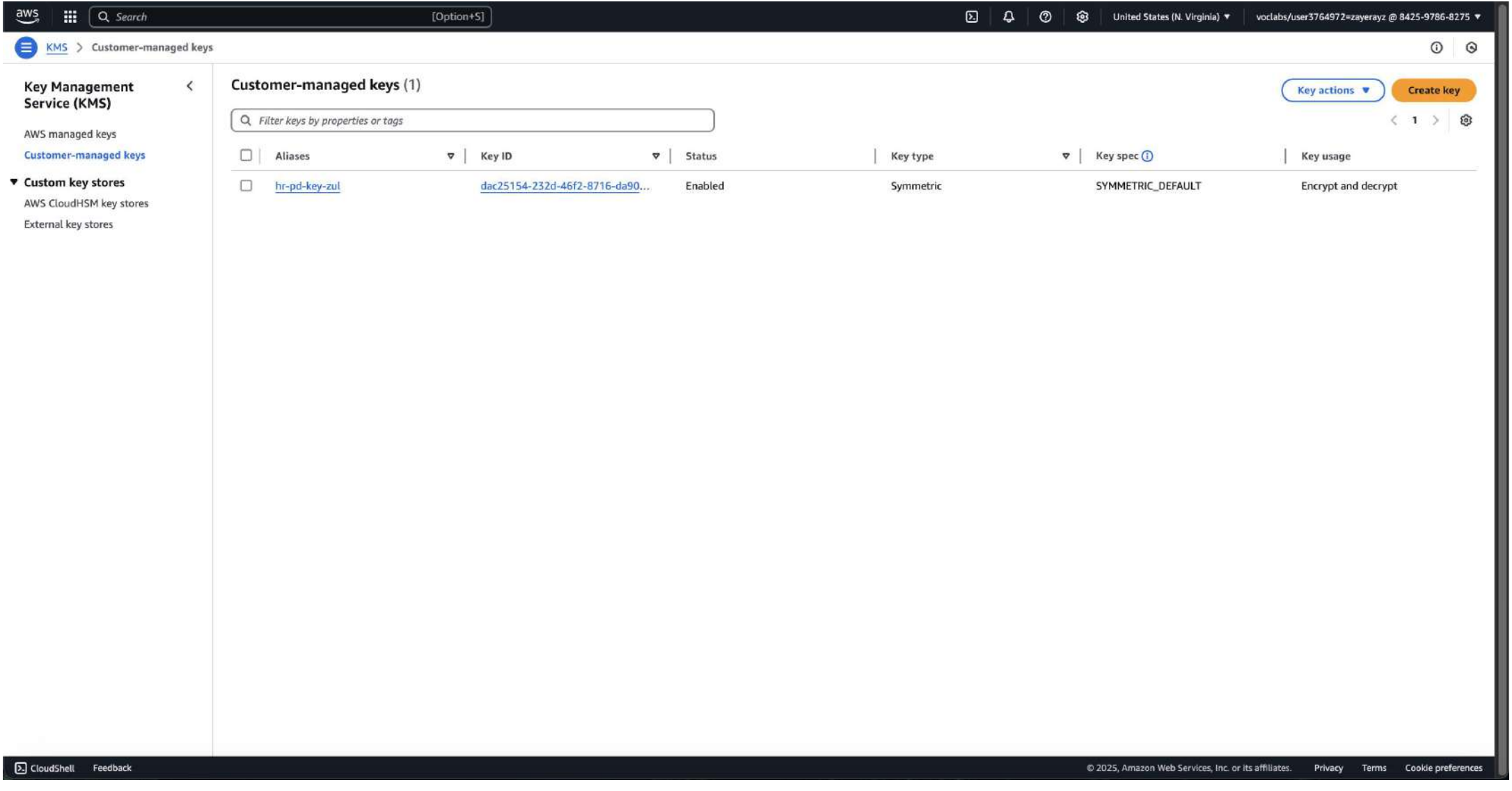
Last modified on 3/12/2025, 12:28:39 PM | Actions | Save | Run

Visual | Script | Job details | Runs | Data quality | Schedules | Version Control




aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > Buckets > hr-trf-zul > professional-development/ > financial-support-list/ > Quality_Check/ > Failed/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
Failed/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Show versions
1
Name
Type
Last modified
Size
Storage class
run-1741807812009-part-r-00000
-
March 12, 2025, 12:30:31 (UTC-07:00)
3.0 KB
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > Buckets > hr-trf-zul > professional-development/ > financial-support-list/ > Quality_Check/ > Passed/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
Passed/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Show versions
1
Name
Type
Last modified
Size
Storage class
run-1741807830796-part-r-00000
-
March 12, 2025, 12:30:33 (UTC-07:00)
1.6 KB
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
KMS > Customer-managed keys
Key Management Service (KMS)
AWS managed keys
Customer-managed keys
Custom key stores
AWS CloudHSM key stores
External key stores
Customer-managed keys (1)
Key actions
Create key
Filter keys by properties or tags
1
Aliases
Key ID
Status
Key type
Key spec
Key usage
hr-pd-key-zul
dac25154-232d-46f2-8716-da90...
Enabled
Symmetric
SYMMETRIC_DEFAULT
Encrypt and decrypt
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Tags (0)

Edit

You can use bucket tags to track storage costs and organize buckets. Learn more

Key	Value

No tags associated with this resource.

Default encryption Info

Edit

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info
Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Encryption key ARN
arn:aws:kms:us-east-1:842597868275:key/dac25154-232d-46f2-8716-da90351ee2c3

Bucket Key
When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. Learn more
Enabled

Intelligent-Tiering Archive configurations (0)

View details | Edit | Delete | Create configuration

Enable objects stored in the Intelligent-Tiering storage class to tier-down to the Archive Access tier or the Deep Archive Access tier which are optimized for objects that will be rarely accessed for long periods of time. Learn more

Find Intelligent-Tiering Archive configurations

Name	Status	Scope	Days until transition to Archive Access t...	Days until transition to Deep Archive A...

No archive configurations
No configurations to display.

Create configuration

Server access logging

Edit

Log requests for access to your bucket. Use CloudWatch to check the health of your server access logging. Learn more

Server access logging
Disabled

hr-cur-zul Info

Objects | Metadata | Properties | Permissions | Metrics | Management | Access Points

Bucket overview

AWS Region
US East (N. Virginia) us-east-1

Amazon Resource Name (ARN)
arn:aws:s3:::hr-cur-zul

Creation date
February 12, 2025, 15:27:53 (UTC-08:00)

Bucket Versioning

Edit

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
Enabled

Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. Learn more
Disabled

Tags (0)

Edit

You can use bucket tags to track storage costs and organize buckets. Learn more

Key	Value

No tags associated with this resource.

Default encryption Info

Edit

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info
Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Encryption key ARN
arn:aws:kms:us-east-1:842597868275:key/dac25154-232d-46f2-8716-da90351ee2c3

Replication configuration successfully updated
If changes to the configuration aren't displayed, choose the refresh button. Changes apply only to new objects. To replicate existing objects with this configuration, choose **Create replication job**.

Create replication job

Replication rules Info

Replication enables automatic and asynchronous copying of objects across buckets in the same or different AWS Regions. A replication configuration is a set of rules that define what options should be applied to a group of objects during replication.

Replication configuration settings

Create replication job | Edit

Configuration settings affect all replication rules in the bucket.

Source bucket
hr-cur-zul

Source Region
US East (N. Virginia) us-east-1

IAM role
LabRole

Replication rules (1)

View details | Edit rule | Delete | Actions | Create replication rule

Use replication rules to define options you want Amazon S3 to apply during replication such as server-side encryption, replica ownership, transitioning replicas to another storage class, and more. Learn more

1

Replication rule name	Status	Destination bucket	Destination Region	Priority	Scope	Storage class	Replica owner	Replication Time Control	KMS-encrypted objects (SSE-KMS or DSSE-KMS)	Replica modification sync
hr-pd-rep-rul-zul	Enabled	s3://hr-cur-bac-zul	US East (N. Virginia) us-east-1	0	Entire bucket	Same as source	Same as source	Disabled	Replicate	Disabled

hr-pd-rep-rul-zul Info

Actions

Replication rule summary

Replication rule name
hr-pd-rep-rul-zul

Status
Enabled

Priority
0

Source bucket

Source bucket name
hr-trf-zul

Source Region
US East (N. Virginia) us-east-1

Scope
Entire bucket

Prefix
-

Tags
-

Destination

Destination bucket name
hr-trf-bac-zul

Destination Region
US East (N. Virginia) us-east-1

Storage class
Same as source

Object ownership
Same as source

Encryption

KMS-encrypted objects (SSE-KMS or DSSE-KMS)
Replicate

AWS KMS key for encrypting destination objects
arn:aws:kms:us-east-1:842597868275:key/dac25154-232d-46f2-8716-da90351ee2c3

Additional replication options

Replication Time Control
Disabled

Replication metrics and notifications
Disabled

Delete marker replication
Disabled

Replica modification sync
Disabled

hr-trf-zul Info

Objects | Metadata | Properties | Permissions | Metrics | Management | Access Points

Bucket overview

AWS Region
US East (N. Virginia) us-east-1

Amazon Resource Name (ARN)
arn:aws:s3:::hr-trf-zul

Creation date
February 4, 2025, 11:45:48 (UTC-08:00)

Bucket Versioning

Edit

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
Enabled

Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. Learn more
Disabled

Tags (0)

Edit

You can use bucket tags to track storage costs and organize buckets. Learn more

Key	Value

No tags associated with this resource.

Default encryption Info

Edit

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info
Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Encryption key ARN
arn:aws:kms:us-east-1:842597868275:key/dac25154-232d-46f2-8716-da90351ee2c3

Tags (0)

Edit

You can use bucket tags to track storage costs and organize buckets. Learn more

Key	Value

No tags associated with this resource.

Default encryption Info

Edit

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info
Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Encryption key ARN
arn:aws:kms:us-east-1:842597868275:key/dac25154-232d-46f2-8716-da90351ee2c3

Bucket Key
When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. Learn more
Enabled

Intelligent-Tiering Archive configurations (0)

View details | Edit | Delete | Create configuration

Enable objects stored in the Intelligent-Tiering storage class to tier-down to the Archive Access tier or the Deep Archive Access tier which are optimized for objects that will be rarely accessed for long periods of time. Learn more

Find Intelligent-Tiering Archive configurations

Name	Status	Scope	Days until transition to Archive Access t...	Days until transition to Deep Archive A...

No archive configurations
No configurations to display.

Create configuration

Server access logging

Edit

Log requests for access to your bucket. Use CloudWatch to check the health of your server access logging. Learn more

Server access logging
Disabled

hr-raw-zul

Objects | Metadata | Properties | Permissions | Metrics | Management | Access Points

Lifecycle configuration

To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their lifecycle. A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. Lifecycle rules run once per day.

Default minimum object size for transitions
All storage classes 128K

Lifecycle rules (2)

View details | Edit | Delete | Actions | Create lifecycle rule

Use lifecycle rules to define actions you want Amazon S3 to take during an object's lifetime such as transitioning objects to another storage class, archiving them, or deleting them after a specified period of time. Learn more

Lifecycle rule name	Status	Scope	Current version actions	Noncurrent versions actions	Expired object delete mar...	Incomplete multipart upl...
hr-emp-lst-lr-zul	Enabled	Filtered	Transition to Glacier Instant Reti	-	-	-
h4-act-lst-lr-zul	Enabled	Filtered	Transition to Glacier Flexible Ret	-	-	-

View lifecycle configuration

Replication rules (1)

View details | Edit rule | Delete | Actions | Create replication rule

Use replication rules to define options you want Amazon S3 to apply during replication such as server-side encryption, replica ownership, transitioning replicas to another storage class, and more. Learn more

Replication rule name	Status	Destination bucket	Destination Region	Priority	Scope	Storage class	Replica owner	Replication Time Control	KMS-encrypted objects (SSE-KMS or DSSE-KMS)	Replica modification sync
hr-pd-rep-rul-zul	Enabled	s3://hr-raw-bac-zul	US East (N. Virginia) us-east-1	0	Entire bucket	Same as source	Same as source	Disabled	Replicate	Disabled

View replication configuration

Inventory configurations (0)

Edit | Delete | Create job from manifest | Create inventory configuration

You can create inventory configurations on a bucket to generate a flat file list of your objects and metadata. These scheduled reports can include all objects in the bucket or be limited to a shared prefix. Learn more

Name	Status	Scope	Destination	Frequency	Last export	Format

Tags (0)

Edit

You can use bucket tags to track storage costs and organize buckets. Learn more

Key	Value

No tags associated with this resource.

Default encryption Info

Edit

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info
Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Encryption key ARN
arn:aws:kms:us-east-1:842597868275:key/dac25154-232d-46f2-8716-da90351ee2c3

Bucket Key
When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. Learn more
Enabled

Intelligent-Tiering Archive configurations (0)

View details | Edit | Delete | Create configuration

Enable objects stored in the Intelligent-Tiering storage class to tier-down to the Archive Access tier or the Deep Archive Access tier which are optimized for objects that will be rarely accessed for long periods of time. Learn more

Find Intelligent-Tiering Archive configurations

Name	Status	Scope	Days until transition to Archive Access t...	Days until transition to Deep Archive A...

No archive configurations
No configurations to display.

Create configuration

Server access logging

Edit

Log requests for access to your bucket. Use CloudWatch to check the health of your server access logging. Learn more

Server access logging
Disabled

hr-raw-zul Info

Objects | Metadata | Properties | Permissions | Metrics | Management | Access Points

Bucket overview

AWS Region
US East (N. Virginia) us-east-1

Amazon Resource Name (ARN)
arn:aws:s3:::hr-raw-zul

Creation date
January 28, 2025, 00:58:06 (UTC-08:00)

Bucket Versioning

Edit

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
Enabled

Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. Learn more
Disabled

Tags (0)

Edit

You can use bucket tags to track storage costs and organize buckets. Learn more

Key	Value

No tags associated with this resource.

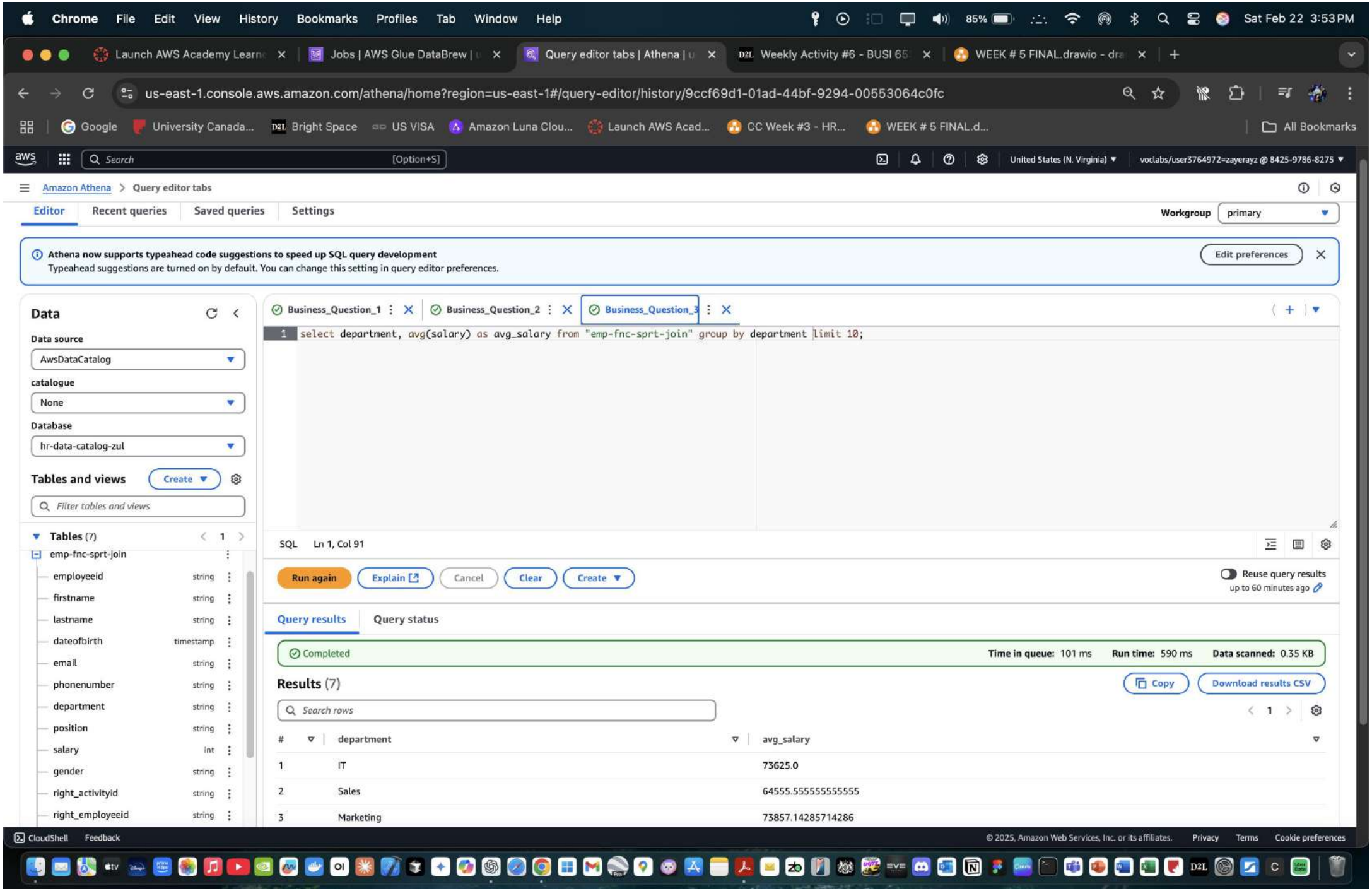
Default encryption Info

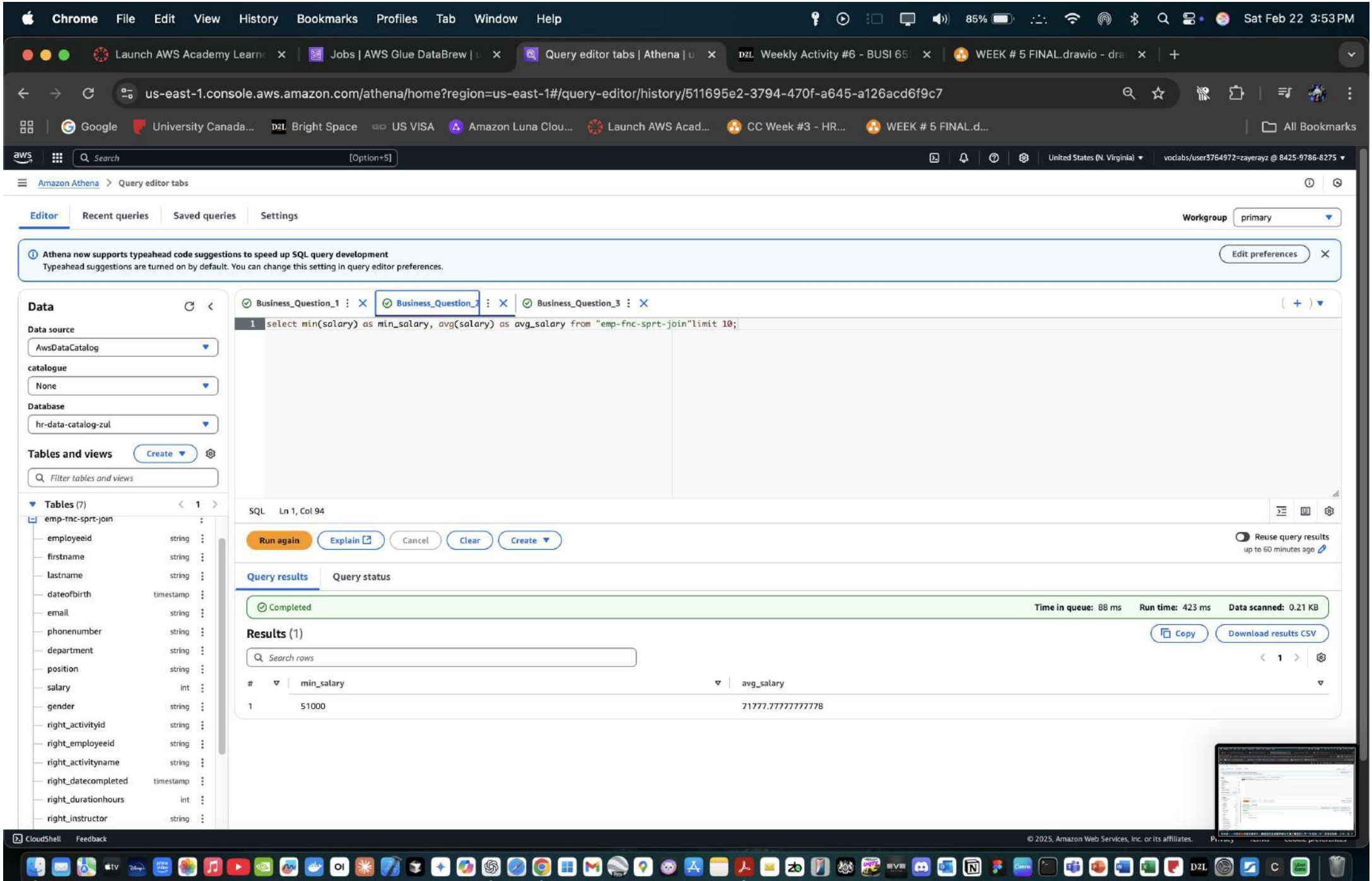
Edit

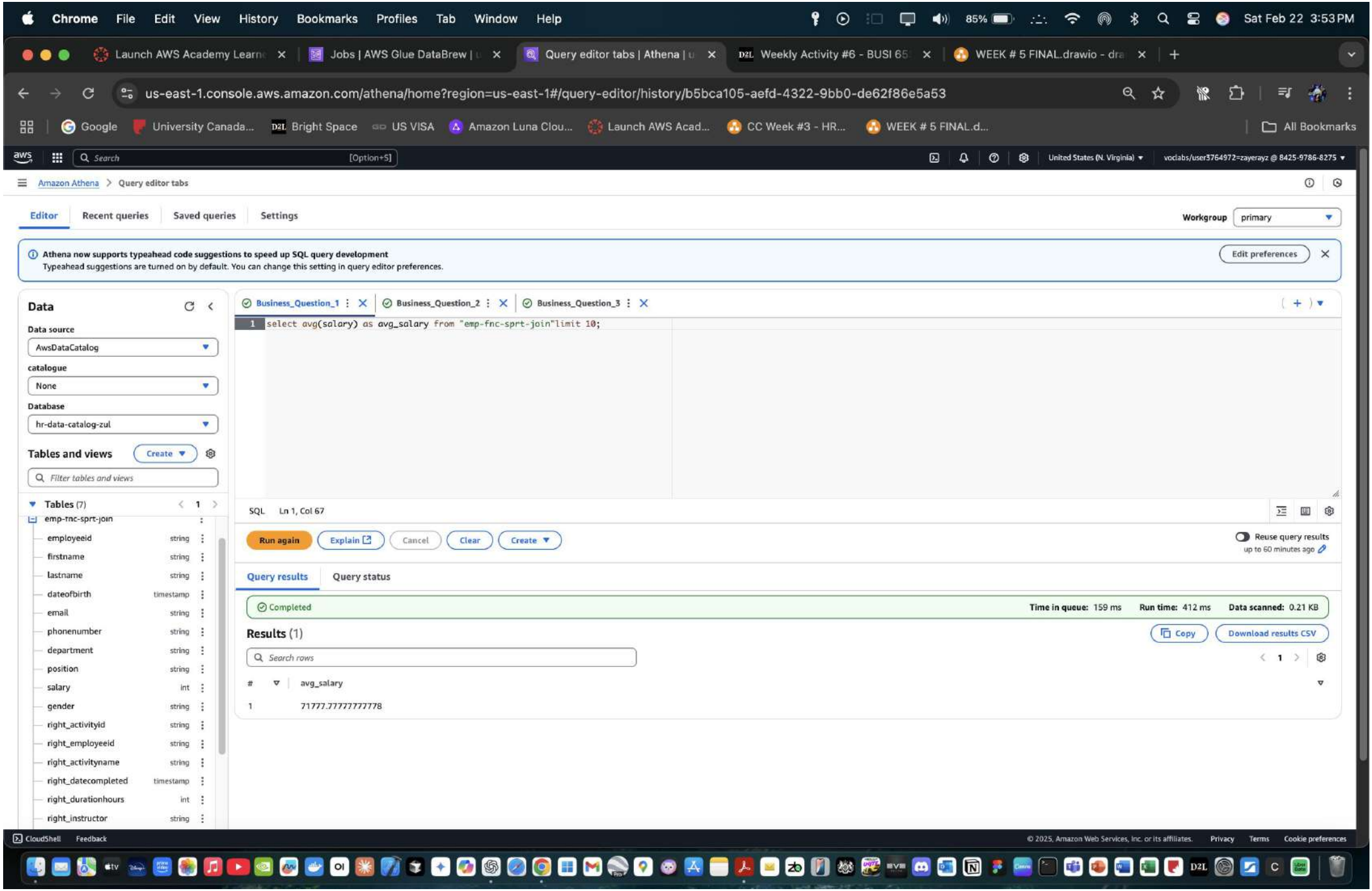
Server-side encryption is automatically applied to new objects stored in this bucket.

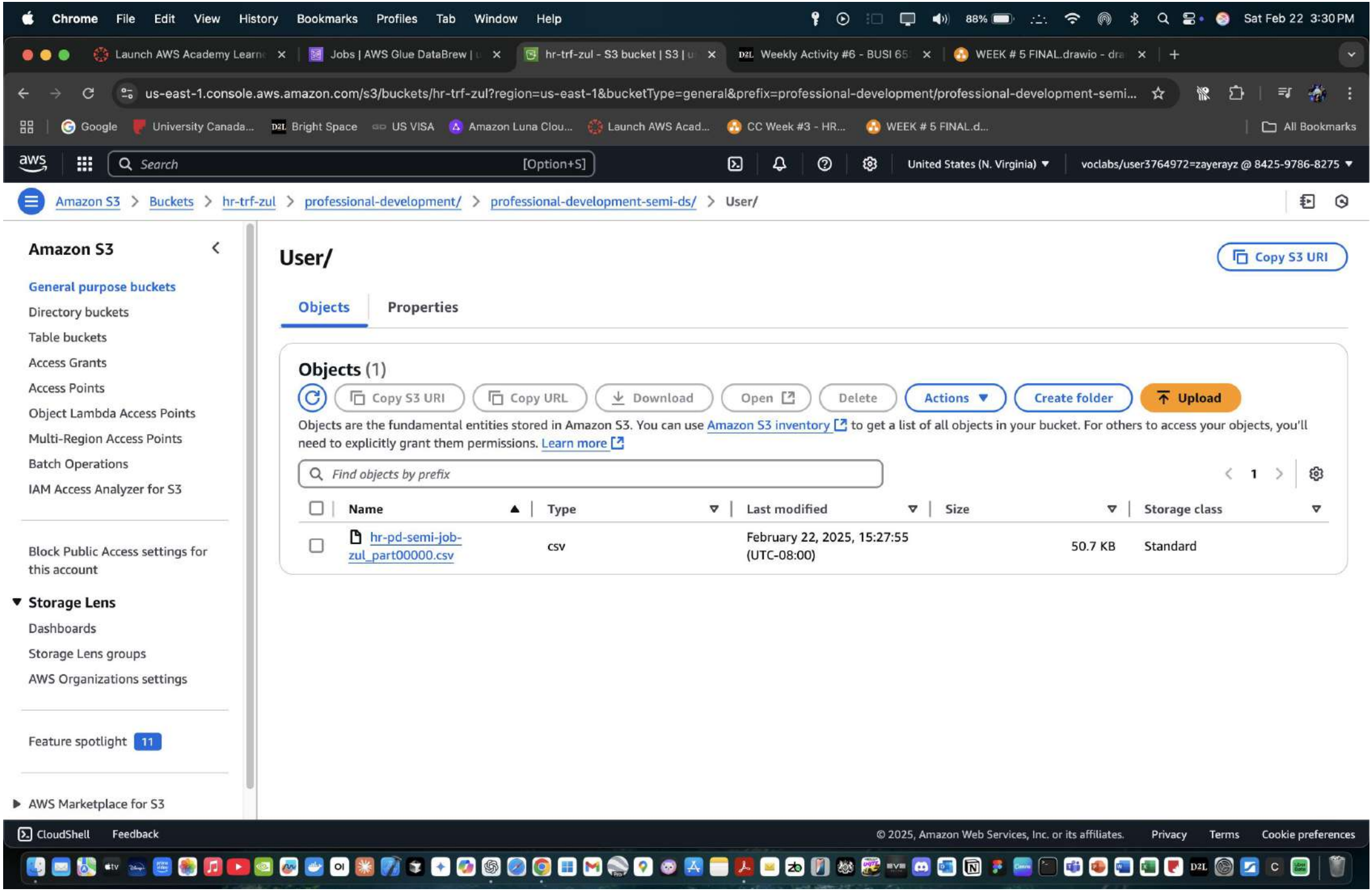
Encryption type Info
Server-side encryption with AWS Key Management Service keys (SSE-KMS)

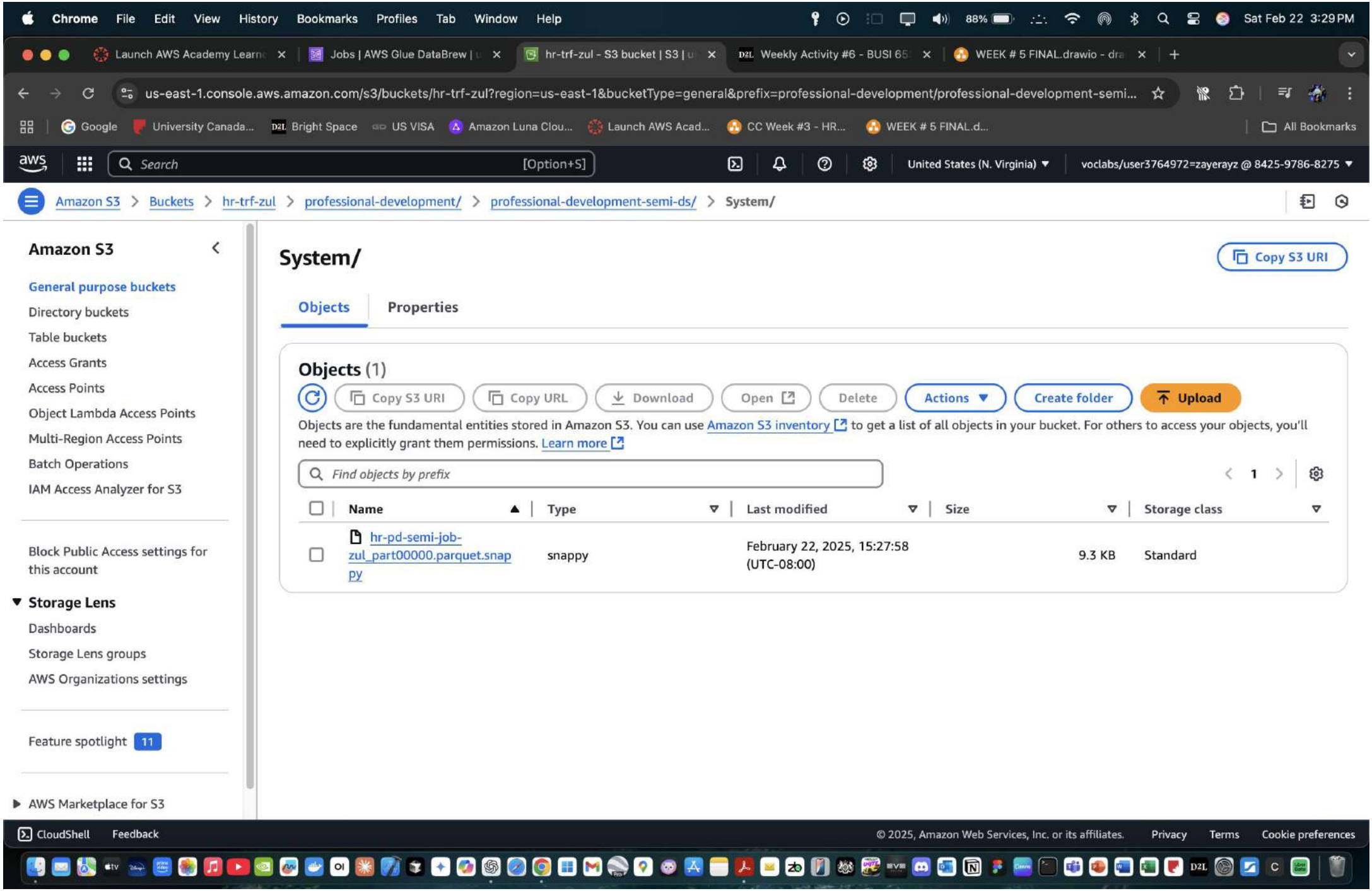
Encryption key ARN
arn:aws:kms:us-east-1:842597868275:key/dac25154-232d-46f2-8716-da90351ee2c3

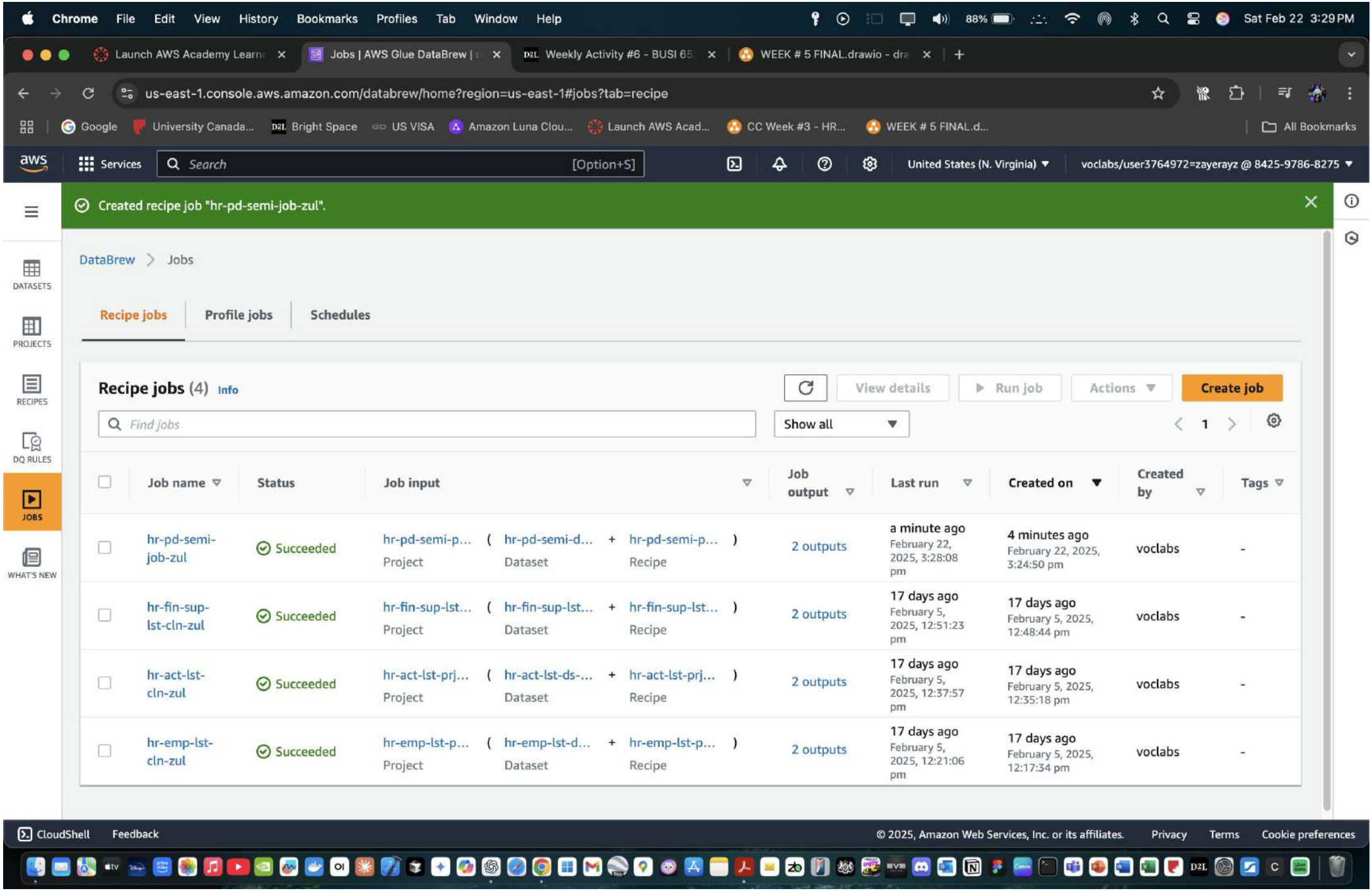

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sat Feb 22 3:53 PM
Launch AWS Academy Learn
Jobs | AWS Glue DataBrew |
Query editor tabs | Athena |
Weekly Activity #6 - BUSI 65
WEEK # 5 FINAL.drawio - dra
us-east-1.console.aws.amazon.com/athena/home?region=us-east-1#/query-editor/history/9ccf69d1-01ad-44bf-9294-00553064c0fc
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
WEEK # 5 FINAL.d...
All Bookmarks
Search [Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon Athena > Query editor tabs
Editor
Recent queries
Saved queries
Settings
Workgroup primary
Athena now supports typeahead code suggestions to speed up SQL query development
Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.
Edit preferences
Data
Data source
AwsDataCatalog
catalogue
None
Database
hr-data-catalog-zul
Tables and views
Create
Filter tables and views
Tables (7)
emp-fnc-sprt-join
employeeid string
firstname string
lastname string
dateofbirth timestamp
email string
phonenumber string
department string
position string
salary int
gender string
right_activityid string
right_employeeid string
Business_Question_1
Business_Question_2
Business_Question_3
1 select department, avg(salary) as avg_salary from "emp-fnc-sprt-join" group by department limit 10;
SQL Ln 1, Col 91
Run again
Explain
Cancel
Clear
Create
Reuse query results
up to 60 minutes ago
Query results
Query status
Completed
Time in queue: 101 ms
Run time: 590 ms
Data scanned: 0.35 KB
Results (7)
Copy
Download results CSV
Search rows
department avg_salary
1 IT 73625.0
2 Sales 64555.555555555555
3 Marketing 73857.14285714286
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

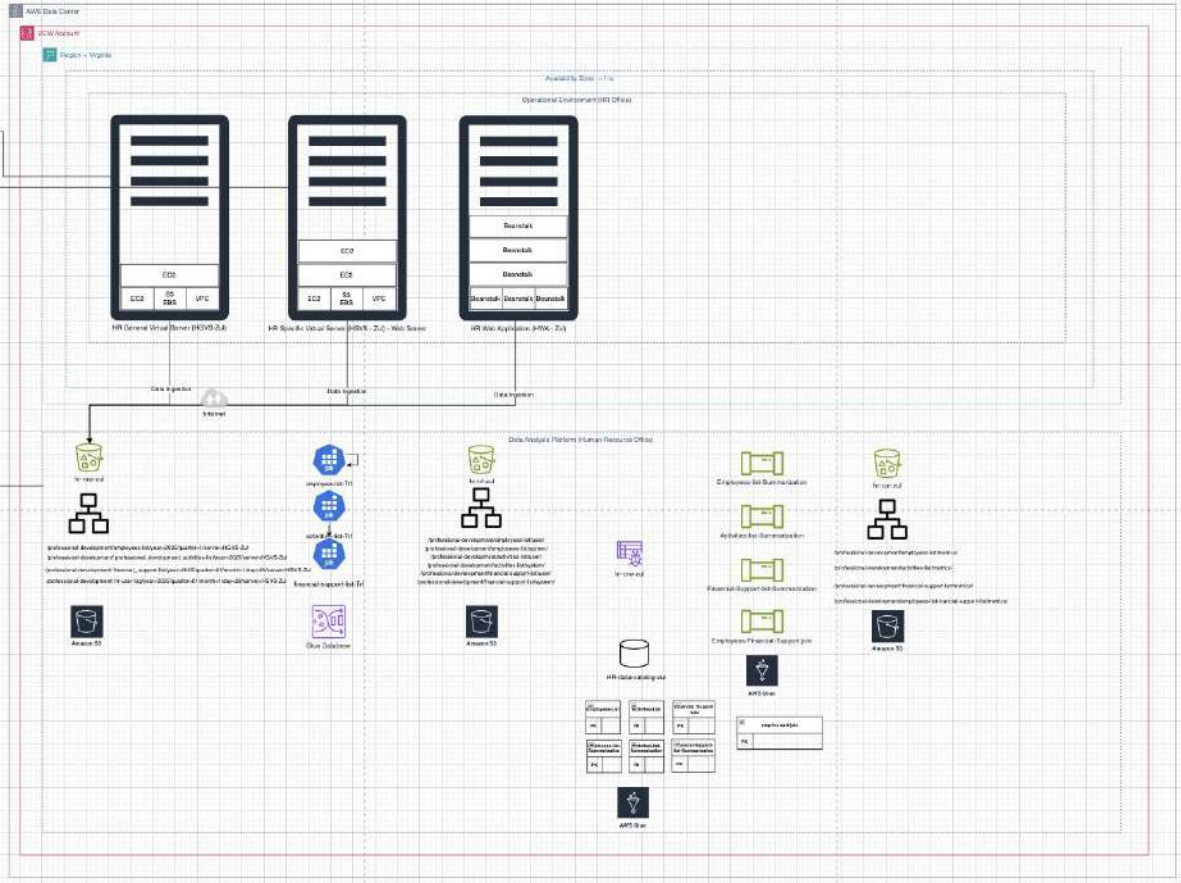

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sat Feb 22 3:53 PM
Launch AWS Academy Learn
Jobs | AWS Glue DataBrew |
Query editor tabs | Athena |
Weekly Activity #6 - BUSI 65
WEEK # 5 FINAL.drawio - dra
us-east-1.console.aws.amazon.com/athena/home?region=us-east-1#/query-editor/history/511695e2-3794-470f-a645-a126acd6f9c7
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
WEEK # 5 FINAL.d...
All Bookmarks
Search [Option+S]
United States (N. Virginia)
Amazon Athena > Query editor tabs
Editor
Recent queries
Saved queries
Settings
Workgroup primary
Athena now supports typeahead code suggestions to speed up SQL query development
Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.
Edit preferences
Data
Data source
AwsDataCatalog
catalogue
None
Database
hr-data-catalog-zul
Tables and views
Create
Filter tables and views
Tables (7)
emp-fnc-sprt-join
employeeid string
firstname string
lastname string
dateofbirth timestamp
email string
phonenumber string
department string
position string
salary int
gender string
right_activityid string
right_employeeid string
right_activityname string
right_datecompleted timestamp
right_durationhours int
right_instructor string
Business_Question_1
Business_Question_2
Business_Question_3
1 select min(salary) as min_salary, avg(salary) as avg_salary from "emp-fnc-sprt-join"limit 10;
SQL Ln 1, Col 94
Run again
Explain
Cancel
Clear
Create
Reuse query results
up to 60 minutes ago
Query results
Query status
Completed
Time in queue: 88 ms
Run time: 423 ms
Data scanned: 0.21 KB
Results (1)
Copy
Download results CSV
Search rows
min_salary avg_salary
1 51000 71777.77777777778
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.

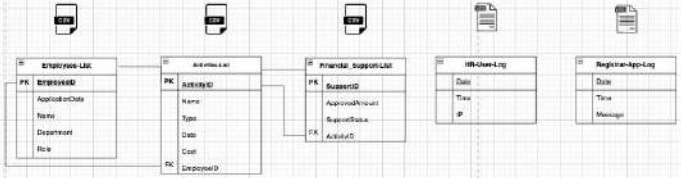

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sat Feb 22 3:53 PM
Launch AWS Academy Learn
Jobs | AWS Glue DataBrew |
Query editor tabs | Athena |
Weekly Activity #6 - BUSI 65
WEEK # 5 FINAL.drawio - dra
us-east-1.console.aws.amazon.com/athena/home?region=us-east-1#/query-editor/history/b5bca105-aefd-4322-9bb0-de62f86e5a53
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
WEEK # 5 FINAL.d...
All Bookmarks
Search [Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon Athena > Query editor tabs
Editor
Recent queries
Saved queries
Settings
Workgroup primary
Athena now supports typeahead code suggestions to speed up SQL query development
Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.
Edit preferences
Data
Data source
AwsDataCatalog
catalogue
None
Database
hr-data-catalog-zul
Tables and views
Create
Filter tables and views
Tables (7)
emp-fnc-sprt-join
employeeid string
firstname string
lastname string
dateofbirth timestamp
email string
phonenumber string
department string
position string
salary int
gender string
right_activityid string
right_employeeid string
right_activityname string
right_datecompleted timestamp
right_durationhours int
right_instructor string
Business_Question_1
Business_Question_2
Business_Question_3
1 select avg(salary) as avg_salary from "emp-fnc-sprt-join"limit 10;
SQL Ln 1, Col 67
Run again
Explain
Cancel
Clear
Create
Reuse query results
up to 60 minutes ago
Query results
Query status
Completed
Time in queue: 159 ms
Run time: 412 ms
Data scanned: 0.21 KB
Results (1)
Copy
Download results CSV
Search rows
avg_salary
1 71777.77777777778
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sat Feb 22 3:30 PM
Launch AWS Academy Learn
Jobs | AWS Glue DataBrew |
hr-trf-zul - S3 bucket | S3 |
Weekly Activity #6 - BUSI 65
WEEK # 5 FINAL.drawio - dra
us-east-1.console.aws.amazon.com/s3/buckets/hr-trf-zul?region=us-east-1&bucketType=general&prefix=professional-development/professional-development-semi...
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
WEEK # 5 FINAL.d...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > Buckets > hr-trf-zul > professional-development/ > professional-development-semi-ds/ > User/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
User/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
hr-pd-semi-job-zul_part00000.csv
csv
February 22, 2025, 15:27:55 (UTC-08:00)
50.7 KB
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sat Feb 22 3:29 PM
Launch AWS Academy Learn
Jobs | AWS Glue DataBrew |
hr-trf-zul - S3 bucket | S3 |
Weekly Activity #6 - BUSI 65
WEEK # 5 FINAL.drawio - dra
us-east-1.console.aws.amazon.com/s3/buckets/hr-trf-zul?region=us-east-1&bucketType=general&prefix=professional-development/professional-development-semi...
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
WEEK # 5 FINAL.d...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > Buckets > hr-trf-zul > professional-development/ > professional-development-semi-ds/ > System/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
System/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
hr-pd-semi-job-zul_part00000.parquet.snappy
snappy
February 22, 2025, 15:27:58 (UTC-08:00)
9.3 KB
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sat Feb 22 3:29 PM
Launch AWS Academy Learn
Jobs | AWS Glue DataBrew |
Weekly Activity #6 - BUSI 65
WEEK # 5 FINAL.drawio - dra
us-east-1.console.aws.amazon.com/databrew/home?region=us-east-1#jobs?tab=recipe
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
WEEK # 5 FINAL.d...
All Bookmarks
aws
Services
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Created recipe job "hr-pd-semi-job-zul".
DataBrew
Jobs
DATASETS
PROJECTS
RECIPES
DQ RULES
JOBS
WHAT'S NEW
Recipe jobs
Profile jobs
Schedules
Recipe jobs (4) Info
View details
Run job
Actions
Create job
Find jobs
Show all
1
Job name
Status
Job input
Job output
Last run
Created on
Created by
Tags
hr-pd-semi-job-zul
Succeeded
hr-pd-semi-p... (hr-pd-semi-d... + hr-pd-semi-p...)
Project Dataset Recipe
2 outputs
a minute ago
February 22, 2025, 3:28:08 pm
4 minutes ago
February 22, 2025, 3:24:50 pm
voclabs
-
hr-fin-sup-lst-cln-zul
Succeeded
hr-fin-sup-lst... (hr-fin-sup-lst... + hr-fin-sup-lst...)
Project Dataset Recipe
2 outputs
17 days ago
February 5, 2025, 12:51:23 pm
17 days ago
February 5, 2025, 12:48:44 pm
voclabs
-
hr-act-lst-cln-zul
Succeeded
hr-act-lst-prj... (hr-act-lst-ds-... + hr-act-lst-prj...)
Project Dataset Recipe
2 outputs
17 days ago
February 5, 2025, 12:37:57 pm
17 days ago
February 5, 2025, 12:35:18 pm
voclabs
-
hr-emp-lst-cln-zul
Succeeded
hr-emp-lst-p... (hr-emp-lst-d... + hr-emp-lst-p...)
Project Dataset Recipe
2 outputs
17 days ago
February 5, 2025, 12:21:06 pm
17 days ago
February 5, 2025, 12:17:34 pm
voclabs
-
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



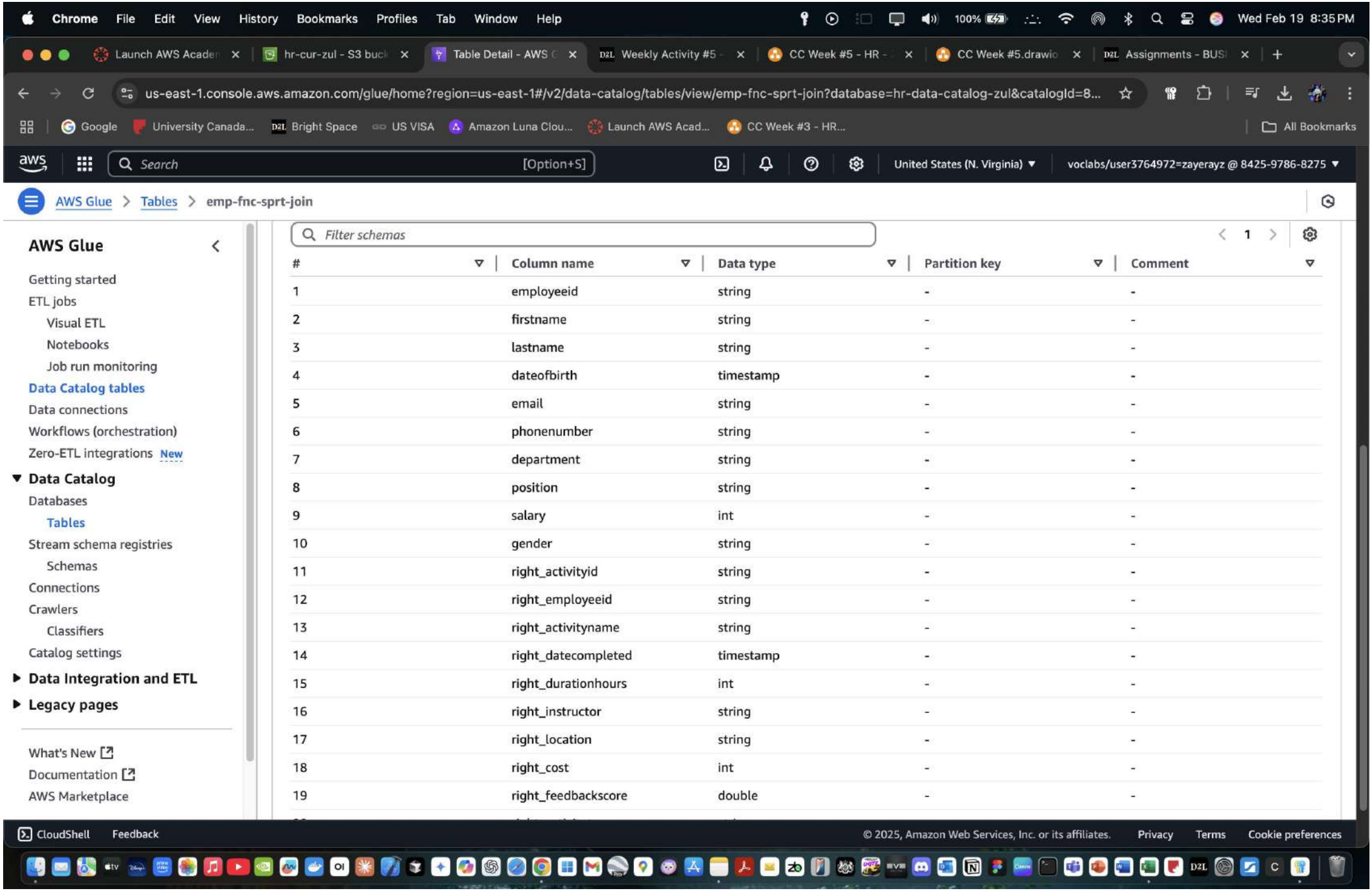


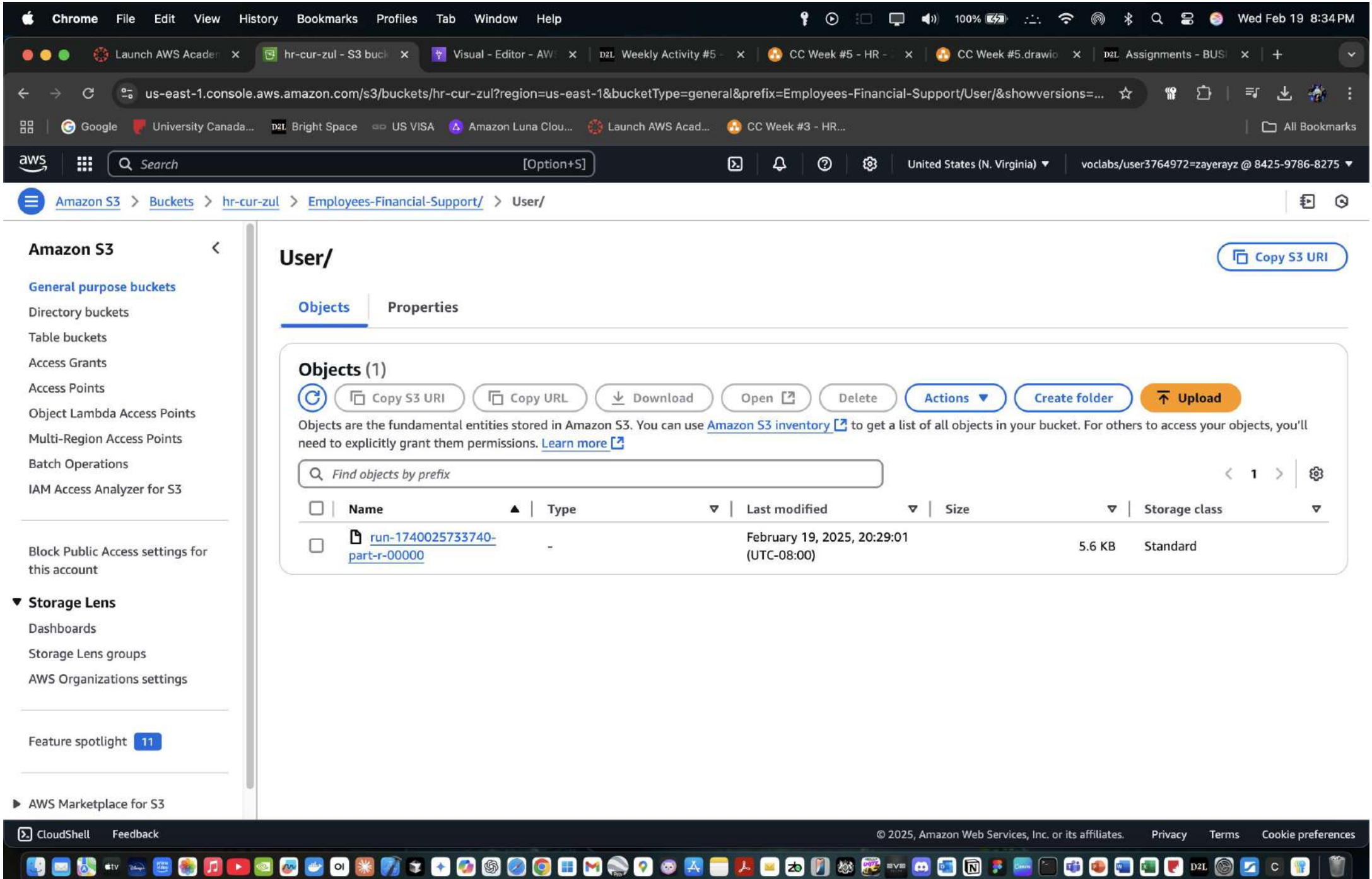
Data Enriching

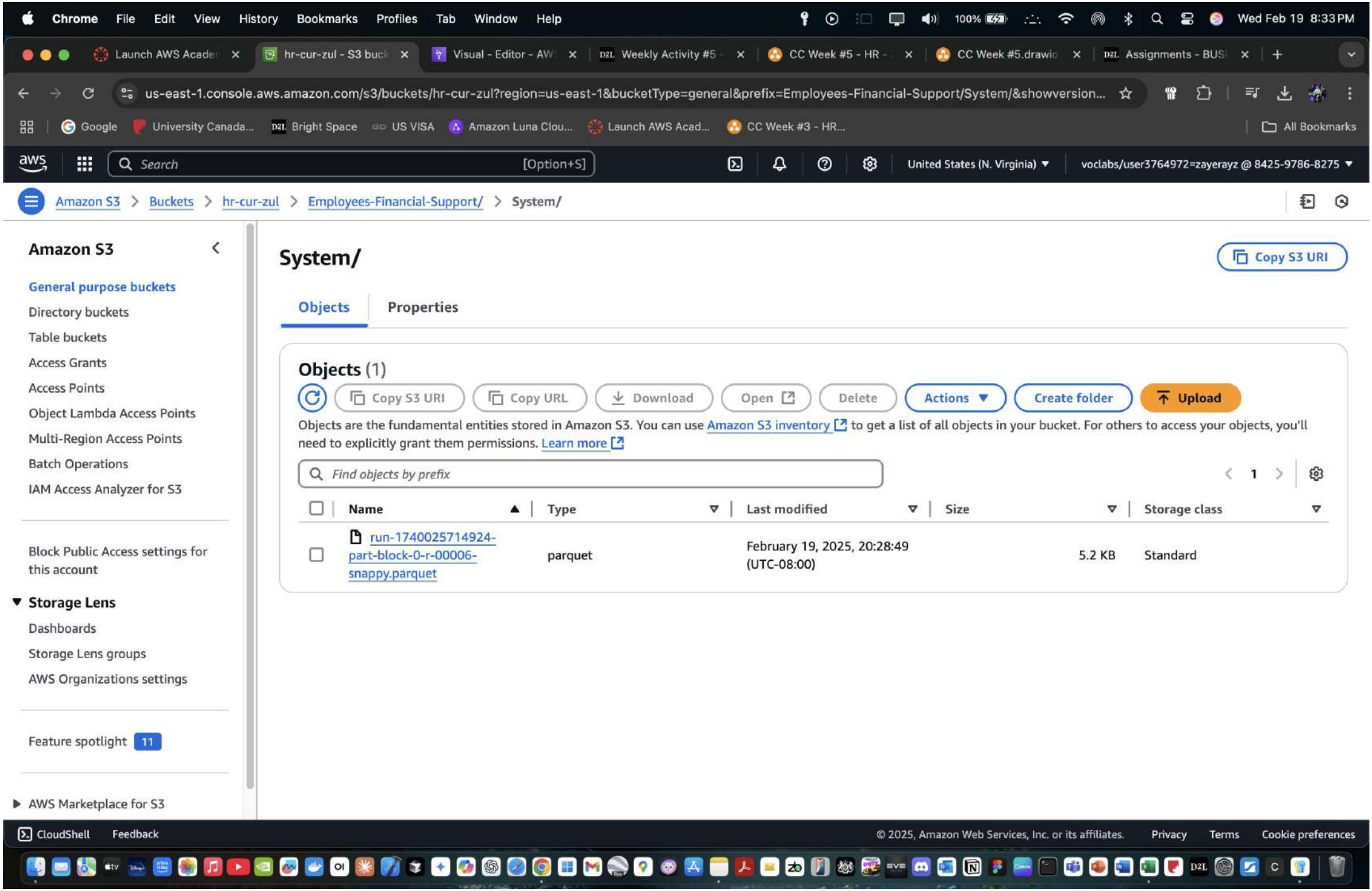
Table	Join
Employees-List Financial_support-List	Left Join

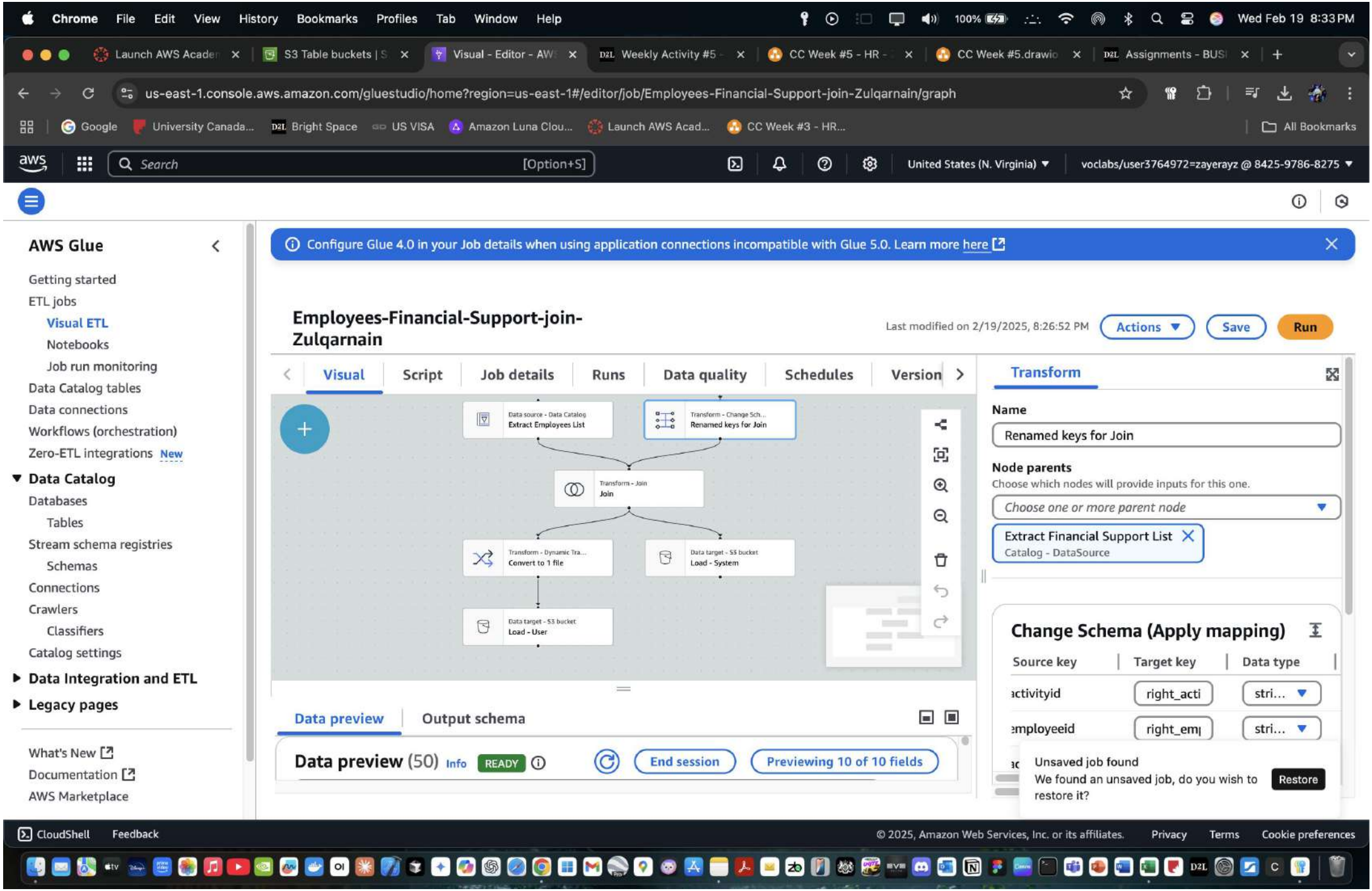
Data Summarization Cost Analysis

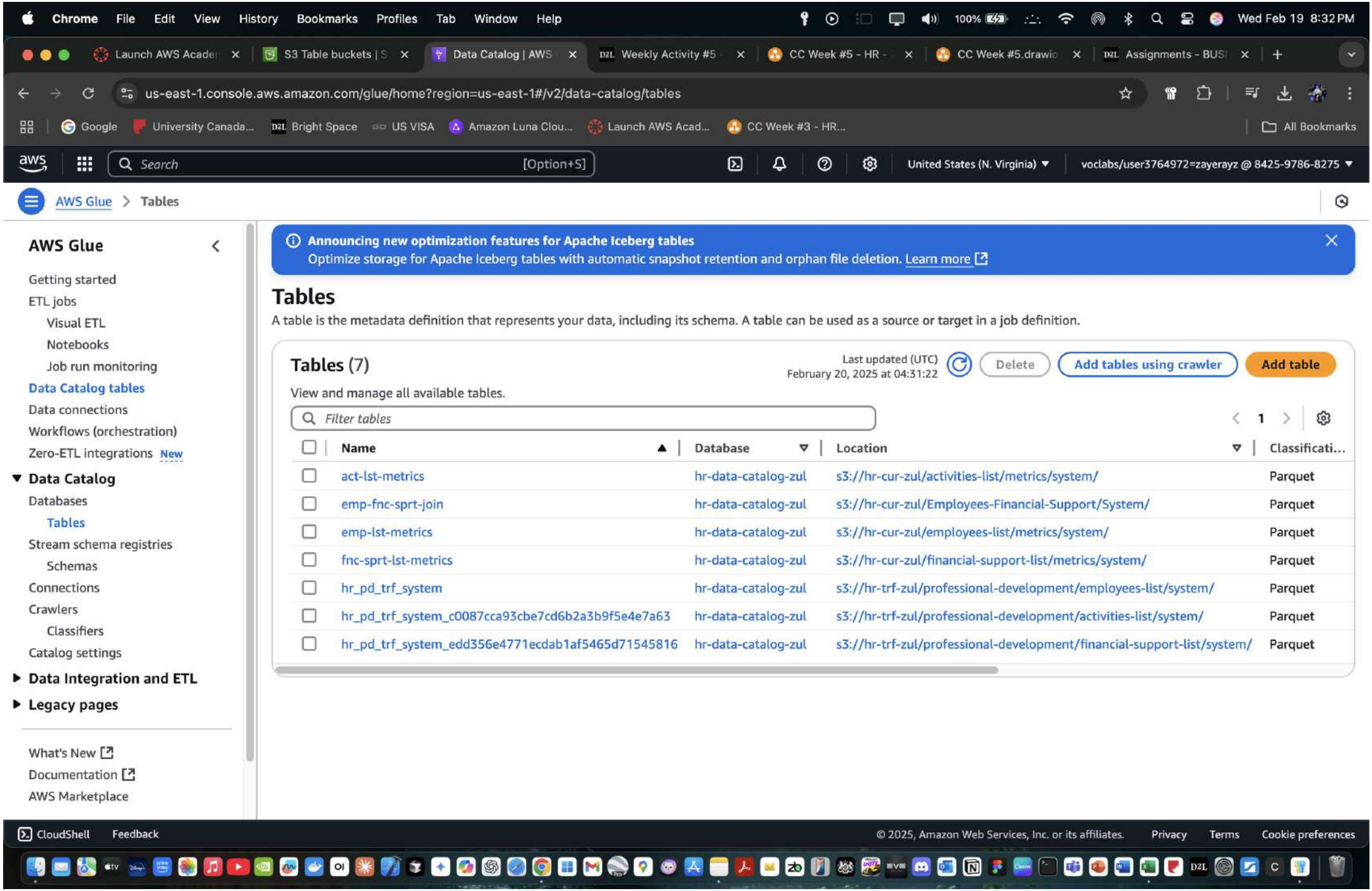
	Glue interactive sessions (#)	Glue interactive sessions Nodes	Glue Job Nodes	Glue Job Duration
Employees-List	3	2	10	0:14 mins
Activities-List	3	2	10	0:33 mins
Financial_Support-List	2	2	10	1:53
HR-User-Log				

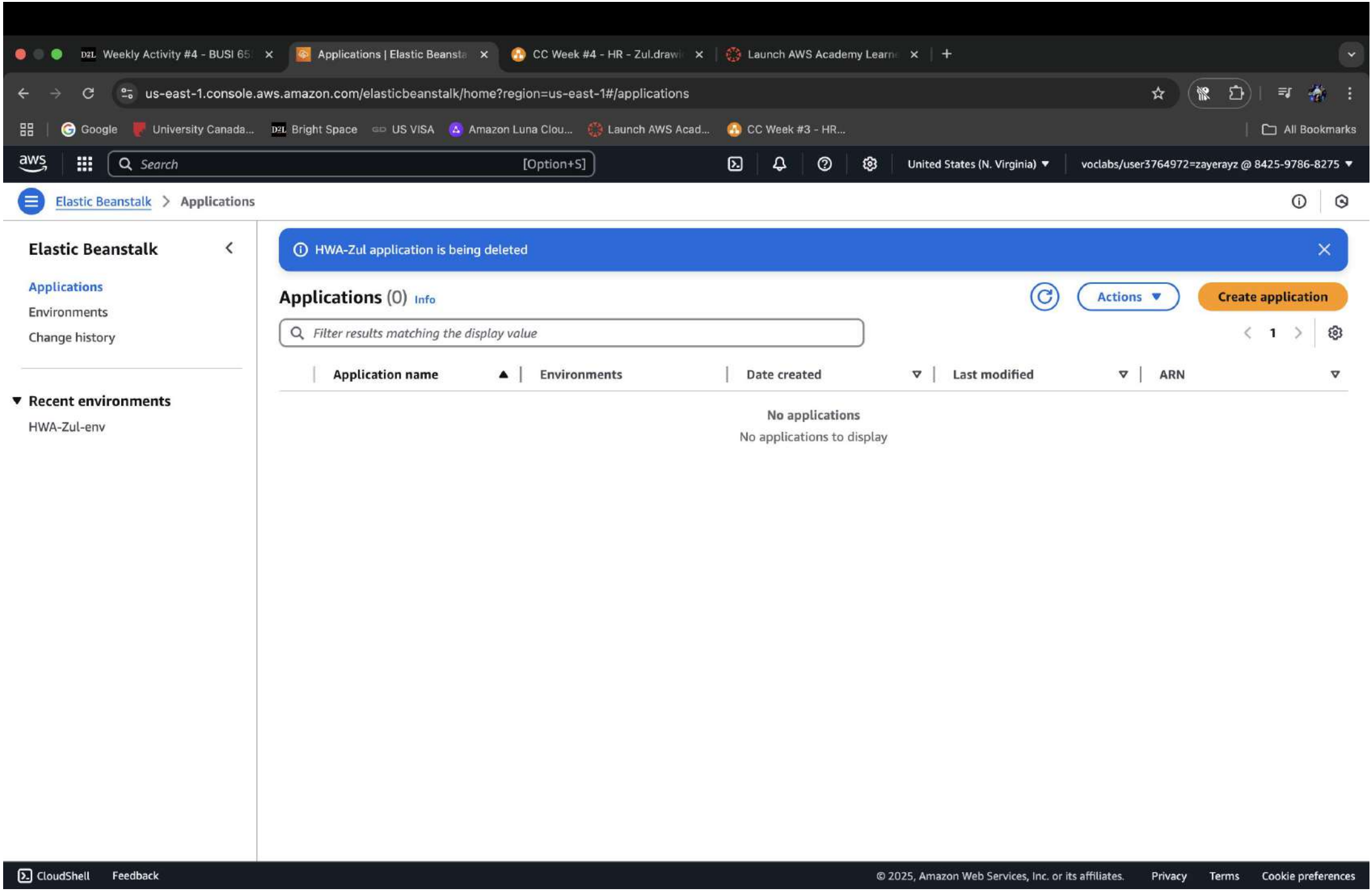

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Feb 19 8:35 PM
Launch AWS Acade
hr-cur-zul - S3 buck
Table Detail - AWS
Weekly Activity #5
CC Week #5 - HR -
CC Week #5.drawio
Assignments - BUS
us-east-1.console.aws.amazon.com/glue/home?region=us-east-1#/v2/data-catalog/tables/view/emp-fnc-sprt-join?database=hr-data-catalog-zul&catalogId=8...
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
AWS Glue > Tables > emp-fnc-sprt-join
AWS Glue
Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Zero-ETL integrations New
Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings
Data Integration and ETL
Legacy pages
What's New
Documentation
AWS Marketplace
Filter schemas
1
| Column name | Data type | Partition key | Comment
1 employeeid string - -
2 firstname string - -
3 lastname string - -
4 dateofbirth timestamp - -
5 email string - -
6 phonenumber string - -
7 department string - -
8 position string - -
9 salary int - -
10 gender string - -
11 right_activityid string - -
12 right_employeeid string - -
13 right_activityname string - -
14 right_datecompleted timestamp - -
15 right_durationhours int - -
16 right_instructor string - -
17 right_location string - -
18 right_cost int - -
19 right_feedbackscore double - -
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

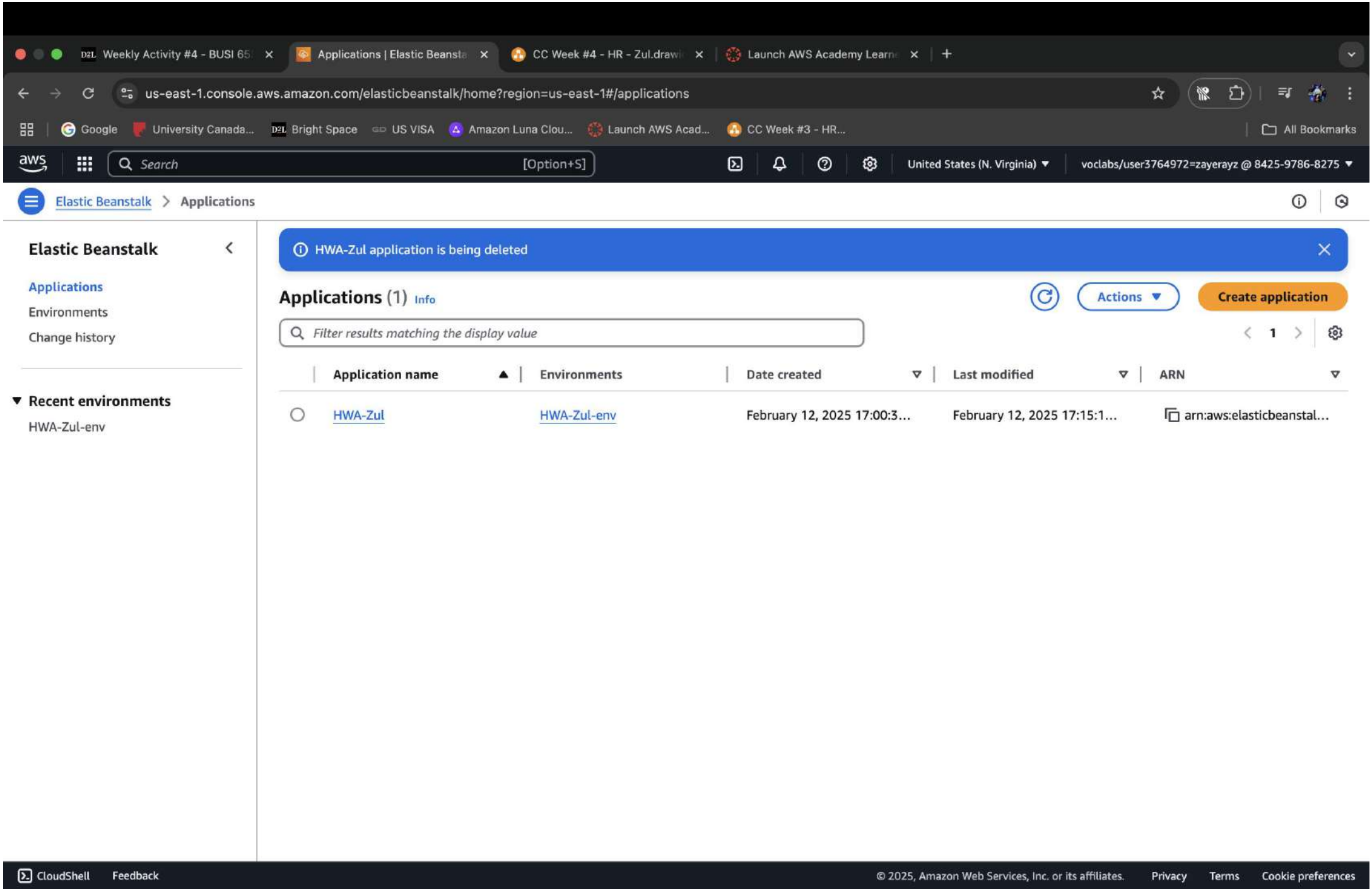

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Feb 19 8:34 PM
us-east-1.console.aws.amazon.com/s3/buckets/hr-cur-zul?region=us-east-1&bucketType=general&prefix=Employees-Financial-Support/User/&showversions=...
Search [Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > Buckets > hr-cur-zul > Employees-Financial-Support/ > User/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
User/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
run-1740025733740-part-r-00000
-
February 19, 2025, 20:29:01 (UTC-08:00)
5.6 KB
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

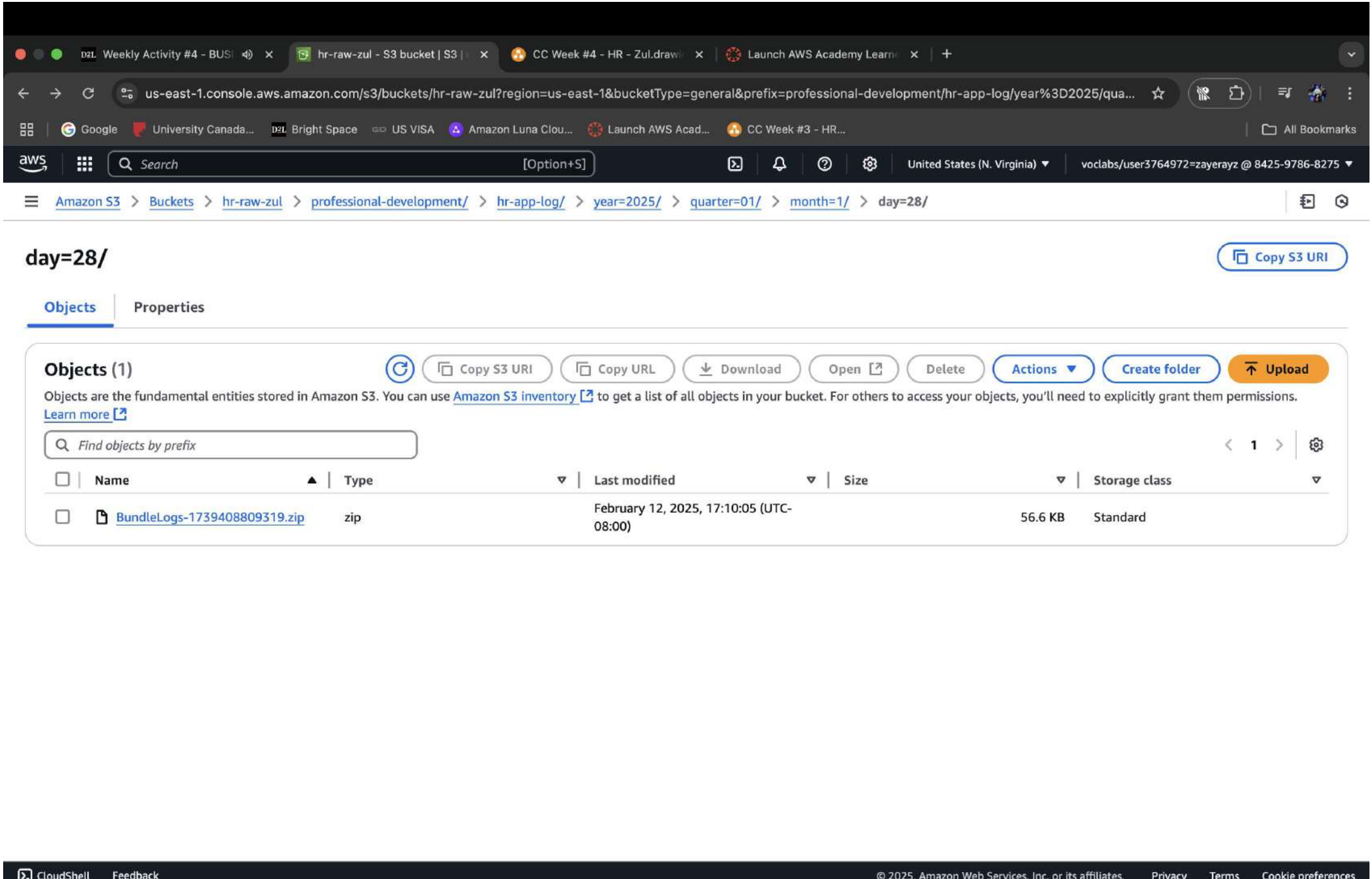

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Feb 19 8:33 PM
us-east-1.console.aws.amazon.com/s3/buckets/hr-cur-zul?region=us-east-1&bucketType=general&prefix=Employees-Financial-Support/System/&showversion...
Search [Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > Buckets > hr-cur-zul > Employees-Financial-Support/ > System/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
System/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
run-1740025714924-part-block-0-r-00006-snappy.parquet
parquet
February 19, 2025, 20:28:49 (UTC-08:00)
5.2 KB
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

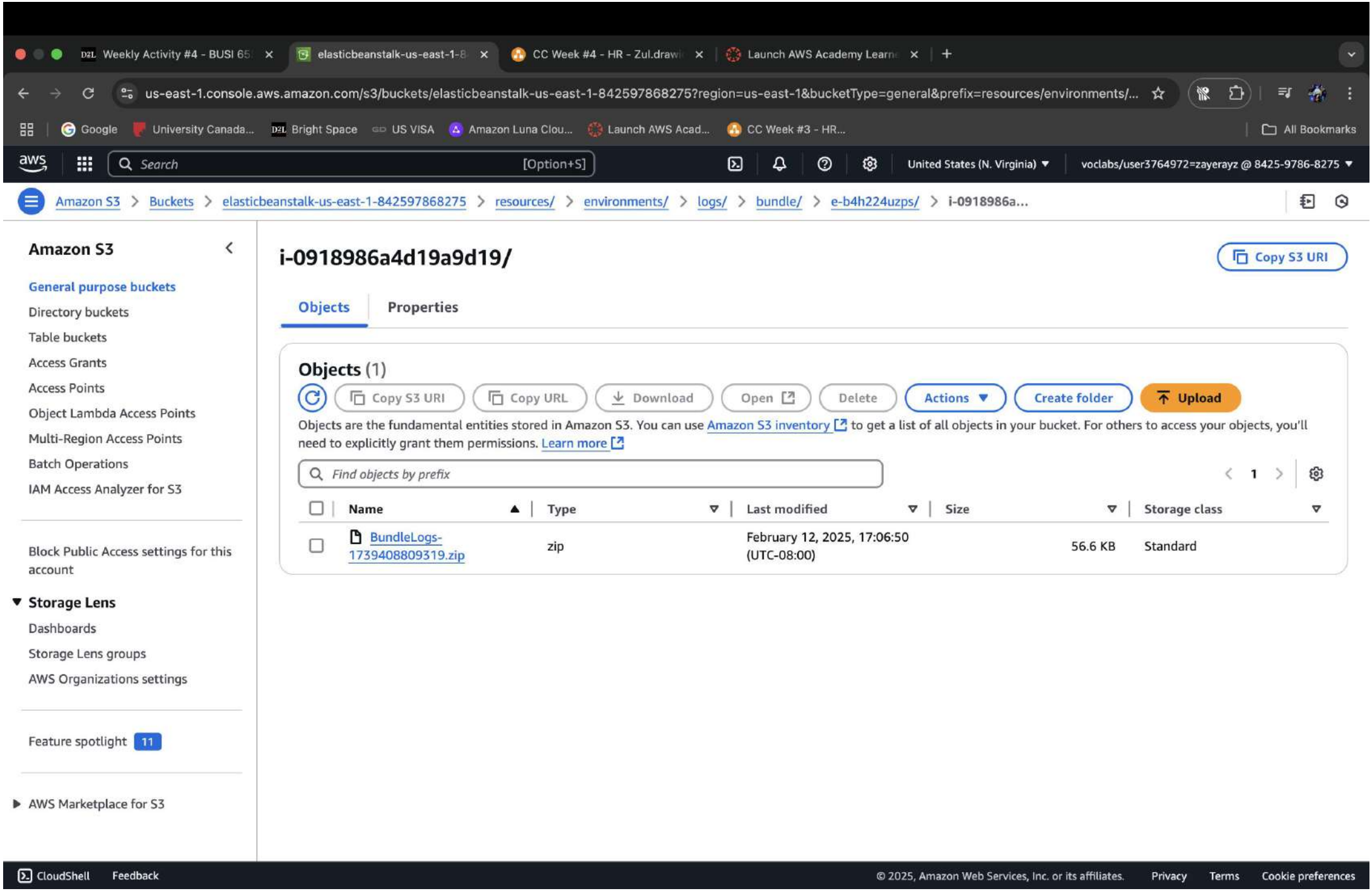

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Feb 19 8:33 PM
us-east-1.console.aws.amazon.com/gluestudio/home?region=us-east-1#/editor/job/Employees-Financial-Support-join-Zulqarnain/graph
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
AWS Glue
Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Zero-ETL integrations New
Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings
Data Integration and ETL
Legacy pages
What's New
Documentation
AWS Marketplace
Configure Glue 4.0 in your Job details when using application connections incompatible with Glue 5.0. Learn more here
Employees-Financial-Support-join-Zulqarnain
Last modified on 2/19/2025, 8:26:52 PM
Actions
Save
Run
Visual
Script
Job details
Runs
Data quality
Schedules
Version
Data source - Data Catalog
Extract Employees List
Transform - Change Sch...
Renamed keys for Join
Transform - Join
Join
Transform - Dynamic Tra...
Convert to 1 file
Data target - S3 bucket
Load - System
Data target - S3 bucket
Load - User
Transform
Name
Renamed keys for Join
Node parents
Choose which nodes will provide inputs for this one.
Choose one or more parent node
Extract Financial Support List
Catalog - DataSource
Change Schema (Apply mapping)
Source key
Target key
Data type
activityid
right_acti
stri...
employeeid
right_emp
stri...
Data preview
Output schema
Data preview (50) Info READY
End session
Previewing 10 of 10 fields
Unsaved job found
We found an unsaved job, do you wish to restore it?
Restore
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

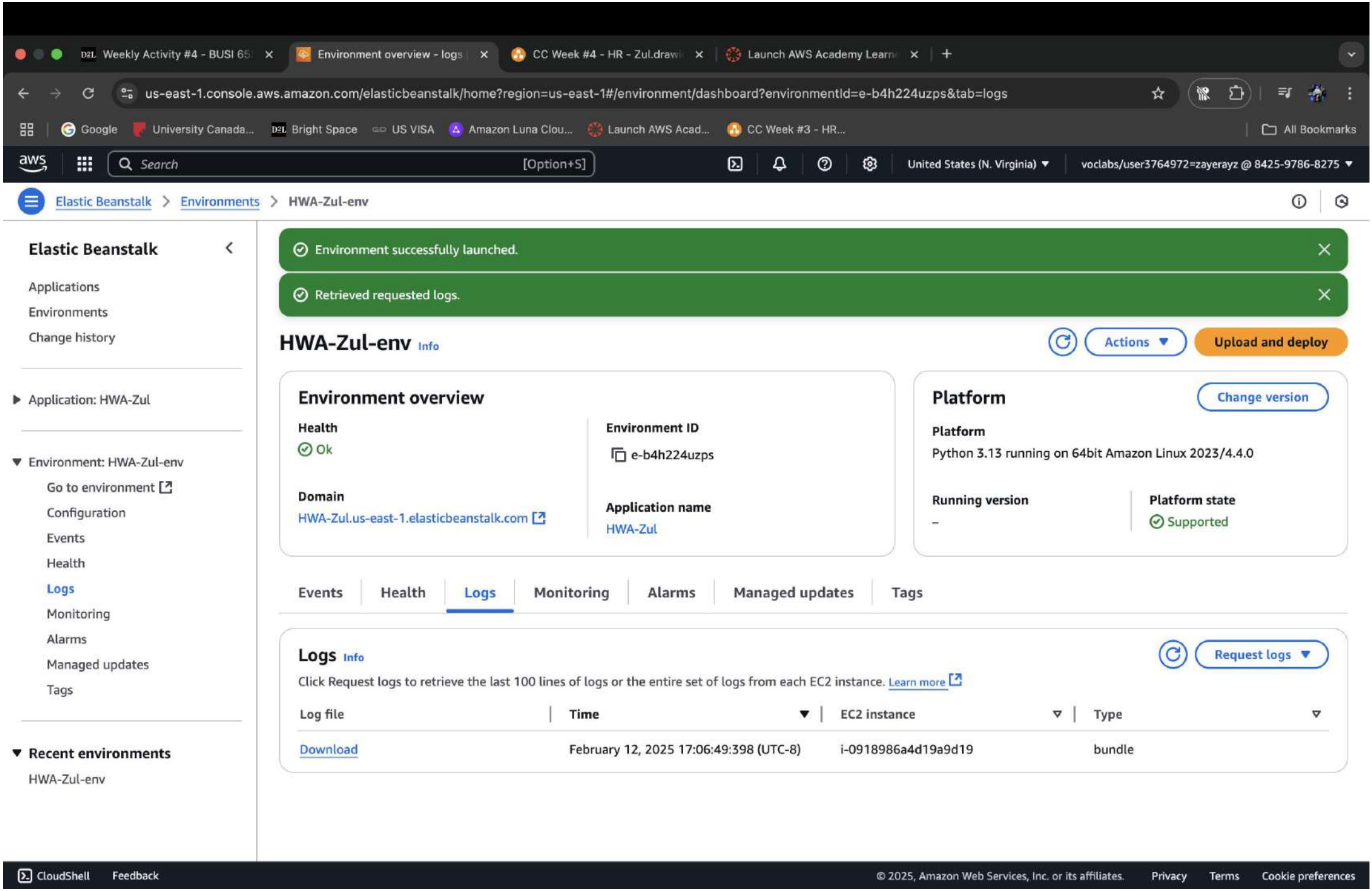

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Feb 19 8:32 PM
Launch AWS Acade
S3 Table buckets | S
Data Catalog | AWS
Weekly Activity #5
CC Week #5 - HR -
CC Week #5.drawio
Assignments - BUS
us-east-1.console.aws.amazon.com/glue/home?region=us-east-1#/v2/data-catalog/tables
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
AWS Glue > Tables
AWS Glue
Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Zero-ETL integrations New
Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings
Data Integration and ETL
Legacy pages
What's New
Documentation
AWS Marketplace
Announcing new optimization features for Apache Iceberg tables
Optimize storage for Apache Iceberg tables with automatic snapshot retention and orphan file deletion. Learn more
Tables
A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.
Tables (7)
Last updated (UTC)
February 20, 2025 at 04:31:22
Delete
Add tables using crawler
Add table
View and manage all available tables.
Filter tables
1
Name
Database
Location
Classificati...
act-lst-metrics hr-data-catalog-zul s3://hr-cur-zul/activities-list/metrics/system/ Parquet
emp-fnc-sprt-join hr-data-catalog-zul s3://hr-cur-zul/Employees-Financial-Support/System/ Parquet
emp-lst-metrics hr-data-catalog-zul s3://hr-cur-zul/employees-list/metrics/system/ Parquet
fnc-sprt-lst-metrics hr-data-catalog-zul s3://hr-cur-zul/financial-support-list/metrics/system/ Parquet
hr_pd_trf_system hr-data-catalog-zul s3://hr-trf-zul/professional-development/employees-list/system/ Parquet
hr_pd_trf_system_c0087cca93cbe7cd6b2a3b9f5e4e7a63 hr-data-catalog-zul s3://hr-trf-zul/professional-development/activities-list/system/ Parquet
hr_pd_trf_system_edd356e4771ecdab1af5465d71545816 hr-data-catalog-zul s3://hr-trf-zul/professional-development/financial-support-list/system/ Parquet
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

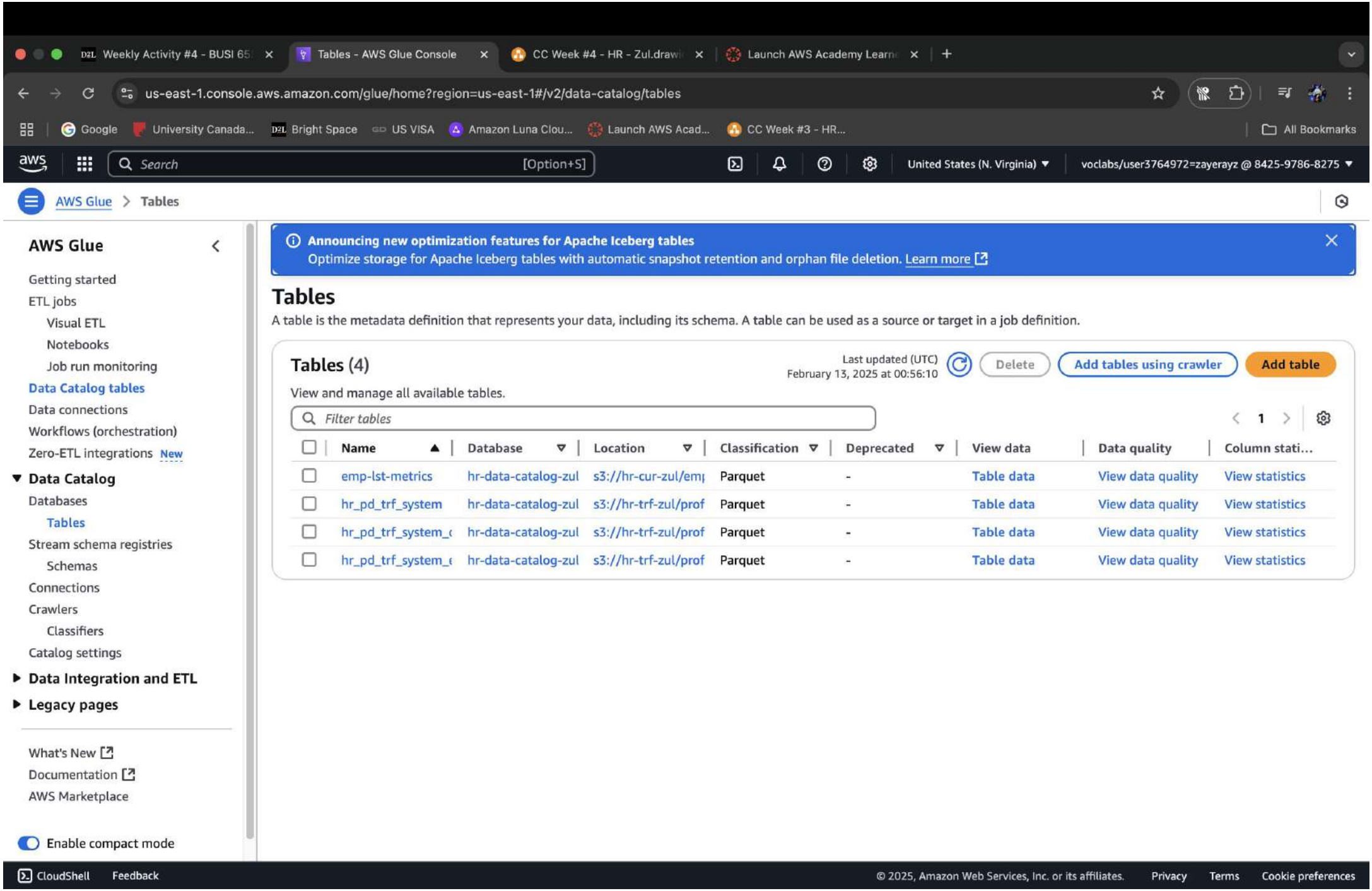

Weekly Activity #4 - BUSI 65
Applications | Elastic Beansta
CC Week #4 - HR - Zul.draw
Launch AWS Academy Learn
us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/applications
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Elastic Beanstalk > Applications
Elastic Beanstalk
Applications
Environments
Change history
Recent environments
HWA-Zul-env
HWA-Zul application is being deleted
Applications (0) Info
Actions
Create application
Filter results matching the display value
1
Application name
Environments
Date created
Last modified
ARN
No applications
No applications to display
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

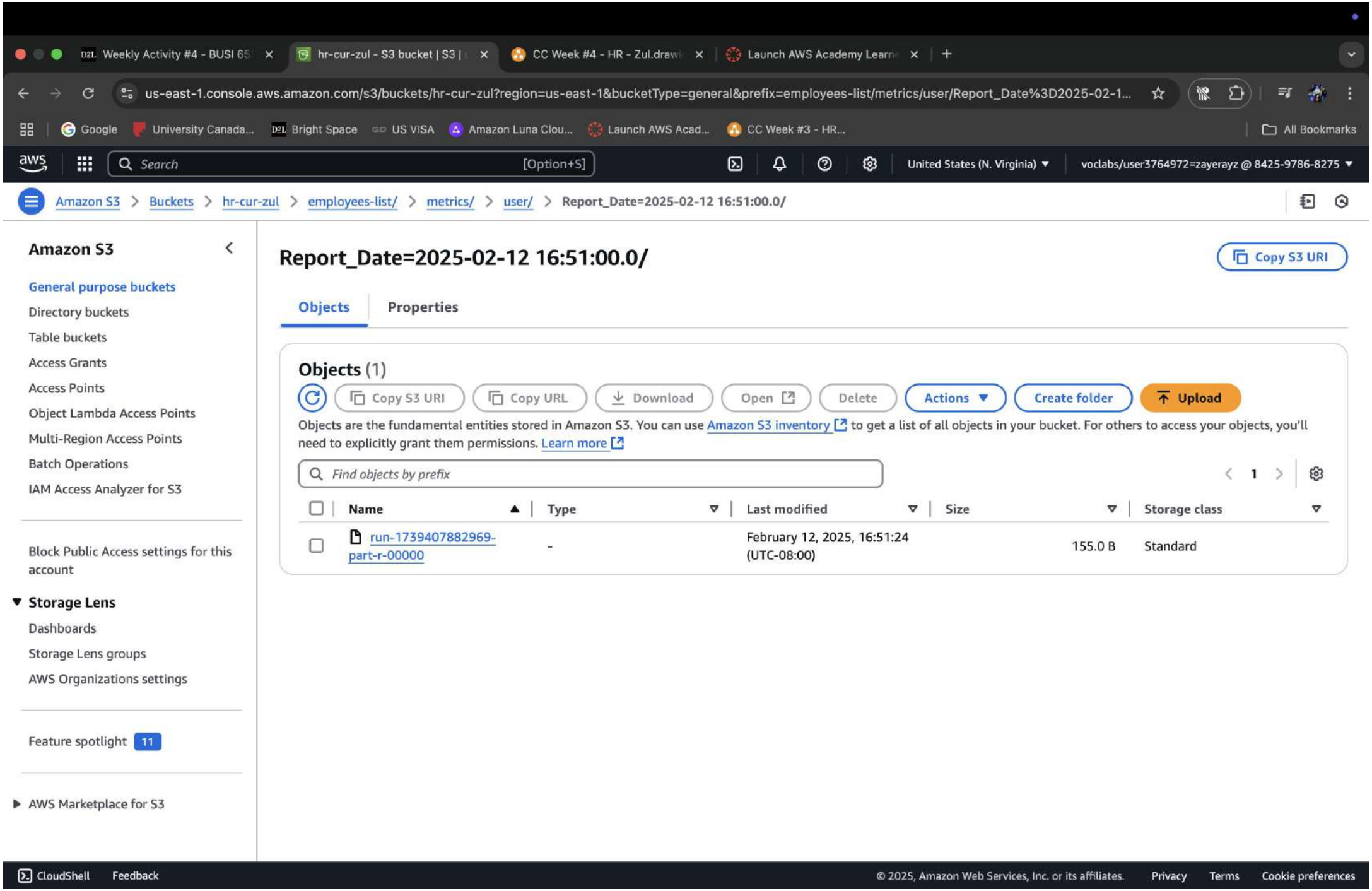

Weekly Activity #4 - BUSI 65
Applications | Elastic Beansta
CC Week #4 - HR - Zul.drawi
Launch AWS Academy Learne
us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/applications
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Elastic Beanstalk > Applications
Elastic Beanstalk
Applications
Environments
Change history
Recent environments
HWA-Zul-env
HWA-Zul application is being deleted
Applications (1) Info
Actions
Create application
Filter results matching the display value
1
Application name
Environments
Date created
Last modified
ARN
HWA-Zul
HWA-Zul-env
February 12, 2025 17:00:3...
February 12, 2025 17:15:1...
arn:aws:elasticbeanstal...
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

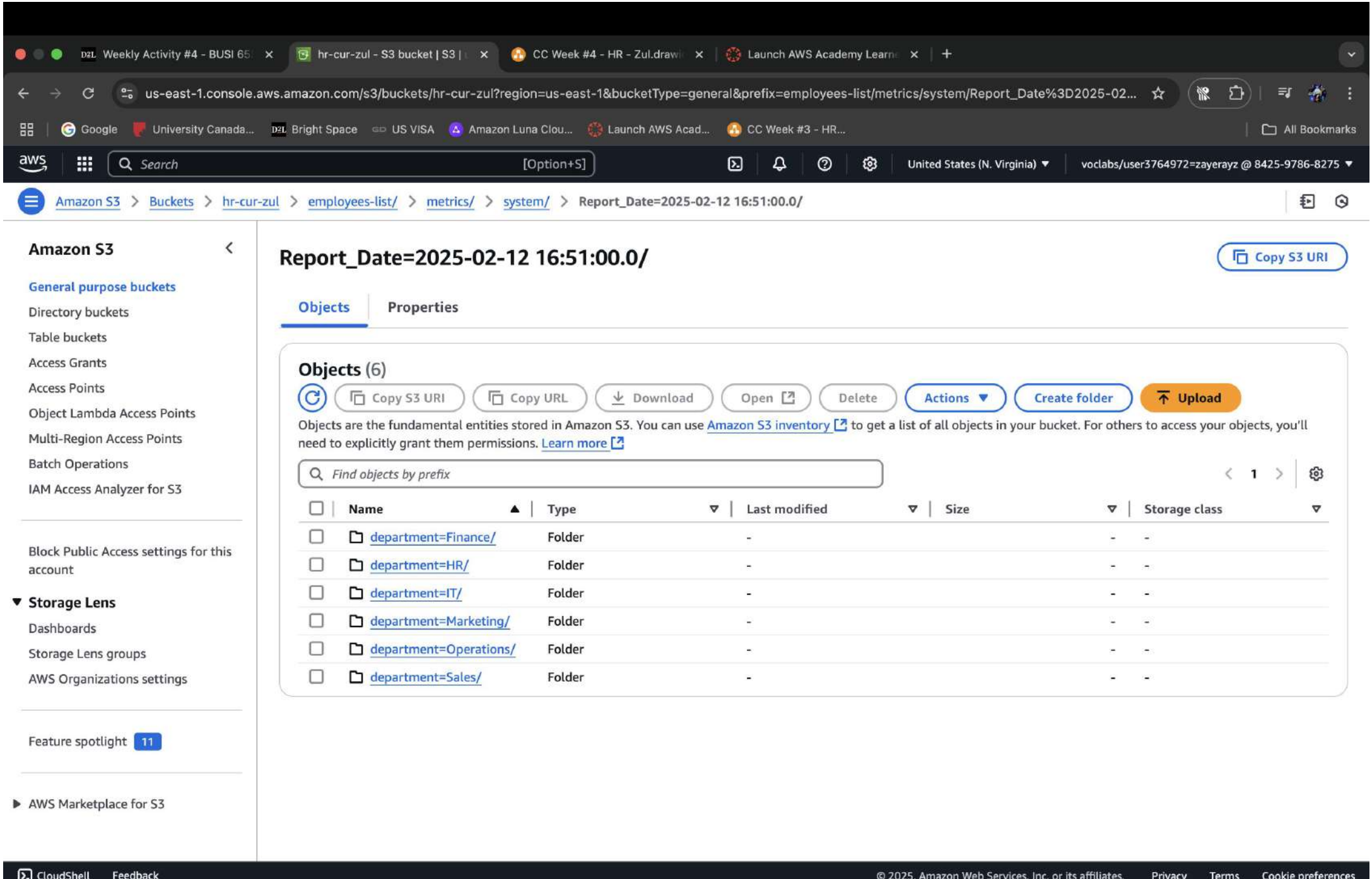

Weekly Activity #4 - BUS
hr-raw-zul - S3 bucket | S3
CC Week #4 - HR - Zul.draw
Launch AWS Academy Learn
us-east-1.console.aws.amazon.com/s3/buckets/hr-raw-zul?region=us-east-1&bucketType=general&prefix=professional-development/hr-app-log/year%3D2025/qua...
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > Buckets > hr-raw-zul > professional-development/ > hr-app-log/ > year=2025/ > quarter=01/ > month=1/ > day=28/
day=28/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions.
Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
BundleLogs-1739408809319.zip
zip
February 12, 2025, 17:10:05 (UTC-08:00)
56.6 KB
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

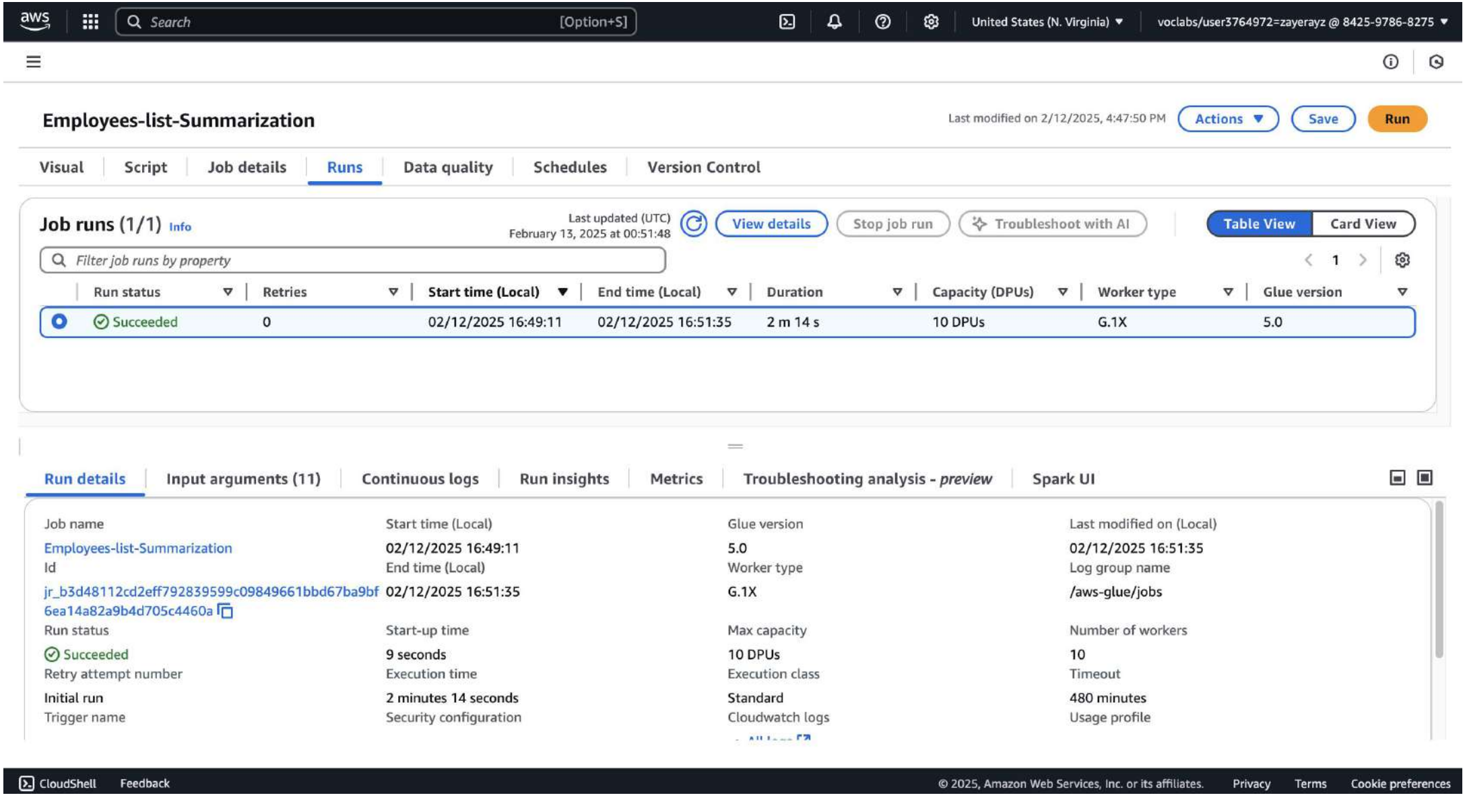

Weekly Activity #4 - BUSI 65
elasticbeanstalk-us-east-1-8
CC Week #4 - HR - Zul.draw
Launch AWS Academy Learn
us-east-1.console.aws.amazon.com/s3/buckets/elasticbeanstalk-us-east-1-842597868275?region=us-east-1&bucketType=general&prefix=resources/environments/...
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > Buckets > elasticbeanstalk-us-east-1-842597868275 > resources/ > environments/ > logs/ > bundle/ > e-b4h224uzps/ > i-0918986a...
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
i-0918986a4d19a9d19/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
BundleLogs-1739408809319.zip
zip
February 12, 2025, 17:06:50 (UTC-08:00)
56.6 KB
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Weekly Activity #4 - BUSI 65
Environment overview - logs
CC Week #4 - HR - Zul.draw
Launch AWS Academy Learn
us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/environment/dashboard?environmentId=e-b4h224uzps&tab=logs
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Elastic Beanstalk > Environments > HWA-Zul-env
Elastic Beanstalk
Applications
Environments
Change history
Application: HWA-Zul
Environment: HWA-Zul-env
Go to environment
Configuration
Events
Health
Logs
Monitoring
Alarms
Managed updates
Tags
Recent environments
HWA-Zul-env
Environment successfully launched.
Retrieved requested logs.
HWA-Zul-env Info
Actions
Upload and deploy
Environment overview
Health
Ok
Environment ID
e-b4h224uzps
Domain
HWA-Zul.us-east-1.elasticbeanstalk.com
Application name
HWA-Zul
Platform
Change version
Platform
Python 3.13 running on 64bit Amazon Linux 2023/4.4.0
Running version
-
Platform state
Supported
Events
Health
Logs
Monitoring
Alarms
Managed updates
Tags
Logs Info
Request logs
Click Request logs to retrieve the last 100 lines of logs or the entire set of logs from each EC2 instance. Learn more
Log file
Time
EC2 instance
Type
Download
February 12, 2025 17:06:49:398 (UTC-8)
i-0918986a4d19a9d19
bundle
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Weekly Activity #4 - BUSI 65
Tables - AWS Glue Console
CC Week #4 - HR - Zul.draw
Launch AWS Academy Learn
us-east-1.console.aws.amazon.com/glue/home?region=us-east-1#/v2/data-catalog/tables
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
AWS Glue > Tables
AWS Glue
Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Zero-ETL integrations New
Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings
Data Integration and ETL
Legacy pages
What's New
Documentation
AWS Marketplace
Enable compact mode
Announcing new optimization features for Apache Iceberg tables
Optimize storage for Apache Iceberg tables with automatic snapshot retention and orphan file deletion. Learn more
Tables
A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.
Tables (4)
Last updated (UTC)
February 13, 2025 at 00:56:10
Delete
Add tables using crawler
Add table
View and manage all available tables.
Filter tables
1
Name
Database
Location
Classification
Deprecated
View data
Data quality
Column stati...
emp-lst-metrics
hr-data-catalog-zul
s3://hr-cur-zul/emp
Parquet
-
Table data
View data quality
View statistics
hr_pd_trf_system
hr-data-catalog-zul
s3://hr-trf-zul/prof
Parquet
-
Table data
View data quality
View statistics
hr_pd_trf_system_c
hr-data-catalog-zul
s3://hr-trf-zul/prof
Parquet
-
Table data
View data quality
View statistics
hr_pd_trf_system_c
hr-data-catalog-zul
s3://hr-trf-zul/prof
Parquet
-
Table data
View data quality
View statistics
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Weekly Activity #4 - BUSI 65
hr-cur-zul - S3 bucket | S3 |
CC Week #4 - HR - Zul.draw
Launch AWS Academy Learn
us-east-1.console.aws.amazon.com/s3/buckets/hr-cur-zul?region=us-east-1&bucketType=general&prefix=employees-list/metrics/user/Report_Date%3D2025-02-1...
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > Buckets > hr-cur-zul > employees-list/ > metrics/ > user/ > Report_Date=2025-02-12 16:51:00.0/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
Report_Date=2025-02-12 16:51:00.0/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
run-1739407882969-part-r-00000
-
February 12, 2025, 16:51:24 (UTC-08:00)
155.0 B
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Weekly Activity #4 - BUSI 65
hr-cur-zul - S3 bucket | S3 |
CC Week #4 - HR - Zul.draw
Launch AWS Academy Learn
us-east-1.console.aws.amazon.com/s3/buckets/hr-cur-zul?region=us-east-1&bucketType=general&prefix=employees-list/metrics/system/Report_Date%3D2025-02...
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
CC Week #3 - HR...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > Buckets > hr-cur-zul > employees-list/ > metrics/ > system/ > Report_Date=2025-02-12 16:51:00.0/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
Report_Date=2025-02-12 16:51:00.0/
Copy S3 URI
Objects
Properties
Objects (6)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
department=Finance/ Folder - - -
department=HR/ Folder - - -
department=IT/ Folder - - -
department=Marketing/ Folder - - -
department=Operations/ Folder - - -
department=Sales/ Folder - - -
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

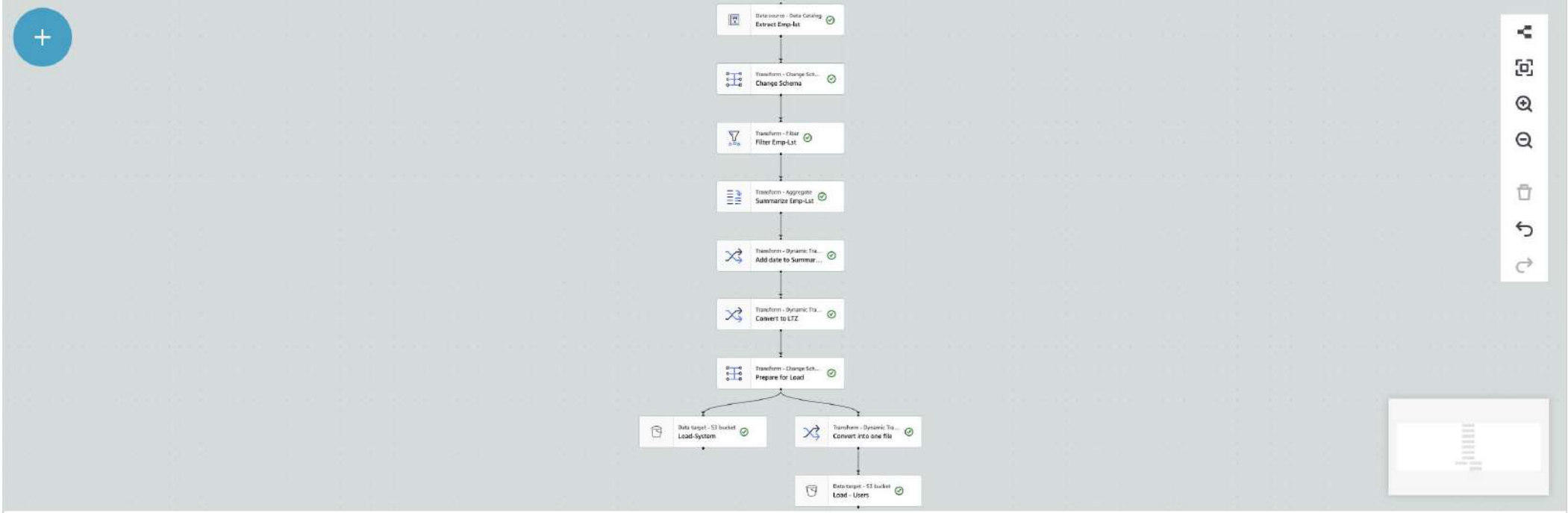

aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Employees-list-Summarization
Last modified on 2/12/2025, 4:47:50 PM
Actions
Save
Run
Visual
Script
Job details
Runs
Data quality
Schedules
Version Control
Job runs (1/1) Info
Last updated (UTC)
February 13, 2025 at 00:51:48
View details
Stop job run
Troubleshoot with AI
Table View
Card View
Filter job runs by property
1
Run status
Retries
Start time (Local)
End time (Local)
Duration
Capacity (DPUs)
Worker type
Glue version
Succeeded
0
02/12/2025 16:49:11
02/12/2025 16:51:35
2 m 14 s
10 DPUs
G.1X
5.0
Run details
Input arguments (11)
Continuous logs
Run insights
Metrics
Troubleshooting analysis - preview
Spark UI
Job name
Employees-list-Summarization
Id
jr_b3d48112cd2eff792839599c09849661bbd67ba9bf6ea14a82a9b4d705c4460a
Run status
Succeeded
Retry attempt number
Initial run
Trigger name
Start time (Local)
02/12/2025 16:49:11
End time (Local)
02/12/2025 16:51:35
Start-up time
9 seconds
Execution time
2 minutes 14 seconds
Security configuration
Glue version
5.0
Worker type
G.1X
Max capacity
10 DPUs
Execution class
Standard
Cloudwatch logs
Last modified on (Local)
02/12/2025 16:51:35
Log group name
/aws-glue/jobs
Number of workers
10
Timeout
480 minutes
Usage profile
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

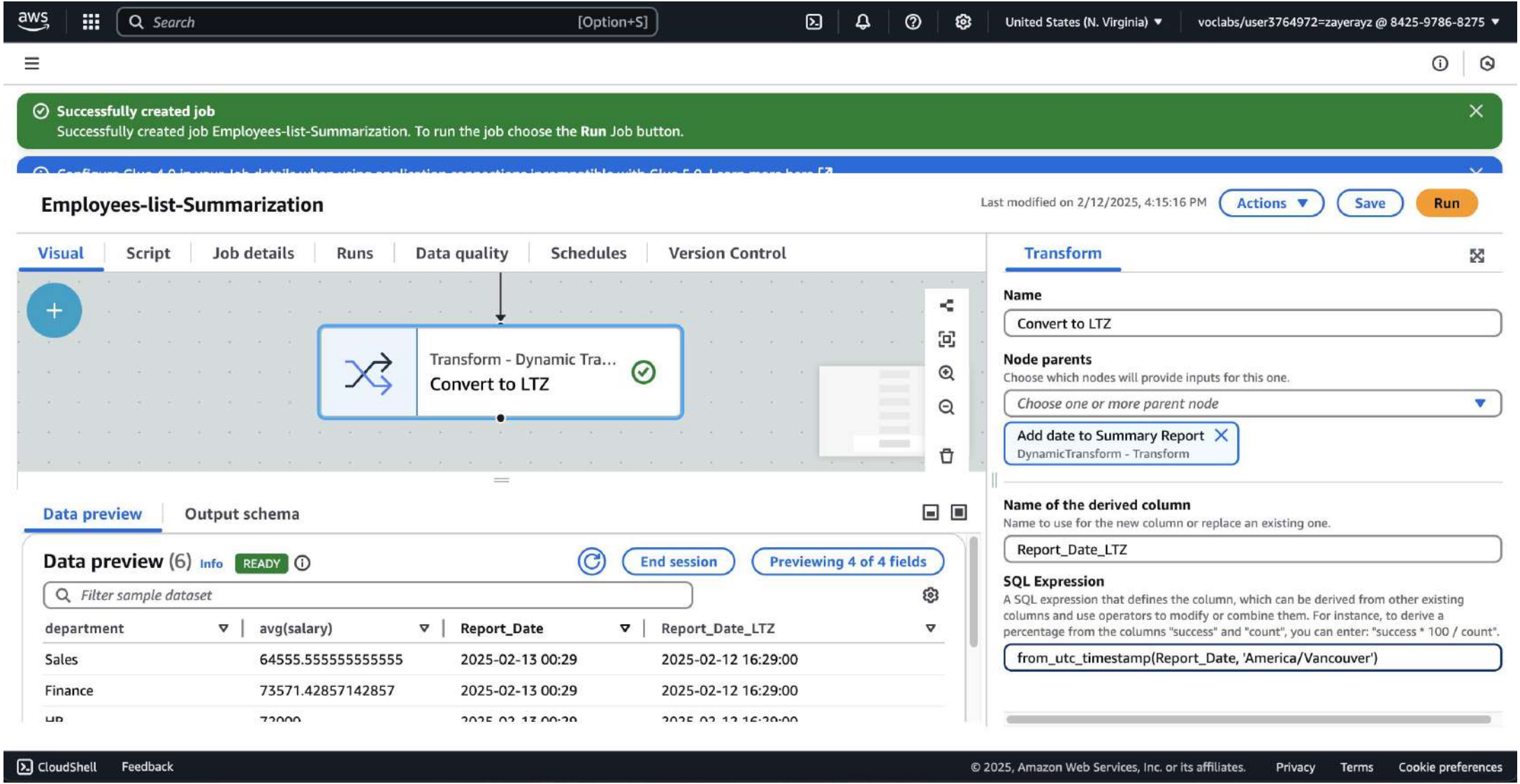
Successfully created job
Successfully created job Employees-list-Summarization. To run the job choose the **Run** Job button.

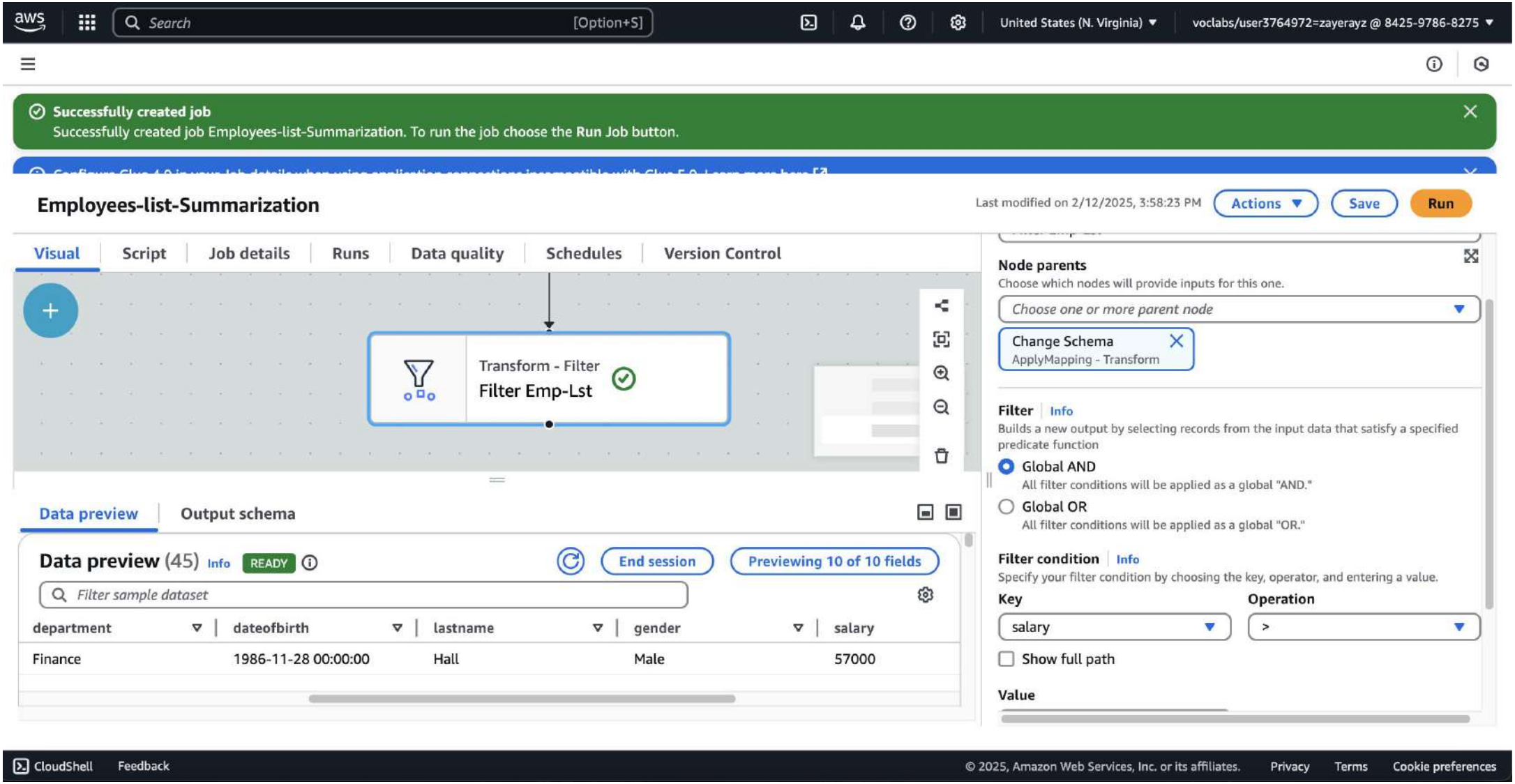
Employees-list-Summarization

Last modified on 2/12/2025, 4:47:50 PM | Actions | Save | Run

Visual | Script | Job details | Runs | Data quality | Schedules | Version Control




Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Successfully created job
Successfully created job Employees-list-Summarization. To run the job choose the Run Job button.
Employees-list-Summarization
Last modified on 2/12/2025, 4:15:16 PM
Actions
Save
Run
Visual
Script
Job details
Runs
Data quality
Schedules
Version Control
Transform - Dynamic Tra...
Convert to LTZ
Transform
Name
Convert to LTZ
Node parents
Choose which nodes will provide inputs for this one.
Choose one or more parent node
Add date to Summary Report
DynamicTransform - Transform
Name of the derived column
Name to use for the new column or replace an existing one.
Report_Date_LTZ
SQL Expression
A SQL expression that defines the column, which can be derived from other existing columns and use operators to modify or combine them. For instance, to derive a percentage from the columns "success" and "count", you can enter: "success * 100 / count".
from_utc_timestamp(Report_Date, 'America/Vancouver')
Data preview
Output schema
Data preview (6)
Info
READY
End session
Previewing 4 of 4 fields
Filter sample dataset
department
avg(salary)
Report_Date
Report_Date_LTZ
Sales
64555.555555555555
2025-02-13 00:29
2025-02-12 16:29:00
Finance
73571.42857142857
2025-02-13 00:29
2025-02-12 16:29:00
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Successfully created job
Successfully created job Employees-list-Summarization. To run the job choose the Run Job button.
Employees-list-Summarization
Last modified on 2/12/2025, 3:58:23 PM
Actions
Save
Run
Visual
Script
Job details
Runs
Data quality
Schedules
Version Control
Transform - Filter
Filter Emp-Lst
Node parents
Choose which nodes will provide inputs for this one.
Choose one or more parent node
Change Schema
ApplyMapping - Transform
Filter
Info
Builds a new output by selecting records from the input data that satisfy a specified predicate function
Global AND
All filter conditions will be applied as a global "AND."
Global OR
All filter conditions will be applied as a global "OR."
Filter condition
Info
Specify your filter condition by choosing the key, operator, and entering a value.
Key
Operation
salary
>
Show full path
Value
Data preview
Output schema
Data preview (45)
Info
READY
End session
Previewing 10 of 10 fields
Filter sample dataset
department
dateofbirth
lastname
gender
salary
Finance
1986-11-28 00:00:00
Hall
Male
57000
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

AWS Glue

Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Zero-ETL integrations New
Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings
Data Integration and ETL
Legacy pages

What's New
Documentation
AWS Marketplace

Announcing new optimization features for Apache Iceberg tables
Optimize storage for Apache Iceberg tables with automatic snapshot retention and orphan file deletion. Learn more

hr-data-catalog-zul

Last updated (UTC)
February 12, 2025 at 23:53:23

Edit | Delete

Database properties

Name	Description	Location	Created on (UTC)
hr-data-catalog-zul	-	-	February 12, 2025 at 23:43:52

Tables (3)

Last updated (UTC)
February 12, 2025 at 23:53:25

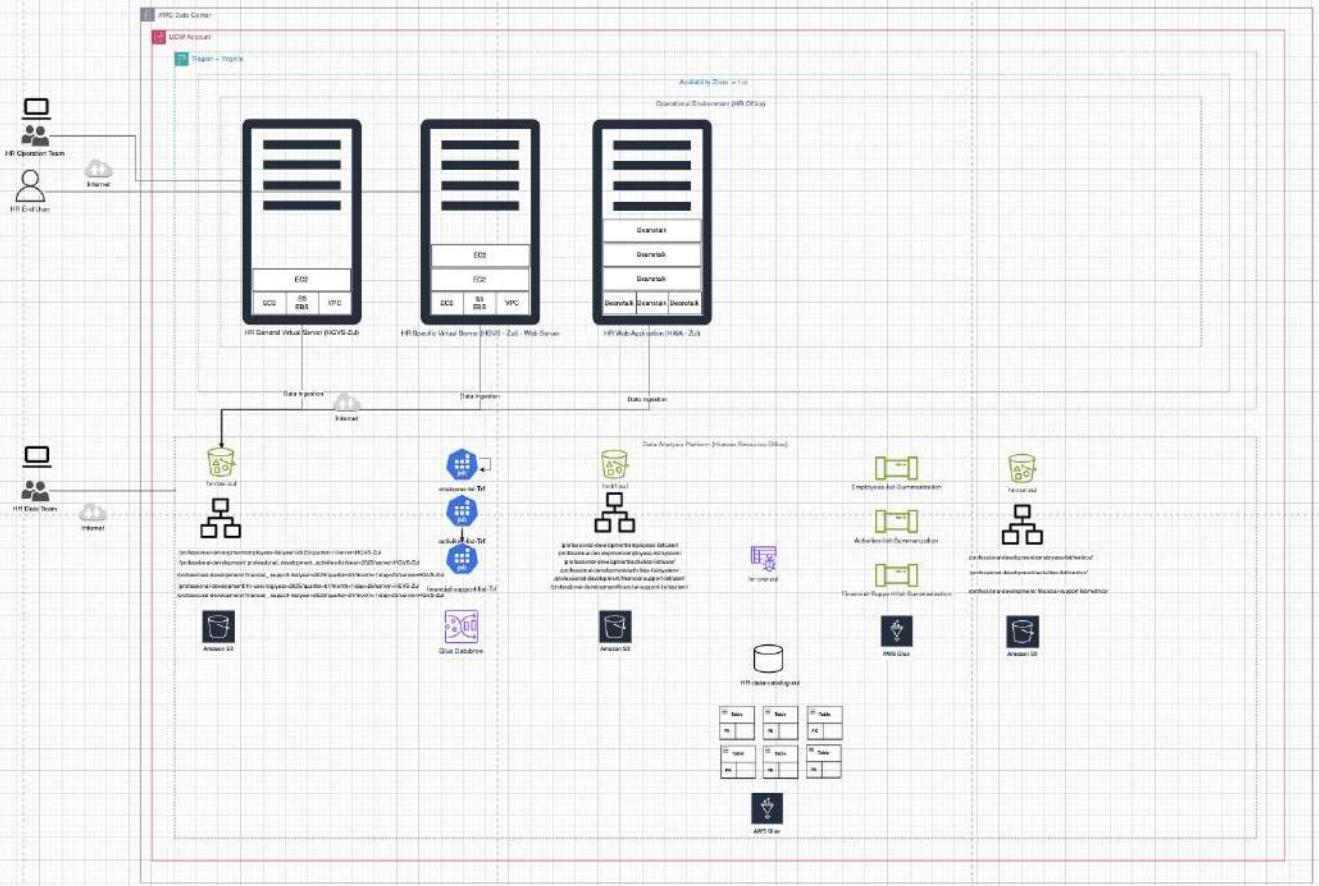
Delete | Add tables using crawler | Add table

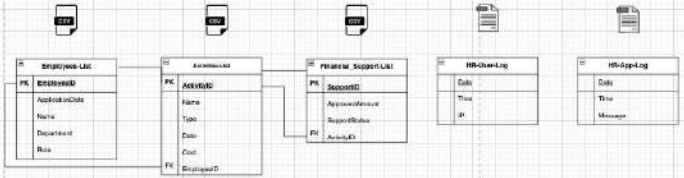
View and manage all available tables.

Filter tables

1

Name	Database	Location	Classification	Deprecated	View data	Data quality	Column stati...
hr_pd_trf_system	hr-data-catalog-zul	s3://hr-trf-zul/prof	Parquet	-	Table data	View data quality	View statistics
hr_pd_trf_system_c	hr-data-catalog-zul	s3://hr-trf-zul/prof	Parquet	-	Table data	View data quality	View statistics
hr_pd_trf_system_e	hr-data-catalog-zul	s3://hr-trf-zul/prof	Parquet	-	Table data	View data quality	View statistics



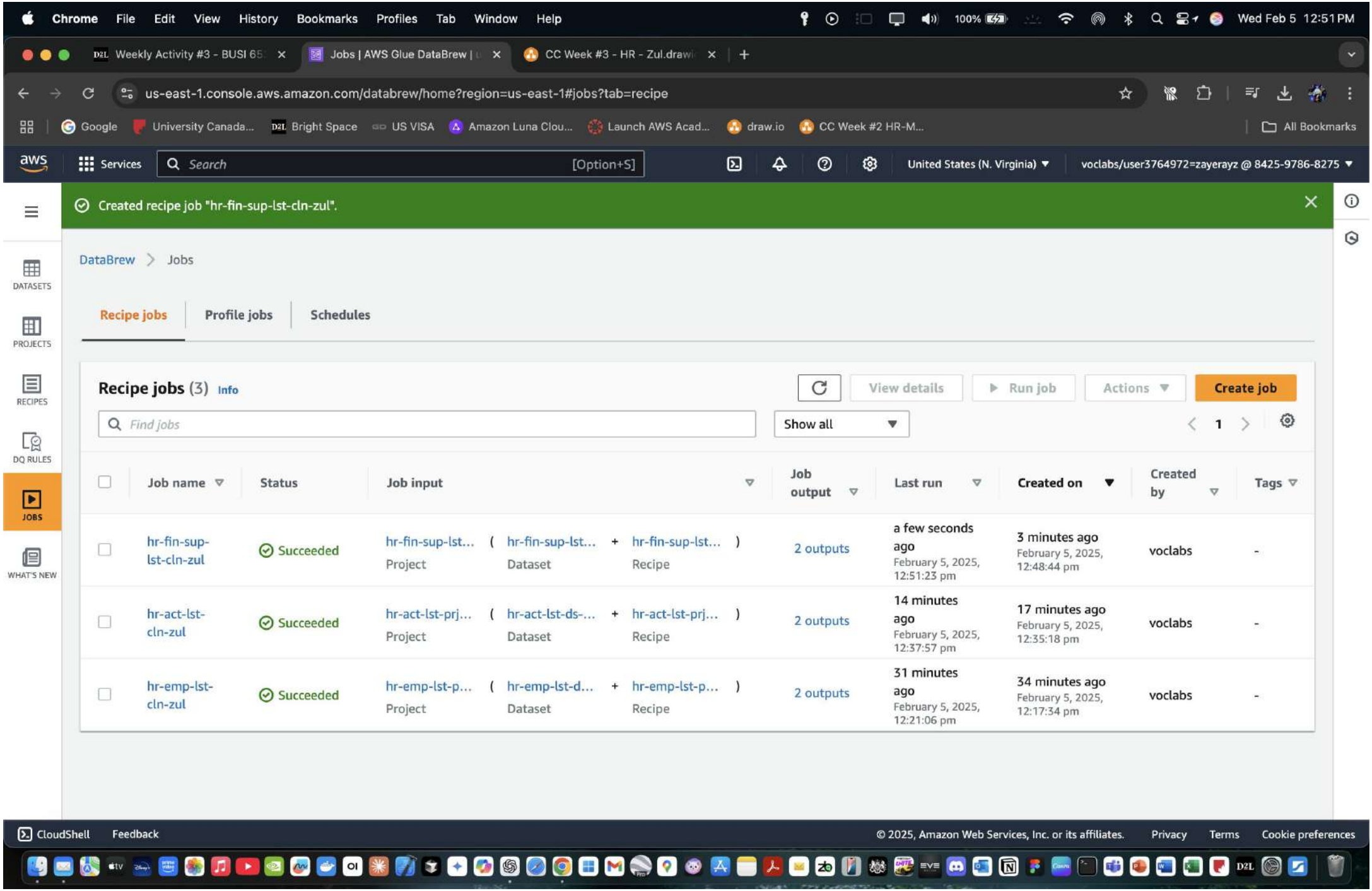


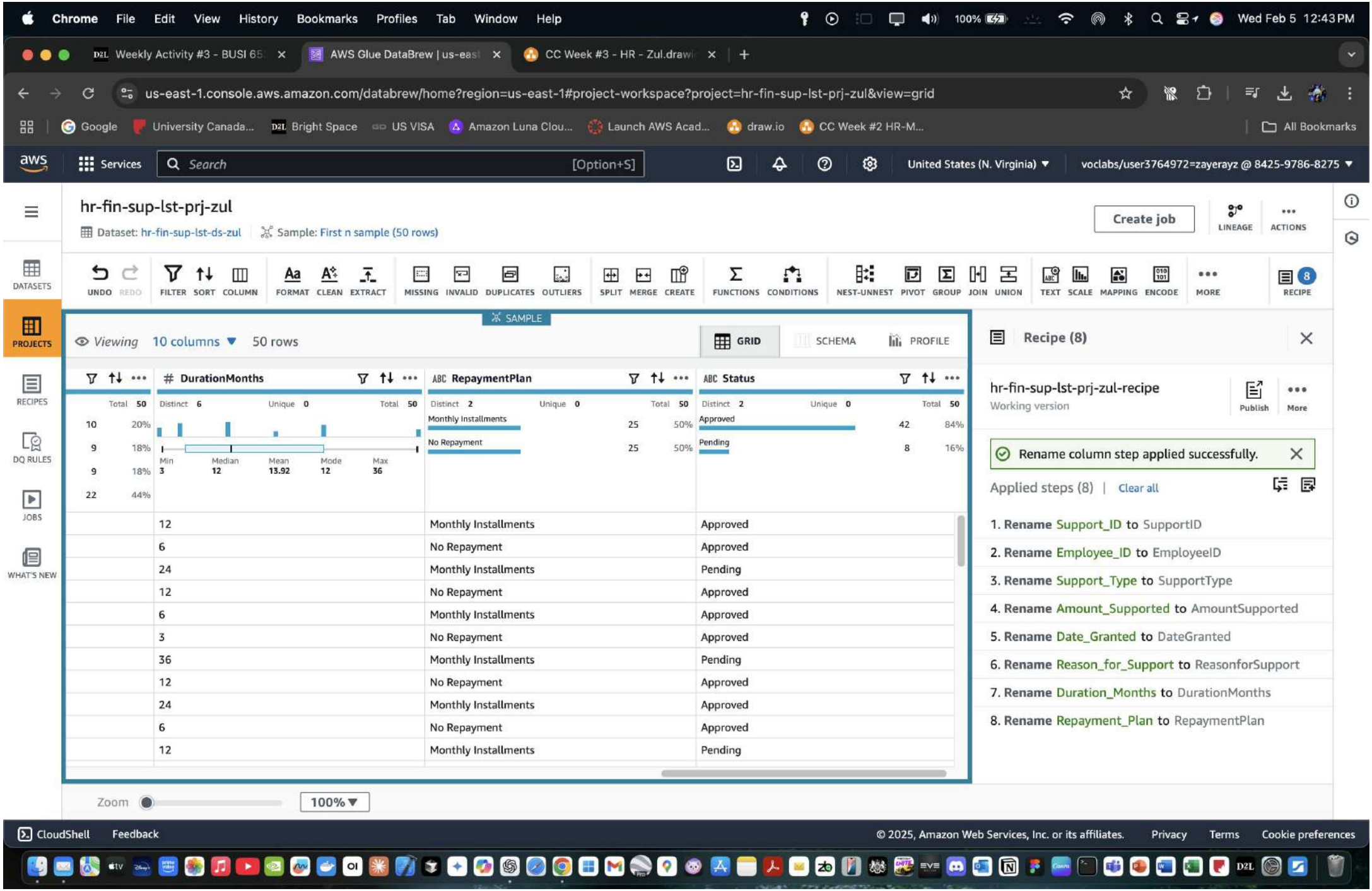
Exploratory Data Analysis

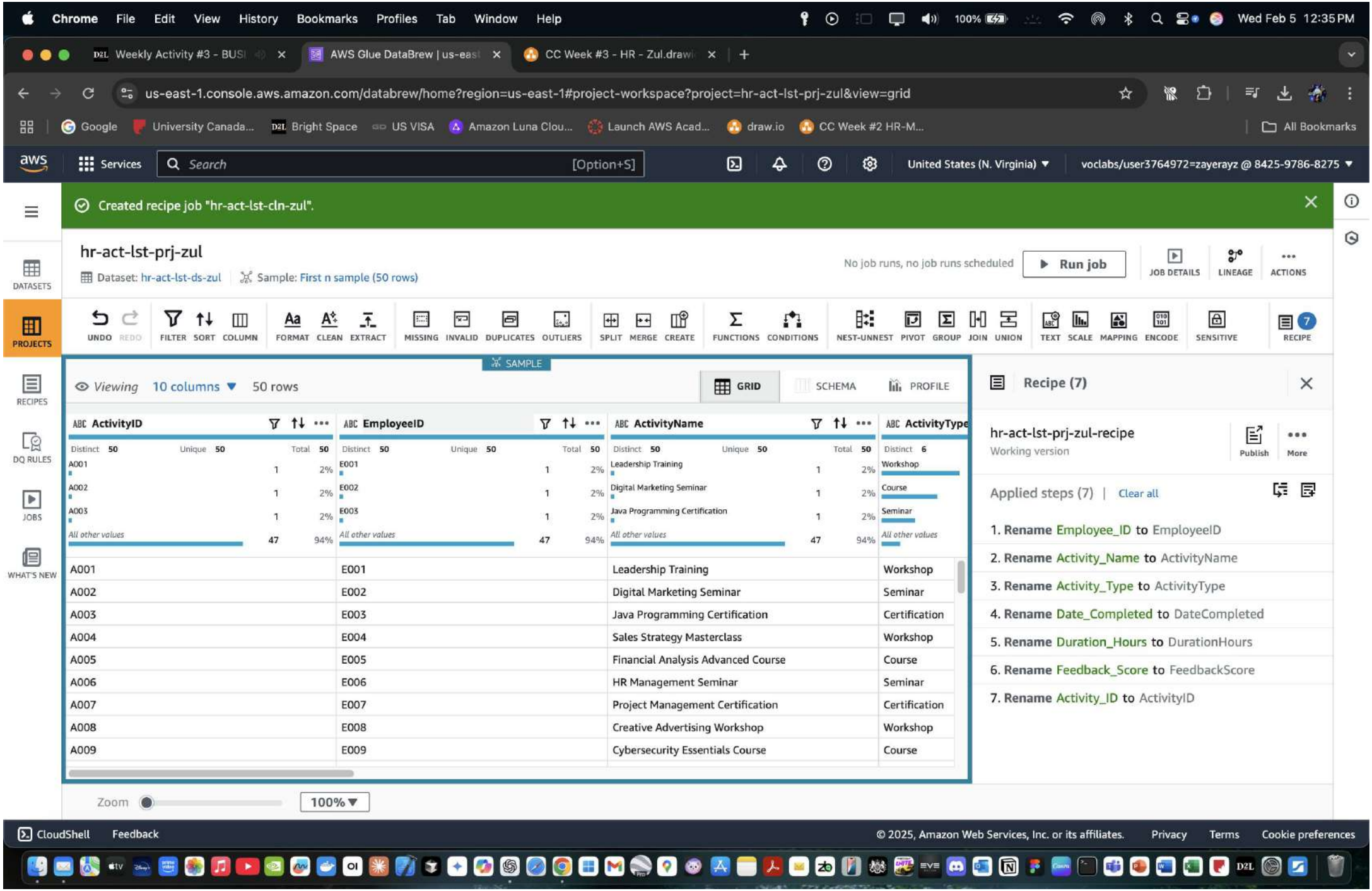
	Summarization Techniques	Patterns
Employees-List	Grouping Filter Projection Min Max Average Sum	Trends Seasonality Correlation Clusters Anomaly
Activities-List		
Financial_Support-List		
Registrar-User-Log		
Registrar-App-Log		

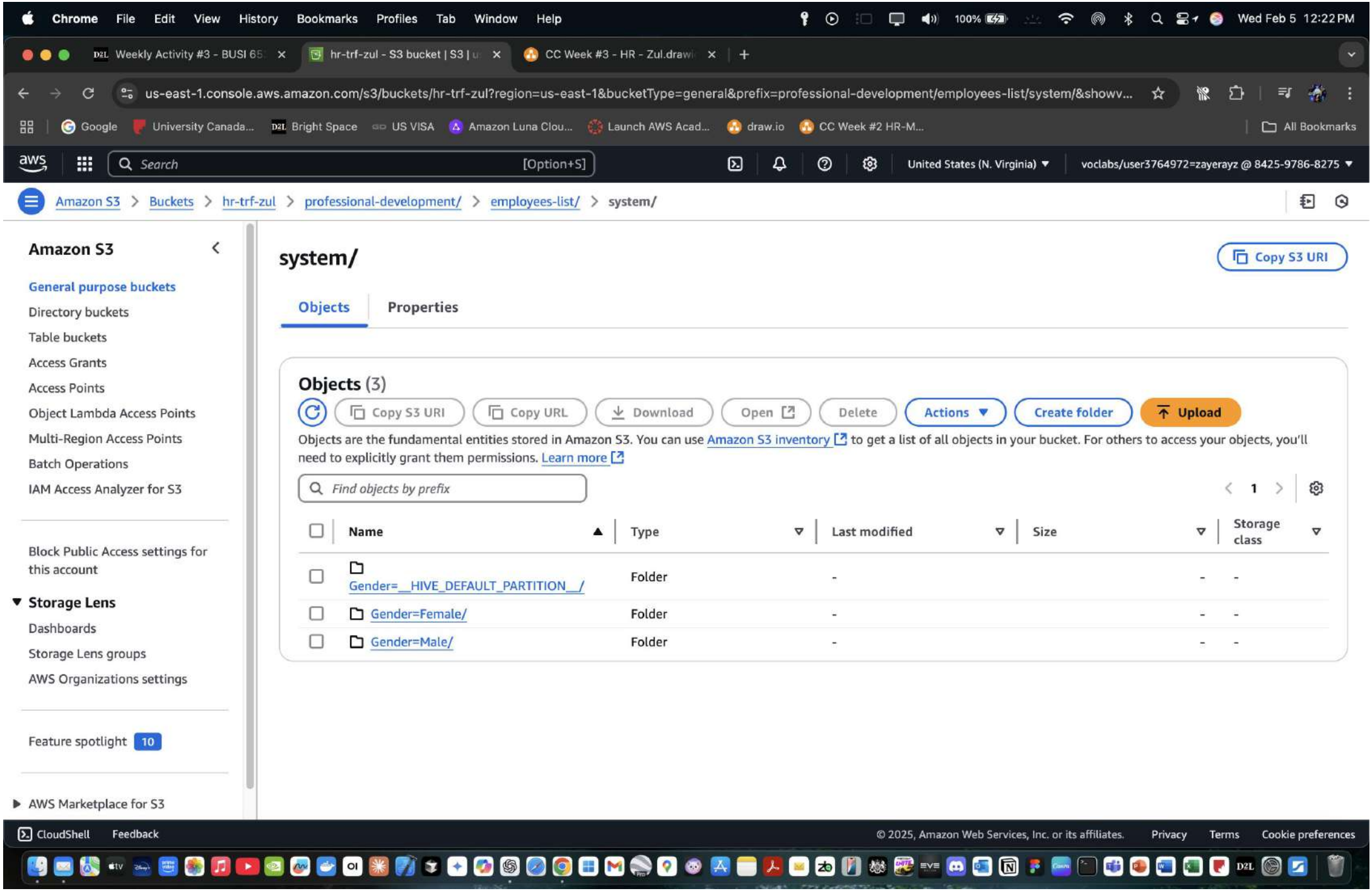
Data Cleaning Cost Analysis

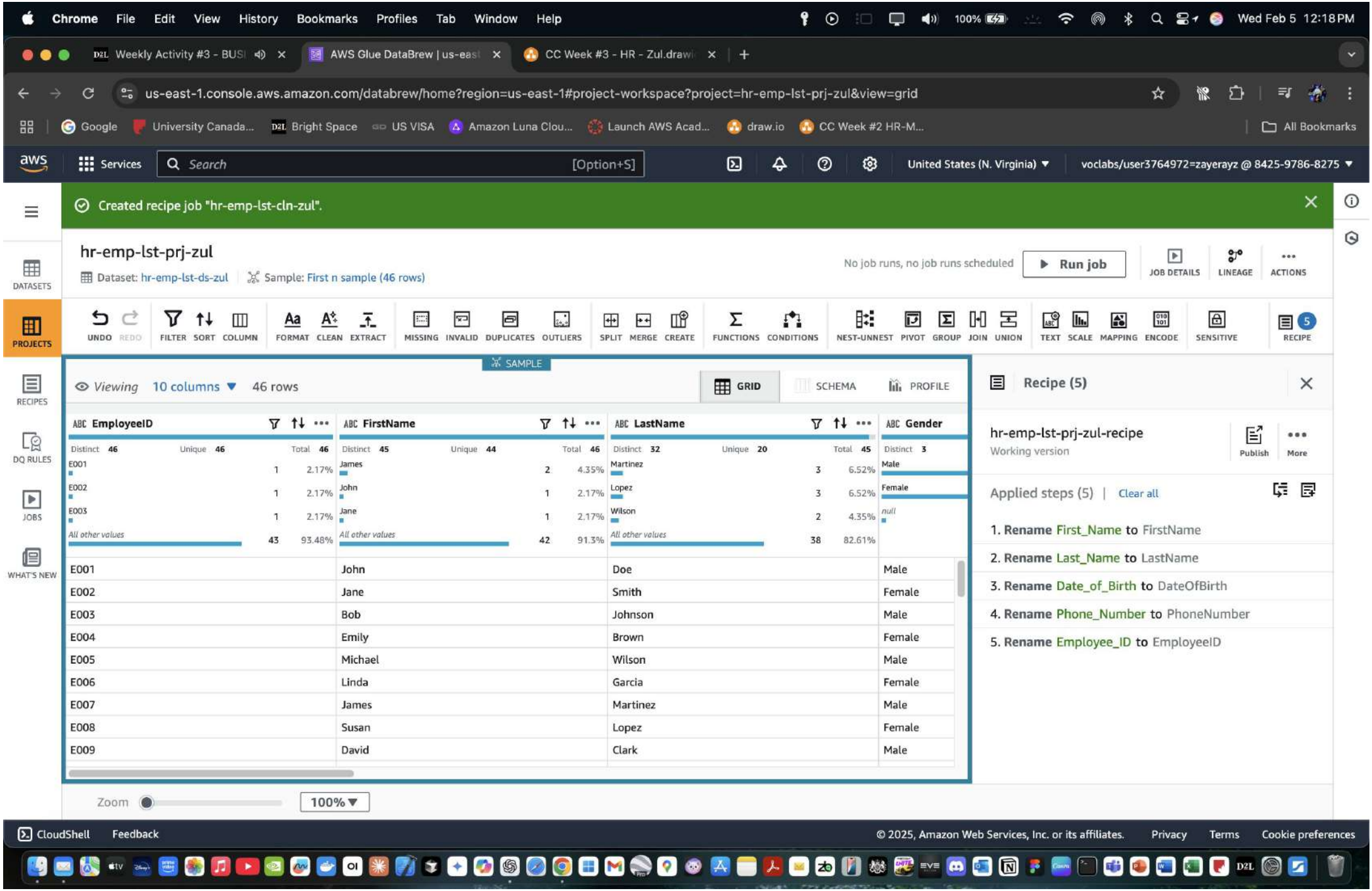
	DataBrew interactive sessions (#)	DataBrew Job Nodes	DataBrew Job Duration
Employees-List	1	5 x G	01:03 x01:42
Activities-List	1	5 x G	00:01x01:58
Financial_Support-List	1	5 x G	01:02x01:32
Registrar-User-Log	1	5 x G	00:13x01:31

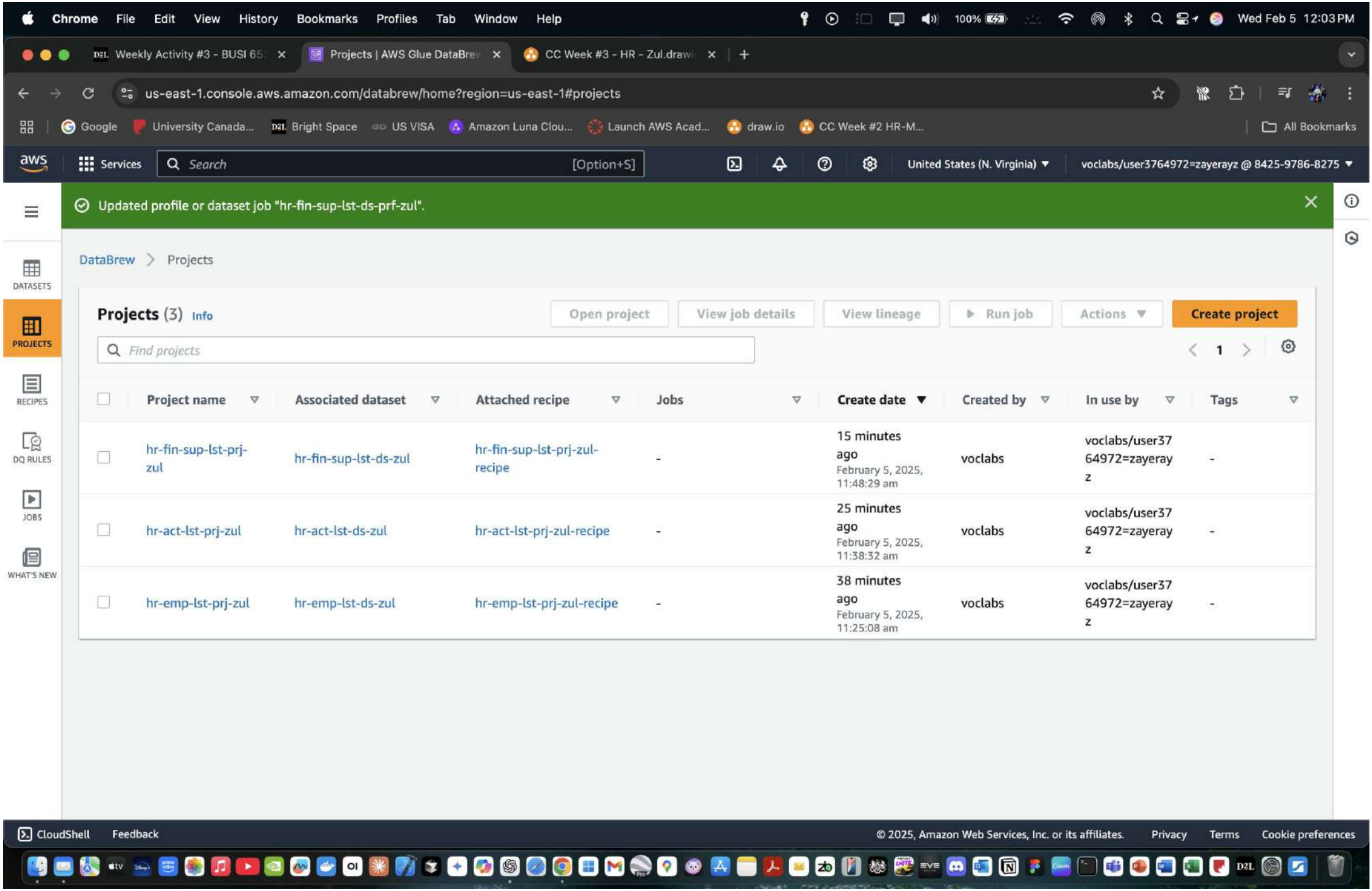

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Feb 5 12:51 PM
Weekly Activity #3 - BUSI 65
Jobs | AWS Glue DataBrew |
CC Week #3 - HR - Zul.draw
us-east-1.console.aws.amazon.com/databrew/home?region=us-east-1#jobs?tab=recipe
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
draw.io
CC Week #2 HR-M...
All Bookmarks
aws
Services
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Created recipe job "hr-fin-sup-lst-cln-zul".
DataBrew
Jobs
DATASETS
PROJECTS
RECIPES
DQ RULES
JOBS
WHAT'S NEW
Recipe jobs
Profile jobs
Schedules
Recipe jobs (3) Info
View details
Run job
Actions
Create job
Find jobs
Show all
1
Job name
Status
Job input
Job output
Last run
Created on
Created by
Tags
hr-fin-sup-lst-cln-zul
Succeeded
hr-fin-sup-lst... (hr-fin-sup-lst... + hr-fin-sup-lst...)
Project Dataset Recipe
2 outputs
a few seconds ago
February 5, 2025, 12:51:23 pm
3 minutes ago
February 5, 2025, 12:48:44 pm
voclabs
-
hr-act-lst-cln-zul
Succeeded
hr-act-lst-prj... (hr-act-lst-ds-... + hr-act-lst-prj...)
Project Dataset Recipe
2 outputs
14 minutes ago
February 5, 2025, 12:37:57 pm
17 minutes ago
February 5, 2025, 12:35:18 pm
voclabs
-
hr-emp-lst-cln-zul
Succeeded
hr-emp-lst-p... (hr-emp-lst-d... + hr-emp-lst-p...)
Project Dataset Recipe
2 outputs
31 minutes ago
February 5, 2025, 12:21:06 pm
34 minutes ago
February 5, 2025, 12:17:34 pm
voclabs
-
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Feb 5 12:43 PM
Weekly Activity #3 - BUSI 65
AWS Glue DataBrew | us-east
CC Week #3 - HR - Zul.draw
us-east-1.console.aws.amazon.com/databrew/home?region=us-east-1#project-workspace?project=hr-fin-sup-lst-prj-zul&view=grid
Google University Canada... Bright Space US VISA Amazon Luna Clou... Launch AWS Acad... draw.io CC Week #2 HR-M... All Bookmarks
Services Search [Option+S] United States (N. Virginia) voclabs/user3764972=zayerayz @ 8425-9786-8275
hr-fin-sup-lst-prj-zul
Dataset: hr-fin-sup-lst-ds-zul Sample: First n sample (50 rows)
Create job LINEAGE ACTIONS
DATASETS PROJECTS RECIPES DQ RULES JOBS WHAT'S NEW
UNDO REDO FILTER SORT COLUMN FORMAT CLEAN EXTRACT MISSING INVALID DUPLICATES OUTLIERS SPLIT MERGE CREATE FUNCTIONS CONDITIONS NEST-UNNEST PIVOT GROUP JOIN UNION TEXT SCALE MAPPING ENCODE MORE RECIPE 8
SAMPLE
Viewing 10 columns 50 rows GRID SCHEMA PROFILE
DurationMonths
Total 50
Distinct 6 Unique 0 Total 50
10 20%
9 18%
9 18%
22 44%
Min 3 Median 12 Mean 13.92 Mode 12 Max 36
ABC RepaymentPlan
Distinct 2 Unique 0 Total 50
Monthly Installments 25 50%
No Repayment 25 50%
ABC Status
Distinct 2 Unique 0 Total 50
Approved 42 84%
Pending 8 16%
12 Monthly Installments Approved
6 No Repayment Approved
24 Monthly Installments Pending
12 No Repayment Approved
6 Monthly Installments Approved
3 No Repayment Approved
36 Monthly Installments Pending
12 No Repayment Approved
24 Monthly Installments Approved
6 No Repayment Approved
12 Monthly Installments Pending
Recipe (8)
hr-fin-sup-lst-prj-zul-recipe
Working version
Publish More
Rename column step applied successfully.
Applied steps (8) | Clear all
1. Rename Support_ID to SupportID
2. Rename Employee_ID to EmployeeID
3. Rename Support_Type to SupportType
4. Rename Amount_Supported to AmountSupported
5. Rename Date_Granted to DateGranted
6. Rename Reason_for_Support to ReasonforSupport
7. Rename Duration_Months to DurationMonths
8. Rename Repayment_Plan to RepaymentPlan
Zoom 100%
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences


Created recipe job "hr-act-lst-cln-zul".
hr-act-lst-prj-zul
Dataset: hr-act-lst-ds-zul
Sample: First n sample (50 rows)
No job runs, no job runs scheduled
Run job
JOB DETAILS
LINEAGE
ACTIONS
DATASETS
PROJECTS
RECIPES
DQ RULES
JOBS
WHAT'S NEW
UNDO REDO FILTER SORT COLUMN FORMAT CLEAN EXTRACT MISSING INVALID DUPLICATES OUTLIERS SPLIT MERGE CREATE FUNCTIONS CONDITIONS NEST-UNNEST PIVOT GROUP JOIN UNION TEXT SCALE MAPPING ENCODE SENSITIVE RECIPE
SAMPLE
Viewing 10 columns 50 rows
GRID SCHEMA PROFILE
ABC ActivityID
ABC EmployeeID
ABC ActivityName
ABC ActivityType
Recipe (7)
hr-act-lst-prj-zul-recipe
Working version
Publish
More
Applied steps (7) | Clear all
1. Rename Employee_ID to EmployeeID
2. Rename Activity_Name to ActivityName
3. Rename Activity_Type to ActivityType
4. Rename Date_Completed to DateCompleted
5. Rename Duration_Hours to DurationHours
6. Rename Feedback_Score to FeedbackScore
7. Rename Activity_ID to ActivityID
A001 E001 Leadership Training Workshop
A002 E002 Digital Marketing Seminar Seminar
A003 E003 Java Programming Certification Certification
A004 E004 Sales Strategy Masterclass Workshop
A005 E005 Financial Analysis Advanced Course Course
A006 E006 HR Management Seminar Seminar
A007 E007 Project Management Certification Certification
A008 E008 Creative Advertising Workshop Workshop
A009 E009 Cybersecurity Essentials Course Course
Zoom 100%
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Feb 5 12:22 PM
Weekly Activity #3 - BUSI 65
hr-trf-zul - S3 bucket | S3 |
CC Week #3 - HR - Zul.draw
us-east-1.console.aws.amazon.com/s3/buckets/hr-trf-zul?region=us-east-1&bucketType=general&prefix=professional-development/employees-list/system/&showv...
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
draw.io
CC Week #2 HR-M...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > Buckets > hr-trf-zul > professional-development/ > employees-list/ > system/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 10
AWS Marketplace for S3
system/
Copy S3 URI
Objects
Properties
Objects (3)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
Gender=__HIVE_DEFAULT_PARTITION__/
Folder
Gender=Female/
Folder
Gender=Male/
Folder
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Created recipe job "hr-emp-lst-cln-zul".
hr-emp-lst-prj-zul
Dataset: hr-emp-lst-ds-zul
Sample: First n sample (46 rows)
No job runs, no job runs scheduled
Run job
JOB DETAILS
LINEAGE
ACTIONS
DATASETS
PROJECTS
RECIPES
DQ RULES
JOBS
WHAT'S NEW
UNDO REDO FILTER SORT COLUMN FORMAT CLEAN EXTRACT MISSING INVALID DUPLICATES OUTLIERS SPLIT MERGE CREATE FUNCTIONS CONDITIONS NEST-UNNEST PIVOT GROUP JOIN UNION TEXT SCALE MAPPING ENCODE SENSITIVE RECIPE 5
SAMPLE
Viewing 10 columns 46 rows
GRID SCHEMA PROFILE
ABC EmployeeID
Distinct 46 Unique 46 Total 46
E001 1 2.17%
E002 1 2.17%
E003 1 2.17%
All other values 43 93.48%
ABC FirstName
Distinct 45 Unique 44 Total 46
James 2 4.35%
John 1 2.17%
Jane 1 2.17%
All other values 42 91.3%
ABC LastName
Distinct 32 Unique 20 Total 45
Martinez 3 6.52%
Lopez 3 6.52%
Wilson 2 4.35%
All other values 38 82.61%
ABC Gender
Distinct 3
Male
Female
null
E001 John Doe Male
E002 Jane Smith Female
E003 Bob Johnson Male
E004 Emily Brown Female
E005 Michael Wilson Male
E006 Linda Garcia Female
E007 James Martinez Male
E008 Susan Lopez Female
E009 David Clark Male
Recipe (5)
hr-emp-lst-prj-zul-recipe
Working version
Publish
More
Applied steps (5) | Clear all
1. Rename First_Name to FirstName
2. Rename Last_Name to LastName
3. Rename Date_of_Birth to DateOfBirth
4. Rename Phone_Number to PhoneNumber
5. Rename Employee_ID to EmployeeID
Zoom 100%
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Feb 5 12:03 PM
Weekly Activity #3 - BUSI 65
Projects | AWS Glue DataBrew
CC Week #3 - HR - Zul.draw
us-east-1.console.aws.amazon.com/databrew/home?region=us-east-1#projects
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
draw.io
CC Week #2 HR-M...
All Bookmarks
aws
Services
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Updated profile or dataset job "hr-fin-sup-lst-ds-prf-zul".
DATASETS
PROJECTS
RECIPES
DQ RULES
JOBS
WHAT'S NEW
DataBrew
Projects
Projects (3) Info
Open project
View job details
View lineage
Run job
Actions
Create project
Find projects
1
Project name
Associated dataset
Attached recipe
Jobs
Create date
Created by
In use by
Tags
hr-fin-sup-lst-prj-zul
hr-fin-sup-lst-ds-zul
hr-fin-sup-lst-prj-zul-recipe
-
15 minutes ago
February 5, 2025, 11:48:29 am
voclabs
voclabs/user3764972=zayerayz
-
hr-act-lst-prj-zul
hr-act-lst-ds-zul
hr-act-lst-prj-zul-recipe
-
25 minutes ago
February 5, 2025, 11:38:32 am
voclabs
voclabs/user3764972=zayerayz
-
hr-emp-lst-prj-zul
hr-emp-lst-ds-zul
hr-emp-lst-prj-zul-recipe
-
38 minutes ago
February 5, 2025, 11:25:08 am
voclabs
voclabs/user3764972=zayerayz
-
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Updated profile or dataset job "hr-fin-sup-lst-ds-prf-zul".

hr-fin-sup-lst-ds-zul

S3

Rerun profile | Create project with this dataset | Actions | JOB DETAILS | DOWNLOAD

Dataset preview | Data profile overview | Column statistics | Data quality rules | Data lineage

Last job run Succeeded **2 minutes ago**, no job runs scheduled

Data profile was run on **custom sample** of first **20,000 rows** of your dataset

Select profile to view: Job run 1 | February 5, 2025, 11:52:54 am

Summary

TOTAL ROWS
50

TOTAL COLUMNS
10

DATA TYPES

INTEGER
2 columns

ABC STRING
7 columns

MISSING CELLS
Missing values count of cells in profiled columns

VALID CELLS
450 100%

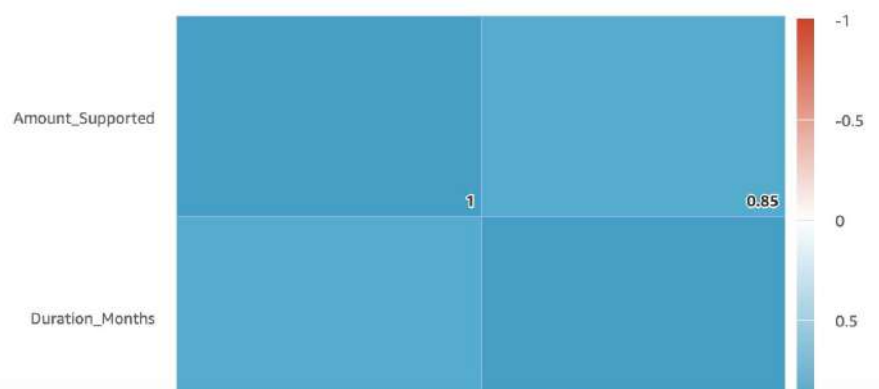
MISSING CELLS
0 0%

DUPLICATE ROWS

Correlations

View: All columns (2)

Correlation coefficient (r) defines how closely two variables are related. It ranges from -1.0 to +1.0 , where 0 means there is no relationship between the variables.



Updated profile or dataset job "hr-act-lst-ds-prf-zul".

hr-act-lst-ds-zul

S3

Rerun profile | Create project with this dataset | Actions | JOB DETAILS | DOWNLOAD

Dataset preview | Data profile overview | Column statistics | Data quality rules | Data lineage

Last job run Succeeded **3 minutes ago**, no job runs scheduled

Data profile was run on **custom sample** of first **20,000 rows** of your dataset

Select profile to view: Job run 1 | February 5, 2025, 11:43:25 am

Summary

TOTAL ROWS
50

TOTAL COLUMNS
10

DATA TYPES

# INTEGER	# DOUBLE	ABC STRING
2 columns	1 columns	6 columns

MISSING CELLS

Missing values count of cells in profiled columns

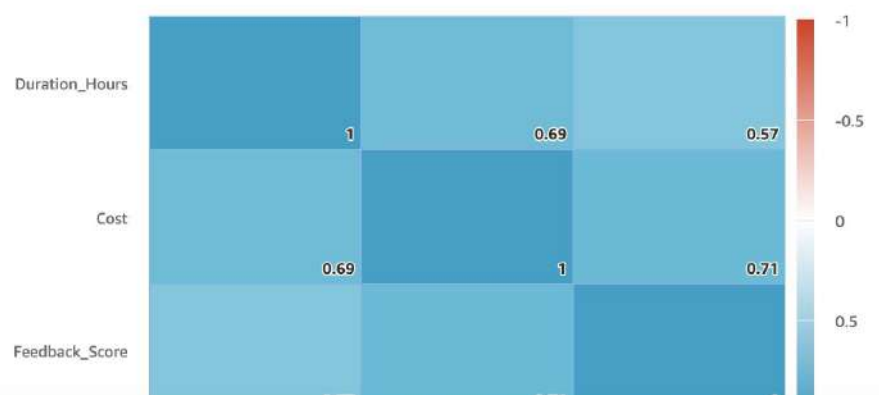
VALID CELLS	MISSING CELLS
450 100%	0 0%

DUPLICATE ROWS

Correlations

View: All columns (3)

Correlation coefficient (r) defines how closely two variables are related. It ranges from -1.0 to +1.0 , where 0 means there is no relationship between the variables.



Updated profile or dataset job "hr-emp-lst-ds-prf-zul".

hr-emp-lst-ds-zul

S3

Rerun profile | Create project with this dataset | Actions | JOB DETAILS | DOWNLOAD

Dataset preview | Data profile overview | Column statistics | Data quality rules | Data lineage

Last job run Succeeded **3 minutes ago**, no job runs scheduled

Data profile was run on **custom sample** of first **20,000 rows** of your dataset

Select profile to view: Job run 1 | February 5, 2025, 11:31:25 am

Summary

TOTAL ROWS
46

TOTAL COLUMNS
10

DATA TYPES

INTEGER
1 columns

ABC STRING
8 columns

MISSING CELLS
Missing values count of cells in profiled columns

VALID CELLS
407 98%

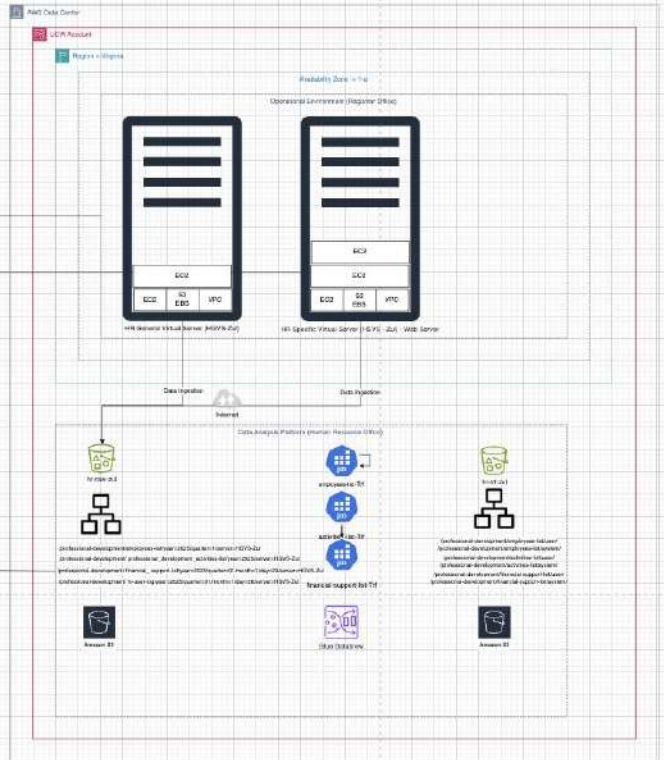
MISSING CELLS
7 2%

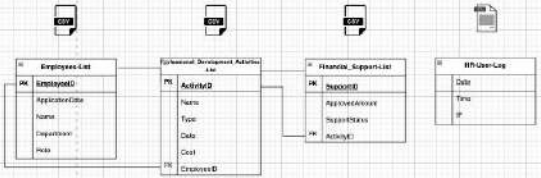
DUPLICATE ROWS

Correlations

View: All columns (1)

Correlation coefficient (r) defines how closely two variables are related. It ranges from -1.0 to +1.0 , where 0 means there is no relationship between the variables.



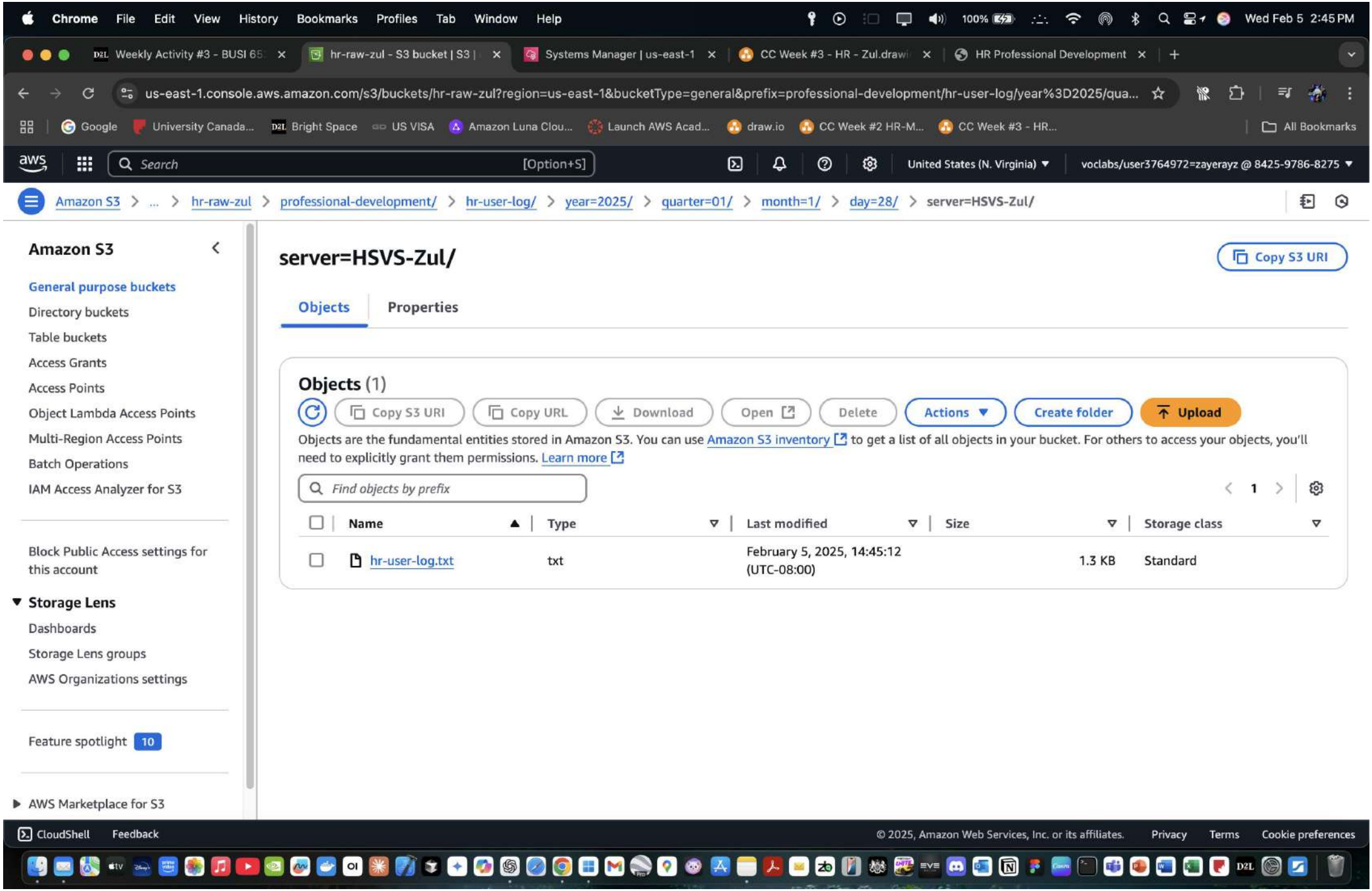


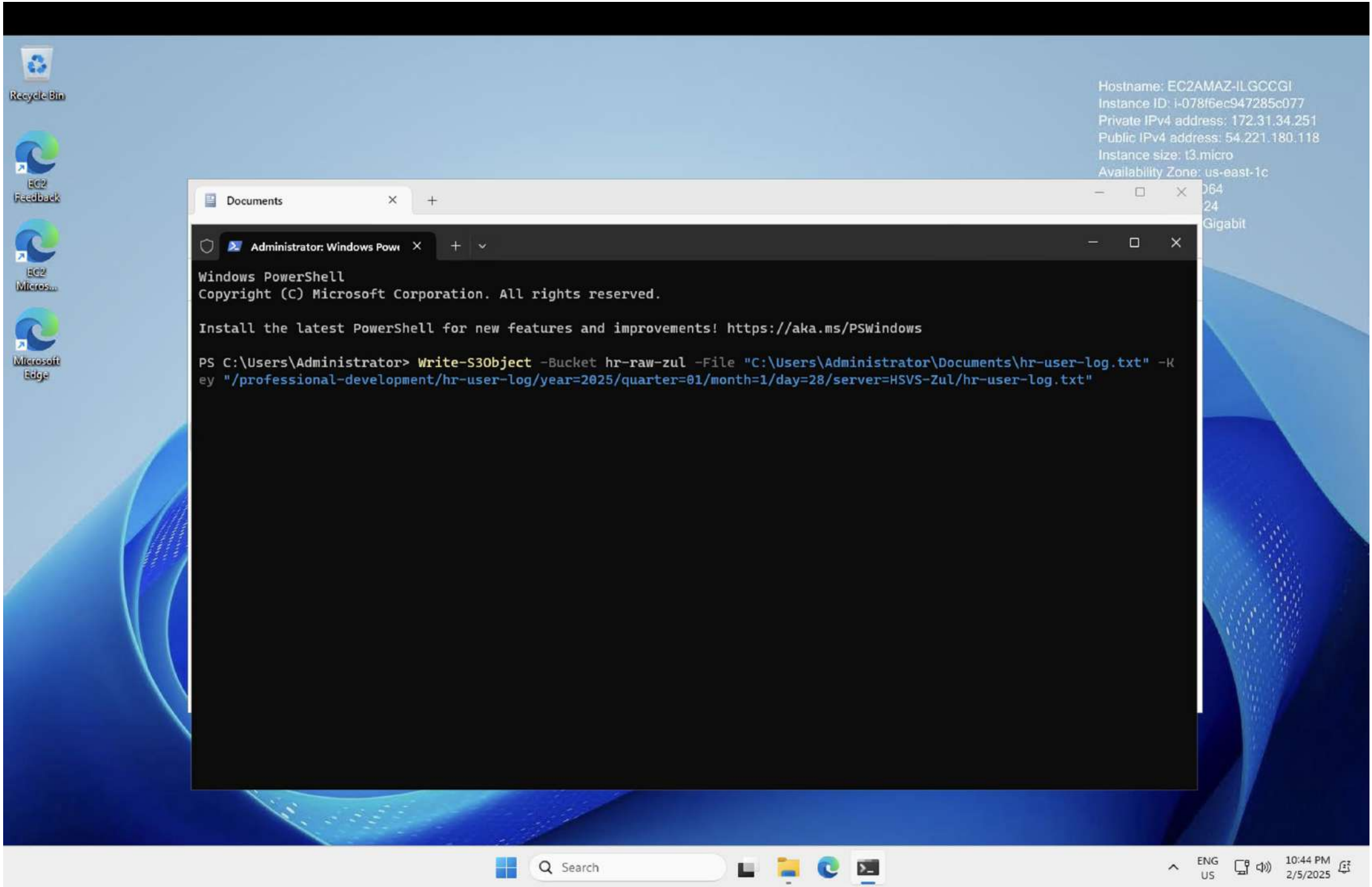
Dataset Quality Analysis

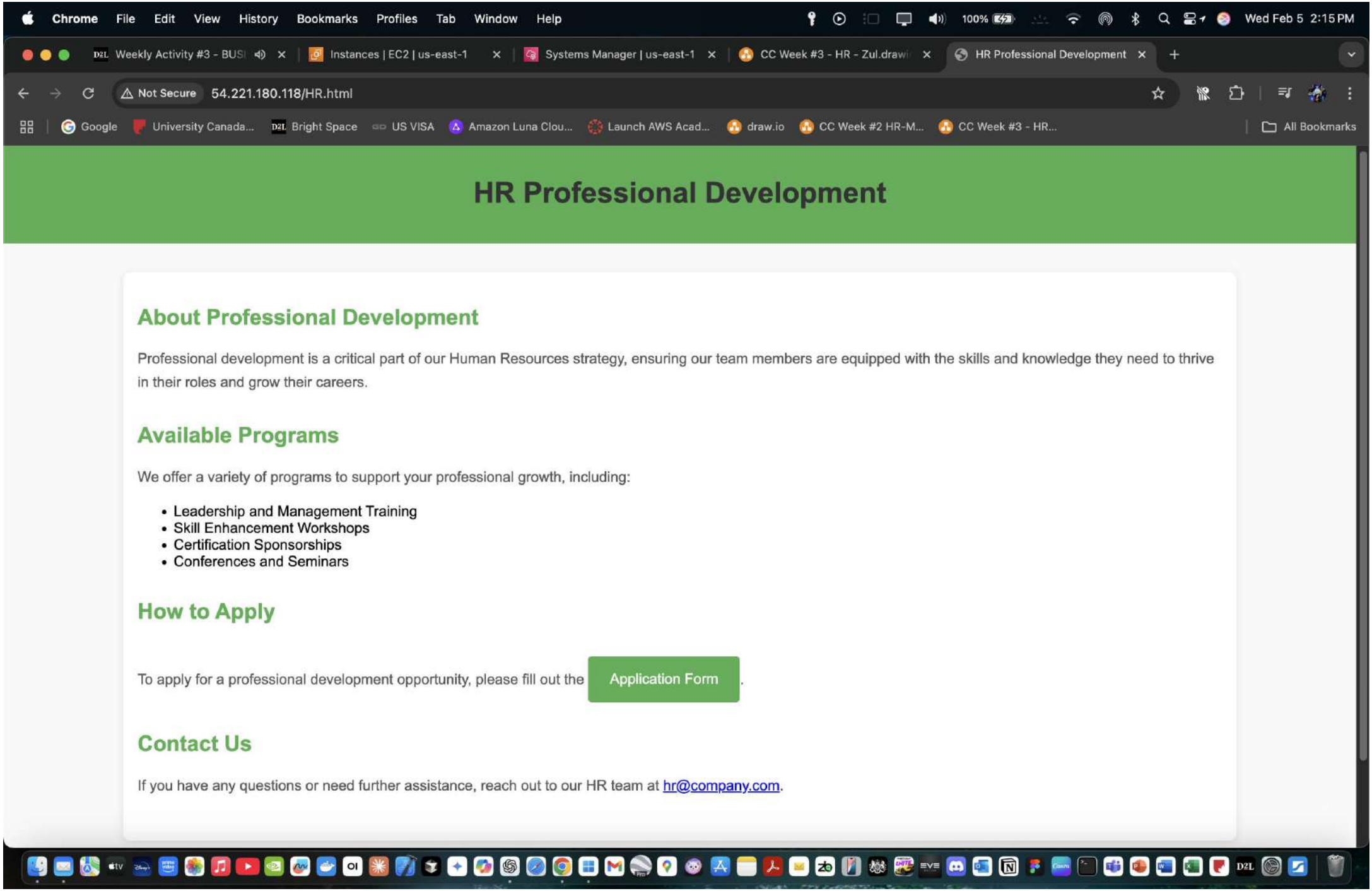
	Invalid Value	Incorrect Value	Outdated Value	Outlier Value	Missing Value	Type Issue	Granularity Issue	Encoding Issue	Incorrect Datatype
Employees-List									
Activities-List									
Financial_Support-List									
HR-User-Log									

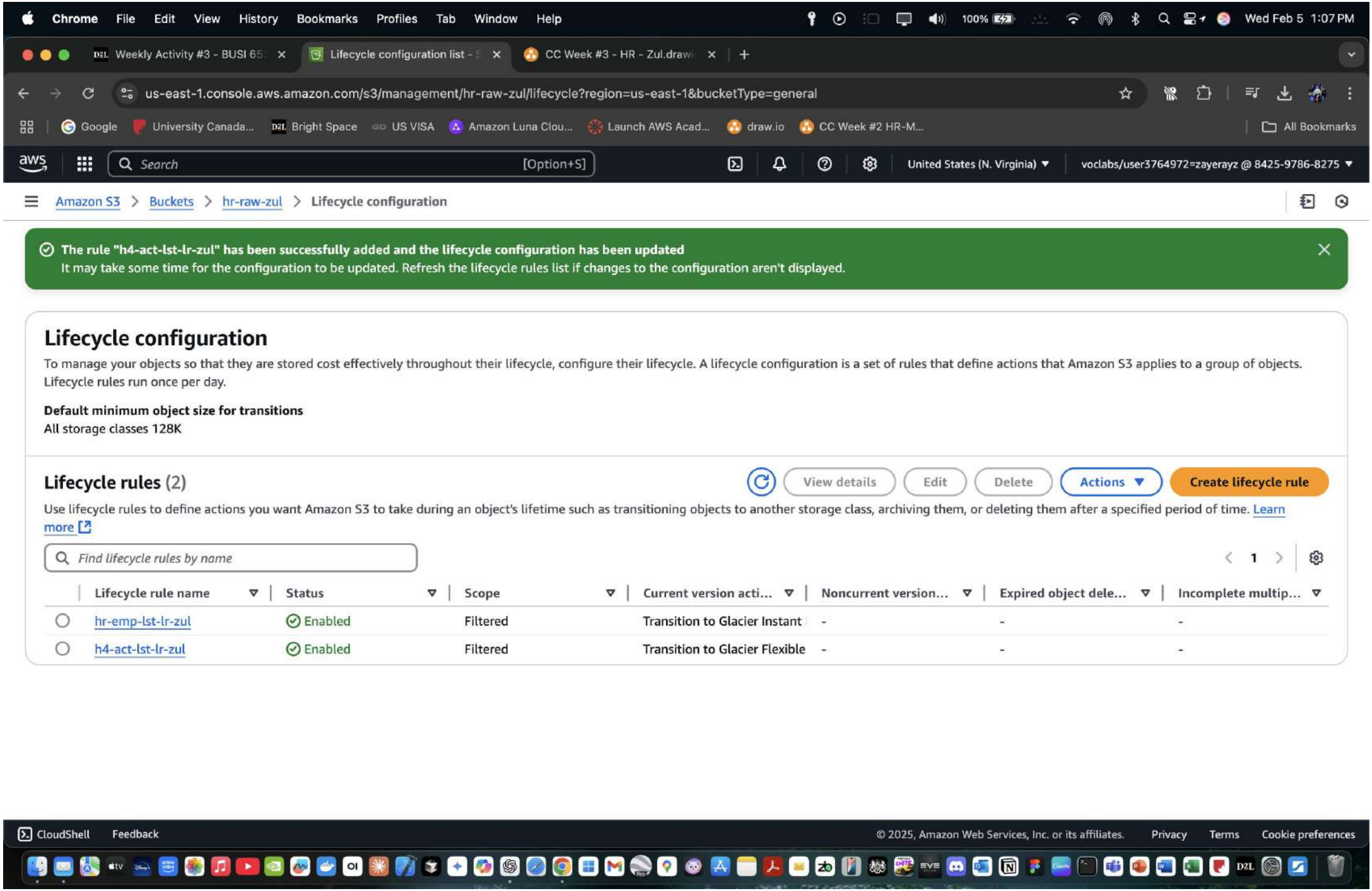
Data Ingestion Cost Analysis

	Storage Class	Object Size	# of Read Requests	# of Write Requests	Total Amount	Inbound Data Transfer	Outbound Data Transfer
Employees-List	Standard Infrequent Access One Zone-Infrequent Access Glacier Glacier Deep Archive Intelligent	4.3 KB	Injector	0	4.3 KB	4.3 KB	0
Activities-List							
Financial_Support-List							
HR-User-Log							


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Feb 5 2:45 PM
Weekly Activity #3 - BUSI 65
hr-raw-zul - S3 bucket | S3 |
Systems Manager | us-east-1
CC Week #3 - HR - Zul.draw
HR Professional Development
us-east-1.console.aws.amazon.com/s3/buckets/hr-raw-zul?region=us-east-1&bucketType=general&prefix=professional-development/hr-user-log/year%3D2025/qua...
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
draw.io
CC Week #2 HR-M...
CC Week #3 - HR...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > ... > hr-raw-zul > professional-development/ > hr-user-log/ > year=2025/ > quarter=01/ > month=1/ > day=28/ > server=HSVS-Zul/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 10
AWS Marketplace for S3
server=HSVS-Zul/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
hr-user-log.txt
txt
February 5, 2025, 14:45:12 (UTC-08:00)
1.3 KB
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Recycle Bin
EC2 Feedback
EC2 Micros...
Microsoft Edge
Hostname: EC2AMAZ-ILGCCGI
Instance ID: i-078f6ec947285c077
Private IPv4 address: 172.31.34.251
Public IPv4 address: 54.221.180.118
Instance size: t3.micro
Availability Zone: us-east-1c
Documents
Administrator: Windows Powe
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\Administrator> Write-S3Object -Bucket hr-raw-zul -File "C:\Users\Administrator\Documents\hr-user-log.txt" -Key "/professional-development/hr-user-log/year=2025/quarter=01/month=1/day=28/server=HSVS-Zul/hr-user-log.txt"
Search
ENG
US
10:44 PM
2/5/2025


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Feb 5 2:15 PM
Weekly Activity #3 - BUSI
Instances | EC2 | us-east-1
Systems Manager | us-east-1
CC Week #3 - HR - Zul.drawi
HR Professional Development
Not Secure
54.221.180.118/HR.html
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
draw.io
CC Week #2 HR-M...
CC Week #3 - HR...
All Bookmarks
HR Professional Development
About Professional Development
Professional development is a critical part of our Human Resources strategy, ensuring our team members are equipped with the skills and knowledge they need to thrive in their roles and grow their careers.
Available Programs
We offer a variety of programs to support your professional growth, including:
Leadership and Management Training
Skill Enhancement Workshops
Certification Sponsorships
Conferences and Seminars
How to Apply
To apply for a professional development opportunity, please fill out the Application Form .
Contact Us
If you have any questions or need further assistance, reach out to our HR team at hr@company.com.


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Feb 5 1:07 PM
Weekly Activity #3 - BUSI 65
Lifecycle configuration list -
CC Week #3 - HR - Zul.draw
us-east-1.console.aws.amazon.com/s3/management/hr-raw-zul/lifecycle?region=us-east-1&bucketType=general
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
draw.io
CC Week #2 HR-M...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Amazon S3 > Buckets > hr-raw-zul > Lifecycle configuration
The rule "h4-act-lst-lr-zul" has been successfully added and the lifecycle configuration has been updated
It may take some time for the configuration to be updated. Refresh the lifecycle rules list if changes to the configuration aren't displayed.
Lifecycle configuration
To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their lifecycle. A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. Lifecycle rules run once per day.
Default minimum object size for transitions
All storage classes 128K
Lifecycle rules (2)
View details
Edit
Delete
Actions
Create lifecycle rule
Use lifecycle rules to define actions you want Amazon S3 to take during an object's lifetime such as transitioning objects to another storage class, archiving them, or deleting them after a specified period of time. Learn more
Find lifecycle rules by name
1
Lifecycle rule name
Status
Scope
Current version acti...
Noncurrent version...
Expired object dele...
Incomplete multip...
hr-emp-lst-lr-zul
Enabled
Filtered
Transition to Glacier Instant
-
-
-
h4-act-lst-lr-zul
Enabled
Filtered
Transition to Glacier Flexible
-
-
-
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

AWS

Public DNS
ec2-34-205-63-68.compute-1.amazonaws.com

Username Info
Administrator

Password

?.h8UPSWHR3fAOMr%RzQX%JP&w32Yro

write-S3Object -Bucket hr-raw-zul -File "C:\Users\Administrator\Documents\Employees-List.csv" -Key "professional-development/Employees-List/year=2025/quarter=1/server=RGVS-Mah/Employees-List.csv"

write-S3Object -Bucket hr-raw-zul -File "C:\Users\Administrator\Documents\Professional_Development_Activities.csv" -Key "professional-development/Professional_Development_Activities -List/professional_development_activities-list/year=2025/server=RGVS-Mah/Professional_Development_Activities.csv"

write-S3Object -Bucket hr-raw-zul -File "C:\Users\Administrator\Documents\Financial_Support-List.csv" -Key "professional-development/Financial_Support-List/year=2025/quarter=01/month=1/server=RGVS-Mah/Financial_Support-List.csv"

AWS

Public DNS
ec2-34-205-63-68.compute-1.amazonaws.com

Username Info
Administrator

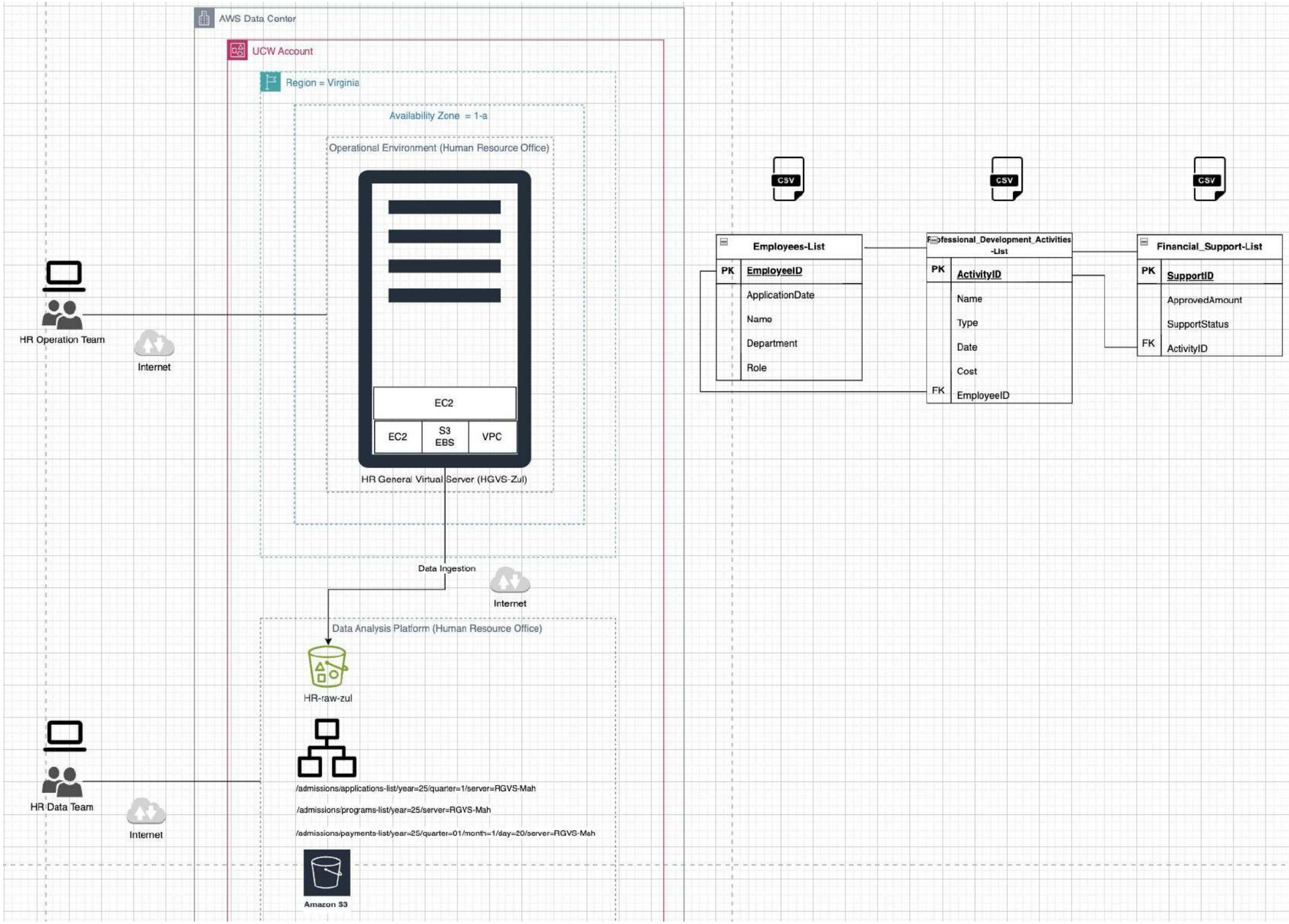
Password

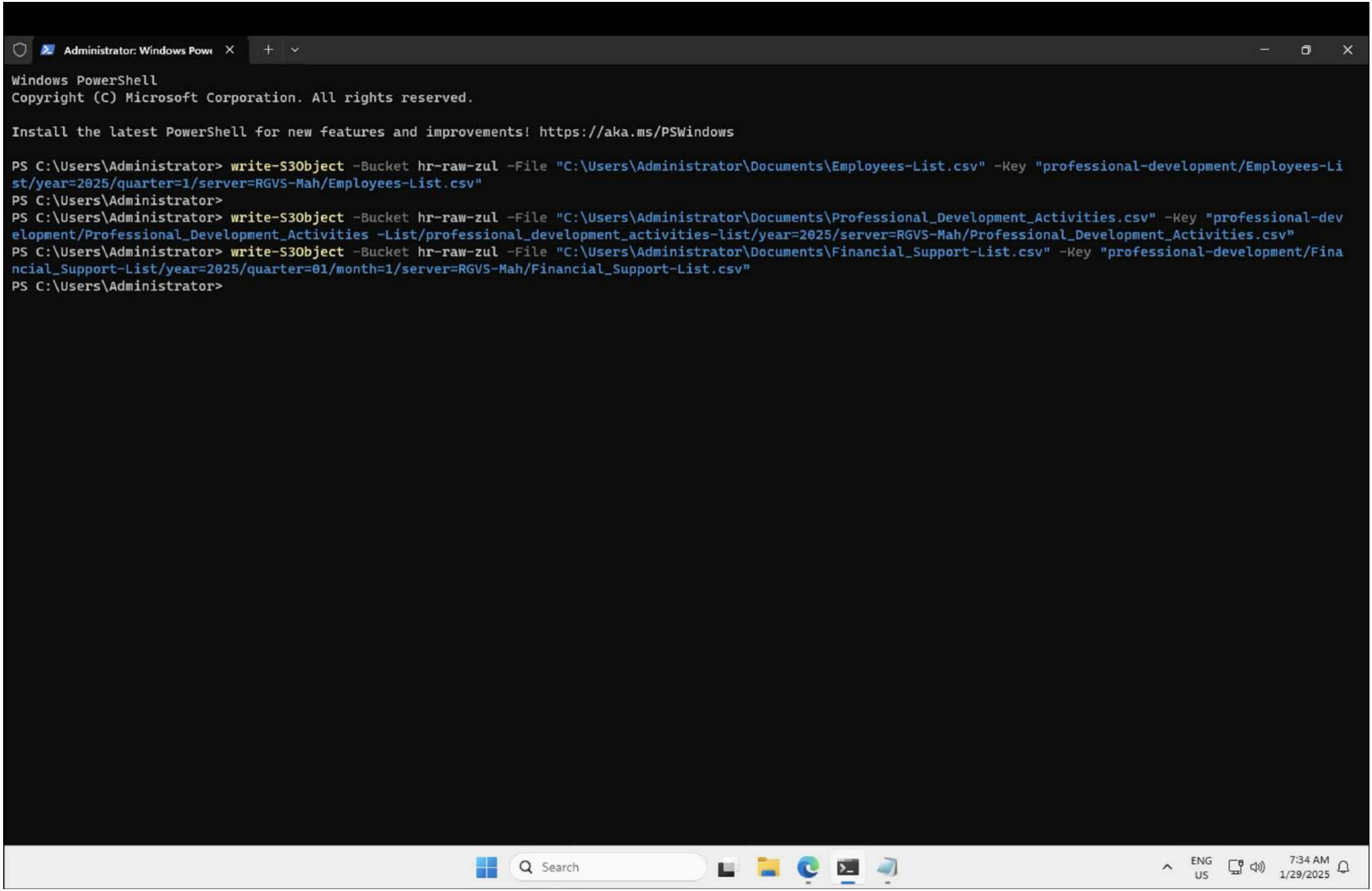
?.h8UPSWHR3fAOMr%RzQX%JP&w32Yro

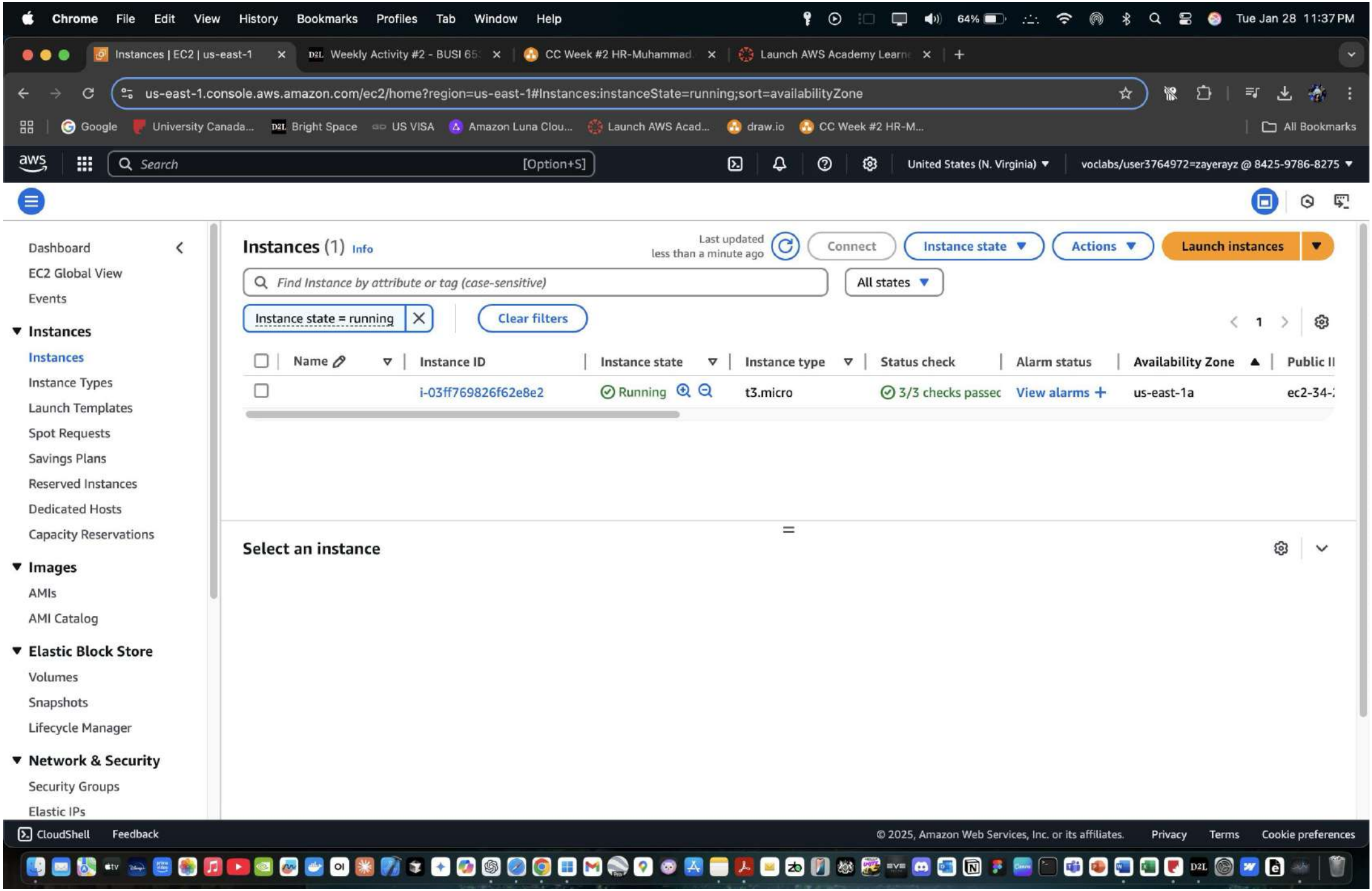
write-S3Object -Bucket hr-raw-zul -File "C:\Users\Administrator\Documents\Employees-List.csv" -Key "professional-development/Employees-List/year=2025/quarter=1/server=RGVS-Mah/Employees-List.csv"

write-S3Object -Bucket hr-raw-zul -File "C:\Users\Administrator\Documents\Professional_Development_Activities.csv" -Key "professional-development/Professional_Development_Activities -List/professional_development_activities-list/year=2025/server=RGVS-Mah/Professional_Development_Activities.csv"

write-S3Object -Bucket hr-raw-zul -File "C:\Users\Administrator\Documents\Financial_Support-List.csv" -Key "professional-development/Financial_Support-List/year=2025/quarter=01/month=1/server=RGVS-Mah/Financial_Support-List.csv"


AWS Data Center
UCW Account
Region = Virginia
Availability Zone = 1-a
Operational Environment (Human Resource Office)
EC2
EC2
S3
EBS
VPC
HR General Virtual Server (HGVS-Zul)
HR Operation Team
Internet
Data Ingestion
Internet
Data Analysis Platform (Human Resource Office)
HR-raw-zul
HR Data Team
Internet
/admissions/applications-list/year=25/quarter=1/server=RGVS-Mah
/admissions/programs-list/year=25/server=RGVS-Mah
/admissions/payments-list/year=25/quarter=01/month=1/day=20/server=RGVS-Mah
Amazon S3
CSV
CSV
CSV
Employees-List
PK
EmployeeID
ApplicationDate
Name
Department
Role
Professional_Development_Activities-List
PK
ActivityID
Name
Type
Date
Cost
FK
EmployeeID
Financial_Support-List
PK
SupportID
ApprovedAmount
SupportStatus
FK
ActivityID


Administrator: Windows Pow
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\Administrator> write-S3Object -Bucket hr-raw-zul -File "C:\Users\Administrator\Documents\Employees-List.csv" -Key "professional-development/Employees-List/year=2025/quarter=1/server=RGVS-Mah/Employees-List.csv"
PS C:\Users\Administrator>
PS C:\Users\Administrator> write-S3Object -Bucket hr-raw-zul -File "C:\Users\Administrator\Documents\Professional_Development_Activities.csv" -Key "professional-development/Professional_Development_Activities -List/professional_development_activities-list/year=2025/server=RGVS-Mah/Professional_Development_Activities.csv"
PS C:\Users\Administrator> write-S3Object -Bucket hr-raw-zul -File "C:\Users\Administrator\Documents\Financial_Support-List.csv" -Key "professional-development/Financial_Support-List/year=2025/quarter=01/month=1/server=RGVS-Mah/Financial_Support-List.csv"
PS C:\Users\Administrator>
Search
ENG
US
7:34 AM
1/29/2025


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tue Jan 28 11:37 PM
Instances | EC2 | us-east-1
Weekly Activity #2 - BUSI 65
CC Week #2 HR-Muhammad
Launch AWS Academy Learne
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:instanceState=running;sort=availabilityZone
Google
University Canada...
Bright Space
US VISA
Amazon Luna Clou...
Launch AWS Acad...
draw.io
CC Week #2 HR-M...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
voclabs/user3764972=zayerayz @ 8425-9786-8275
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Instances (1) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Instance state = running
Clear filters
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public II
i-03ff769826f62e8e2
Running
t3.micro
3/3 checks passec
View alarms +
us-east-1a
ec2-34-
Select an instance
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Team	ID	Priority
G1	BP 1.1	Required
G1	BP 2.4	Recommended

Best Practice
Validate the data quality of source systems before transferring data for analytics.
Build standard operating procedures for deployment, test, rollback, and backfill tasks.

Evaluation (0: Not implemented, 1: Partially Implemented, 2: Implemented)
2
0

Recommendation for the next semster	Presenter (Team Member)
Draft and document everything rollback and backfill scenarios.	

Team	ID	Priority
G1	BP 3.5	Required
G1	BP 4.2	Required
G1	BP 5.2	Required

Best Practice
Record data classifications into the Data Catalog so that analytics workload can understand.
Build user identity solutions that uniquely identify people and systems.
Implement least privilege policies for source and downstream systems.

Evaluation (0: Not implemented, 1: Partially Implemented, 2: Implemented)
2
2
2

Recommendation for the next semster	Presenter (Team Member)

Team	ID	Priority
G1	BP 6.3	Required
G1	BP 7.3	Required

Best Practice
Notify stakeholders about analytics or ETL job failures.
Trace data lineage.

Evaluation (0: Not implemented, 1: Partially Implemented, 2: Implemented)
0
2

Recommendation for the next semster	Presenter (Team Member)
Notify stakeholders about analytics or ETL job failures.	

Team	ID	Priority
G1	BP 9.1	Highly recommended
G1	BP 10.3	Recommended

Best Practice
Identify critical performance criteria for your storage workload.
Utilize compression techniques to both decrease storage requirements and enhance I/O efficiency.

Evaluation (0: Not implemented, 1: Partially Implemented, 2: Implemented)
0
2

Recommendation for the next semster	Presenter (Team Member)
Identify critical performance criteria for your storage workload.	

Team	ID	Priority
G1	BP 11.4	Recommended
G1	BP 13.1	Recommended

Best Practice
Use auto scaling where appropriate.
Remove unused data and infrastructure.

Evaluation (0: Not implemented, 1: Partially Implemented, 2: Implemented)
0
0

Recommendation for the next semster	Presenter (Team Member)
Implement auto scaling for compute resources such as EC2, ECS, or EMR based on workload metr	

ics using CloudWatch.

Team	ID	Priority
G1	BP 15.1	Recommended
Class	BP 15.6	Recommended
Class	BP 15.7	Recommended

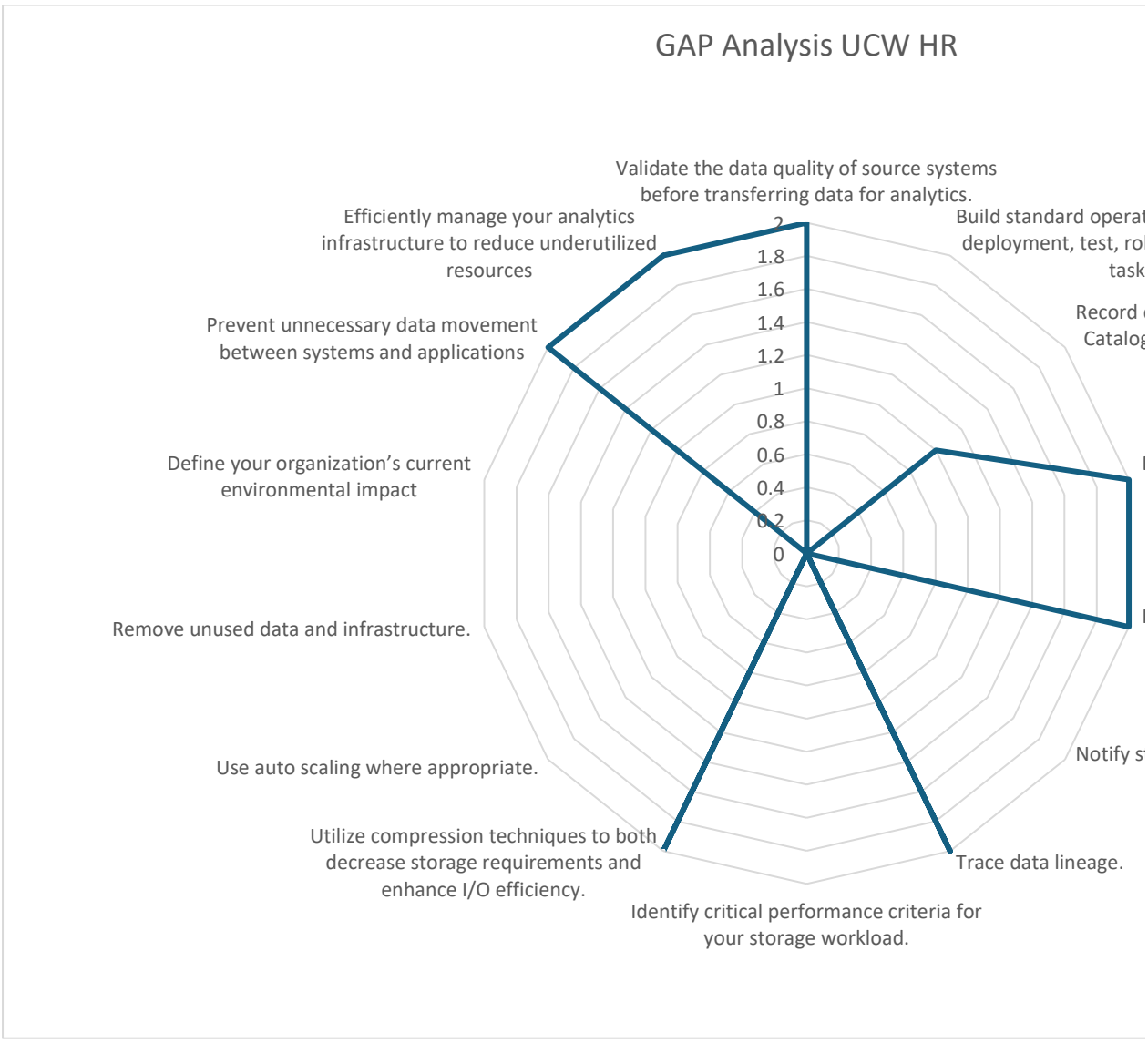
Best Practice
Define your organization’s current environmental impact
Prevent unnecessary data movement between systems and applications
Efficiently manage your analytics infrastructure to reduce underutilized resources

Evaluation (0: Not implemented, 1: Partially Implemented, 2: Implemented)
0
2
2

Recommendation for the next semster	Presenter (Team Member)
UCW should begin tracking its environmental impact by leveraging AWS sustainability tools such as t	

:he AWS Customer Carbon Footprint Tool.0

Validate the data quality of source systems before transferring data for analytic	2
Build standard operating procedures for deployment, test, rollback, and backfill	0
Record data classifications into the Data Catalog so that analytics workload car	1
Build user identity solutions that uniquely identify people and systems.	2
Implement least privilege policies for source and downstream systems.	2
Notify stakeholders about analytics or ETL job failures.	0
Trace data lineage.	2
Identify critical performance criteria for your storage workload.	0
Utilize compression techniques to both decrease storage requirements and enh	2
Use auto scaling where appropriate.	0
Remove unused data and infrastructure.	0
Define your organization’s current environmental impact	0
Prevent unnecessary data movement between systems and applications	2
Efficiently manage your analytics infrastructure to reduce underutilized resource	2


GAP Analysis UCW HR
Validate the data quality of source systems before transferring data for analytics.
Build standard operat
deployment, test, ro
task
Record
Catalog
Notify s
Trace data lineage.
Identify critical performance criteria for your storage workload.
Utilize compression techniques to both decrease storage requirements and enhance I/O efficiency.
Use auto scaling where appropriate.
Remove unused data and infrastructure.
Define your organization’s current environmental impact
Prevent unnecessary data movement between systems and applications
Efficiently manage your analytics infrastructure to reduce underutilized resources
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0

ting procedures for
llback, and backfill
:s.

data classifications into the Data
g so that analytics workload can
understand.

Build user identity solutions that uniquely
identify people and systems.

Implement least privilege policies for
source and downstream systems.

takeholders about analytics or ETL
job failures.

GAP Analysis for UCW HR

Muhammad Zulqarnain Shahzad (2306553)

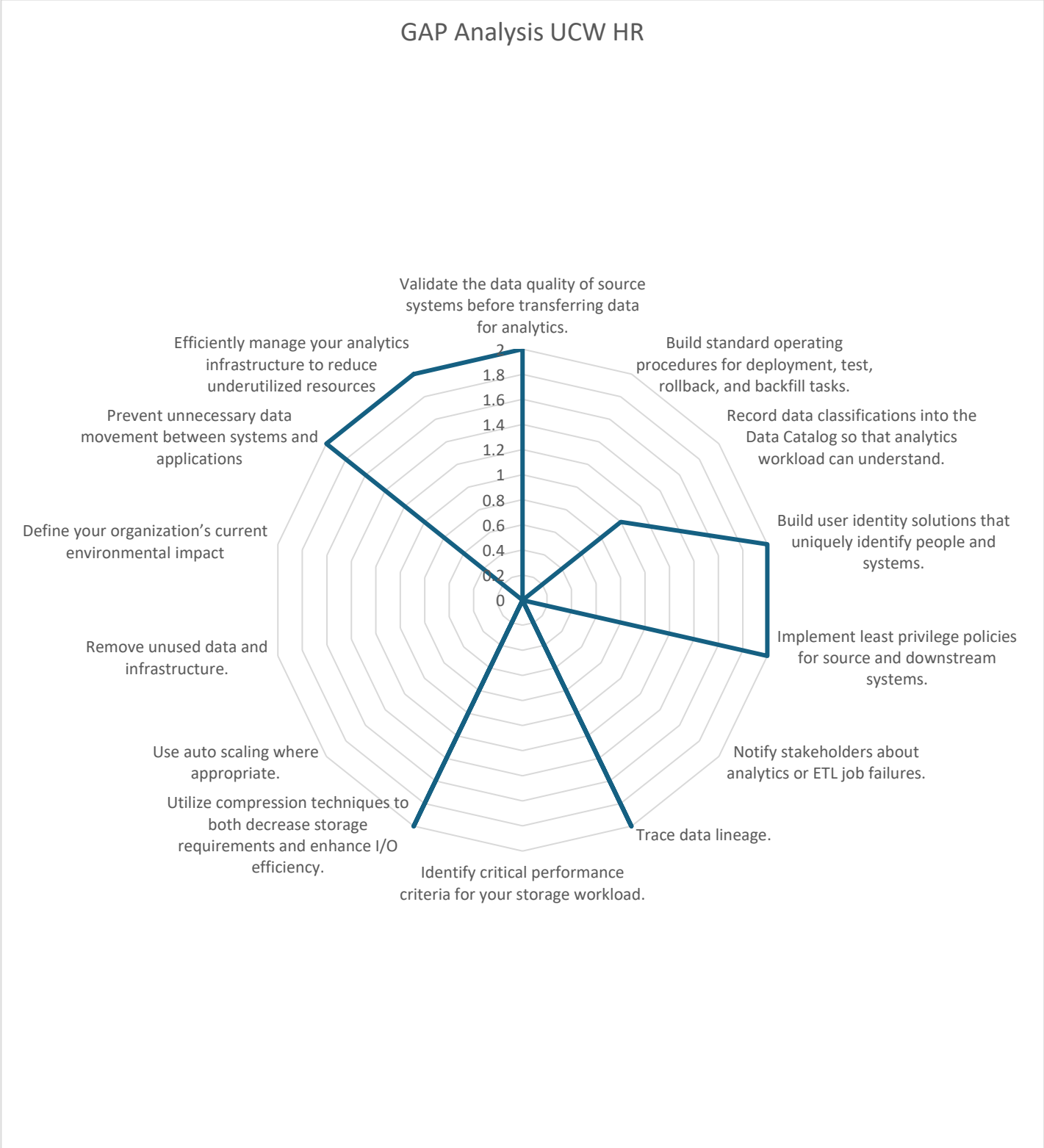
University Canada West

BUSI 653: Cloud Computing Technologies

Professor. Mahmood Mortazavi Dehkordi

March 20, 2025

Fig: 1 Gap Analysis Radar Chart for UCW HR



Note. This graph was prepared by the authors using the excel.

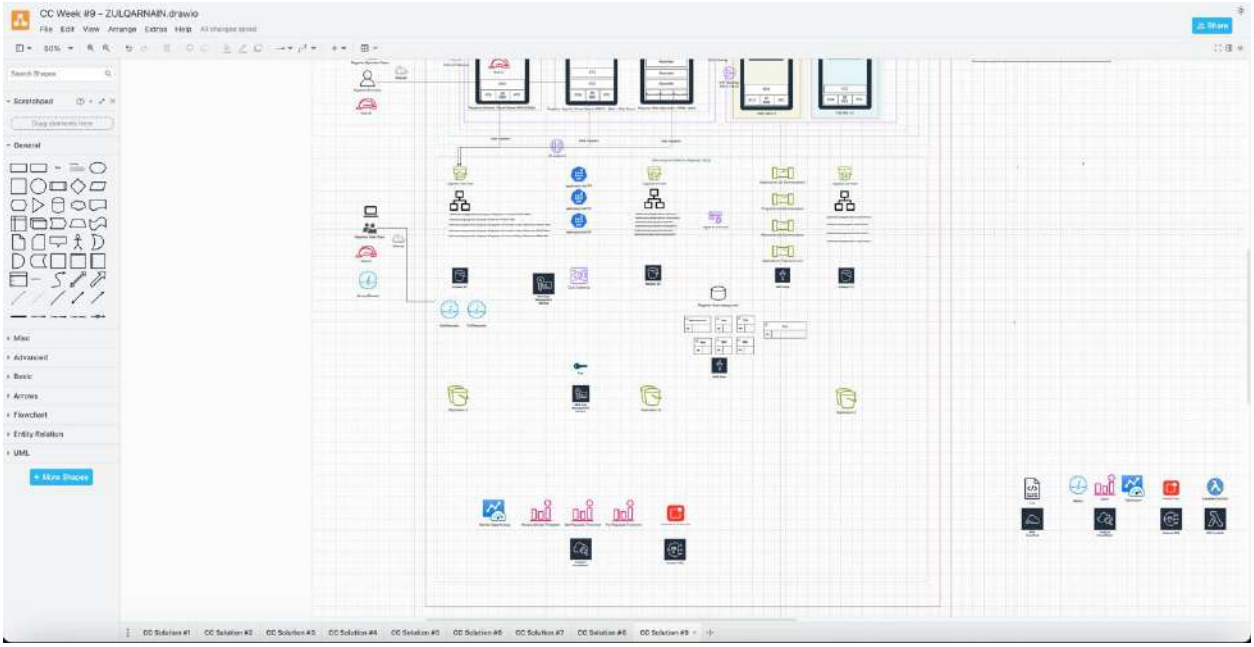
Muhammad Zulqarnain Shahzad (2306553)

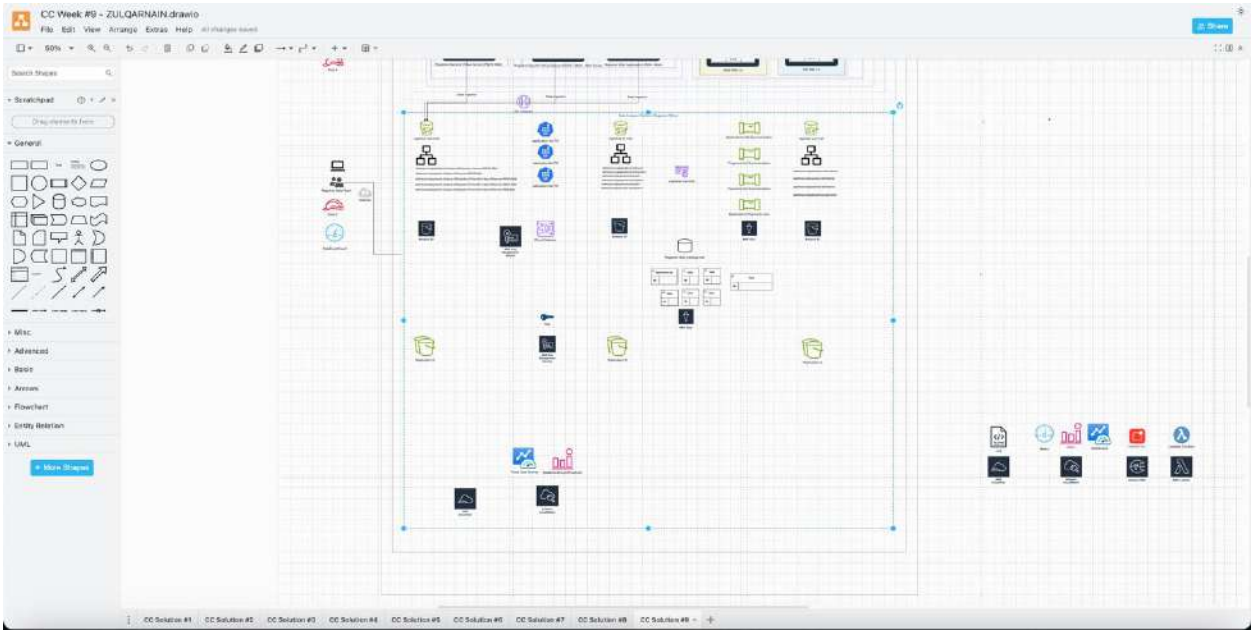
University Canada West

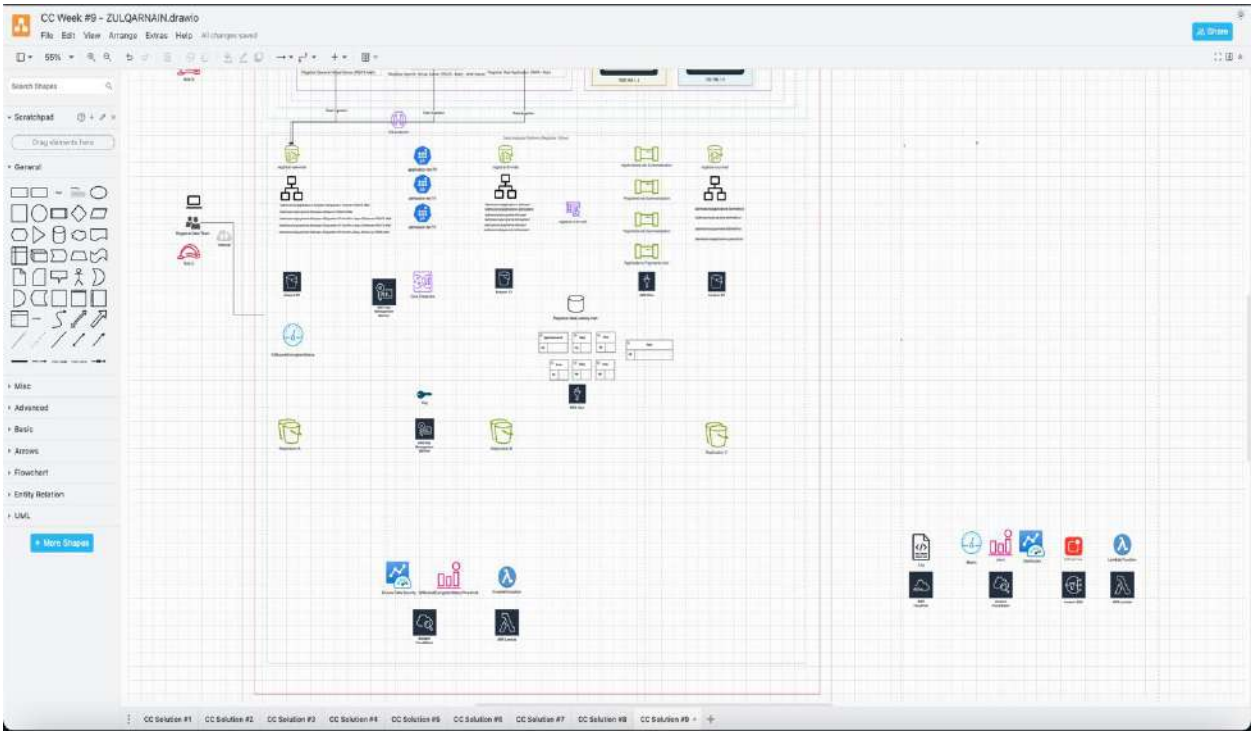
BUSI 653: Cloud Computing Technologies

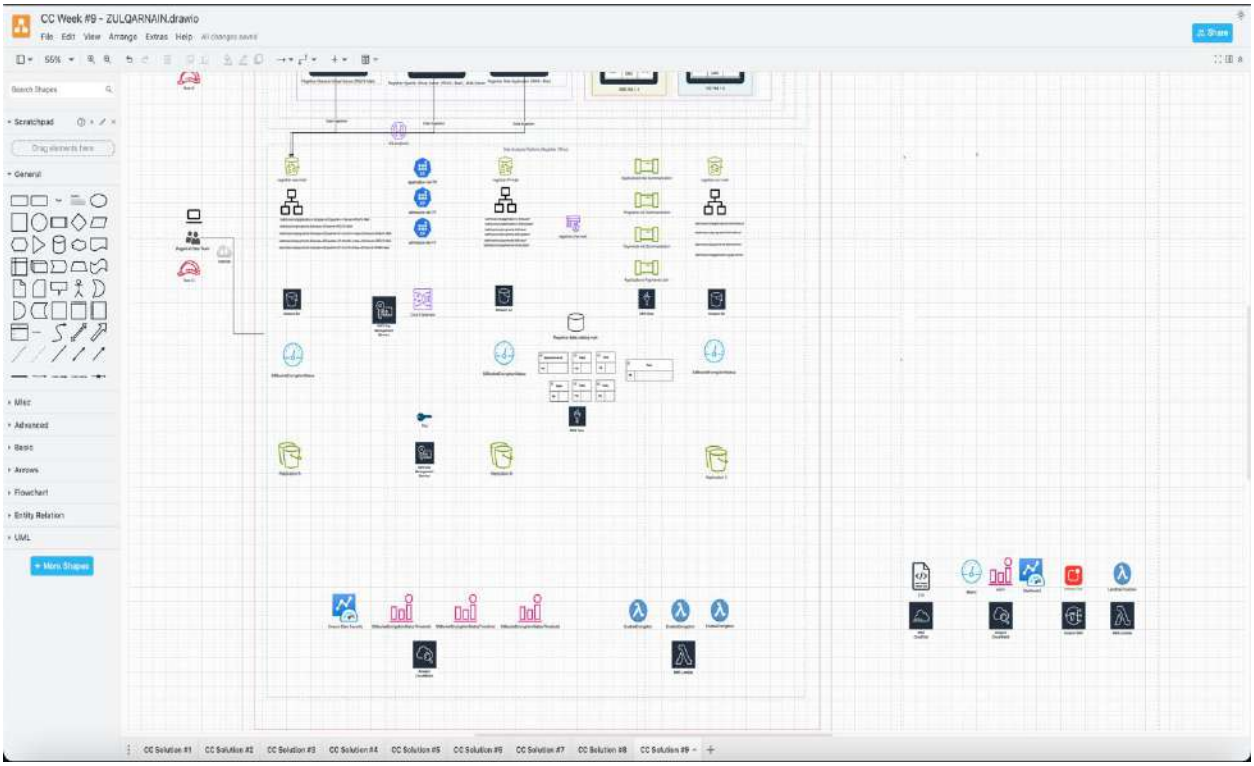
Professor. Mahmood Mortazavi Dehkordi

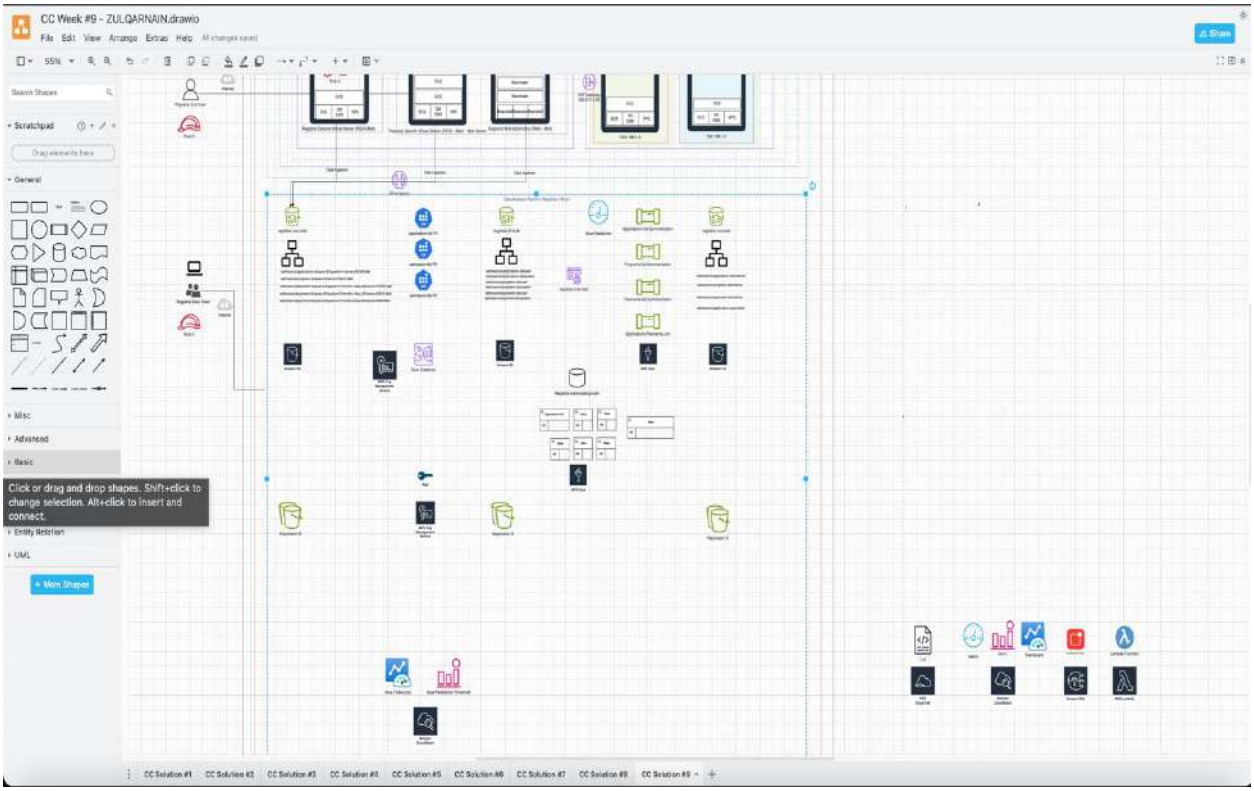
March 13, 2025

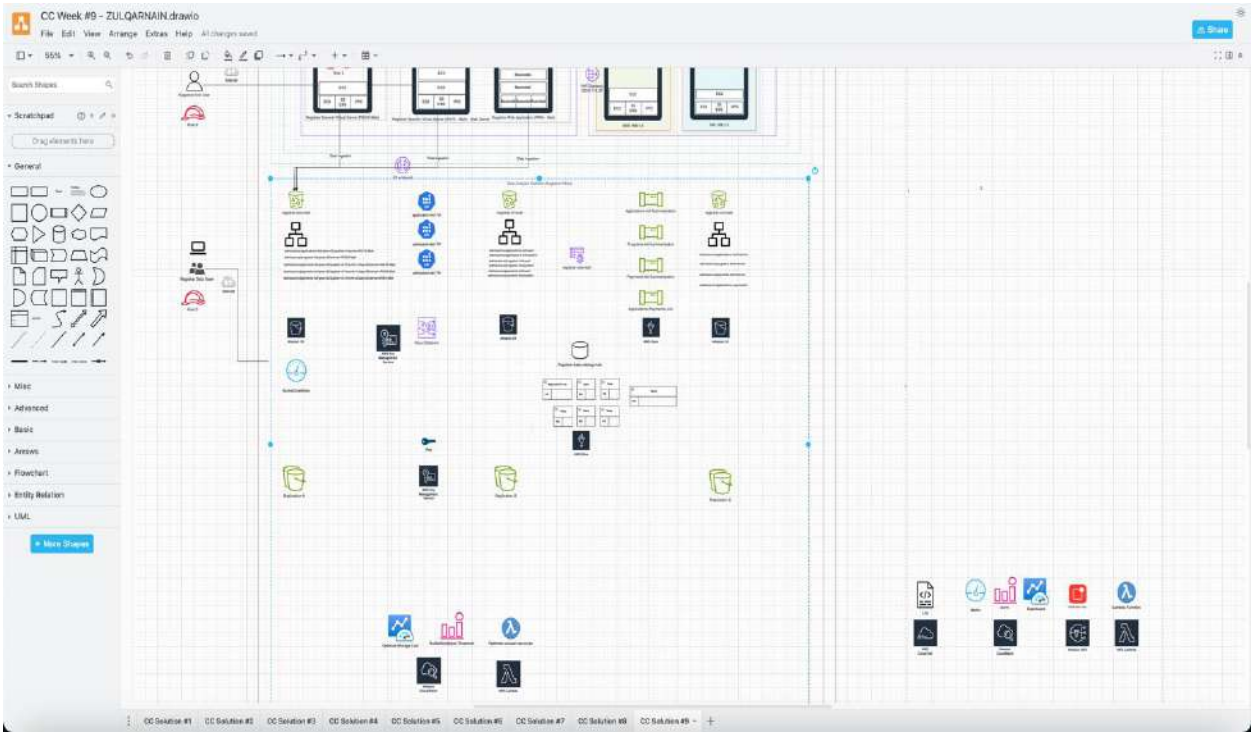


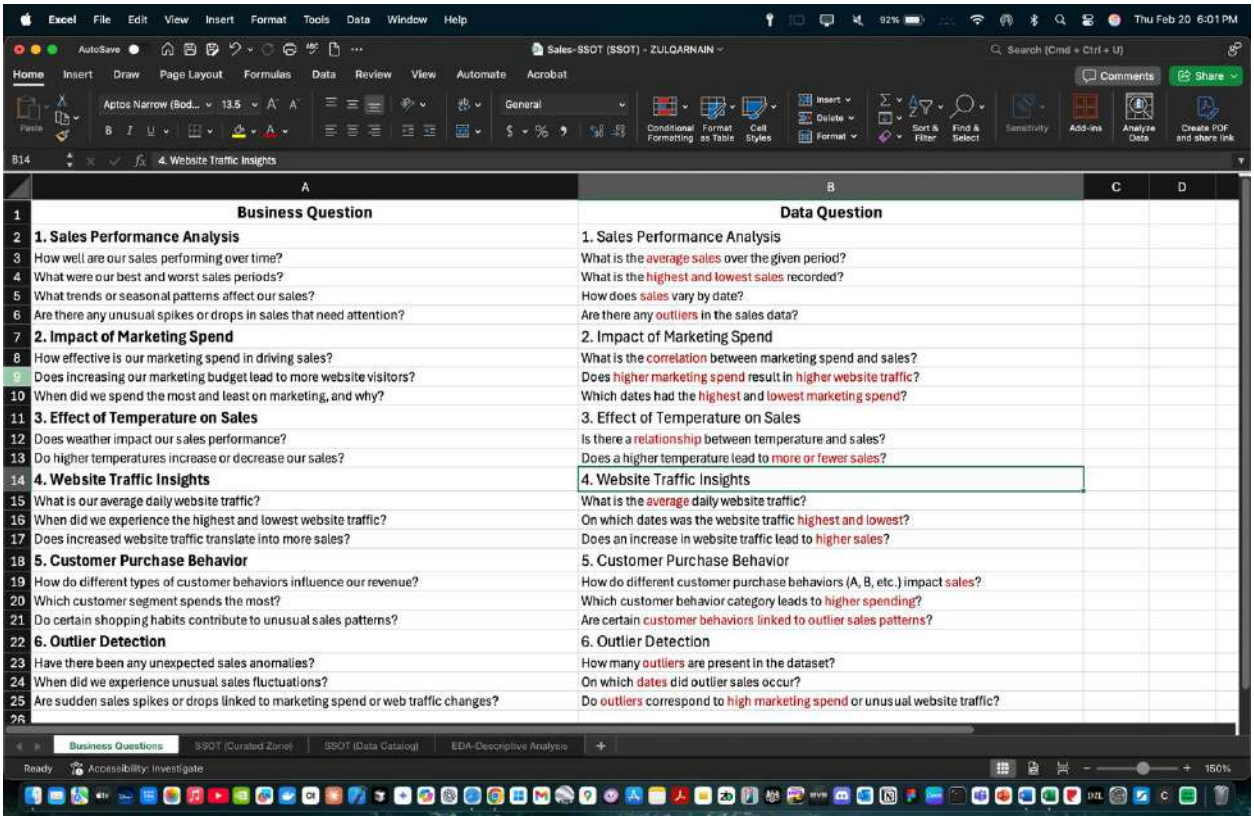



CC Week #9 - ZULQARNAIN.drawio
File Edit View Arrange Extras Help
Share
Search Shapes
Scratchpad
General
Misc
Advanced
Basic
Arrows
Flowchart
Entity Relation
UML
+ More Shapes
CC Solution #1
CC Solution #2
CC Solution #3
CC Solution #4
CC Solution #5
CC Solution #6
CC Solution #7
CC Solution #8
CC Solution #9

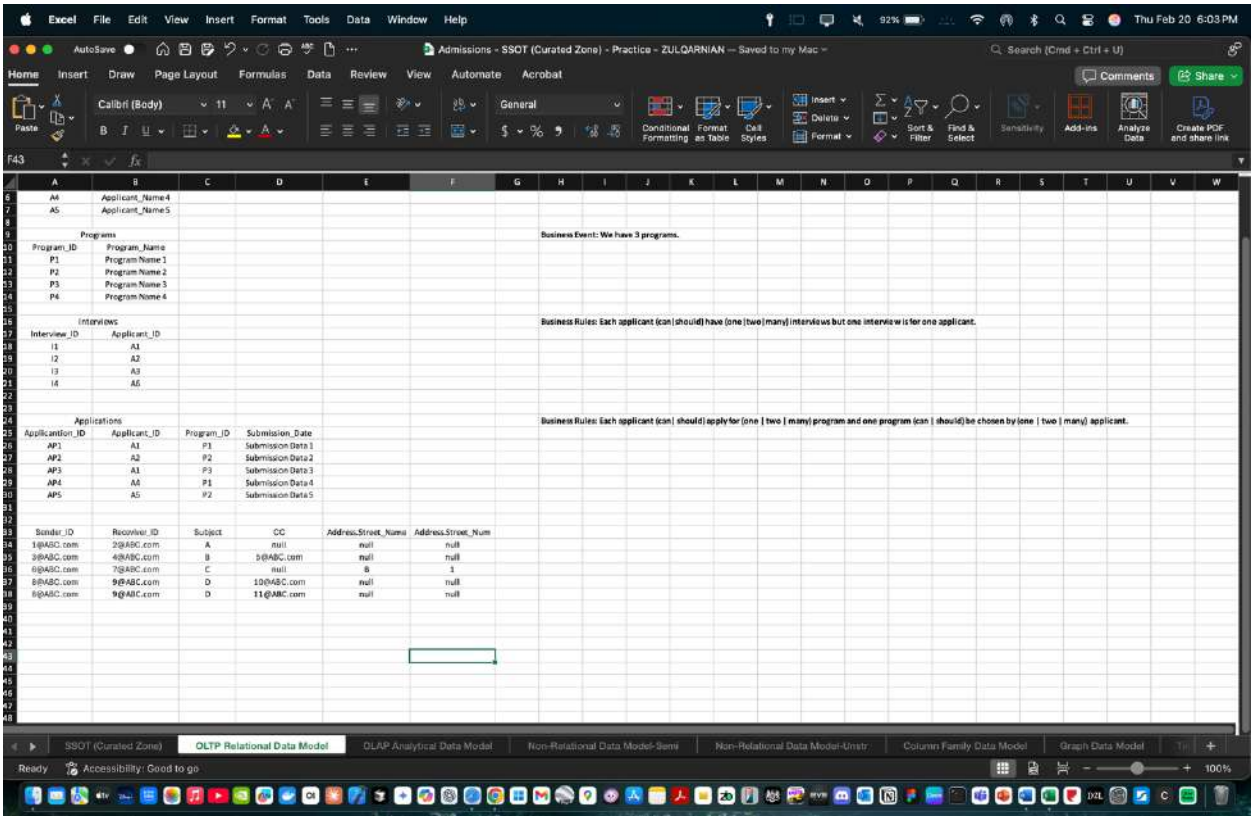

CC Week #9 - ZULQARNAIN.drawio
File Edit View Arrange Extras Help
Share
Search Shapes
Scratchpad
General
Misc
Advanced
Basic
Arrows
Flowchart
Entity Relation
UML
+ More Shapes
CC Solution #1
CC Solution #2
CC Solution #3
CC Solution #4
CC Solution #5
CC Solution #6
CC Solution #7
CC Solution #8
CC Solution #9

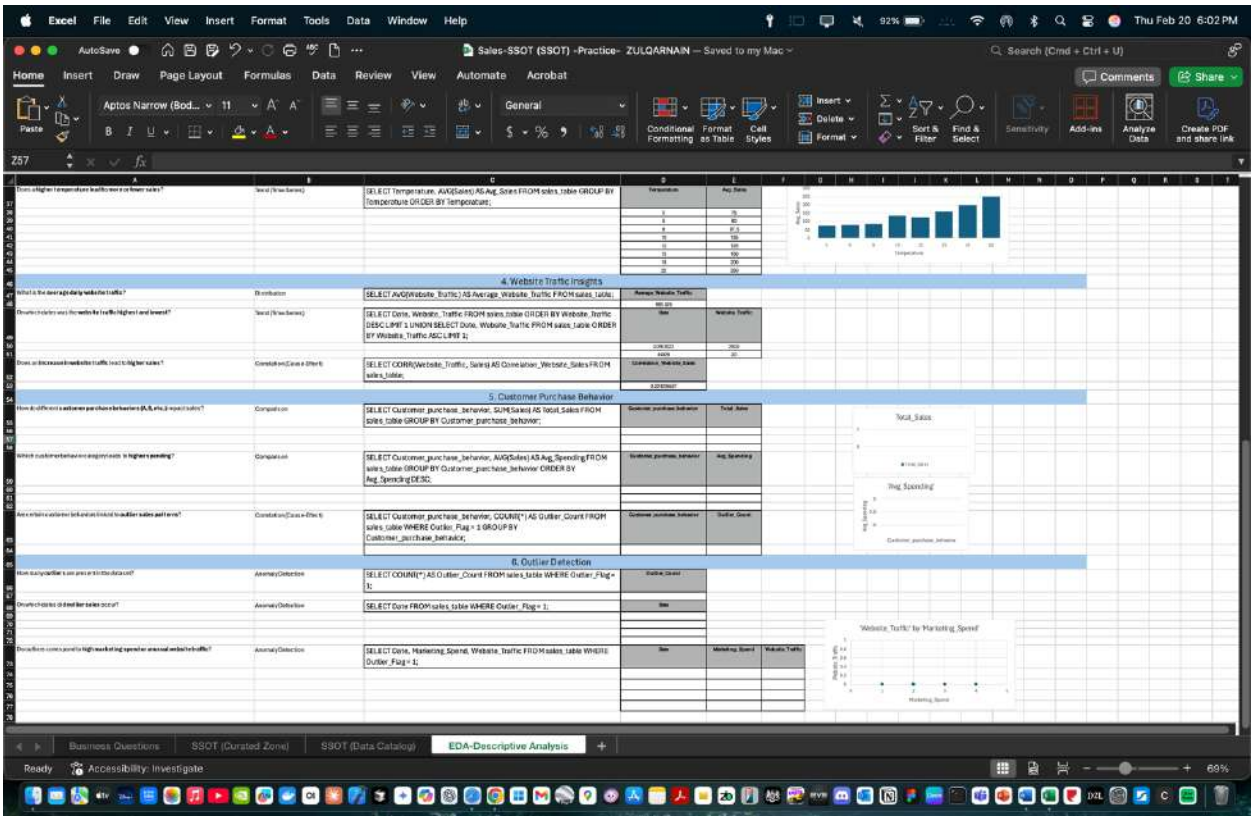

CC Week #9 - ZULQARNAIN.drawio
File Edit View Arrange Extras Help
Share
Search Shapes
Scratchpad
General
Misc
Advanced
Basic
Click or drag and drop shapes. Shift+click to change selection. Alt+click to insert and connect.
Entity Relation
UML
+ More Shapes


CC Week #9 - ZULQARNAIN.drawio
File Edit View Arrange Extras Help
Share
Search Shapes
Scratchpad
General
Misc
Advanced
Basic
Arrows
Flowchart
Entity Relation
UML
+ More Shapes



	A	B	C	D
1	**Business Question**	**Data Question**		
2	**1. Sales Performance Analysis**	1. Sales Performance Analysis		
3	How well are our sales performing over time?	What is the average sales over the given period?		
4	What were our best and worst sales periods?	What is the highest and lowest sales recorded?		
5	What trends or seasonal patterns affect our sales?	How does sales vary by date?		
6	Are there any unusual spikes or drops in sales that need attention?	Are there any outliers in the sales data?		
7	**2. Impact of Marketing Spend**	2. Impact of Marketing Spend		
8	How effective is our marketing spend in driving sales?	What is the correlation between marketing spend and sales?		
9	Does increasing our marketing budget lead to more website visitors?	Does higher marketing spend result in higher website traffic?		
10	When did we spend the most and least on marketing, and why?	Which dates had the highest and lowest marketing spend?		
11	**3. Effect of Temperature on Sales**	3. Effect of Temperature on Sales		
12	Does weather impact our sales performance?	Is there a relationship between temperature and sales?		
13	Do higher temperatures increase or decrease our sales?	Does a higher temperature lead to more or fewer sales?		
14	**4. Website Traffic Insights**	4. Website Traffic Insights		
15	What is our average daily website traffic?	What is the average daily website traffic?		
16	When did we experience the highest and lowest website traffic?	On which dates was the website traffic highest and lowest?		
17	Does increased website traffic translate into more sales?	Does an increase in website traffic lead to higher sales?		
18	**5. Customer Purchase Behavior**	5. Customer Purchase Behavior		
19	How do different types of customer behaviors influence our revenue?	How do different customer purchase behaviors (A, B, etc.) impact sales?		
20	Which customer segment spends the most?	Which customer behavior category leads to higher spending?		
21	Do certain shopping habits contribute to unusual sales patterns?	Are certain customer behaviors linked to outlier sales patterns?		
22	**6. Outlier Detection**	6. Outlier Detection		
23	Have there been any unexpected sales anomalies?	How many outliers are present in the dataset?		
24	When did we experience unusual sales fluctuations?	On which dates did outlier sales occur?		
25	Are sudden sales spikes or drops linked to marketing spend or web traffic changes?	Do outliers correspond to high marketing spend or unusual website traffic?		
26				


Excel File Edit View Insert Format Tools Data Window Help
Thu Feb 20 6:03 PM
Admissions - SSOT (Curated Zone) - Practice - ZULQARNIAN — Saved to my Mac
Home Insert Draw Page Layout Formulas Data Review View Automate Acrobat
Comments Share
Calibri (Body) 11
General
Conditional Formatting
Format as Table
Cell Styles
Insert Delete Format
Sort & Filter
Find & Select
Sensitivity
Add-ins
Analyze Data
Create PDF and share link
F43
A4 Applicant_Name4
A5 Applicant_Name5
Programs
Program_ID Program_Name
P1 Program Name 1
P2 Program Name 2
P3 Program Name 3
P4 Program Name 4
Interviews
Interview_ID Applicant_ID
I1 A1
I2 A2
I3 A3
I4 A5
Applications
Applicantion_ID Applicant_ID Program_ID Submission_Date
AP1 A1 P1 Submission Data 1
AP2 A2 P2 Submission Data 2
AP3 A1 P3 Submission Data 3
AP4 A4 P1 Submission Data 4
AP5 A5 P2 Submission Data 5
Sender_ID Receiver_ID Subject CC Address.Street_Name Address.Street_Num
1@ABC.com 2@ABC.com A null null null
3@ABC.com 4@ABC.com B 5@ABC.com null null
6@ABC.com 7@ABC.com C null B 1
8@ABC.com 9@ABC.com D 10@ABC.com null null
8@ABC.com 9@ABC.com D 11@ABC.com null null
Business Event: We have 3 programs.
Business Rules: Each applicant (can | should) have (one | two | many) interviews but one interview is for one applicant.
Business Rules: Each applicant (can | should) apply for (one | two | many) program and one program (can | should) be chosen by (one | two | many) applicant.
SSOT (Curated Zone)
OLTP Relational Data Model
OLAP Analytical Data Model
Non-Relational Data Model-Semi
Non-Relational Data Model-Unstr
Column Family Data Model
Graph Data Model
Ready
Accessibility: Good to go
100%


Excel File Edit View Insert Format Tools Data Window Help
Thu Feb 20 6:02 PM
Sales-SSOT (SSOT) -Practice- ZULQARNAIN — Saved to my Mac
Home Insert Draw Page Layout Formulas Data Review View Automate Acrobat
Comments Share
Aptos Narrow (Bod... 11
General
Conditional Formatting
Format as Table
Cell Styles
Insert Delete Format
Sort & Filter
Find & Select
Sensitivity
Add-ins
Analyze Data
Create PDF and share link
Z57
SELECT Temperature, AVG(Sales) AS Avg_Sales FROM sales_table GROUP BY Temperature ORDER BY Temperature;
4. Website Traffic Insights
SELECT AVG(Website_Traffic) AS Average_Website_Traffic FROM sales_table;
SELECT Date, Website_Traffic FROM sales_table ORDER BY Website_Traffic DESC LIMIT 1 UNION SELECT Date, Website_Traffic FROM sales_table ORDER BY Website_Traffic ASC LIMIT 1;
SELECT CORR(Website_Traffic, Sales) AS Correlation_Website_Sales FROM sales_table;
5. Customer Purchase Behavior
SELECT Customer_purchase_behavior, SUM(Sales) AS Total_Sales FROM sales_table GROUP BY Customer_purchase_behavior;
SELECT Customer_purchase_behavior, AVG(Sales) AS Avg_Spending FROM sales_table GROUP BY Customer_purchase_behavior ORDER BY Avg_Spending DESC;
SELECT Customer_purchase_behavior, COUNT(*) AS Outlier_Count FROM sales_table WHERE Outlier_Flag = 1 GROUP BY Customer_purchase_behavior;
6. Outlier Detection
SELECT COUNT(*) AS Outlier_Count FROM sales_table WHERE Outlier_Flag = 1;
SELECT Date FROM sales_table WHERE Outlier_Flag = 1;
SELECT Date, Marketing_Spend, Website_Traffic FROM sales_table WHERE Outlier_Flag = 1;
Total_Sales
Avg_Spending
'Website_Traffic' by 'Marketing_Spend'
Business Questions
SSOT (Curated Zone)
SSOT (Data Catalog)
EDA-Descriptive Analysis
Ready
Accessibility: Investigate
69%

Date	Student_ID	Attendance	Avg_Exam_Score	AVG_Study_Hours
1/1/2024	S001	86	79.8135028	5.444160367
1/2/2024	S002	99	64.02699842	5.704595464
1/3/2024	S003	94	73.40306095	4.847869165
1/4/2024	S004	90	74.60196257	1.228772141
1/5/2024	S005	87	58.3184505	1.971022843
1/6/2024	S006	86	93.63130825	1.282862671
1/7/2024	S007	98	84.88097705	6.727693701
1/8/2024	S008	90	92.27745237	3.82920383
1/9/2024	S009	90	90.26723077	5.57713622
1/10/2024	S010	83	76.90549905	9.168098265
1/11/2024	S011	87	91.48434058	3.243630062
1/12/2024	S012	82	53.98216259	4.693446307
1/13/2024	S013	81	58.81922881	7.799960247
1/14/2024	S014	91	52.035228	3.059183489
1/15/2024	S015	85	64.63986488	1.692819188
1/16/2024	S016	81	67.49047804	3.607763076
1/17/2024	S017	80	62.21070643	2.450991585
1/18/2024	S018	91	87.29318791	9.367278871
1/19/2024	S019	91	66.0538997	8.273083416
1/20/2024	S020	96	62.64205294	6.700633809
1/21/2024	S021	89	74.42132374	8.843145312
1/22/2024	S022	95	56.34159012	8.233048692
1/23/2024	S023	94	86.09886413	2.67913053
1/24/2024	S024	94	53.35477897	9.033030986
1/25/2024	S025	98	94.40991215	5.854080177
1/26/2024	S026	91	84.75101462	8.266961396
1/27/2024	S027	99	58.94220567	9.064821699
1/28/2024	S028	82	50.24849527	3.862031275
1/29/2024	S029	84	86.69576428	1.990467321
1/30/2024	S030	98	81.80858047	3.051416463
1/31/2024	S031	86	82.80532256	4.843970098
2/1/2024	S032	88	84.7071656	8.362132893
2/2/2024	S033	86	53.33200933	8.746575249
2/3/2024	S034	97	66.13095778	1.062569175
2/4/2024	S035	83	55.21410768	5.596725723
2/5/2024	S036	93	88.83965416	4.756699028
2/6/2024	S037	97	78.04841571	2.998970294
2/7/2024	S038	88	64.89041112	2.078788306
2/8/2024	S039	81	52.86012576	4.038536543
2/9/2024	S040	99	63.99420448	9.486187335
2/10/2024	S041	94	64.63324949	3.908826388
2/11/2024	S042	86	82.83227803	5.669115596
2/12/2024	S043	91	78.69008621	7.32717063
2/13/2024	S044	87	89.92457342	4.272666421
2/14/2024	S045	94	71.24967163	9.746038744
2/15/2024	S046	82	55.38174107	9.662025654

2/16/2024	S047	93	82.09601543	3.266040662
2/17/2024	S048	96	84.23532719	5.475236553
2/18/2024	S049	83	75.25747389	3.707904788
2/19/2024	S050	97	84.6935231	3.563564449

Field	Descriptive Analysis (What Happened)	Diagnostic analysis (Why did it happen)	Predictive analysis (What will happen)	Prescriptive analysis (what should we do)
Date				
Student_ID				
Attendance	What are the average attendance of students?			
Avg_Exam_Score		Why do some students score better than others?		
AVG_Study_Hours			How will study hours impact student scores in exame?	
Learning_Style				How can UCW manager personalize education based on students learning styles?
Engagement_Level				

Learning_Style	Engagement_Level
Visual	Medium
Kinesthetic	High
Visual	High
Auditory	Medium
Auditory	Medium
Auditory	High
Kinesthetic	High
Visual	Medium
Visual	Low
Visual	Low
Kinesthetic	Medium
Auditory	Low
Auditory	Medium
Visual	Low
Auditory	Low
Auditory	High
Kinesthetic	High
Kinesthetic	Low
Kinesthetic	Low
Kinesthetic	High
Visual	High
Kinesthetic	High
Auditory	Medium
Visual	Low
Auditory	High
Auditory	Low
Auditory	Medium
Kinesthetic	Medium
Kinesthetic	High
Visual	Medium
Visual	Low
Kinesthetic	High
Auditory	Low
Visual	Low
Kinesthetic	Medium
Kinesthetic	Medium
Kinesthetic	Medium
Auditory	Medium
Kinesthetic	Medium
Kinesthetic	High
Kinesthetic	Low
Kinesthetic	Low
Visual	Low
Visual	Low
Kinesthetic	Medium
Auditory	Low

Visual	High
Kinesthetic	High
Visual	Low
Visual	Low

Causal analysis
(what should happen if)

If engagement increases, how will student scores will be affected?

Date	Student_ID	Attendance
1/1/2024	1001	95
1/15/2024	1002	80
2/1/2024	1003	60
2/20/2024	1004	90
3/10/2024	1005	50
3/25/2024	1006	85
4/5/2024	1007	92
4/20/2024	1008	45

Field	**Summarization** (Max, Min, Sum, Avg, Filter, Projection, Grouping...)	**Trends** A long-term increase or decrease in data over time.
Date		
Student_ID		No
Attendance	Filtring	
Avg_Exam_Score	Avg	
AVG_Study_Hours	Avg	
Learning_Style		
Engagement_Level		

Avg_Exam_Score	AVG_Study_Hours	Learning_Style	Engagement_Level
85	12	Visual	High
78	8	Auditory	Medium
55	4	Kinesthetic	Low
88	14	Visual	High
40	2	Kinesthetic	Low
82	10	Auditory	Medium
90	15	Visual	High
30	1	Kinesthetic	Low

Seasonality	**Correlation**	**Clusters**	**Anomaly**
Recurring patterns at regular intervals.	A relationship between two or more variables.	Grouping of similar data points.	Data points that do not follow the general pattern.

No

	YES	Yes
Exam Score	YES	Yes
Study Hours	YES	Yes
	YES	
	YES	

ID	Name	Age	Date of Birth
1	Alice Johnson	30	01-02-1993
2	Bob Smith	29	5/12/1994
3	Charlie Brown	-5	10/10/2015
4	Dana Lee	45	11/11/1978
5	Eve Adams		7/25/1985
6	Frank Miller	35	12-12-1988
7	Alice Johnson	30	01-02-1993

Duplicated Issues

Missing Value

Invalid Value

Inconsistant Formating

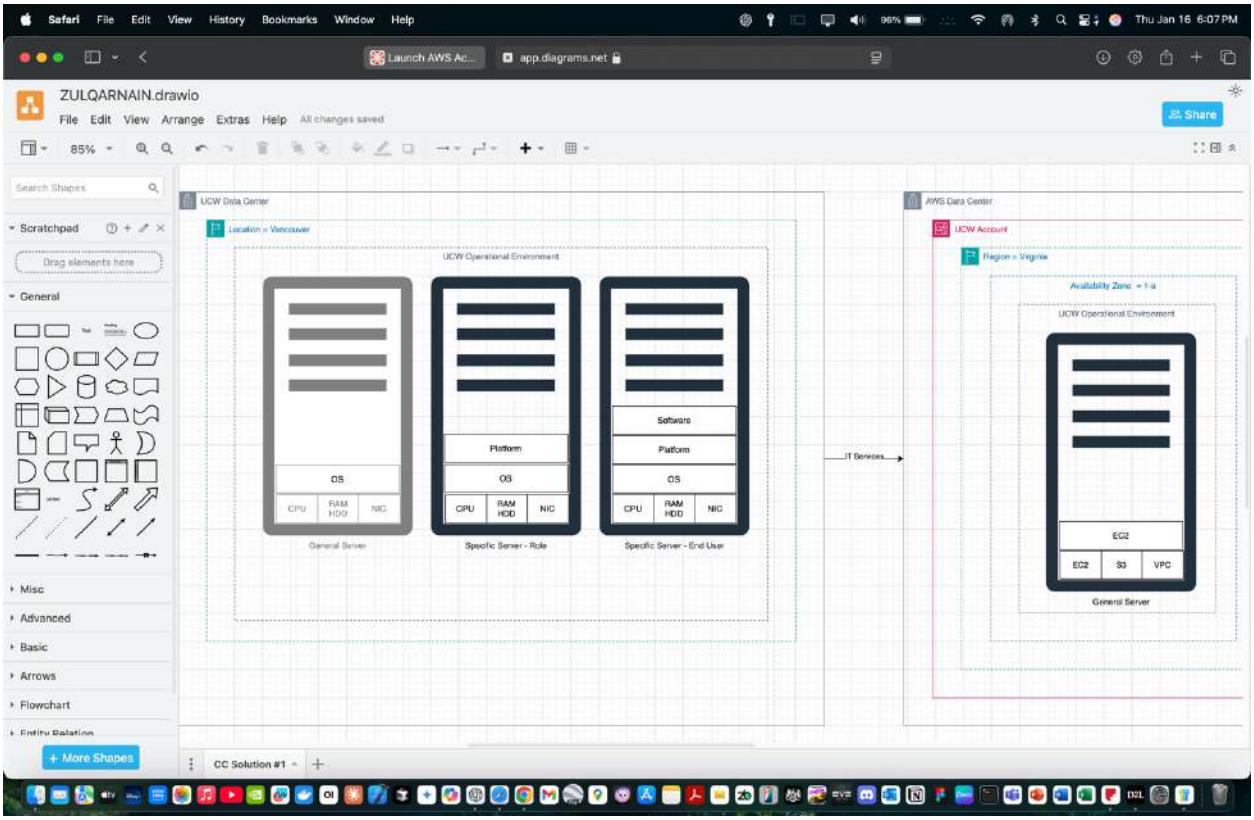
Country	Income	Gender
USA	$50,000	F
United States	55000	Male
US	$100,000	M
USA	,Ç¨60,000	female
USA	$70,000	F
	$120,000	M
USA	$50,000	F

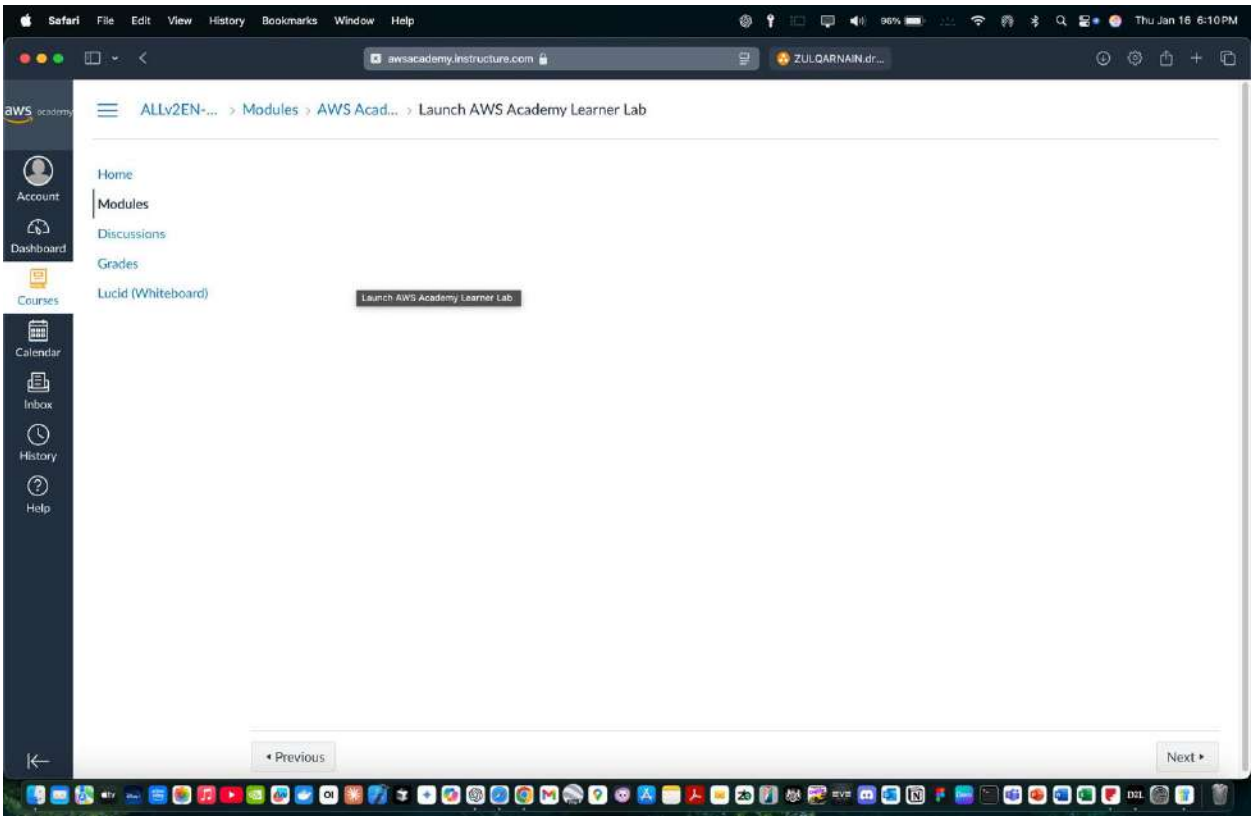
Missing Value
Encoding issue
Inconsistent Formate
Inconsistant Formating
Typo

Transaction Date	Height (cm)	Satisfaction
2/5/2023	170	7
05-02-2023	180	8
2/1/2023		6
13-02-2023	165	9
2/15/2023	172	
2023.02.10	175	10
2/5/2023	170	7

Inconsistent Formating	Missing Value	Missing Value

Duplicated Row


Safari File Edit View History Bookmarks Window Help
Thu Jan 16 6:07 PM
Launch AWS Ac...
app.diagrams.net
ZULQARNAIN.drawio
File Edit View Arrange Extras Help All changes saved
Share
85%
Search Shapes
Scratchpad
Drag elements here
General
Misc
Advanced
Basic
Arrows
Flowchart
Entity Relation
+ More Shapes
UCW Data Center
Location = Vancouver
UCW Operational Environment
OS
CPU
RAM HDD
NIC
General Server
Platform
Specific Server - Role
Software
Specific Server - End User
IT Services
AWS Data Center
UCW Account
Region = Virginia
Availability Zone = 1-a
EC2
S3
VPC
CC Solution #1


Safari File Edit View History Bookmarks Window Help
Thu Jan 16 6:10 PM
awsacademy.instructure.com
ZULQARNAIN.dr...
aws academy
ALLv2EN-... › Modules › AWS Acad... › Launch AWS Academy Learner Lab
Account
Dashboard
Courses
Calendar
Inbox
History
Help
Home
Modules
Discussions
Grades
Lucid (Whiteboard)
Launch AWS Academy Learner Lab
Previous
Next

Professor, I’m not able to open this I will try this again at home!

Group Project- AWS Data Analytic Platform for The City of Vancouver

Gyanvi Bhardwaj (2320329)

Peng Wang (2306811)

Muhammad Zulqarnain Shahzad (2306553)

Haoran Xu (2317218)

University Canada West

BUSI653: Cloud Computing Technologies (HBD-WINTER25-03)

Dr Mahmood Mortazavi Dehkordi

March 05, 2025

Table of Contents

Introduction

City of Vancouver requires a Data Analytics Platform (DAP) to process and summarise bulk datasets with efficiency to make decisions. By utilising AWS services, scalability and accuracy can be ensured to understand city-based data.

The foundation of creating a data analytics platform for the City of Vancouver involves choosing relevant datasets followed by their preparatory processing. The project adopted four datasets namely Public Trees, 3-1-1 Contact Metrics, Business Licenses and Animal Control Inventory Register. The dataset filtering process aimed to maintain the right data amount. The data required profiling alongside cleaning steps which produced accurate data for analysis purposes.

The next step involved data preparation after which the data required cataloging. AWS S3 operated as a scalable storage platform that provided secure data access capabilities. AWS Glue performed data transformation after which it structured each dataset to allow analysis. The data formatting process as a part of this step ensured that information prepared for querying requirements correctly.

AWS Glue served as the platform to conduct data summary and analysis tasks after processing and storage took place. The data team implemented diverse analytical methods to obtain useful conclusions. The analysis of Public Trees data evaluated tree diameter versus height range, but the 3-1-1 Contact Metrics data supported standard service levels analysis. The analysis of business license records revealed worker-based trends and evaluated the Animal Control data for finding animal breed ratios and impoundment statistics.

The analysis generated valuable recommendations about municipal changes together with business enhancements that were derived from the statistical results. Data pattern assessments

from these analyses helped provide knowledge about environmental monitoring methods together with service efficiency improvements alongside policy effects. Through this structured process the City of Vancouver can create optimal public services which enable data-based decision-making for community welfare.

DAP Design and Implementation (Individual work – Gyanvi Bhardwaj)

Descriptive Analysis

Dataset selection

The selected dataset is sourced from the ***'Animal Control Inventory Register'*** under the theme of ***'Parks, Recreation and Pets'*** from the City of Vancouver data portal for the year 2024 (*Animal Control Inventory - Register*, 2024).

Link: Animal Control Inventory 2024

This dataset provides a comprehensive list of all the animals that have been reported to the City of Vancouver Animal Control Department and have been taken into their custody. It is one of the three interlinked datasets about animal reporting in Vancouver. The other related datasets include lost and found animals, and deceased animals datasets. It includes categorization based on various factors such as medical status, breed, colour and any known history of the animal.

Why the 'Animal-control-inventory-register' dataset?

I have chosen the 'Animal-Control-Inventory-Register' dataset due to its relevance in decision-making for the animal control department. The well-structured dataset consists of a comprehensive overview of the various animals that are in the custody of the City of Vancouver, differentiating them based on various attributes like breed, sex, colour, age category etc. By

utilizing this data, animal control officials can improve resource distribution, shelter capacity and trends in impoundment. By understanding breed or age categories, we can better evaluate animals that need urgent attention. By including operational details like source and status, we can reduce the number of stray animals and understand trends of impoundment. This relevance makes the dataset an invaluable tool for informing data-driven policies, improving animal welfare, and ensuring more efficient and responsive animal control practices.

Goal:

To analyze the distribution of animal breeds across age categories and identify the most recent impound date to make informed animal control strategies.

Analytical questions:

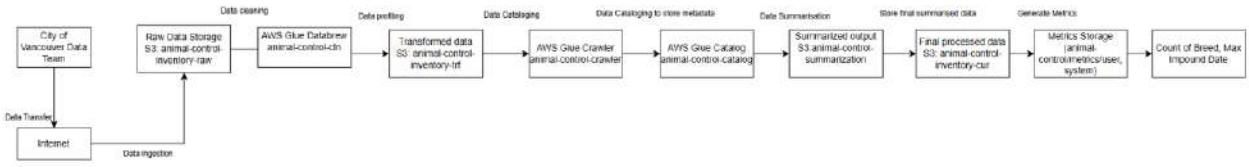
- What is the distribution of animal breeds across different age categories?
- What is the most recent date of animal impoundments?

Descriptive Metric:

- Count of animals (by breed)
- Maximum impound date (most recent impoundment date)

Figure 1

Components of Data Analytics Platform for the City of Vancouver



Note. Components of Data Analytics Platform (DAP) for the City of Vancouver for Animal-control-inventory dataset. (Source: *self-work*, created on draw.io)

The components involved in the DAP begin from the City of Vancouver data portal. The data team of the City of Vancouver works on the raw data ingested into the raw data S3 bucket. This raw data is then sent to AWS Glue DataBrew for data profiling and cleaning. The cleaned and transformed are stored in the transformed S3 bucket. A crawler is created for data cataloging and summarisation using AWS Glue services. The curated data is stored in the curated S3 bucket.

Step 1: Data Ingestion

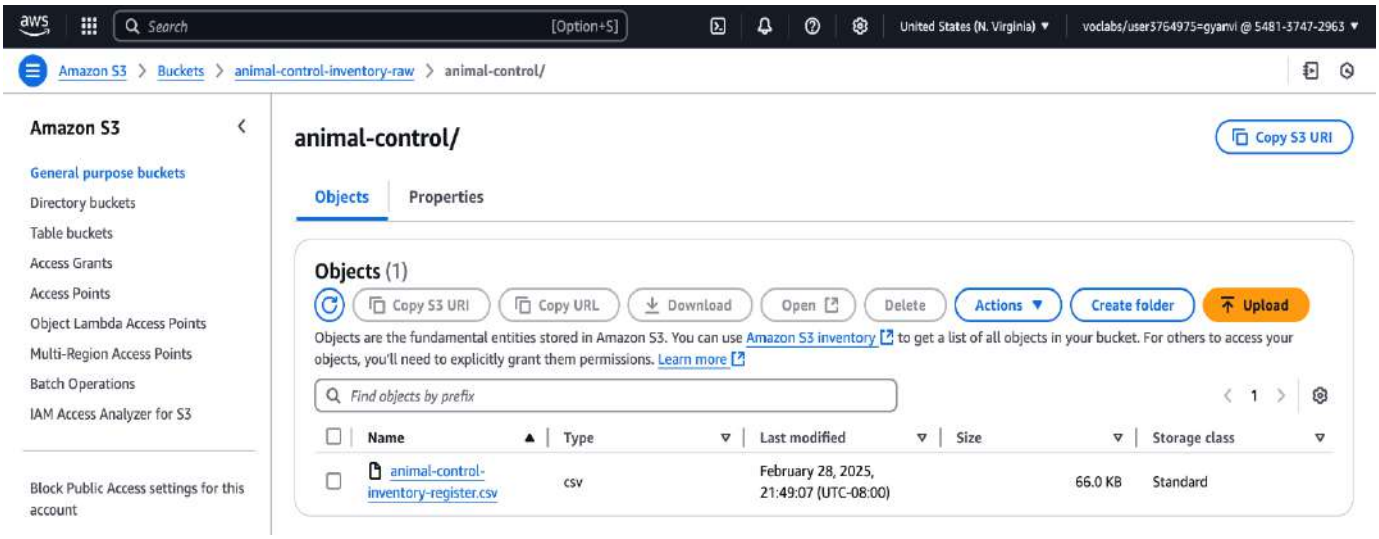
The data ingestion step includes extracting data from its source and loading it into an AWS S3 bucket. In my case, I have exported a CSV file named ‘Animal-control-inventory-register’, filtered for 2024 with a total of 510 rows and 18 columns from the data portal of the City of Vancouver.

Using an AWS storage service named S3, I have created a raw bucket called ‘animal-control-inventory-raw’ to ingest the raw data into a folder called ‘animal-control’. Our ingestion

rate is daily and since the dataset is already filtered for 2024, I have not created any other sub-folder to keep visualization easier. I have successfully loaded my CSV dataset into the animal-control folder.

Figure 2

AWS S3 bucket with raw ingested data



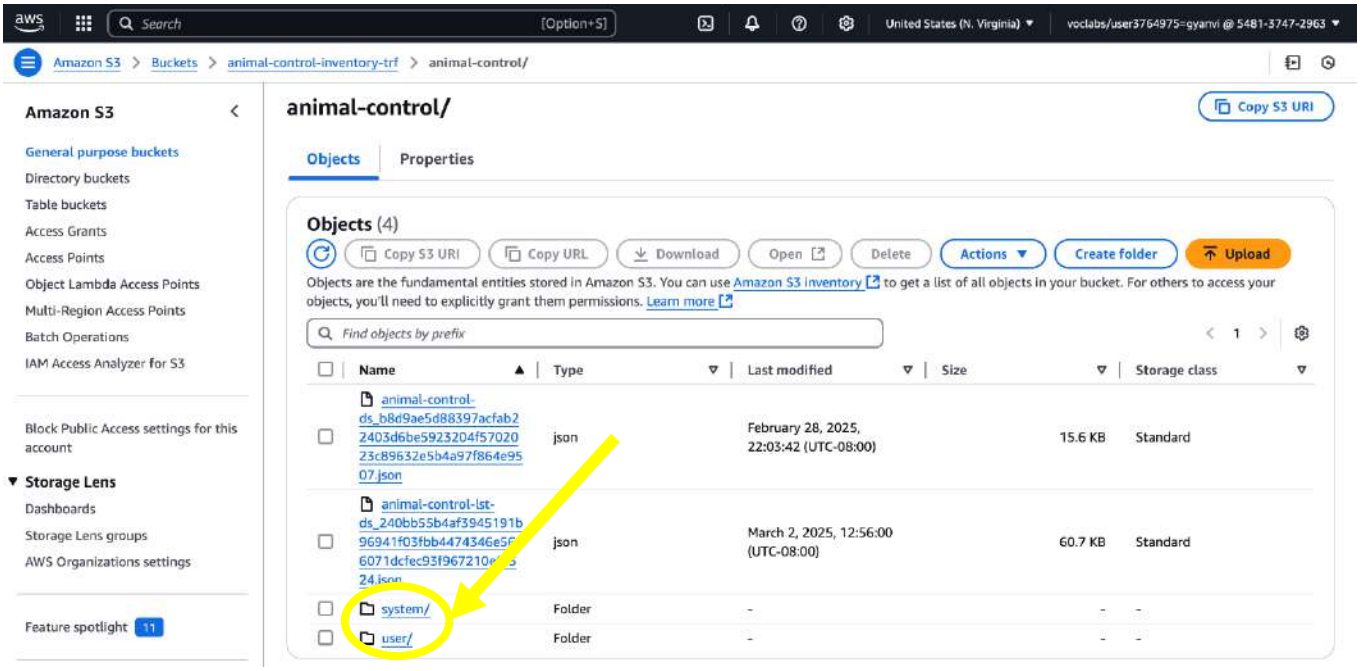
Note. The data ingestion step is performed by ingesting data from the source- City of Vancouver data portal and loaded into our raw S3 bucket. Source: AWS S3, *self-work*

Step 2: Data Profiling

Data profiling is a crucial step as it helps us to understand the structure and identify various possible issues with the dataset. I have created another S3 bucket named 'animal-control-inventory-trf' to store transformed data after performing data profiling and data cleaning. Inside this bucket, we have created another sub-folder called 'animal-control', which consists of two more sub-folders named 'user' and 'system'. The user folder will have data in a user-friendly readable format, which is .csv with a semi-colon as the delimiter and will generate a single output file. The system folder will contain multiple files of snappy type which is easy for the system to read.

Figure 3

AWS S3 bucket with transformed data and two sub-folders

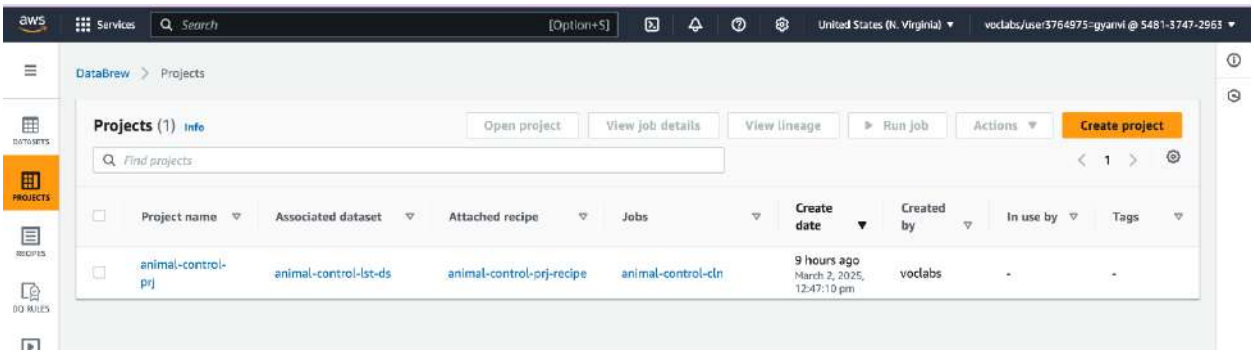


Note. The data profiling step is initiated by creating a new S3 bucket with two folders-user and system to store the transformed data. Source: AWS S3, *self-work*

Next, I utilized AWS Glue DataBrew to visualize data for preparation without writing any code. The first step includes creating a project. I have created a project called ‘animal-control-prj’ and its associated dataset called ‘animal-control-lst-ds’ is derived from the raw bucket.

Figure 4

Project created for data profiling



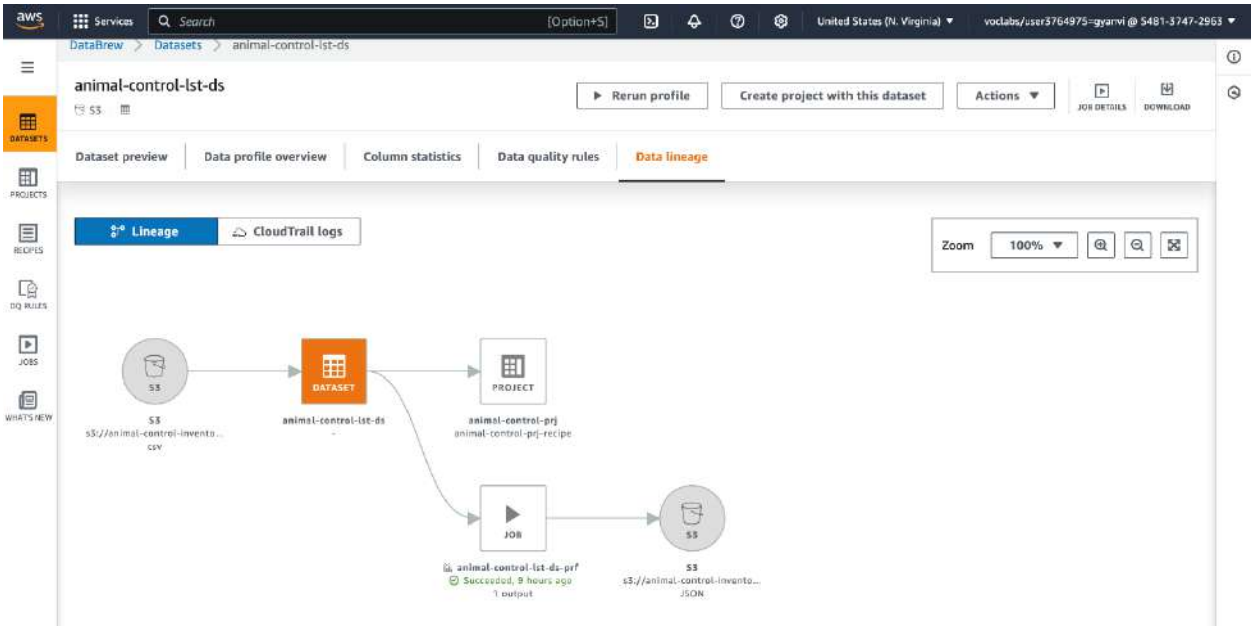
Note. A project to visualize data for data profiling is created using AWS Glue DataBrew. Source: AWS Glue DataBrew, self-work

Step 3: Data Cleaning

Data cleaning is very important to clean and prepare the raw data for analysis. It handles all errors in the dataset such as missing values, duplicate values, outdated values, etc. All 17 issues are taken care of using AWS DataBrew cleaning techniques by creating a cleaning job named ‘animal-control-cln’. The recipe created handles all the cleaning steps that I have used to clean my dataset. There is a total of 24 recipe steps implemented to clean and store our data in the transformed S3 bucket.

Figure 5

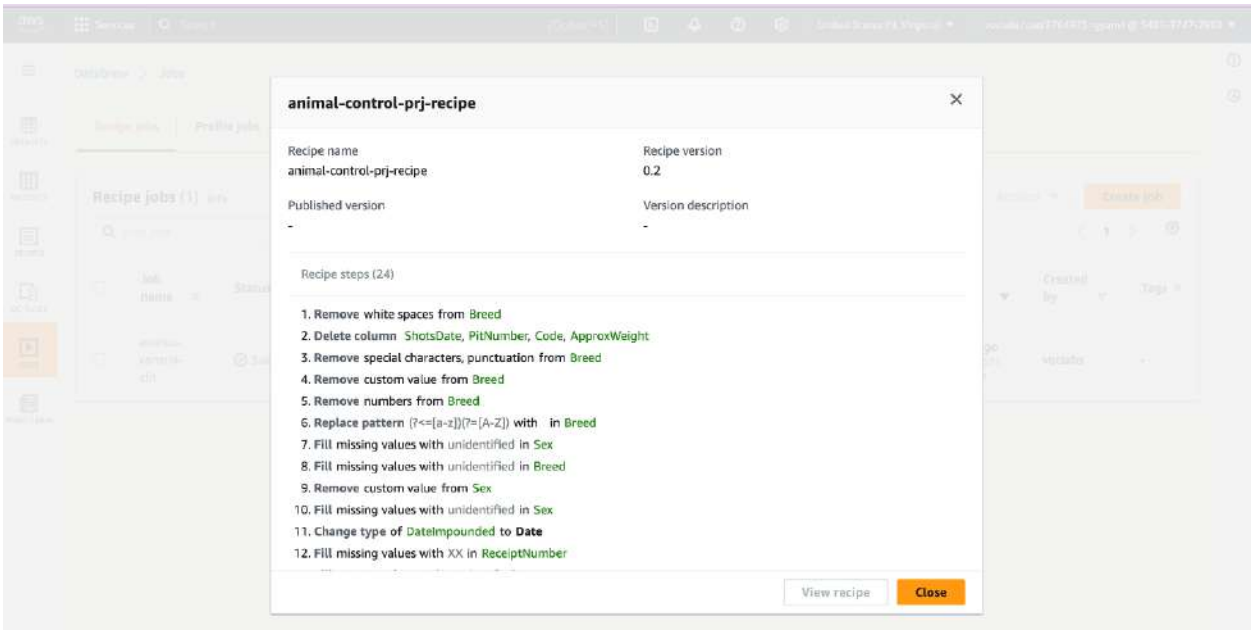
Data lineage visualization for the job created



Note. Data lineage visualization in AWS Glue DataBrew shows how the dataset is taken for the project from the raw bucket and the job is run to clean the raw data. Source: AWS Glue DataBrew, *self-work*

Figure 6

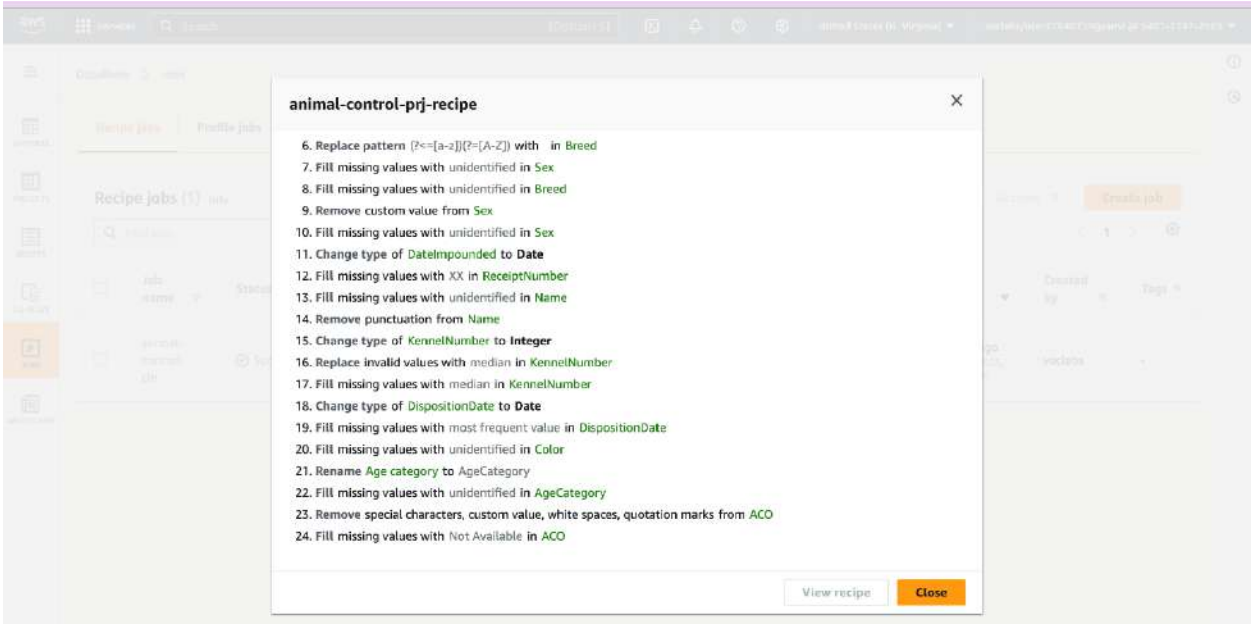
Recipe steps for data cleaning



Note. The overview of the recipe steps used to clean our animal-control-inventory dataset is listed using AWS Glue DataBrew. Source: AWS Glue DataBrew, *self-work*

Figure 7

Recipe steps for data cleaning



Note. The overview of the recipe steps used to clean our animal-control-inventory dataset is listed using AWS Glue DataBrew *(contd.)*. Source: AWS Glue DataBrew, *self-work*

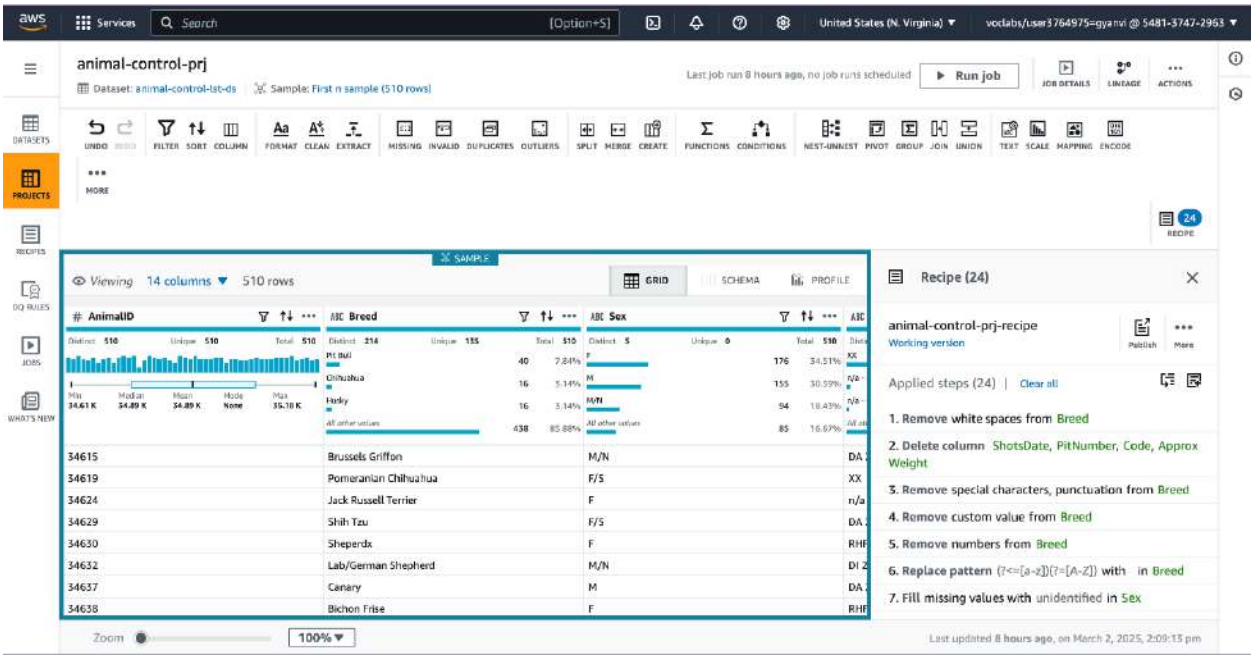
Data cleaning steps used:

1. Removing all leading and trailing white spaces
2. Delete four columns that had no data.
3. Removed special characters and unwanted punctuation.
4. Removed numbers from the string datatype columns.
5. Filled custom value using Regex code for column Breed to put space between sentence cases.
6. Changed format from timestamp to date.
7. Filled missing values with ‘unidentified’ for better understanding.

8. Changed the datatype of column Kennel number from string to integer.
9. Changed the structure of the column name to sentence case.
10. Filled custom value using most frequent data for disposition date.

Figure 8

Cleaned dataset ready for analysis



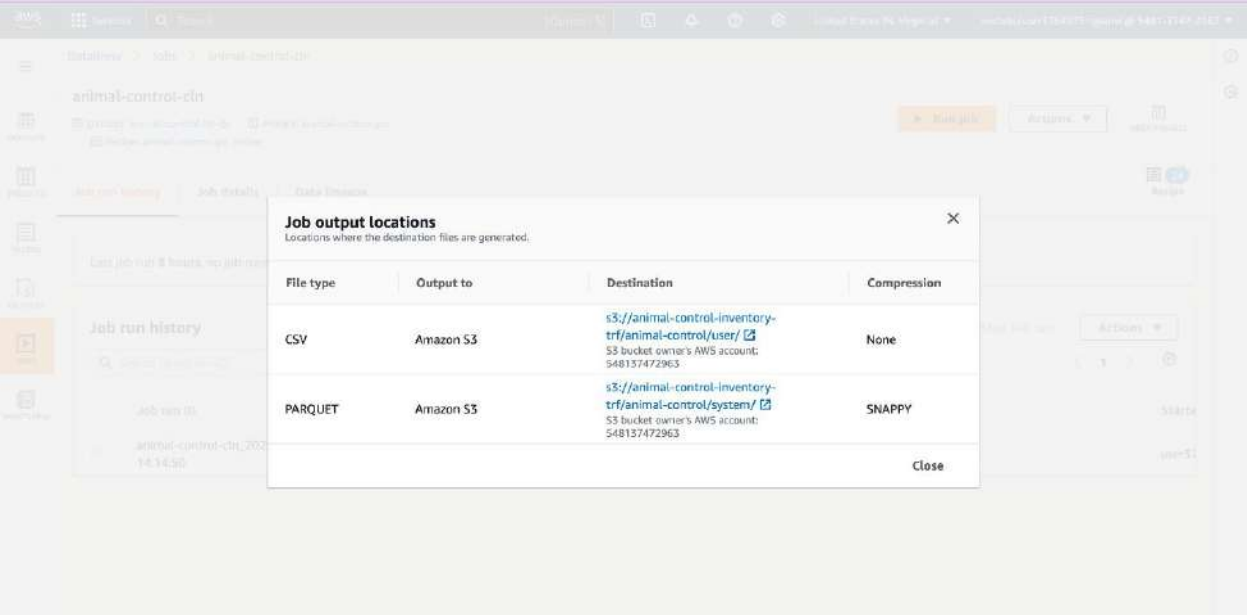
Note. The cleaned dataset consists of 14 columns and 510 rows and has no dataset issues.

Source: AWS Glue DataBrew, *self-work*

The transformed data after the cleaning procedure is stored in the bucket 'animal-control-inventory-trf'. After the cleaning job is run, the output is stored in two files- CSV and PARQUET. The former is user-friendly, and the latter is system friendly. The partition key for the system is the *Age Category* of the animals. It is the most suitable categorical column available in the dataset.

Figure 9

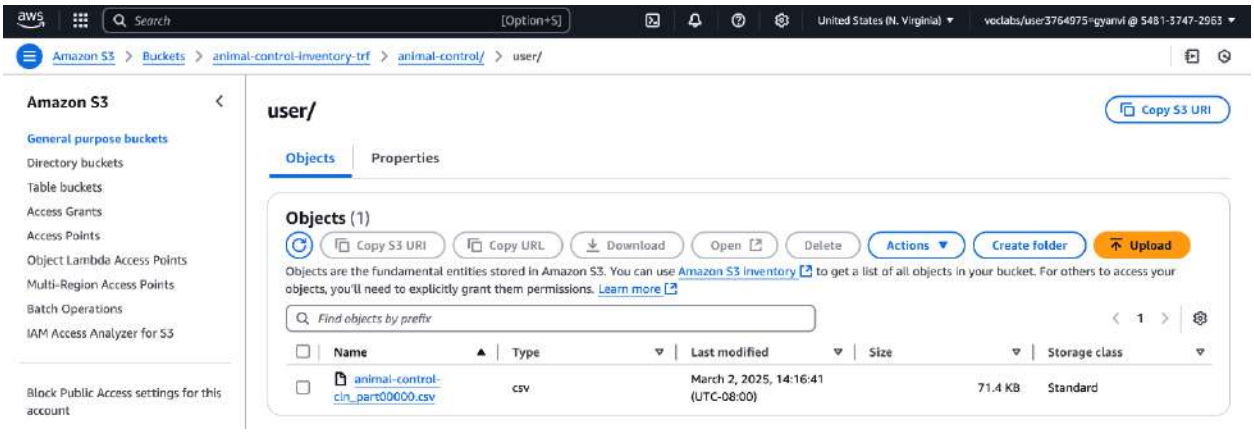
Output of the job run into two folders of the transformed bucket



Note. The image shows a summary of the output locations of the job that was created and run, consisting of two file types stored in a transformed S3 bucket. Source: AWS S3, *self-work*

Figure 10

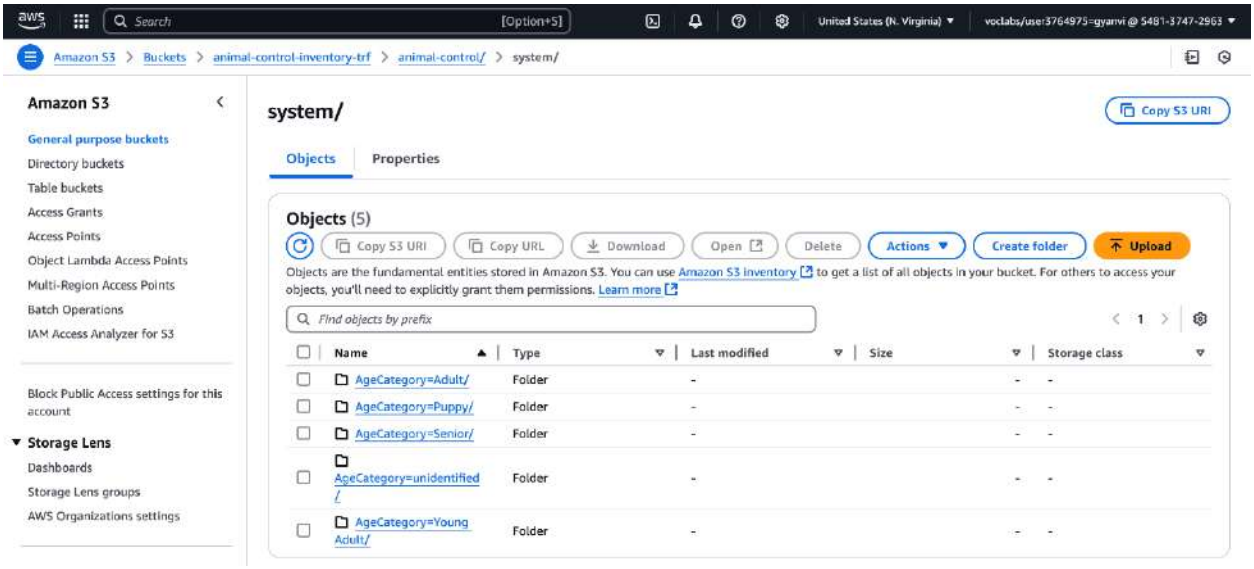
User folder of animal-control-inventory-trf bucket



Note. The user folder of the transformed bucket contains the cleaned dataset file without any partitions. Source: AWS S3, *self-work*

Figure 11

System folder of animal-control-inventory-trf bucket



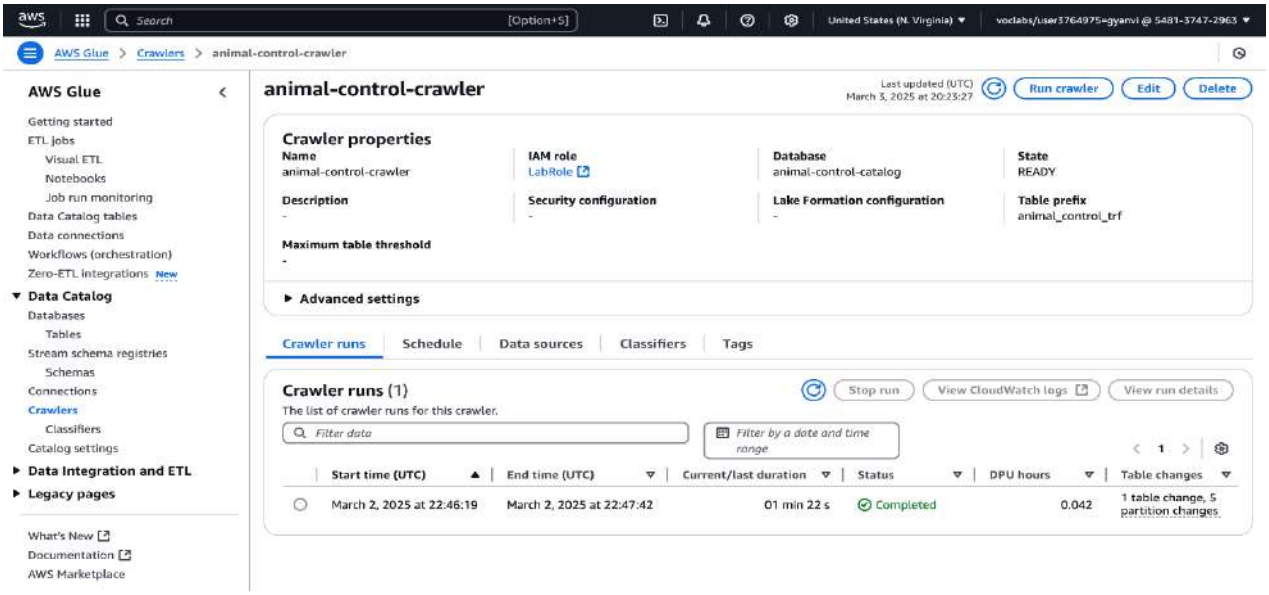
Note. The system folder of the transformed bucket contains the folders partitioned according to the age category of the animals. Source: AWS S3, *self-work*

Step 4: Data Cataloging

The data catalog stands as a single source of truth for my metadata. I started by creating a database on AWS Glue named ‘animal-control-catalog’. A crawler is created and named ‘animal-control-crawler’ to create the meta table in my data catalog.

Figure 12

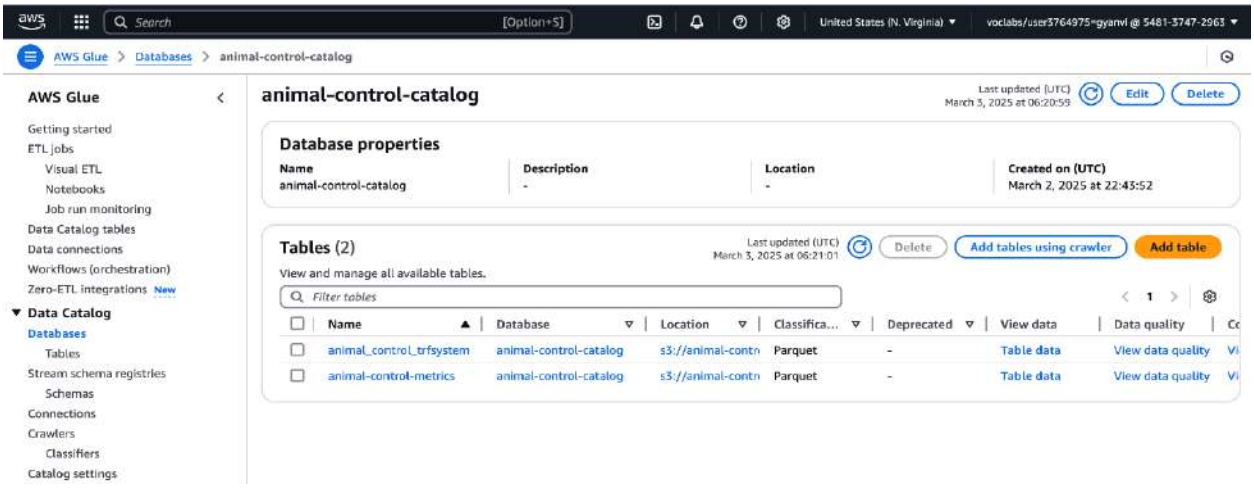
Crawler creation



Note. A crawler is created and named ‘animal-control-crawler’ to create the meta table in my data catalog. Source: AWS Glue, *self-work*

Figure 13

Data catalog with meta tables created



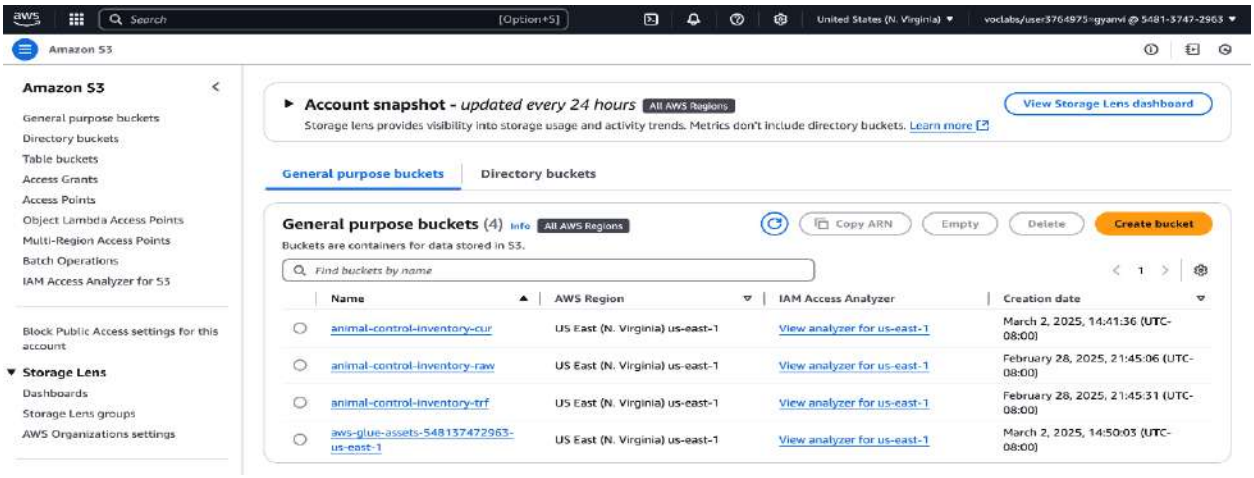
Note. Two meta tables, one with the desired metrics were created in my data catalog. Source: AWS Glue, *self-work*

Moving back to AWS S3, another bucket to store data in the curated zone is created with the name ‘animal-control-inventory-cur’. It has a folder named ‘animal-control’ and then another

folder called ‘metrics’ and two sub-folders named user and system to store user and system-friendly output. Now we have S3 buckets.

Figure 14

AWS S3 buckets- raw, transformed and curated buckets



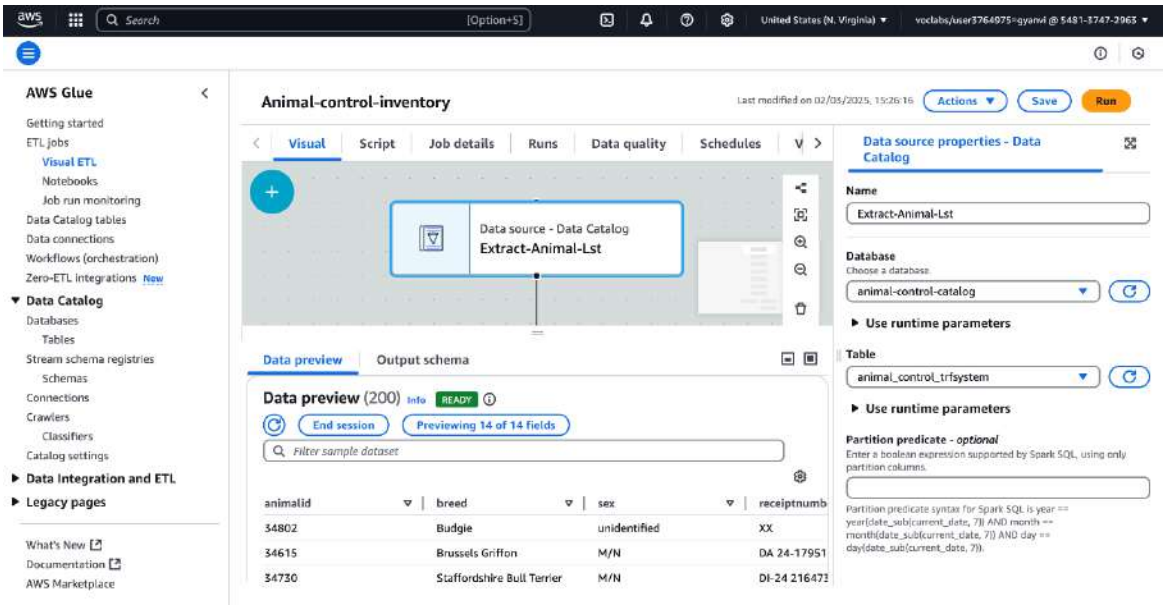
Note. A list of all the buckets created in the AWS S3 environment, including raw, transformed and curated buckets. Source: AWS S3, *self-work*

Step 5: Data Summarization

For data summarization, I implemented an ETL job named ‘Animal-control-inventory’ to create an ETL pipeline. The first step is Extraction which includes inputting a data catalog. The database chosen is ‘animal-control-catalog’ and the table is ‘animal_control_trfsystem’.

Figure 15

ETL Step 1

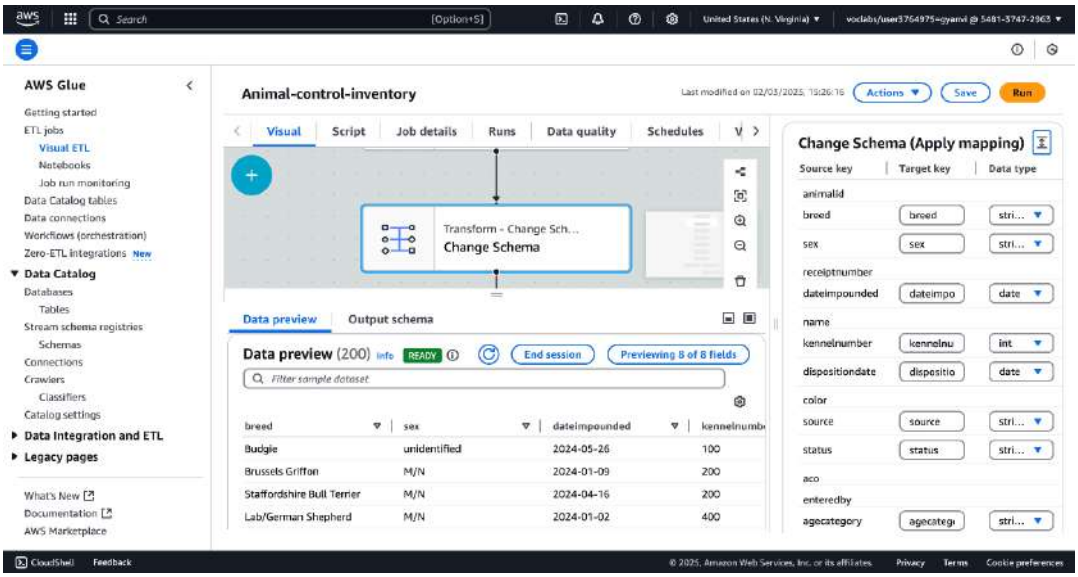


Note. The first step in the visual ETL called extraction is carried out using a data catalog Source: AWS Glue, *self-work*

Next, under transformation, I applied the change schema technique by dropping a few columns called Animal Id, receipt number, name, colour, and ACO, entered by as they do not contribute to summarization.

Figure 16

ETL Step 2

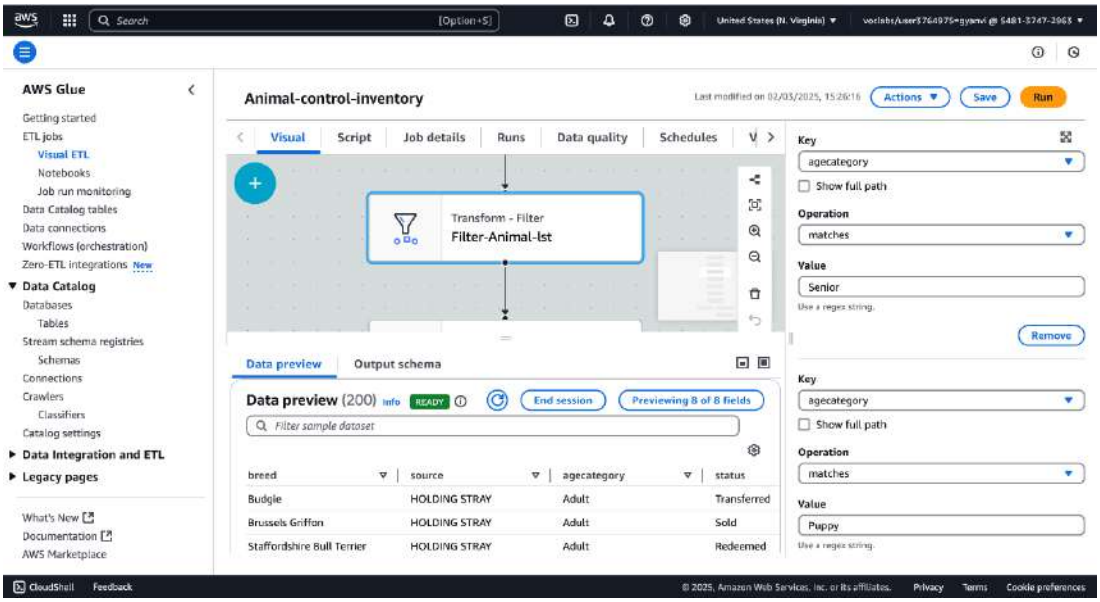


Note. Transformation step called change schema to apply mapping techniques. Source: AWS Glue, *self-work*

Another step under transformation is filter, where I filtered using Global OR and applied four filter conditions. The key is the age category and the operation is ‘matches’. The four values are young, senior, puppy and adult.

Figure 17

ETL Step 3

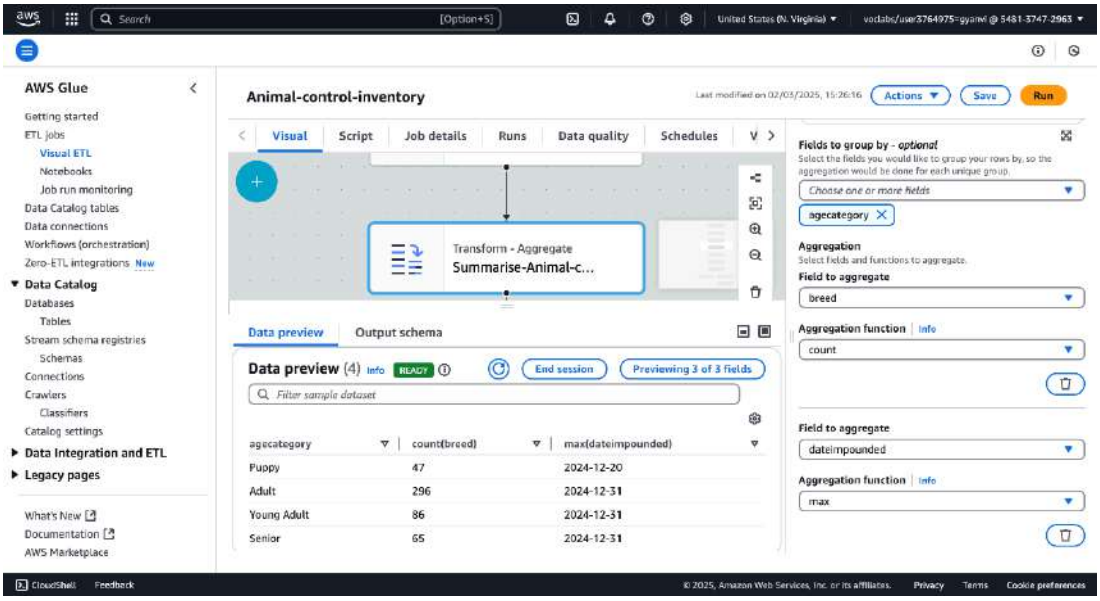


Note. The transformation step called filtering to apply conditions to dataset Source: AWS Glue, *self-work*

Another transformation step is aggregate. I chose the age category as the field to group by as it is the most suitable categorical data. The fields to aggregate are breed and date impounded with count and max as their respective aggregate functions.

Figure 18

ETL Step 4

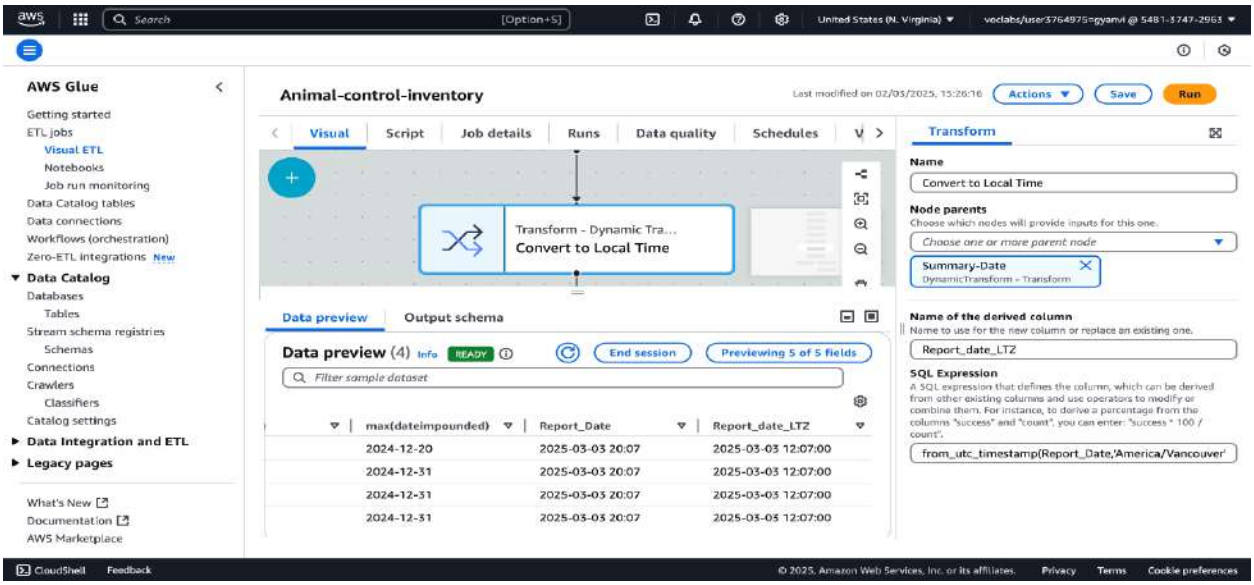


Note. Transformation step called aggregate to perform count and max operations Source: AWS Glue, *self-work*

Next, I added a date and timestamp column under the transformation technique and then used the derived column to change the time zone from American UTC to Vancouver local time. The SQL expression used to do this conversion is from_utc_timestamp(Report_Date,'America/Vancouver').

Figure 19

ETL Step 5 and 6



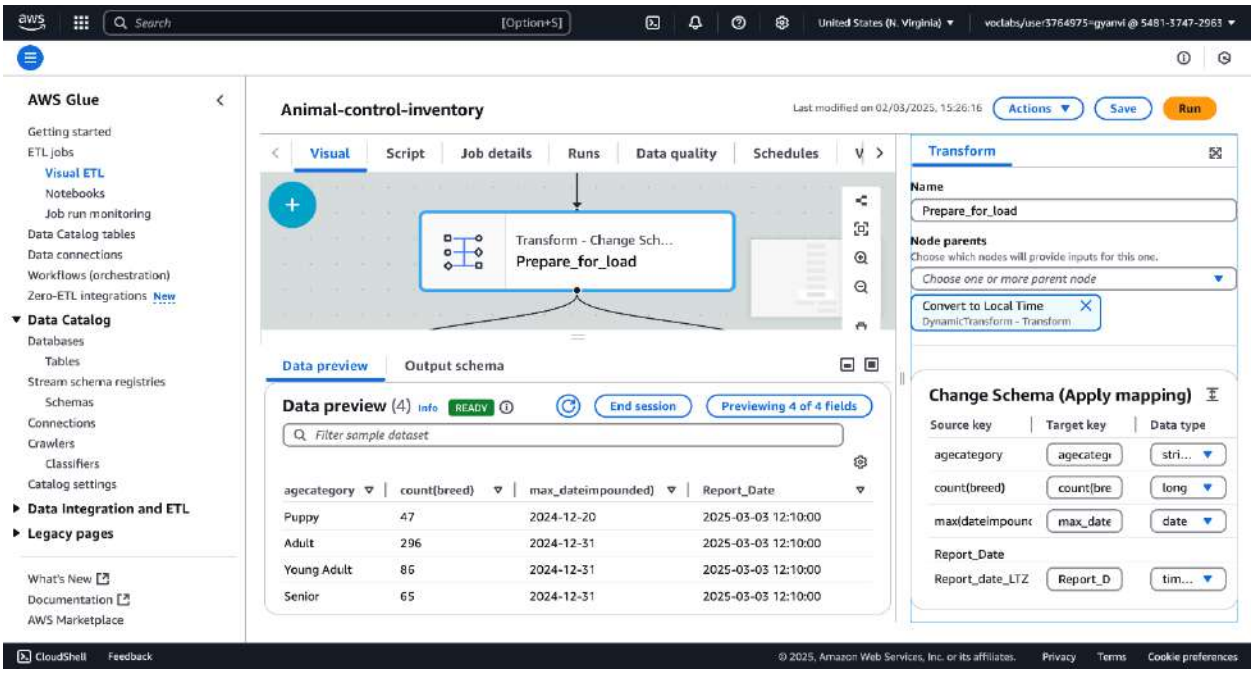
Note. Transformation technique to insert current timestamp and convert it into local time zone

Source: AWS Glue, *self-work*

We will now use the change schema technique to drop the report date column and keep the updated report date to prepare for the load.

Figure 20

ETL Step 7

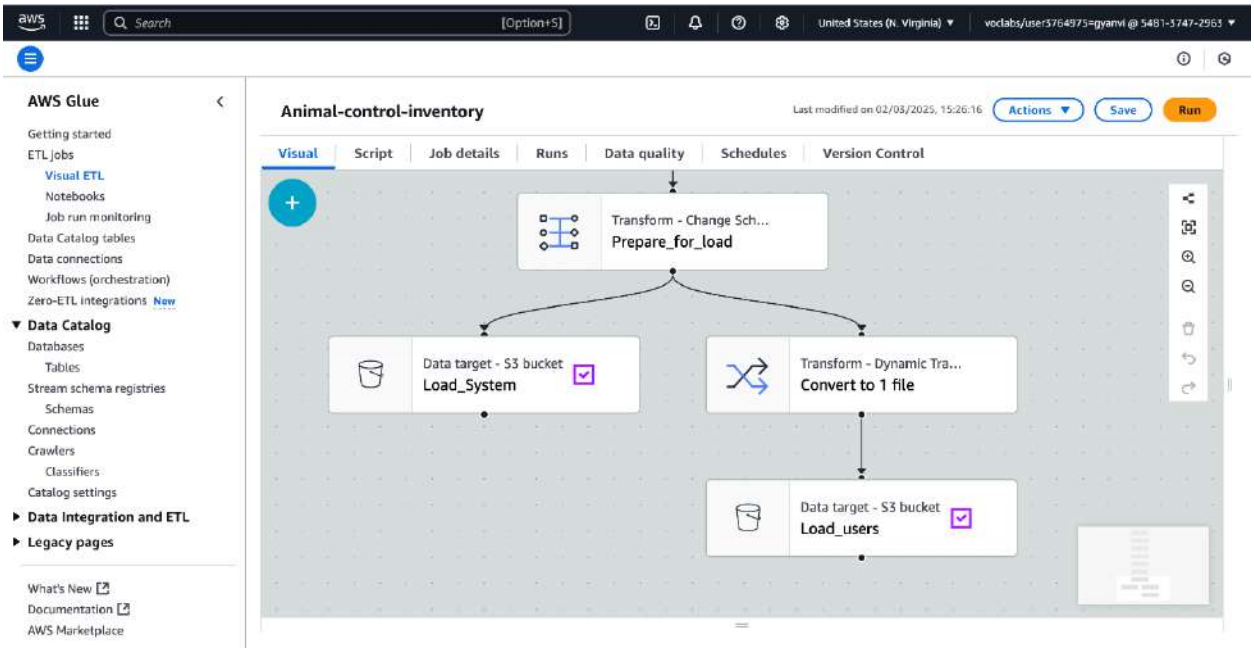


Note. Transformation step to change schema and prepare for loading by dropping American time zone column for report date. Source: AWS Glue, *self-work*

The last step in summarization concludes with two S3 buckets, one for the system and one for users as mentioned in the curated bucket. Before loading the user's S3 bucket, I implemented auto-balancing to convert the output into one file, so it is easier for the user to understand. The table name for the outputs is ‘animal-control-metrics’. The first partition category for the system file is report date and the second partition is using age category.

Figure 21

ETL Step 8, 9, 10

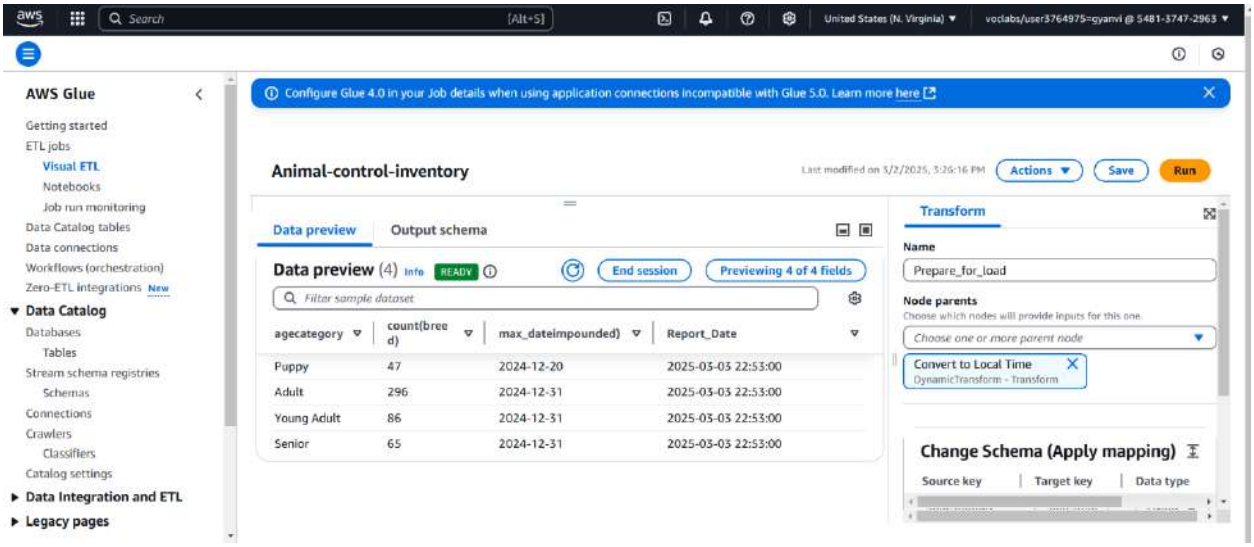


Note. Created two S3 buckets to load system and user data. Source: AWS Glue, *self-work*

The only partition for user output is the report date. Both the metrics- count of breed and maximum impound date are based on the age category of the animal.

Figure 22

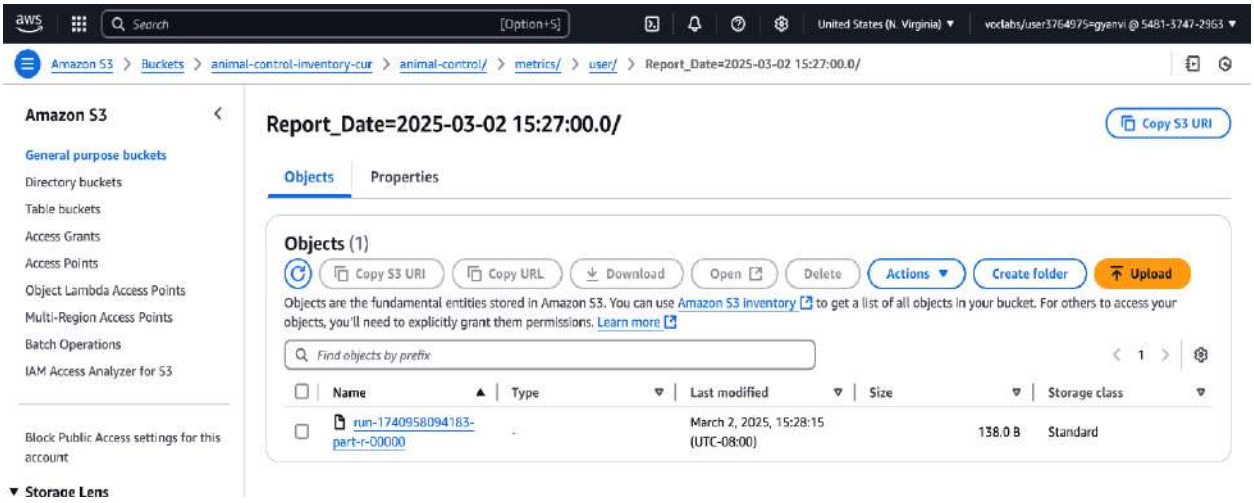
Output of summarization



Note. The two metrics created are counting the number of breeds and the latest (maximum) date of impoundment according to the age category Source: AWS Glue, *self-work*

Figure 23

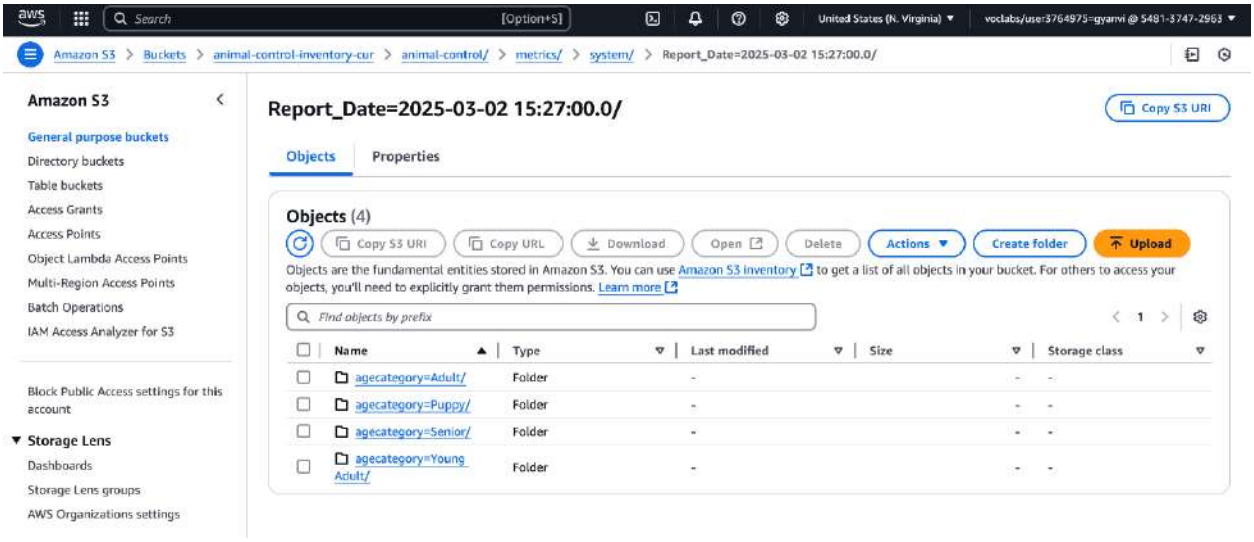
User output in the curated bucket partitioned by the report date



Note. The user folder contains the report date folder which contains the user output file Source: AWS S3, *self-work*

Figure 24

System output based on the age category and partitioned by the report date



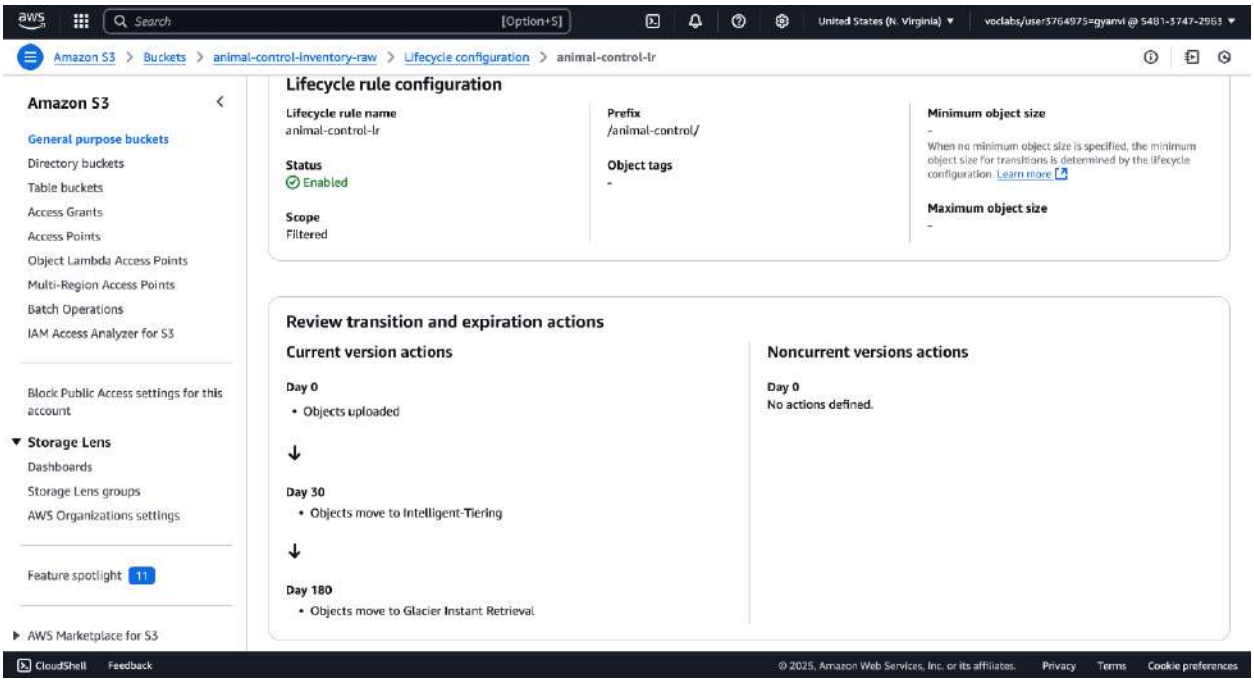
Note. The system folder is partitioned by the report date and this folder consists of other folders partitioned by the age category. Source: AWS S3, *self-work*

Cost Estimation

To further optimize my cost, I have created a lifecycle rule for cost management. The present storage class for my S3 object is standard as the animal control inventory data needs to be accessed frequently and retrieved within milliseconds. However, after 30 days, our storage class can shift to Intelligent tiering as the data trends can vary- there might be fluctuations in accessing data depending on the animals being taken into the custody of the City of Vancouver. Finally, after 180 days, the storage class can shift to glacier instant retrieval for data that sits for a long time and is hardly accessed but when needed, it can be retrieved within milliseconds.

Figure 25

Lifecycle Configuration



Note. Lifecycle rule configuration shows the transition from the standard storage class to Intelligent tiering and to Glacier Instant Retrieval. Source: AWS S3, *self-work*

DAP Design and Implementation (Individual work – Peng Wang)

Descriptive Analysis

I selected 3-1-1 contact metrics, including the data in 2025 and 2024, which has a total of around 400 rows, and the components included date, calls offered, calls handled, calls abandoned, average speed of answer, service level, and BI ID number. Additionally, I implemented this dataset into the AWS platform and gave advice on questions. The first question is about whether the average speed of answer is standard. The second question is about whether call abandoned time is also standard.

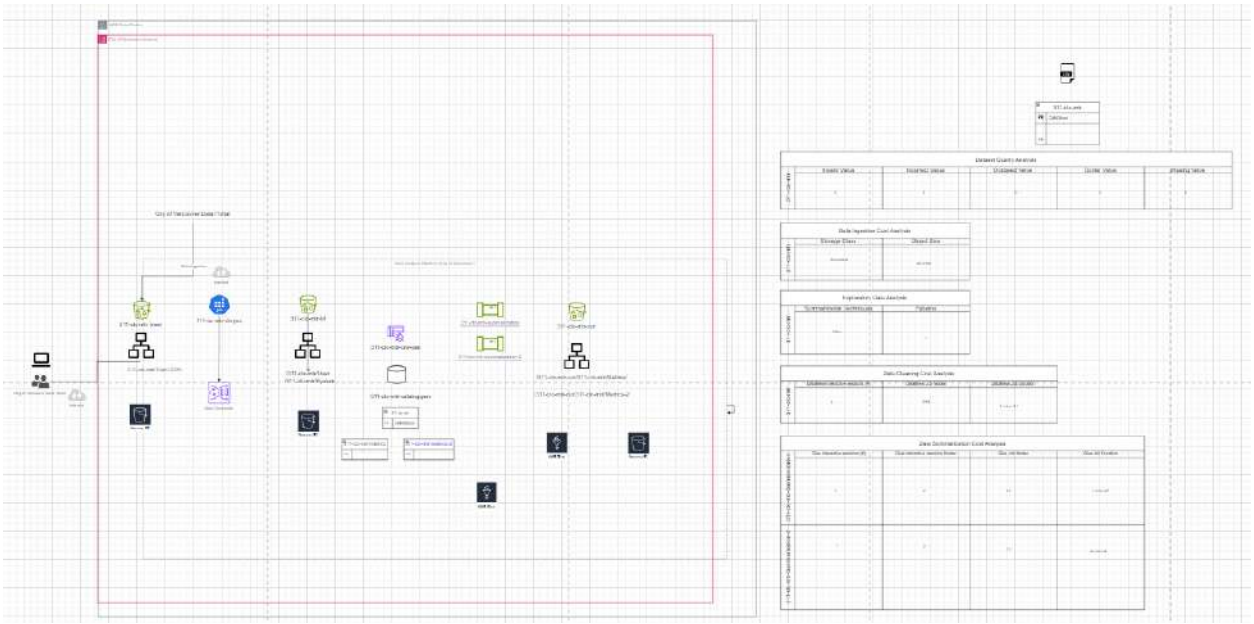
On the left side of the draw.io file, I downloaded the datasets from the City of Vancouver Data Portal and then implemented the data ingestion by creating a raw bucket and sub-folders in Amazon S3 and then uploaded this dataset into this raw bucket. Next, I ran a job in AWS Glue

Databrew and created transfer buckets in AWS S3 and sub-folders including the system and the user to receive the cleaning data. Next, I achieved the Crawlers to prepare for the data catalog, which was classified by two metrics. Finally, I created a new bucket in AWS S3 to receive the summarization data, which is in AWS Glue, into those two metrics.

Additionally, on the right side of the draw.io file, it shows the dataset information. The dataset was named 311-ctc-mtr, and its primary key is Call-Offered. Next, the data quality is all good because there are not any invalid values, incorrect values, and so on. When proceeding with data ingestion, the storage class is standard, and the object size is 20 KB. I only use filter to do the EDA because this dataset only shows numerical values. Next, when proceeding with data cleaning cost analysis, the databrew interactive session is 1, and databrew job notes are 5+5, and databrew job duration is 1 minute 31 seconds. When proceeding with data summarization cost analysis, the glue interactive session, the glue interactive session node, and the glue job node are the same for those two summarizations, which are 1, 2, and 10, respectively, but the difference is the glue job duration. The first summarization ran for 1 minute 47 seconds, and the second one ran for 2 minutes 3 seconds.

Figure 26

All Steps Using draw.io

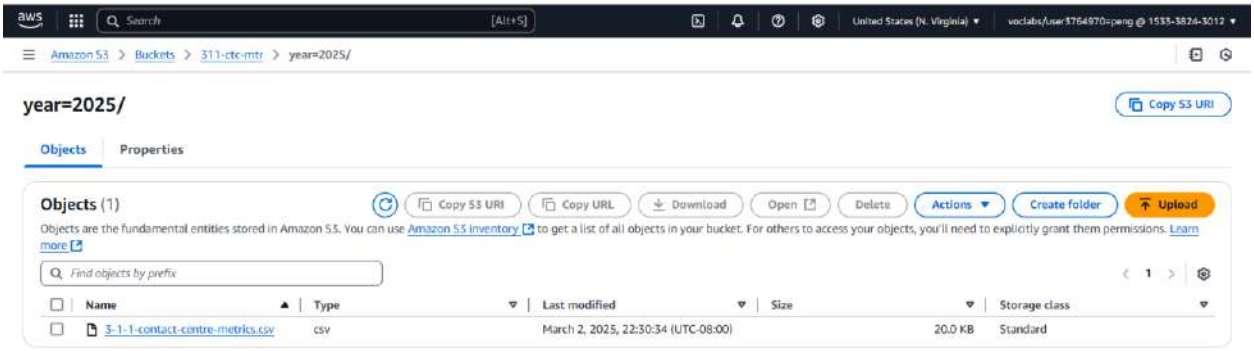


Note. The screenshot shows all the steps using the draw.io file.

Step 1: Data Ingestion

Figure 27

S3 Bucket Building



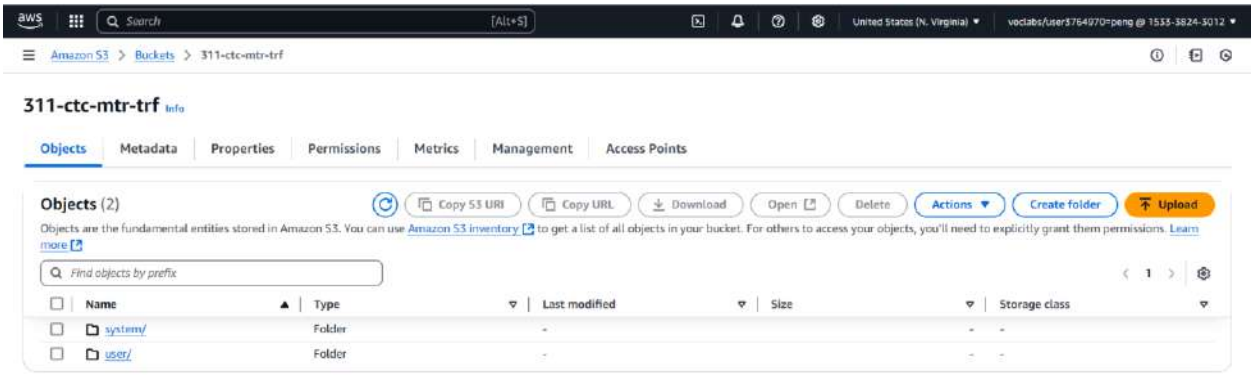
Note. The screenshot shows the S3 bucket. Source from AWS.

Explanation: I downloaded the datasets from the City of Vancouver data portal and created buckets called 311-ctc-mtr as my raw bucket, and then I also created sub-folders called year=2025 which received my dataset when uploading.

Step 2: Data Profiling

Figure 28

Transfer Buckets Building

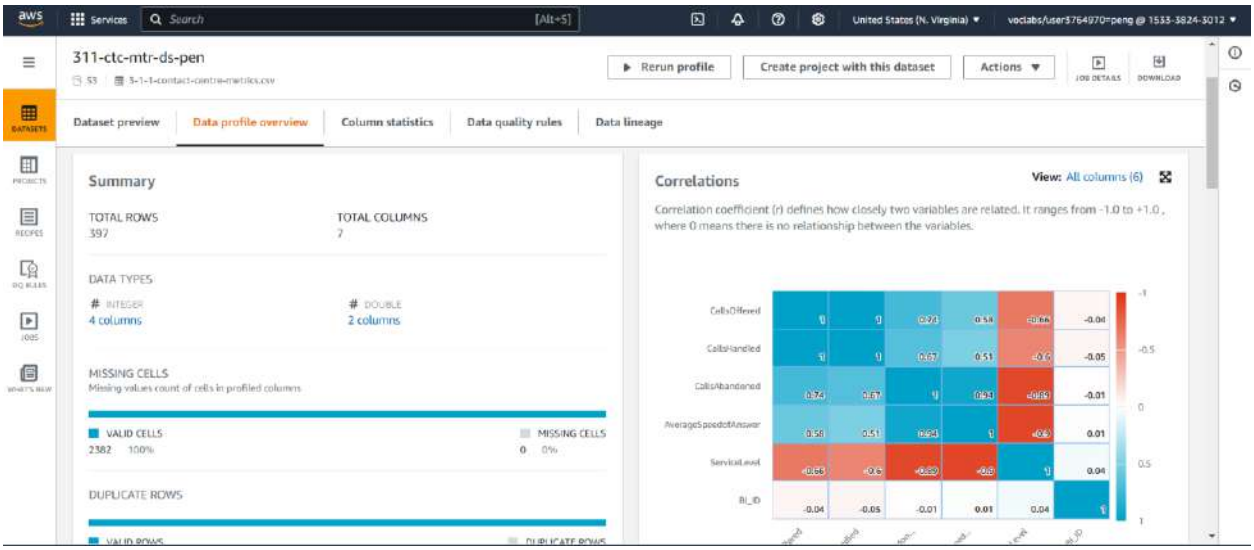


Note. The screenshot shows the transfer bucket has already been built. Source from AWS.

Explanation: I created another bucket called 311-ctc-mtr-trf, which is for data transfer. After that, I created the sub-folders, including system and user, for different uses.

Figure 29

Data Profile Overview



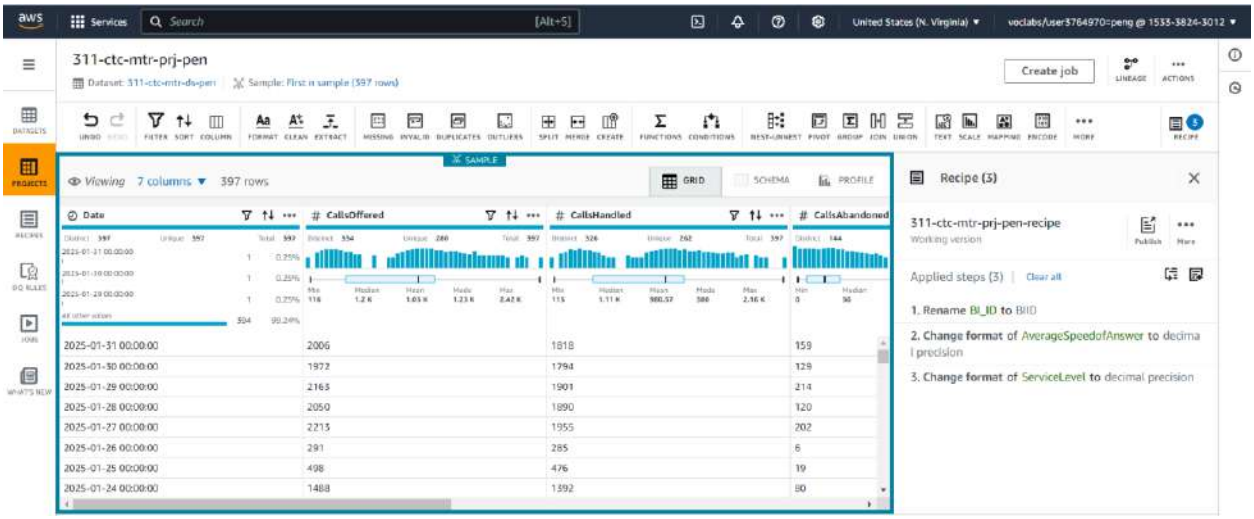
Note. The screenshot shows the data profile overview. Source from AWS.

Explanation: This is the data profile screenshot after running the job.

Step 3: Data Cleaning

Figure 30

Data Cleaning

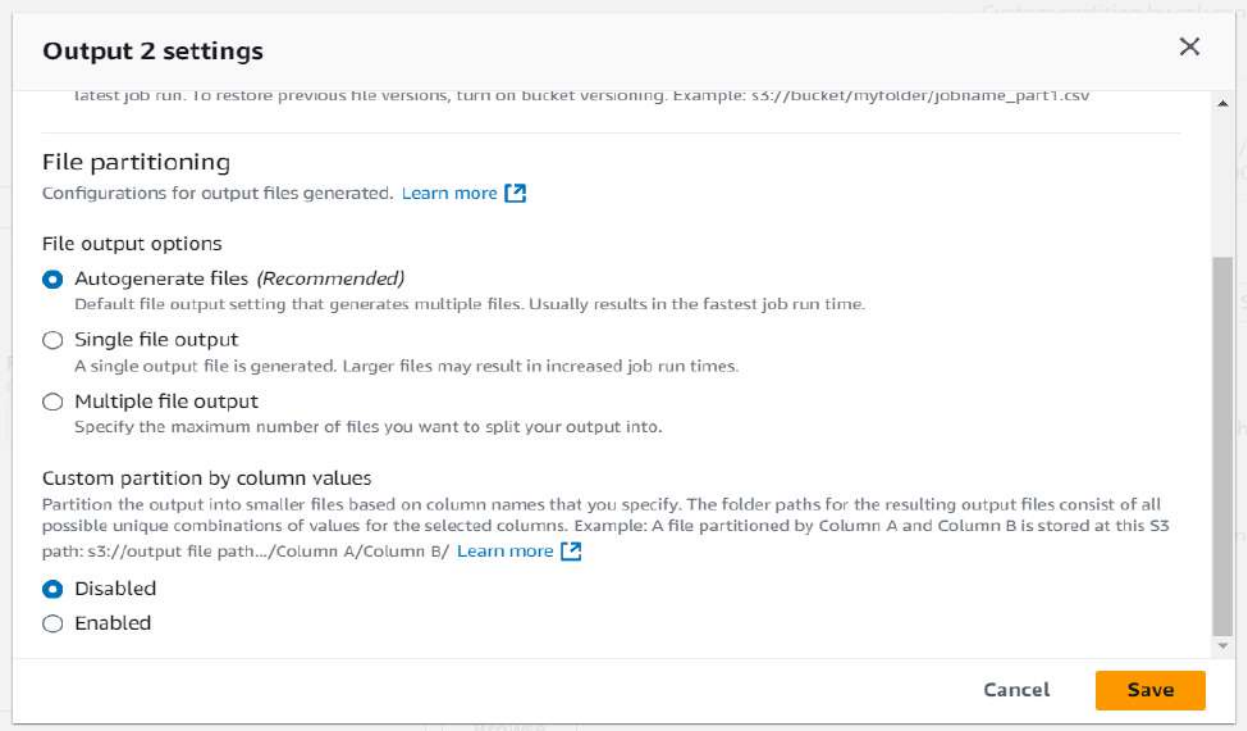


Note. The screenshot shows the data cleaning. Source from AWS.

Explanation: I cleaned the datasets, including renaming, and changing the format to decimal precision of 2 values. That is because it is more directly visual to analyze and keep the data consistently visually.

Figure 31

Output 2 settings

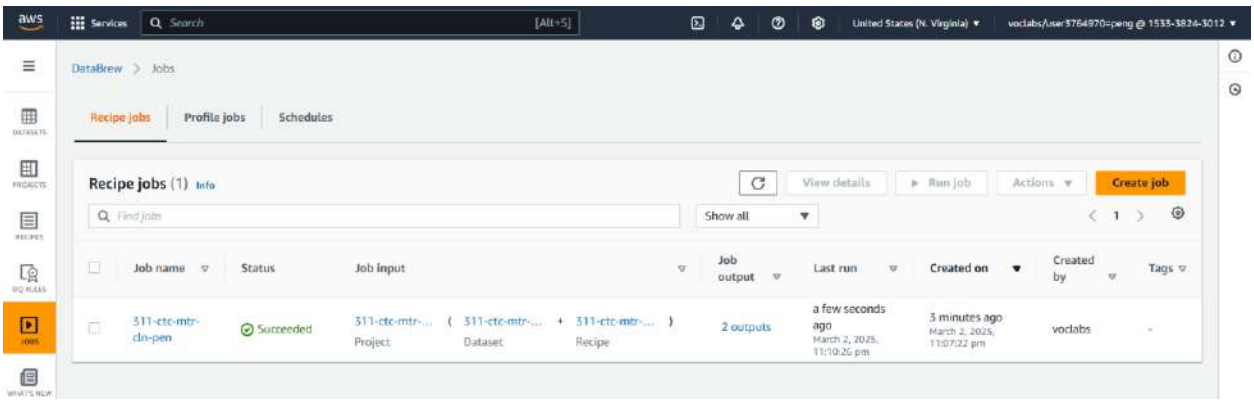


Note. The screenshot shows output 2 settings and no selection of "Enabled" for the custom partition. Source from AWS.

Explanation: When proceeding with data cleaning, I didn't choose custom partition as enabled because there is no column data in my datasets, so it is not effective when classifying the numerical values.

Figure 32

Recipe Jobs

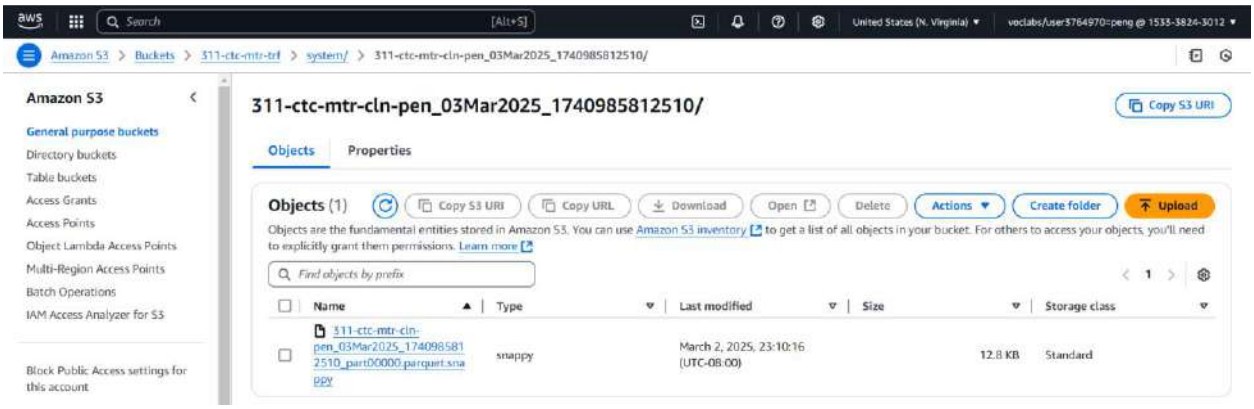


Note. The screenshot shows recipe jobs. Source from AWS.

Explanation: when the job is run successfully, the cleaning data can be found in system and user buckets.

Figure 33

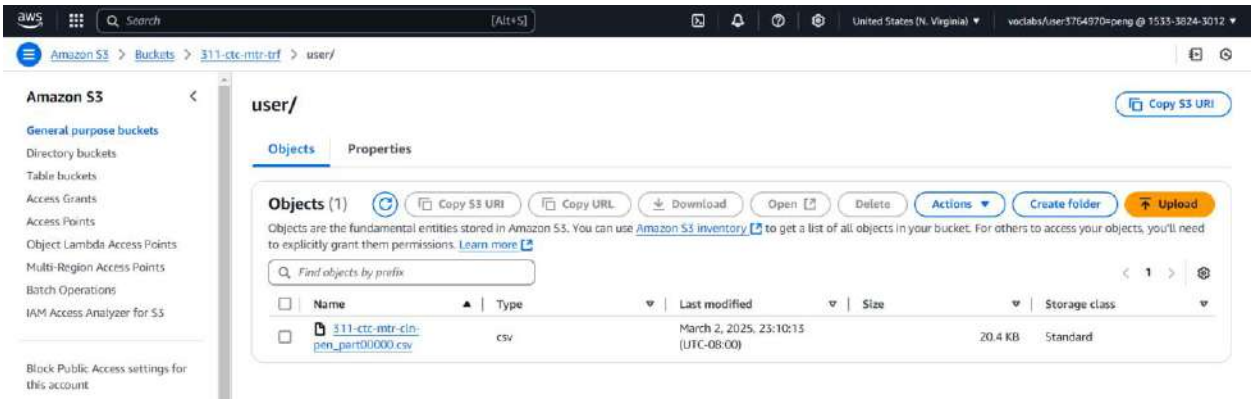
System Bucket for Cleaning Data



Note. The screenshot shows recipe jobs. Source from AWS.

Figure 34

User Bucket for Cleaning Data



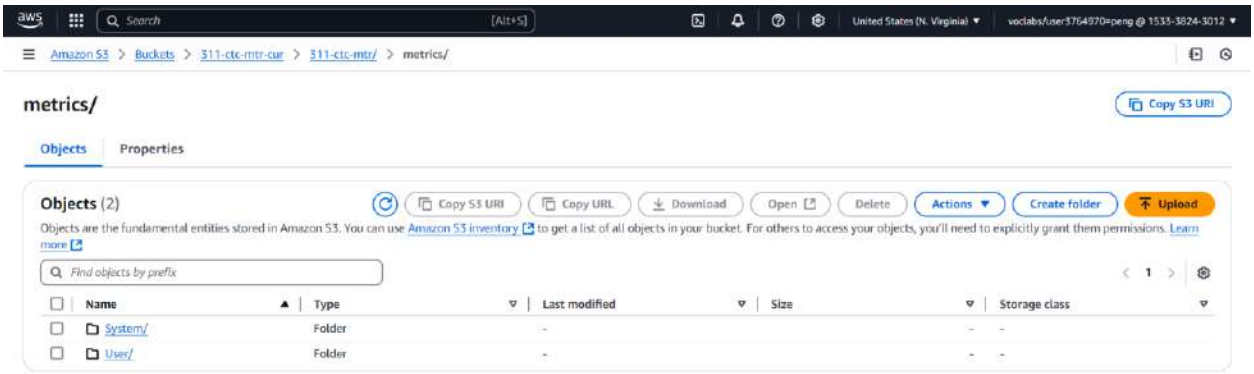
Explanation: Now, the cleaning data has already been in the transfer bucket, which will be used for data cataloging.

Step 4

Data Cataloging

Figure 35

Metrics Folders Building

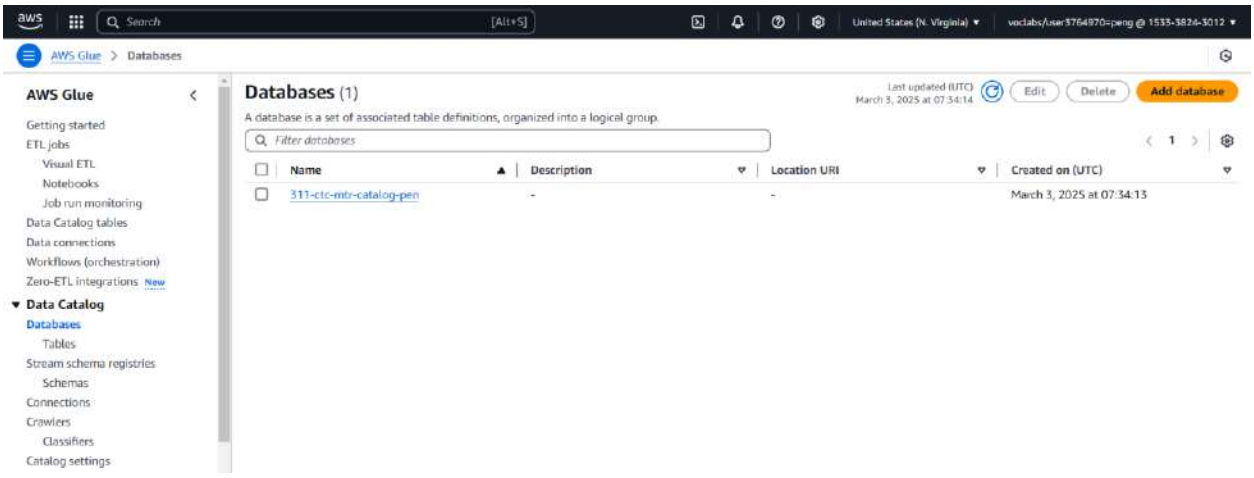


Note. The screenshot shows the metrics folder building. Source from AWS.

Explanation: I created a new bucket called 311-ctc-mtr-cur and sub-folders for metrics including system and user folders, which will be stored for summarization.

Figure 36

Databases Building

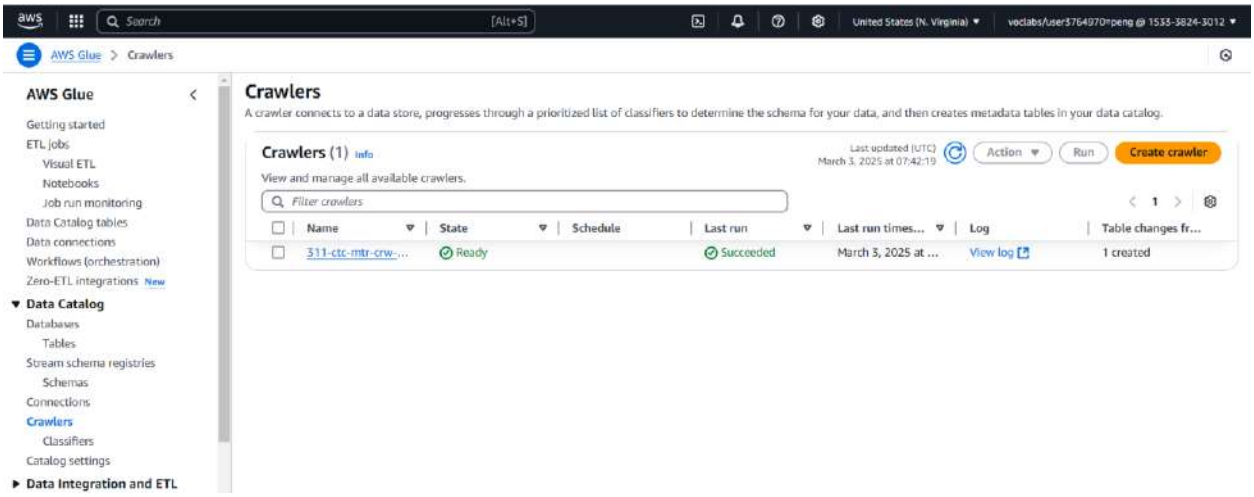


Note. The screenshot shows the database building. Source from AWS.

Explanation: I created a database called 311-ctc-mtr-catalog-pen to receive the cleaning data.

Figure 37

Crawlers Building

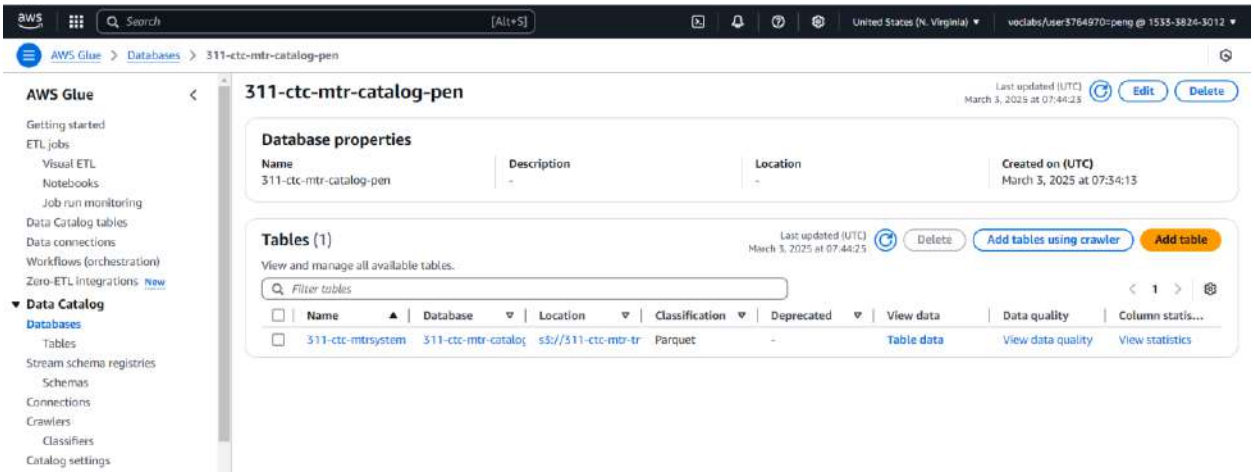


Note. The screenshot shows the crawler building. Source from AWS.

Explanation: I created a crawler to prepare for the data catalog.

Figure 38

Transfer Cleaning Data to Databases for catalog



Note. The screenshot shows the cleaning data has already been transferred to datasets for catalog.

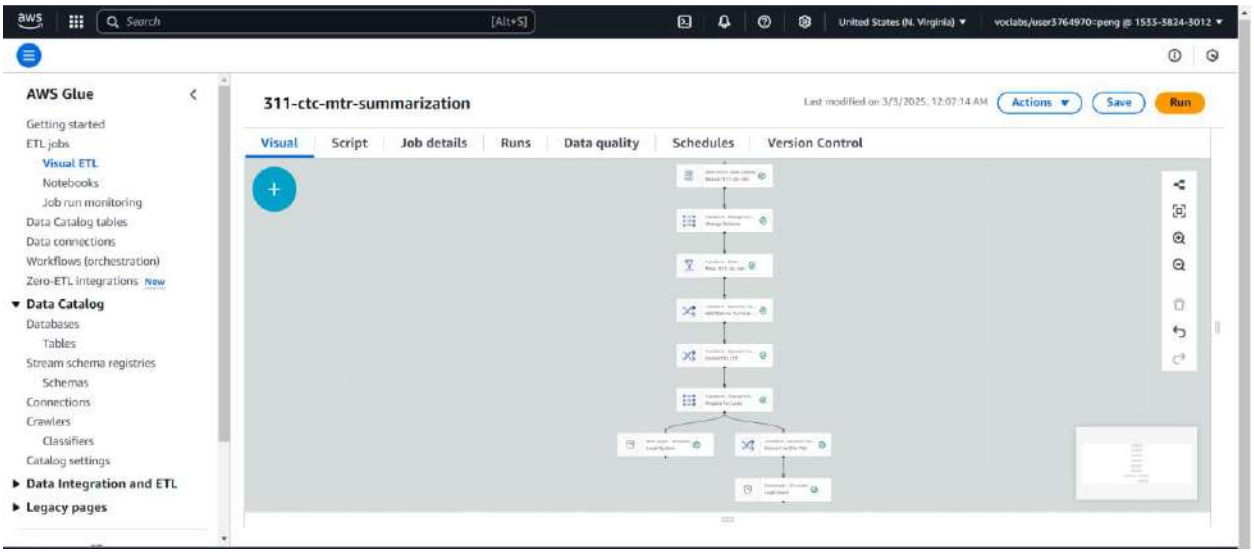
Source from AWS.

Explanation: When the Crawler was ready, the cleaning data was transferred to databases for cataloging and prepared for summarization.

Step 5: Data Summarization

Figure 39

ETL pipeline 1

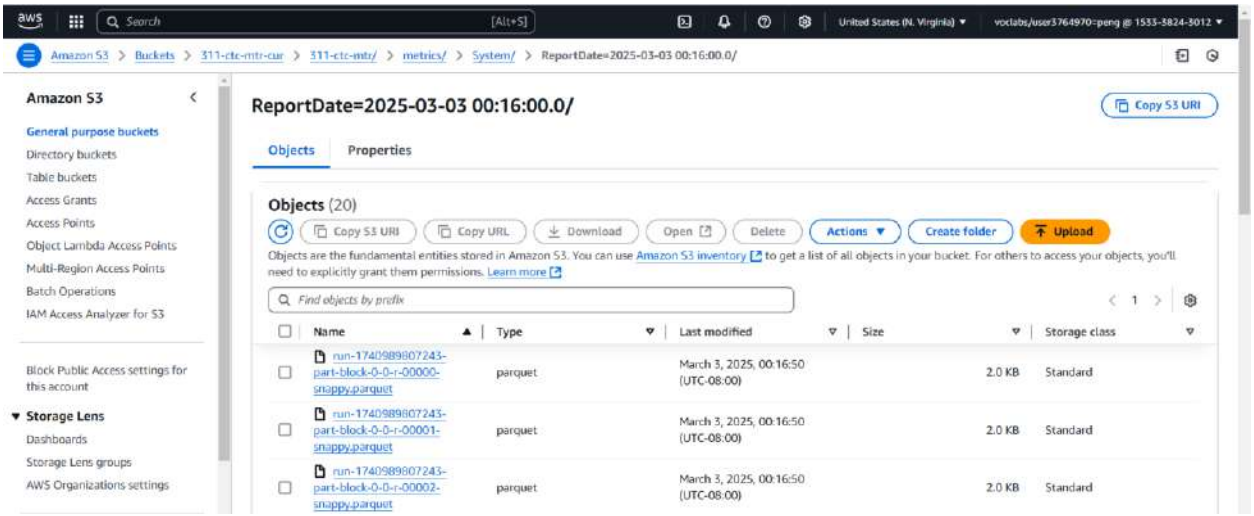


Note. The screenshot shows dataset summarization. Source from AWS.

Explanation: I extracted the data from the system, and then I didn't change the schema because all the data was good. Next, I did a filter for the key to the average speed of the answer less than 100. Next, I added the date to the summary report and changed the date to local time in Vancouver, which dropped the date and the original report date. Next, I loaded the data into the system. Meanwhile, I loaded the data into users by CSV file by converting one file.

Figure 40

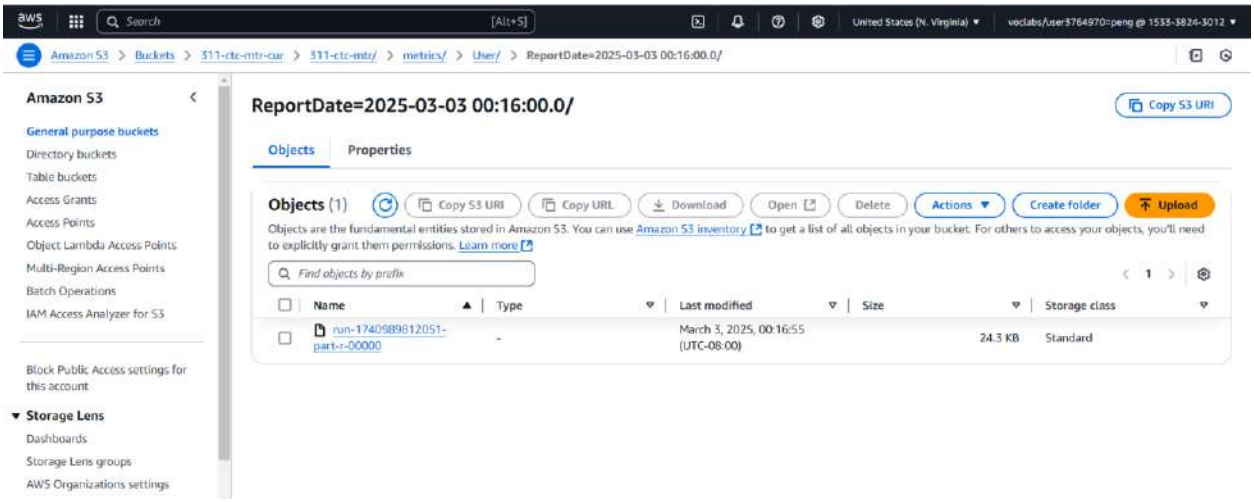
Metrics Report for System



Note. The screenshot shows the metrics report for the system. Source from AWS.

Figure 41

Metrics Report for User



Note. The screenshot shows the metrics report for users. Source from AWS.

Explanation: When I ran my pipeline successfully, the report data was shown in the system and users, respectively, in order for summarization. For example, users can download the final dataset after cleaning.

Figure 42

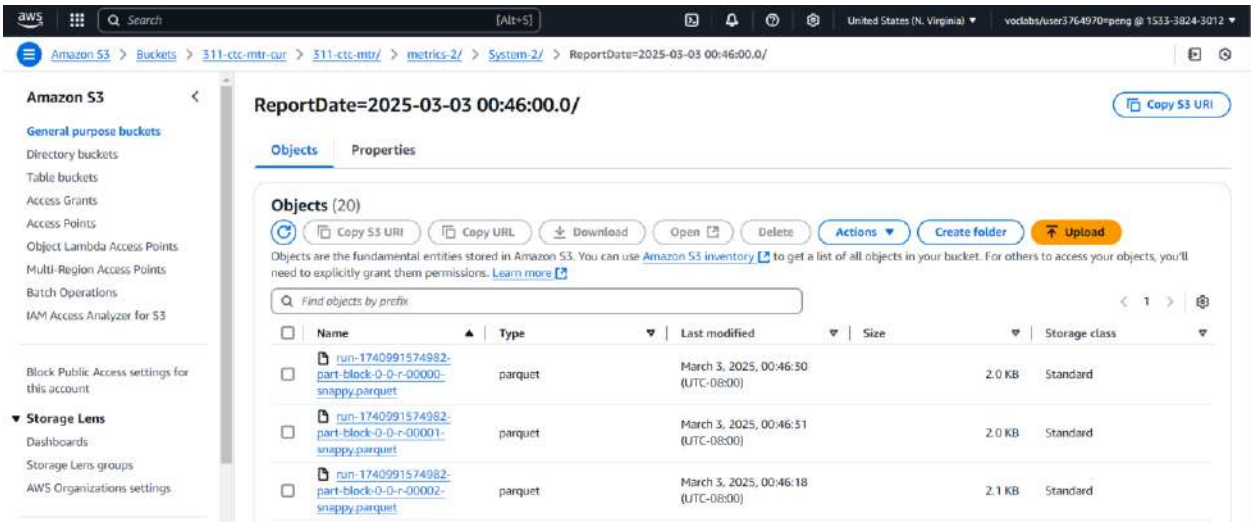
ETL pipeline 2

Note. The screenshot shows dataset summarization-2. Source from AWS.

Explanation: I extracted the data from the system, and then I did a filter for the key to the call abandoned less than 100. Next, I added the date to the summary report and changed the date to local time in Vancouver, which dropped the date and the original report date. Next, I loaded the data into the system. Meanwhile, I loaded the data into users by CSV file by converting one file.

Figure 43

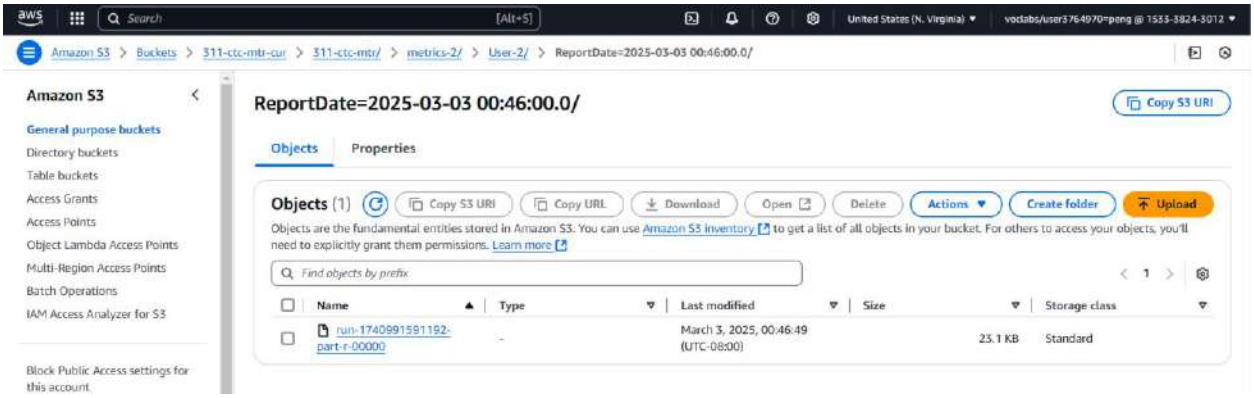
Metrics Report for System - 2



Note. The screenshot shows the metrics report for the system - 2. Source from AWS.

Figure 44

Metrics Report for User – 2



Note. The screenshot shows the metrics report for users – 2. Source from AWS.

Explanation: When I ran my pipeline successfully again, the report data was shown in the system and users, respectively, in order for summarization. For example, users can download the final dataset after cleaning.

DAP Design and Implementation (Individual work – Muhammad Zulqarnain Shahzad)

Descriptive Analysis

Dataset Chosen

- Business Licences – Consulting and Management Services dataset from the City of Vancouver Open Data Portal.
- Link: Consulting and Management Services 2024

Why This Dataset

- It meets the required volume (approximately 500 rows) for the assignment.

- It contains key information on employee counts and license fees, which is crucial for understanding the performance of different management consulting subdomains.

Goal

To analyze the performance of management consulting subdomains by determining which ones have the highest employee counts and evaluating key cost metrics (such as license fee paid).

Analytical Questions

- Which management consulting subdomain has the highest number of employees?
- What is the maximum, minimum, and average employee counts across these subdomains?
- What is the average license fee paid by organizations in each subdomain?

Metrics

Employee Count Metrics

- Maximum employee count per subdomain
- Minimum employee count per subdomain
- Average employee count per subdomain

License Fee Metric

- Average license fee paid

Overview: What I Did and Why I Did It

In this project, I downloaded the dataset from the City of Vancouver data portal website and selected the December 2024 data, as it contained approximately 500 rows the required size for this project. I then uploaded the original CSV file (which uses a semicolon as its delimiter) to an S3 bucket named “bl-cms-dec-2024-zul” to keep the raw data intact. I did this to ensure that the dataset would be properly analysed later on. Next, I used AWS Glue DataBrew to profile and clean the data, where I corrected the delimiter, standardized column names, reformatted date fields, and dropped unnecessary columns. This cleaning step made the dataset consistent and easier to work with. After that, I ran an AWS Glue Crawler to automatically create a database schema, making it simpler to query the cleaned data with Athena. Finally, I built a single ETL pipeline in AWS Glue Studio to transform, filter, and aggregate the data (for example, to calculate how many licenses were issued or such as the number of employees > 50) and stored the final results in the transform, system, and user folders. This streamlined process ensures that the data is quickly accessible, neatly organized, and ready for further analysis. For a more detailed step-by-step breakdown, please refer to the subsequent sections of this document.

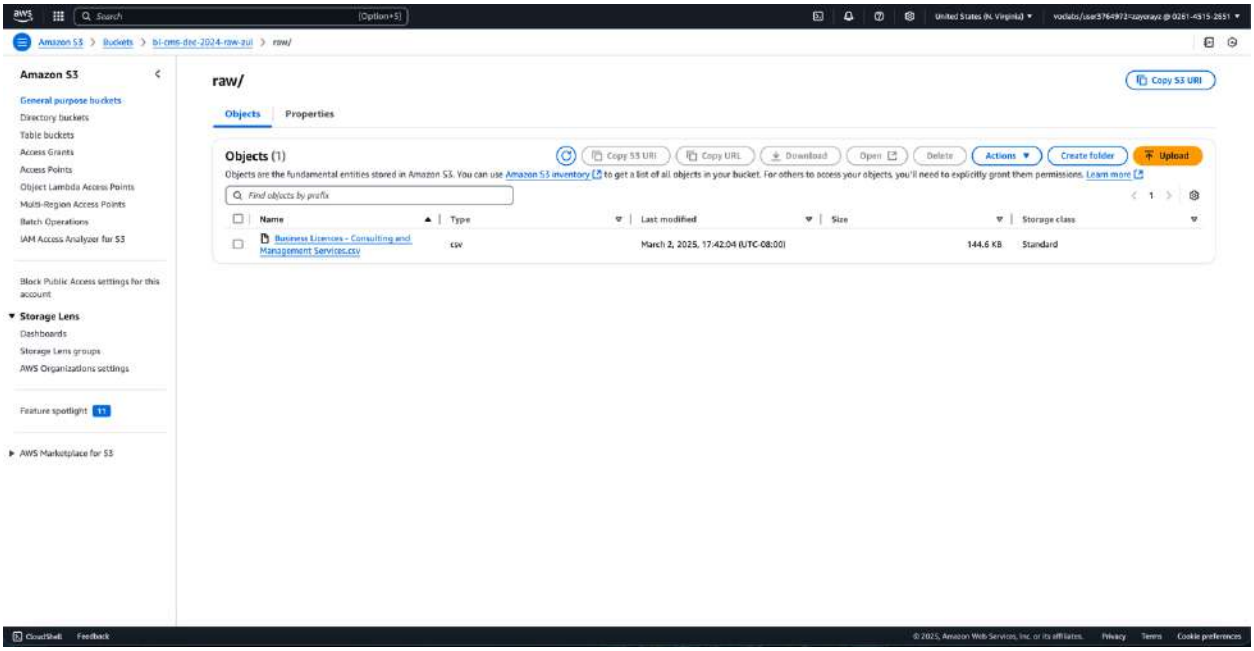
Step 1: Data Ingestion

I started by creating an S3 bucket named “bl-cms-dec-2024-zul” to serve as the centralized repository for my project data. Within this bucket, I created a folder named “raw” where I uploaded the original CSV file. The file contains Business Licenses – Consulting and Management Services data and uses a semicolon as its delimiter. It was essential to preserve this detail during ingestion so that the data would be parsed correctly in later steps. I verified the file

properties (such as file size, upload date, etc.) in the S3 console to ensure that the raw data was intact and accessible.

Figure 45

S3 Bucket – Raw Folder and Uploaded CSV



Note. Screenshot from AWS S3 console showing bucket “bl-cms-dec-2024-zul” with the “raw” folder containing the original Business Licences – Consulting and Management Services CSV file. Source: AWS, *self-work*

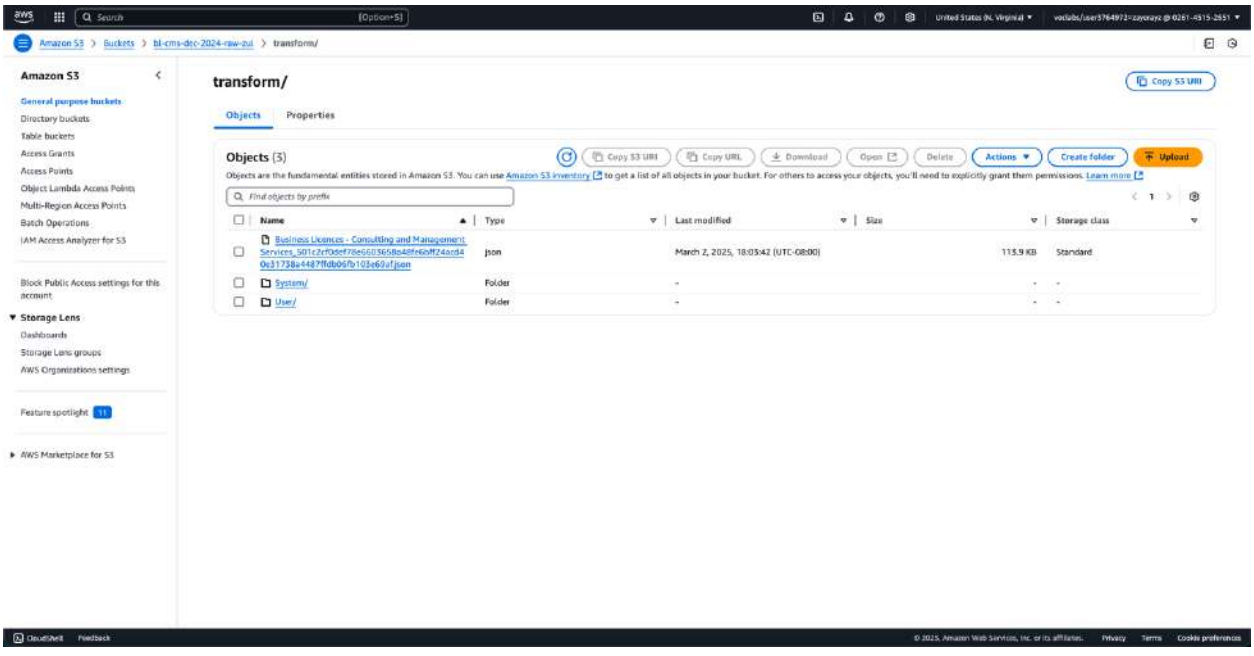
I show my S3 bucket named “bl-raw-zul” with the “raw” folder clearly visible. Here, I uploaded the original CSV file containing Business Licenses – Consulting and Management Services data. Since the CSV uses a semicolon as its delimiter, I made sure to preserve that detail during ingestion. I also verified the file’s properties such as size and upload date to confirm that the data was intact before moving on.

Step 2: Data Profiling

Next, I set up an AWS Glue DataBrew project to profile the dataset. Initially, I encountered an issue where the data was not splitting into columns correctly because the default CSV delimiter was a comma. I corrected this by configuring the project to use a semicolon as the delimiter exactly as specified on the data portal. With the correct configuration, the data preview in DataBrew displayed the columns properly, and summary statistics (data types, null counts, etc.) were generated. This profiling step was crucial to identify any data quality issues and to plan the cleaning steps accordingly.

Figure 46

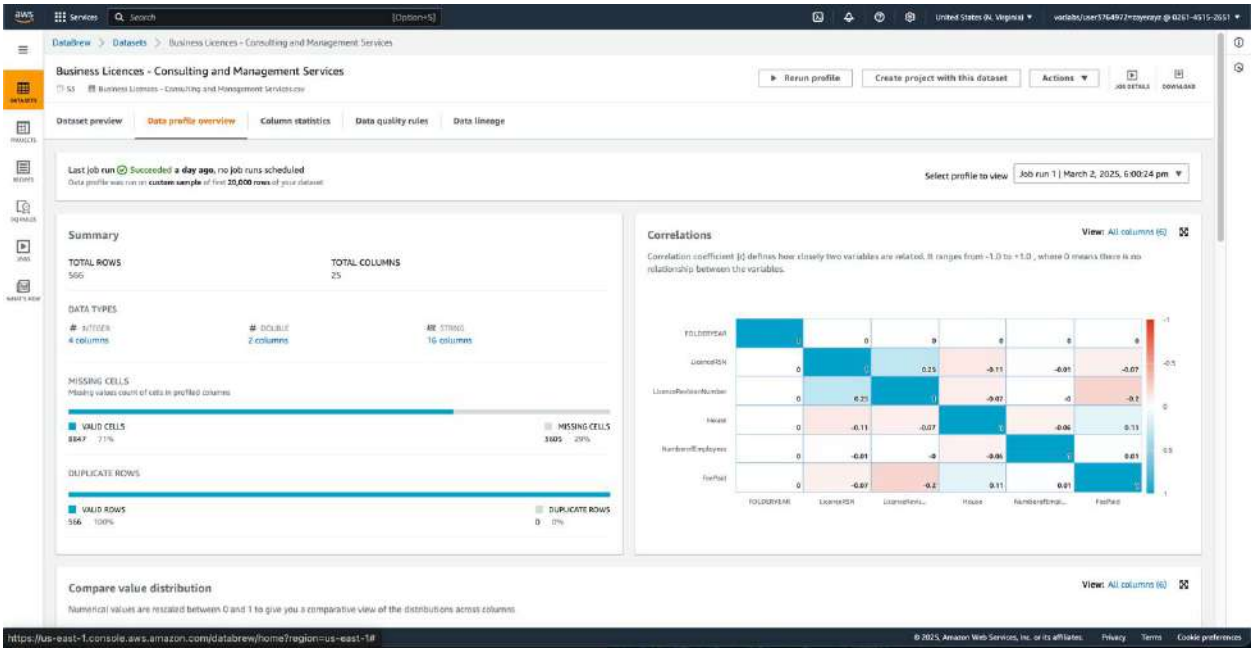
Transform



Note. Screenshot from AWS S3 console showing Transform bucket + System and User folders. Source: AWS, *self-work*

Figure 47

DataBrew Project – Data Preview



Note. Screenshot showing the data preview in DataBrew with summary statistics (data types, null counts) confirming proper column splitting. Source: AWS, *self-work*

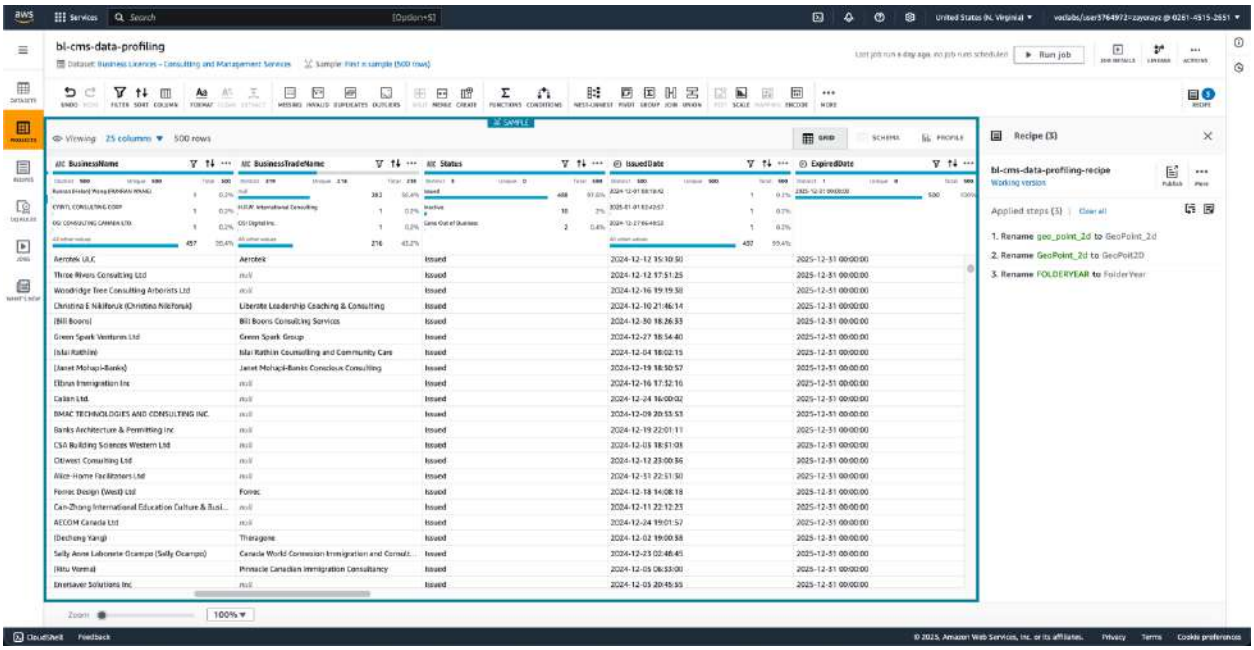
Here, I set up an AWS Glue DataBrew project to profile my dataset. Initially, the data did not split into columns because the default delimiter was a comma. After configuring the project to use a semicolon, the preview now shows the columns correctly. This step was key to identifying data quality issues early on. This shows the summary statistics generated by DataBrew, including data types and null counts. The view reassured me that all columns are correctly interpreted, and I'm set to proceed to cleaning.

Step 3: Data Cleaning

Once the data profile was verified, I created a cleaning recipe in DataBrew. In the recipe editor, I standardized the column names to be more descriptive and reformatted the date fields into the “yyyy-mm-dd” format. I also reviewed all columns and dropped those that were not needed for the analysis (for example, columns like “geom” and “geopoit2d”). This process streamlined the dataset, reducing unnecessary clutter and improving processing efficiency. The cleaned data was then output to two separate locations: a “system” folder for technical verification and a “user” folder for the final analysis-ready data.

Figure 48

DataBrew Recipe – Cleaning Steps Overview

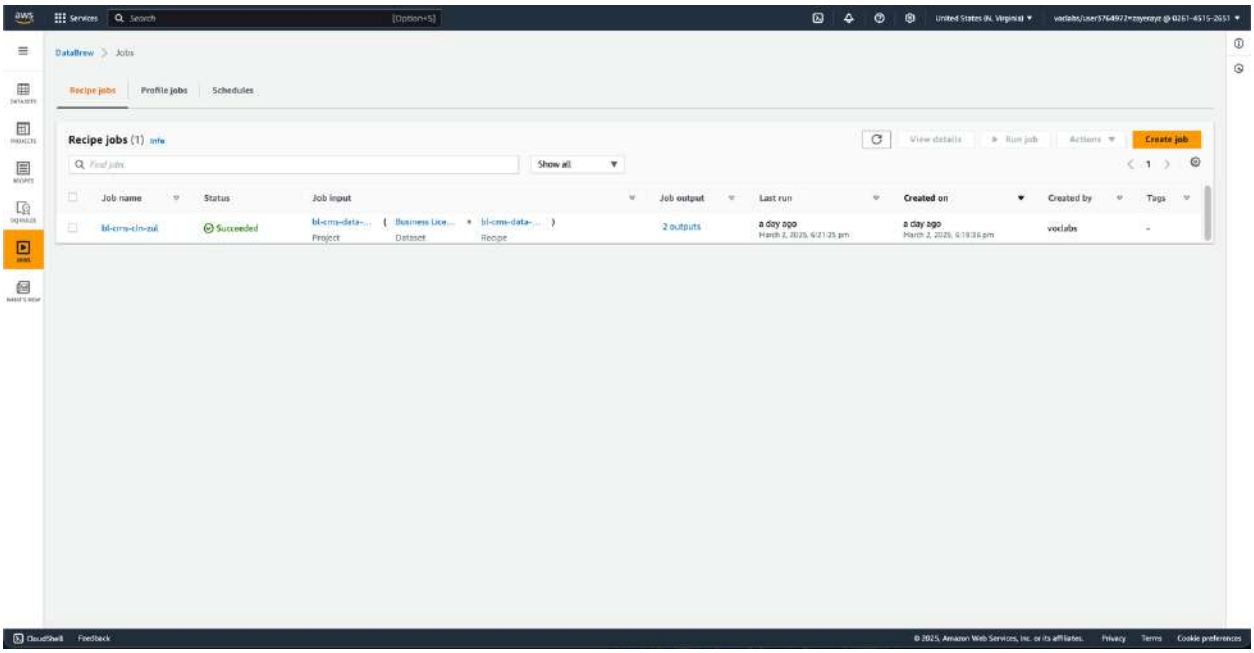


Note. Screenshot of the cleaning recipe in DataBrew where column names are standardized and date fields reformatted to “yyyy-mm-dd,” with unnecessary columns dropped. Source: AWS, *self-work*

In this image, you can see the cleaning recipe in DataBrew. I standardized the column names to be more descriptive and reformatted date fields into the “yyyy-mm-dd” format. Additionally, I dropped unnecessary columns (e.g., geographic coordinate fields) to simplify the dataset and improve processing efficiency.

Figure 49

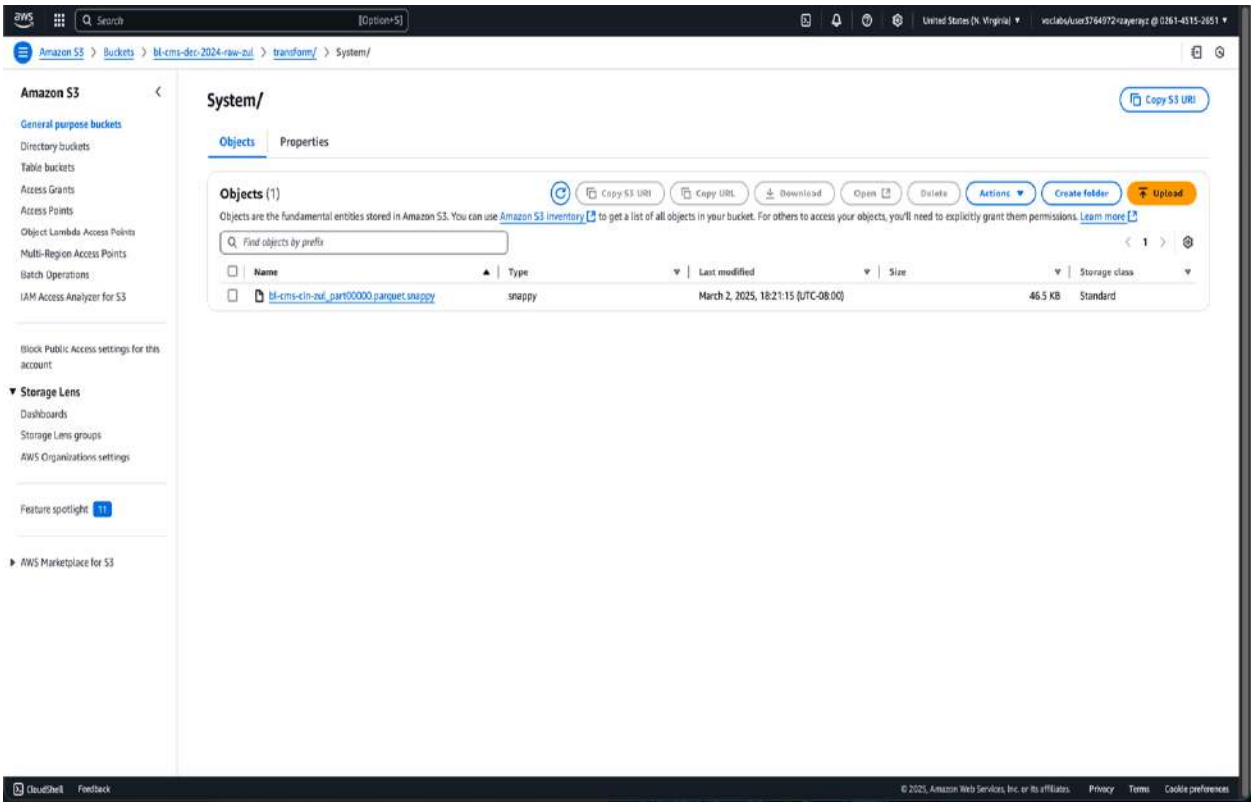
Recipe Job Successful



Note. Screenshot indicating the successful execution of the cleaning recipe job in DataBrew. Source: AWS, *self-work*

Figure 50

Cleaned Data Output - System Folder

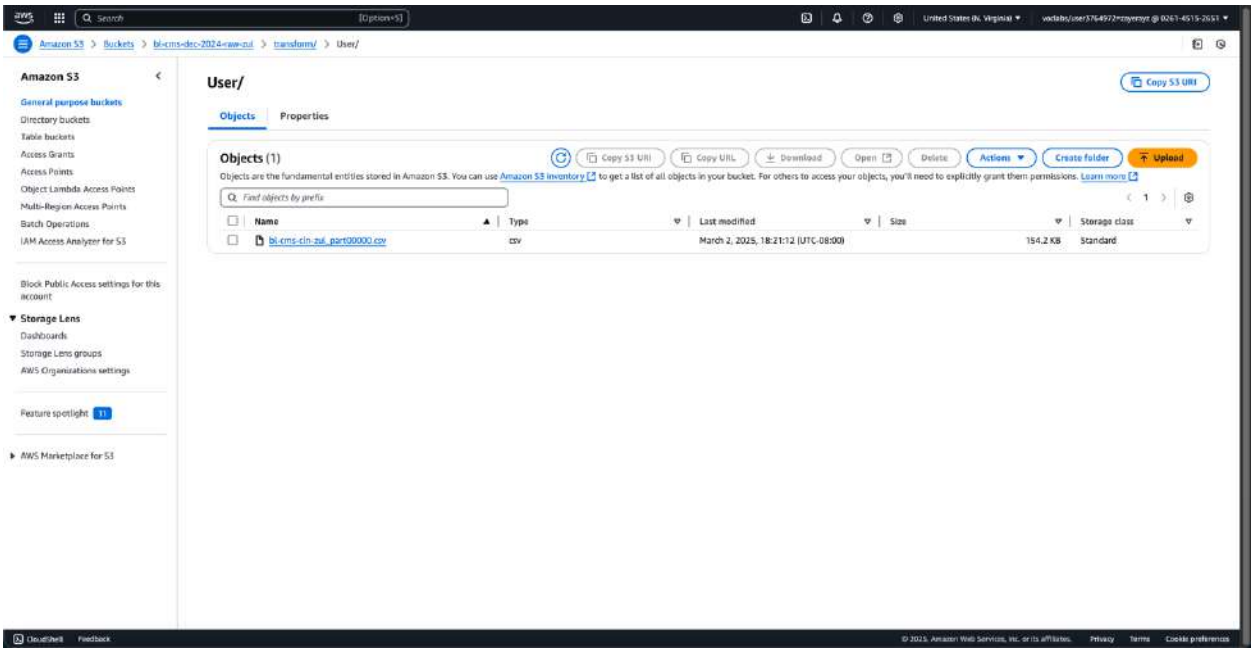


Note. Screenshot from AWS S3 showing the cleaned dataset stored in the system folder for technical verification. Source: AWS, *self-work*

This screenshot displays the cleaned dataset as it appears in the system folder. This output is used for technical verification; you can see that only the essential columns remain and that the formatting is consistent.

Figure 51

Cleaned Data Output - User Folder



Note. Screenshot from AWS S3 showing the final cleaned CSV stored in the user folder, ready for analysis. Source: AWS, *self-work*

Here, the final cleaned CSV is stored in the user folder. This version is what I will use for analysis, ensuring that all extraneous data has been removed, and the dataset is ready for the next phase.

Step 4

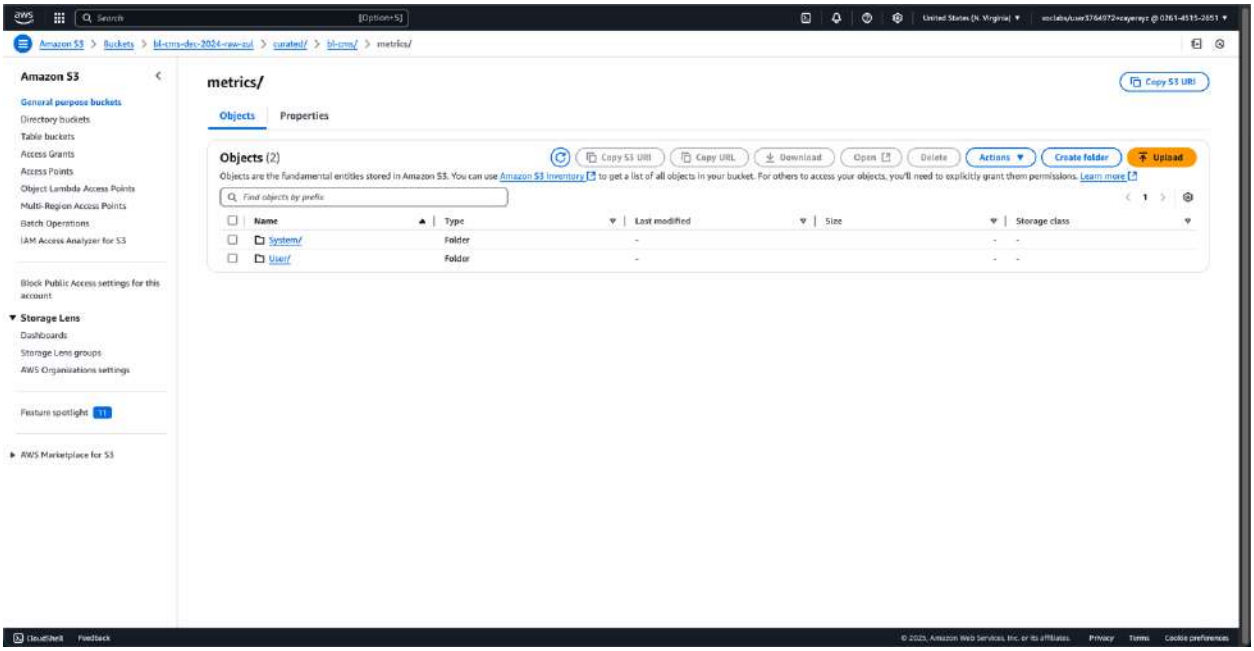
Data Cataloging

After cleaning, I used AWS Glue Crawler to catalog the dataset. I configured the crawler to scan the cleaned data stored in the designated folder (it could be the “user” folder or a dedicated cataloging folder, depending on your setup). The crawler automatically created a database in AWS Glue, which organizes the data with a well-defined schema. This step is

essential for enabling efficient querying in AWS Athena and ensuring that the dataset is accessible for further ETL processing.

Figure 52

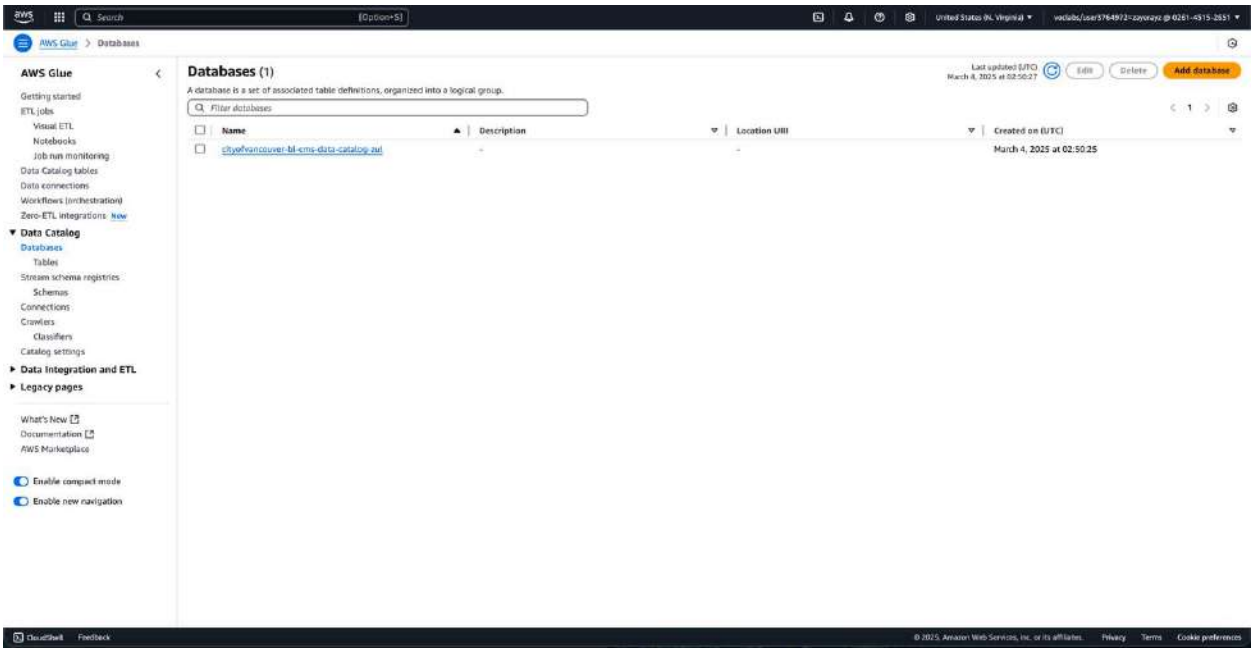
Metrics Folder



Note. Screenshot displaying the metrics folder created for further processing and cataloging. Source: AWS, *self-work*

Figure 53

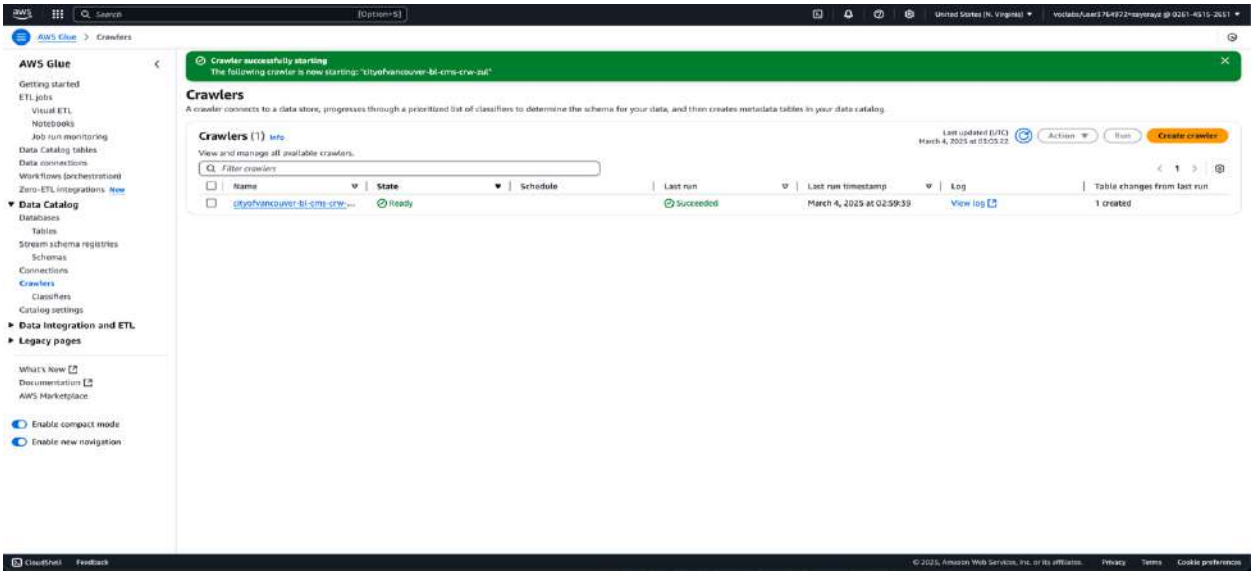
AWS Glue Database – Cataloging the Cleaned Data



Note. Screenshot showing the database created by AWS Glue Crawler that catalogs the cleaned data from the user folder. Source: AWS, *self-work*

Figure 54

AWS Glue Crawler

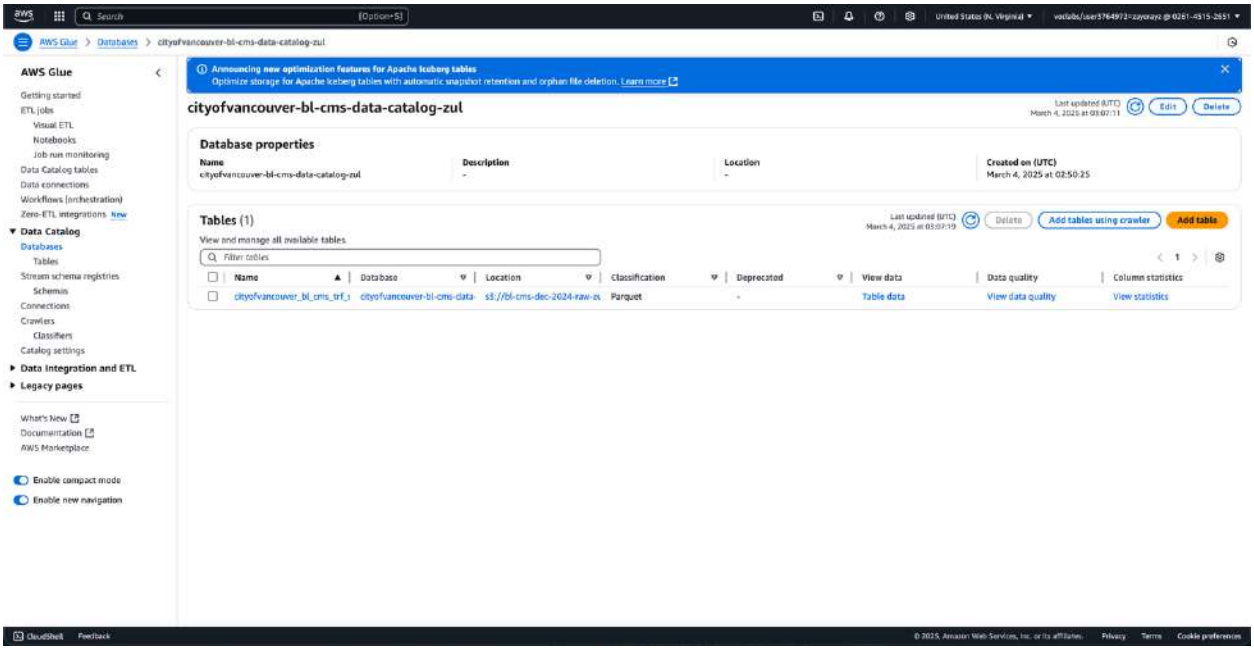


Note. Screenshot capturing the Glue Crawler scanning the cleaned data to build the database schema. Source: AWS, *self-work*

After cleaning, I ran an AWS Glue Crawler on the cleaned dataset stored in the user folder. This screenshot shows the crawler in action as it scans the data and automatically creates a database with a well-defined schema.

Figure 55

Cataloged Database and Table Schema in AWS Glue



Note. Screenshot of the final database and table schema generated by the crawler, confirming organized data for querying. Source: AWS, *self-work*

This image displays the resulting database and table schema from the crawler. It confirms that the data is organized and accessible for querying.

Step 5

ETL Pipeline and Data Summarization

For the transformation and summarization phase, I built a single, continuous ETL pipeline in AWS Glue Studio. Unlike some projects that employ two separate pipelines for multiple metrics, my analysis was accomplished with one streamlined flow. The pipeline starts with the “Prepare for Load” node, which connects directly to an output node configured to write the transformed data into the “transform” folder of my S3 bucket.

Within this pipeline, I applied several key transformation steps:

• **Filtering:** I filtered out rows that did not meet the specific criteria relevant to my analysis.

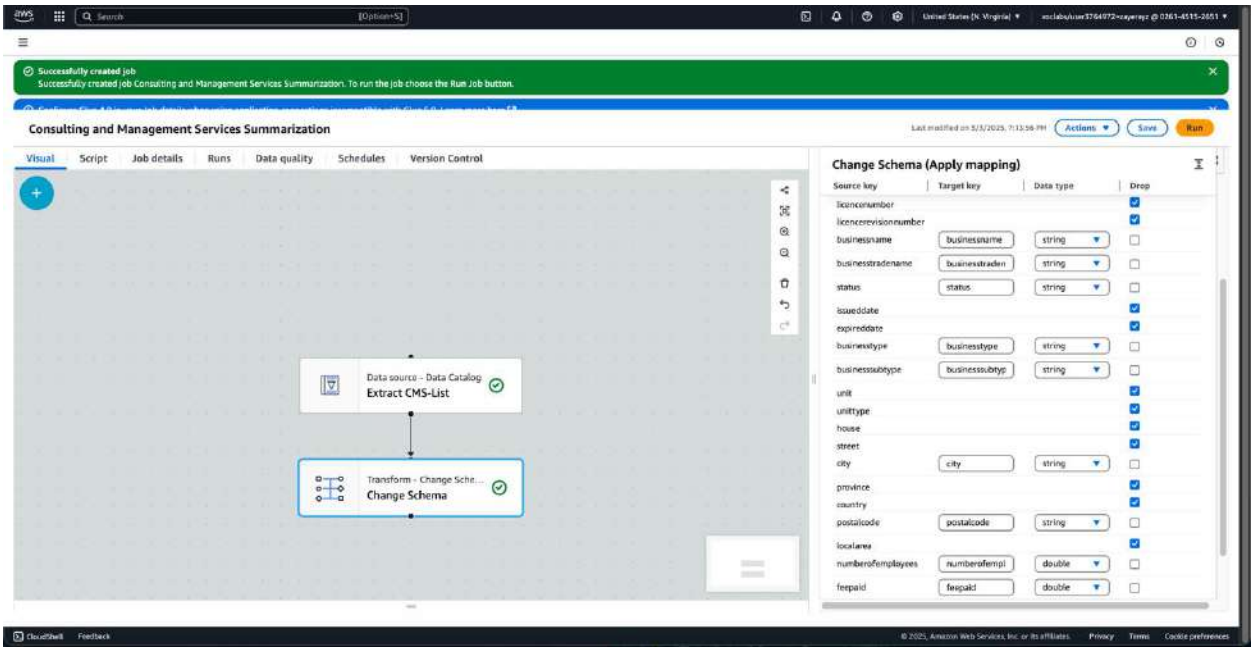
• **Aggregation:** I performed aggregation operations to compute my primary metric (for instance, the total number of licenses issued or the average processing time).

• **Schema Changes:** I re-mapped column names and dropped fields that were unnecessary, ensuring that the final output contained only the relevant information.

The final output of the ETL pipeline is then written to designated folders. In my case, the output is available in both the “system” folder (for technical verification) and the “user” folder (as the final, summarized dataset). Additionally, I ensured that the final data appears in the “transform” folder if that’s where intermediate or final transformation files are meant to be stored according to the project structure.

Figure 56

Schema Changes and Final Mapping

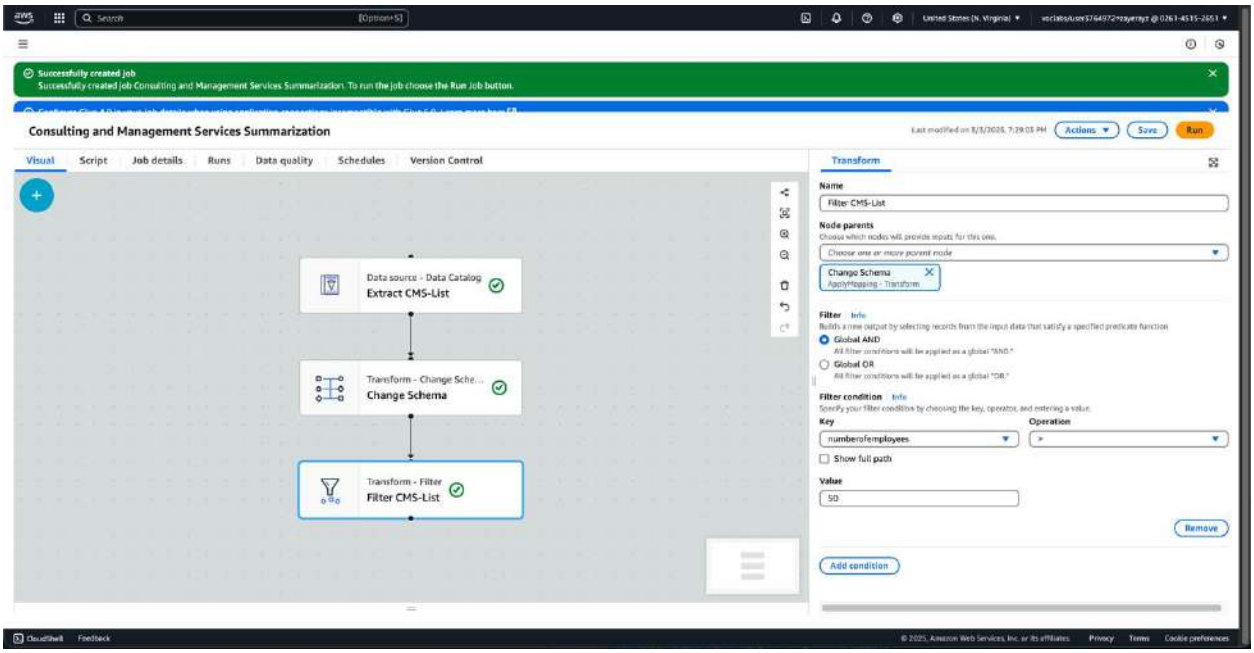


Note. Screenshot of the final database and table schema generated by the crawler, confirming organized data for querying. Source: AWS, *self-work*

This screenshot focuses on the schema change step, where I re-mapped column names and dropped unnecessary fields. These adjustments ensure that the final output includes only relevant information and is easy to interpret.

Figure 57

Detailed Transformation - Filtering and Aggregation

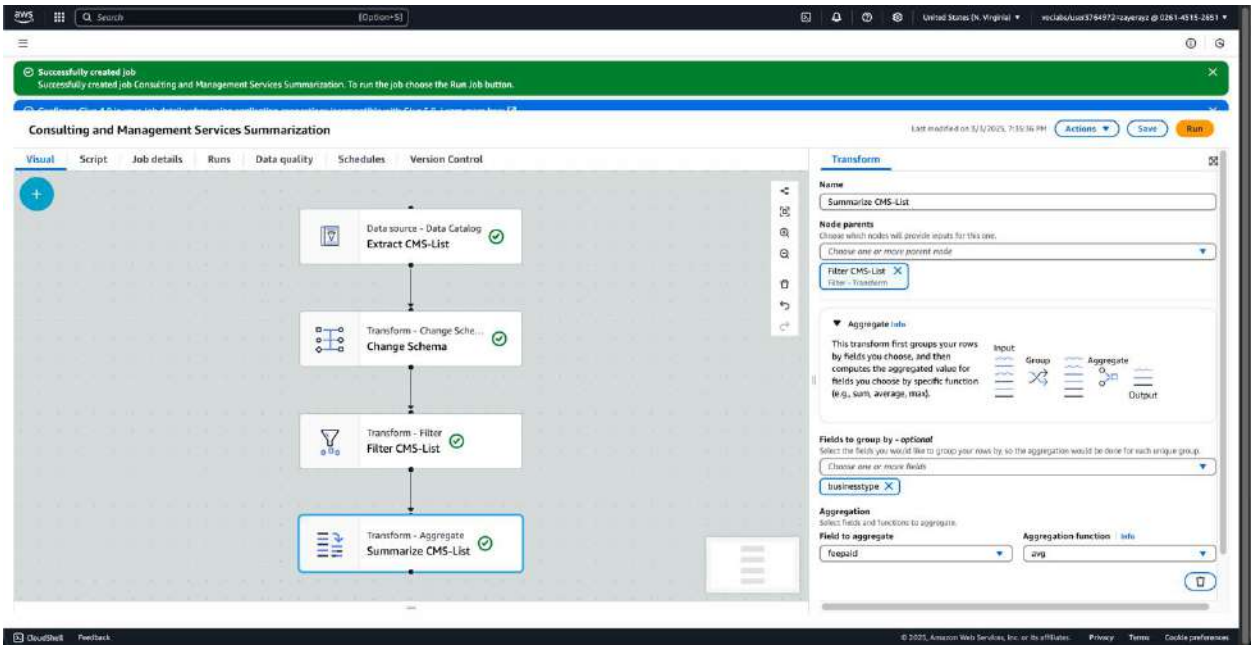


Note. Screenshot highlighting the filtering and aggregation steps used to compute the primary metrics in the ETL process. Source: AWS, *self-work*

In this image, I highlight the transformation steps within the pipeline. I applied filtering to remove rows that do not meet the required criteria and used aggregation functions to calculate my primary metric (such as the number of employees > 50).

Figure 58

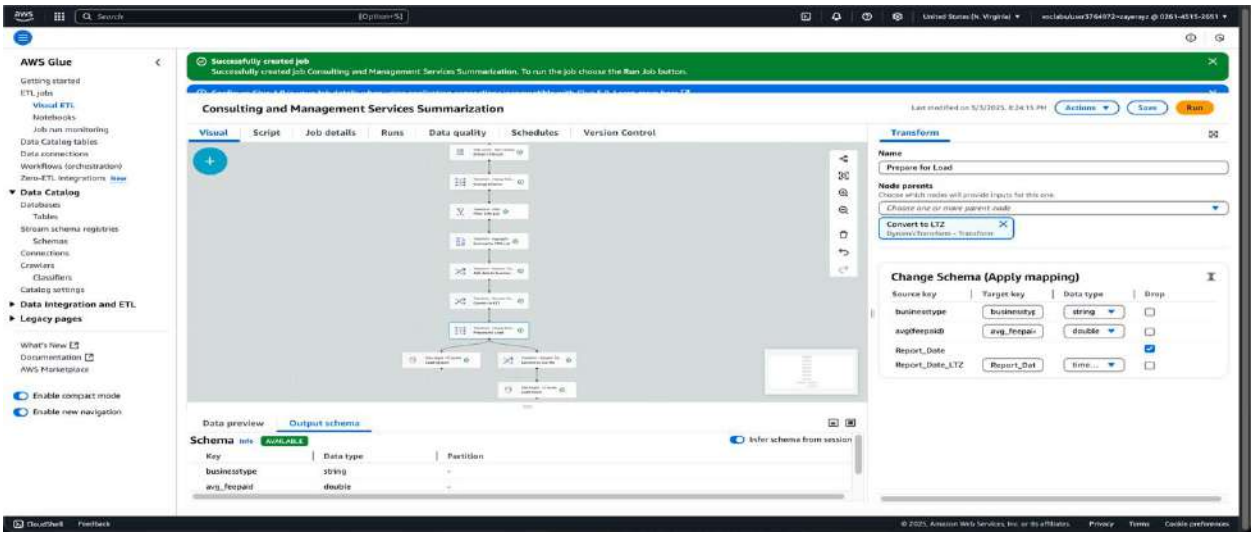
Aggregate



Note. Screenshot showing the aggregated data output after the transformation step in the ETL pipeline. Source: AWS, *self-work*

Figure 59

Final Summarized Data – System Folder and User Folder

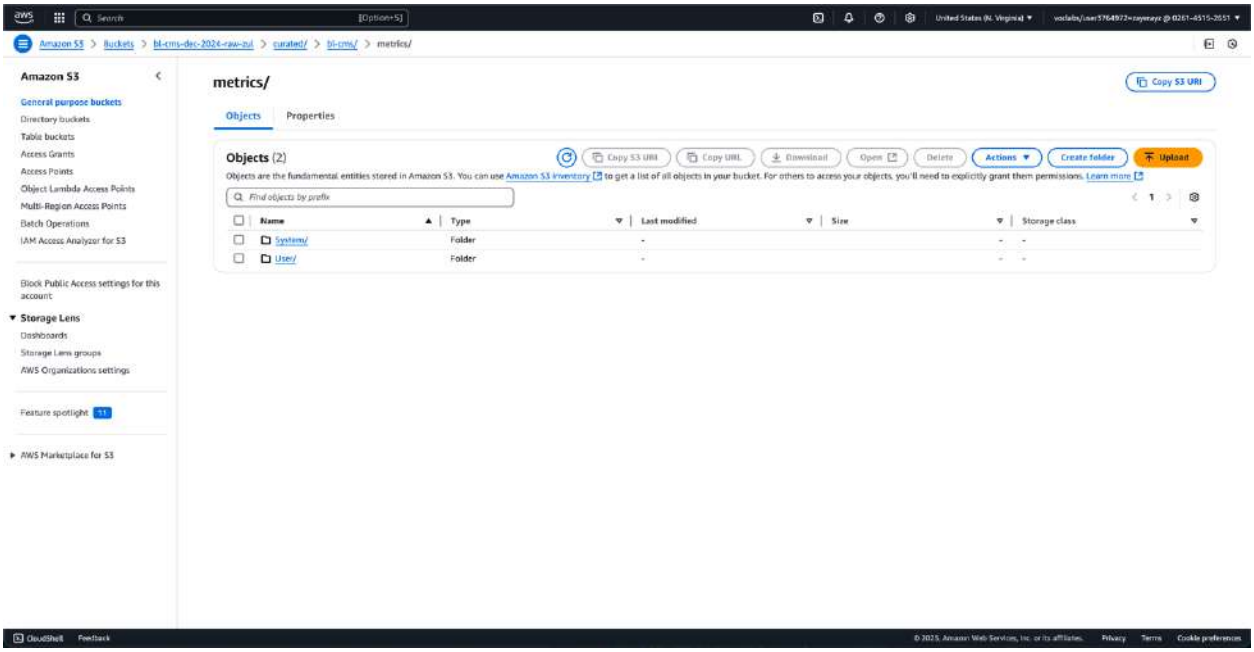


Note. Screenshot displaying the final summarized dataset stored in both the system and user folders, confirming successful ETL processing. Source: AWS, *self-work*

This screenshot displays the final output stored in the system folder and user folder. It confirms that the pipeline has successfully produced the transformed and aggregated dataset, which is now ready for analysis.

Figure 60

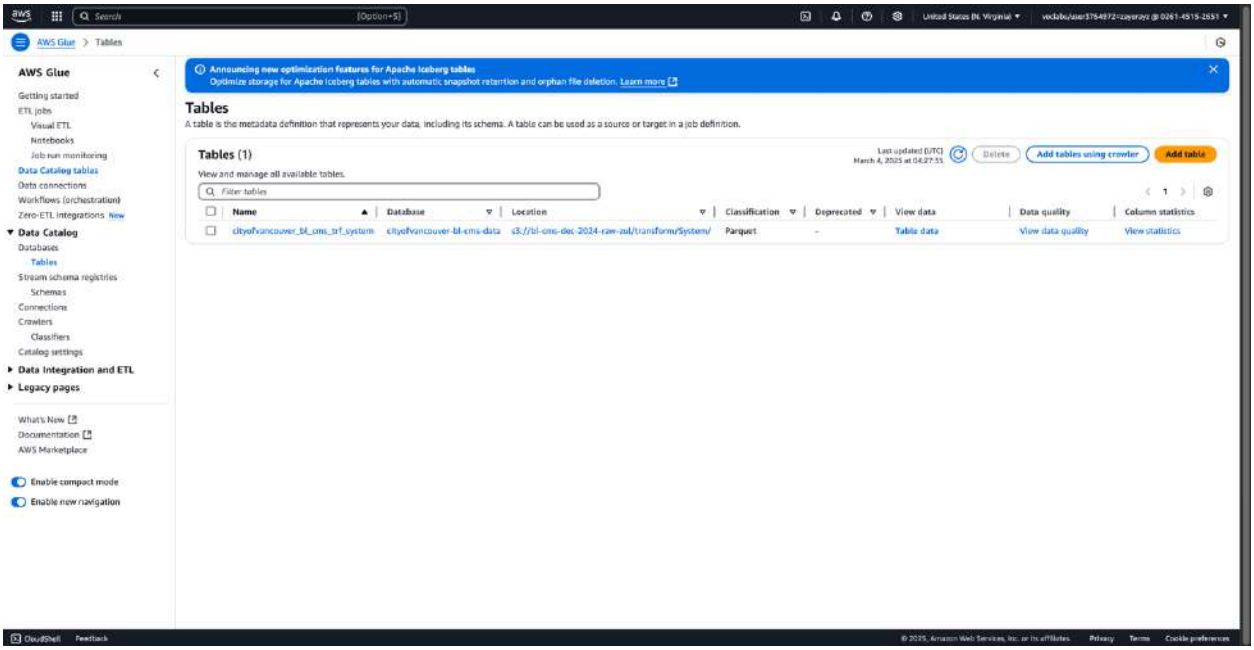
Curated System and Users



Note. Screenshot illustrating the organized folder structure in the curated S3 bucket where the final outputs are stored. Source: AWS, *self-work*

Figure 61

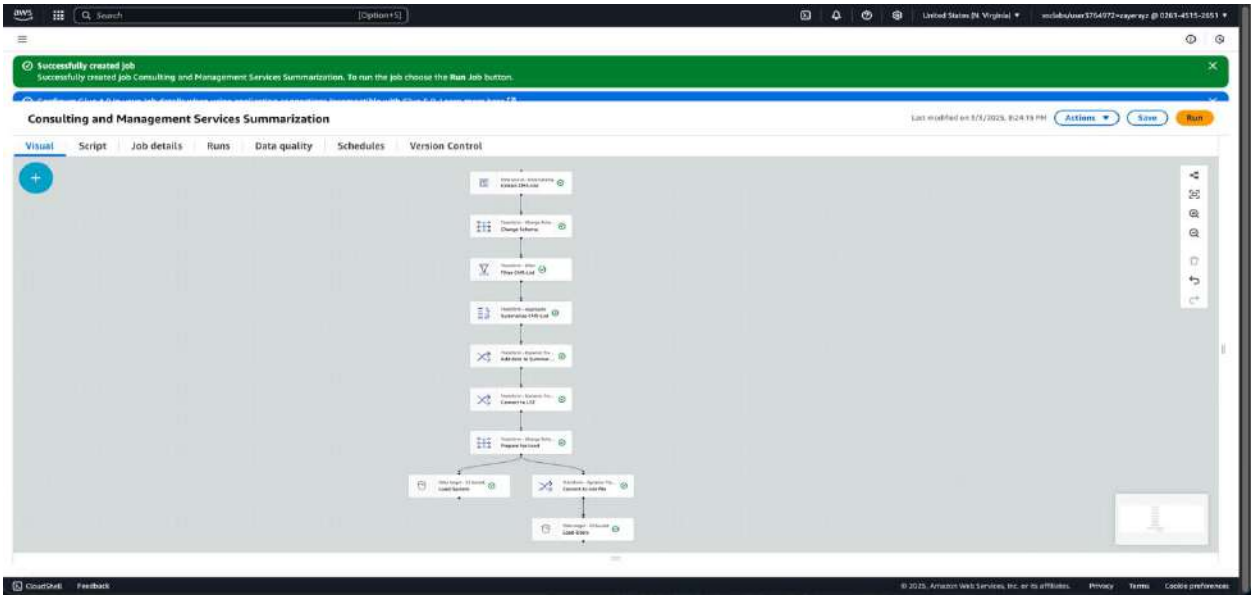
Tables



Note. Screenshot showing the final data tables generated from the ETL process, ready for analysis. Source: AWS, *self-work*

Figure 62

Overview of the ETL Pipeline in AWS Glue

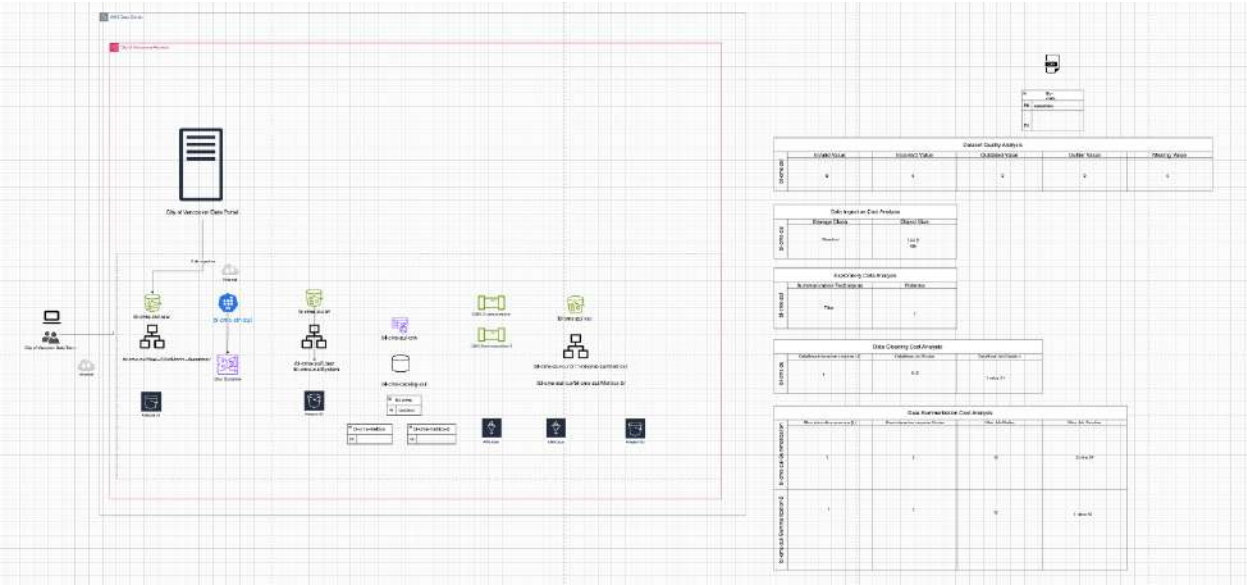


Note. Screenshot depicting the overall ETL pipeline flow from raw data ingestion to final output. Source: AWS, *self-work*

This screenshot shows my single, continuous ETL pipeline in Glue. The pipeline starts at the “Extract CMS-List” node and flows into an output node configured. This visual confirms that the pipeline is set up as a continuous flow.

Figure 63

Data Pipeline Architecture Diagram



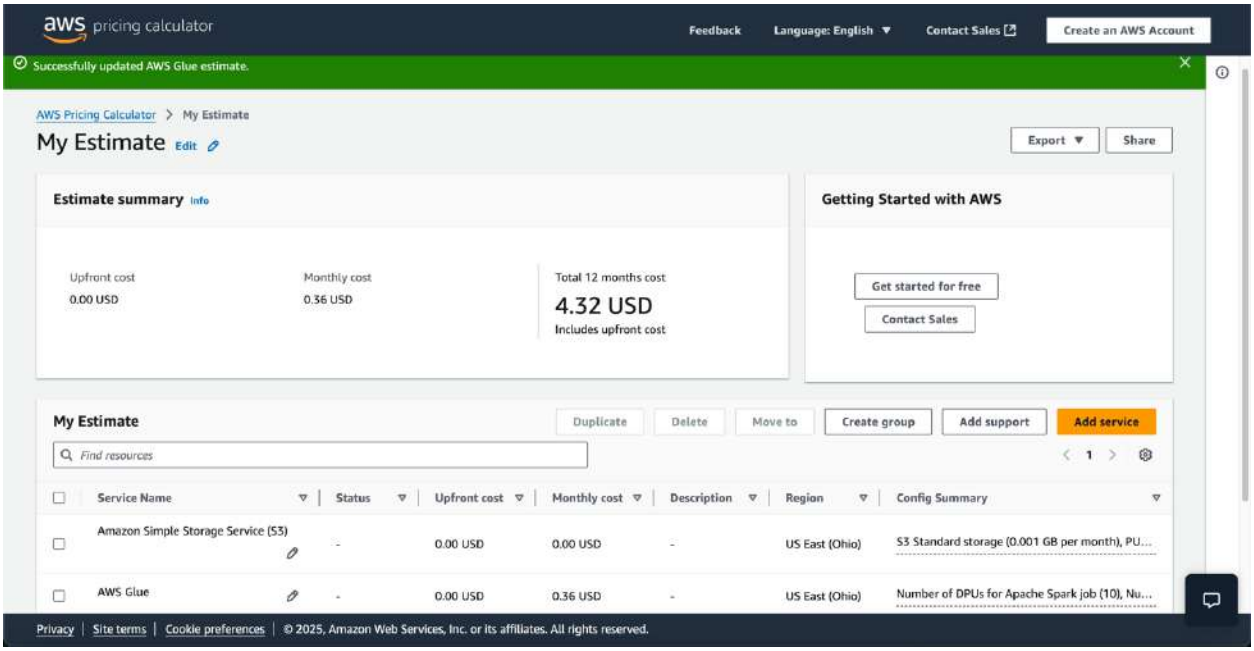
Note. Diagram created using draw.io that visually summarizes the entire data analytic process, from S3 ingestion and DataBrew cleaning to Glue cataloging and ETL transformation. (Source: *self-work*, created on draw.io)

This diagram provides a comprehensive view of the entire process I implemented. On the left, you can see how the raw CSV file is stored in Amazon S3 (bucket “bl-cms-dec-2024-zul”). Moving right, the dataset is profiled and cleaned with AWS Glue DataBrew, where I corrected the delimiter to semicolon, renamed columns, and removed unnecessary fields. Next, the AWS

Glue Crawler automatically creates a database and table schema. Finally, a single ETL pipeline in AWS Glue Studio applies transformations (filtering, aggregation, schema changes) and writes the results to the "transform," "system," and "user" folders in S3. This end-to-end flow ensures that the City of Vancouver's Business Licenses data is properly ingested, cleansed, cataloged, and summarized for further analysis.

Figure 65

AWS Pricing Calculator - Final Cost Estimate



Note. Screenshot of the AWS Pricing Calculator showing an estimated annual cost of $4.32, which includes S3 storage and AWS Glue usage. Source: AWS Cost calculator, *self-work*

In this screenshot, I ran my project on AWS Pricing Calculator for the Business Licenses, Consulting and Management Services dataset on AWS, and I ended up with a total of about $4.32 for 12 months. That cost includes my S3 storage (which is tiny, since the dataset is only a few hundred KB) and my AWS Glue jobs (for data cleaning, cataloging, and the ETL pipeline).

Because my data is so small and my ETL job runs for just 2 minutes, the monthly bill is super low. I didn't add any cost for Athena, because I either haven't run queries there or only ran a few small ones, which would be negligible anyway. So, if I only used S3 and Glue, $4.32 a year totally makes sense.

Basically, I'm paying a few cents a month to store the raw and cleaned data in S3, and the rest of the cost comes from Glue usage during my short ETL runs. If I ever scale up the data or run more frequent jobs, this cost will increase but for now, it's perfect for a small project. This also shows that AWS can be pretty cheap when your dataset and usage are minimal.

DAP Design and Implementation (Individual work – Haoran Xu)

Descriptive Analysis

I selected the dataset called "Public Trees", also to make sure my data volume is neither too large nor small. I filter the dataset on the data portal by "Species Name", and chose "Acutissima" as my target, it contains around 500 rows.

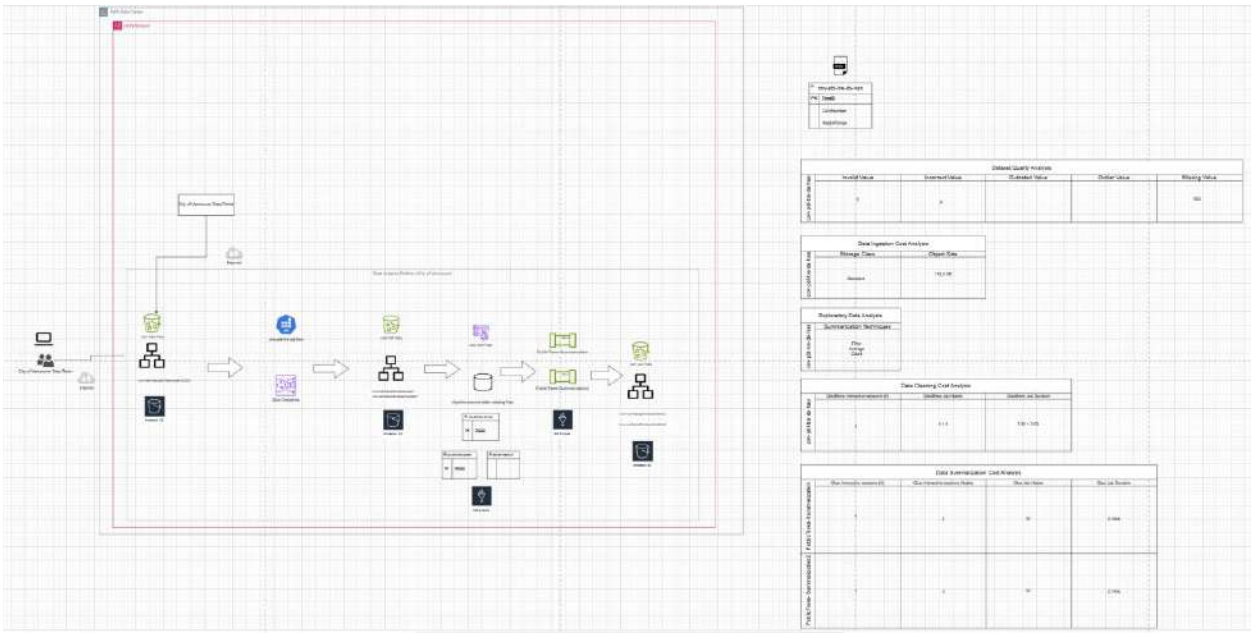
The dataset contains columns like TreeID, CivicNumber, STD Street, etc. My descriptive analysis questions are in the filter with Diameter greater than or equal to 10, is there a direct relationship between diameter and height range? And under the filter with Diameter greater than or equal to 10, and in a certain number of samples, what is the proportion of each different height range?

To answer this descriptive analysis question, I filtered and removed some irrelevant columns (which would be reflected in my ETL pipeline). The remaining columns are Diameters (will be AvgDiameter later), HeightrangeID and Heightrange. The screenshot below shows the entire process of my plan on drowio before implementation.

The first step is data ingestion, which means I download the corresponding dataset from the data portal, create a raw bucket in the S3 function and upload the dataset. The second step is data profiling. In this step, we use AWS Glue Databrew. I created a profile for my dataset. This allows me to easily browse if there are any data quality issues in my table, and I can also correct the corresponding problems. And we also need to create a bucket in S3 for data transformation. To store the clean data after cleaning, we put it in the system and user folders respectively. The third step is data cleaning. We also perform this step in Glue Databrew. We can browse the schema of our dataset to check if they have format problems or change the column name to make them clearer. Cleaning is a very important step because without clean data we cannot move on to the catalog. The fourth step is data cataloging. I first created a curated data bucket using the S3 function. Then I will create our database in AWS Glue and use crawler to get our data and store it in it ready for summarization. The last step is data summarization, which is also performed in AWS Glue. This is a crucial step to get the "descriptive analysis" we use to answer the descriptive analysis question. We will use ETL to create a series of pipelines, including uploading data, changing the schema format, filtering our data if necessary, adding our premises and assumptions, and attaching the time in the local timezone, and finally uploading the results to the curated data bucket we created in advance.

Figure 64

Drowio

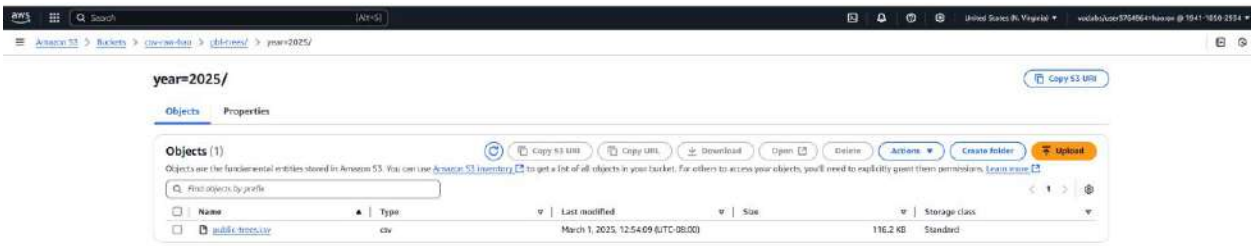


Note. This screenshot shows the drowio design part including the cost estimation at the right-side. Source: Made by author using Drow.io

Step 1: Data Ingestion

Figure 65

Raw Bucket



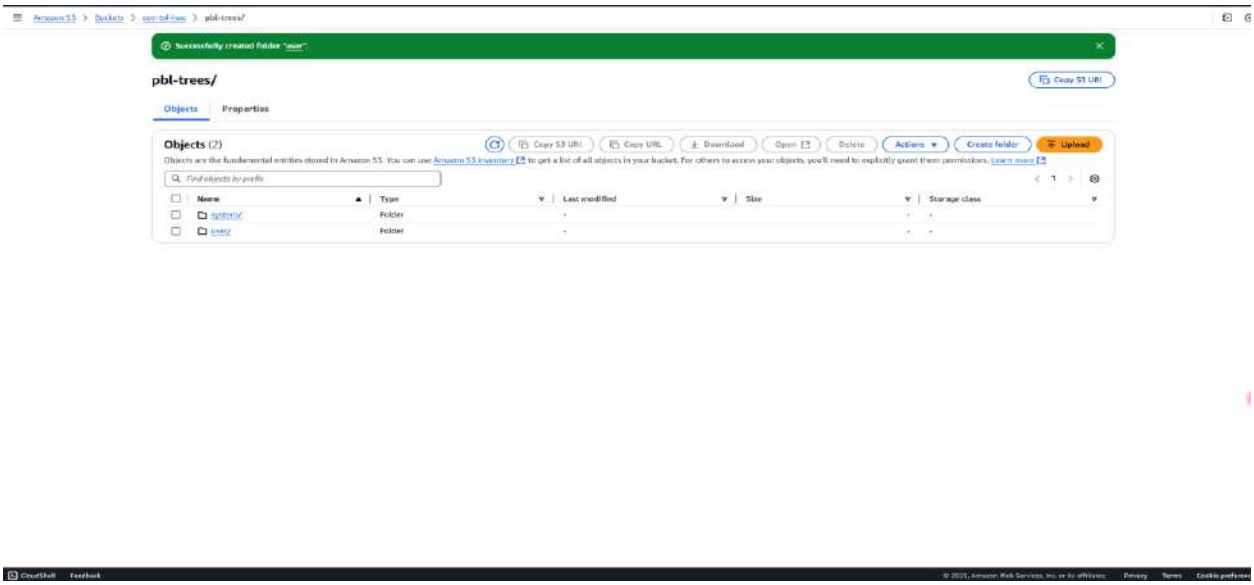
Note. This screenshot shows the raw bucket in AWS S3. Source: AWS S3

After downloading the datasets from City of Vancouver data portal, I created a bucket called cov-raw-hao as my raw data bucket. It is for storing my raw dataset. I create sub-folder inside it until year=2025 and then upload my downloaded dataset there as my raw dataset.

Step 2: Data Profiling

Figure 66

Transfer Bucket

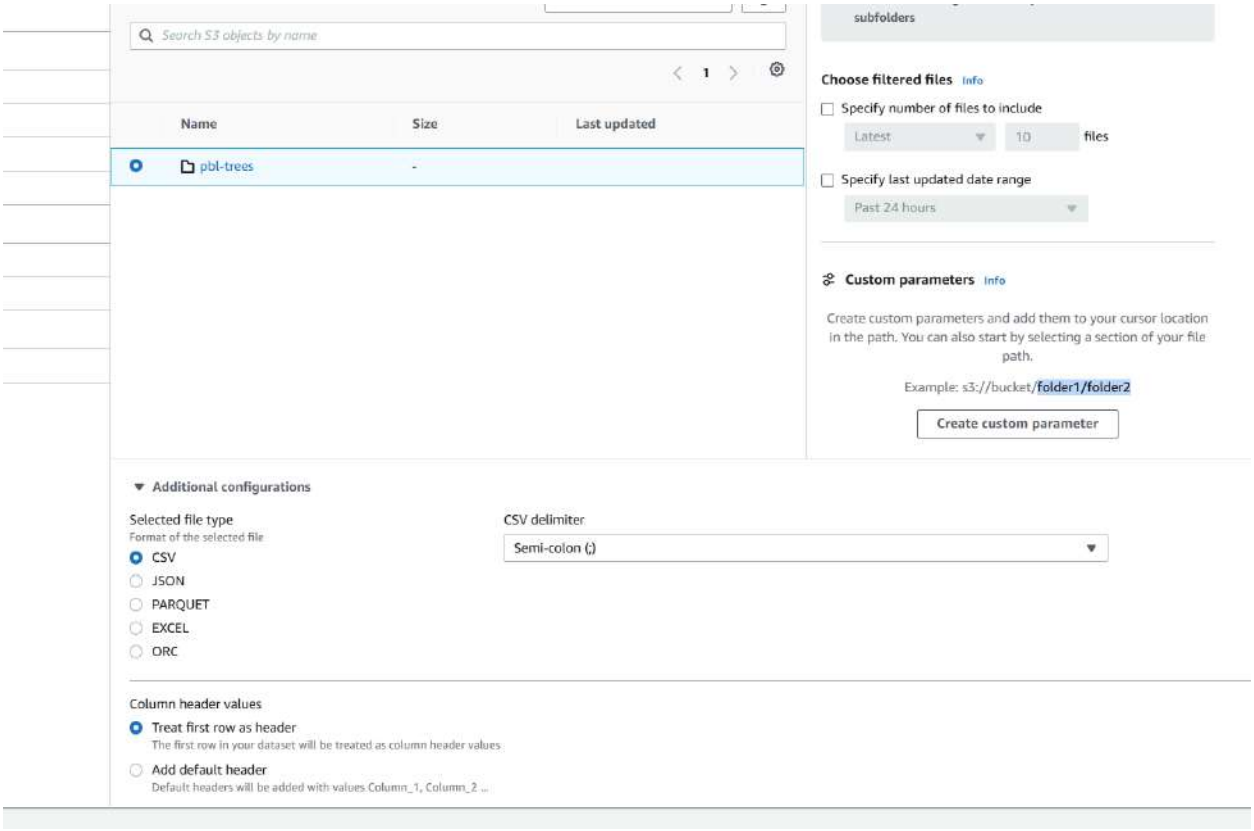


Note. This screenshot shows the transfer bucket in S3. Source: AWS S3

After Data ingestion, it's time for data profiling. Firstly, I choose to create another bucket called cov-tsf-hao which is for transfer bucket. And I create the sub-folder inside it until system and user folder. Currently, we have two different sub-folders for different uses.

Figure 67

Process of Profiling

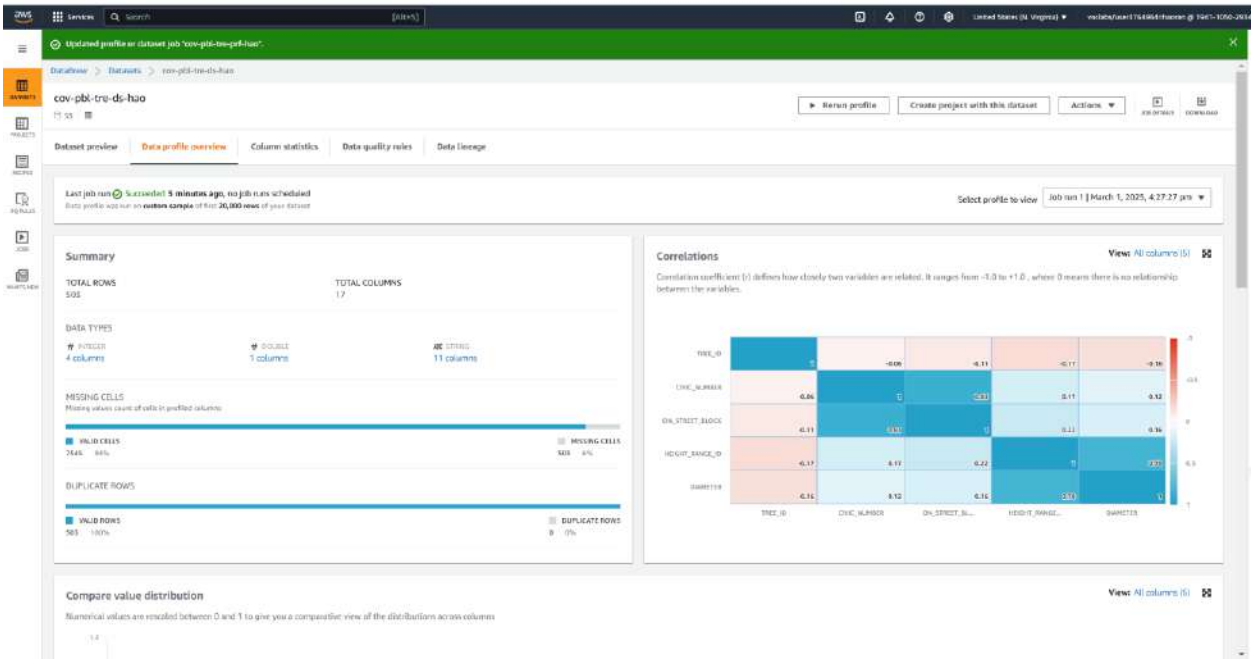


Note. This screenshot shows the process of creating data profiling. Source: AWS Glue Databrew

Here's when I'm creating the project, I failed last time because I don't change the CSV delimiter to Semi-colon (it's described on the data portal that the delimiter for csv file is semi-colon), I change the correct delimiter this time.

Figure 68

Data Profiling Completed



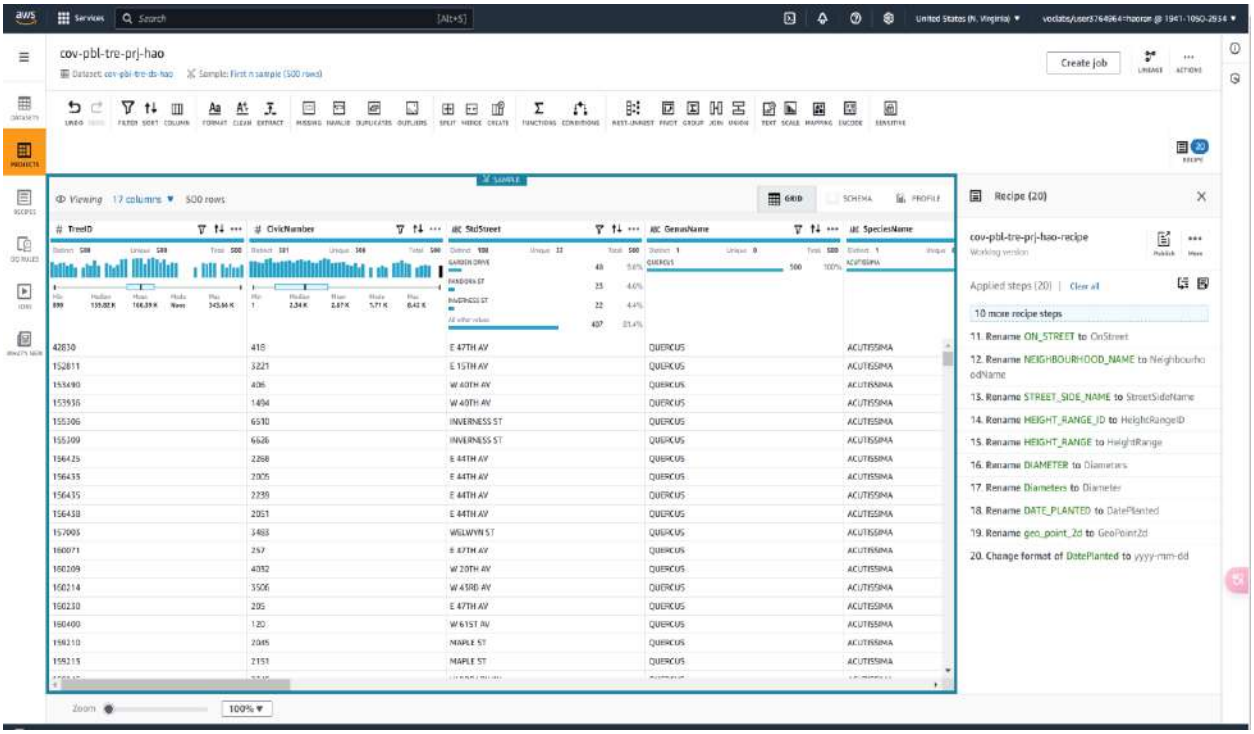
Note. This screenshot shows the profiling result in AWS Glue Databrew. Source: AWS Glue Databrew

Here's the profile screenshot after I create and run the job. And now it's time for data cleaning

Step 3: Data Cleaning

Figure 69

Data Cleaning Process



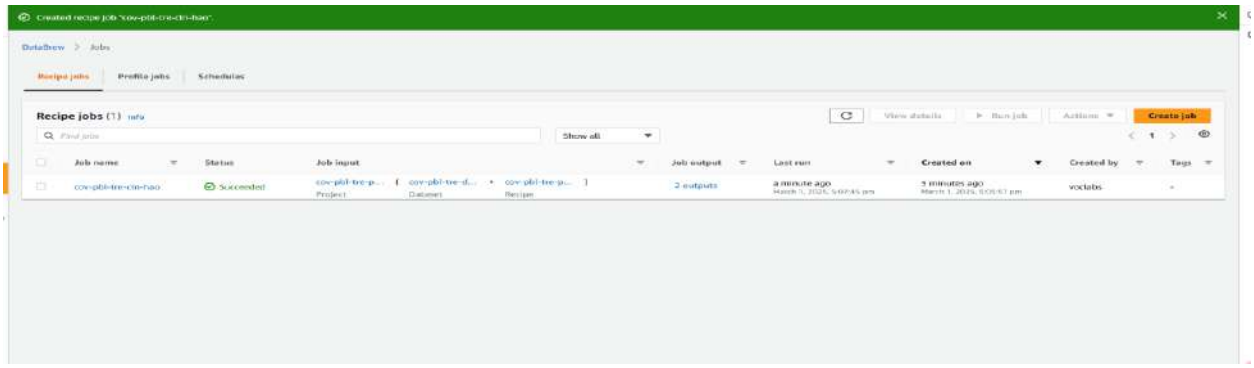
Note. This screenshot shows the Data cleaning process in AWS Glue Databrew. Source: AWS Glue Databrew

For the data cleaning, step 1 I changed all the column title to a clearer format to watch. I also changed the format of time-date format to yyyy-mm-dd.

All the formats are clean and structured here and I'm planning to create and run the cleaning job.

Figure 70

Cleaning Job Completed

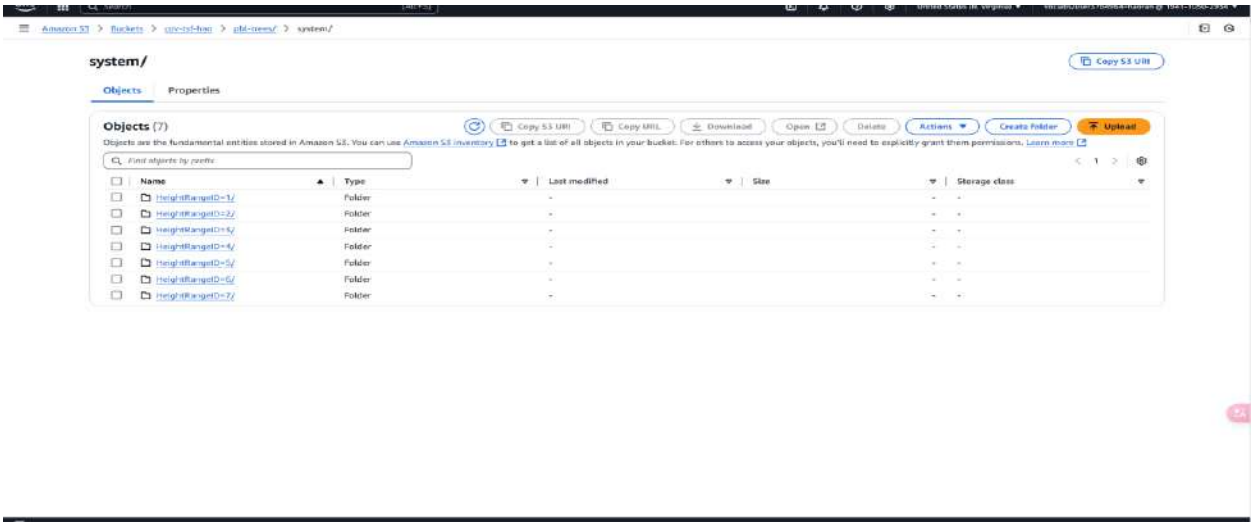


Note. This screenshot shows completion of cleaning job in AWS Glue Databrew. Source: AWS Glue Databrew

Here's the screenshot after I create and run the cleaning job.

Figure 71

System Output of Cleaned Dataset

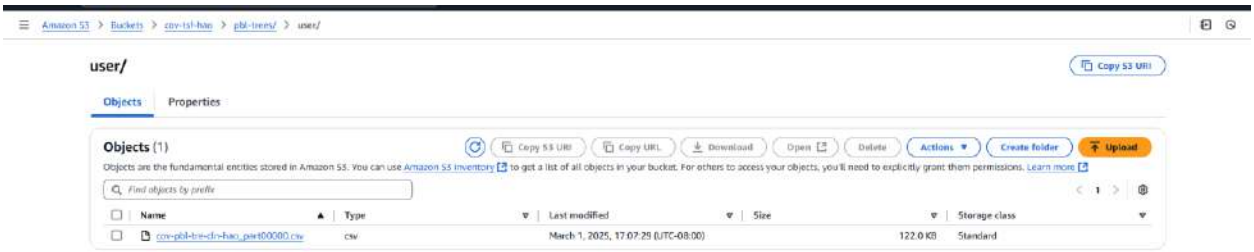


Note. This screenshot shows the cleaned dataset in system folder in AWS S3. Source: AWS S3

Here's the system folder of parq format and I used "Columns to partition by" and choose HeightRangeID as the output.

Figure 72

User output of Cleaned Dataset



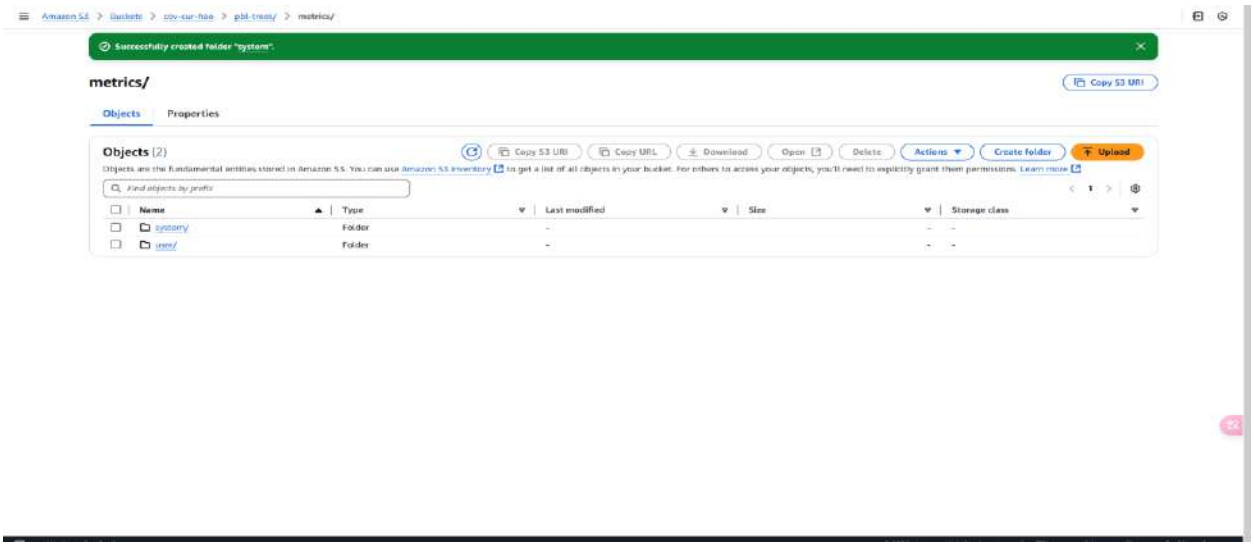
Note. This screenshot shows the cleaned data set-in user-friendly format in AWS S3. Source: AWS S3

Here's the user folder, cleaned csv file is stored here. Now we are going to Data Cataloging for next part.

Step 4: Data Cataloging

Figure 73

Curated Bucket Creation

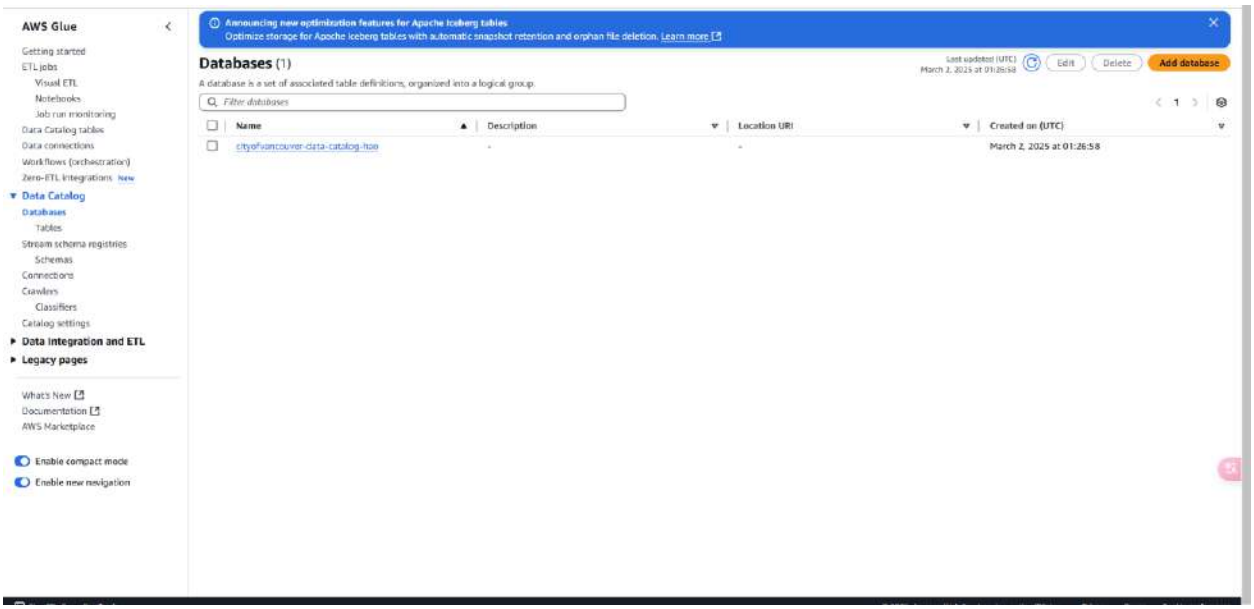


Note. This screenshot shows the curated bucket in AWS S3. Source: AWS S3

Firstly, I create a new bucket called cov-cur-hao and sub-folders for metrics until system and user folders.

Figure 74

Databased Creation

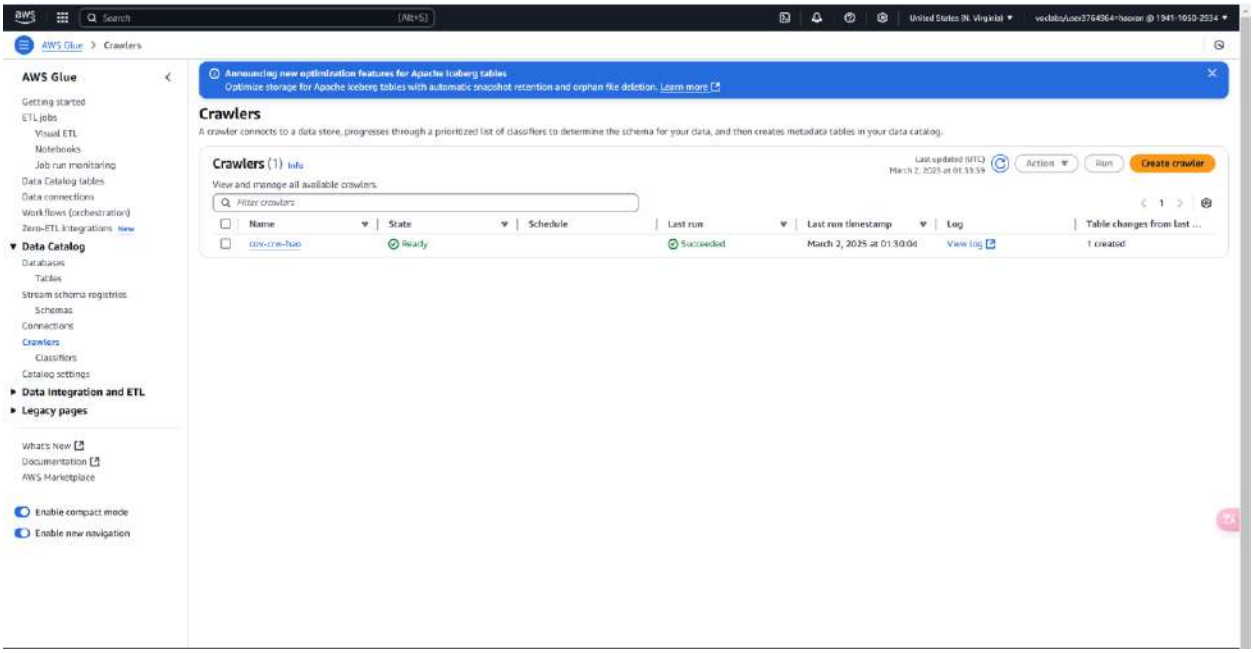


Note. This screenshot shows the database creation in AWS Glue. Source: AWS Glue

I created the database here.

Figure 75

Crawler Creation

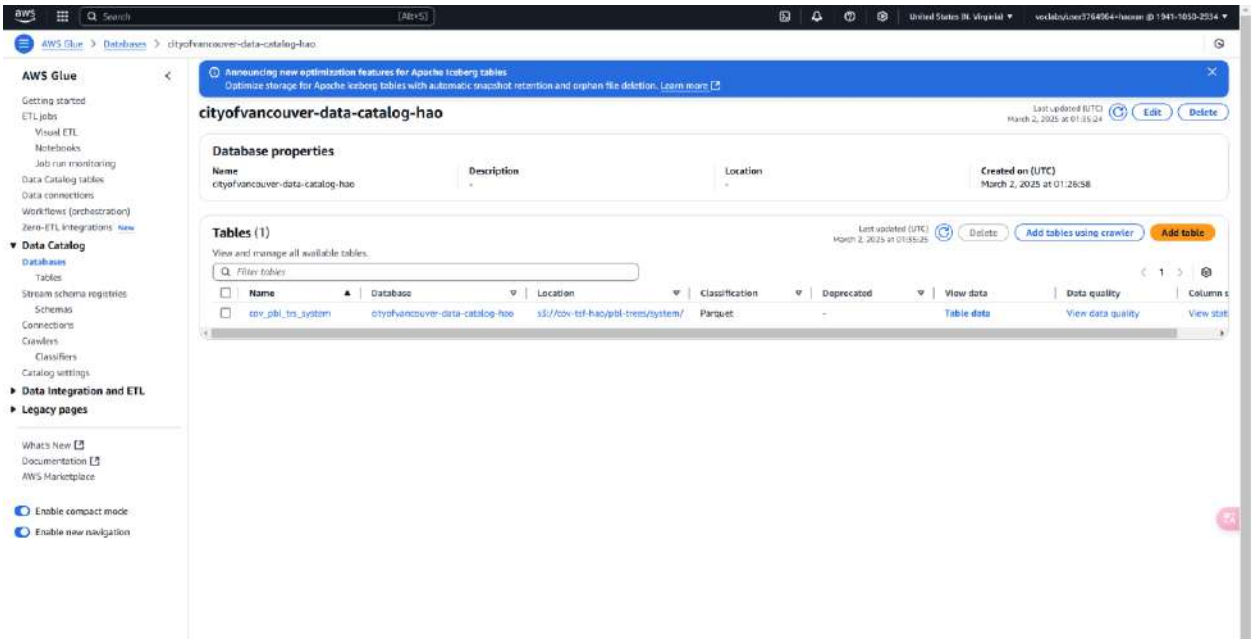


Note. This screenshot shows the successful run of Crawler in AWS Glue. Source: AWS Glue

Here I created a crawler to get the data for me automatically and store them into my database.

Figure 76

Result of Running Crawler



Note. This screenshot shows the output after running crawler in AWS Glue. Source: AWS Glue

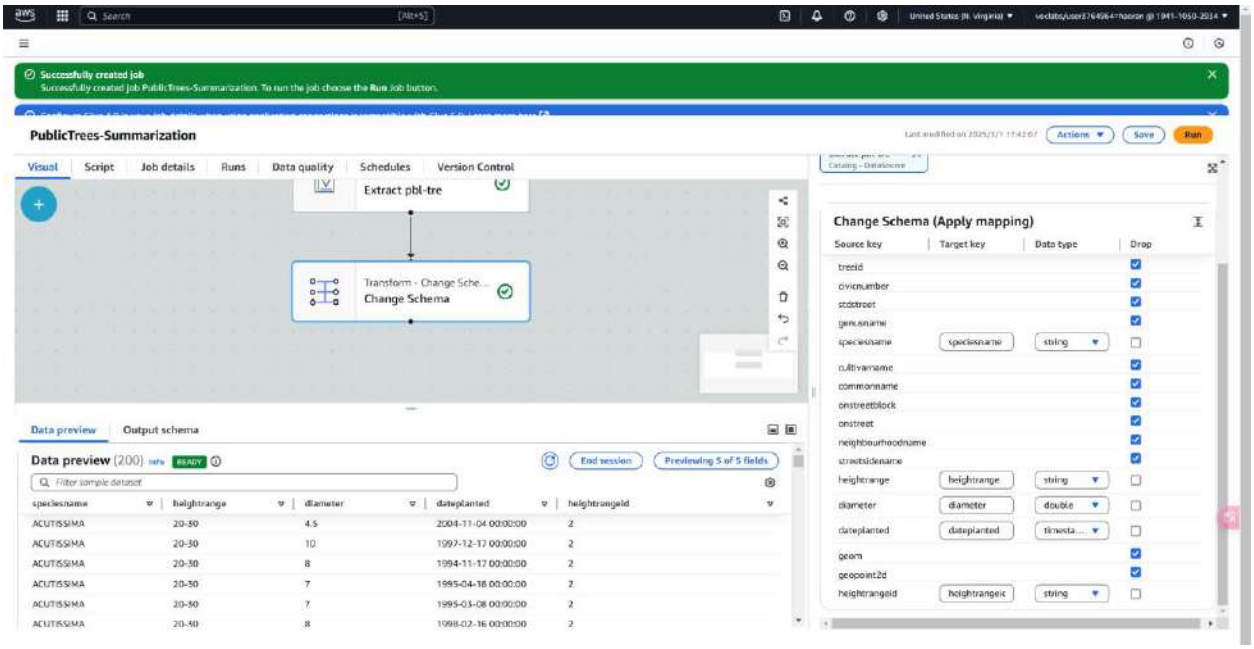
Here's the result of running my crawler. And this is the end of cataloging. Next, we are going to summarization part.

Step 5: Data Summarization

Summarization Metric 1 (ETL pipeline 1)

Figure 77

Change Schema of Summarization 1

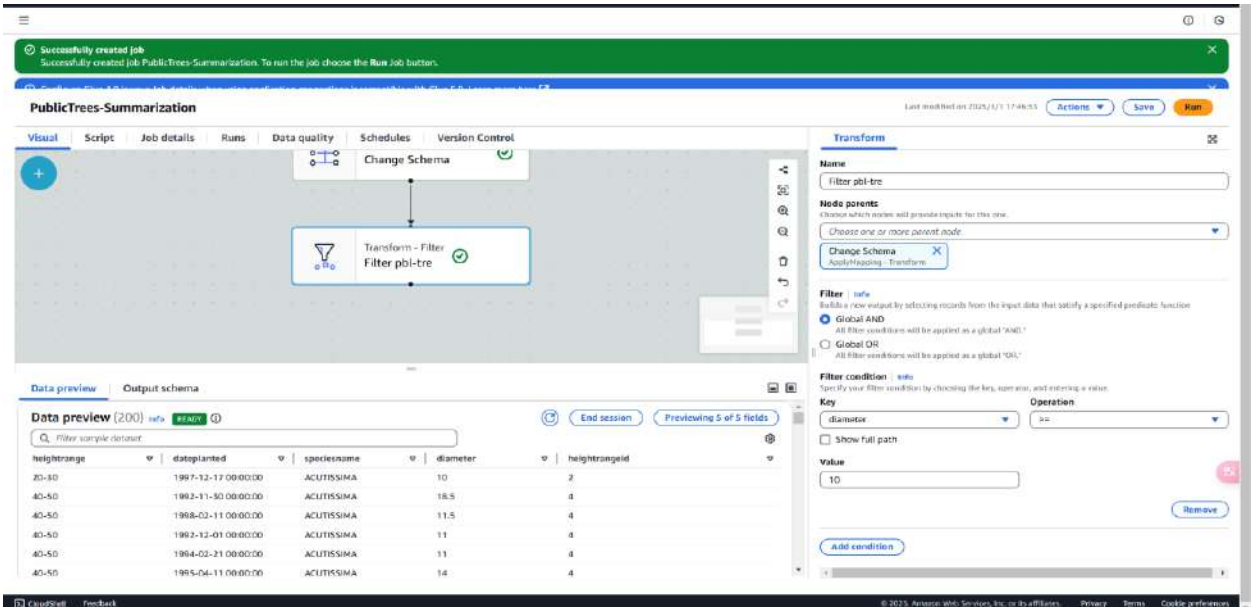


Note. This screenshot shows the change schema step in AWS Glue Visual ETL. Source: AWS Glue

I use “Change Schema” to drop some useless columns which I don’t want it to appear in my summarization. And I change the data type of dateplanted to timetable format. Finally, I only keep 5 columns here.

Figure 78

Filter of Summarization 1

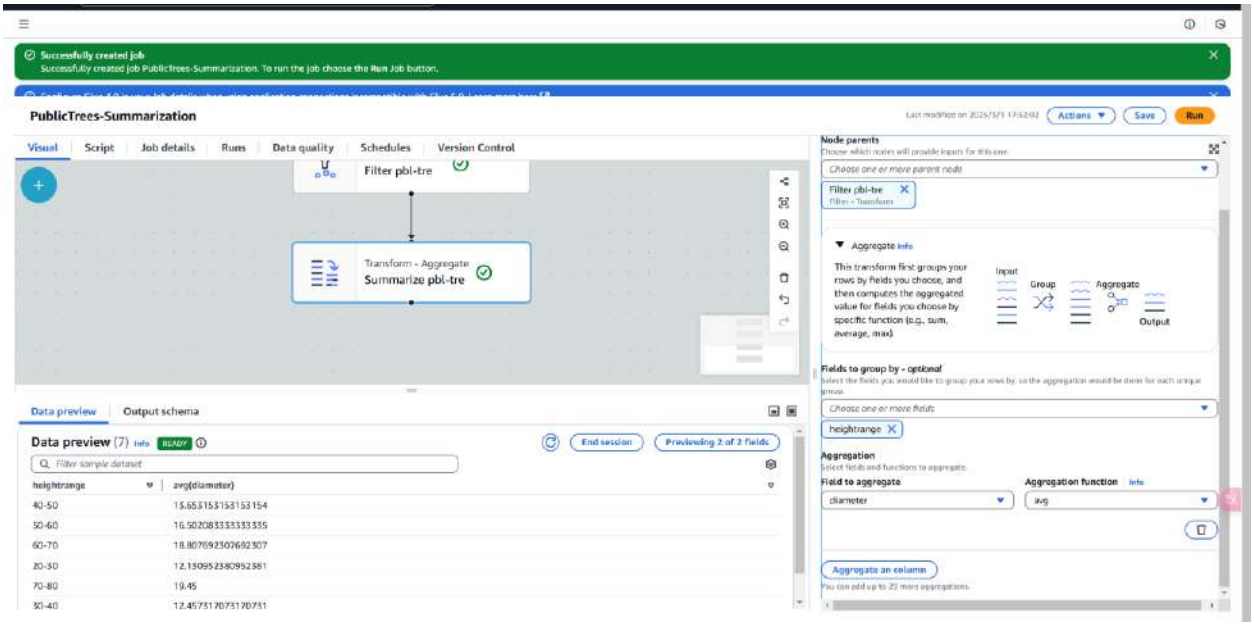


Note. This screenshot shows the filter step in AWS Glue Visual ETL. Source: AWS Glue

I use filter to filter my rows and keep those the diameter is equal or greater than 10. Now I only have 200 rows cut from 500 rows.

Figure 79

Aggregate of Summarization 1

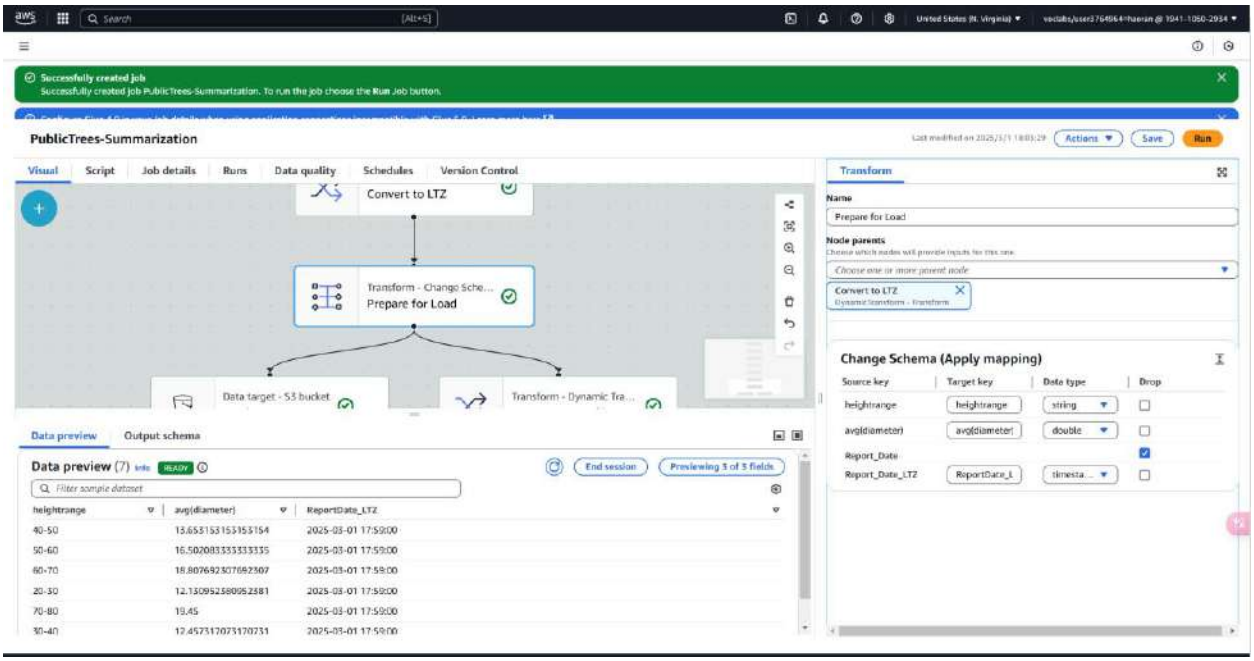


Note. This screenshot shows the aggregate step of summarization 1 in AWS Glue Visual ETL. Source: AWS Glue

Now I use aggregate to do summarization for my first metric. I want to know the average diameter of different heightrange of trees. (And I only count those diameters are equal or greater than 10 from last step).

Figure 80

Prepare for Load of Summarization 1

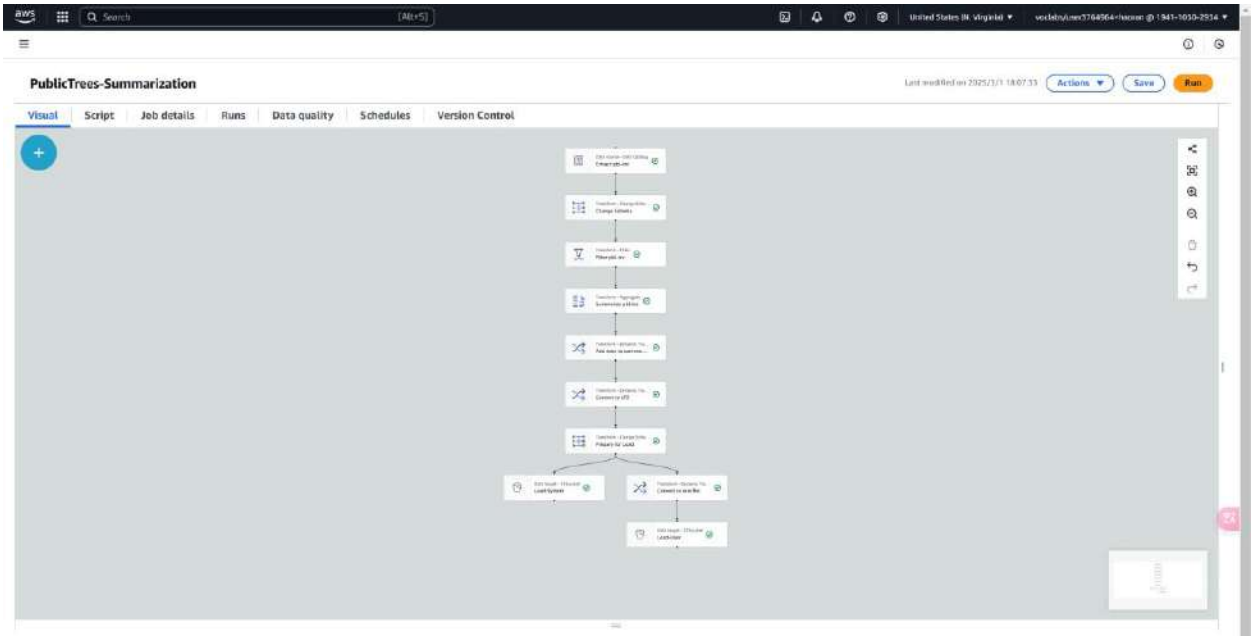


Note. This screenshot shows the prepare for load step in AWS Glue Visual ETL. Source: AWS Glue

I create the add current-timesstamp, convert the timezone to local time zone, and change the schema again here, to drop the old time data column and edit the column name for LTZ to clearer one.

Figure 81

Overall Look of Summarization 1



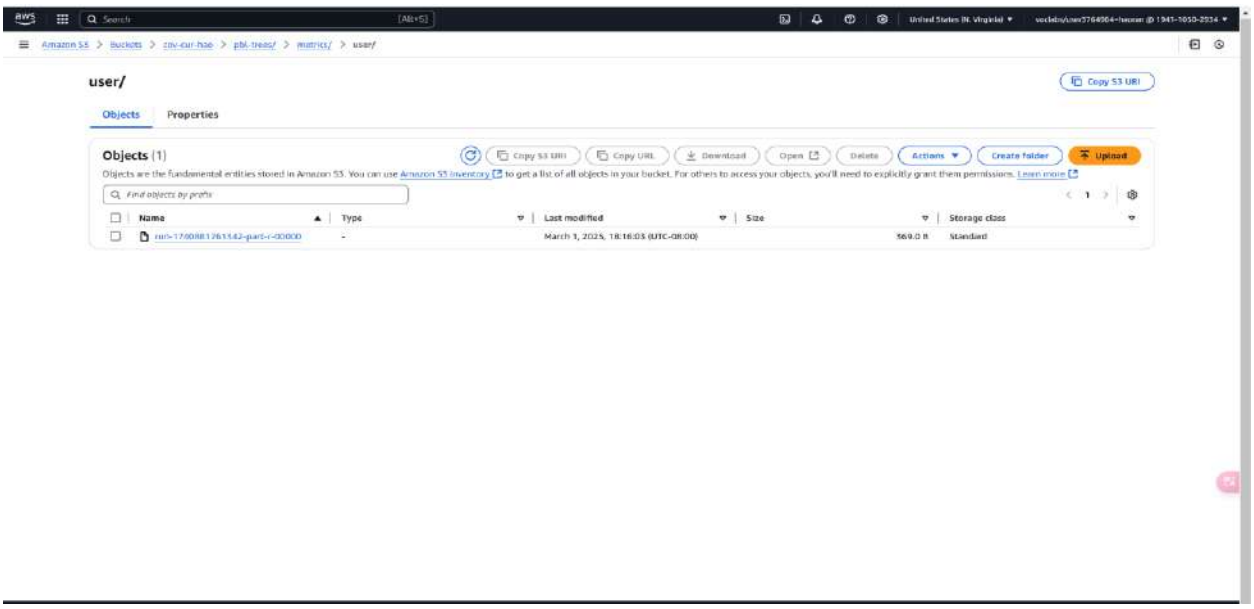
Note. This screenshot shows the overall look of summarization 1 in AWS Glue Visual ETL.

Source: AWS Glue

Here's the final look of the ETL pipeline of metric1.

Figure 82

User Output of Summarization 1

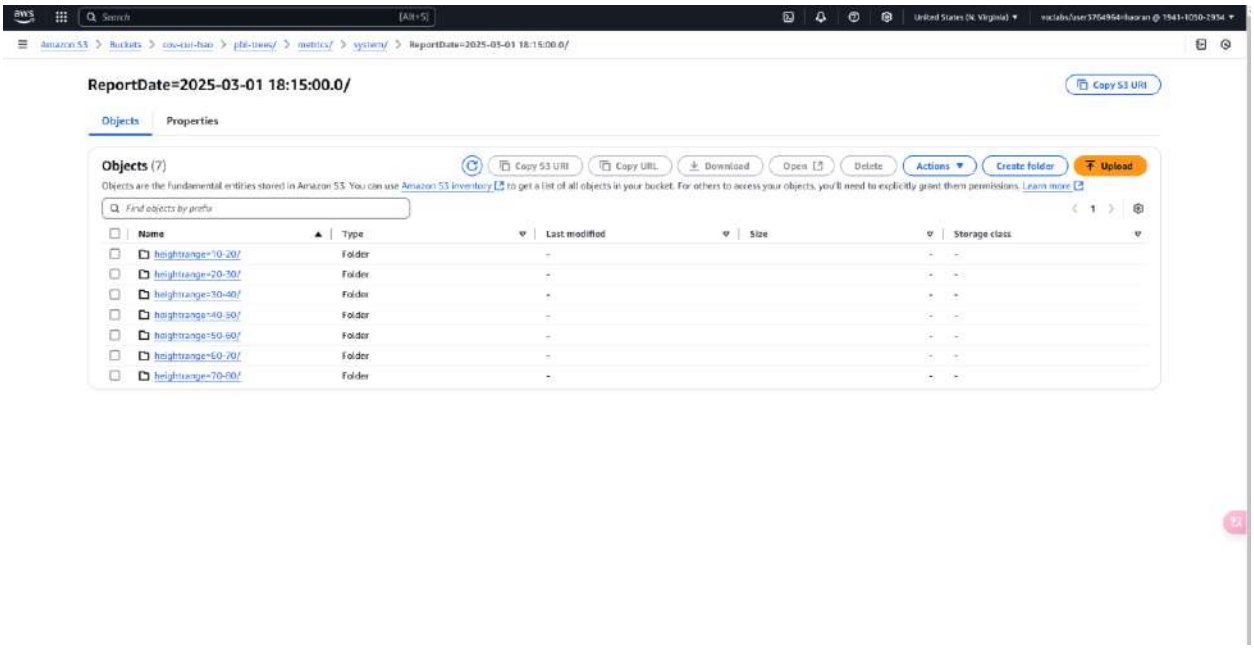


Note. This screenshot shows the user output of summarization 1 in AWS S3. Source: AWS S3.

Here’s the output after summarization. It is stored as user-friendly format in the user-folder .

Figure 83

System Output of Summarization 1

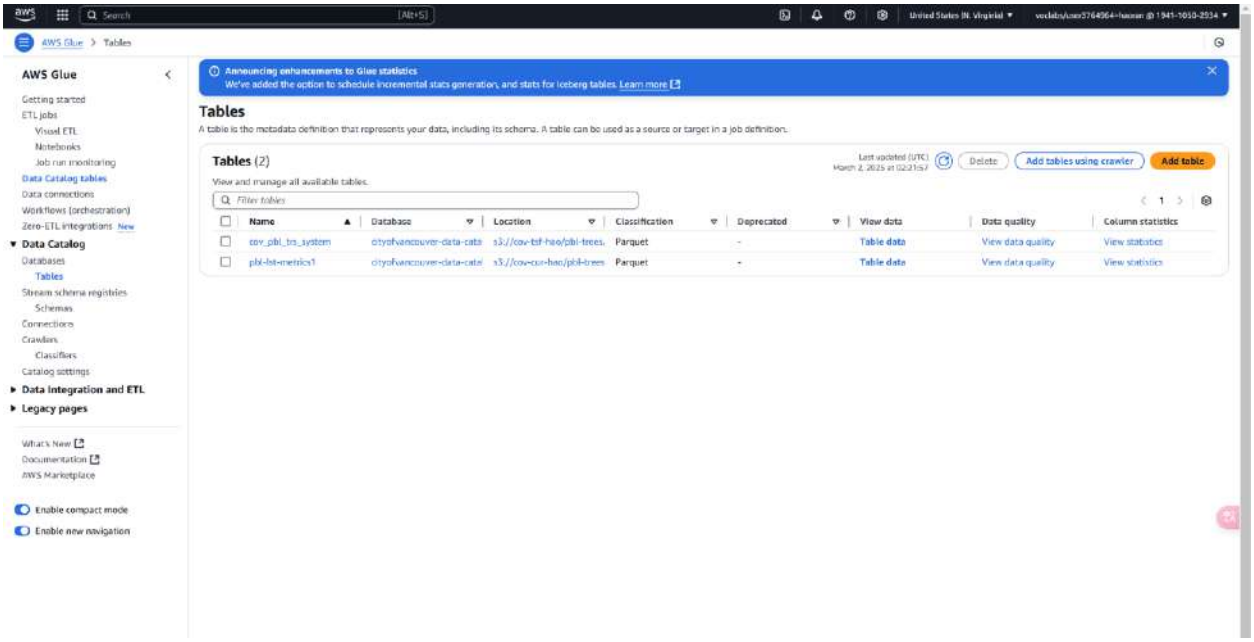


Note. This screenshot shows the system output of summarization 1 in AWS S3. Source: AWS S3

Here’s the output of summarization 1. It is stored as system-friendly format in the system folder.

Figure 84

Tables Screenshot

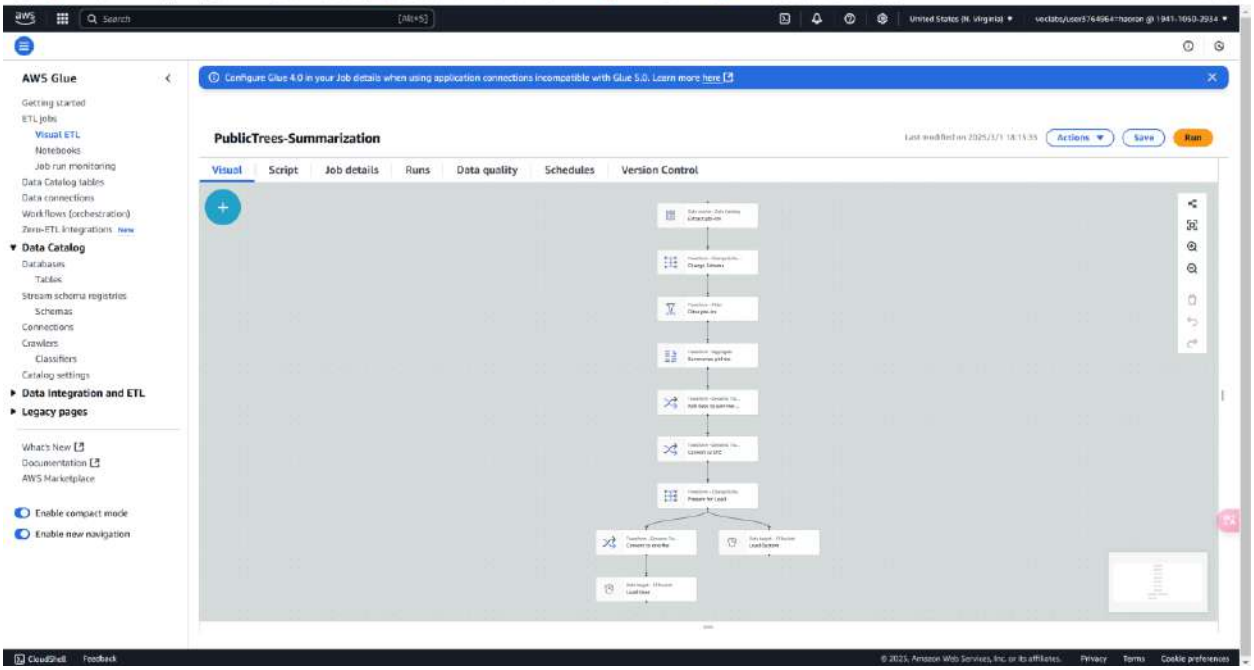


Note. This screenshot shows the tables output after summarization 1 in AWS Glue. Source: AWS Glue

Here's Tables screenshot in AWS Glue after running the summarization 1.

Figure 85

Overall look of Metric 1.



Note. This screenshot shows the overall look of summarization 1 in AWS Glue Visual ETL.

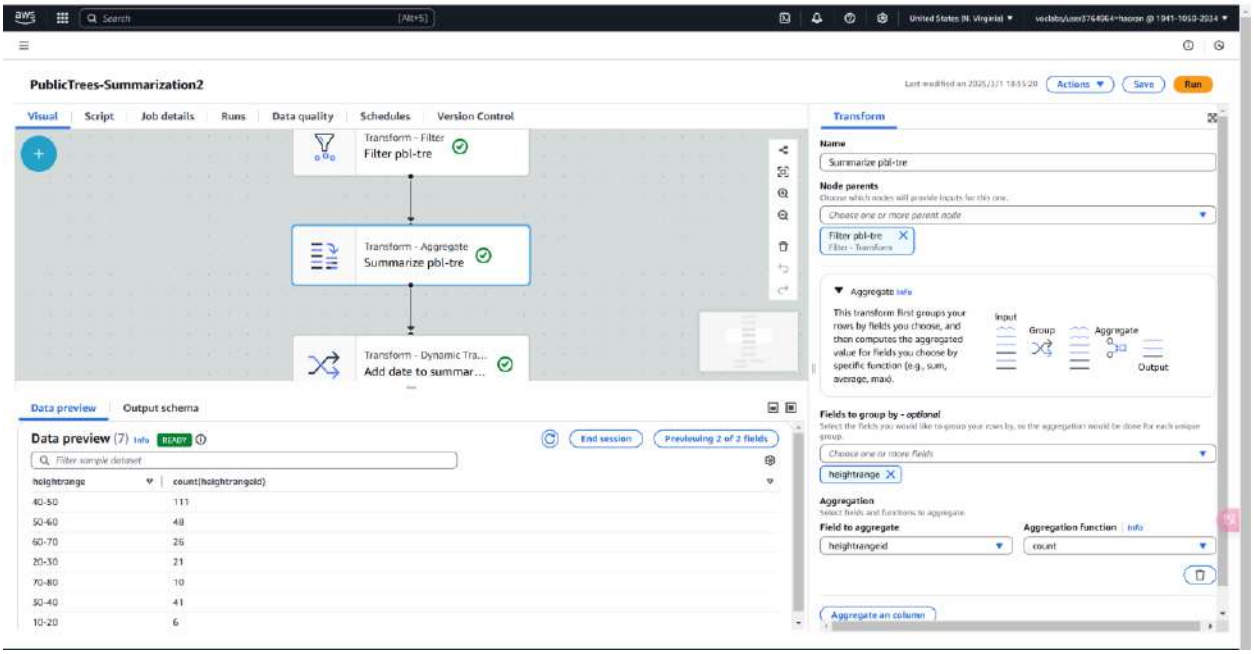
Source: AWS Glue

Here's the overall look of Metric 1 before running it.

Summarization Metric 2 (ETL pipeline 2)

Figure 86

Aggregate of Summarization 2



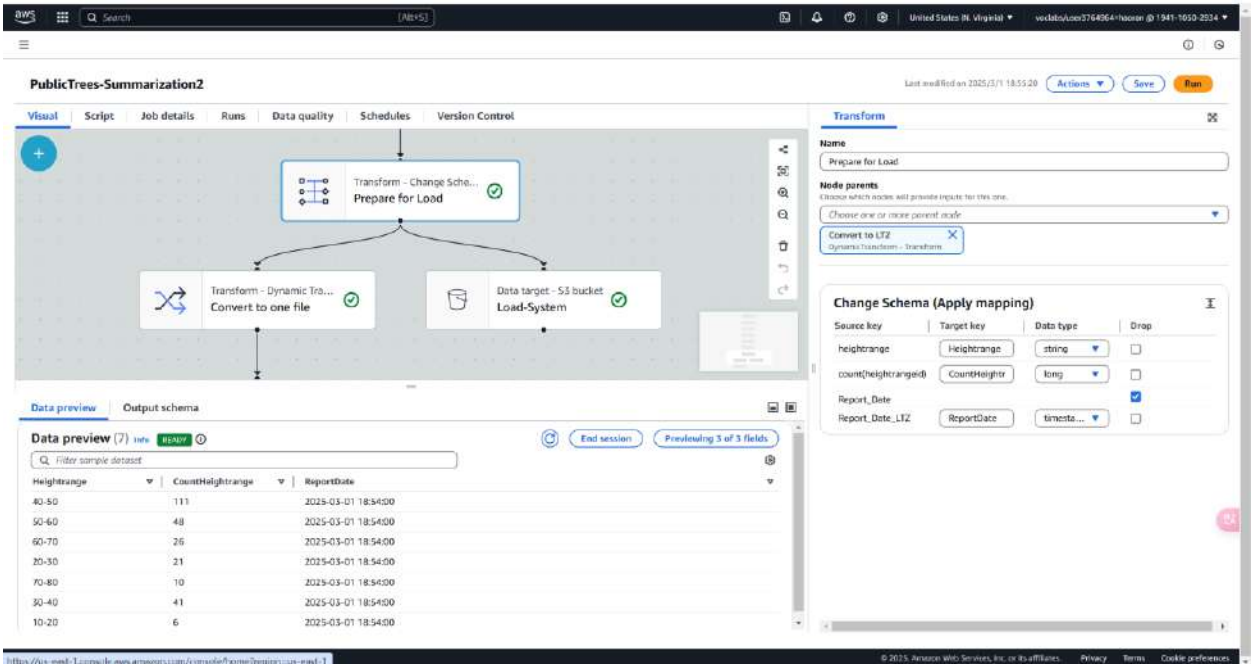
Note. This screenshot shows the aggregate step of summarization 2 in AWS Glue Visual ETL.

Source: AWS Glue

I created a new ETL pipeline2 by downloading and uploading pipeline1, because most of the parts will be the same. And I change the Summarization part to count how many are they for each heightrange.

Figure 87

Prepare for Load of Summarization 2

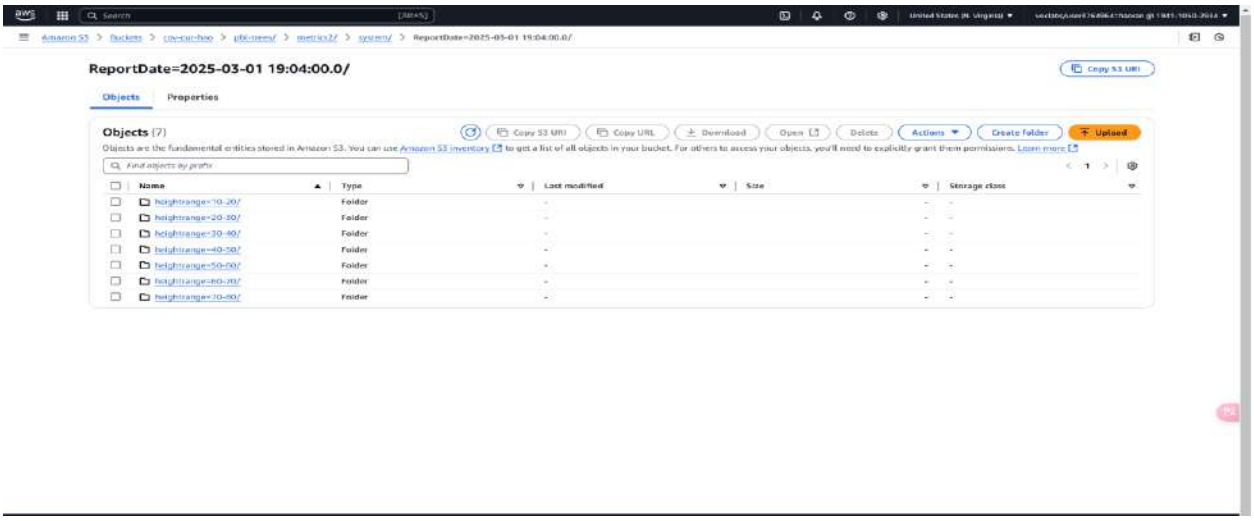


Note. This screenshot shows the prepare for load step in AWS Glue Visual ETL. Source: AWS Glue

I also change the column title of countheightrangeid to a clearer name “CountHeight” by using change schema before loading them to system and user folders.

Figure 88

System Output of Summarization 2

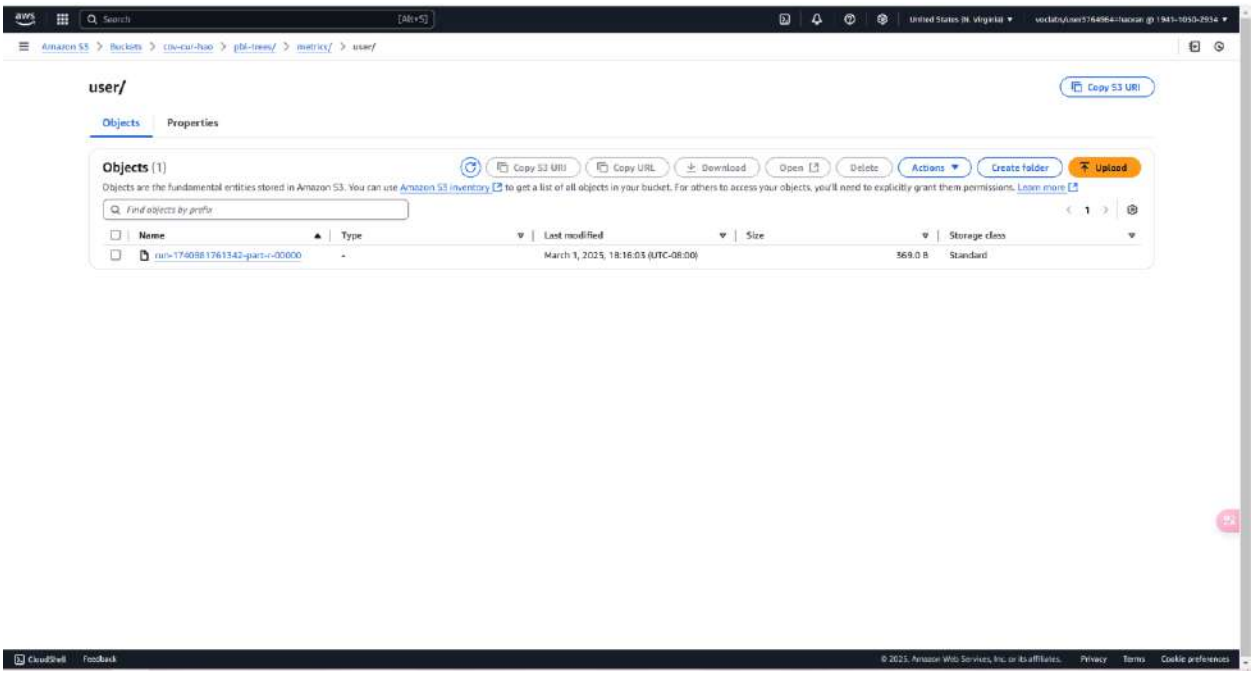


Note. This screenshot shows the system output of summarization 2 in AWS S3 Source: AWS S3

Here's the System-folder screenshot, storing the output dataset of summarization 2 in system-friendly format.

Figure 89

User Output of Summarization 2

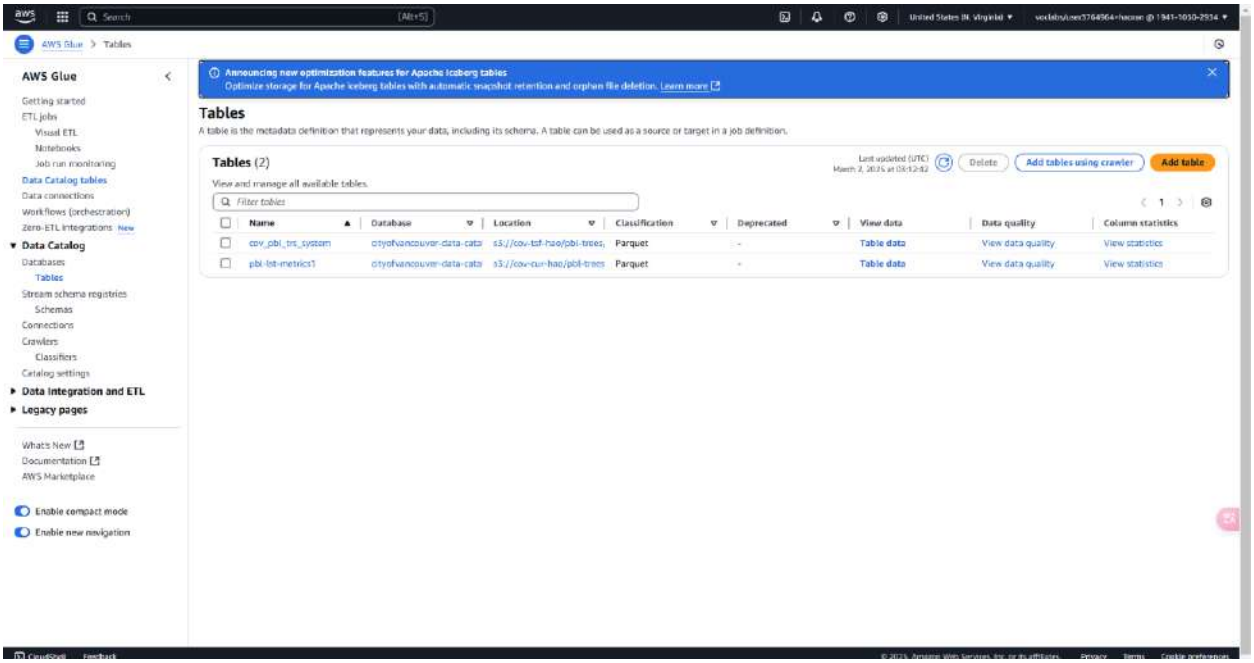


Note. This screenshot shows the user output of summarization 2 in AWS S3 Source: AWS S3

Here's the user-folder screenshot, storing the output dataset of summarization 2 in user-friendly format.

Figure 90

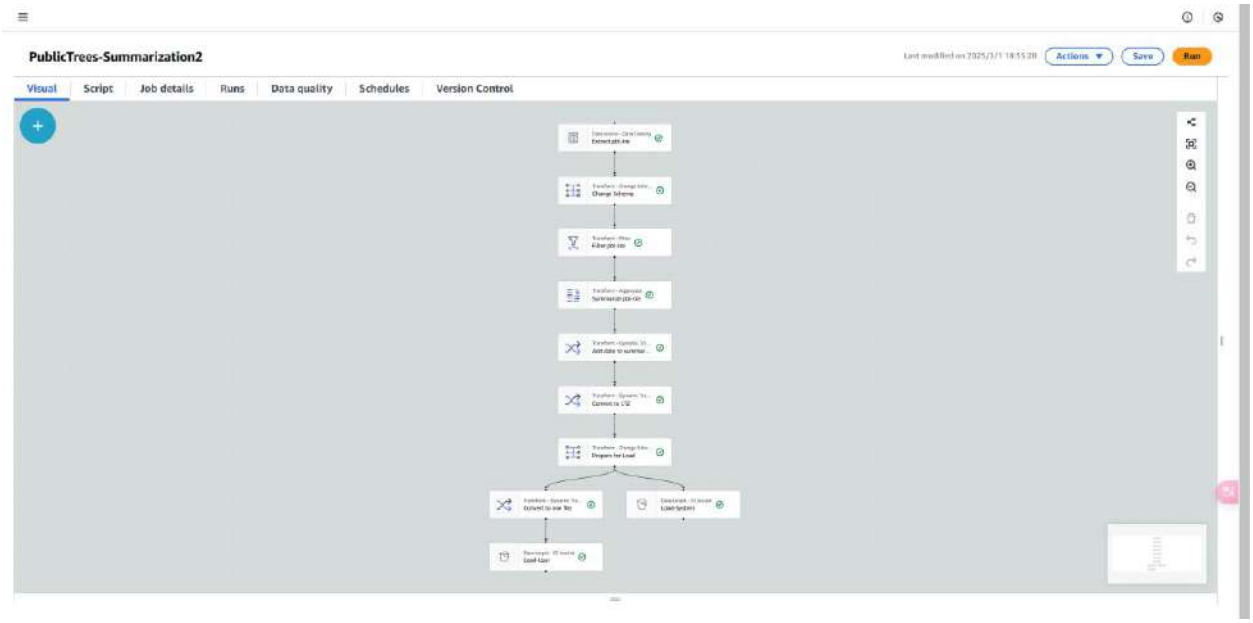
Tables Output of Summarization 2



Note. This screenshot shows the tables output of summarization 2 in AWS Glue. Source: AWS Glue

Figure 91

Overall Look of Metric 2



Note. This screenshot shows the final overall look of summarization 2 in AWS Glue. Source: AWS Glue

Here's the final overall look of summarization 2 for metric2.

DAP Estimated Cost (Teamwork)

Total Estimated Cost of The Team

The estimated total cost for our team would be USD 68.04 for 12 months, and the specific explanation for each member is below.

DAP Estimated Cost (Gyanvi Bhardwaj)

To estimate the monthly and yearly costs, I have used the AWS cost calculator. The first service added is AWS S3. Under this, I have added S3 Standard, S3 Intelligent Tiering, S3 Glacier Instant retrieval and data transfer. My object storage size is 0.0000629425 GB. The

lifecycle request per month is 1 as the only transition needed after 30 days would be from standard to intelligent tiering. Data returned and scanned by S3 Select is 1 GB per month.

The next service added is AWS Glue. Under Glue, the following services are added: AWS Glue ETL jobs and interactive sessions, AWS Glue Data Catalog storage requests, AWS Glue DataBrew interactive sessions and AWS Crawlers. There are 10 DPUs for the Apache Spark job, 0.0625 DPUs for the Python Shell job and the Apache Spark ETL job run duration is 2 minutes. The DataBrew jobs created is 1 and the number of nodes consumed for the DataBrew job is 5. The duration of the DataBrew job is approximately 2 minutes (1 minute 53 seconds). There is 1 crawler created with the duration for each crawler is 82 seconds.

Therefore, in total the two services with their sub-services cost 28.20 USD yearly and 2.35 USD monthly with 0 USD as the upfront cost.

Figure 92

Cost Estimation of Gyanvi Bhardwaj

aws

Contact your AWS representative: Contact Sales

Export Date: 03/04/2025 **Language:** English

Estimate url

Estimate summary

Upfront cost	Monthly cost	Total 12 months cost
0.00 USD	2.35 USD	28.20 USD Includes upfront cost

Detailed Estimate

Name	Group	Region	Upfront cost	Monthly cost
Amazon Simple Storage Service (S3)	-	US East (Ohio)	0.00 USD	0.00 USD
Status	-			
Description:	-			
Config summary	PUT, COPY, POST, LIST requests to S3 Standard (2), GET, SELECT, and all other requests from S3 Standard (1), S3 Standard storage (0.0000629425 GB per month) S3 INT Average Object Size (16 MB), Percentage of Storage in INT-Frequent Access Tier (1), Lifecycle Transition requests (1)			

Name	Group	Region	Upfront cost	Monthly cost
AWS Glue	-	US East (Ohio)	0.00 USD	2.35 USD
Status	-			
Description:	-			
Config summary	Number of DPUs for Apache Spark job (10), Number of DPUs for Python Shell job (0.0625) Number of interactive sessions for DataBrew (2), Number of nodes consumed for DataBrew job (5) Number of crawlers (1)			

Acknowledgement

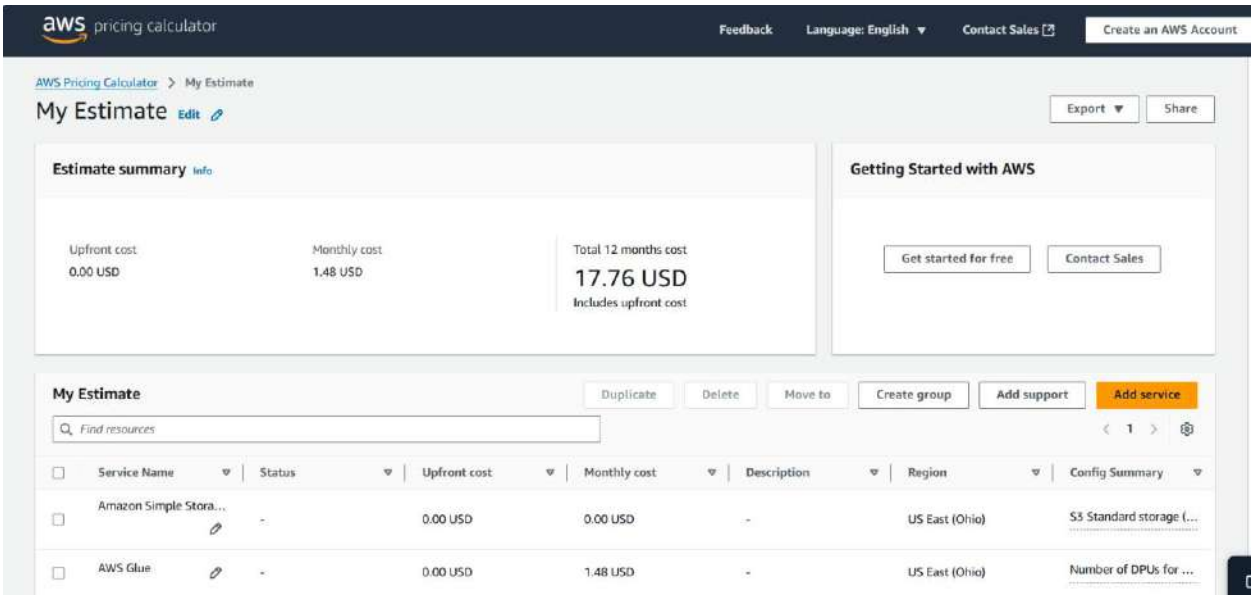
AWS Pricing Calculator provides only an estimate of your AWS fees and doesn't include any taxes that might apply. Your actual fees depend on a variety of factors, including your actual usage of AWS services. Learn more

Note. The total yearly cost of availing the mentioned AWS services is 28.20 USD which also includes the upfront cost. Source: AWS Cost calculator, *self-work*

DAP Estimated Cost (Peng Wang)

Figure 93

Cost Estimation of Peng Wang



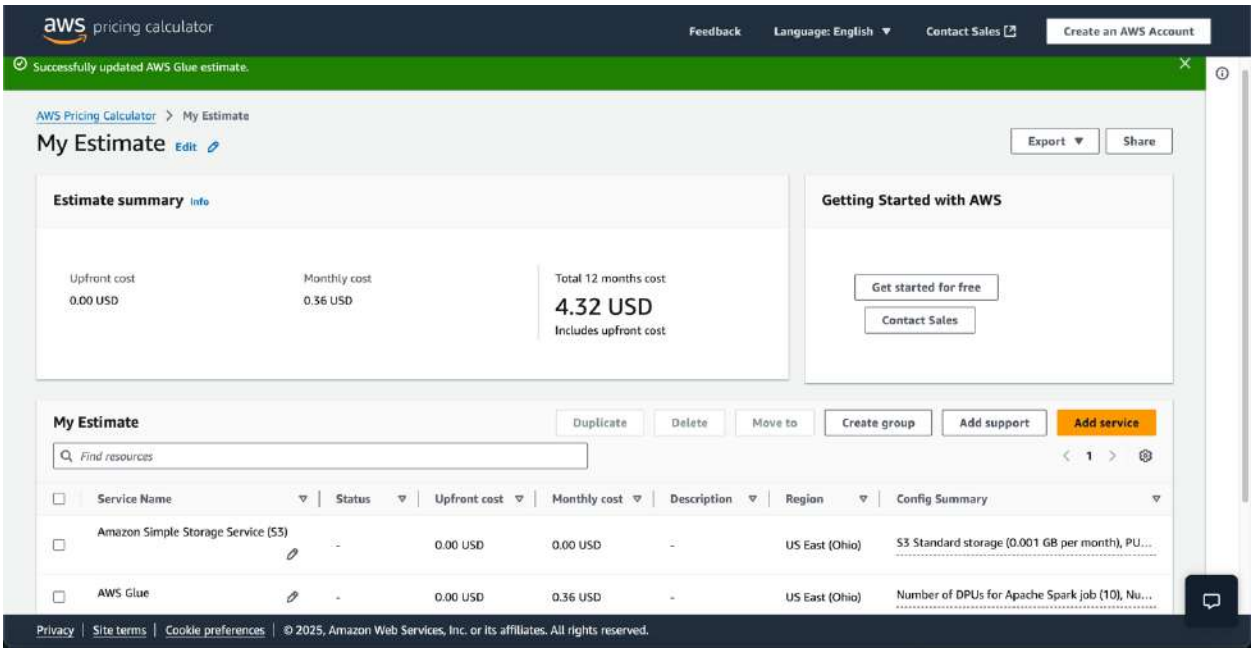
Note. The screenshot shows the cost estimation. Source: AWS cost calculator.

Explanation: My individual cost is 17.76 USD for total 12 months. Specifically, I used AWS S3 and AWS Glue. Regarding AWS S3, I used 0.000928 GB per month as the standard storage, and others are not necessary. Regarding AWS Glue, I used 10 DPUs for the Apache Spark job, 0.0625 DPUs for the Python Shell job, 0.000003 million per month for an object stored, and 0.000003 million per month for access requests, one interactive session for DataBrew, five nodes consumed for DataBrew job, and one crawler.

DAP Estimated Cost (Muhammad Zulqarnain Shahzad)

Figure 94

Cost Estimation of Muhammad Zulqarnain Shahzad



In this screenshot, I ran my project on AWS Pricing Calculator for the Business Licenses, Consulting and Management Services dataset on AWS, and I ended up with a total of about $4.32 for 12 months. That cost includes my S3 storage (which is tiny, since the dataset is only a few hundred KB) and my AWS Glue jobs (for data cleaning, cataloging, and the ETL pipeline). Because my data is so small and my ETL job runs for just 2 minutes, the monthly bill is super low. I didn't add any cost for Athena, because I either haven't run queries there or only ran a few small ones, which would be negligible anyway. So, if I only used S3 and Glue, $4.32 a year totally makes sense.

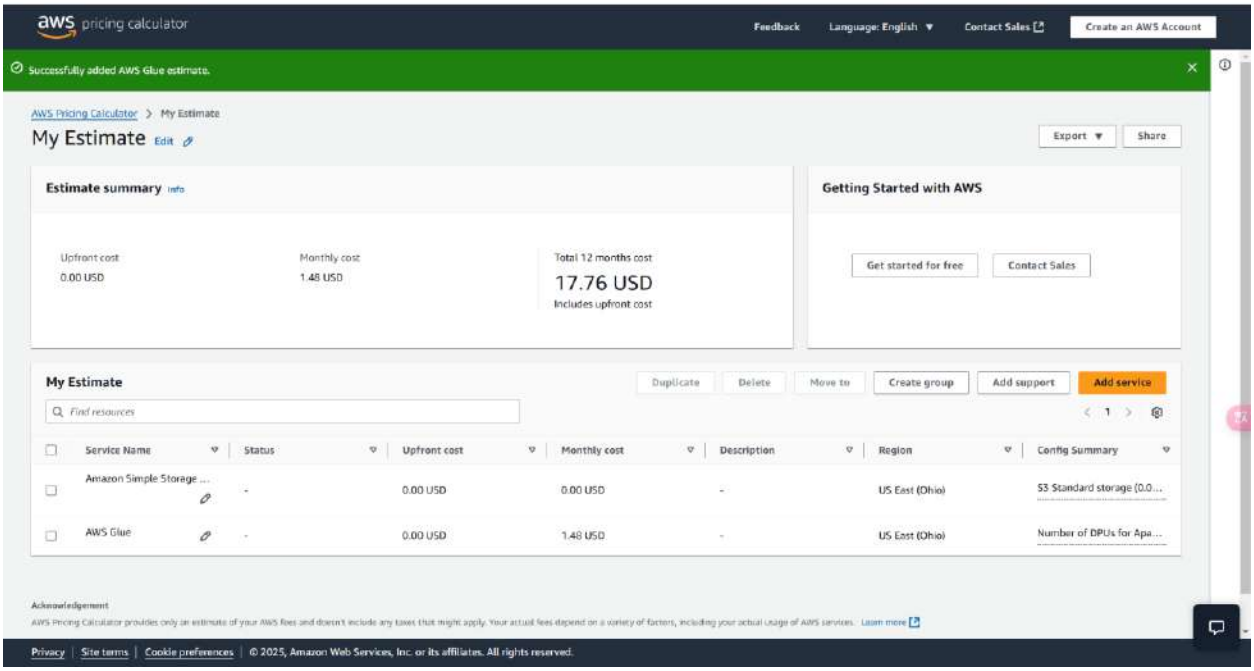
Basically, I'm paying a few cents a month to store the raw and cleaned data in S3, and the rest of the cost comes from Glue usage during my short ETL runs. If I ever scale up the data or

run more frequent jobs, this cost will increase but for now, it's perfect for a small project. This also shows that AWS can be pretty cheap when your dataset and usage are minimal.

DAP Estimated Cost (Haoran Xu)

Figure 95

Cost Estimation of Haoran Xu



Note. This screenshot is showing the estimate cost on AWS pricing calculator. Source: AWS pricing calculator.

My total cost estimation contains two main parts. Firstly, I used 0.0133GB monthly S3 standard storage. And I don't use other services like PUT, COPY, POST, GET, etc.

Secondly, I use AWS Glue for these uses:

1. Profiling and Cleaning in AWS Data Gluebrew. It contains 1 interactive session for Data Gluebrew.

2. Cataloging, Crawler and Summarization in AWS Glue. It contains both 0.000002 million objects stored and accesses per month. Using 10 DPUs for Apache Spark job and 0.0625 DPUs for Python shell job.

Total will be 1.48 USD cost per month and 17.76 USD cost annually.

Conclusion

Our project demonstrates a comprehensive, end-to-end data analytic platform built on AWS for the City of Vancouver, showcasing the power of cloud-based analytics across multiple datasets. As a team, we collectively tackled every required step, from data ingestion to cost estimation, while each member focused on a distinct dataset and set of analytical questions.

We began by extracting our datasets from the City of Vancouver Open Data Portal, each dataset was carefully selected to meet our volume requirements and provide valuable insights. Whether it was the Public Trees, 3-1-1 Contact Metrics, Business Licences for Consulting and Management Services, or the Animal Control Inventory Register, every dataset was first uploaded to dedicated raw S3 buckets. This ensured that the source data, including critical details like delimiters (e.g., the semicolon in our CSV files), was preserved without alteration.

Next, using AWS Glue DataBrew, each team member profiled and cleaned their dataset. We addressed data quality issues by correcting delimiter settings, standardizing column names, reformatting date fields, and removing unnecessary or redundant columns. This uniform cleaning process not only enhanced data consistency but also laid a strong foundation for accurate analysis.

To organize our cleaned data for subsequent processing, we used AWS Glue Crawlers to automatically catalog the datasets, generating well-defined database schemas. This step was

crucial for enabling efficient querying and integration of our data through AWS Athena and subsequent ETL processes.

For the transformation and summarization phases, each member implemented their own ETL pipelines in AWS Glue applying targeted filtering, aggregation, and schema mapping to derive the key metrics required by our descriptive analysis questions. While the specific transformation details varied depending on the dataset (e.g., average tree diameter versus height range, call handling metrics, Number of licenses and average number of employees, or animal impoundment statistics), the underlying methodology was consistent. We ensured that our final outputs were stored in clearly designated folders (transform, system, and user) for both technical verification and user-friendly access.

Finally, we conducted a detailed cost estimation using the AWS Pricing Calculator. By carefully accounting for our minimal storage needs, short duration ETL jobs, and light query loads, our estimated annual cost remained very low. This not only highlights the cost-efficiency of AWS for small-scale projects but also demonstrates the scalability of our platform should the data volume or processing frequency increase, our approach could easily adapt.

In summary, our group project fulfills all the requirements: we designed and implemented a data analytic platform that supports descriptive analysis and provided a thorough report that breaks down each step of the process. By integrating individual efforts into a cohesive, cloud-based solution, we've shown that the City of Vancouver can leverage AWS to drive data-based decision-making with minimal cost and high efficiency. Our project stands as a robust, scalable foundation for future data analytic endeavors, and we're proud of the collaborative effort that made it possible.

References

Animal control inventory - register. (2019). Vancouver.ca. https://opendata.vancouver.ca/explore/dataset/animal-control-inventory-register/information/?sort=dateimpounded&refine.dateimpounded=2024

City of Vancouver. (2025). *Business licences – consulting and management services*. City of Vancouver Open Data Portal. Retrieved February 25, 2025, from https://opendata.vancouver.ca/

Group Project - AWS Data Analytic Platform for The City of Vancouver

Gyanvi Bhardwaj (2320329)

Peng Wang (2306811)

Muhammad Zulqarnain Shahzad (2306553)

Haoran Xu (2317218)

University Canada West

BUSI653: Cloud Computing Technologies (HBD-WINTER25-03)

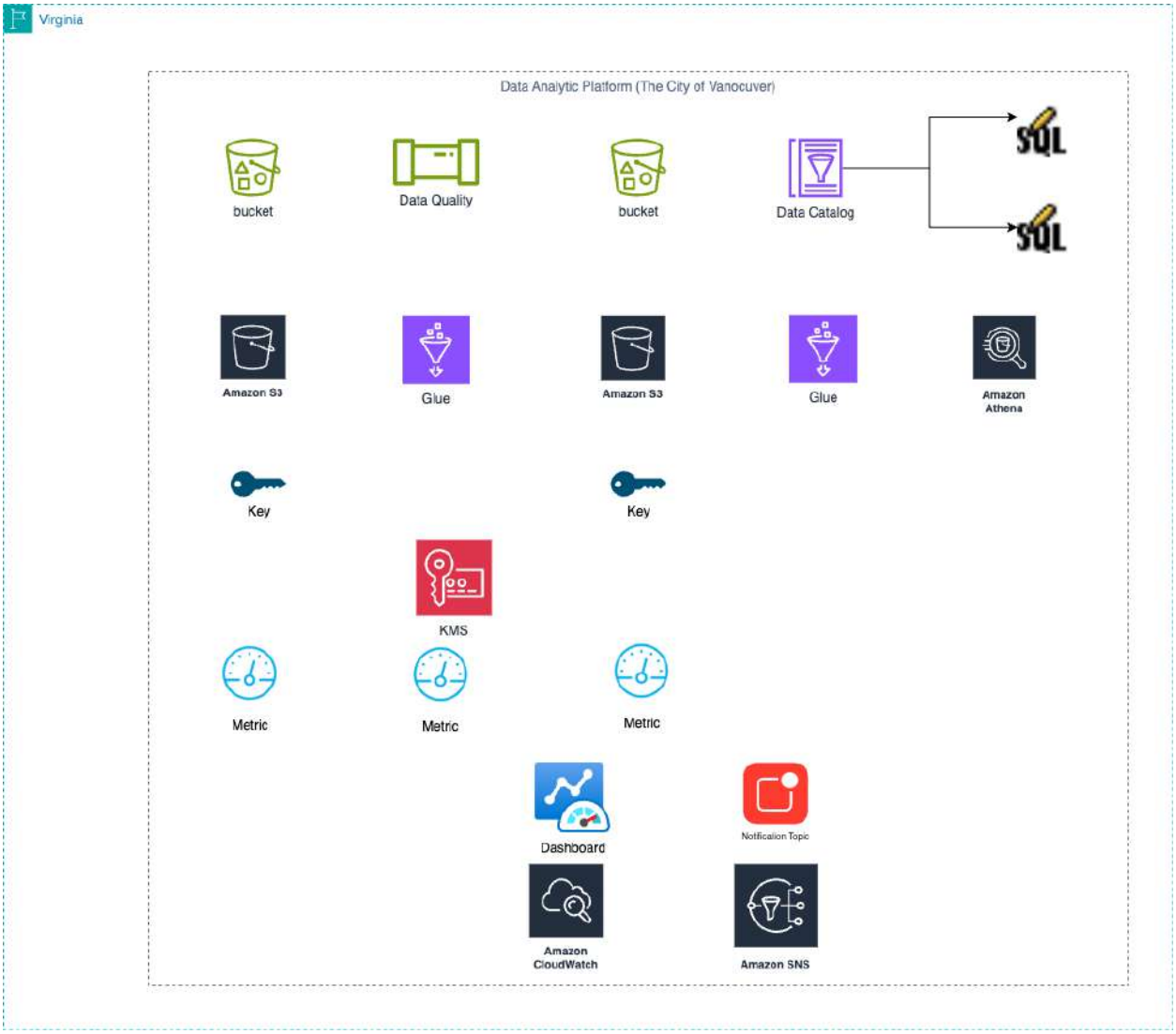
Dr Mahmood Mortazavi Dehkordi

March 23, 2025

Table of Contents

Figure 1

DAP requirements for the City of Vancouver



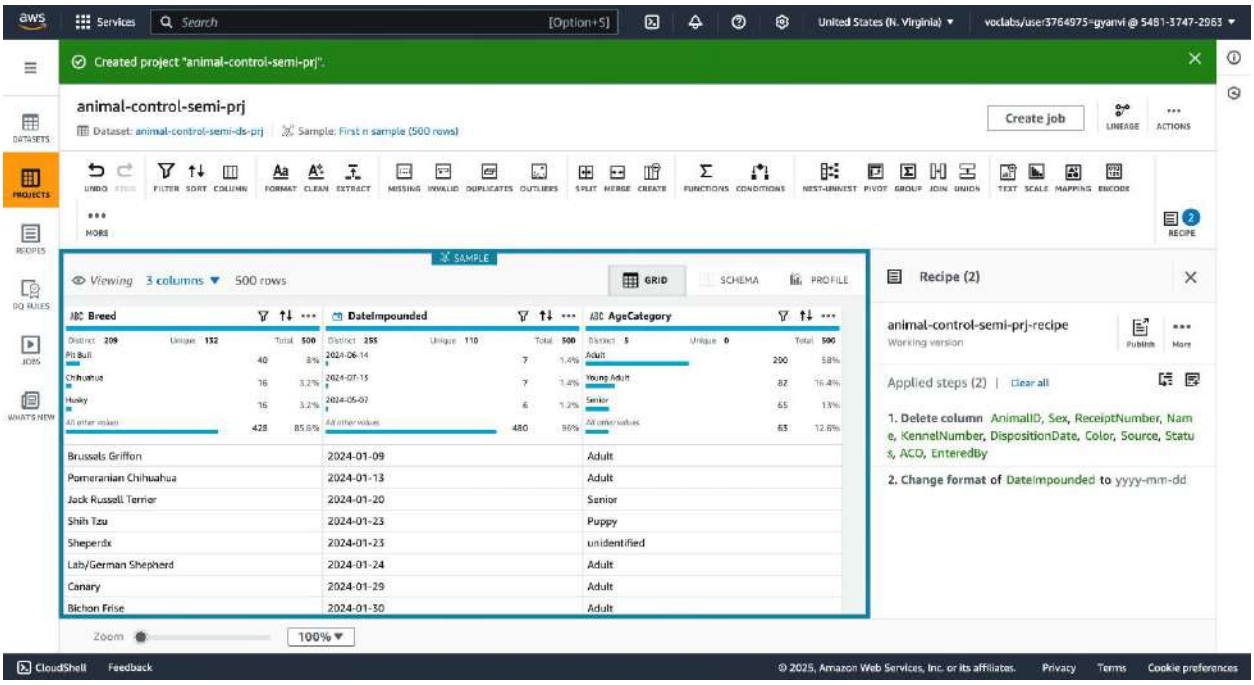
Note. This picture shows the procedure our team needs to implement the DAP to support the requirements of the City of Vancouver *(Draw.io, self-work).*

DAP Implementation (Individual work - Gyanvi Bhardwaj)

Step 5: Data Analysis

The data analysis step was carried out using AWS Athena which employs SQL queries to analyze data. A new folder was created in the raw zone of the AWS S3 bucket and named 'animal-control-semi-ds'. This contained the cleaned dataset. Using Glue Databrew, the columns that were not required to answer the two business questions were dropped. This also ensures that the cost associated with scanning the data is lowered as only the required data is available in the dataset provided for analysis to Athena. Now, the dataset is structured and ready for analysis using Athena.

Figure 2: Cleaned dataset ready for analysis

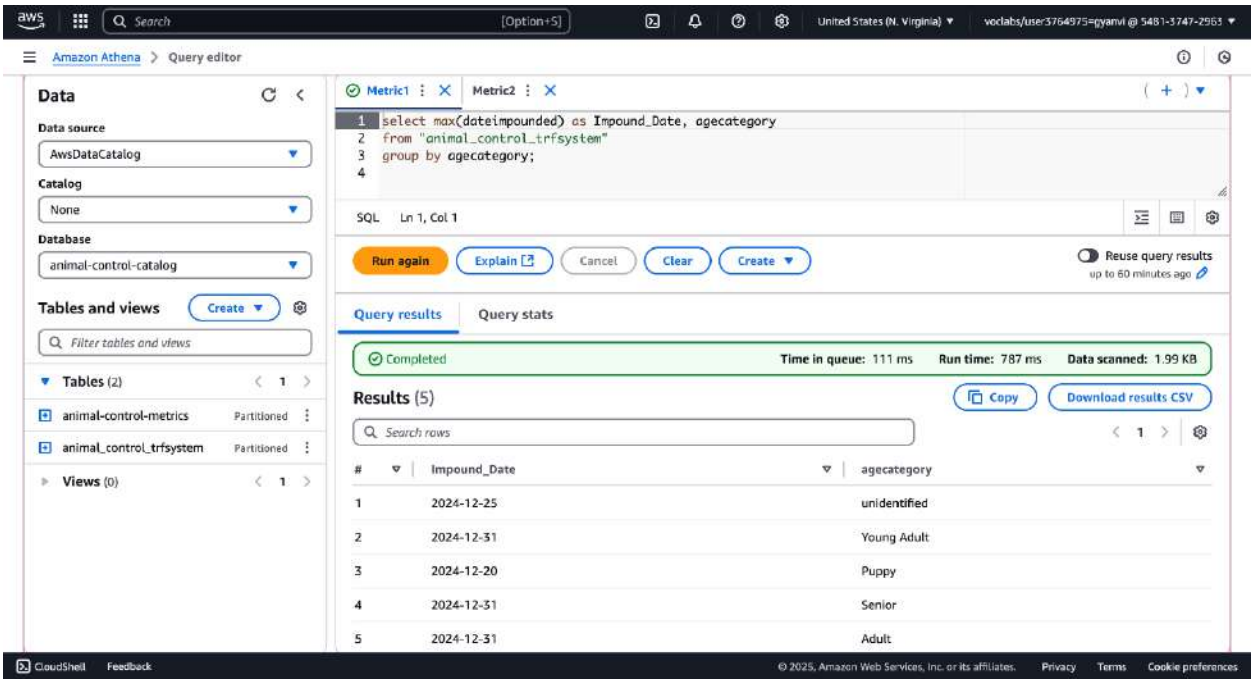


Note. The cleaned dataset is further formatted to drop unnecessary columns. (Source: AWS Glue, *self-work*)

AWS Athena query editor takes our Data catalog as the data source and answers the two business questions. For our analysis, we need only 3 columns- date impounded, breed and age category.

The SQL query for business question 1 selects the maximum date of impoundment and creates an alias as ‘Impound_Date’ and groups it by the age category. The result gives the recent impound date for each animal age category.

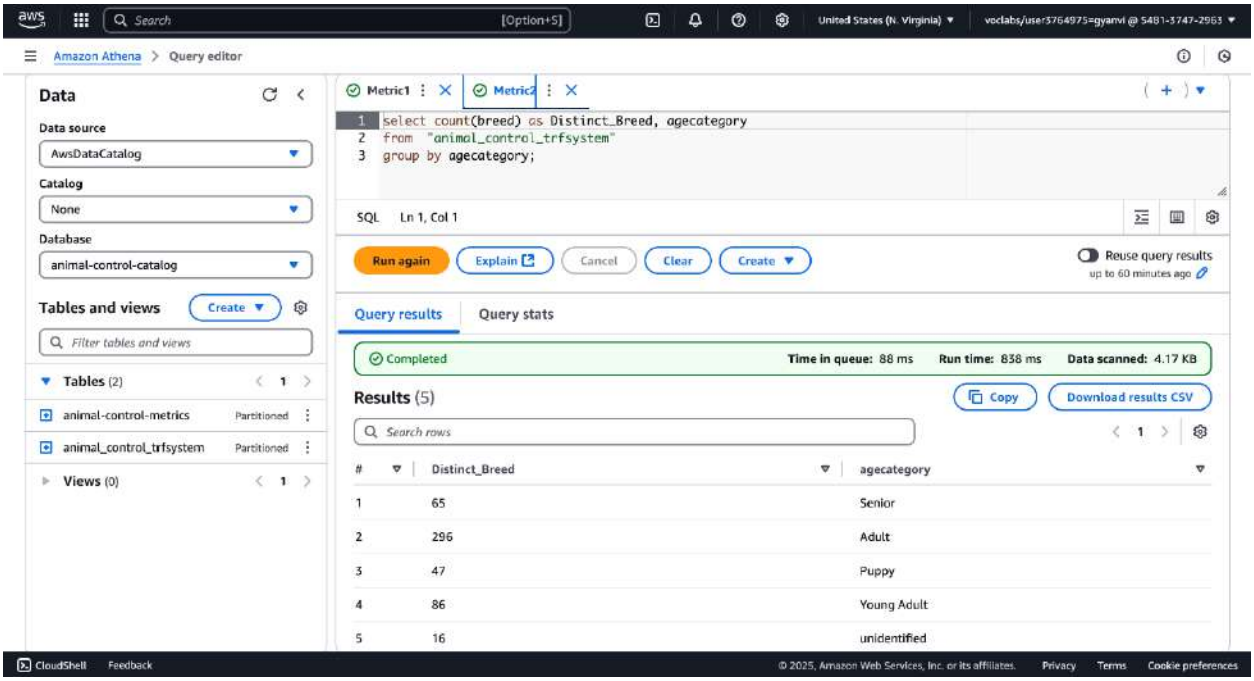
Figure 3: AWS Athena Business Question 1 SQL query



Note. AWS Athena SQL query for business question 1. (Source: AWS Athena, *self-work*)

The SQL query for business question 2 selects the number of breeds and creates an alias as ‘Distinct_Breed’ and groups it by the age category. The result gives the number of distinct breeds for each animal age category.

Figure 4: AWS Athena Business Question 2 SQL query

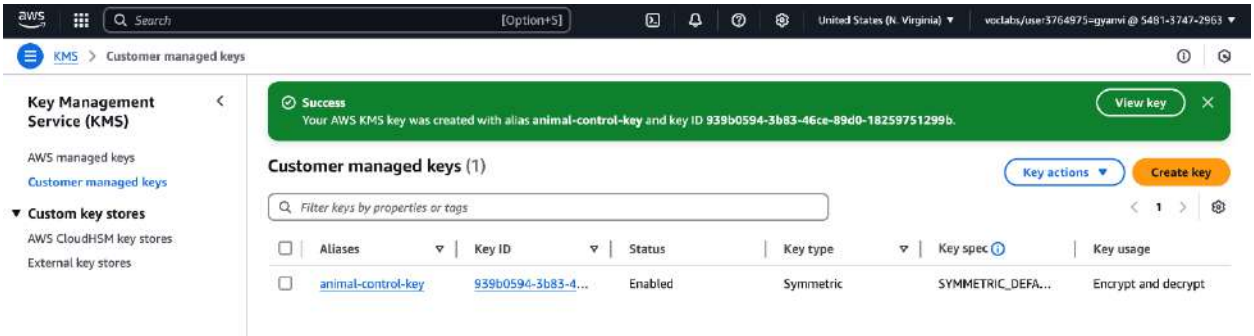


Note. AWS Athena SQL query for business question 2. (Source: AWS Athena, *self-work*)

Step 6: Data Security

Data security deals with creating and managing keys while controlling encryption activities to protect our data. Amazon Key Management System has been used to create a key called ‘animal-control-key’. LabRole is the key administrator and the key user. Encryption has been enabled for this key.

Figure 5: Key created for animal control inventory data



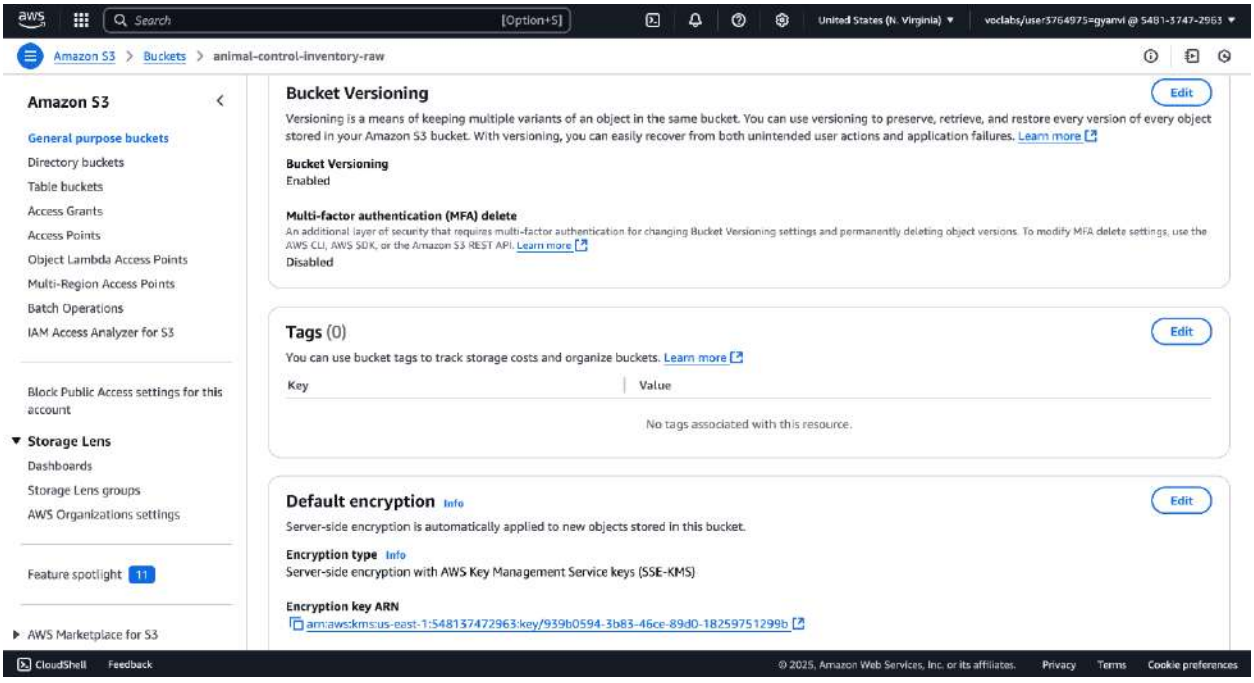
Note. Key created for encryption to be used in animal control inventory dataset. (Source: AWS KMS, *self-work*)

Next, the raw and transformed buckets were managed to provide encryption and replication for enhancing data security. The key ‘animal-control-key’ is used for both the raw and transformed buckets to enable encryption.

For bucket versioning, we have chosen this option to keep our data safe by creating multiple copies of it for any unforeseen event. By creating a new S3 bucket called ‘animal-control-inventory-backup’ for replication rules, the replication rule named ‘animal-control-inventory-replication’ is stored in the new bucket.

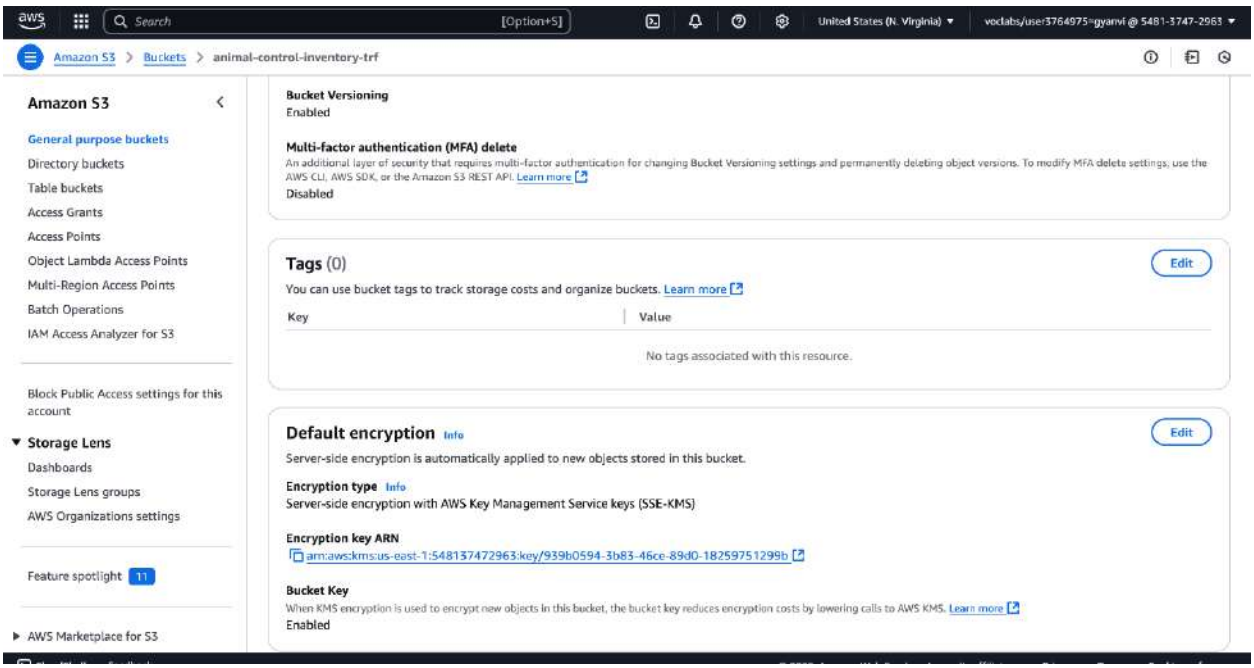
Similarly, for creating the replication rule for the transformed bucket, we have created another S3 bucket called ‘animal-control-trf-backup’. The replication rule called ‘animal-control-trf-replication’ is stored in this new bucket created.

Figure 6: Bucket versioning and encryption for raw bucket



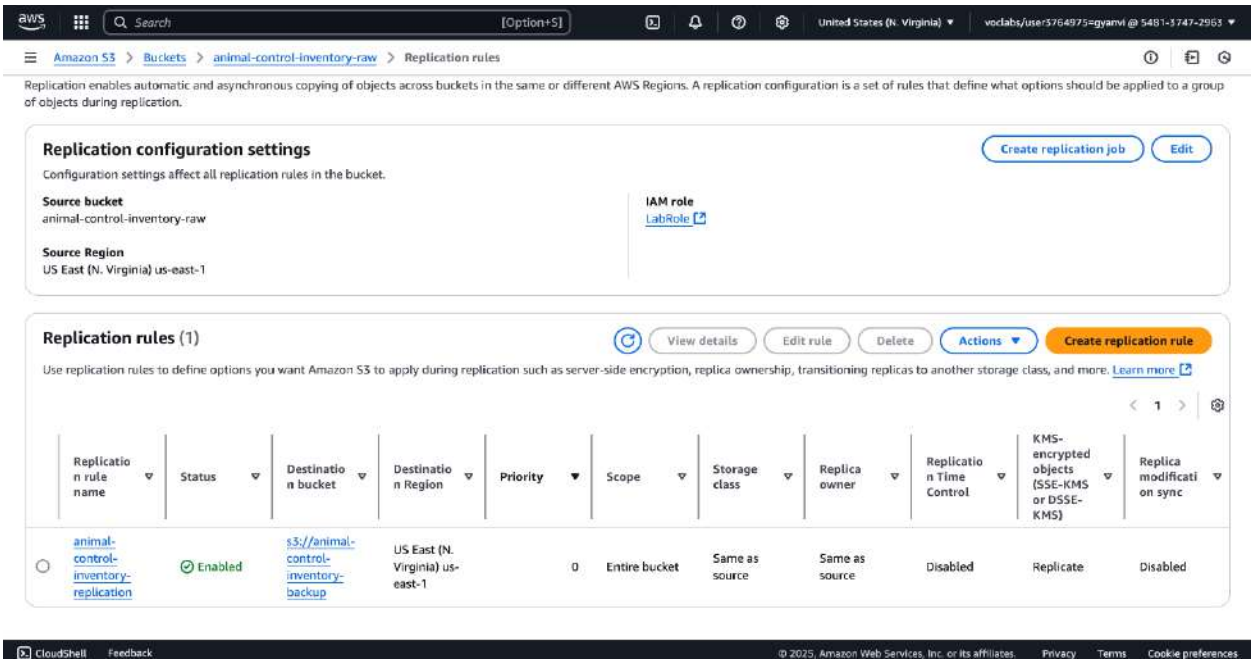
Note. Bucket versioning and encryption for the animal-control-raw bucket. (AWS S3, *self-work).*

Figure 7: Bucket versioning and encryption for transformed bucket



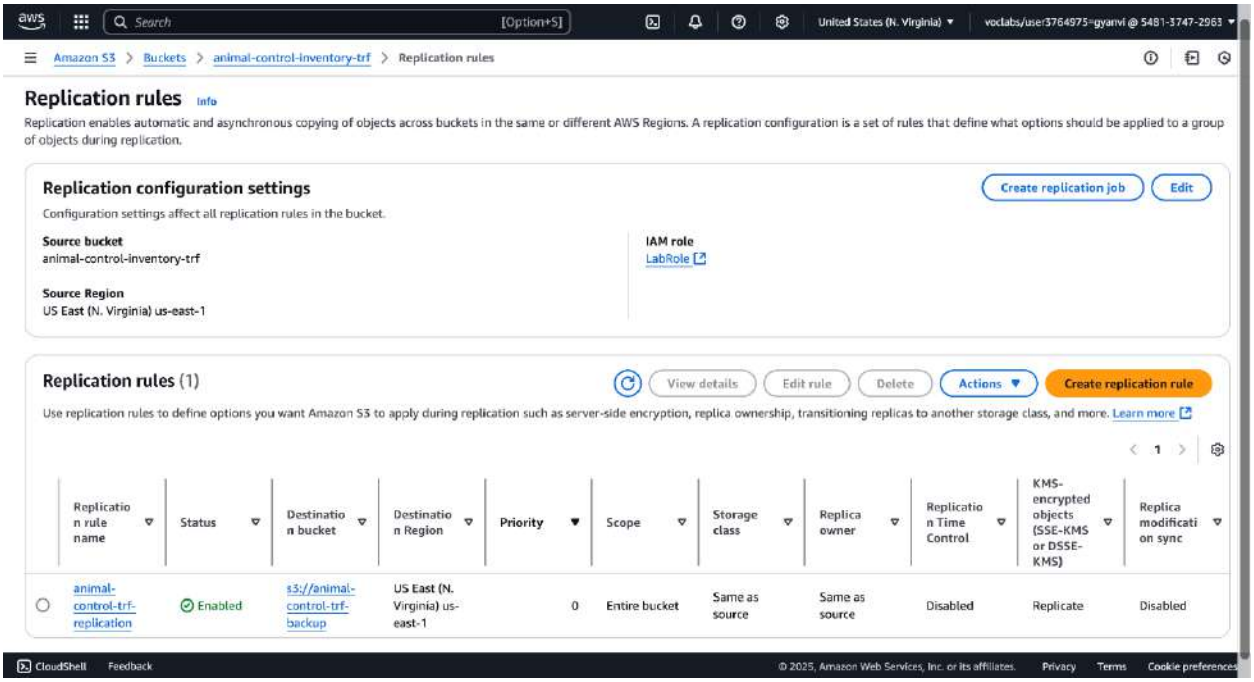
Note. Bucket versioning and encryption for the animal-control-trf bucket. (AWS S3, *self-work).*

Figure 8: Replication rule created for raw bucket



Note. A replication rule was created for the raw zone in the new raw-backup bucket. (AWS S3, *self-work).*

Figure 9: Replication rule created for transformed bucket



Note. A replication rule was created for the transformed zone in the new trf-backup bucket. (AWS S3, *self-work).*

Step 7: Data Governance

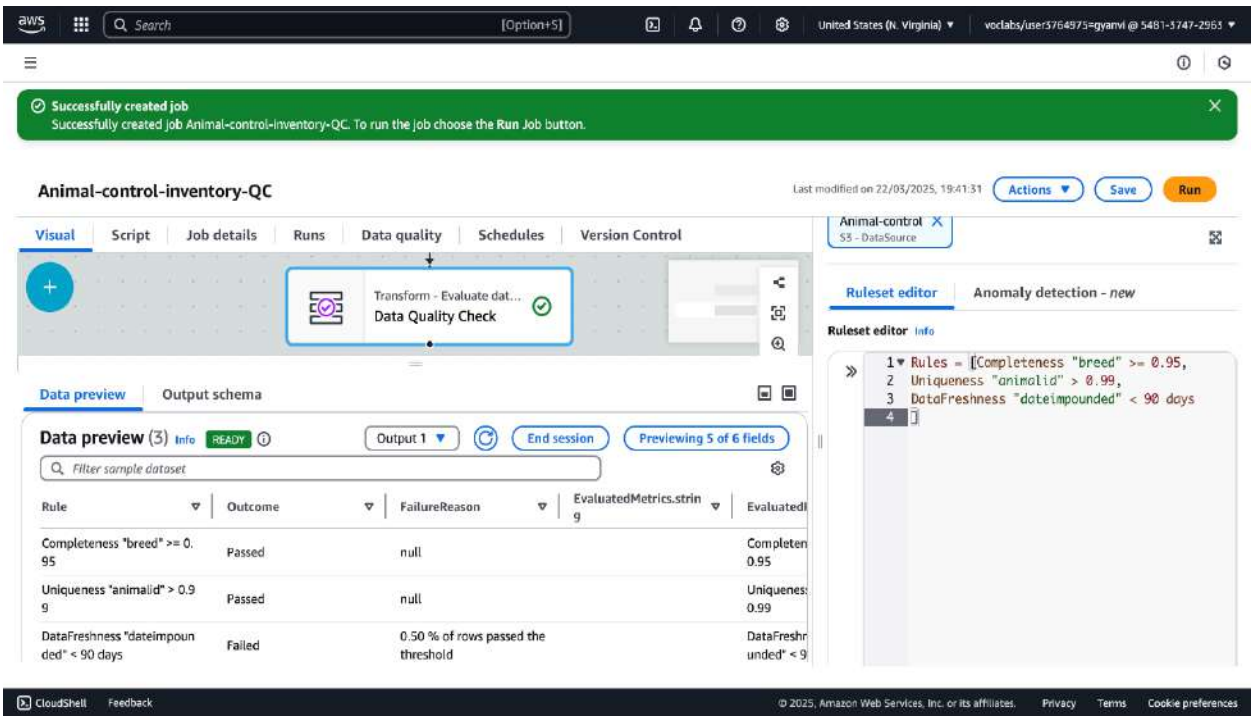
The data governance step was carried out in AWS Glue where the visual ETL pipeline was created. This pipeline involves checking data for its quality using three parameters namely, completeness, uniqueness and freshness. The pipeline is named ‘Animal-control-inventory-QC’ and takes the data from the animal-control-inventory raw bucket. Next, the ruleset editor creates the following three rules:

1. Completeness of the *‘breed’* column should be greater than or equal to 0.95

2. Uniqueness concerns the primary key column called *'animalid'* which should be greater than 0.99

3. Freshness has a date column called *'dateimpounded'* less than 90 days

Figure 10: Rules created for animal-control-inventory-QC



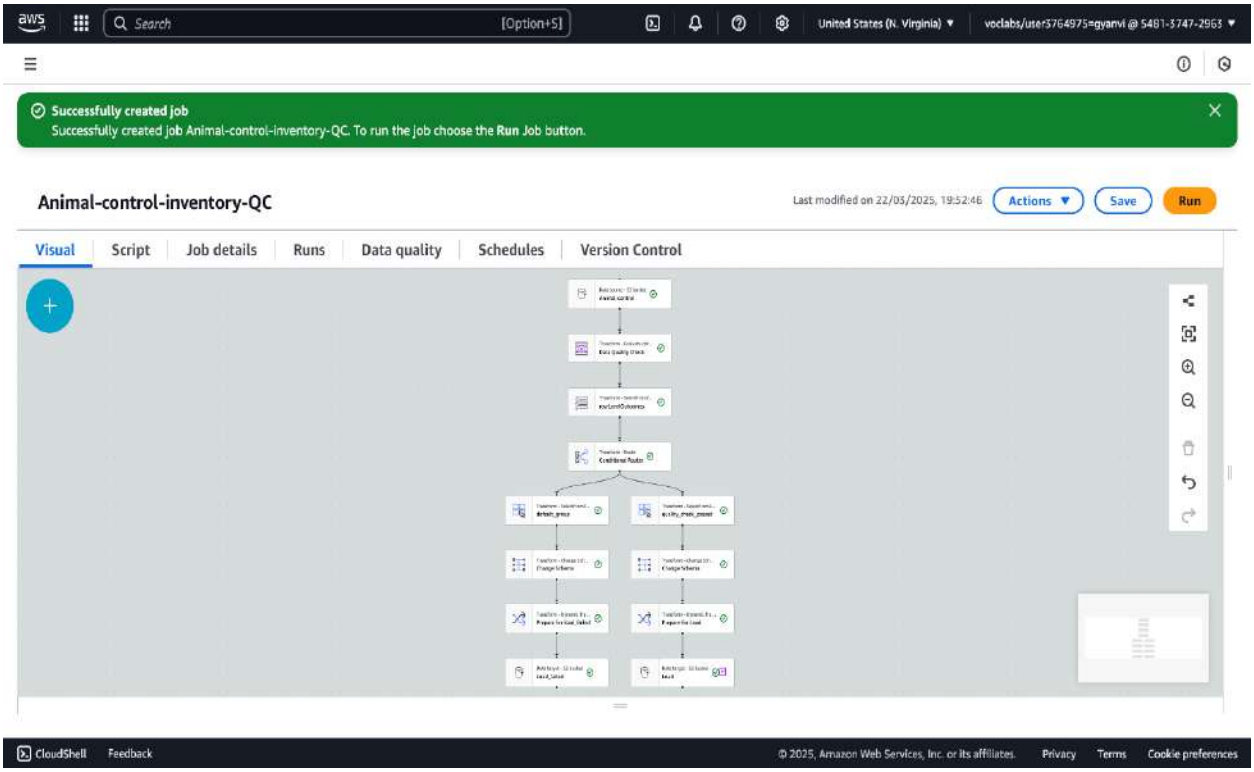
Note. Evaluate data quality under the transform section in the ETL pipeline. (AWS Glue, *self-work).*

Two of the rules have passed and one has failed based on the conditions provided under the ruleset editor. Now, the following steps have used transform features like transform-route for creating two groups called, '*quality_check_passed'* and the other is the default group.

The logical operator is AND and the key is the Data quality evaluation result. The operation is 'matches' as "Passed" is the string value. By employing the usual steps of dropping

unnecessary columns of data quality check under the Change Schema feature and then using auto-balancing for a single output, we perform the steps for both passed and failed sets.

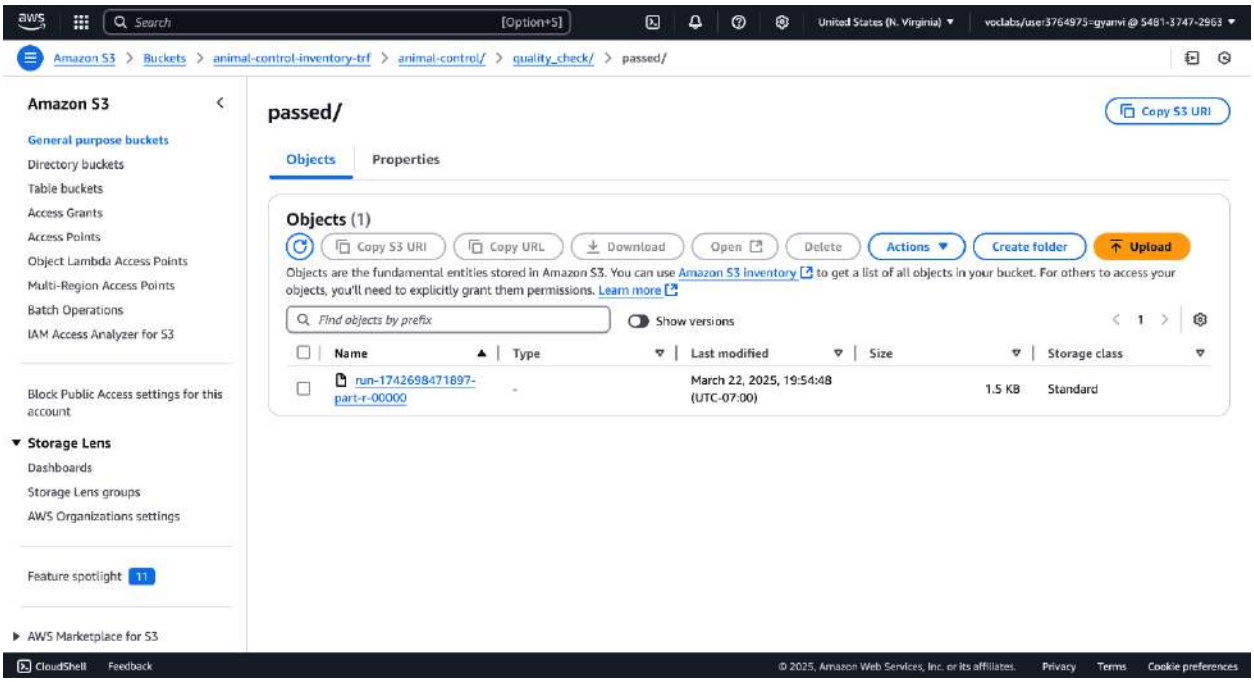
Figure 11: ETL pipeline for animal control inventory quality check



Note. The entire ETL pipeline was created for the data governance step for quality check of animal control inventory. (AWS Glue, *self-work).*

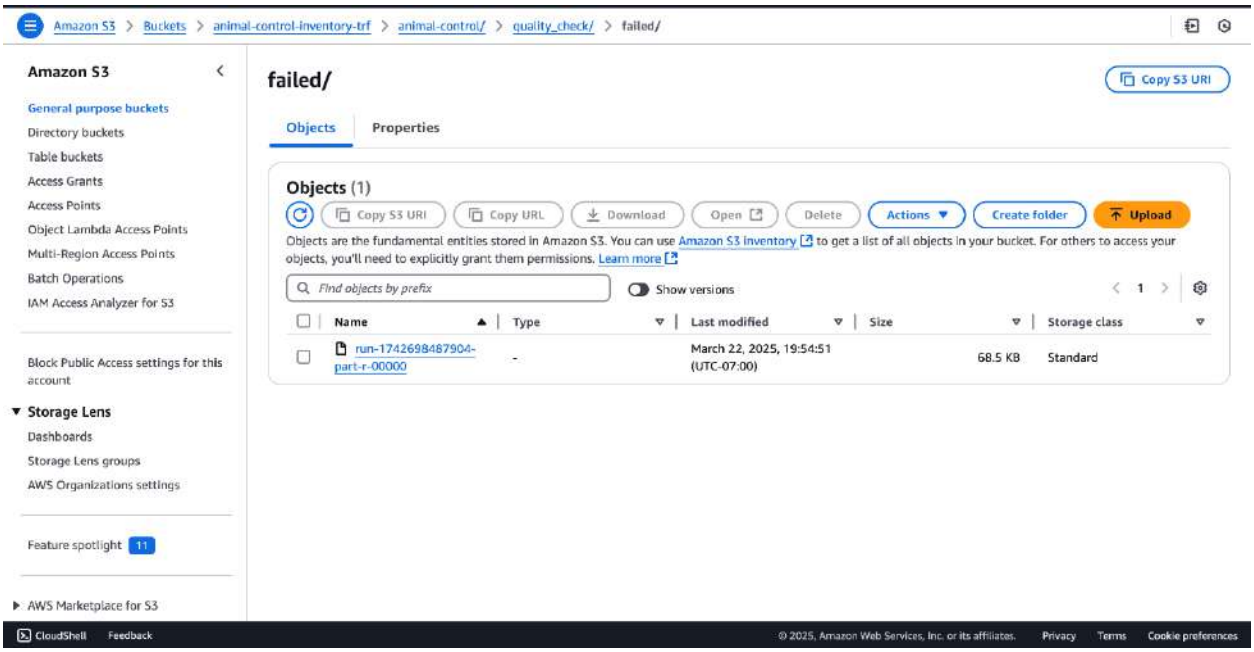
In the transformed bucket, we created a folder called ‘Quality_check’ which has two sub-folders called ‘Passed’ and ‘Failed’. The single CSV output for passed goes into the passed folder and the same goes for the failed folder.

Figure 12: Job output for ‘Passed’ folder



Note. AWS S3 transformed bucket containing the output of the passed data quality check (AWS S3, *self-work).*

Figure 13: Job output for the 'Failed' folder

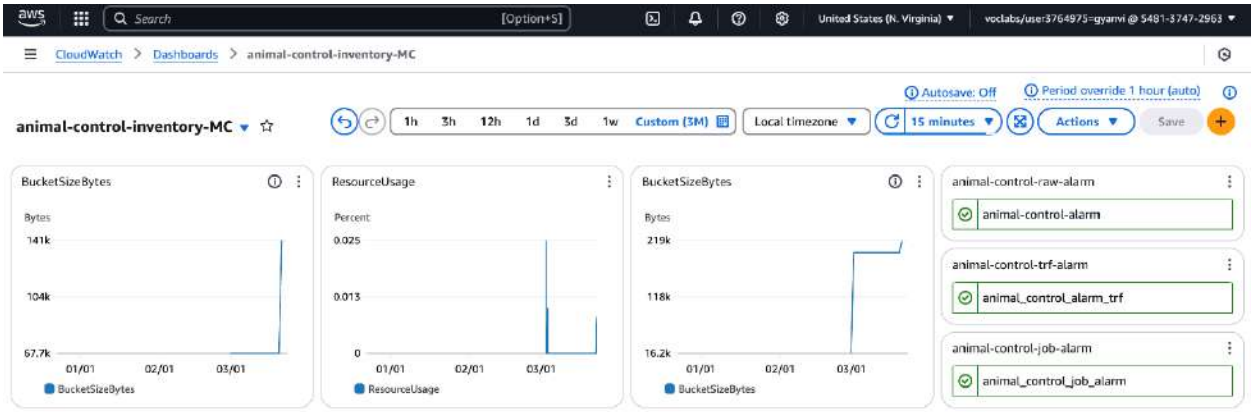


Note. AWS S3 transformed bucket containing the output of the failed data quality check (AWS S3, *self-work).*

Step 8: Data Monitoring

For data monitoring, we have used the AWS CloudWatch feature that monitors resource usage which is essential in cost and storage optimization. The dashboard created is named 'animal-control-inventory-MC' which contains 3 metrics and 3 alarms for the respective metrics. A simple line widget is used for the three metrics- Raw S3 bucket, Transformed S3 bucket and Glue JobRun. The timeline is customized for 3 months.

Figure 14: Dashboard created on CloudWatch



Note. The overview of the dashboard contains the three metrics and three alarms. (AWS CloudWatch, *self-work).*

Next, the alarms created for the S3 service have an average as the stat and 1 day as the time period. The Glue service has a time period of 5 minutes.

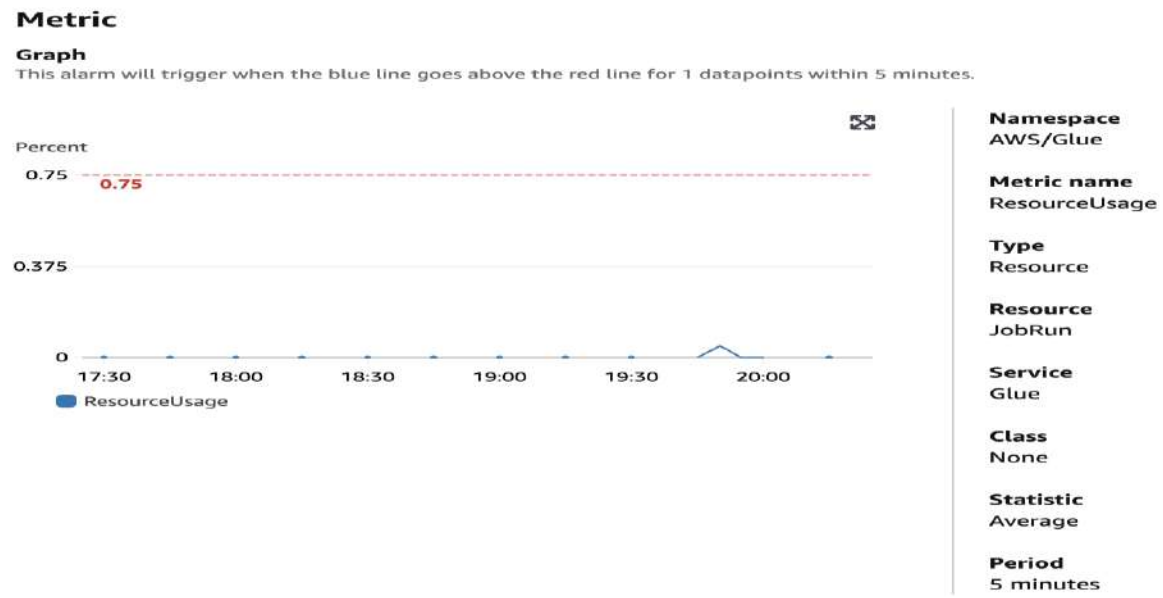
The threshold is set for the three metrics as:

1. Raw S3 bucket bucket size bytes: 180,000

2. Transformed S3 bucket size bytes: 300,000

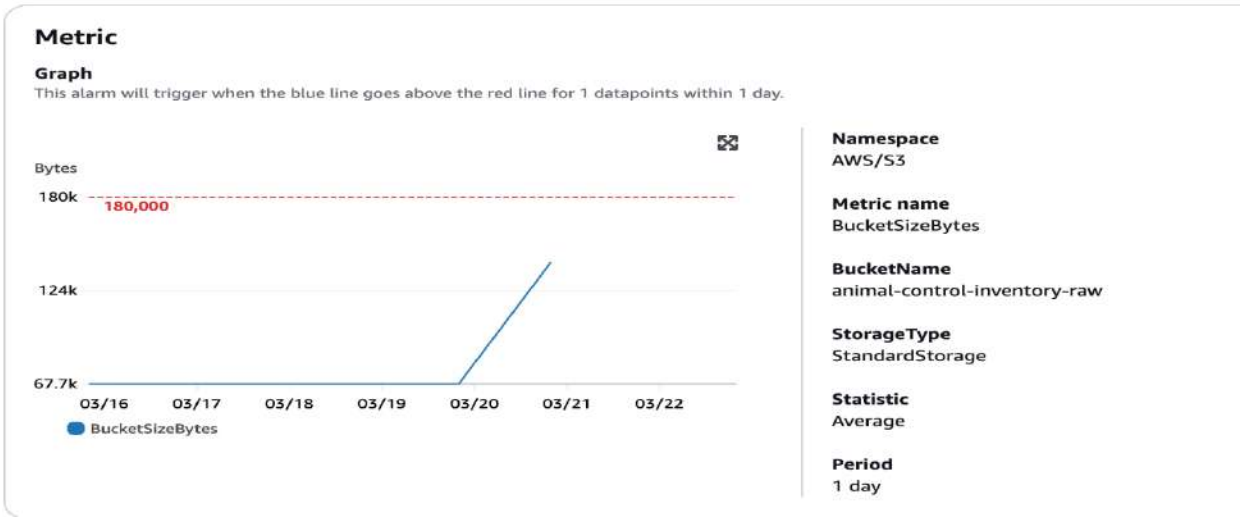
3. JobRun percent: 0.75

Figure 15: AWS Glue alarm



Note. Overview of the AWS Glue alarm with 0.75 percent as threshold. (AWS CloudWatch, *self-work).*

Figure 16: AWS S3 Raw bucket alarm



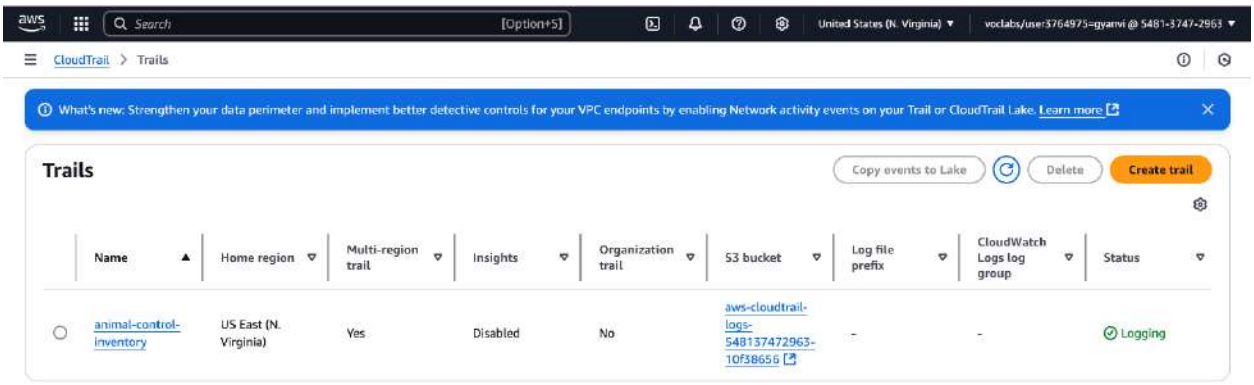
Note. Overview of the AWS S3 Raw bucket alarm with 180,000 bytes as threshold (AWS CloudWatch, *self-work).*

Figure 17: AWS S3 transformed bucket alarm

Note. Overview of the AWS S3 Transformed bucket alarm with 300,000 bytes as threshold (AWS CloudWatch, *self-work).*

We can use AWS Cloudtrail to monitor user activity and understand patterns of usage. For this, we have created a trail called ‘animal-control-inventory’ which contains all steps taken during the session.

Figure 18: AWS CloudTrail used to create a trail

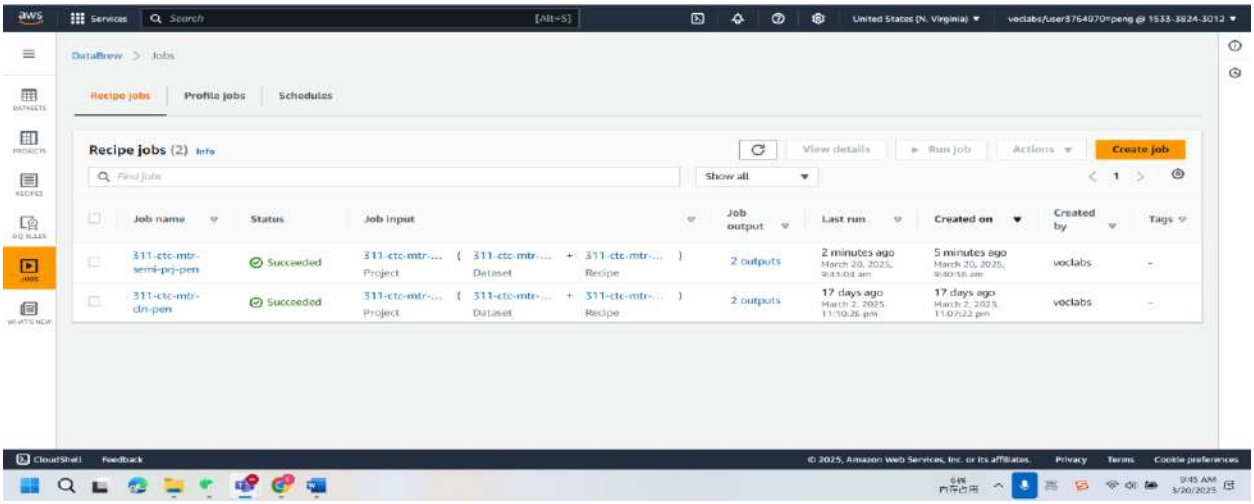


Note. Trail created in AWS CloudTrail to understand user activity and usage patterns for cost and resource optimization (AWS CloudTrail, *self-work).*

DAP Implementation (Individual work – Peng Wang)

Step 5: Data Analysis

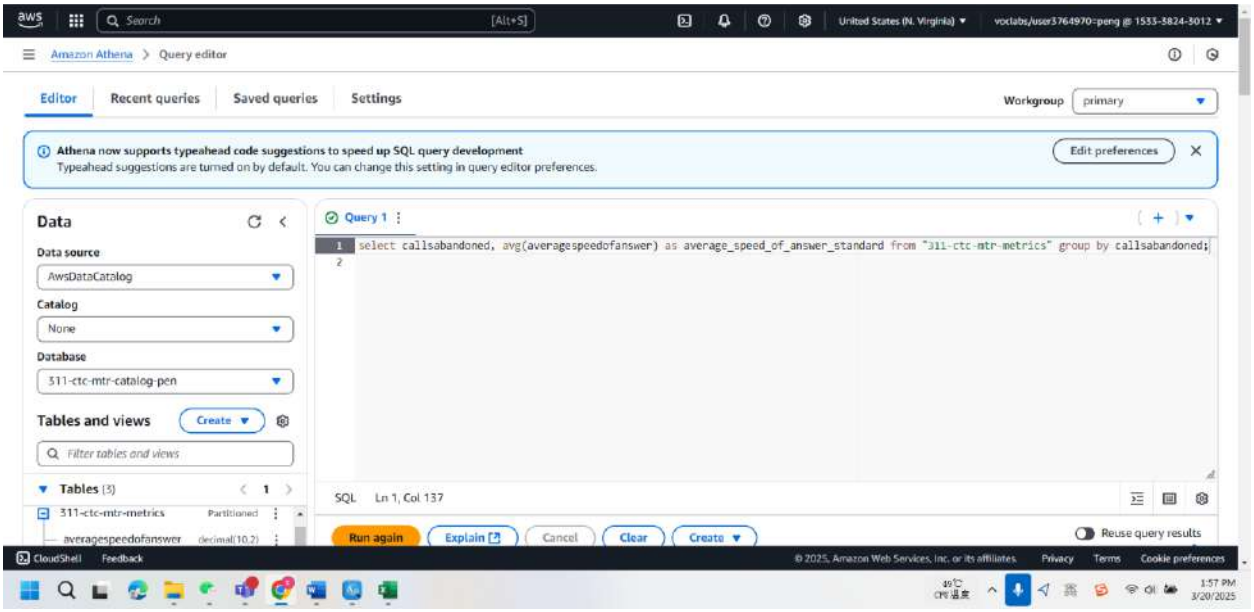
Figure 19: Semi-dataset jobs



Note. The screenshot shows the semi-dataset jobs. Source from AWS.

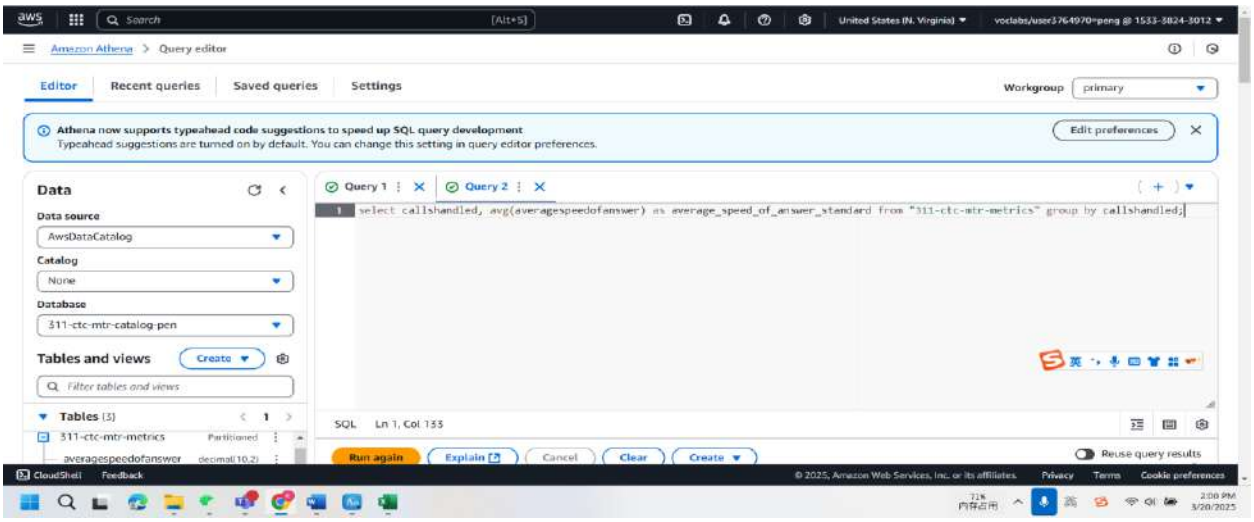
Explanation: I created the semi-dataset, but I found I have already created the cleaned dataset, which can be used for data analysis, so I think it is not necessary to create the semi-dataset.

Figure 20: SQL Syntax in Athena



Note. The screenshot shows the first SQL syntax in Athena. Source from AWS.

Figure 21: SQL Syntax in Athena



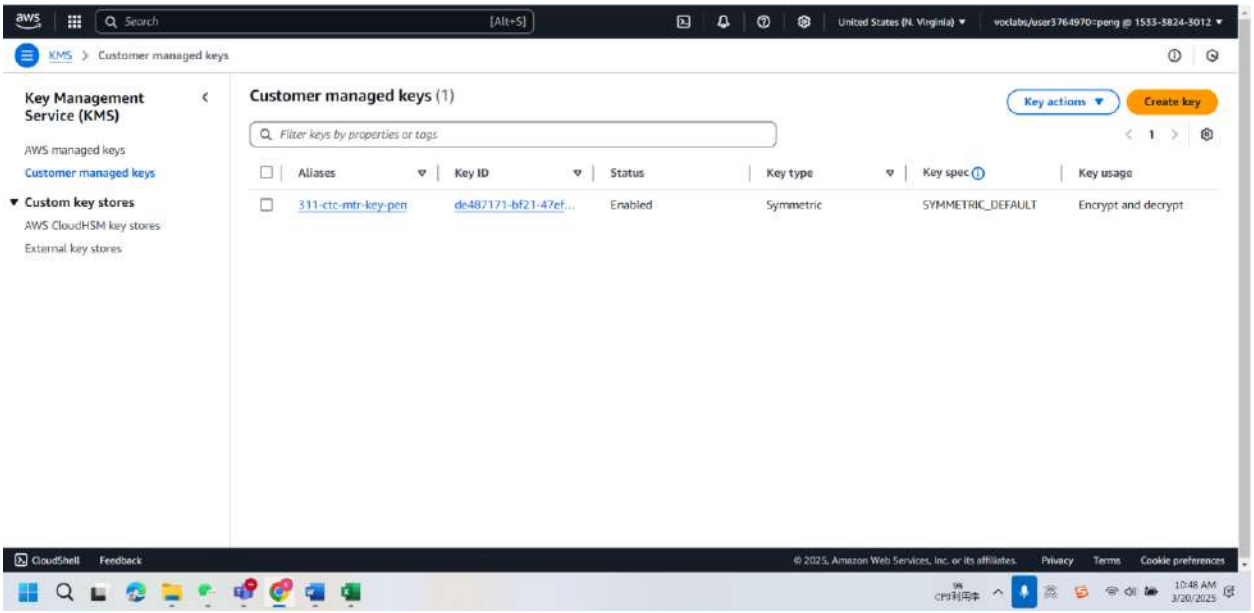
Note. The screenshot shows the second SQL syntax in Athena. Source from AWS.

Explanation: I analyzed through SQL syntax in Athena. That is because Athena provides the functions to proceed with business questions by SQL syntax. There are two questions that need

to be analyzed. The first one is the calls abandoned analysis, compared with an average speed of answer. The second one is the calls handled analysis, compared with an average speed of answer. Specifically, I set the standard value as the average of average speed of the answer. In this case, calls abandoned data and calls handled data can be shown whether it is more than this standard value or less than the standard value.

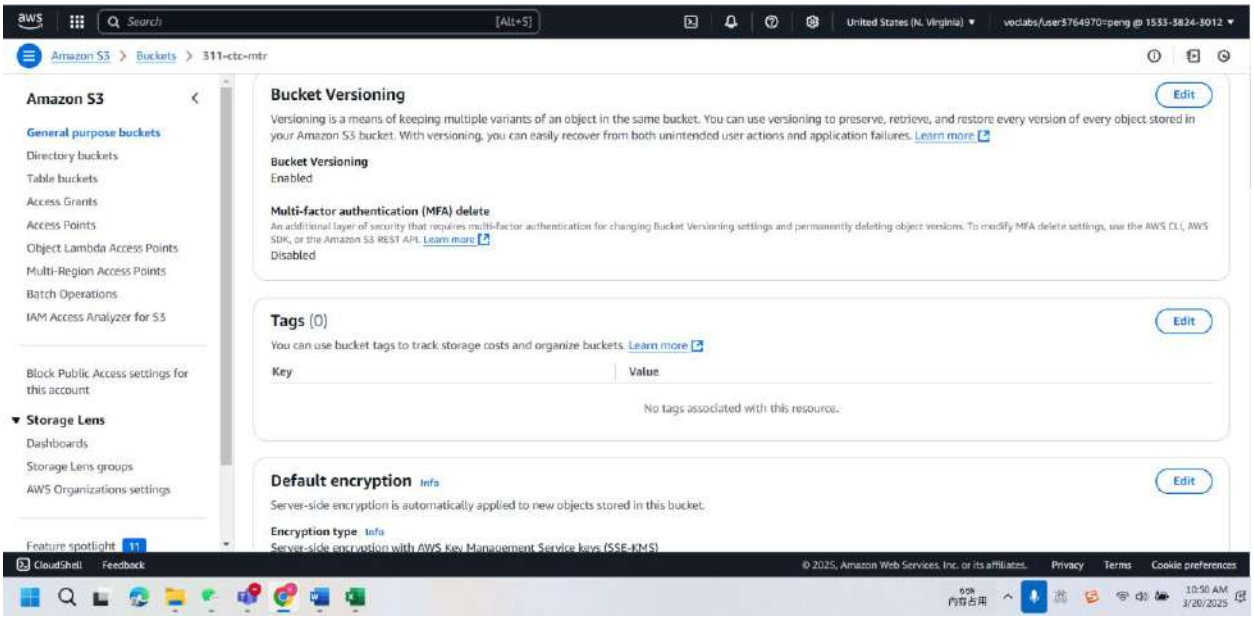
Step 6: Data Security

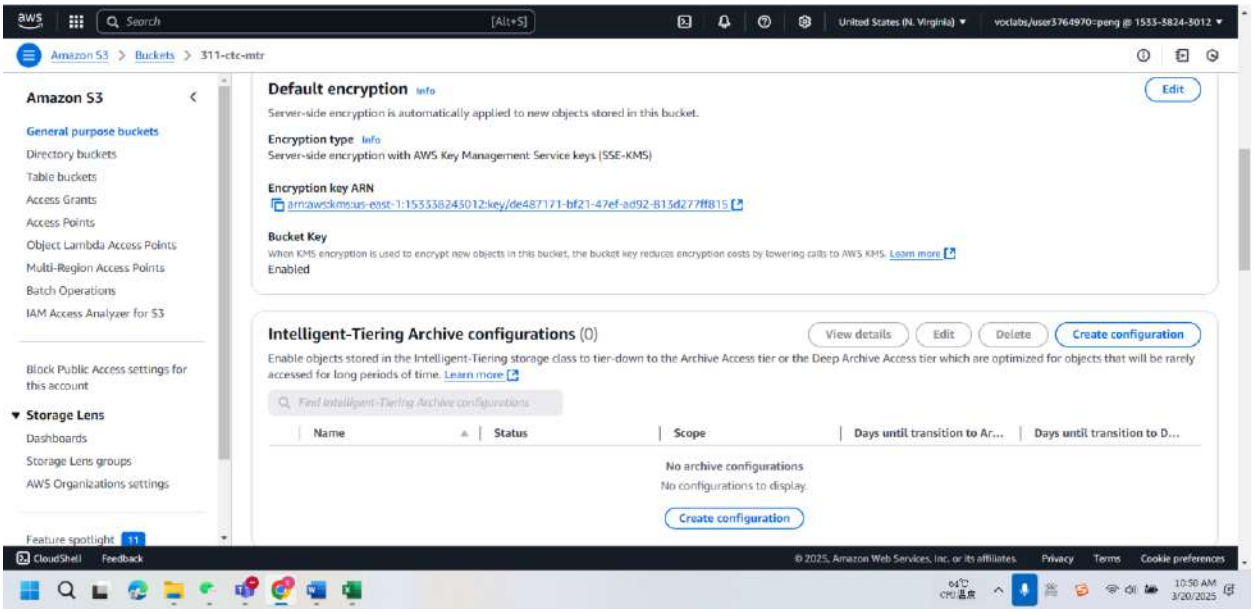
Figure 22: Key Management Service



Note. The screenshot shows the key management service. Source from AWS.

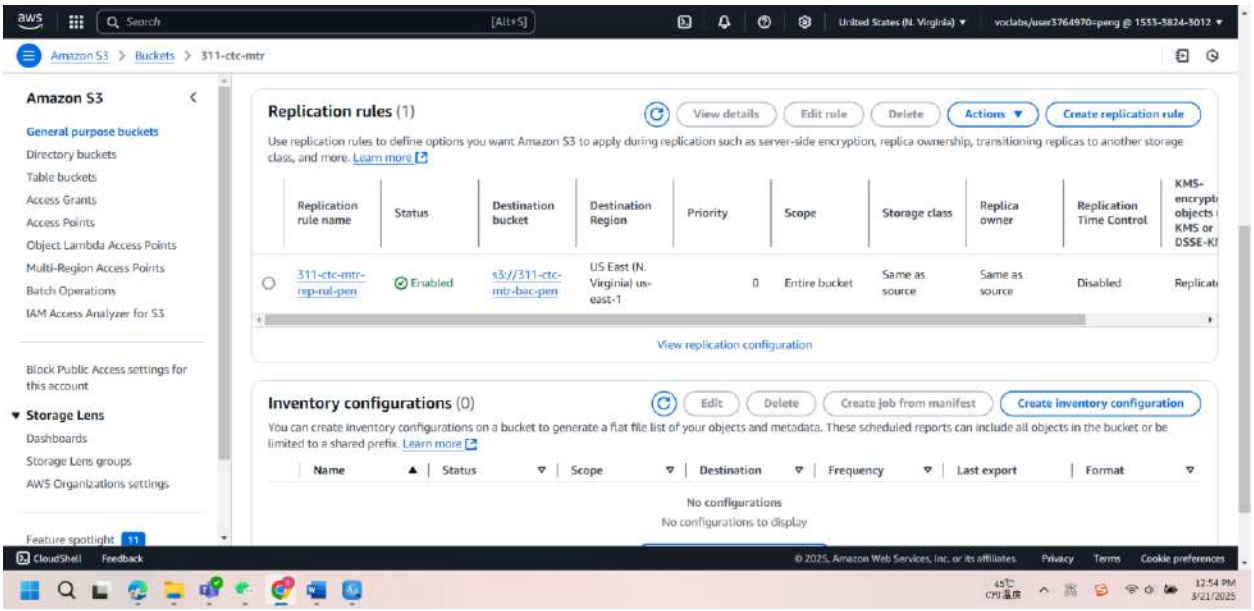
Figure 23: Raw Bucket Versioning and Default Encryption





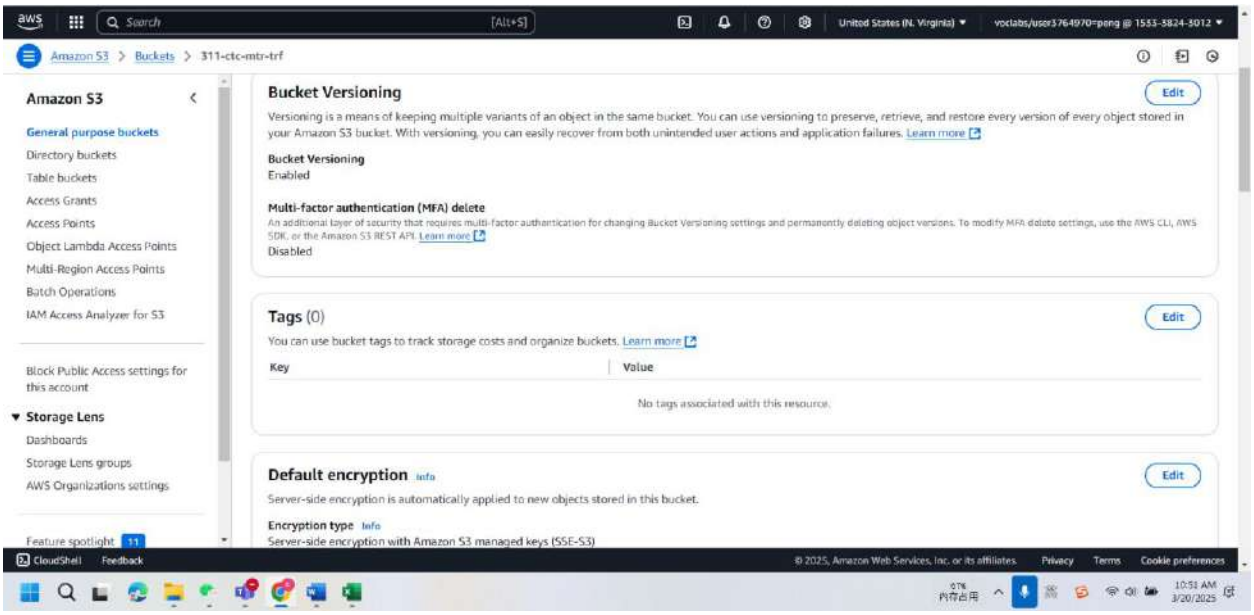
Note. The screenshot shows the raw bucket versioning and default encryption. Source from AWS.

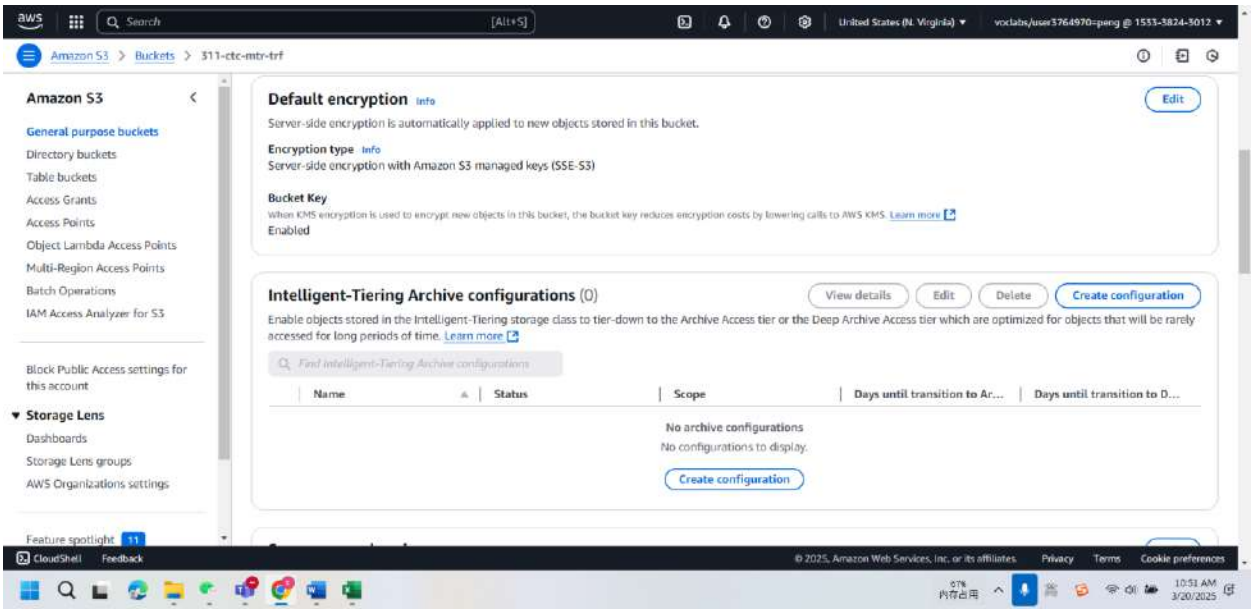
Figure 24: Replication Rules in Raw Bucket



Note. The screenshot shows the replication rules in the raw bucket. Source from AWS.

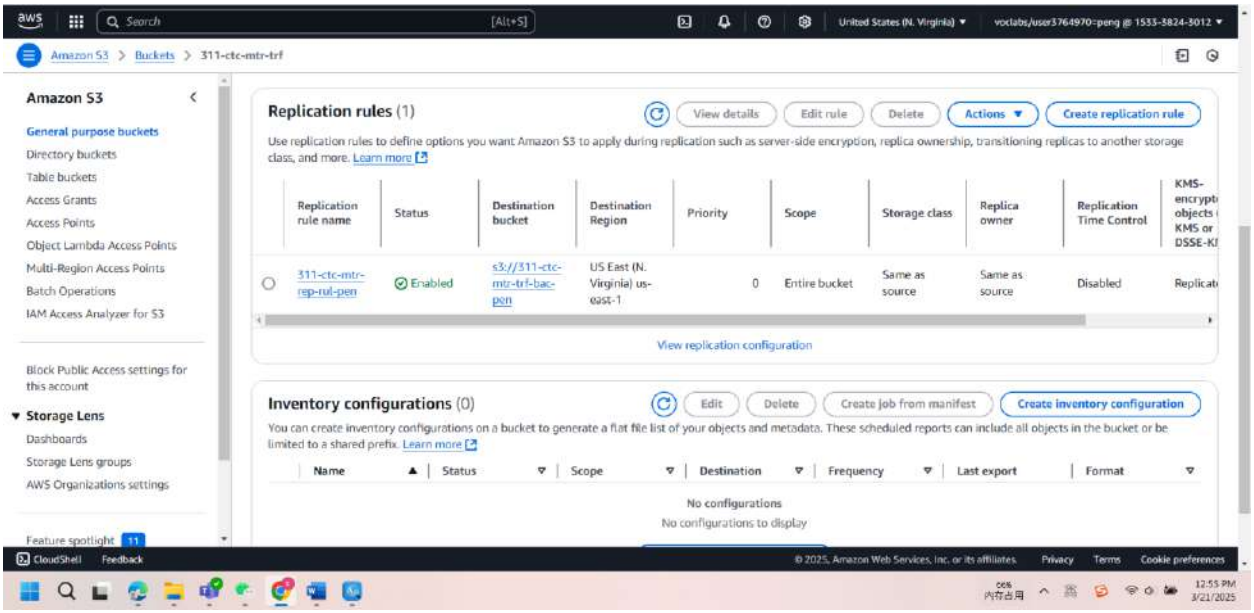
Figure 25: Transfer Bucket Versioning and Default Encryption





Note. The screenshot shows the transfer bucket versioning and default encryption. Source from AWS.

Figure 26: Replication Rules in Transfer Bucket

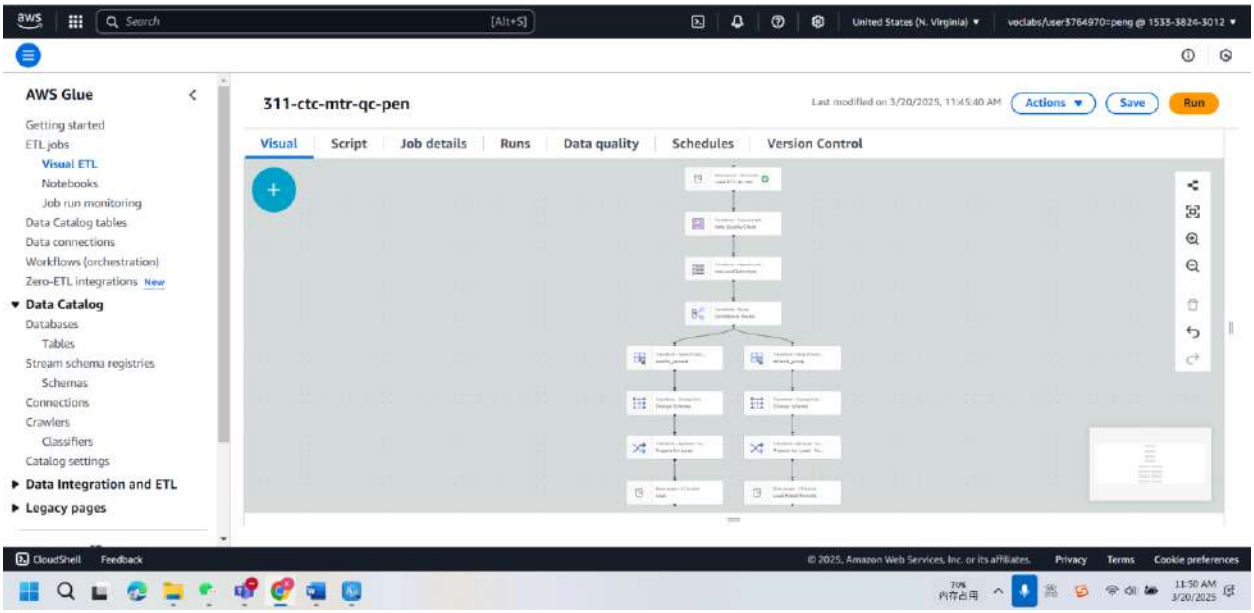


Note. The screenshot shows the replication rules in the transfer bucket. Source from AWS.

Explanation: I set the key through KMS, and then, I set the bucket versioning and default encryption. Finally, it is necessary to set the replication rules in the raw bucket and transfer bucket. The reason why I set the key in KMS is that KMS kyes can reduce the risk and encrypt the sensitive data, the bucket versioning is mainly to prevent data loss, and the default encryption is mainly to avoid unauthorized access to stored data. Overall, this operation can ensure data security.

Step 7: Data Governance

Figure 27: Quality Check in Visual ETL

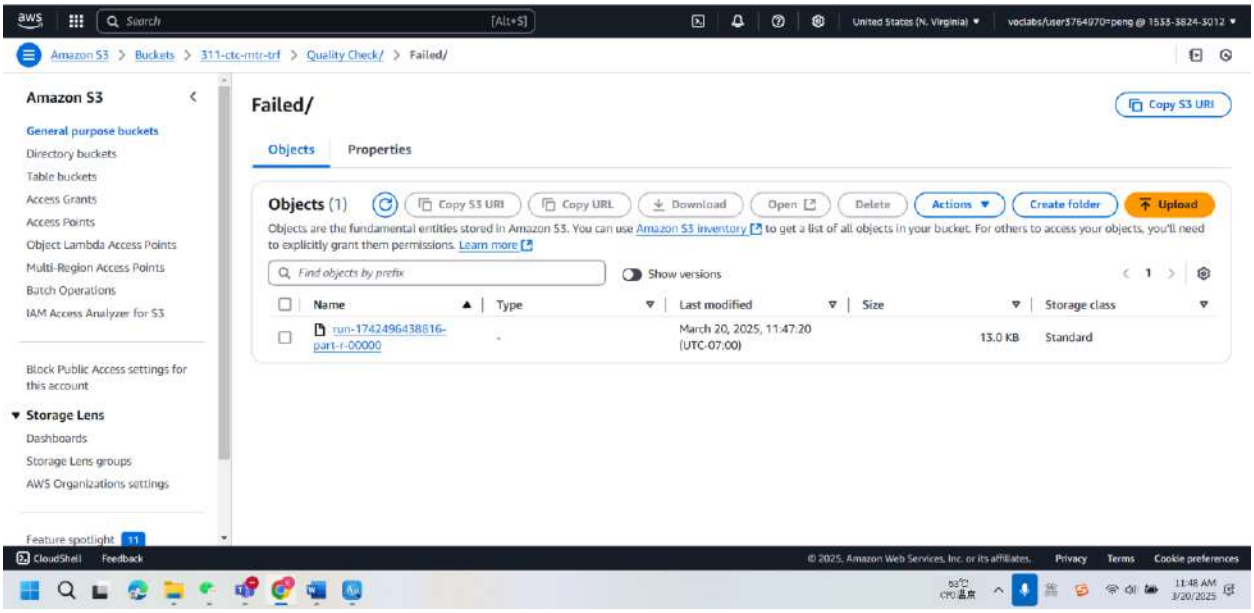


Note. The screenshot shows the visual ETL for quality check. Source from AWS.

Explanation: I did the ruleset for data quality and then set the conditional router, including passed and failed, and then changed the schema before load, including dropping the useless columns. Finally, I loaded the Passed and Failed into the corresponding folders. This part mainly focuses on passed data and failed data based on specific rules.

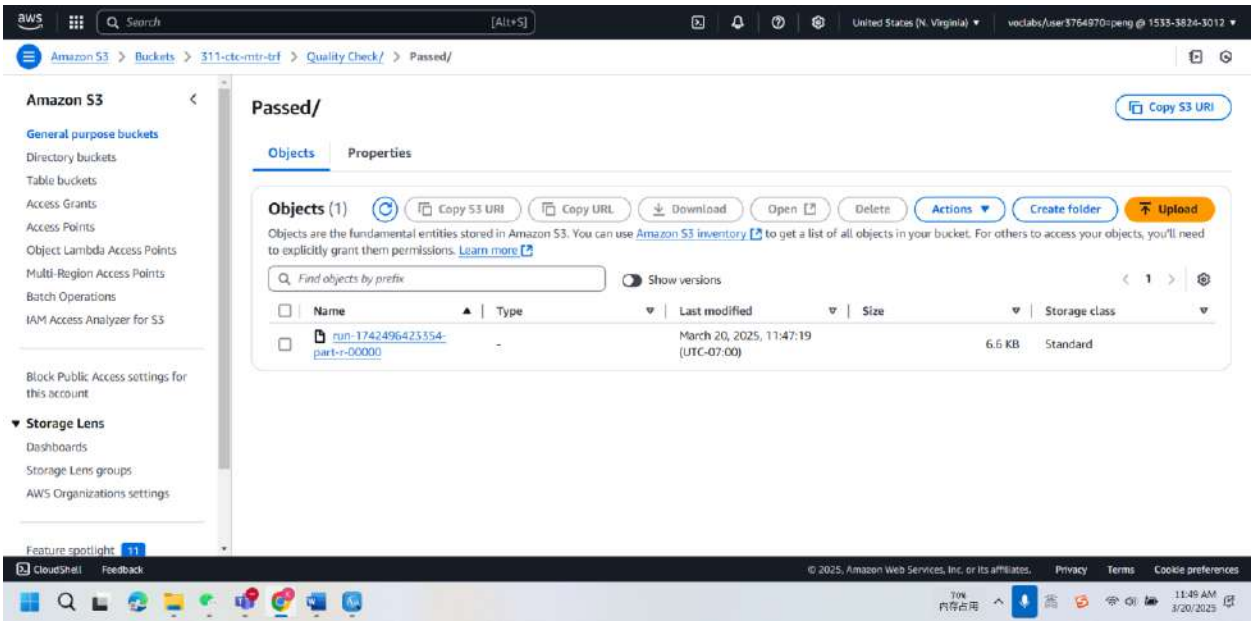
The whole operation is mainly to organize the data better. For example, the data in the Passed subfolder is cleaned and validated, which can be processed successfully. However, the data in the Failed subfolder is incorrect and incomplete, which cannot be processed better. In this case, better data quality control can be achieved. Moreover, visual ETL is beneficial in reducing errors and speeding up data transformation.

Figure 28: Failed Dataset in Transfer Bucket



Note. The screenshot shows the failed dataset in the transfer bucket. Source from AWS.

Figure 29: Passed Dataset in Transfer Bucket

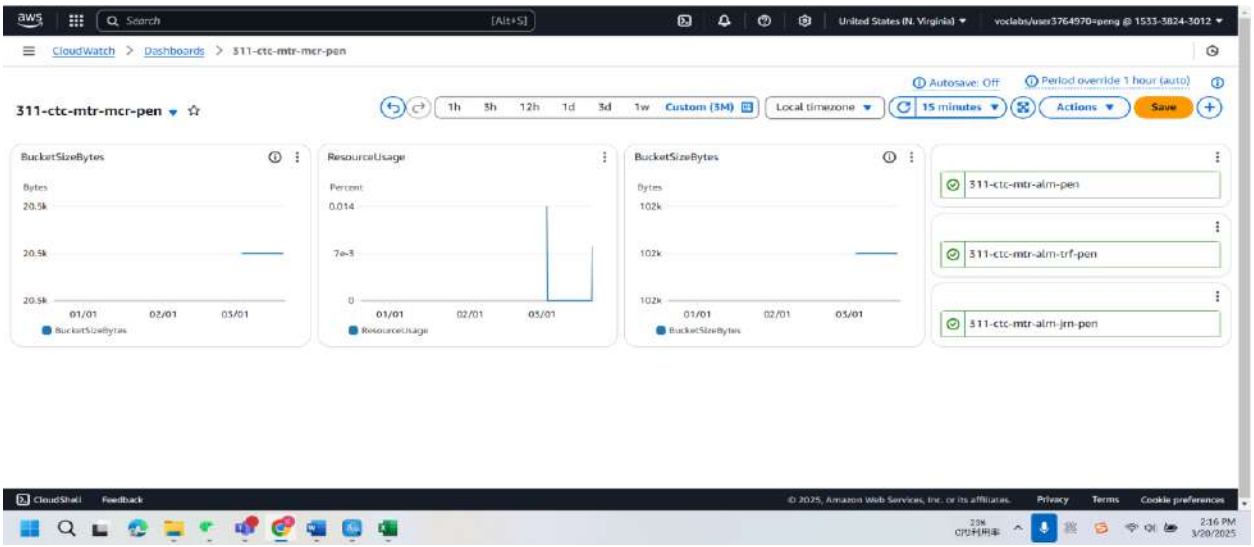


Note. The screenshot shows the passed dataset in the transfer bucket. Source from AWS.

Explanation: After loading the passed and failed in Visual ETL, the corresponding datasets have already been published into each sub-bucket in the transfer bucket, which can be downloaded for different uses.

Step 8: Data Monitoring

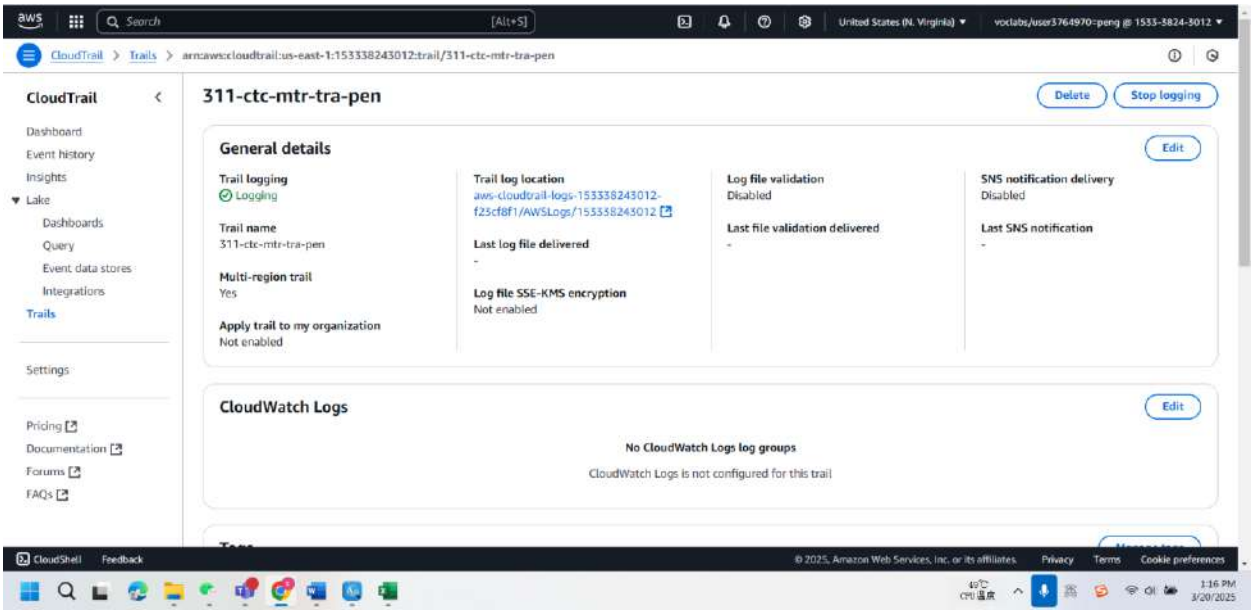
Figure 30: Dashboard and Alarms in CloudWatch



Note. The screenshot shows the dashboard and alarms in CloudWatch. Source from AWS.

Explanation: I established the dashboard in CloudWatch, including the raw bucket and the transfer bucket in S3, and the JobRun in Glue. Finally, I also established three alarms for the raw bucket, transfer bucket, and the jobrun, which were added to the dashboard. Regarding the alarms, it is necessary to set the threshold for each alarm, and the threshold is based on the dataset. For example, I set the threshold as 40000 for the bucket size bytes and 1 day to monitor the data. That is because this threshold is suitable for this bucket, which is originally upper. The period of 1-day monitoring data is mainly to reduce the cost. Finally, the dashboard can visualize the comprehensive view of key metrics, which can achieve the goal of monitoring the data, and alarms are mainly related to thresholds, which can monitor the issue before impacting users.

Figure 31: Trail in CloudTrail



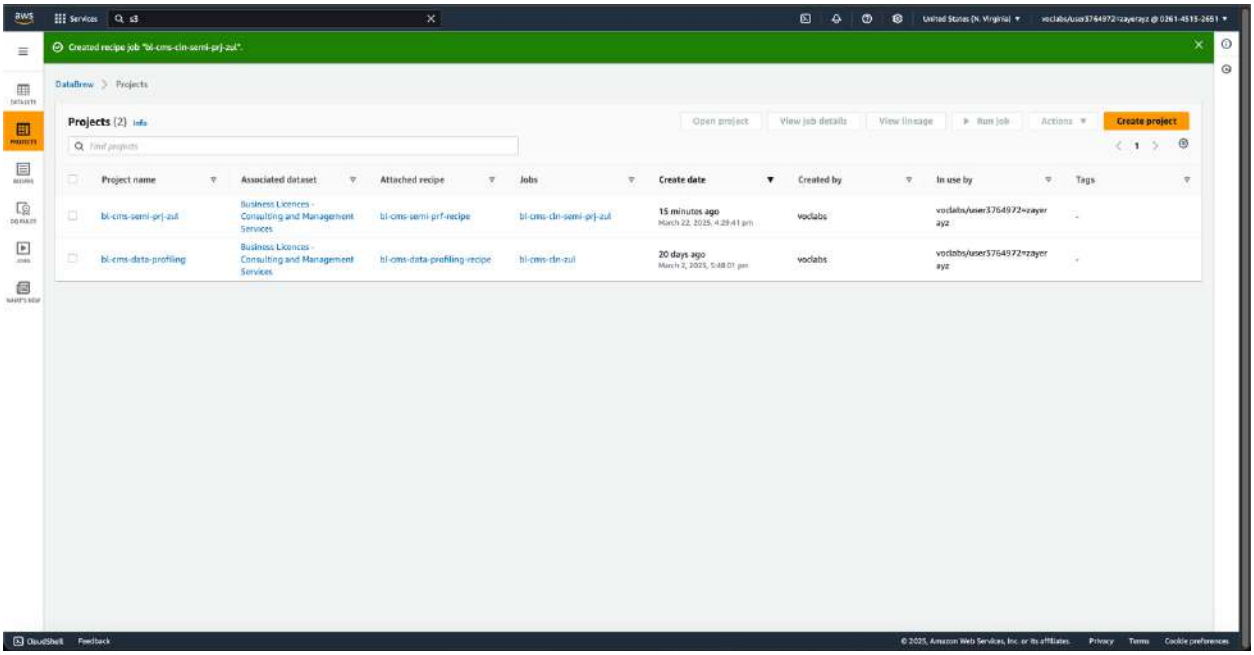
Note. The screenshot shows the trail in CloudTrail. Source from AWS.

Explanation: I established the trail in CloudTrail. That is because I think this operation is essential to do the data monitoring by detecting threats and unauthorized access.

DAP Implementation (Individual work - Muhammad Zulqarnain Shahzad)

Step 5: Data Analysis

Figure 32: Data Brew Projects – Semi Dataset Setup



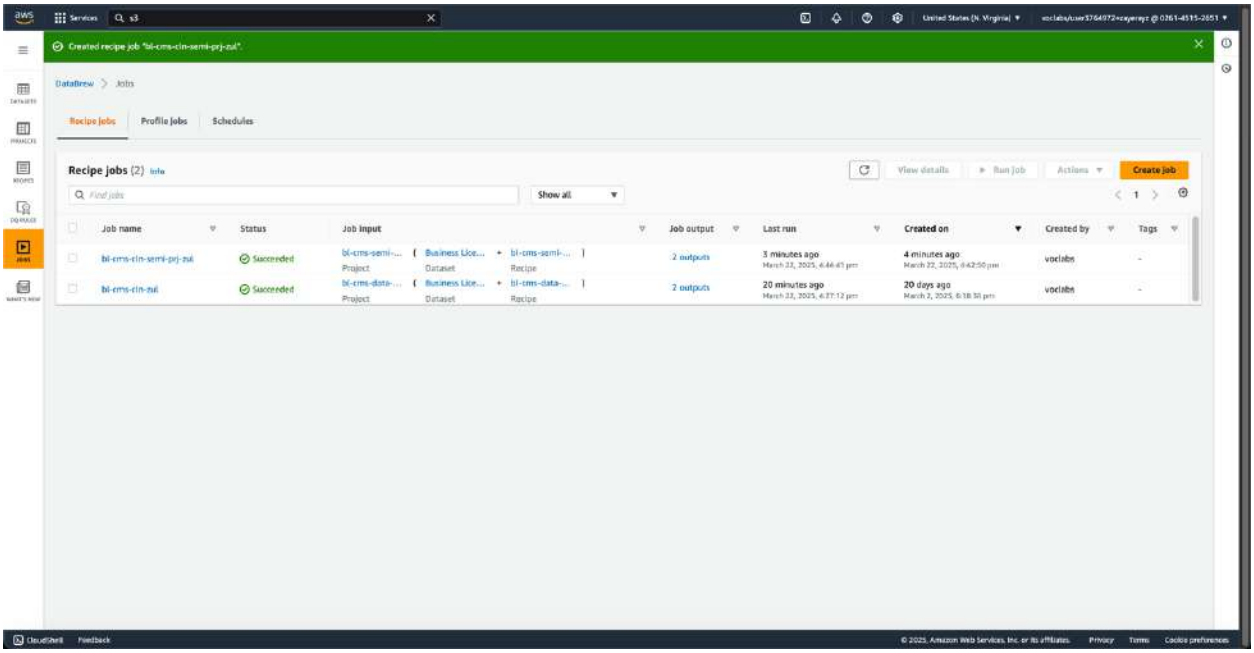
Note. This screenshot shows that I created a new Glue DataBrew project named bl-cms-semi-prj-zul to work on my semi dataset based on the raw version of Business Licences – Consulting and Management Services. Source: AWS Glue DataBrew, *self-work*

Explanation

I just copy my old project from part 1 and made new one for part 2 to show I can handle semi dataset too. It's same data but just to make it look structured like everyone. Professor said

we don’t need new data, but structure is important. So, I use same dataset but made new project with new name.

Figure 33: Data Brew Recipe Jobs – Semi Cleaning Job Execution

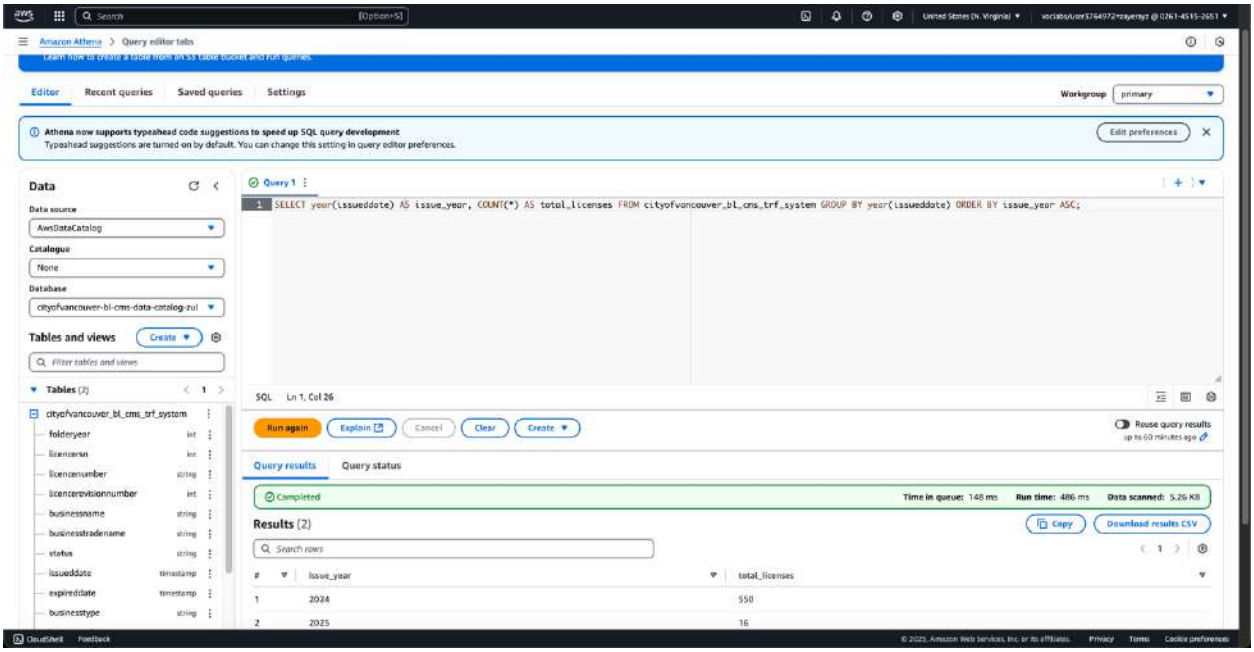


Note. This screenshot shows the job bl-cms-cln-semi-prj-zul was created and successfully ran to process the semi dataset and store the output in S3 semi/user and semi/system. Source: AWS Glue DataBrew, *self-work*

Explanation

This is the job I made from the semi project to do a basic run on the dataset. I didn’t need to do much cleaning again because it was already clean from before. But I still ran the job just to complete the step and make sure I can track how the data flows from raw to semi. It saves the output into the right folders in S3 and shows that I understand how to structure things with different stages like raw, semi, and transform. So, this is more about keeping a clear data process, even if the content didn’t change much.

Figure 34: Business Licenses Issued by Year – Athena SQL Query

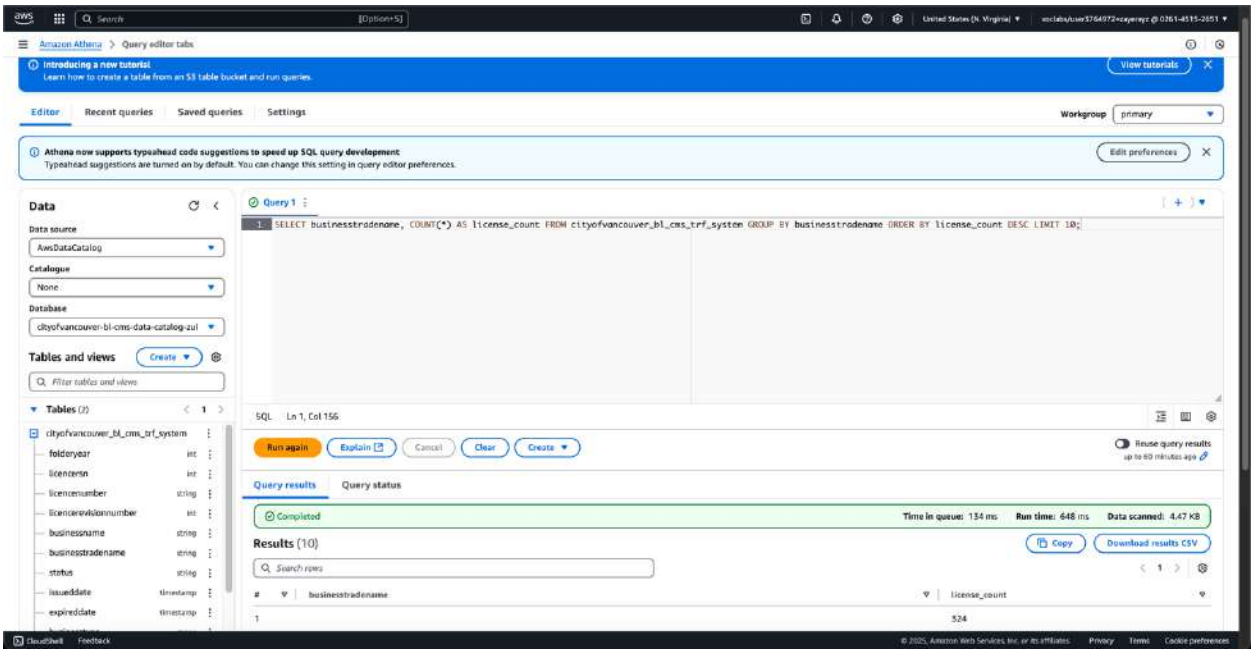


Note. This query shows the number of consulting licenses issued each year using the issued date field. The result shows 550 licenses in 2024. Source: Self-work, AWS Athena Query Editor

Explanation

I wanted to see like how many licenses are given each year, so I ran this query. It's just counting how many new consulting businesses got licenses by year. 2024 is big, 2025 is just started. This helps to see the trend and business activity, if it's going up or down.

Figure 35: Top 10 Business Trade Names by License Count – Athena SQL Query

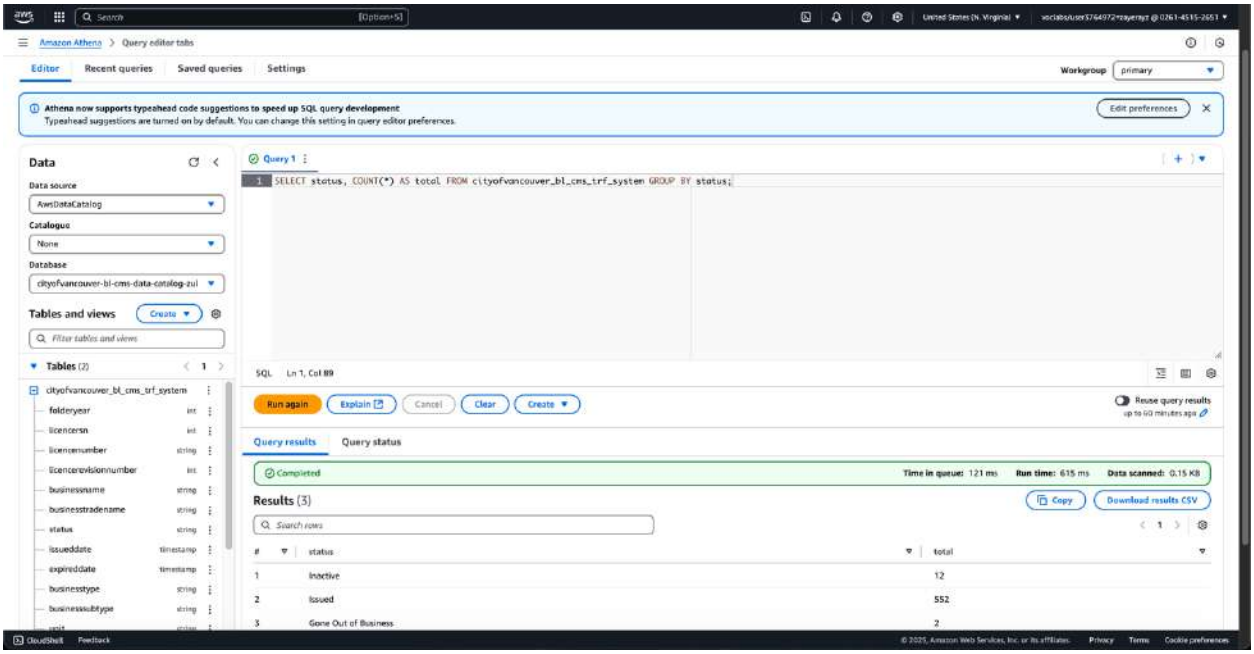


Note. This query returns the most frequent business trade names from the consulting dataset, ordered by license count in descending order. Source: Self-work, AWS Athena Query Editor

Explanation

This one show which business names come most often in the consulting list. I used COUNT to count how many times each name comes. Then I sorted from biggest to smallest. The top one has 324, that's a lot. It helps to know who is most active or popular in this business.

Figure 36: License Status Breakdown – Active vs Inactive



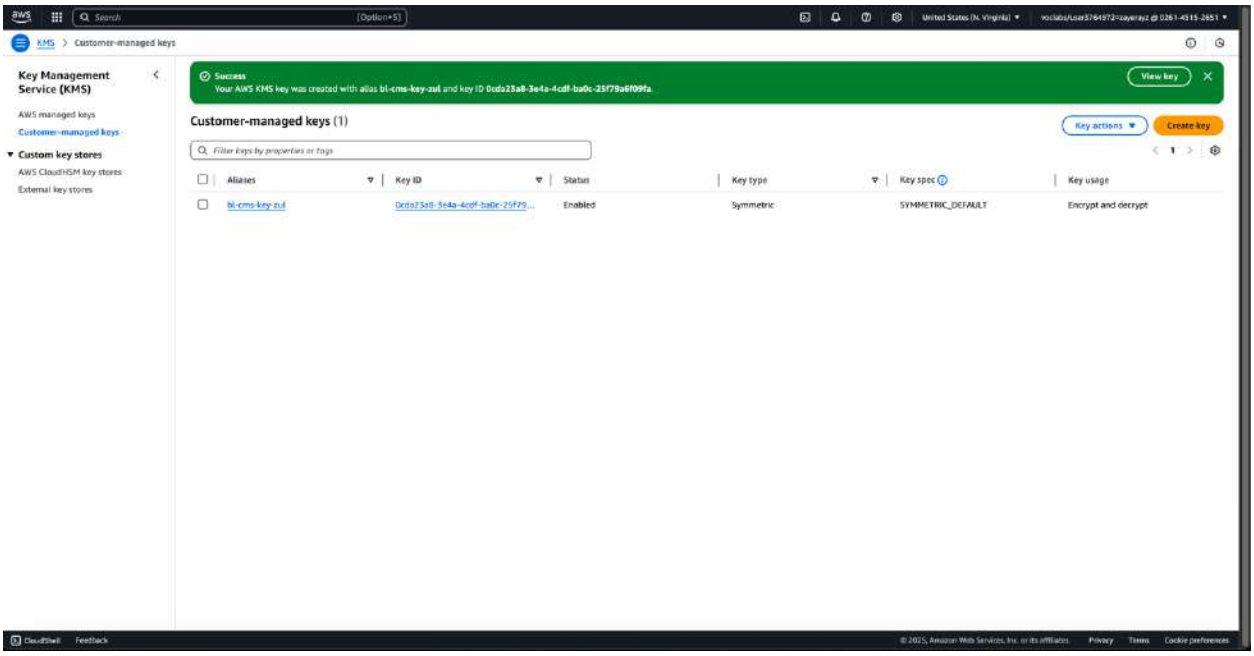
Note. This query shows the distribution of license statuses, including issued, inactive, and gone out of business. Source: Self-work, AWS Athena Query Editor

Explanation

Here I checked how many licenses are active and how many are not. So, I just grouped by status and counted. Most of them are issued (like active), but some are inactive or gone out of business. It's useful to know which ones are still running and which stopped.

Step 6: Data Security

Figure 37: AWS KMS Key Creation – bl-cms-key-zul

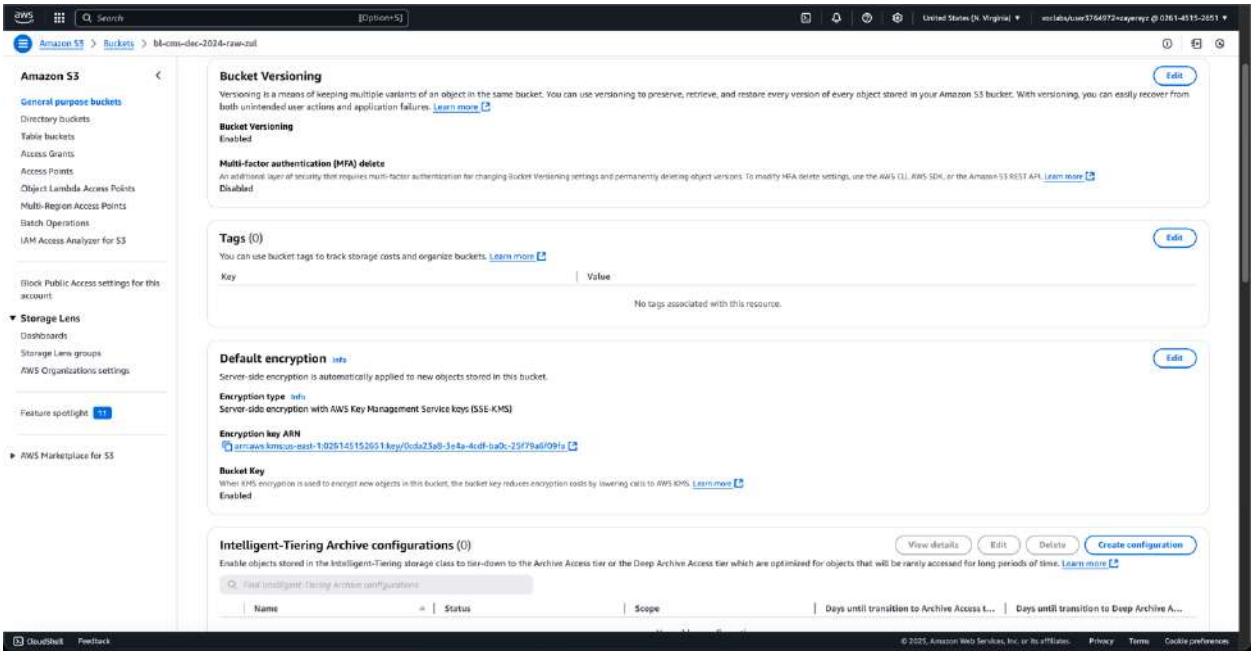


Note. This screenshot shows the KMS key I created in AWS for encrypting S3 data. The key is symmetric and will be used for both encryption and decryption of business license files. Source: Self-work, AWS Key Management Service

Explanation

This is my own KMS key. I make this to lock my files in S3 buckets. I named it bl-cms-key-zul to match my consulting dataset project. It's symmetric so I can use it for both encrypt and decrypt. I will use this in S3 buckets in next steps to make the data secure.

Figure 38: S3 Bucket Encryption and Versioning – Raw Zone

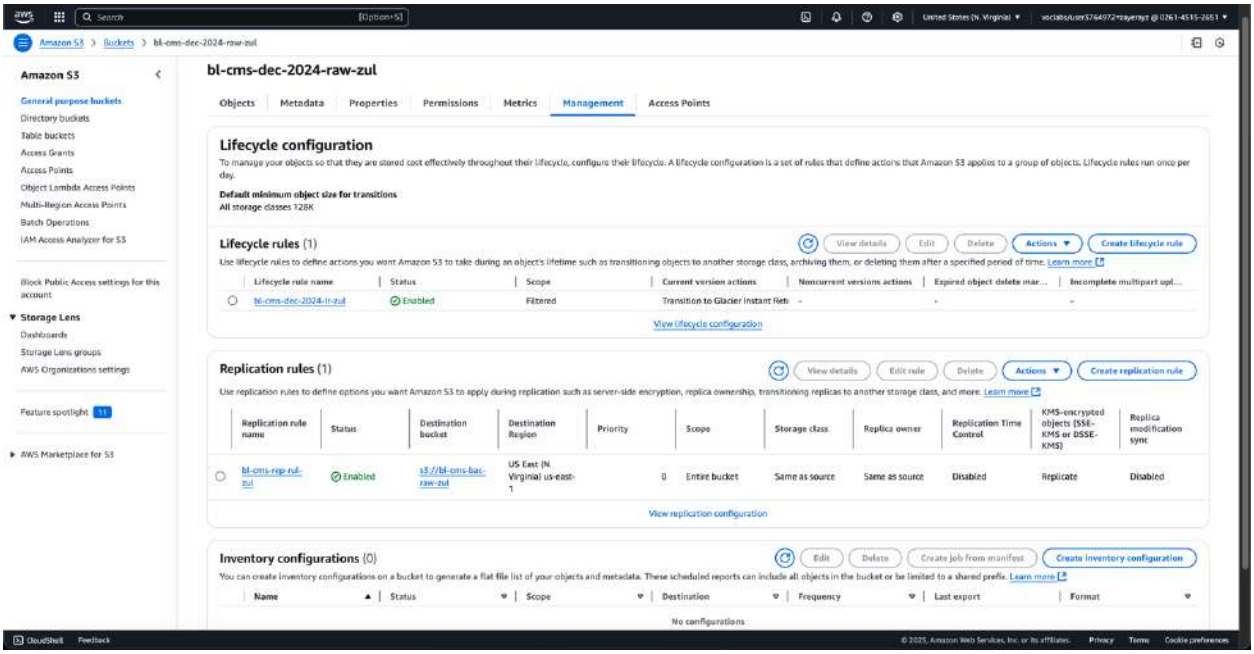


Note. This screenshot shows the settings of my raw S3 bucket bl-cms-dec-2024-raw-zul. I enabled versioning and set default encryption using the KMS key I created earlier. Source: Self-work, AWS S3 Bucket Settings

Explanation

Here I turn on versioning and encryption for my raw bucket. Versioning is for saving old copies if something changes or delete. Encryption is to protect my raw business license files. I select my KMS key to lock them. Now every file I upload here will be safe and can't be read by others.

Figure 39: S3 Replication and Lifecycle Rules for Raw Bucket



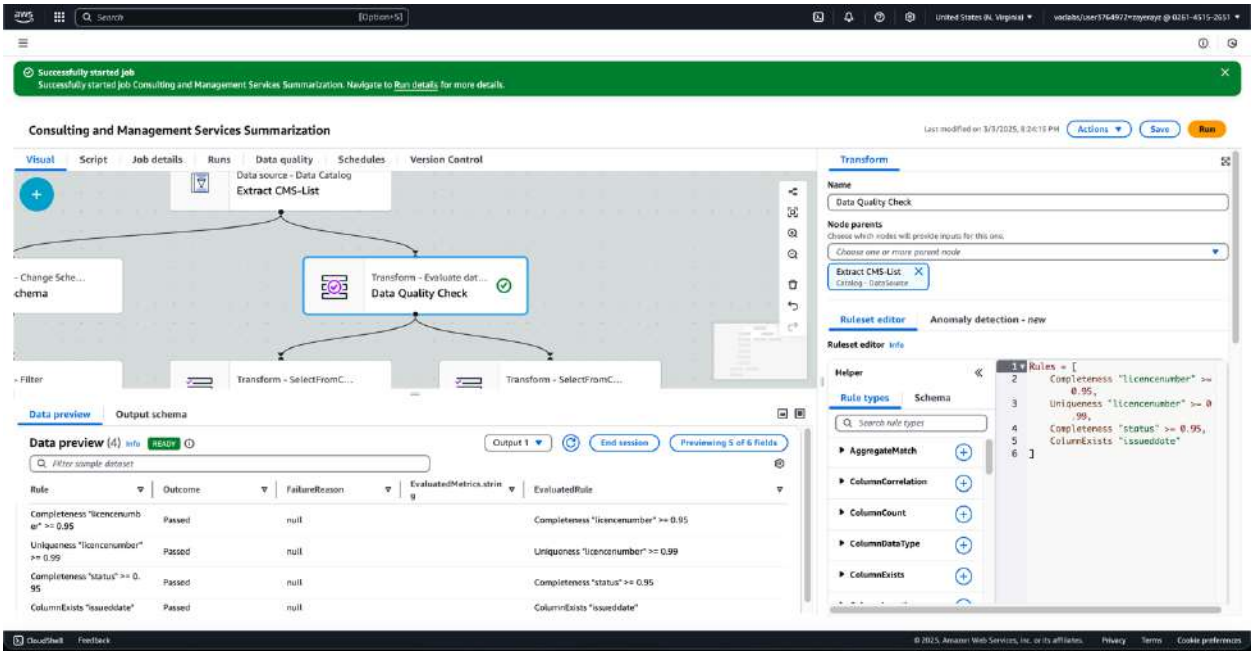
Note. This screenshot shows the replication and lifecycle settings applied to my raw zone bucket. The replication rule sends data to a backup bucket using the same KMS key. The lifecycle rule is used to manage long-term storage. Source: Self-work, AWS S3 Management Tab

Explanation

In this I setup replication from my raw bucket to another backup one. So, my files don't get lost if main bucket got error. I also made lifecycle rule to move old files to glacier storage after some time. It helps to save cost and make system more professional and safer. Replication is using same KMS key I made before.

Step 7: Data Governance

Figure 40: Data Quality Rules, Execution

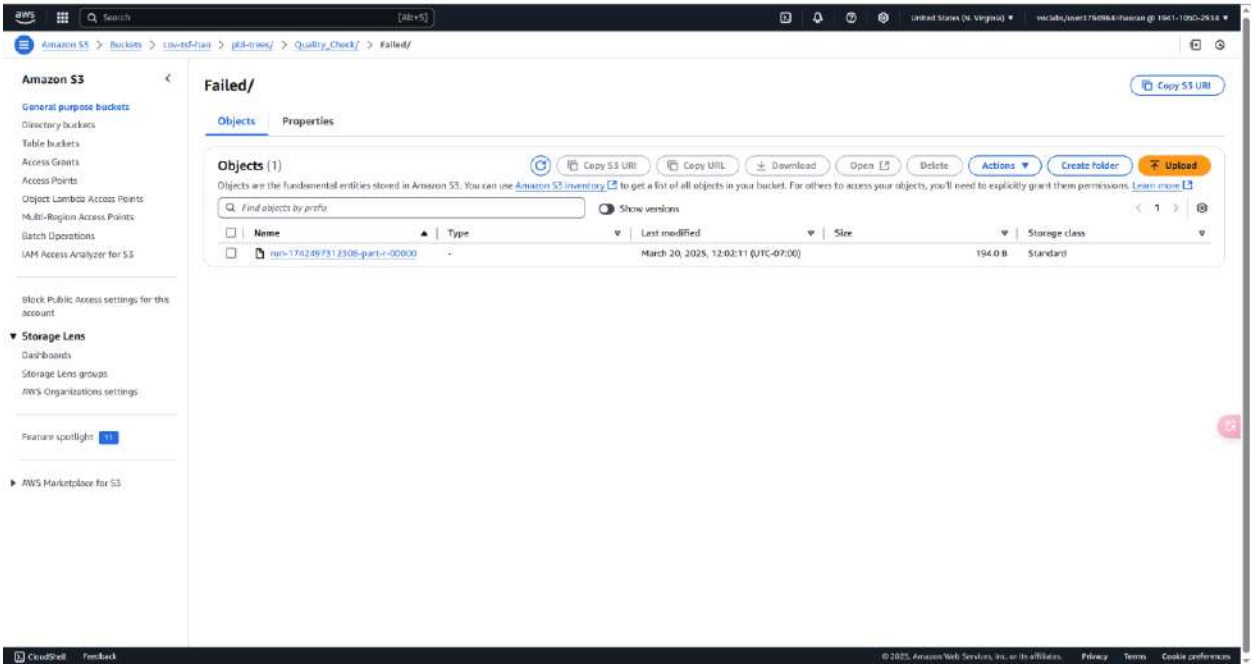


Note. This screenshot shows the main data quality rules I created and executed in AWS Glue Studio. It includes completeness check on “licencenumber” and “status” columns (≥ 95%), uniqueness for “licencenumber” (≥ 99%), and existence check for the “issueddate” column. Source: Self-work, AWS Glue Studio – Data Quality Rule Editor

Explanation

This was the part where I added my data quality check block to the ETL pipeline. I tried to make sure important fields are filled and unique, and check if the column even there. All passed and result came green. I was able to do this part fully, and now data looks ready for next steps.

Figure 41: S3 Output Folder – Failed Results Preview



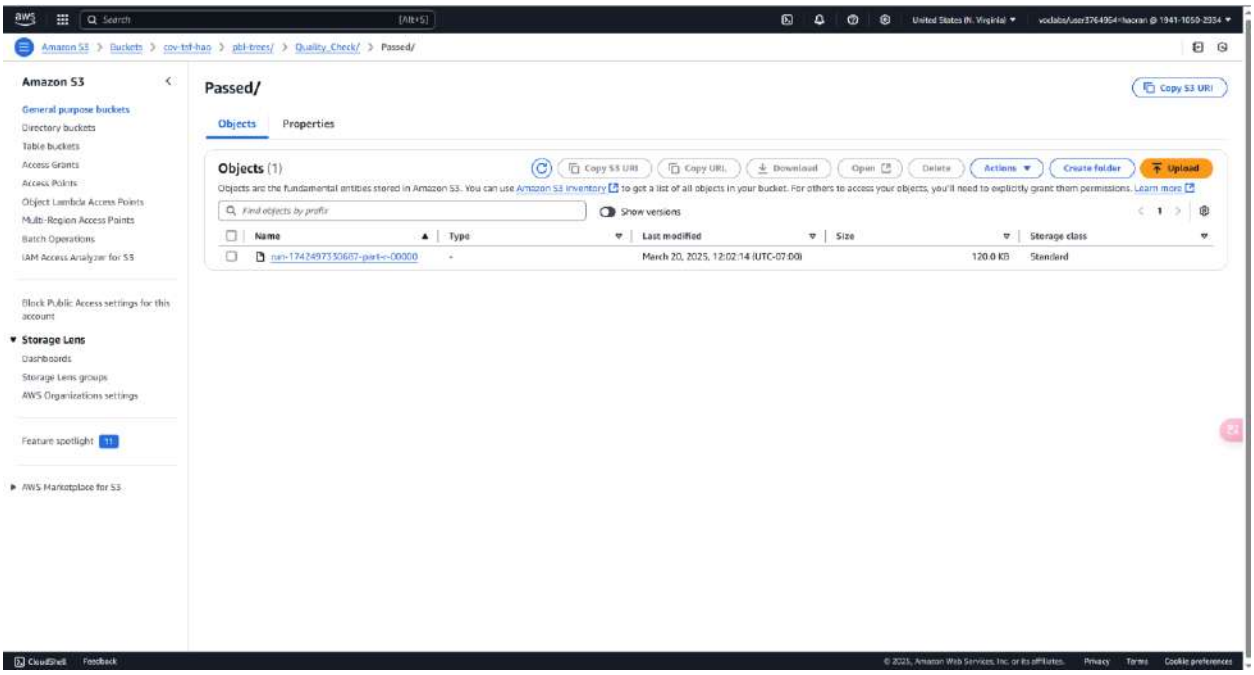
Note. This screenshot shows the Failed folder inside the S3 bucket where failed data quality records are stored. This part was done to separate any data that didn't meet the defined rules.

Source: Peer provided screenshot, AWS S3 – Quality Check Output

Explanation

This part is important to show failed and passed results but honestly, I was struggling here. I couldn't find the right video or remember the exact steps to separate output in passed and failed like this. I understand the logic and how it works, but I couldn't finish it myself so I'm just attaching to show what it looks like.

Figure 42: S3 Output Folder – Passed Results Preview



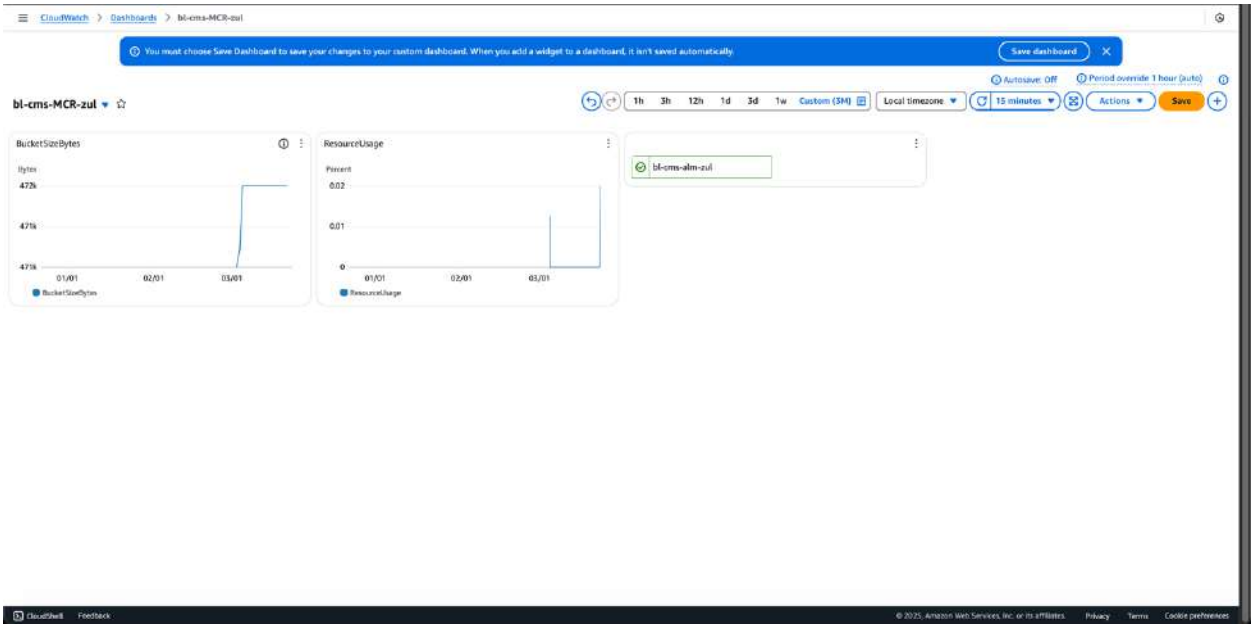
Note. This screenshot shows the Passed folder where records that met all data quality rules were saved. It's part of the validation output done automatically when job is set properly. *Source:* Peer provided screenshot, AWS S3 – Quality Check Output

Explanation

Again, same as before. Just couldn't locate full guide or setup during work session. I understand what's happening in this screenshot and I'll fix mine before final submission. Just showing this to explain I know where it should go.

Step 8: Data Monitoring

Figure 43: Custom CloudWatch Dashboard for Monitoring

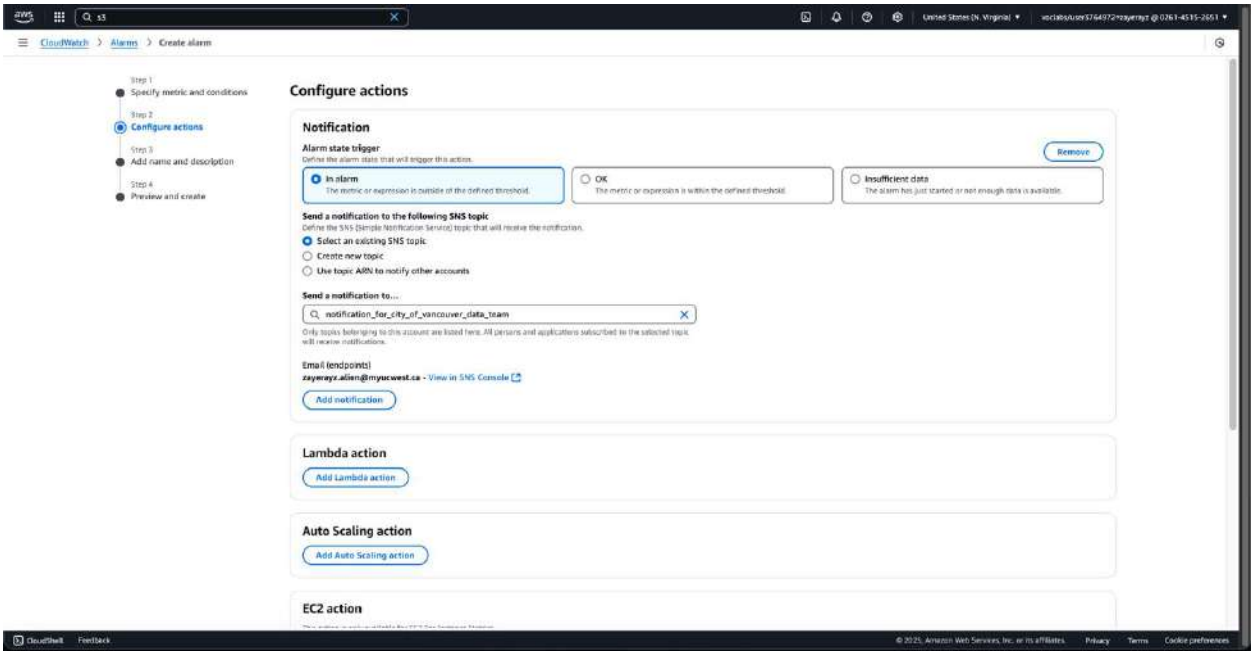


Note. This screenshot shows the custom dashboard bl-cms-MCR-zul with two widgets: one tracking bucket size and another for overall resource usage. *Source:* Self-work, AWS CloudWatch, Dashboards

Explanation

I added dashboard to keep visual on stuff like how much storage I am using and what's happening. I saw bucket size go up in March, which is when I started running my job. It helps to check fast if something crazy happen without opening full AWS pages.

Figure 44: Notification Setup for CloudWatch Alarm

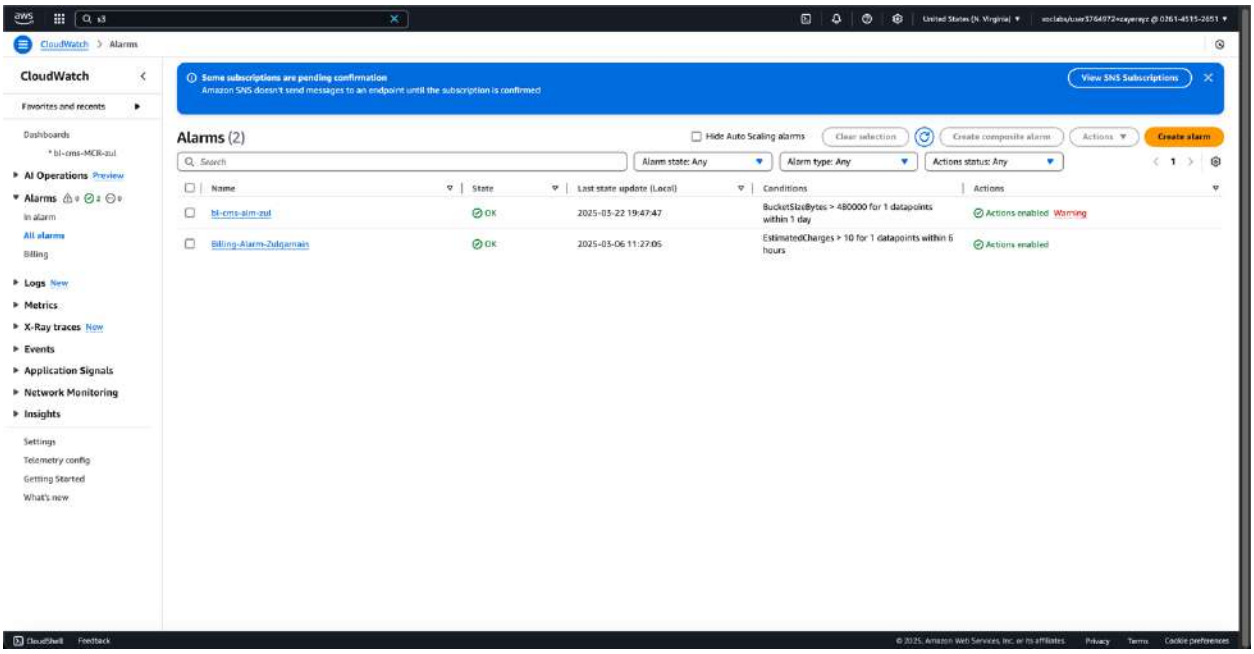


Note. This screenshot shows the setup during alarm creation, with email endpoint zayerayz.alien@myucwest.ca added for receiving bucket alert notifications. *Source:* Self-work, AWS CloudWatch, Alarms, Configure Actions

Explanation

I tried to make email go to me if bucket go above limit. I use topic name like "notification_for_city_of_vancouver_data_team". I click add notification and put my UCW email. Still waiting on confirm I think, but idea is to know if something weird happen.

Figure 45: CloudWatch Alarm Setup – Billing and Storage Thresholds

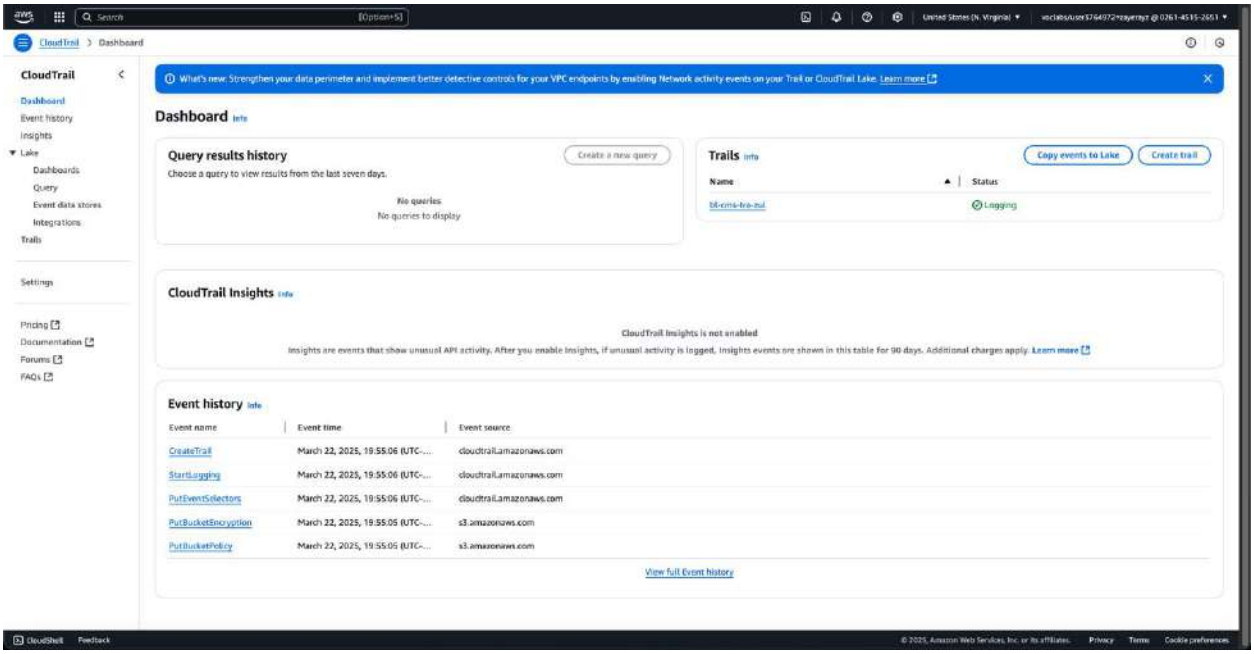


Note. This screenshot shows two CloudWatch alarms set up for my project: one for S3 bucket storage usage (bl-cms-alm-zul) and one for estimated billing cost (Billing-Alarm-Zulqarnain). Both are in OK state and active. *Source:* Self-work, AWS CloudWatch, All Alarms

Explanation:

Here I set one new alarm just to be safe and monitor stuff. First one is for storage, I set it to alert me if my S3 bucket goes above 480,000 bytes in one day. I use that to keep an eye on my storage cost and make sure nothing unexpected is filling up the bucket.

Figure 46: CloudTrail Setup with Logging Events



Note. This screenshot shows the bl-cms-tra-zul CloudTrail trail created with logging enabled. Recent logged events include trail creation and S3 policy updates. *Source:* Self-work, AWS CloudTrail, Dashboard

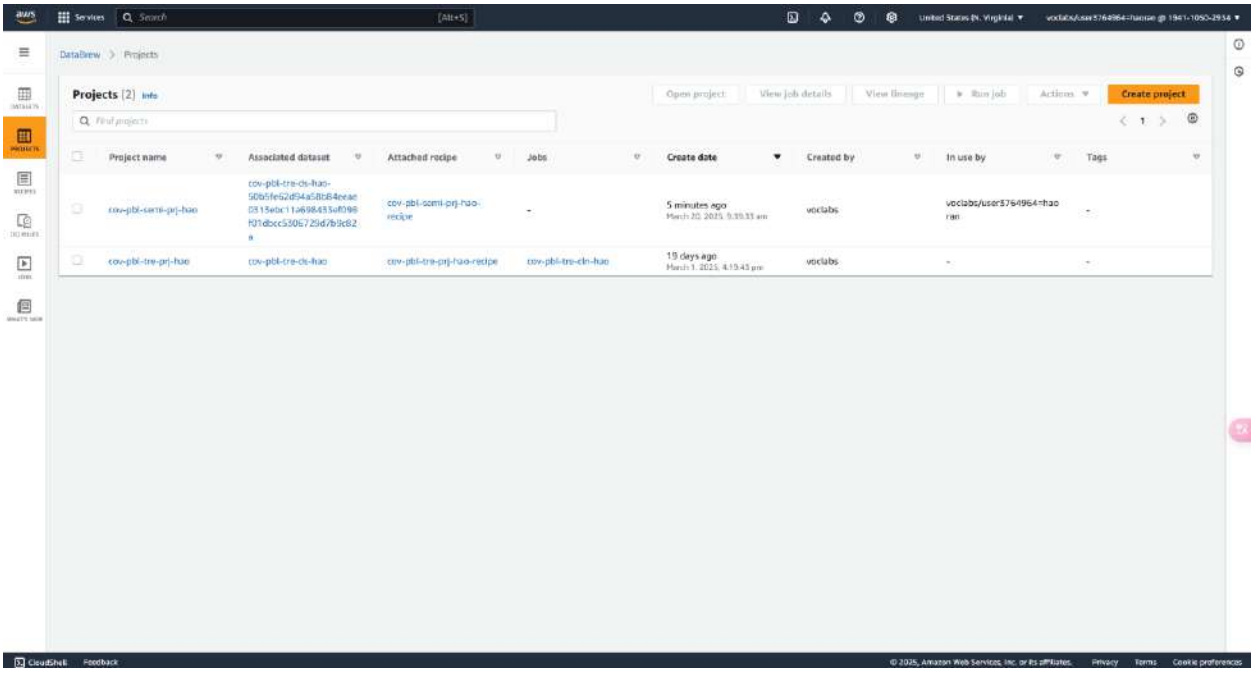
Explanation

Professor, so this is where I set up CloudTrail to track stuff I do in AWS. It shows logs like when I created this trail, turned logging on, added bucket policy, encryption, all that. According to Chat GPT It's not really for querying right now but it's super helpful to see if something weird happen like if someone touch the config or bucket without me knowing. Just keeping track of my own work and for security too. If anything goes wrong, this history helps us to trace back what we did or what someone else did.

DAP Implementation (Individual work – Haoran Xu)

Step 5: Data Analysis

Figure 47: Project Screenshot

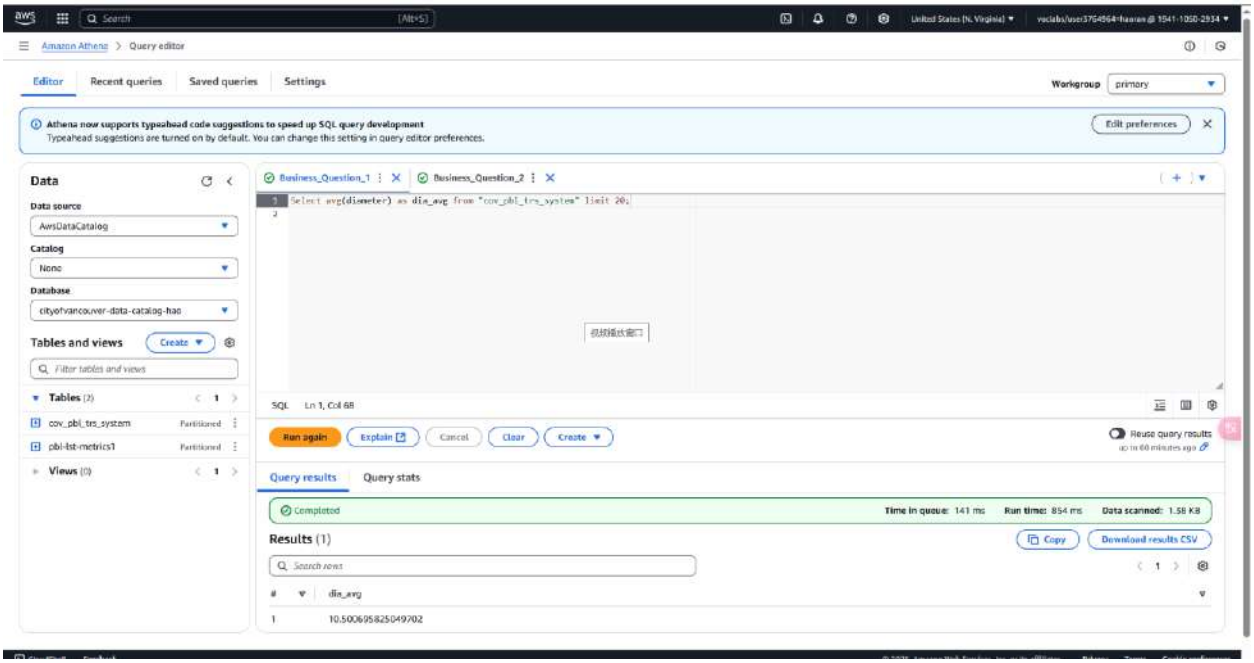


Note. This figure shows the screenshot of the project in AWS Glue Data Brew. Source: AWS Glue

I created the project to check if my dataset has a semi-structured issue. But I found that my dataset is cleaned and can be directly used for analysis.

Now it's time to present the business questions in Athena using SQL.

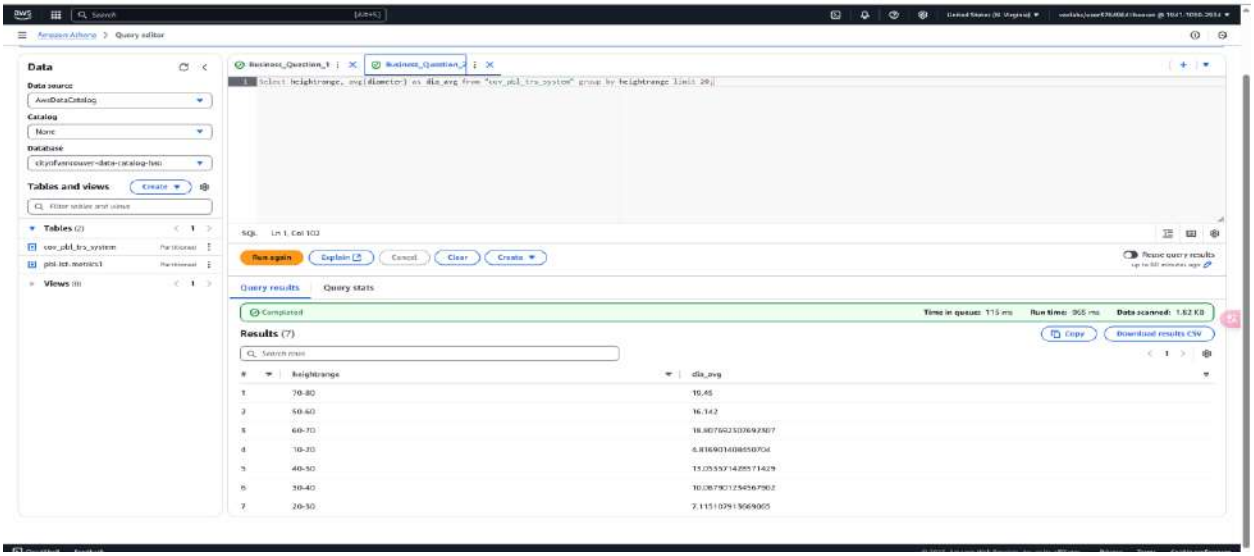
Figure 48: Athena Business Question 1



Note. This figure shows the screenshot of the business question 1 in Athena. Source: Athena

The first business question is simple, it will ask for calculating the average of diameter for the first 20 rows data. And make the output column named “dia_avg”.

Figure 49: Athena Business Question 2

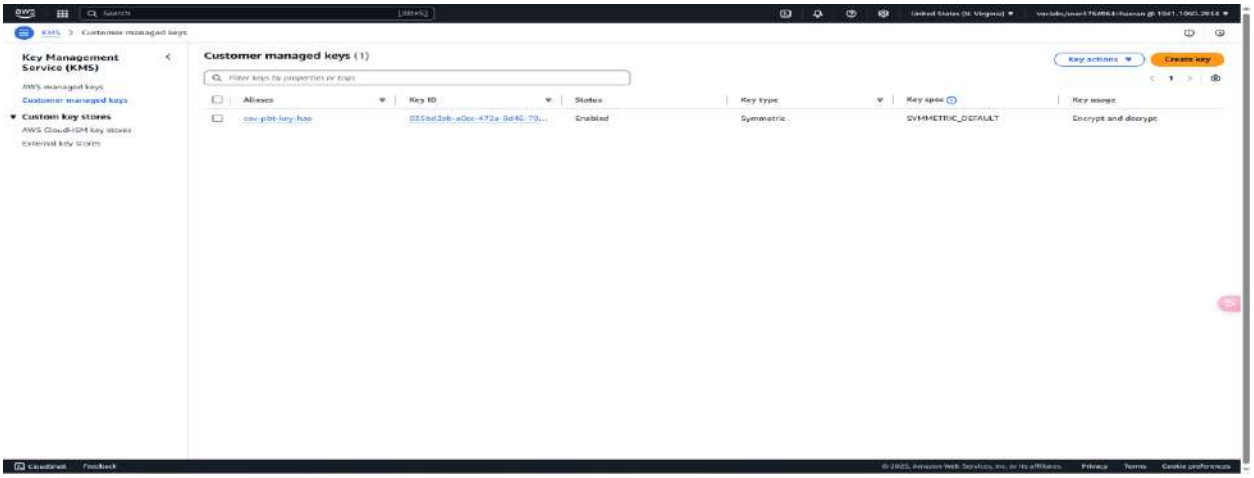


Note. This figure shows the screenshot of the business question 2 in Athena. Source: Athena

The second business question will ask whether the average diameter is directly related to height range under different height_range conditions, and group the output according to height_range. We can conclude from the data that average diameter is indeed correlated with height range and is positively correlated.

Step 6: Data Security

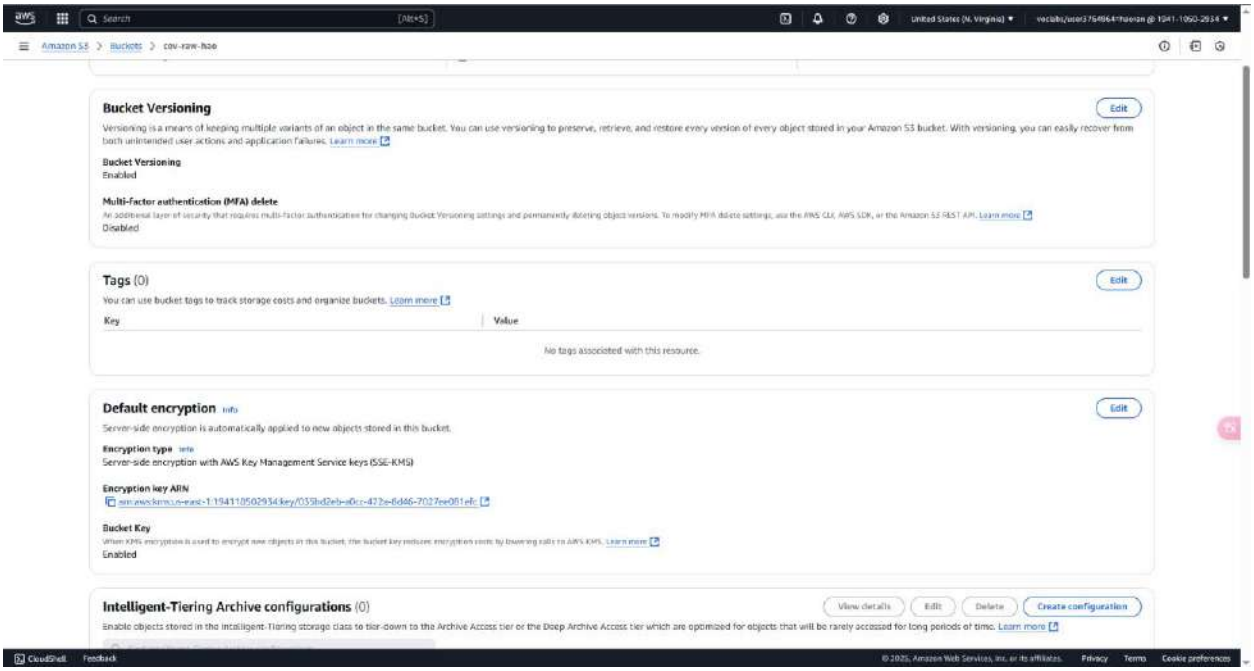
Figure 50: Create KMS Key



Note. This figure shows the screenshot of the key created in AWS KMS. Source: AWS KMS.

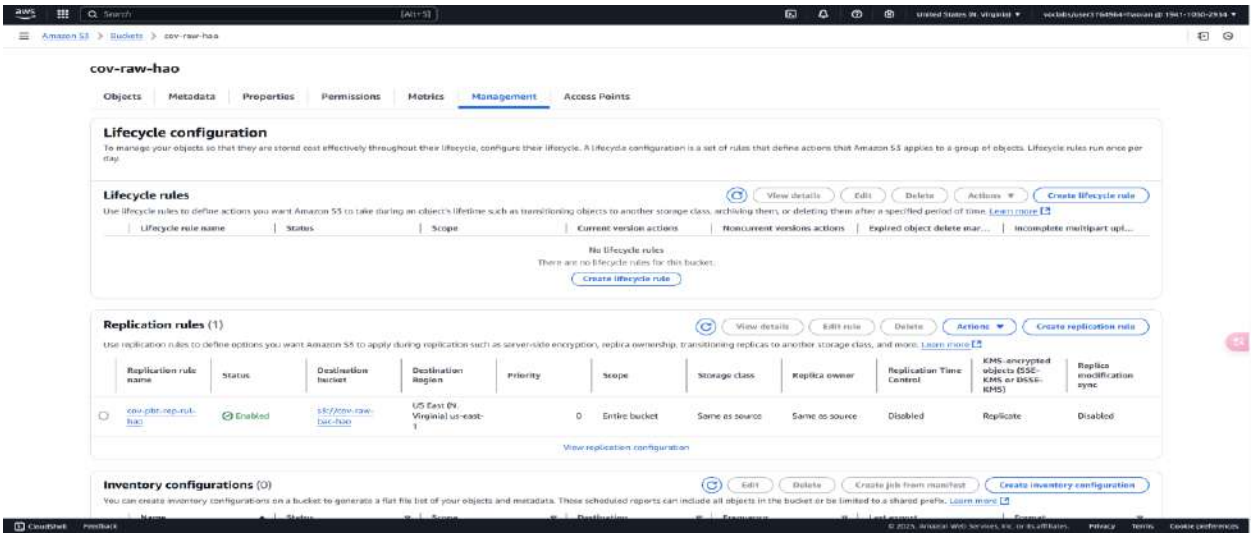
Here, I create a key, it's for sharing with Raw bucket and Transfer Bucket.

Figure 51: Raw-bucket-Versioning and Encryption



Note. This figure shows the screenshot of the versioning and encryption of raw bucket created in S3. Source: AWS S3.

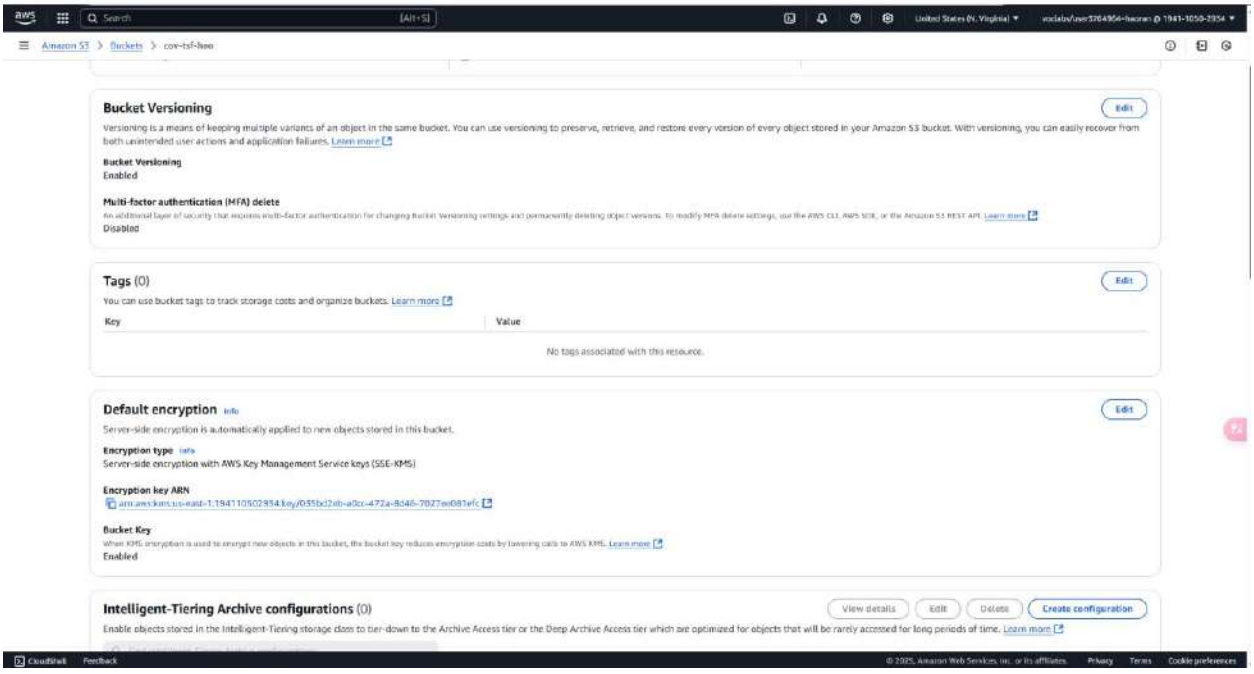
Figure 52: Raw-bucket-Replication Rules



Note. This figure shows the replication rules of raw bucket created in S3. Source: AWS S3.

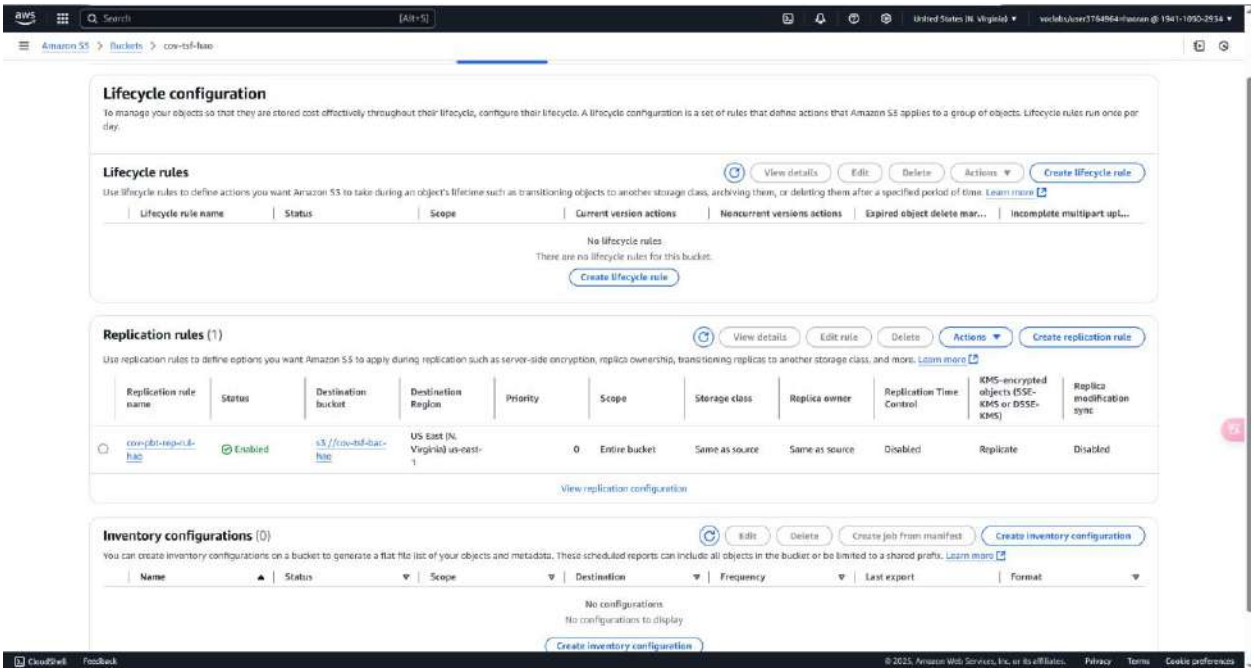
Before this step, I also created the bac-folder for raw bucket in S3. And this is the destination of output.

Figure 53: Transfer-bucket-Versioning and Encryption



Note. This figure shows the screenshot of the versioning and encryption of transfer bucket created in S3. Source: AWS S3.

Figure 54: Transfer-bucket-Replication Rules

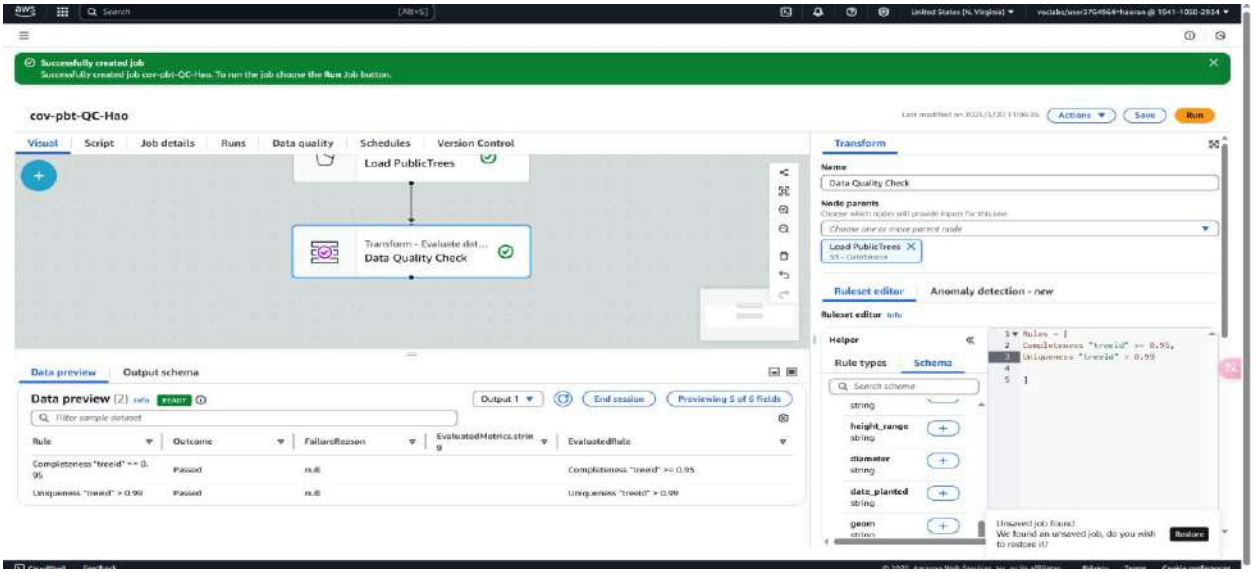


Note. This figure shows the replication rules of transfer bucket created in S3. Source: AWS S3.

Before this step, I also created the bac-folder for transfer bucket in S3. And this is the destination of output.

Step 7: Data Governance

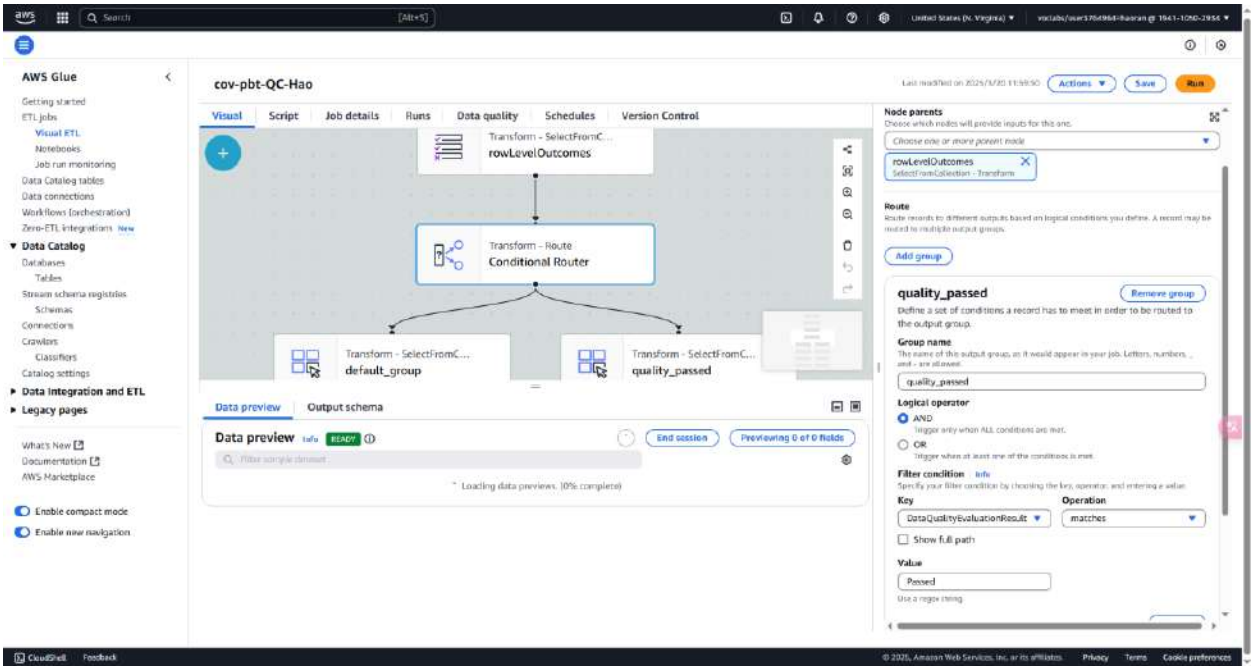
Figure 55: Data Quality Check Rules



Note. This figure shows the screenshot of the data quality check step in visual ETL created in Glue. Source: AWS Glue

In this step, I checked the completeness and uniqueness for “treeid” column. To see if the completeness is equal or greater than 95% and if Uniqueness is greater than 99%.

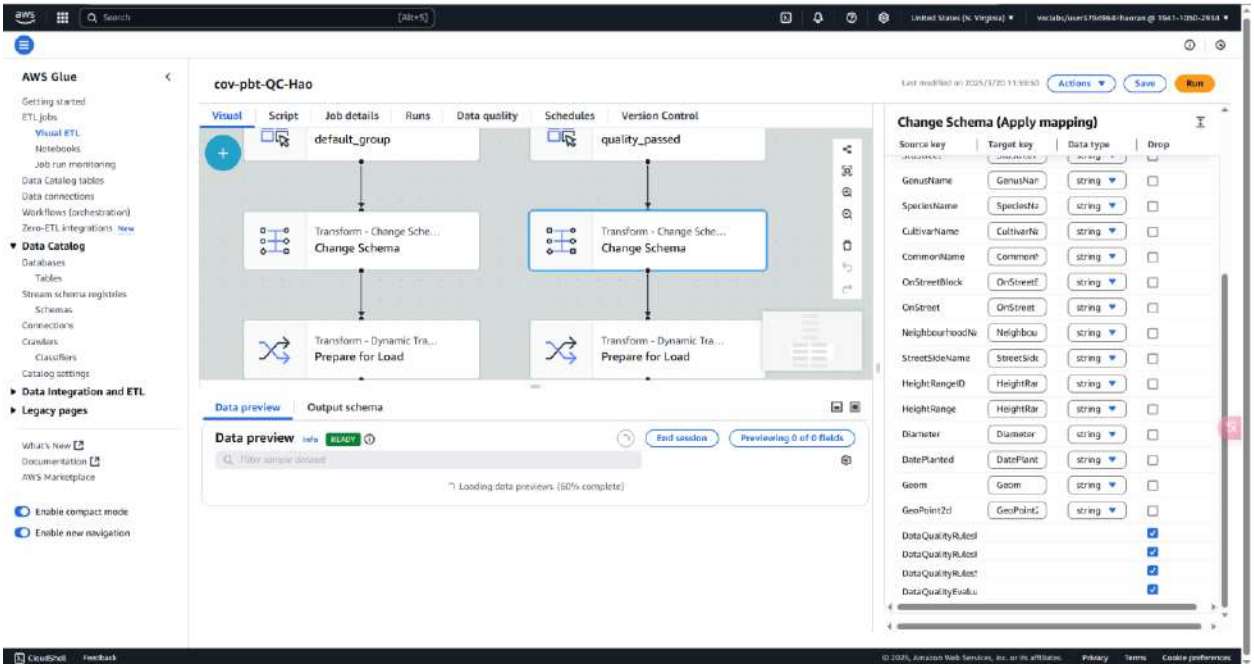
Figure 56: Conditional Router



Note. This figure shows the screenshot of the conditional router step in visual ETL created in Glue. Source: AWS Glue

This step is to separate the quality check output as two files: passed and failed into two different prepared folders.

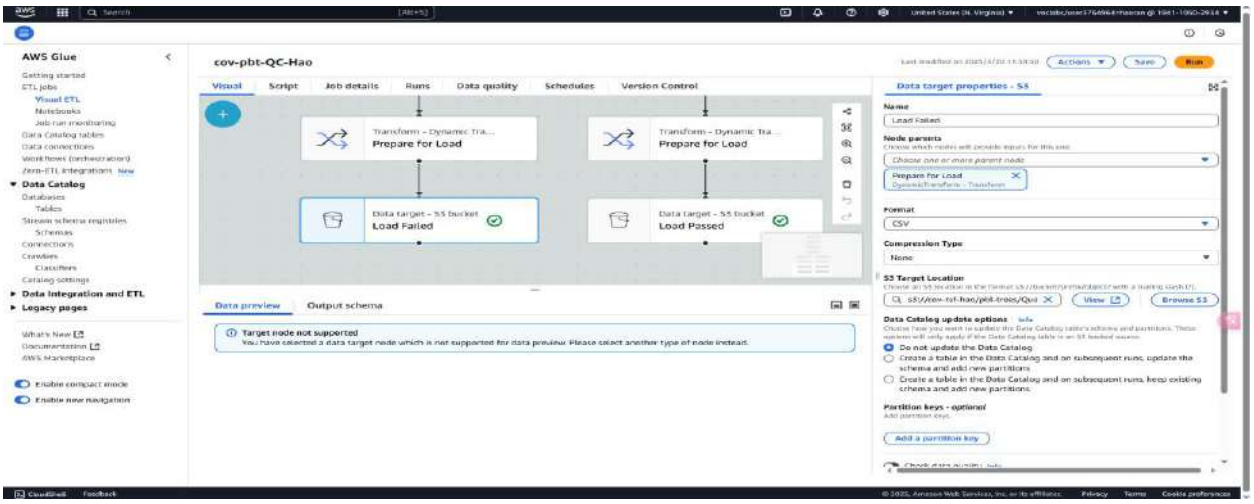
Figure 57: Change Schema Before Load



Note. This figure shows the screenshot of the change schema step in visual ETL created in Glue. Source: AWS Glue

The meaning of this step is to drop the useless columns which are for data quality checking for both passed and failed output files. So it reduces our costs and make the output file looks cleaner.

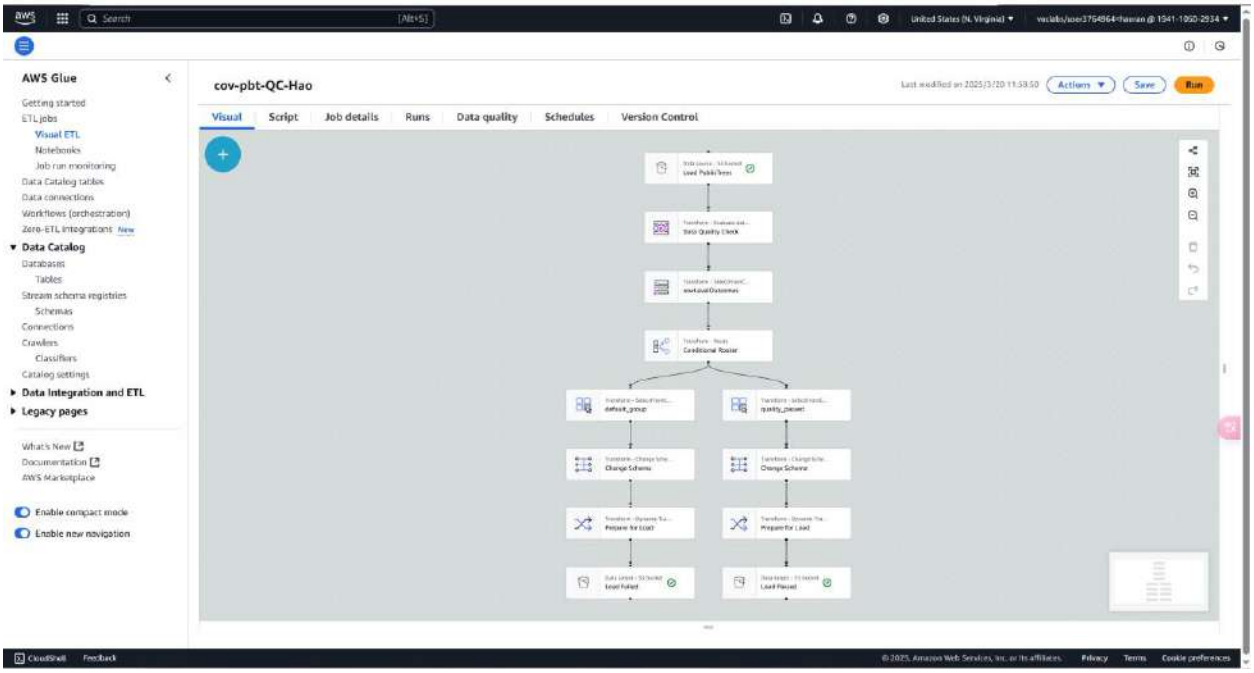
Figure 58: Load Both Passed and Failed into Corresponding Folders



Note. This figure shows the screenshot of the Load Failed/Load Passed step in visual ETL created in Glue. Source: AWS Glue

These two steps are the last step before running the visual ETL. We select the output file format as csv and choose the specified location to output.

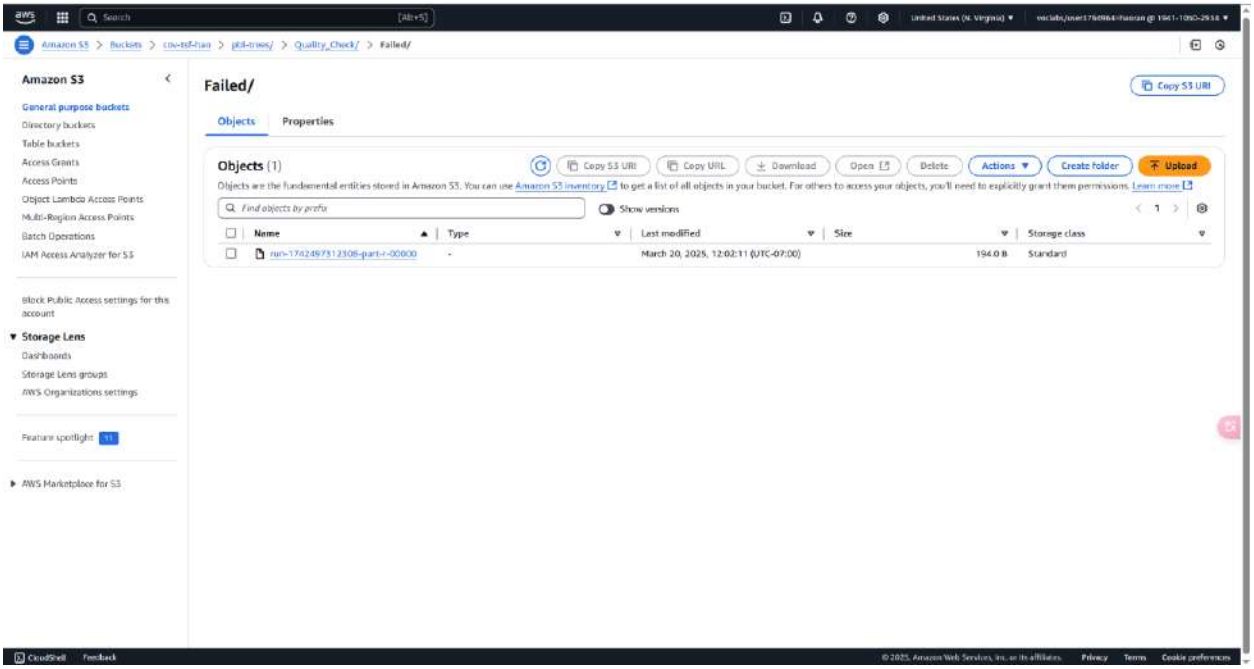
Figure 59: Overall Look of Quality Check Visual ETL



Note. This figure shows the screenshot of the overall look of Quality Check Pipeline in visual ETL created in Glue. Source: AWS Glue

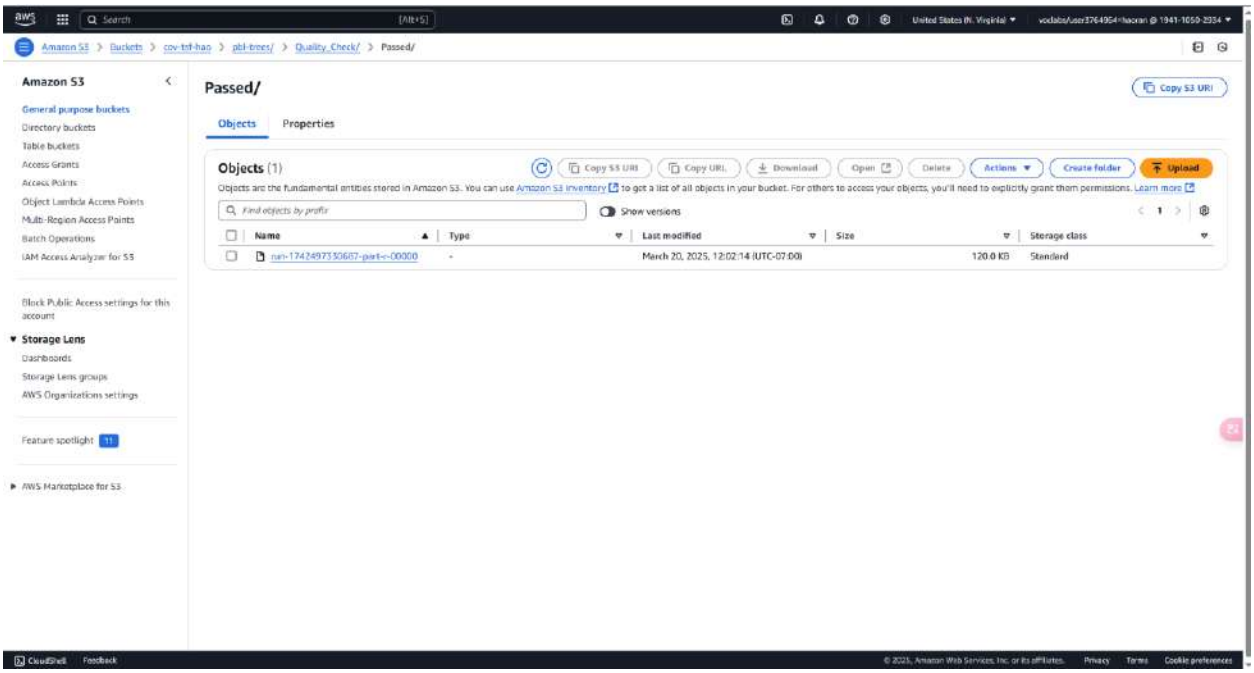
This is the overall look of the Quality Check Visual Pipeline. We can see which part of dataset passed/failed the quality check in the S3 bucket below.

Figure 60: Quality Check Failed Output



Note. This figure shows the screenshot of the Failed output in Transfer Bucket of S3. Source: AWS S3

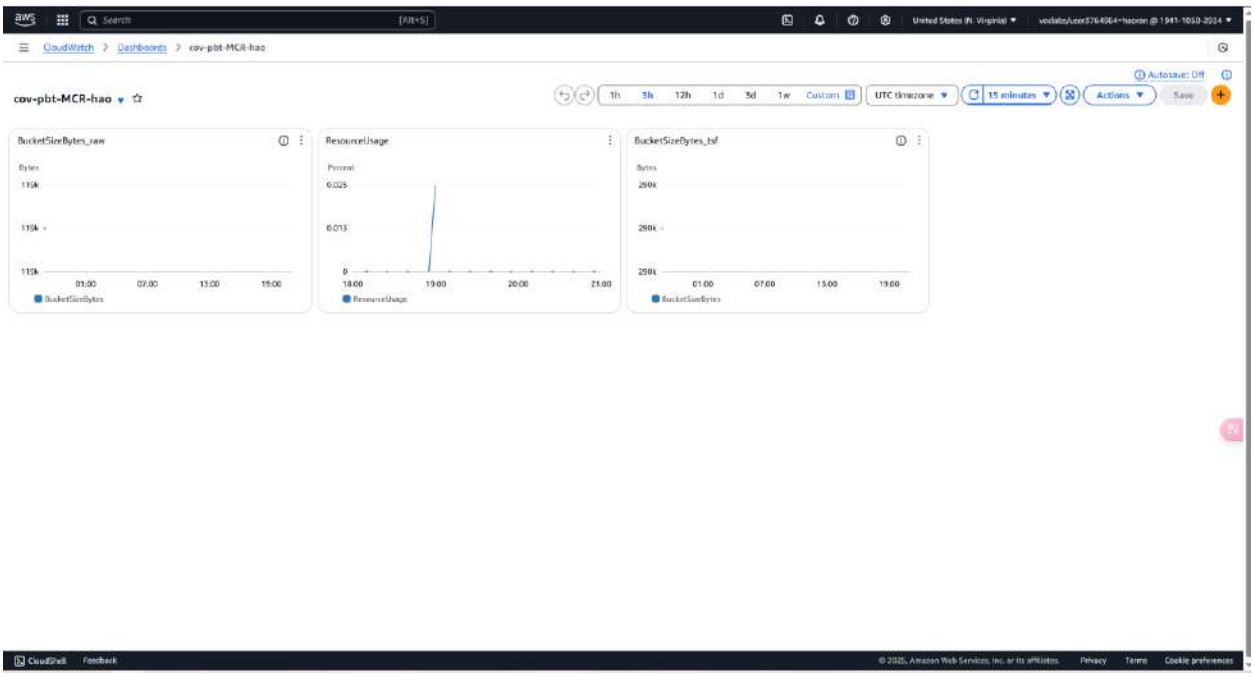
Figure 61: Quality Check Passed Output



Note. This figure shows the screenshot of the Passed output in Transfer Bucket of S3. Source: AWS S3

Step 8: Data Monitoring

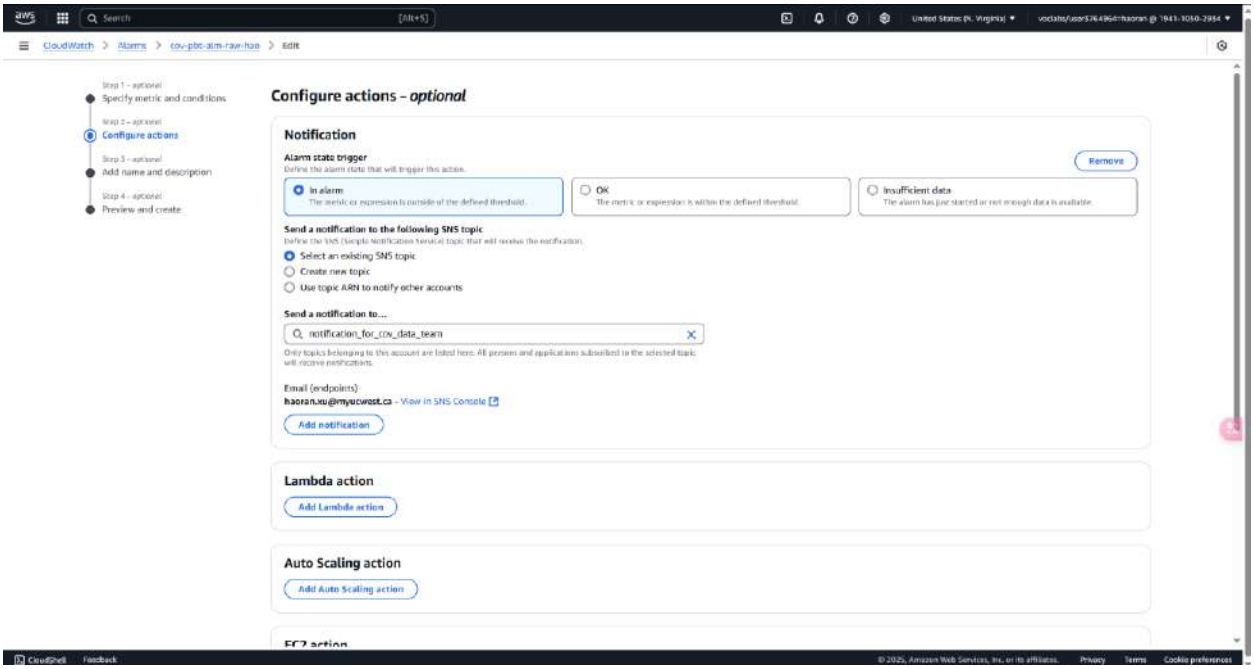
Figure 62: Dashboard of CloudWatch



Note. This figure shows the screenshot of the dashboard created in CloudWatch. Source: AWS CloudWatch

It’s for monitoring the usage of raw bucket size, job run resource usage and usage of transfer bucket size.

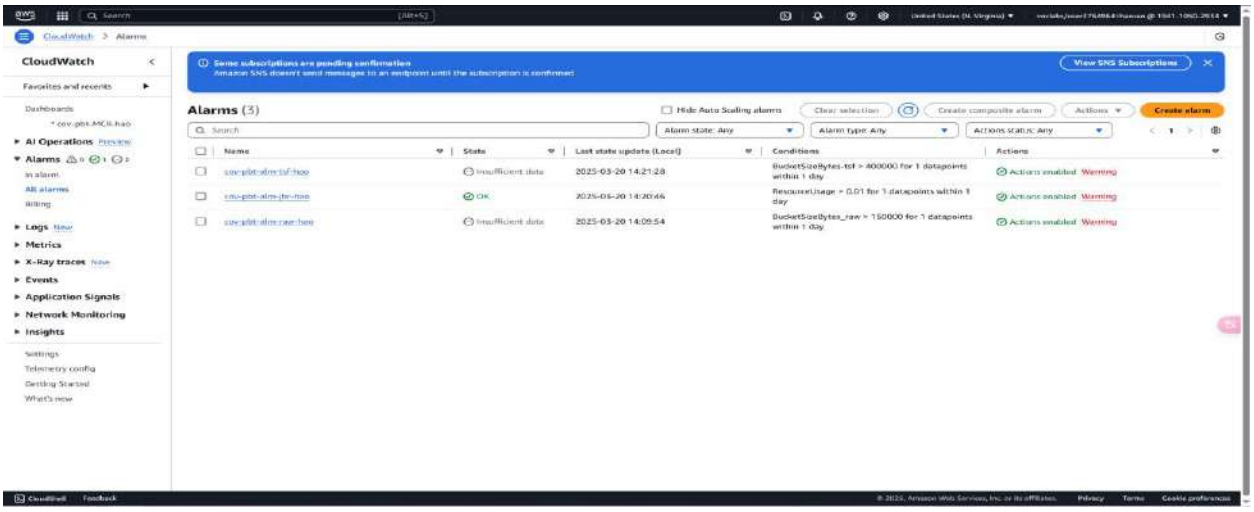
Figure 63: Alarm-Create New Topic for SNS



Note. This figure shows the screenshot of creating new SNS topic in CloudWatch. Source: AWS CloudWatch

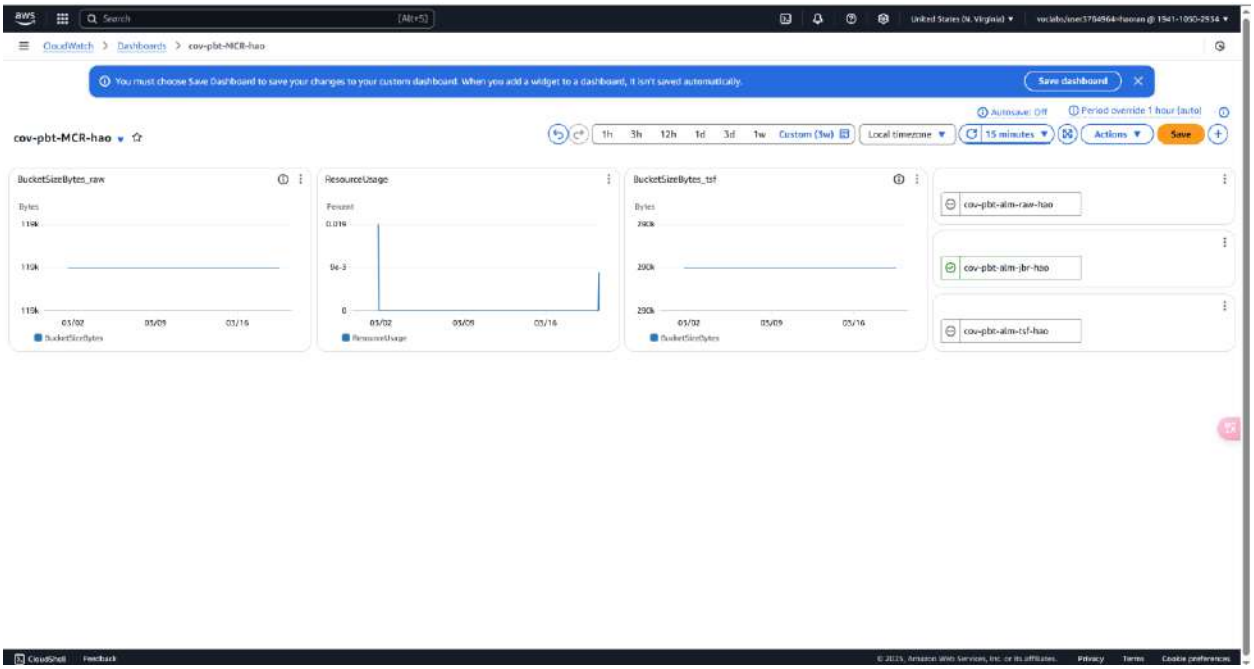
Here, I'm creating Alarm to monitoring the usage of three metrics. If the usage exceed specific amount that I set, I'll get the alarm message from AWS CloudWatch to notify me through email, so I can do some adjustment.

Figure 64: Three Alarms Screenshot



Note. This figure shows the screenshot of all alarms in CloudWatch. Source: AWS CloudWatch

Figure 65: Adding Alarms into the Dashboard



Note. This figure shows the screenshot of all alarms added into the dashboard in CloudWatch. Source: AWS CloudWatch

---

DAP Architecture Analysis (Teamwork)

Operational Excellence *(Haoran)*

Operational excellence is definitely a very important part when running DAP. Effective operational excellence management can enable various projects to operate effectively, and the main responsibilities are to create, monitor and maintain the operation of the system. At the beginning of Project 1, we downloaded data from the City of Vancouver data portal to prepare raw data. Here we found and created the raw dataset. After that, we went through profiling and

cleaning steps to ensure that our data was ready to use before transfer. Here, our role in operational excellence is to ensure the integrity and cleanliness of the dataset, paving the way for subsequent use by the data team. We also need to monitor the system in daily use, which can be reflected that all our steps on AWS were recorded, which ensured that modifications were traceable. This helped us ensure system stability and data integrity in our operational excellence. At the same time, Project 2 does not require us to use cloud trail to monitor various actions. However, this is very helpful to ensure operational excellence. If we encounter problems or failures, we can find the problems from the past log history and learn from them to improve the system. We also need to adjust our operating environment and strategies at any time according to customer needs and changes in the business environment. This is also an important part of ensuring effective operational excellence.

Security *(Muhammad)*

In our group project, we all worked with our own datasets and cloud environments, but we followed the same architecture guidelines and AWS best practices. So even though we planned things together, every one of us set up and implemented security individually. This section summarizes how each team member handled security in their own setup.

• **Kyle** worked on the *Vancouver Street Trees* dataset. He created his own KMS key and encrypted his buckets, enabled versioning, used CloudTrail for auditing, and set up alarms for billing thresholds. He also ran transformation jobs in Glue and used Athena for querying.

• **Haoran** used the *Parking Tickets* dataset. His environment included encrypted buckets with KMS, CloudTrail logs, S3 replication for backup, and detailed Glue pipelines with multiple transformations. He also had a separate bucket for storing processed data and applied logging and monitoring.

• **Gyanvi** worked with the *Animal Control Inventory* dataset. She also created her own KMS key, enabled S3 versioning, and configured lifecycle rules, CloudTrail logging, and CloudWatch monitoring. She created her Glue jobs and followed the same structure.

• **Zulqarnain** used the *City of Vancouver Business Licenses* dataset. He created a custom KMS key, encrypted all S3 buckets, enabled versioning, used lifecycle rules, and set up CloudTrail for tracking. He also made CloudWatch alarms for usage and billing and enabled SNS email notifications.

All of us implemented our own identity control using IAM and CloudTrail. We didn't share IAM roles or users each person had their own environment to manage. No EC2 instances were used, so most of our security work focused on S3, Glue, CloudTrail, and CloudWatch.

Implement a Strong Identity Foundation

Did we use it? Yes

How: Everyone used CloudTrail to track activity and manage identity access in their own AWS accounts. IAM permissions were applied based on the least privilege principle. No shared accounts or public access.

Enable Traceability

Did we use it? Yes

How: All of us enabled CloudTrail and used it to track every action in AWS. We also set up CloudWatch alarms. For example, Zulqarnain and Kyle monitored billing and S3 activity with custom alarms.

Apply Security at All Layers

Did we use it? Yes

How: We applied encryption, versioning, and blocking public access to all S3 buckets. CloudTrail monitored user actions. CloudWatch helped track S3 storage and Glue job activity.

Automate Security Best Practices

Did we use it? Yes

How: Everyone configured alarms in CloudWatch. Zulqarnain added SNS for email alerts. Although no auto-remediation was used, our alerts helped us respond faster.

Protect Data in Transit and at Rest

Did we use it? Yes

How: All S3 buckets were encrypted with KMS keys. CloudTrail logs were stored in encrypted buckets too. This covered both transit (data moving across services) and at rest (data in storage).

Keep People Away from Data

Did we use it? Yes

How: CloudTrail logs and datasets were encrypted and not directly accessed. Glue jobs processed the data, and logging was automatic, reducing manual exposure.

Prepare for Security Events

Did we use it? Yes

How: We didn't run a full incident simulation, but CloudTrail logs and CloudWatch alarms gave us a solid setup to catch anything unusual. If something went wrong, we would know right away.

Table 1: Security Summary Table

Security Pillar	Did we use it?	How it was applied
Implement a Strong Identity Foundation	Yes	IAM and CloudTrail used by everyone individually
Enable Traceability	Yes	CloudTrail logs + CloudWatch alarms
Apply Security at All Layers	Yes	S3 encryption, versioning, monitoring
Automate Security Best Practices	Yes	Alarms + SNS email alerts
Protect Data in Transit and at Rest	Yes	KMS encryption + secure storage
Keep People Away from Data	Yes	Logs encrypted, no manual access
Prepare for Security Events	Yes	Alerts and logs prepared us for issues

Note: This is the self-made table about the security summary

Reliability *(Muhammad)*

Just like with security, all of us managed our own reliability setup in our own AWS accounts. We followed the same structure, so the monitoring and fault tolerance ideas were the same just applied individually. We didn't use EC2 or advanced networking, but we did rely on S3, Glue, CloudTrail, and CloudWatch to keep things reliable and monitored.

Automatically Recover from Failure

Did we use it? Partially

How: Everyone set up CloudWatch alarms for job failures, billing, and usage. But we didn't use auto-recovery like restarting jobs or scaling resources automatically.

Test Recovery Procedures

Did we use it? No

How: We didn't simulate any failures or test recovery plans. This is something we could add in future projects.

Scale Horizontally to Increase Availability

Did we use it? No

How: We didn't use horizontal scaling or load balancing since we weren't working with EC2 or dynamic web services.

Stop Guessing Capacity

Did we use it? Partially

How: We used CloudWatch to monitor storage and costs, but we didn't scale any services automatically. Glue jobs had retried, which helped a little.

Manage Change in Automation

Did we use it? Yes

How: Everyone used CloudTrail and CloudWatch to track changes. This reduced the need for manual checking.

Table 2: Reliability Questions Overview

Reliability Area	***Did we use it?***	***Notes***
Automatically Recover from Failure	Partially	Alarms set up, but no automatic fixes
Test Recovery Procedures	No	No simulations or recovery drills done
Scale Horizontally to Increase Availability	No	Not applicable for this project
Stop Guessing Capacity	Partially	CloudWatch monitored usage, but no autoscaling
Manage Change in Automation	Yes	All changes logged and tracked with alerts
Service Quotas & Network Topology	No	Not included in our scope

Workload Architecture	Partially	Each setup worked well but lacked redundancy or fault isolation
Change Management - Monitor Resources	Yes	Done using CloudWatch
Change Management - Adapt to Demand	No	No scaling features implemented
Change Management - Implement Change	Yes	Automated monitoring helped manage changes
Failure Management - Data Backup	No	No formal backup strategy set up
Failure Management - Fault Isolation	No	Everything ran in one region/account per person
Failure Management - Withstand Failures	No	No failover or load balancing done
Failure Management - Test Reliability	No	Didn't test for reliability under failure conditions
Failure Management - Disaster Recovery	No	No disaster recovery plan made

Note: This is the self-made table about Reliability Questions Overview

Performance Efficiency *(Gyanvi)*

Table 3: Performance Efficiency Questions

Performance efficiency questions	**Implemented (Yes/No)**

Selection	
How do you select the best-performing architecture?	No
How do you select your compute solution?	No
How do you select your storage solution?	Yes
How do you select your database solution?	No
How do you configure your networking solution?	No
Review	
How do you evolve your workload to take advantage of new releases?	No
Monitoring	
How do you monitor your resources to ensure they are performing?	Yes
Trade-offs	
How do you use trade-offs to improve performance?	Yes

Note: This is the self-made table about Performance Efficiency Questions

Performance efficiency questions:

1. Selection

1.1. How do you select the best-performing architecture?

We did not implement this specifically as we used the default architecture setup for performance. We can improve by evaluating different architecture based on our requirements and selecting the best-performing one.

1.2. How do you select your compute solution?

We did not create an EC2 instance for our project nor did we choose the auto-scaling option to automatically create EC2 instances based on demand. We can improve by selecting the EC2 instance based on our workload.

1.3. How do you select your storage solution?

We used the S3 standard option for storage in our project, however, Gyanvi implemented a lifecycle rule to switch storage class from standard to Intelligent tiering after 30 days and then to Glacier Instant Retrieval after 180 days to optimize cost. We can improve by exploring more options for storage solutions like EBS for enhancing cost and performance efficiency.

1.4. How do you select your database solution?

We did not implement a database solution in our project. We can improve by using database services like Amazon RDS to optimize performance.

1.5. How do you configure your networking solution?

We did not implement any networking solution configuration. We can improve by using VPC or content delivery networks to enhance performance.

2. Review

2.1. How do you evolve your workload to take advantage of new releases?

We did not evolve our workload to take advantage of the new releases. We can improve by implementing more features offered by AWS to improve performance.

3. Monitor

3.1. How do you monitor your resources to ensure they are performing?

We implemented resource monitoring by using the AWS CloudWatch feature. We created three metrics and three alarms in a dashboard to track billing and CPU usage in real time. We can improve by utilizing enhanced CloudWatch features to optimize performance.

4. Trade-offs

4.1. How do you use trade-offs to improve performance?

We implemented trade-offs using compression techniques like storing system files in parquet format with snappy compression type. However, we can improve by enabling caching for read-heavy workloads or compressing log data to improve performance.

Performance efficiency design principles

1. Democratize Advanced Technologies

We democratised advanced technologies like AWS CloudWatch and CloudTrail which handles monitoring and user activity logging. We did not build any custom solutions for infrastructure management.

2. Go Global in Minutes

We did not deploy our system in multiple AWS regions to reduce latency or improve the global user experience. This was not under the scope of our project.

3. Use Serverless Architectures

We have used many serverless architectures like CloudWatch, and CloudTrail for monitoring and logging user activities.

4. Experiment More Often

We did not perform comparative analysis and experimentation of various storage and computing techniques to choose the best-performing configuration.

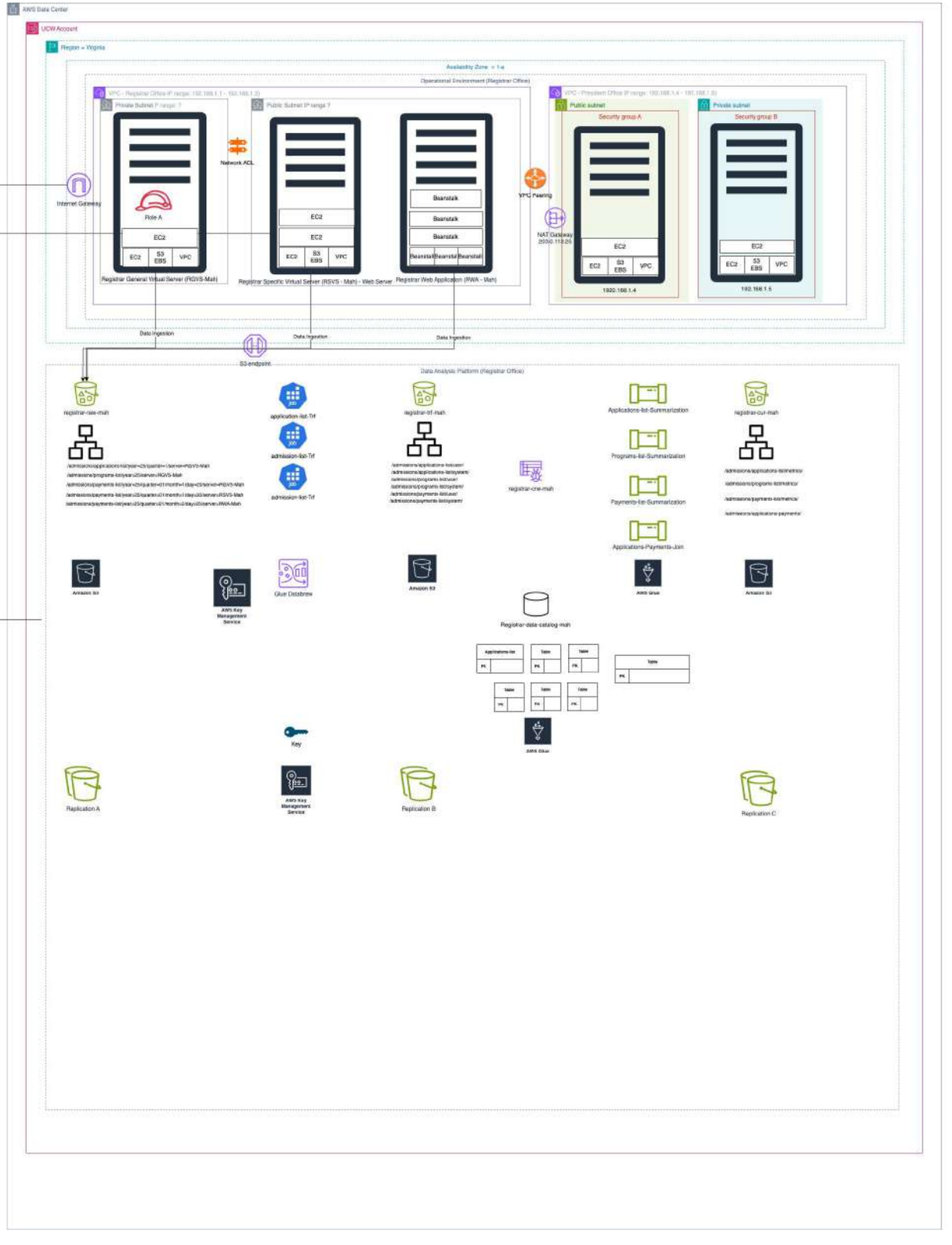
5. Have Mechanical Sympathy

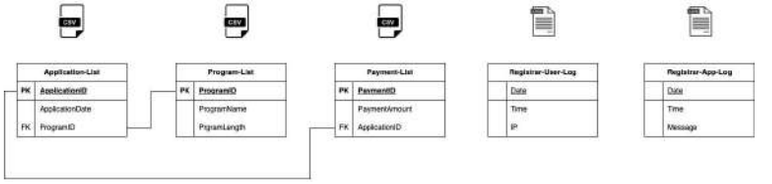
We selected the appropriate AWS services such as CloudWatch and CloudTrail for monitoring and logging. We did not deeply analyze data access patterns or optimize databases or storage.

Cost Optimization *(Peng)*

We have already adopted a consumption model for both parts of the project. That is because we paid only for the AWS services that we require based on the requirements of the City of Vancouver rather than using elaborate forecasting. For example, the city of Vancouver clearly requires us to do the data ingestion in part 1 of the project and the data monitoring in part 2 of the project. In this case, we paid only for buckets in S3 and Dashboard and Alarms in CloudWatch. Additionally, we didn't measure overall efficiency, but we operated some functions with the awareness of cost reduction. For example, in part 2 of the project, although we did not measure overall efficiency, we considered the cost and set 1 day for monitoring data for setting alarms. Moreover, we stopped spending money on data center operations in both parts of this project. That is because we mainly focus on the City of Vancouver to proceed with its business projects. Next, we analyzed and attributed expenditure in part 1 of the project but not in part 2 of the project. That is because we have already done the cost estimation for 12 months in part 1 of the project, and our team has an opportunity to optimize the resources and reduce the costs. Finally, we did not technically use managed and application-level services to reduce the cost of ownership in both parts of the project, but we considered the cost reduction when operating

services at AWS. Although we did not technically reduce the operational burden of maintaining servers for tasks such as sending emails or managing databases, we considered the cost when operating services at AWS. For example, we considered the cost and set 1 day for monitoring data for setting alarms, which not only reduced the cost but also managed databases.


AWS Data Center
UCW Account
Region = Virginia
Availability Zone = 1a
Operational Environment (Registrar Office)
VPC - Registrar Office IP range: 192.168.1.1 - 192.168.1.3)
Private Subnet ? range: ?
Public Subnet IP range ?
VPC - Prestem Office IP range: 192.168.1.4 - 192.168.1.5)
Public subnet
Security group A
Private subnet
Security group B
Role C
Registrar Operation Team
Internet Gateway
Internet
Registrar End User
Role B
Role A
EC2
S3
EBS
VPC
Network ACL
Beanstalk
VPC Peering
NAT Gateway
2030.113.25
192.168.1.4
192.168.1.5
Registrar General Virtual Server (RGVS-Mah)
Registrar Specific Virtual Server (RSVS - Mah) - Web Server
Registrar Web Application (RWA - Mah)
Data Ingestion
S3-endpoint
Data Analysis Platform (Registrar Office)
registrar-raw-mah
application-list-Trf
admission-list-Trf
registrar-trf-mah
registrar-cur-mah
Applications-list-Summarization
Programs-list-Summarization
Payments-list-Summarization
Applications-Payments-Join
registrar-cnw-mah
Registrar Data Team
Amazon S3
AWS Key Management Service
Glue Databrew
AWS Glue
Registrar-data-catalog-mah
Key
Replication A
Replication B
Replication C


Application-List
PK ApplicationID
ApplicationDate
FK ProgramID
Program-List
PK ProgramID
ProgramName
PgramLength
Payment-List
PK PaymentID
PaymentAmount
FK ApplicationID
Registrar-User-Log
Date
Time
IP
Registrar-App-Log
Date
Time
Message


Log
Metric
alarm
Dashboard
Lambda Function
AWS CloudTrail
Amazon CloudWatch
Amazon SNS
AWS Lambda

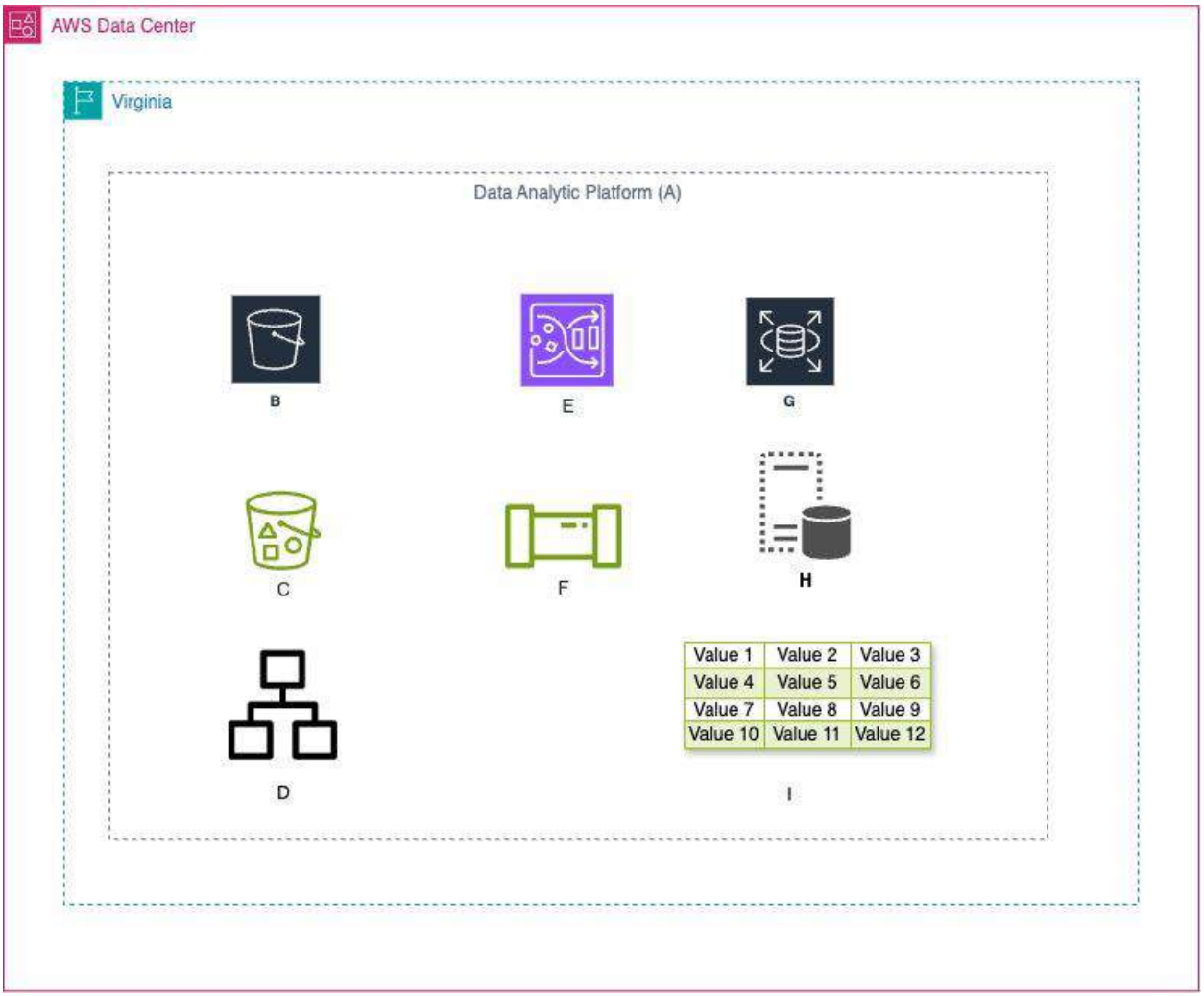
Sample Question and Answer for BUSI 653

Question:

Business Requirements

Managers in your domain decided to store their day-to-day datasets in the AWS Cloud. They prefer to store and prepare their datasets of each procedure in separate places for different types of analysis. For each procedure, they have one structured high-quality relational dataset and one unstructured low-quality dataset. Which AWS solution do you suggest. Please draw it and explain each component of your solution (What is the component, and why do they need it, and detail of the component).

Answer:



A: *The name of your domain*

B: *The name of the service, why did you select this service?*

C: *The name of the resource based on your procedure, why did you select this resource?*

D: *The structure of the resource based on your procedure, why did you consider this structure?*

E: *The name of the service, why did you select this service?*

F: *The name of the resource based on your procedure, why did you select this resource? The details about the resource and the reasons.*

G: *The name of the service, why did you select this service?*

H: *The name of the resource based on your procedure, why did you select this resource?*

I: *The structure of the resource based on your procedure, why did you consider this structure?*

Exploratory Data Analysis

Project Description: Exploratory Data Analysis (EDA) on Titanic Dataset

Project Title: Surviving the Titanic: An Exploratory Data Analysis

Objective: The primary goal of this project is to perform an exploratory data analysis (EDA) on the Titanic dataset to uncover patterns, trends, and insights related to passenger survival. By analyzing various features such as age, gender, class, and fare, we aim to understand the factors that influenced the likelihood of survival during the Titanic disaster.

Dataset: The Titanic dataset consists of passenger information from the ill-fated voyage of the RMS Titanic, including details such as:

- Passenger ID: Unique identifier for each passenger
- Survived: Survival status (0 = No, 1 = Yes)

- Pclass: Passenger class (1st, 2nd, 3rd)
- Name: Name of the passenger
- Sex: Gender of the passenger
- Age: Age of the passenger
- SibSp: Number of siblings/spouses aboard
- Parch: Number of parents/children aboard
- Ticket: Ticket number
- Fare: Fare paid by the passenger
- Embarked: Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

Methodology:

1- Data Collection and Preparation:
   - Load the Titanic dataset using Python libraries like Pandas.
   - Perform initial data cleaning, which includes handling missing values, correcting data types, and renaming columns for clarity.

2- Descriptive Statistics:
   - Generate summary statistics (mean, median, mode) for numerical features (like Age and Fare) and frequency distributions for categorical features (like Pclass and Sex).

3- Data Visualization:
   - Create visualizations to illustrate key insights:
     - Histograms and Boxplots: Analyze the distribution of continuous variables like Age and Fare.
     - Bar Charts: Showcase survival rates across different categories (e.g., Sex, Pclass).
     - Heatmaps: Visualize correlations between numerical variables.

4- Survival Analysis:
   - Compare survival rates:
     - By gender: Determine if there is a significant difference in survival rates between male and female passengers.
     - By class: Analyze how passenger class affected survival chances.
     - By age group: Create age bins to assess survival across different age demographics.

5- Insights and Findings:
   - Summarize the findings based on data visualizations and statistical analyses, highlighting notable trends and patterns (e.g., women and children had higher survival rates, first-class passengers had a significant survival advantage).

6- Conclusion:
    - Discuss the implications of the findings and suggest further analyses or data-driven decisions that could be explored, such as building predictive models to classify survival based on passenger features.

Tools and Technologies:

- Python (Pandas, NumPy, Matplotlib, Seaborn)
- Jupyter Notebook for interactive data exploration
- Any additional data visualization tools like Tableau or Power BI (optional)

Deliverables:

- A comprehensive Jupyter Notebook containing all steps of the analysis, including code, visualizations, and narrative explanations of findings.
- A presentation summarizing key insights and visualizations for stakeholders or peers.

This EDA project not only demonstrates your analytical and programming skills but also highlights your ability to derive meaningful insights from data, making it a valuable addition to your data analyst portfolio.

Descriptive Analysis

Project Description: Descriptive Analysis of Customer Purchase Patterns

Project Title: Understanding Customer Purchase Patterns at XYZ Retail

Objective: The primary goal of this project is to conduct a descriptive analysis of customer purchase data at XYZ Retail. Through this analysis, we aim to summarize key characteristics of customer purchases, identify trends, and generate insights that can inform marketing strategies and inventory management.

Dataset: The dataset includes transactional data from XYZ Retail over the past year, containing the following key features:

- Transaction ID: Unique identifier for each purchase
- Customer ID: Identification number for each customer
- Purchase Date: Date and time of the transaction

- Product Category: Category of the purchased product (e.g., electronics, clothing, groceries)
- Quantity: Number of items purchased
- Price: Total price of the transaction
- Payment Method: Method used for payment (e.g., credit card, cash, digital payment)
- Location: Store location where the purchase was made

Methodology:

1- Data Collection and Preparation:
   - Load the dataset using data analysis tools (e.g., Python, Excel).
   - Perform data cleaning to address missing values, correct data types, and remove duplicates.
2- Descriptive Statistics:
   - Calculate summary statistics for key variables, including:
     - Total sales and average transaction value
     - Number of transactions per month
     - Distribution of purchases by product category
     - Average quantity purchased per transaction
3- Data Visualization:
   - Create visual representations to illustrate findings:
     - Time series graphs showing sales trends over the year.
     - Bar charts displaying the most popular product categories.
     - Pie charts representing the share of different payment methods.
     - Heatmaps of sales by location and time of day.
4- Customer Segmentation:
   - Segment customers based on their purchasing behavior (e.g., high-frequency vs. low-frequency buyers).
   - Analyze the purchasing patterns of different segments.
5- Insights and Findings:
   - Summarize the insights derived from the analysis, highlighting:
     - Peak shopping periods (e.g., holidays, weekends)
     - Trends in product category sales over time
     - Preferences in payment methods across customer segments
6- Recommendations:
   - Provide actionable recommendations based on the findings to inform inventory management, targeted marketing campaigns, and promotional strategies.

Tools and Technologies:

- Python (Pandas, Matplotlib, Seaborn) or Excel for data analysis
- Data visualization tools (Tableau or Power BI) for creating dashboards

Deliverables:

- A detailed report summarizing the methods, findings, and recommendations.
- Visualizations and dashboards to present key insights clearly.
- A presentation for stakeholders to communicate important findings and suggestions for future action.

This descriptive analysis project aims to provide a comprehensive understanding of customer purchase behaviors, enabling XYZ Retail to optimize its operations and enhance customer satisfaction.

Diagnostic Analysis

Project Description: Diagnostic Analysis of Sales Decline at XYZ Retail

Project Title: Investigating the Causes of Sales Decline at XYZ Retail

Objective: The primary goal of this project is to conduct a diagnostic analysis to identify the underlying causes of a recent decline in sales at XYZ Retail. By analyzing various data sources, we aim to uncover the factors contributing to this decline and provide actionable insights for management to formulate effective strategies for improvement.

Background: Over the past six months, XYZ Retail has experienced a noticeable decline in sales, which has prompted management to seek a thorough understanding of the reasons behind this trend. This analysis will not only help in identifying contributing factors but also guide remedial actions to reverse the decline.

Dataset: The analysis will utilize multiple datasets, including:

- Sales Data: Daily sales transactions from the past year, including product categories, transaction amounts, and customer demographics.
- Inventory Data: Records of stock levels for each product category over the same period, indicating any shortages or overstock situations.

- Customer Feedback: Survey data and feedback directly from customers regarding their shopping experiences, preferences, and complaints.
- Market Data: External factors, including regional economic indicators, competitor pricing, and market trends.

Methodology:

1- Data Collection and Preparation:
   - Consolidate and clean datasets from multiple sources to ensure accuracy and consistency.
   - Normalize data to make it suitable for analysis.
2- Trend Analysis:
   - Perform a thorough analysis of sales trends over the last year to identify specific periods and product categories where declines were most significant.
3- Correlation Analysis:
   - Identify correlations between sales decline and other variables, such as inventory levels, customer feedback, and market conditions.
   - Use statistical methods (like regression analysis) to quantify how strongly these factors influence sales.
4- Root Cause Analysis:
   - Conduct focus group discussions or interviews (if feasible) with store staff and management to gather qualitative insights on observed changes in customer behavior and inventory practices.
   - Utilize techniques such as the "5 Whys" or Fishbone Diagram to systematically investigate potential causes.
5- Segmentation Analysis:
   - Segment customers based on behaviors (e.g., frequency of purchase, average transaction size) to analyze differing impacts across segments.
6- Synthesis of Findings:
   - Integrate quantitative and qualitative data to uncover patterns and themes that indicate the most significant factors contributing to the sales decline.

Tools and Technologies:

- Data analysis tools (Python with libraries like Pandas and Scikit-learn, R, or SQL) for metrics calculations and correlation analysis.
- Visualization tools (Tableau or Power BI) for presenting findings and trends clearly to stakeholders.

Deliverables:

- A comprehensive diagnostic report that outlines the analysis process, findings, and confirmed root causes of the sales decline.
- Visualizations and dashboards summarizing key metrics and trends.
- Actionable recommendations for management, focusing on strategies to address identified issues and improve sales performance.

Timeline:

- Expected completion of the project: 6 weeks from project kickoff, including regular check-ins with stakeholders to align findings and recommendations.

This diagnostic analysis aims to provide a clear understanding of the reasons behind the decline in sales at XYZ Retail, enabling management to take targeted actions to boost revenue and enhance customer satisfaction in the future.

Data Wrangling

Project Description: Data Wrangling for Customer Analytics at XYZ Company

Project Title: Data Wrangling for Enhanced Customer Analytics at XYZ Company

Objective: The primary goal of this project is to perform comprehensive data wrangling to prepare a robust dataset for customer analytics at XYZ Company. By cleaning, transforming, and consolidating data from various sources, the project aims to enhance the accuracy and usability of customer data for subsequent analysis and reporting.

Background: XYZ Company has accumulated customer data from multiple channels, including sales transactions, customer service interactions, and marketing campaigns. However, this data is often inconsistent, incomplete, or fragmented, making it challenging to derive meaningful insights. Effective data wrangling will facilitate better decision-making and more targeted marketing strategies.

Dataset: The data wrangling process will involve various datasets, including:

- Sales Data: Transaction records that include customer IDs, purchase amounts, product details, and timestamps.

- Customer Information: Demographic details such as age, gender, location, and account creation date.
- Customer Service Records: Logs of customer inquiries, complaints, and resolutions.
- Marketing Interaction Data: Email and campaign response data, including open rates and click-through rates.

Methodology:

1- Data Collection:
   - Gather datasets from various sources, including internal databases, CRM systems, and third-party marketing platforms.
   - Ensure that all relevant datasets are identified for a comprehensive customer profile.

2- Data Assessment:
   - Conduct an initial assessment of the data quality to identify issues such as missing values, duplicates, and inconsistencies across different datasets.
   - Document data types, formats, and any discrepancies.

3- Data Cleaning:
   - Address missing values through appropriate methods (e.g., imputation or exclusion) based on their significance and context.
   - Remove duplicate records and correct inconsistencies in data formats (e.g., date formats, naming conventions).
   - Normalize categorical variables to ensure consistency across datasets (e.g., standardizing customer status as "active," "inactive," etc.).

4- Data Transformation:
   - Perform data type conversions to ensure that all fields are in suitable formats for analysis (e.g., converting strings to datetime objects).
   - Derive new features that may aid in analytics, such as total purchase amounts, frequency of purchases, or customer tenure.
   - Aggregate data as necessary to ensure that it aligns with the intended analysis (e.g., summarizing monthly sales per customer).

5- Data Consolidation:
   - Merge datasets into a unified customer database, ensuring that all relevant information is linked accurately through unique identifiers (e.g., customer ID).
   - Create a comprehensive view of each customer by combining sales, support, and marketing data.

7- Documentation and Validation:
   - Document the data wrangling process, including data sources, cleaning methods, and transformations applied to the dataset.

- Validate the final dataset through exploratory data analysis (EDA) to confirm accuracy and completeness.

Tools and Technologies:

- Python (using libraries like Pandas and NumPy) or R for data manipulation and cleaning.
- SQL for data extraction and initial assessment of data from relational databases.
- Jupyter Notebook or RStudio for interactive data wrangling and documentation.
- Visualization tools (like Matplotlib or Seaborn) to assist with EDA and quality checks.

Deliverables:

- A cleaned and transformed customer dataset ready for analysis, available in a suitable format (e.g., CSV, Excel Database).
- A comprehensive report documenting the data wrangling process, including challenges encountered, methods employed, and final dataset characteristics.
- Visualizations illustrating the key data insights and confirmations of data quality checks conducted during the process.

Timeline:

- Expected completion of the project: 6 weeks, including phases for assessment, cleaning, transformation, and documentation.

This data wrangling project aims to establish a high-quality dataset that enables XYZ Company to conduct effective customer analytics, ultimately enhancing marketing strategies, improving customer service, and driving overall business growth.

Data Quality Control

Project Description: Data Quality Control Initiative at ABC Enterprises

Project Title: Implementation of Data Quality Control Measures at ABC Enterprises

Objective: The primary objective of this project is to establish a comprehensive Data Quality Control (DQC) framework at ABC Enterprises. This framework will ensure the

accuracy, completeness, consistency, and reliability of the organization's data, enhancing decision-making processes and overall business performance.

Background: As ABC Enterprises continues to expand its operations and data sources, issues related to data quality have surfaced, including inaccuracies, duplicate records, and inconsistent formats. Poor data quality can lead to misguided business strategies, inefficiencies, and regulatory compliance risks. This project aims to implement robust data quality control measures to mitigate these issues.

Scope: The project will focus on the following key areas:

- Data Profiling: Analyzing existing datasets to assess quality levels.
- Data Cleansing: Developing processes to correct inaccuracies and eliminate duplicates.
- Data Validation: Implementing validation rules and checks to ensure data integrity.
- Monitoring and Reporting: Establishing ongoing monitoring processes and dashboards to track data quality metrics.
- Training and Awareness: Creating training programs for staff on data quality best practices.

Methodology:

1- Current State Assessment:
   - Conduct a thorough analysis of current data sources, workflows, and existing data quality challenges.
   - Identify the key datasets that significantly impact business operations and decision-making.

2- Data Profiling:
   - Utilize data profiling tools to assess the quality of identified datasets, focusing on completeness, uniqueness, validity, consistency, and accuracy.
   - Document findings to highlight areas requiring immediate attention.

3- Establish Data Quality Metrics:
   - Define clear data quality metrics and key performance indicators (KPIs) to evaluate and track data quality over time, such as error rates, duplicate records, and compliance with data standards.

4- Data Cleansing Processes:
   - Develop and implement procedures for data cleansing, which may include:
     - Removing duplicates and correcting errors.
     - Standardizing data formats and values.
     - Filling in missing values using appropriate imputation techniques.

5- Validation Rules and Procedures:
    - Set up validation rules for new data entries to reduce the risk of poor-quality data being introduced into the system.
    - Create data entry guidelines to promote consistency and accuracy.
6- Monitoring and Reporting:
    - Implement monitoring tools and dashboards that provide real-time data quality metrics and alerts for significant deviations.
    - Schedule regular reports to review data quality trends and performance against established KPIs.
7- Training and Best Practices:
    - Develop training materials and conduct workshops to educate employees on data quality principles, the importance of maintaining data integrity, and procedural best practices.
    - Foster a culture of accountability where employees recognize their role in ensuring data quality.
8- Feedback Mechanism:
    - Establish a feedback loop to continually assess and improve data quality processes based on user input and observed results.

Tools and Technologies:

- Data quality tools (such as Informatica Data Quality, Talend, or Trifacta) for profiling and cleansing.
- Data visualization tools (like Tableau or Power BI) for monitoring and reporting on data quality metrics.
- SQL or Python for data cleansing and automated validation scripts.

Deliverables:

- A comprehensive Data Quality Control plan detailing processes, metrics, and responsibilities.
- Documentation of data quality metrics and KPIs being tracked.
- Cleaned and validated datasets ready for analysis and reporting.
- Training resources, including materials and workshops, are designed to educate staff on data quality practices.
- A monitoring dashboard that visualizes data quality metrics in real-time.

Timeline:

- Expected completion of the project: 8 weeks, including assessment, implementation, training, and monitoring setup.

This Data Quality Control initiative aims to empower ABC Enterprises to enhance its data integrity and reliability, resulting in improved decision-making, operational efficiency, and compliance with regulatory requirements.

Course Completion Badge

Share your course completion badge in your portfolio. You can claim it using a module named “Badges and completion certificates” in your AWS Academy CF course.

Exploratory Data Analysis

Project Description: Exploratory Data Analysis (EDA) on Titanic Dataset

Project Title: Surviving the Titanic: An Exploratory Data Analysis

Objective: The primary goal of this project is to perform an exploratory data analysis (EDA) on the Titanic dataset to uncover patterns, trends, and insights related to passenger survival. By analyzing various features such as age, gender, class, and fare, we aim to understand the factors that influenced the likelihood of survival during the Titanic disaster.

Dataset: The Titanic dataset consists of passenger information from the ill-fated voyage of the RMS Titanic, including details such as:

- Passenger ID: Unique identifier for each passenger
- Survived: Survival status (0 = No, 1 = Yes)

- Pclass: Passenger class (1st, 2nd, 3rd)
- Name: Name of the passenger
- Sex: Gender of the passenger
- Age: Age of the passenger
- SibSp: Number of siblings/spouses aboard
- Parch: Number of parents/children aboard
- Ticket: Ticket number
- Fare: Fare paid by the passenger
- Embarked: Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

Methodology:

1- Data Collection and Preparation:
   - Load the Titanic dataset using Python libraries like Pandas.
   - Perform initial data cleaning, which includes handling missing values, correcting data types, and renaming columns for clarity.

2- Descriptive Statistics:
   - Generate summary statistics (mean, median, mode) for numerical features (like Age and Fare) and frequency distributions for categorical features (like Pclass and Sex).

3- Data Visualization:
   - Create visualizations to illustrate key insights:
     - Histograms and Boxplots: Analyze the distribution of continuous variables like Age and Fare.
     - Bar Charts: Showcase survival rates across different categories (e.g., Sex, Pclass).
     - Heatmaps: Visualize correlations between numerical variables.

4- Survival Analysis:
   - Compare survival rates:
     - By gender: Determine if there is a significant difference in survival rates between male and female passengers.
     - By class: Analyze how passenger class affected survival chances.
     - By age group: Create age bins to assess survival across different age demographics.

5- Insights and Findings:
   - Summarize the findings based on data visualizations and statistical analyses, highlighting notable trends and patterns (e.g., women and children had higher survival rates, first-class passengers had a significant survival advantage).

6- Conclusion:
    - Discuss the implications of the findings and suggest further analyses or data-driven decisions that could be explored, such as building predictive models to classify survival based on passenger features.

Tools and Technologies:

- Python (Pandas, NumPy, Matplotlib, Seaborn)
- Jupyter Notebook for interactive data exploration
- Any additional data visualization tools like Tableau or Power BI (optional)

Deliverables:

- A comprehensive Jupyter Notebook containing all steps of the analysis, including code, visualizations, and narrative explanations of findings.
- A presentation summarizing key insights and visualizations for stakeholders or peers.

This EDA project not only demonstrates your analytical and programming skills but also highlights your ability to derive meaningful insights from data, making it a valuable addition to your data analyst portfolio.

Descriptive Analysis

Project Description: Descriptive Analysis of Customer Purchase Patterns

Project Title: Understanding Customer Purchase Patterns at XYZ Retail

Objective: The primary goal of this project is to conduct a descriptive analysis of customer purchase data at XYZ Retail. Through this analysis, we aim to summarize key characteristics of customer purchases, identify trends, and generate insights that can inform marketing strategies and inventory management.

Dataset: The dataset includes transactional data from XYZ Retail over the past year, containing the following key features:

- Transaction ID: Unique identifier for each purchase
- Customer ID: Identification number for each customer
- Purchase Date: Date and time of the transaction

- Product Category: Category of the purchased product (e.g., electronics, clothing, groceries)
- Quantity: Number of items purchased
- Price: Total price of the transaction
- Payment Method: Method used for payment (e.g., credit card, cash, digital payment)
- Location: Store location where the purchase was made

Methodology:

1- Data Collection and Preparation:
   - Load the dataset using data analysis tools (e.g., Python, Excel).
   - Perform data cleaning to address missing values, correct data types, and remove duplicates.
2- Descriptive Statistics:
   - Calculate summary statistics for key variables, including:
     - Total sales and average transaction value
     - Number of transactions per month
     - Distribution of purchases by product category
     - Average quantity purchased per transaction
3- Data Visualization:
   - Create visual representations to illustrate findings:
     - Time series graphs showing sales trends over the year.
     - Bar charts displaying the most popular product categories.
     - Pie charts representing the share of different payment methods.
     - Heatmaps of sales by location and time of day.
4- Customer Segmentation:
   - Segment customers based on their purchasing behavior (e.g., high-frequency vs. low-frequency buyers).
   - Analyze the purchasing patterns of different segments.
5- Insights and Findings:
   - Summarize the insights derived from the analysis, highlighting:
     - Peak shopping periods (e.g., holidays, weekends)
     - Trends in product category sales over time
     - Preferences in payment methods across customer segments
6- Recommendations:
   - Provide actionable recommendations based on the findings to inform inventory management, targeted marketing campaigns, and promotional strategies.

Tools and Technologies:

- Python (Pandas, Matplotlib, Seaborn) or Excel for data analysis
- Data visualization tools (Tableau or Power BI) for creating dashboards

Deliverables:

- A detailed report summarizing the methods, findings, and recommendations.
- Visualizations and dashboards to present key insights clearly.
- A presentation for stakeholders to communicate important findings and suggestions for future action.

This descriptive analysis project aims to provide a comprehensive understanding of customer purchase behaviors, enabling XYZ Retail to optimize its operations and enhance customer satisfaction.

Diagnostic Analysis

Project Description: Diagnostic Analysis of Sales Decline at XYZ Retail

Project Title: Investigating the Causes of Sales Decline at XYZ Retail

Objective: The primary goal of this project is to conduct a diagnostic analysis to identify the underlying causes of a recent decline in sales at XYZ Retail. By analyzing various data sources, we aim to uncover the factors contributing to this decline and provide actionable insights for management to formulate effective strategies for improvement.

Background: Over the past six months, XYZ Retail has experienced a noticeable decline in sales, which has prompted management to seek a thorough understanding of the reasons behind this trend. This analysis will not only help in identifying contributing factors but also guide remedial actions to reverse the decline.

Dataset: The analysis will utilize multiple datasets, including:

- Sales Data: Daily sales transactions from the past year, including product categories, transaction amounts, and customer demographics.
- Inventory Data: Records of stock levels for each product category over the same period, indicating any shortages or overstock situations.

- Customer Feedback: Survey data and feedback directly from customers regarding their shopping experiences, preferences, and complaints.
- Market Data: External factors, including regional economic indicators, competitor pricing, and market trends.

Methodology:

1- Data Collection and Preparation:
   - Consolidate and clean datasets from multiple sources to ensure accuracy and consistency.
   - Normalize data to make it suitable for analysis.
2- Trend Analysis:
   - Perform a thorough analysis of sales trends over the last year to identify specific periods and product categories where declines were most significant.
3- Correlation Analysis:
   - Identify correlations between sales decline and other variables, such as inventory levels, customer feedback, and market conditions.
   - Use statistical methods (like regression analysis) to quantify how strongly these factors influence sales.
4- Root Cause Analysis:
   - Conduct focus group discussions or interviews (if feasible) with store staff and management to gather qualitative insights on observed changes in customer behavior and inventory practices.
   - Utilize techniques such as the "5 Whys" or Fishbone Diagram to systematically investigate potential causes.
5- Segmentation Analysis:
   - Segment customers based on behaviors (e.g., frequency of purchase, average transaction size) to analyze differing impacts across segments.
6- Synthesis of Findings:
   - Integrate quantitative and qualitative data to uncover patterns and themes that indicate the most significant factors contributing to the sales decline.

Tools and Technologies:

- Data analysis tools (Python with libraries like Pandas and Scikit-learn, R, or SQL) for metrics calculations and correlation analysis.
- Visualization tools (Tableau or Power BI) for presenting findings and trends clearly to stakeholders.

Deliverables:

- A comprehensive diagnostic report that outlines the analysis process, findings, and confirmed root causes of the sales decline.
- Visualizations and dashboards summarizing key metrics and trends.
- Actionable recommendations for management, focusing on strategies to address identified issues and improve sales performance.

Timeline:

- Expected completion of the project: 6 weeks from project kickoff, including regular check-ins with stakeholders to align findings and recommendations.

This diagnostic analysis aims to provide a clear understanding of the reasons behind the decline in sales at XYZ Retail, enabling management to take targeted actions to boost revenue and enhance customer satisfaction in the future.

Data Wrangling

Project Description: Data Wrangling for Customer Analytics at XYZ Company

Project Title: Data Wrangling for Enhanced Customer Analytics at XYZ Company

Objective: The primary goal of this project is to perform comprehensive data wrangling to prepare a robust dataset for customer analytics at XYZ Company. By cleaning, transforming, and consolidating data from various sources, the project aims to enhance the accuracy and usability of customer data for subsequent analysis and reporting.

Background: XYZ Company has accumulated customer data from multiple channels, including sales transactions, customer service interactions, and marketing campaigns. However, this data is often inconsistent, incomplete, or fragmented, making it challenging to derive meaningful insights. Effective data wrangling will facilitate better decision-making and more targeted marketing strategies.

Dataset: The data wrangling process will involve various datasets, including:

- Sales Data: Transaction records that include customer IDs, purchase amounts, product details, and timestamps.

- Customer Information: Demographic details such as age, gender, location, and account creation date.
- Customer Service Records: Logs of customer inquiries, complaints, and resolutions.
- Marketing Interaction Data: Email and campaign response data, including open rates and click-through rates.

Methodology:

1- Data Collection:
   - Gather datasets from various sources, including internal databases, CRM systems, and third-party marketing platforms.
   - Ensure that all relevant datasets are identified for a comprehensive customer profile.

2- Data Assessment:
   - Conduct an initial assessment of the data quality to identify issues such as missing values, duplicates, and inconsistencies across different datasets.
   - Document data types, formats, and any discrepancies.

3- Data Cleaning:
   - Address missing values through appropriate methods (e.g., imputation or exclusion) based on their significance and context.
   - Remove duplicate records and correct inconsistencies in data formats (e.g., date formats, naming conventions).
   - Normalize categorical variables to ensure consistency across datasets (e.g., standardizing customer status as "active," "inactive," etc.).

4- Data Transformation:
   - Perform data type conversions to ensure that all fields are in suitable formats for analysis (e.g., converting strings to datetime objects).
   - Derive new features that may aid in analytics, such as total purchase amounts, frequency of purchases, or customer tenure.
   - Aggregate data as necessary to ensure that it aligns with the intended analysis (e.g., summarizing monthly sales per customer).

5- Data Consolidation:
   - Merge datasets into a unified customer database, ensuring that all relevant information is linked accurately through unique identifiers (e.g., customer ID).
   - Create a comprehensive view of each customer by combining sales, support, and marketing data.

7- Documentation and Validation:
   - Document the data wrangling process, including data sources, cleaning methods, and transformations applied to the dataset.

- Validate the final dataset through exploratory data analysis (EDA) to confirm accuracy and completeness.

Tools and Technologies:

- Python (using libraries like Pandas and NumPy) or R for data manipulation and cleaning.
- SQL for data extraction and initial assessment of data from relational databases.
- Jupyter Notebook or RStudio for interactive data wrangling and documentation.
- Visualization tools (like Matplotlib or Seaborn) to assist with EDA and quality checks.

Deliverables:

- A cleaned and transformed customer dataset ready for analysis, available in a suitable format (e.g., CSV, Excel Database).
- A comprehensive report documenting the data wrangling process, including challenges encountered, methods employed, and final dataset characteristics.
- Visualizations illustrating the key data insights and confirmations of data quality checks conducted during the process.

Timeline:

- Expected completion of the project: 6 weeks, including phases for assessment, cleaning, transformation, and documentation.

This data wrangling project aims to establish a high-quality dataset that enables XYZ Company to conduct effective customer analytics, ultimately enhancing marketing strategies, improving customer service, and driving overall business growth.

Data Quality Control

Project Description: Data Quality Control Initiative at ABC Enterprises

Project Title: Implementation of Data Quality Control Measures at ABC Enterprises

Objective: The primary objective of this project is to establish a comprehensive Data Quality Control (DQC) framework at ABC Enterprises. This framework will ensure the

accuracy, completeness, consistency, and reliability of the organization's data, enhancing decision-making processes and overall business performance.

Background: As ABC Enterprises continues to expand its operations and data sources, issues related to data quality have surfaced, including inaccuracies, duplicate records, and inconsistent formats. Poor data quality can lead to misguided business strategies, inefficiencies, and regulatory compliance risks. This project aims to implement robust data quality control measures to mitigate these issues.

Scope: The project will focus on the following key areas:

- Data Profiling: Analyzing existing datasets to assess quality levels.
- Data Cleansing: Developing processes to correct inaccuracies and eliminate duplicates.
- Data Validation: Implementing validation rules and checks to ensure data integrity.
- Monitoring and Reporting: Establishing ongoing monitoring processes and dashboards to track data quality metrics.
- Training and Awareness: Creating training programs for staff on data quality best practices.

Methodology:

1- Current State Assessment:
   - Conduct a thorough analysis of current data sources, workflows, and existing data quality challenges.
   - Identify the key datasets that significantly impact business operations and decision-making.

2- Data Profiling:
   - Utilize data profiling tools to assess the quality of identified datasets, focusing on completeness, uniqueness, validity, consistency, and accuracy.
   - Document findings to highlight areas requiring immediate attention.

3- Establish Data Quality Metrics:
   - Define clear data quality metrics and key performance indicators (KPIs) to evaluate and track data quality over time, such as error rates, duplicate records, and compliance with data standards.

4- Data Cleansing Processes:
   - Develop and implement procedures for data cleansing, which may include:
     - Removing duplicates and correcting errors.
     - Standardizing data formats and values.
     - Filling in missing values using appropriate imputation techniques.

5- Validation Rules and Procedures:
    - Set up validation rules for new data entries to reduce the risk of poor-quality data being introduced into the system.
    - Create data entry guidelines to promote consistency and accuracy.
6- Monitoring and Reporting:
    - Implement monitoring tools and dashboards that provide real-time data quality metrics and alerts for significant deviations.
    - Schedule regular reports to review data quality trends and performance against established KPIs.
7- Training and Best Practices:
    - Develop training materials and conduct workshops to educate employees on data quality principles, the importance of maintaining data integrity, and procedural best practices.
    - Foster a culture of accountability where employees recognize their role in ensuring data quality.
8- Feedback Mechanism:
    - Establish a feedback loop to continually assess and improve data quality processes based on user input and observed results.

Tools and Technologies:

- Data quality tools (such as Informatica Data Quality, Talend, or Trifacta) for profiling and cleansing.
- Data visualization tools (like Tableau or Power BI) for monitoring and reporting on data quality metrics.
- SQL or Python for data cleansing and automated validation scripts.

Deliverables:

- A comprehensive Data Quality Control plan detailing processes, metrics, and responsibilities.
- Documentation of data quality metrics and KPIs being tracked.
- Cleaned and validated datasets ready for analysis and reporting.
- Training resources, including materials and workshops, are designed to educate staff on data quality practices.
- A monitoring dashboard that visualizes data quality metrics in real-time.

Timeline:

- Expected completion of the project: 8 weeks, including assessment, implementation, training, and monitoring setup.

This Data Quality Control initiative aims to empower ABC Enterprises to enhance its data integrity and reliability, resulting in improved decision-making, operational efficiency, and compliance with regulatory requirements.

Course Completion Badge

Share your course completion badge in your portfolio. You can claim it using a module named “Badges and completion certificates” in your AWS Academy CF course.