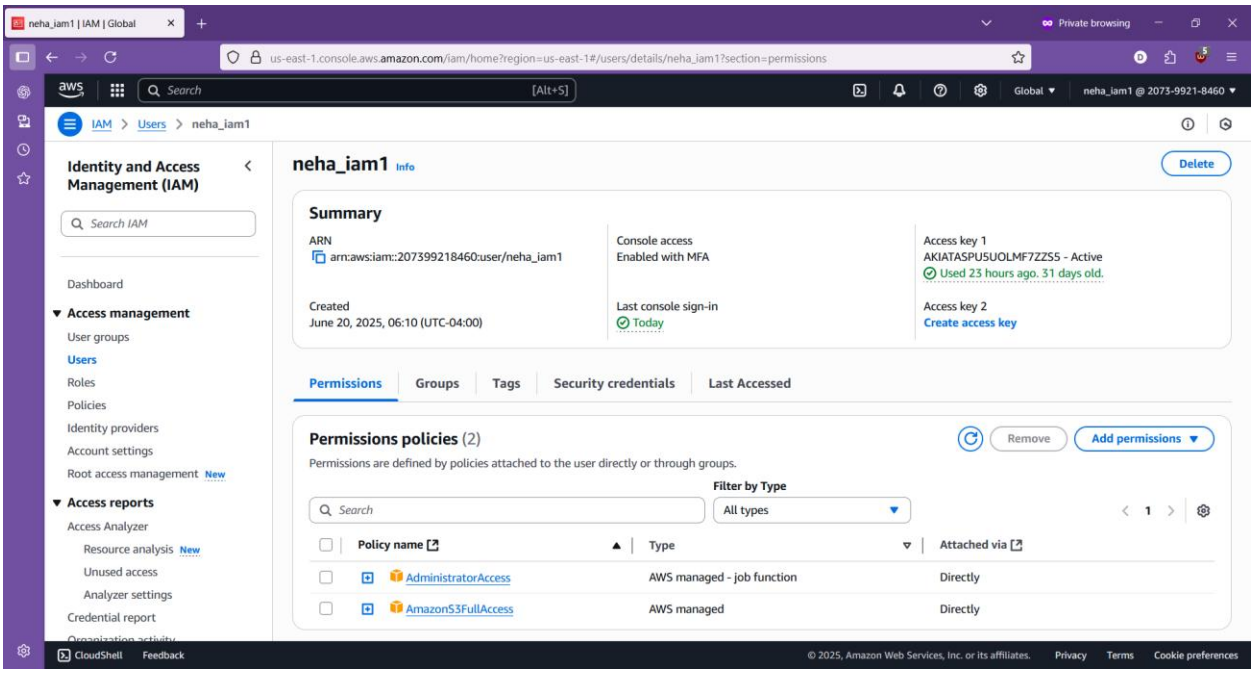
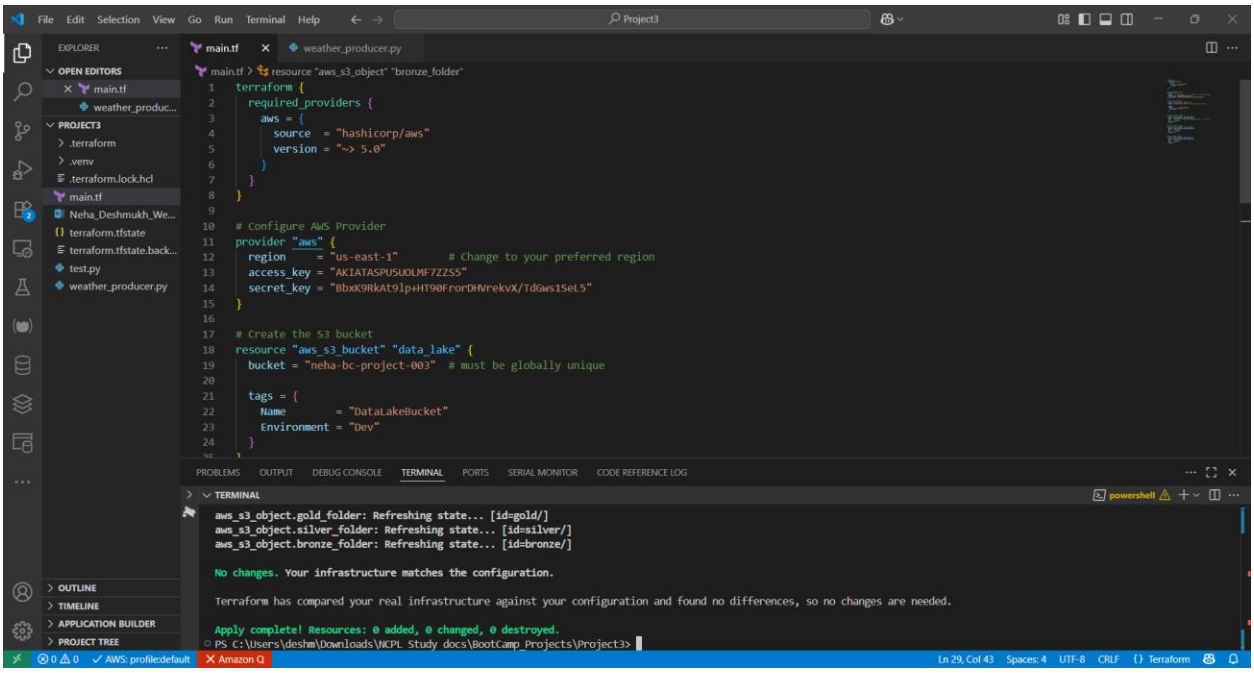
Building a Real-Time Weather Data Pipeline for Weather Analytics

1. Create S3 Bucket using Terraform

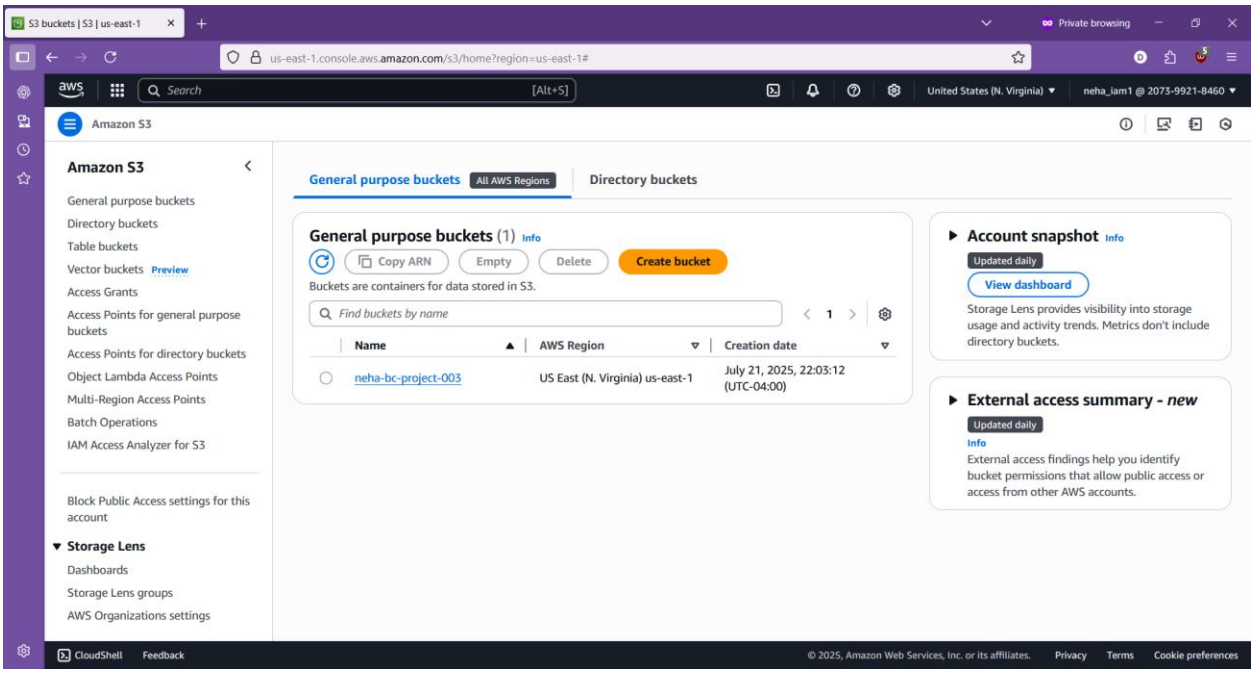
IAM User with Access code to create S3 bucket from Terraform

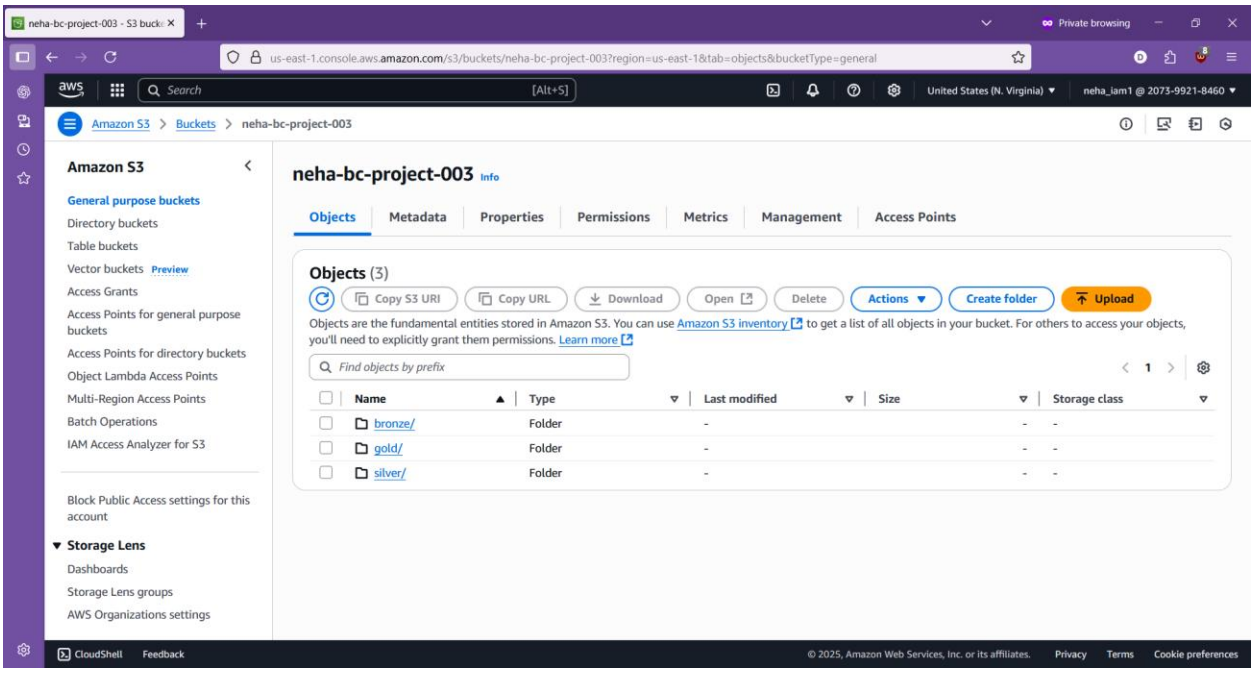


Create a Terraform Configuration File and run it:



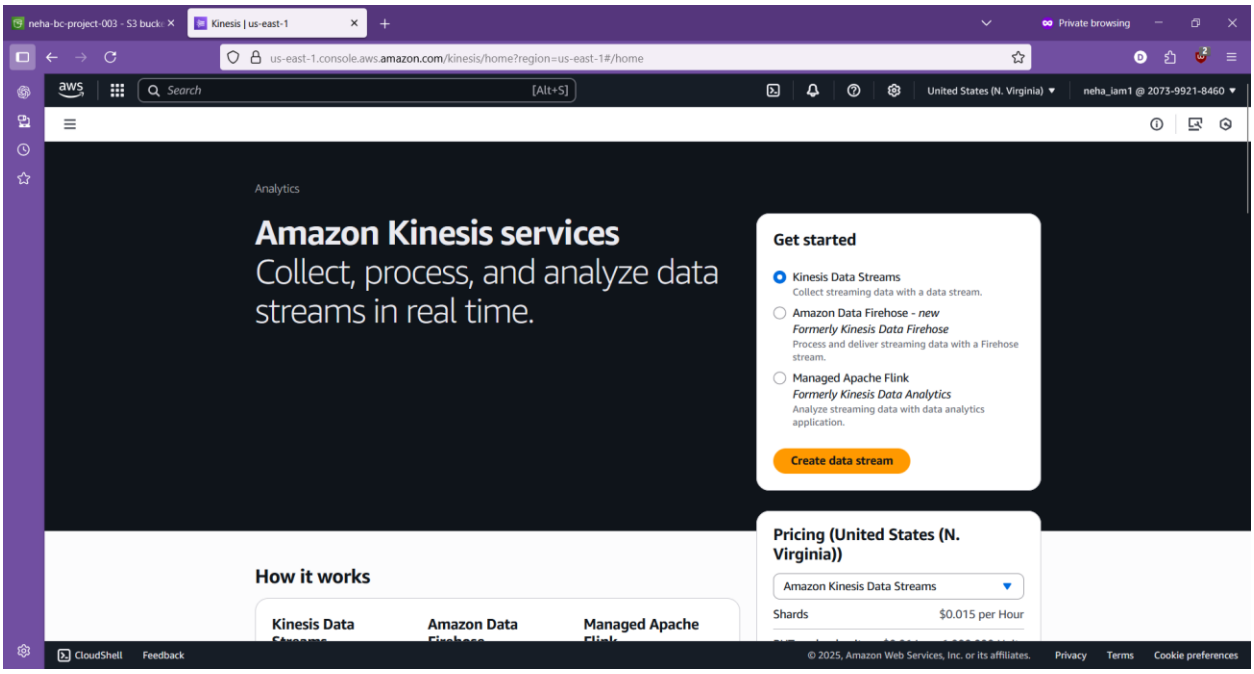
Verify whether the bucket is created as expected:

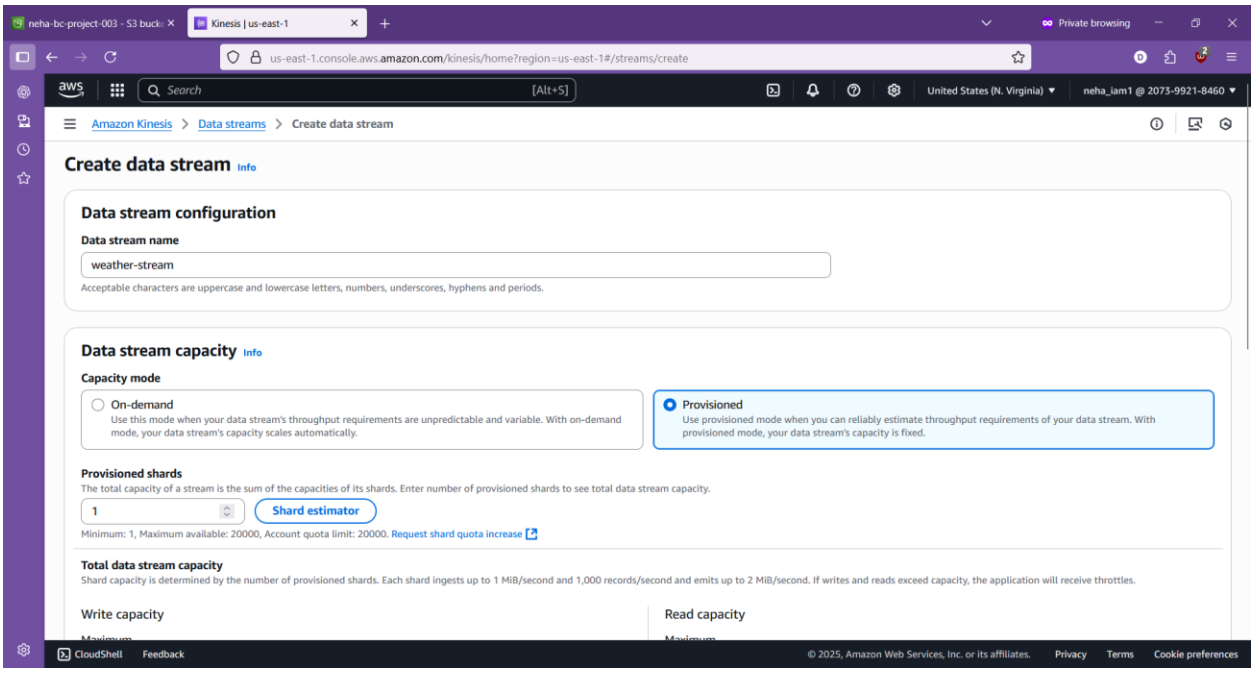


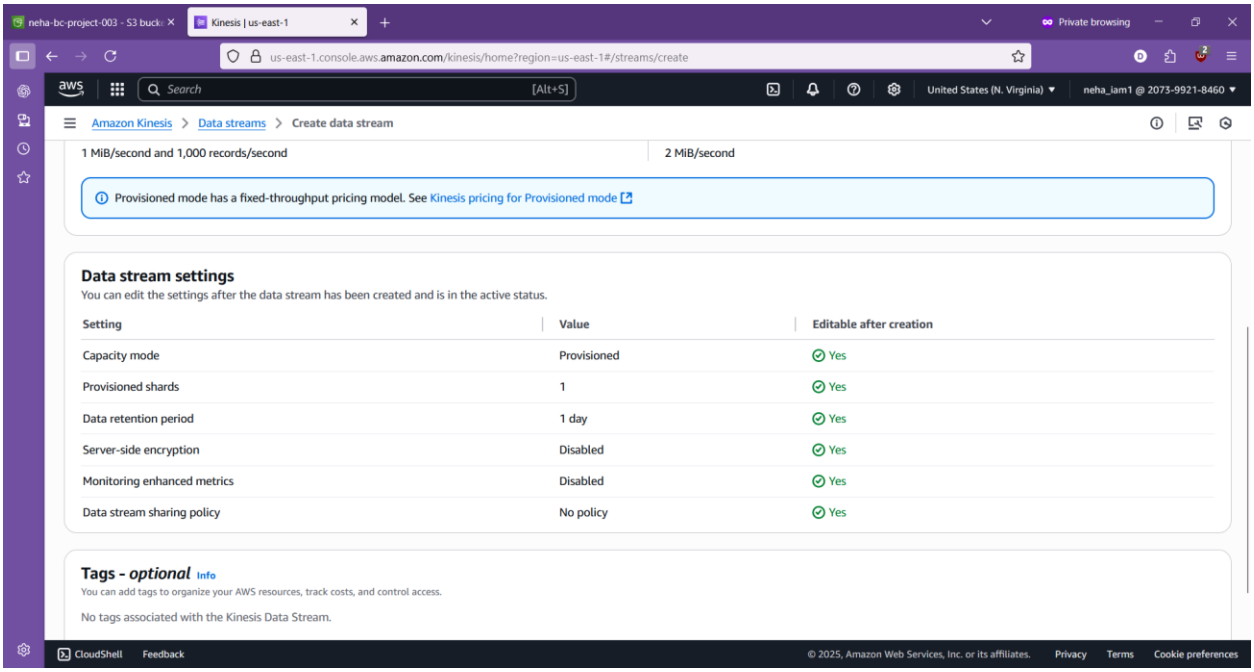


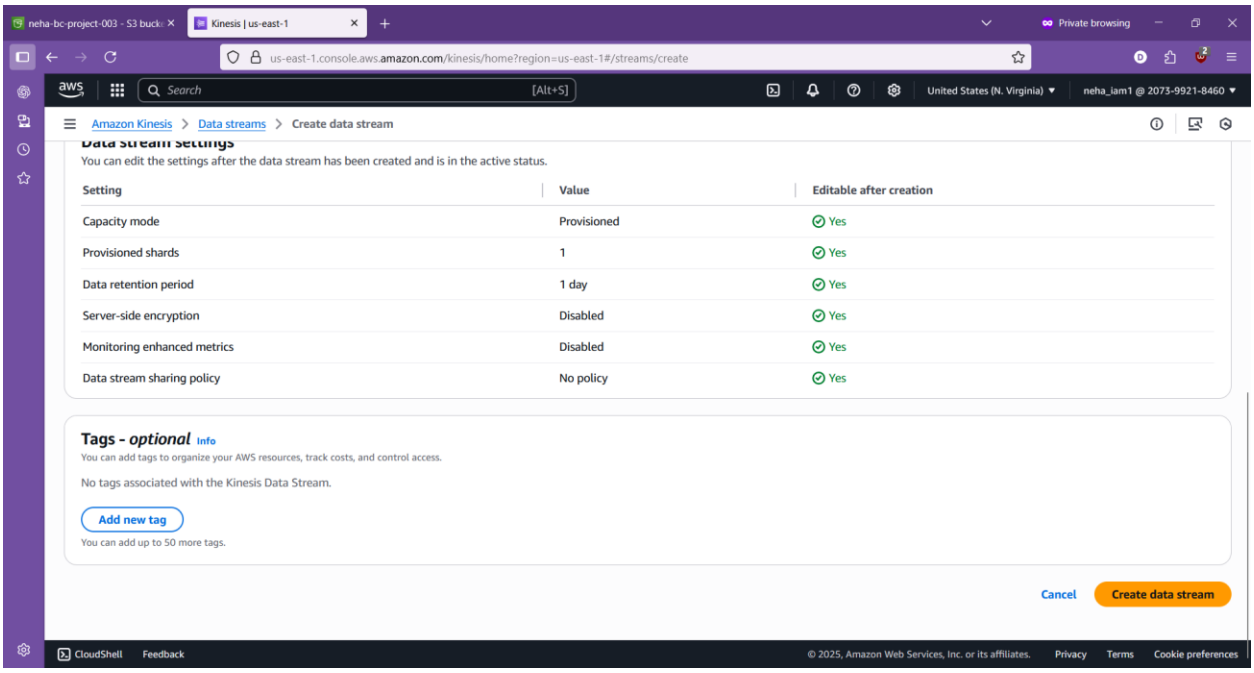
Bronze Layer

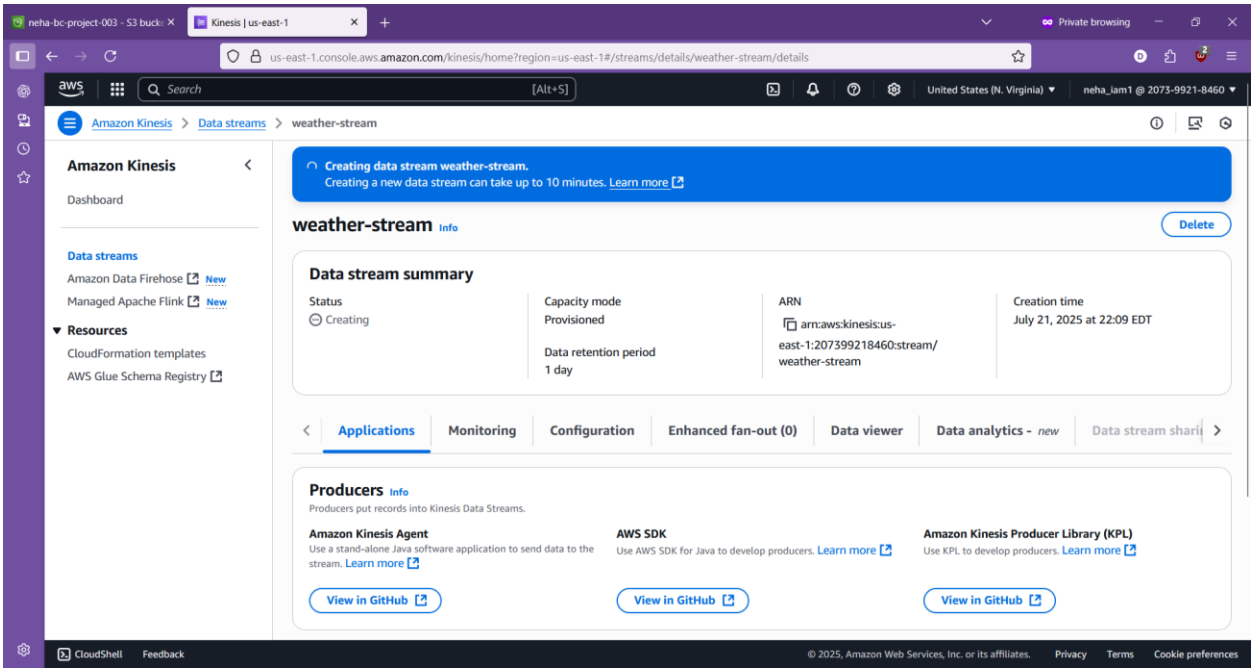
1. Create Kinesis Data Stream

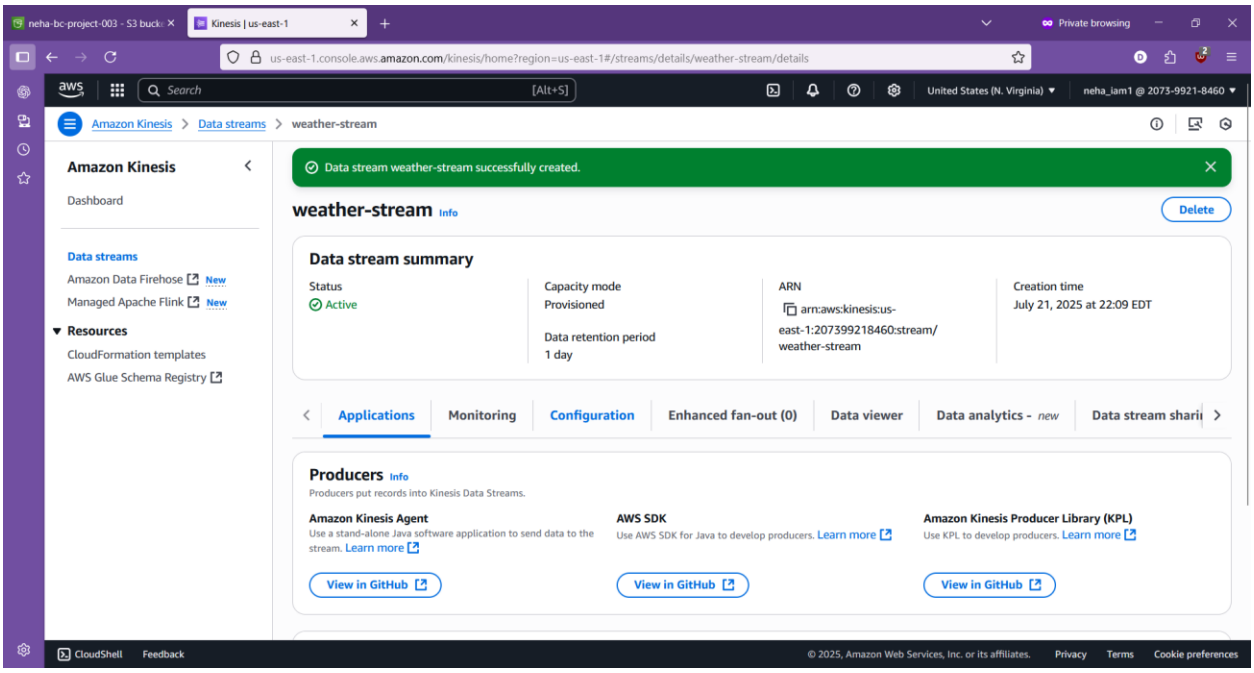


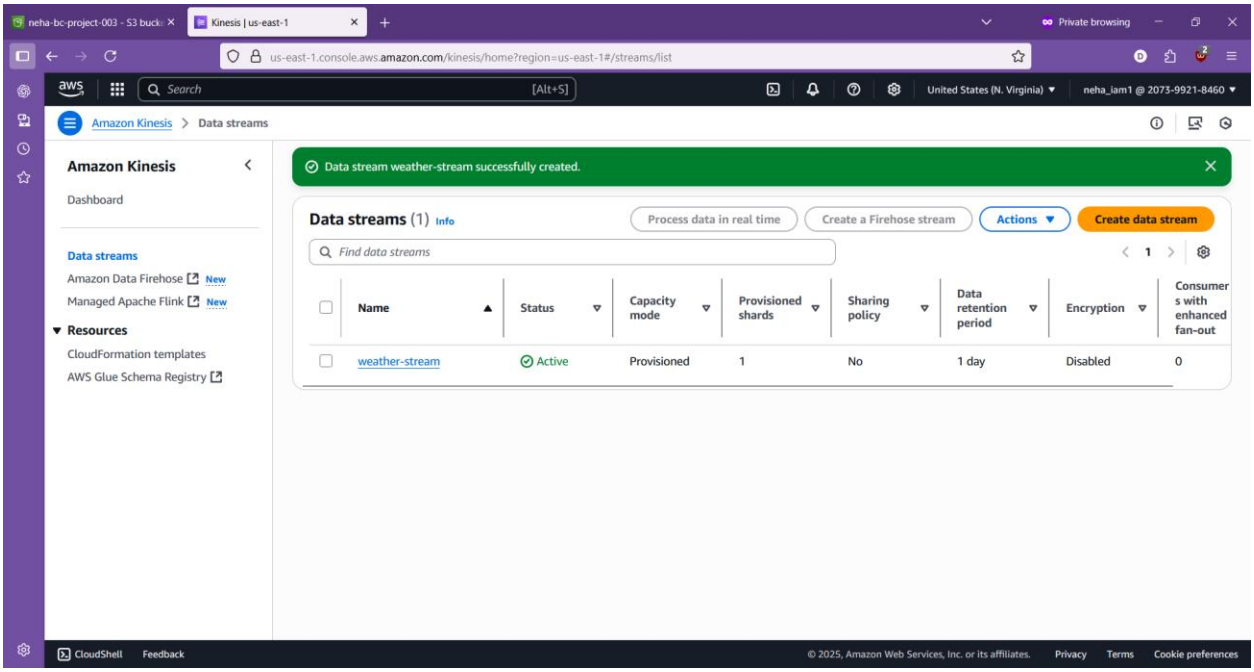

neha-bc-project-003 - S3 buck
Kinesis | us-east-1
Private browsing
us-east-1.console.aws.amazon.com/kinesis/home?region=us-east-1#/streams/create
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
Amazon Kinesis
Data streams
Create data stream
Create data stream Info
Data stream configuration
Data stream name
weather-stream
Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens and periods.
Data stream capacity Info
Capacity mode
On-demand
Use this mode when your data stream's throughput requirements are unpredictable and variable. With on-demand mode, your data stream's capacity scales automatically.
Provisioned
Use provisioned mode when you can reliably estimate throughput requirements of your data stream. With provisioned mode, your data stream's capacity is fixed.
Provisioned shards
The total capacity of a stream is the sum of the capacities of its shards. Enter number of provisioned shards to see total data stream capacity.
1
Shard estimator
Minimum: 1, Maximum available: 20000, Account quota limit: 20000. Request shard quota increase
Total data stream capacity
Shard capacity is determined by the number of provisioned shards. Each shard ingests up to 1 MiB/second and 1,000 records/second and emits up to 2 MiB/second. If writes and reads exceed capacity, the application will receive throttles.
Write capacity
Read capacity
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


neha-bc-project-003 - S3 buck
Kinesis | us-east-1
Private browsing
us-east-1.console.aws.amazon.com/kinesis/home?region=us-east-1#/streams/create
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
Amazon Kinesis
Data streams
Create data stream
1 MiB/second and 1,000 records/second
2 MiB/second
Provisioned mode has a fixed-throughput pricing model. See Kinesis pricing for Provisioned mode
Data stream settings
You can edit the settings after the data stream has been created and is in the active status.
Setting
Value
Editable after creation
Capacity mode
Provisioned
Yes
Provisioned shards
1
Yes
Data retention period
1 day
Yes
Server-side encryption
Disabled
Yes
Monitoring enhanced metrics
Disabled
Yes
Data stream sharing policy
No policy
Yes
Tags - optional Info
You can add tags to organize your AWS resources, track costs, and control access.
No tags associated with the Kinesis Data Stream.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Kinesis | us-east-1
Private browsing
us-east-1.console.aws.amazon.com/kinesis/home?region=us-east-1#/streams/create
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
Amazon Kinesis > Data streams > Create data stream
You can edit the settings after the data stream has been created and is in the active status.
Setting
Value
Editable after creation
Capacity mode
Provisioned
Yes
Provisioned shards
1
Yes
Data retention period
1 day
Yes
Server-side encryption
Disabled
Yes
Monitoring enhanced metrics
Disabled
Yes
Data stream sharing policy
No policy
Yes
Tags - optional Info
You can add tags to organize your AWS resources, track costs, and control access.
No tags associated with the Kinesis Data Stream.
Add new tag
You can add up to 50 more tags.
Cancel
Create data stream
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

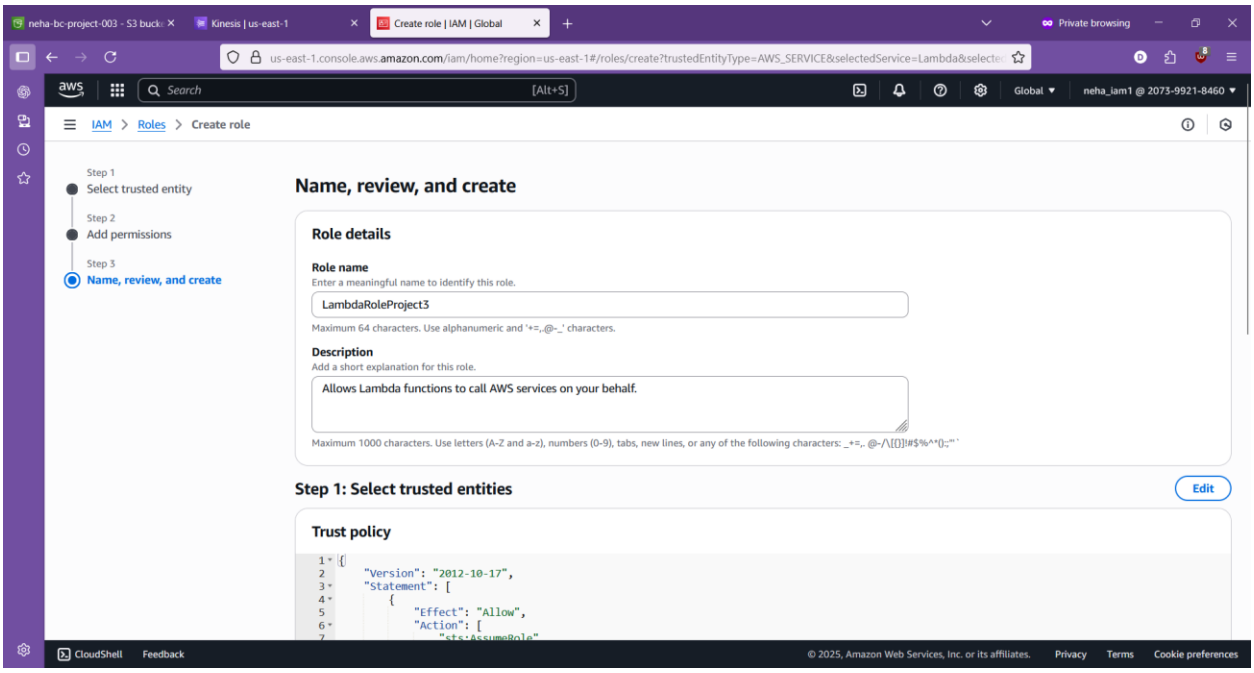

Kinesis | us-east-1
Private browsing
us-east-1.console.aws.amazon.com/kinesis/home?region=us-east-1#/streams/details/weather-stream/details
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
Amazon Kinesis > Data streams > weather-stream
Amazon Kinesis
Dashboard
Data streams
Amazon Data Firehose New
Managed Apache Flink New
Resources
CloudFormation templates
AWS Glue Schema Registry
Creating data stream weather-stream.
Creating a new data stream can take up to 10 minutes. Learn more
weather-stream Info
Delete
Data stream summary
Status
Creating
Capacity mode
Provisioned
Data retention period
1 day
ARN
arn:aws:kinesis:us-east-1:207399218460:stream/weather-stream
Creation time
July 21, 2025 at 22:09 EDT
Applications
Monitoring
Configuration
Enhanced fan-out (0)
Data viewer
Data analytics - new
Producers Info
Producers put records into Kinesis Data Streams.
Amazon Kinesis Agent
Use a stand-alone Java software application to send data to the stream. Learn more
View in GitHub
AWS SDK
Use AWS SDK for Java to develop producers. Learn more
View in GitHub
Amazon Kinesis Producer Library (KPL)
Use KPL to develop producers. Learn more
View in GitHub
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

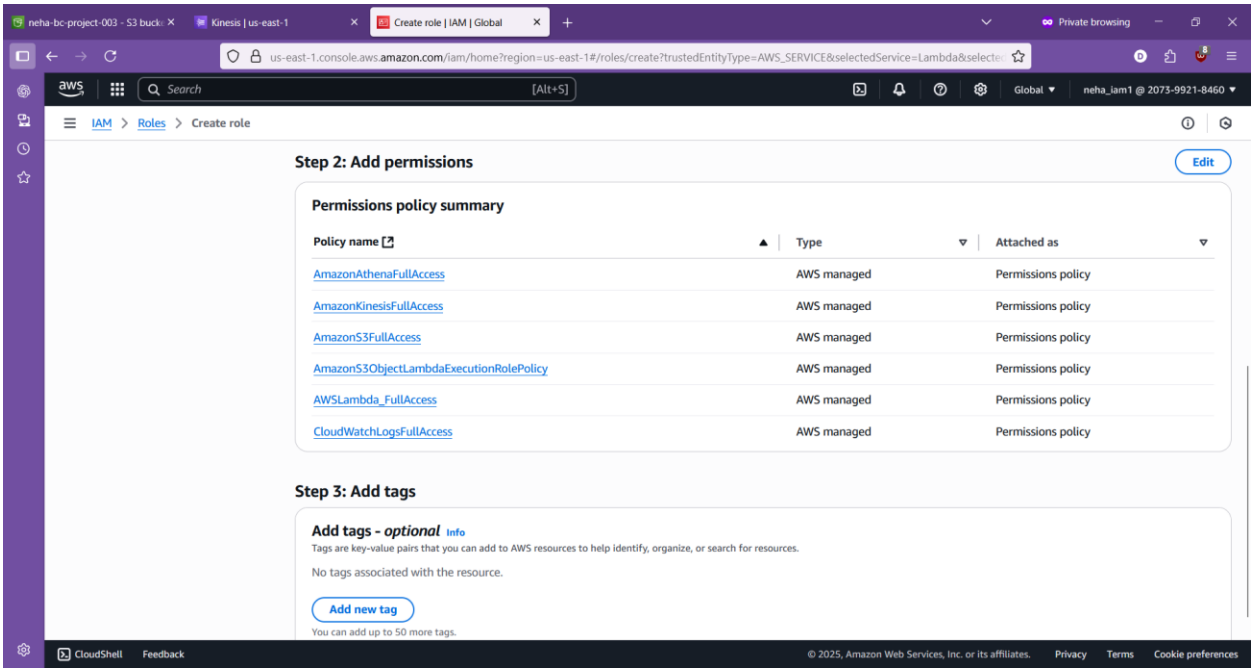


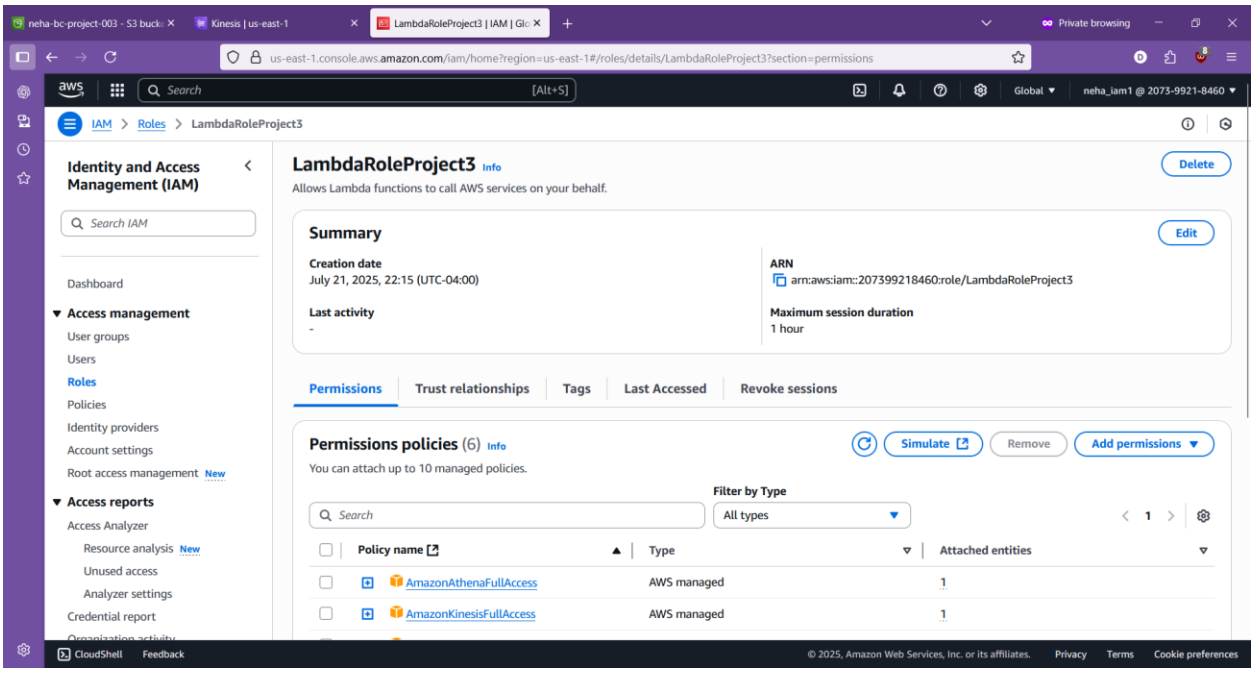


2. Prepare and Run Script to Stream Weather Data

First, create IAM Role with lambda as use case:


Create role | IAM | Global
Private browsing
us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create?trustedEntityType=AWS_SERVICE&selectedService=Lambda&selecte
Search
[Alt+S]
Global
neha_iam1 @ 2073-9921-8460
IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
LambdaRoleProject3
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Allows Lambda functions to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*();:"'`
Step 1: Select trusted entities
Edit
Trust policy
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"sts:AssumeRole"
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Create role | IAM | Global
Private browsing
us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create?trustedEntityType=AWS_SERVICE&selectedService=Lambda&selecte
Search
[Alt+S]
Global
neha_iam1 @ 2073-9921-8460
IAM > Roles > Create role
Step 2: Add permissions
Edit
Permissions policy summary
Policy name
Type
Attached as
AmazonAthenaFullAccess
AWS managed
Permissions policy
AmazonKinesisFullAccess
AWS managed
Permissions policy
AmazonS3FullAccess
AWS managed
Permissions policy
AmazonS3ObjectLambdaExecutionRolePolicy
AWS managed
Permissions policy
AWSLambda_FullAccess
AWS managed
Permissions policy
CloudWatchLogsFullAccess
AWS managed
Permissions policy
Step 3: Add tags
Add tags - optional
Info
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



To run in CloudShell

1. Click the terminal icon on the AWS Console (bottom nav bar).

2. To Open the file:

    a. nano weather_stream-project-3.py

3. Paste the streaming script into a file:

4. To save and exit nano:

    a. Press Ctrl + O to write (save) the file

    b. Press Enter to confirm

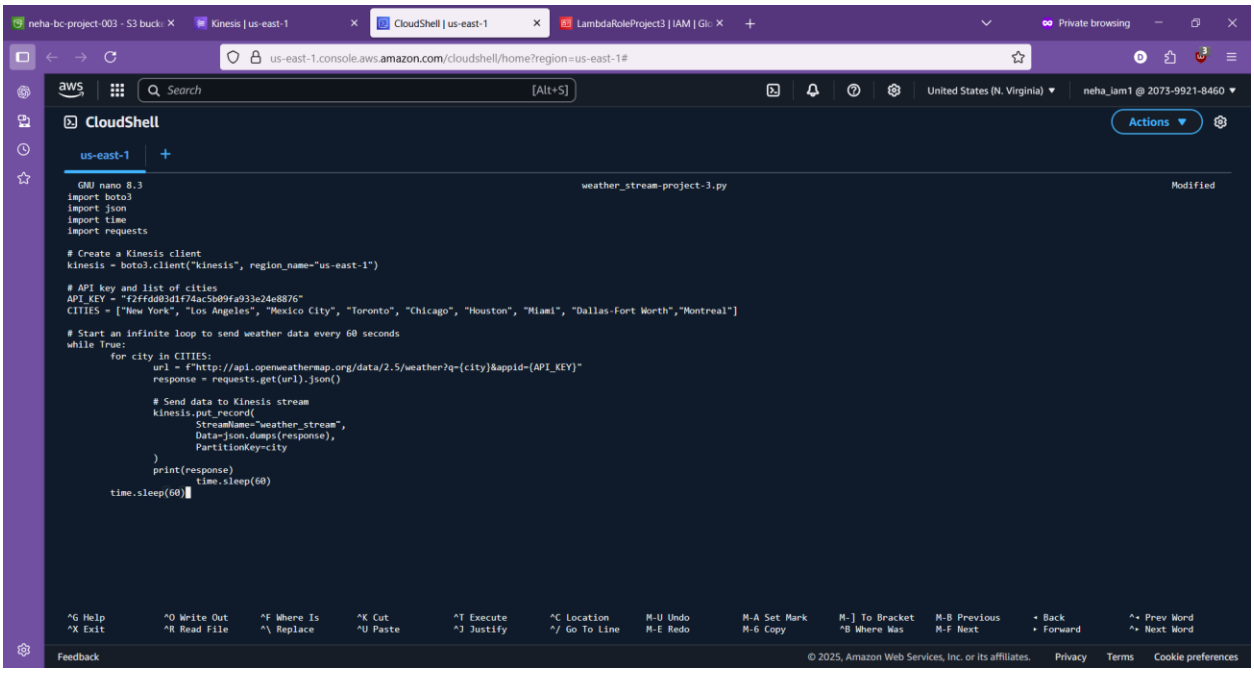
    c. Press Ctrl + X to exit

5. Make it executable:

    a. chmod +x weather_stream-project-3.py

6. Run it:

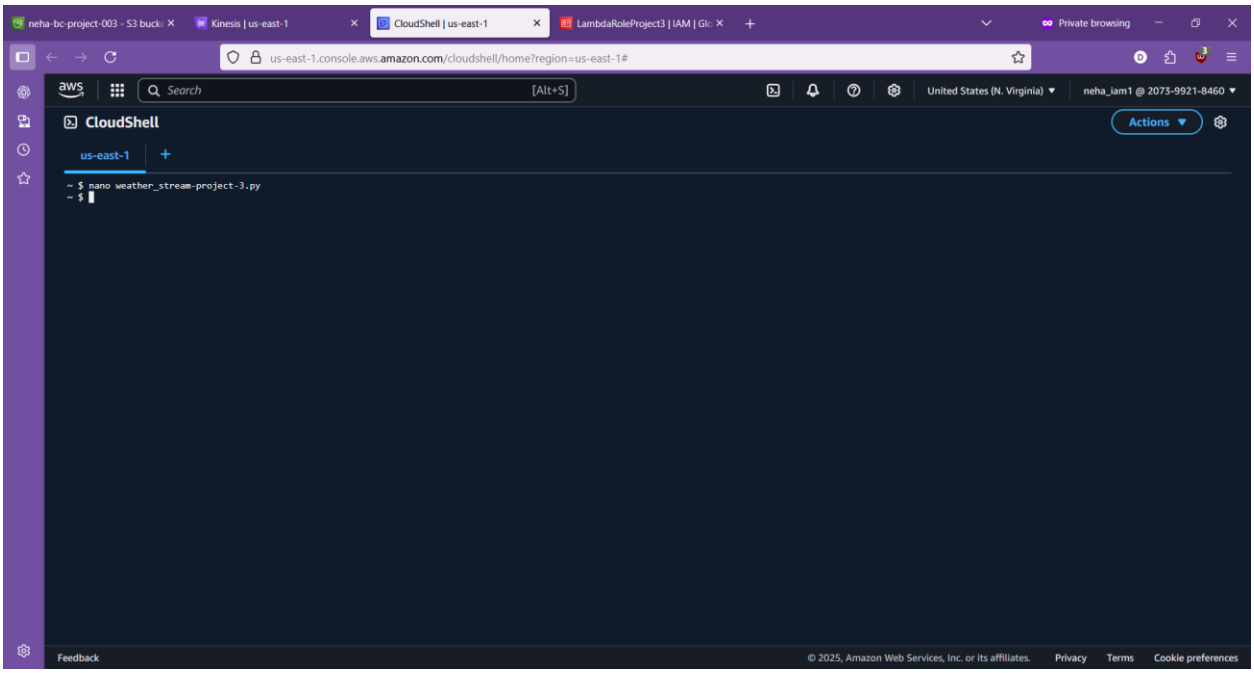
    a. Python3 weather_stream-project-3.py

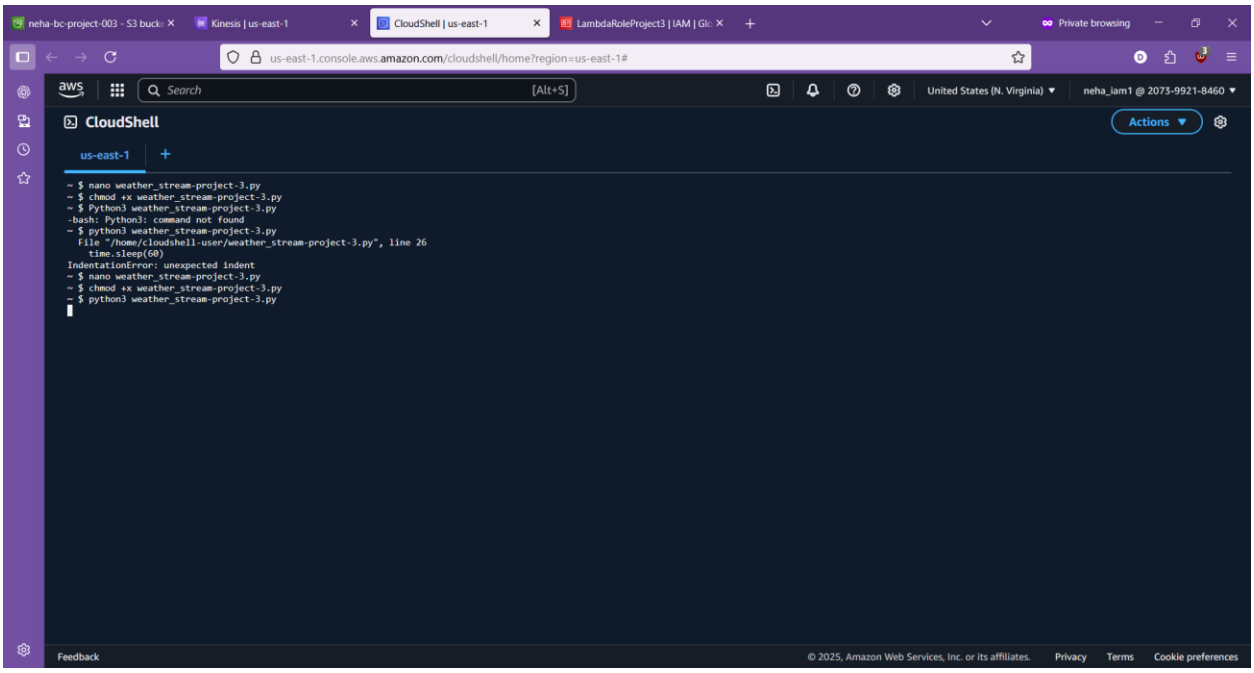


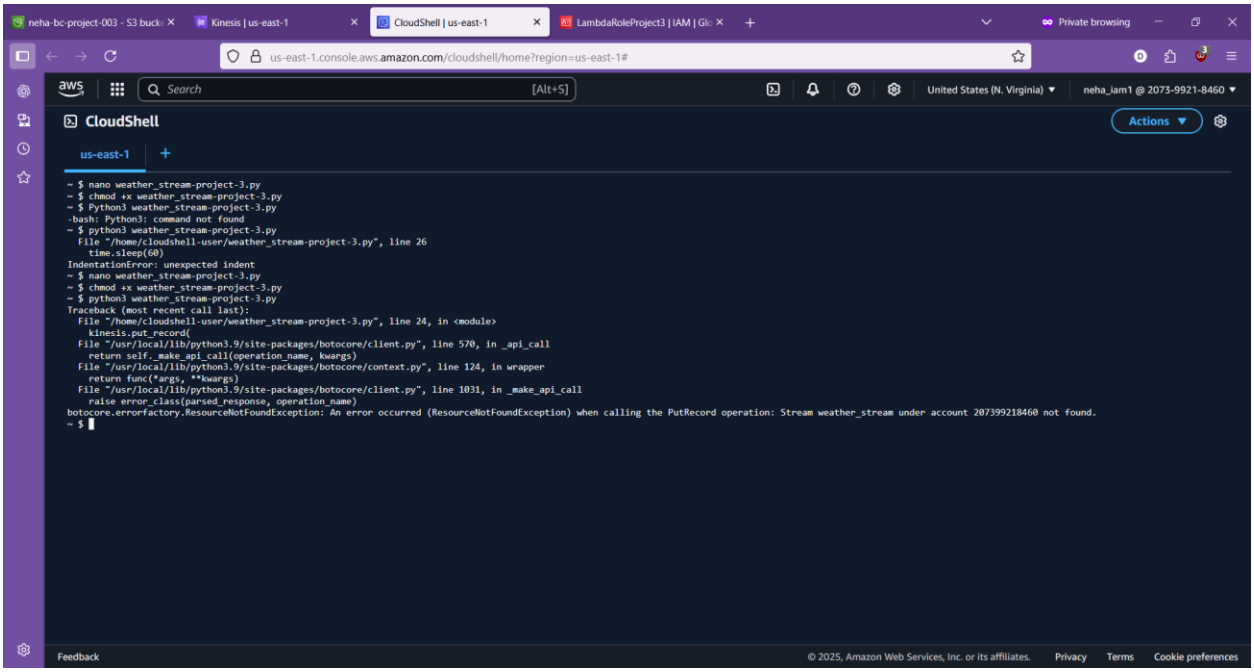
Script:

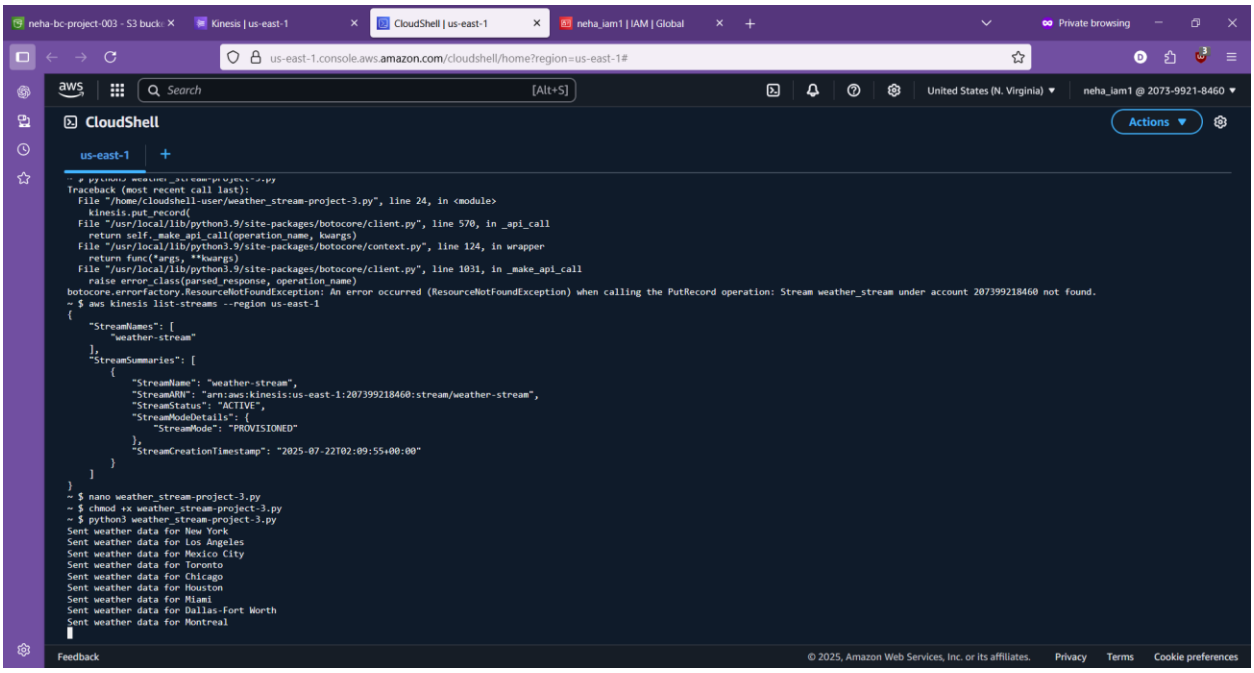
```
import boto3
import json
import time
import requests


# Create a Kinesis client
kinesis = boto3.client("kinesis", region_name="us-east-1")


# API key and list of cities
API_KEY = "f2ffdd03d1f74ac5b09fa933e24e8876"
CITIES = [
    "New York", "Los Angeles", "Mexico City",
    "Toronto", "Chicago", "Houston",
    "Miami", "Dallas-Fort Worth", "Montreal"
]
```

```
# Infinite loop to send weather data every 60 seconds
while True:
    for city in CITIES:
        url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={API_KEY}"
        response = requests.get(url).json()

        # Send data to Kinesis stream
        kinesis.put_record(
            StreamName="weather-stream",
            Data=json.dumps(response),
            PartitionKey=city
        )
        print(f"Sent weather data for {city}")

    # Wait 60 seconds before the next batch
    time.sleep(60)
```



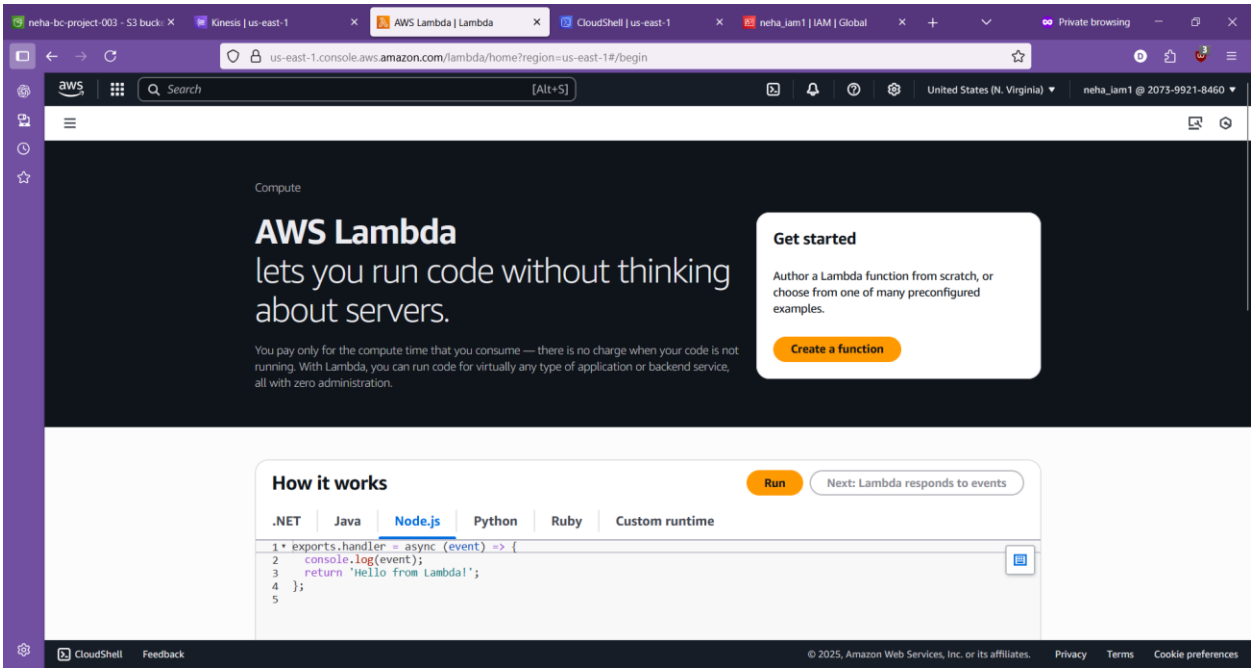

neha-bc-project-003 - S3 buck
Kinesis | us-east-1
CloudShell | us-east-1
LambdaRoleProject3 | IAM | Gl
Private browsing
us-east-1.console.aws.amazon.com/cloudshell/home?region=us-east-1#
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
CloudShell
Actions
us-east-1
~ $ nano weather_stream-project-3.py
~ $ chmod +x weather_stream-project-3.py
~ $ Python3 weather_stream-project-3.py
-bash: Python3: command not found
~ $ python3 weather_stream-project-3.py
File "/home/cloudshell-user/weather_stream-project-3.py", line 26
time.sleep(60)
IndentationError: unexpected indent
~ $ nano weather_stream-project-3.py
~ $ chmod +x weather_stream-project-3.py
~ $ python3 weather_stream-project-3.py
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

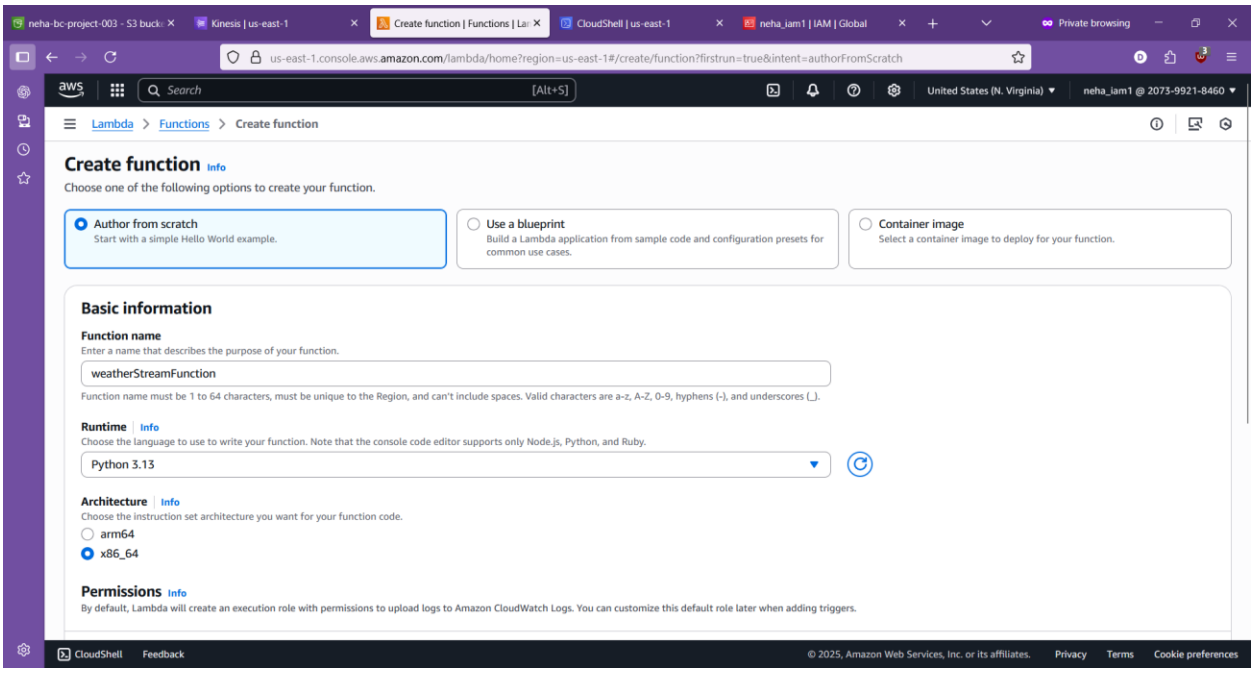

neha-bc-project-003 - S3 buck
Kinesis | us-east-1
CloudShell | us-east-1
LambdaRoleProject3 | IAM | Gl
Private browsing
us-east-1.console.aws.amazon.com/cloudshell/home?region=us-east-1#
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
CloudShell
Actions
us-east-1
~ $ nano weather_stream-project-3.py
~ $ chmod +x weather_stream-project-3.py
~ $ Python3 weather_stream-project-3.py
-bash: Python3: command not found
~ $ python3 weather_stream-project-3.py
File "/home/cloudshell-user/weather_stream-project-3.py", line 26
time.sleep(60)
IndentationError: unexpected indent
~ $ nano weather_stream-project-3.py
~ $ chmod +x weather_stream-project-3.py
~ $ python3 weather_stream-project-3.py
Traceback (most recent call last):
File "/home/cloudshell-user/weather_stream-project-3.py", line 24, in <module>
kinesis.put_record(
File "/usr/local/lib/python3.9/site-packages/botocore/client.py", line 570, in _api_call
return self._make_api_call(operation_name, kwargs)
File "/usr/local/lib/python3.9/site-packages/botocore/context.py", line 124, in wrapper
return func(*args, **kwargs)
File "/usr/local/lib/python3.9/site-packages/botocore/client.py", line 1031, in _make_api_call
raise error_class(parsed_response, operation_name)
botocore.errorfactory.ResourceNotFoundException: An error occurred (ResourceNotFoundException) when calling the PutRecord operation: Stream weather_stream under account 207399218460 not found.
~ $
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

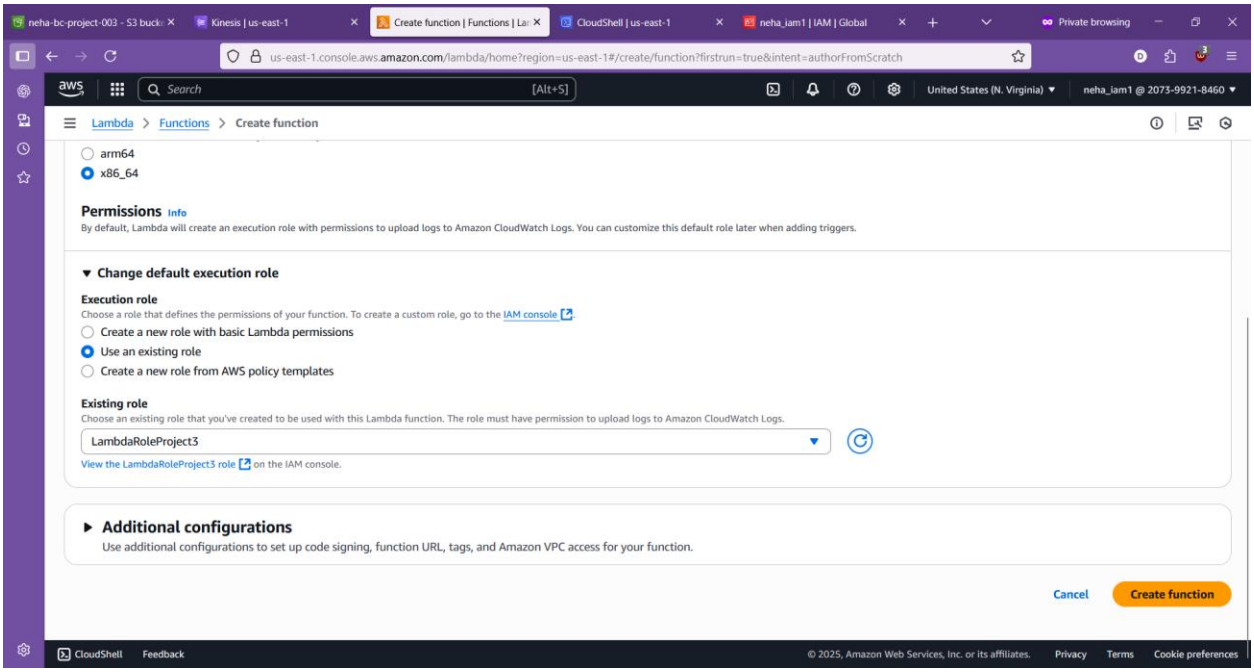


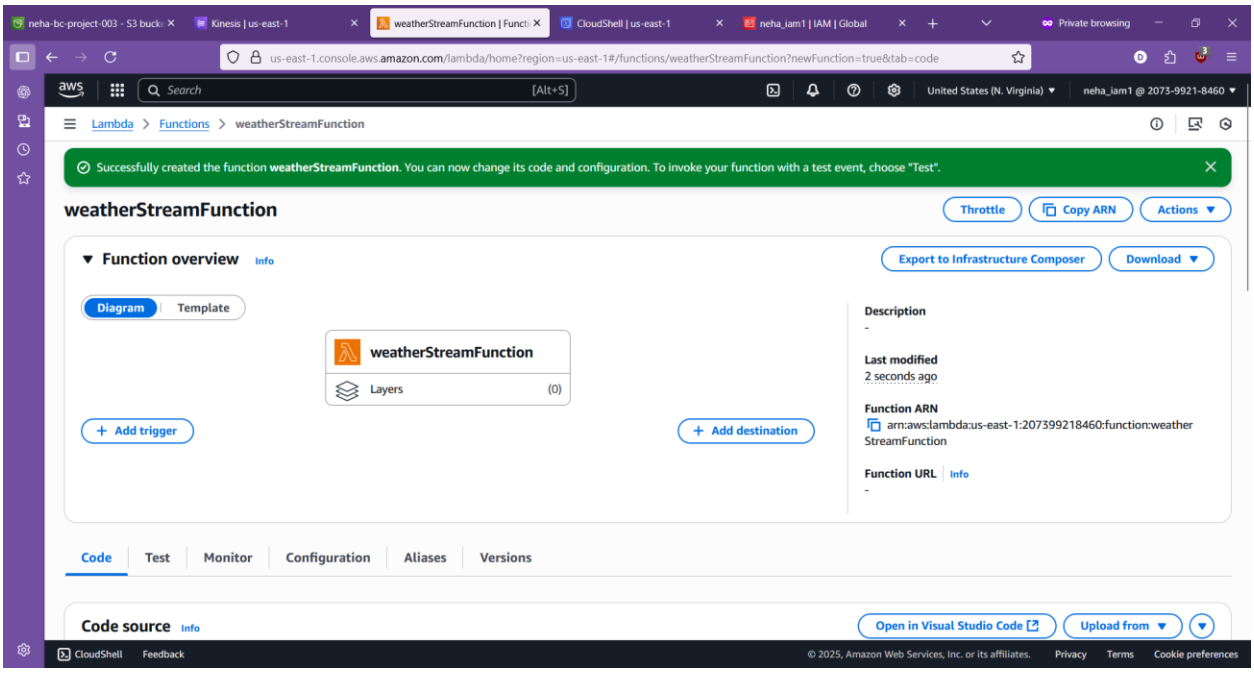
Silver Layer

1. Create Lambda function to handle the weather data




Create function | Functions | La
Kinesis | us-east-1
CloudShell | us-east-1
neha_iam1 | IAM | Global
Private browsing
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function?firstrun=true&intent=authorFromScratch
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
Lambda > Functions > Create function
Create function Info
Choose one of the following options to create your function.
Author from scratch
Start with a simple Hello World example.
Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.
Container image
Select a container image to deploy for your function.
Basic information
Function name
Enter a name that describes the purpose of your function.
weatherStreamFunction
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).
Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.13
Architecture Info
Choose the instruction set architecture you want for your function code.
arm64
x86_64
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

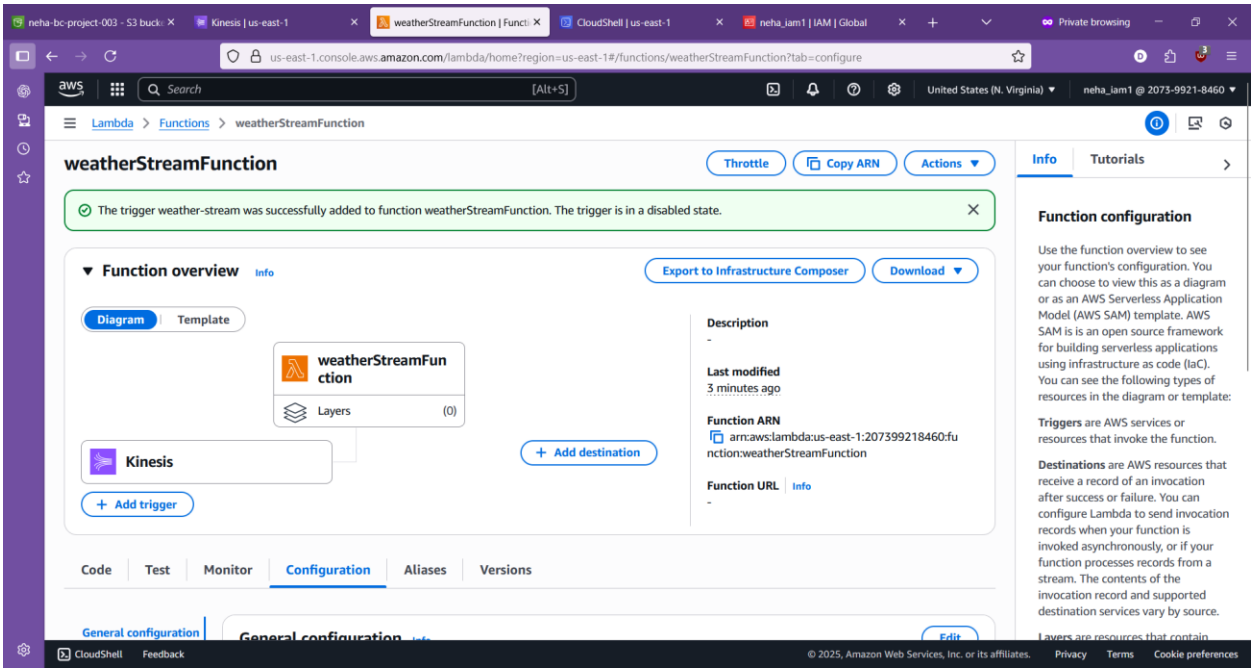

Create function | Functions | La
Kinesis | us-east-1
CloudShell | us-east-1
neha_iam1 | IAM | Global
Private browsing
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function?firstrun=true&intent=authorFromScratch
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
Lambda > Functions > Create function
arm64
x86_64
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.
Change default execution role
Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
Create a new role with basic Lambda permissions
Use an existing role
Create a new role from AWS policy templates
Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
LambdaRoleProject3
View the LambdaRoleProject3 role on the IAM console.
Additional configurations
Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.
Cancel
Create function
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

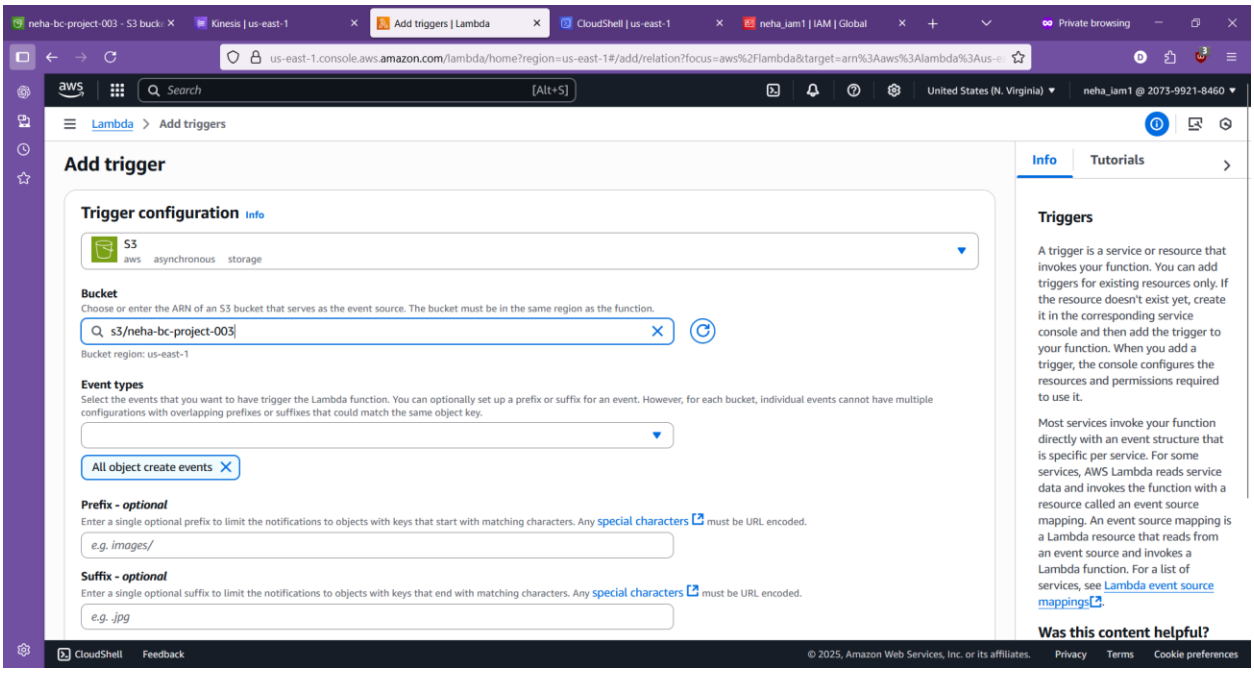


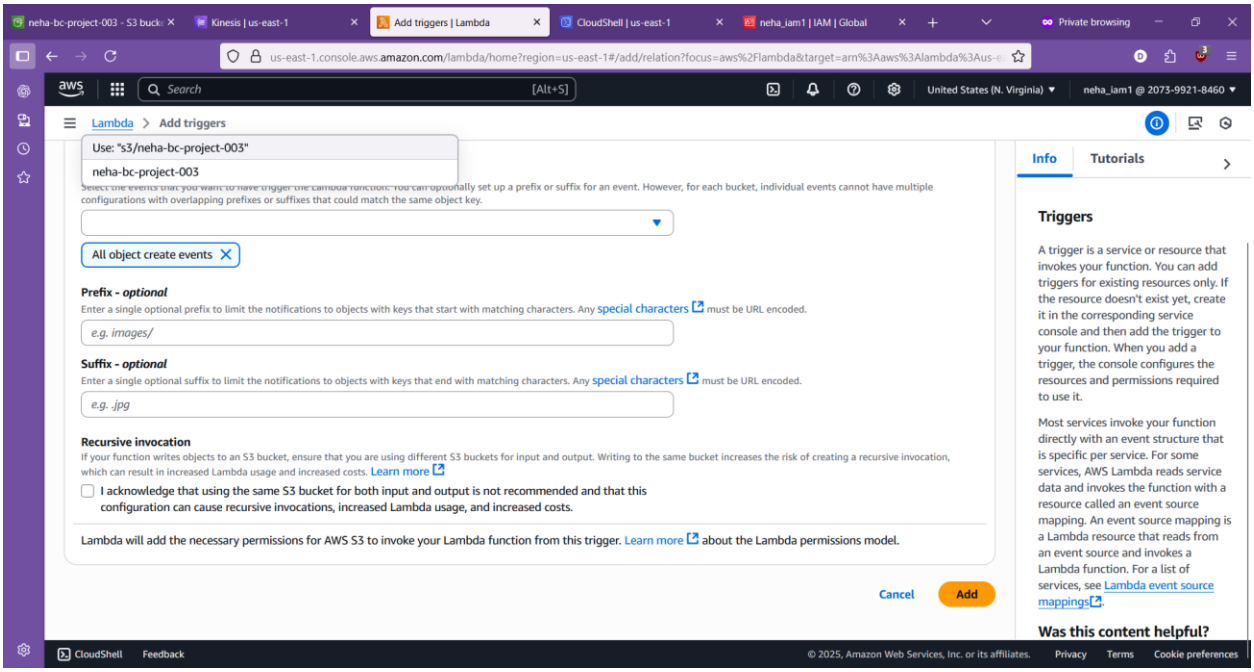
2. Add the following Triggers

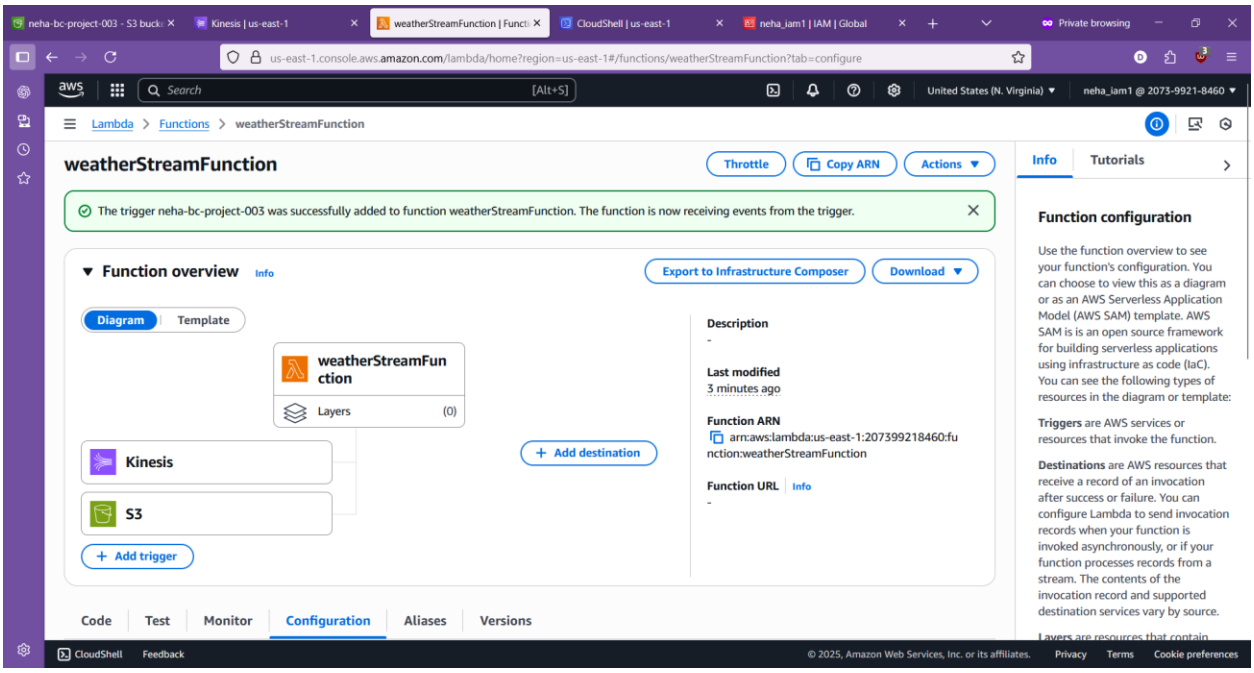
Kinesis - to fetch the real time data from kinesis

S3 - to store the raw (Bronze layer) data and Cleaned/processed (Siler layer) data in S3

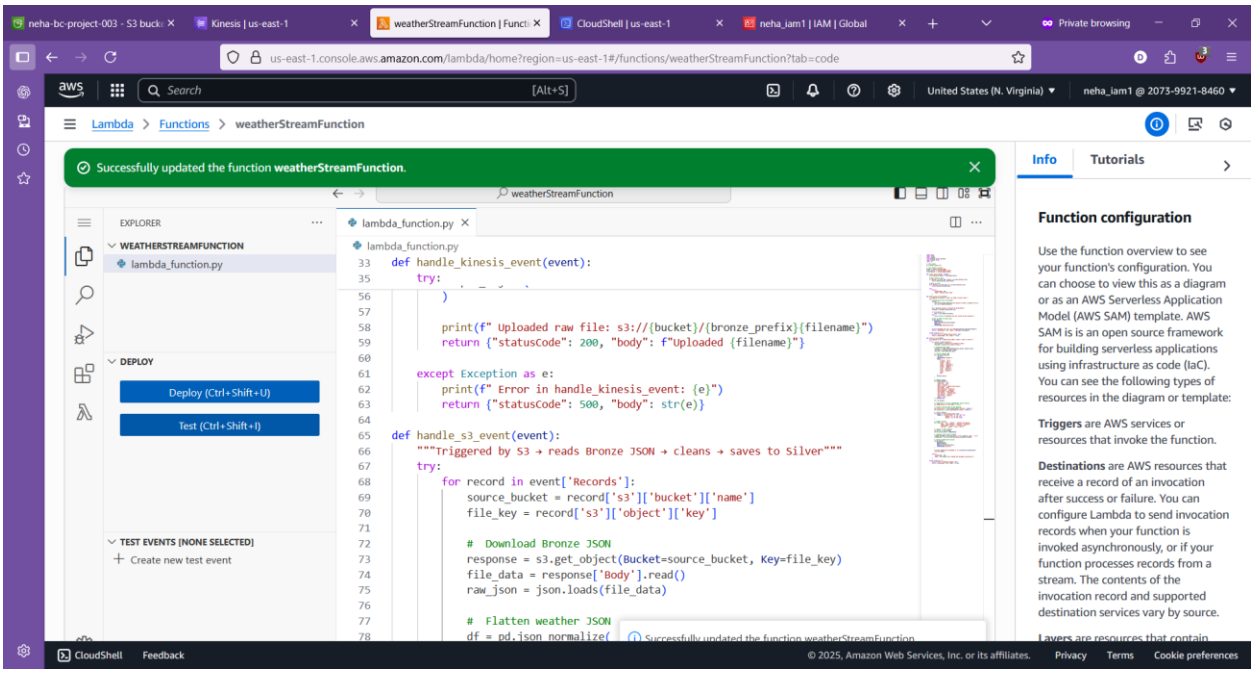



Add triggers | Lambda
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
Lambda > Add triggers
Add trigger
Trigger configuration Info
S3
aws asynchronous storage
Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.
s3/neha-bc-project-003
Bucket region: us-east-1
Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.
All object create events
Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.
e.g. images/
Suffix - optional
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any special characters must be URL encoded.
e.g. .jpg
Info
Tutorials
Triggers
A trigger is a service or resource that invokes your function. You can add triggers for existing resources only. If the resource doesn't exist yet, create it in the corresponding service console and then add the trigger to your function. When you add a trigger, the console configures the resources and permissions required to use it.
Most services invoke your function directly with an event structure that is specific per service. For some services, AWS Lambda reads service data and invokes the function with a resource called an event source mapping. An event source mapping is a Lambda resource that reads from an event source and invokes a Lambda function. For a list of services, see Lambda event source mappings.
Was this content helpful?
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Add triggers | Lambda
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
Lambda > Add triggers
Use: "s3/neha-bc-project-003"
neha-bc-project-003
All object create events
Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.
e.g. images/
Suffix - optional
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any special characters must be URL encoded.
e.g. .jpg
Recursive invocation
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. Learn more
I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.
Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.
Cancel
Add
Info
Tutorials
Triggers
A trigger is a service or resource that invokes your function. You can add triggers for existing resources only. If the resource doesn't exist yet, create it in the corresponding service console and then add the trigger to your function. When you add a trigger, the console configures the resources and permissions required to use it.
Most services invoke your function directly with an event structure that is specific per service. For some services, AWS Lambda reads service data and invokes the function with a resource called an event source mapping. An event source mapping is a Lambda resource that reads from an event source and invokes a Lambda function. For a list of services, see Lambda event source mappings.
Was this content helpful?
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



Paste the lambda function code in the “code” tab and Click “deploy”



Code:

import json

import boto3

import base64

from datetime import datetime

```
import pandas as pd
import io

# AWS clients
s3 = boto3.client('s3')

# bucket & prefixes
bucket = "neha-bc-project-003"
bronze_prefix = "bronze/weather_data/"
silver_prefix = "silver/weather_data/"

def lambda_handler(event, context):
    print("Received event:", json.dumps(event))

    # Detect Kinesis event
    if 'Records' in event and 'kinesis' in event['Records'][0]:
        return handle_kinesis_event(event)

    # Detect S3 event
    elif 'Records' in event and 's3' in event['Records'][0]:
        return handle_s3_event(event)

    else:
        return {
            "statusCode": 400,
            "body": "Unknown event type"
        }

def handle_kinesis_event(event):
```

```
"""Triggered by Kinesis : stores raw JSON in Bronze layer (handles multiple records)"""
try:
    # Decode all records in the batch
    records = [
        json.loads(base64.b64decode(rec['kinesis']['data']).decode('utf-8'))
        for rec in event['Records']
    ]

    now = datetime.utcnow().strftime("%Y-%m-%d-%H-%M")
    filename = f"weather_data_{now}.json"

    # Save as a JSON array
    body = json.dumps(records, indent=2)

    # Upload to Bronze layer
    s3.put_object(
        Bucket=bucket,
        Key=f"{bronze_prefix}{filename}",
        Body=body,
        ContentType='application/json'
    )

    print(f"Uploaded {len(records)} records to s3://{bucket}/{bronze_prefix}{filename}")
    return {"statusCode": 200, "body": f"Uploaded {len(records)} records"}

except Exception as e:
    print(f"Error in handle_kinesis_event: {e}")
    return {"statusCode": 500, "body": str(e)}
```

```
def handle_s3_event(event):
    """Triggered by S3 : reads Bronze JSON → cleans → saves to Silver"""
    try:
        for record in event['Records']:
            source_bucket = record['s3']['bucket']['name']
            file_key = record['s3']['object']['key']

            # Download Bronze JSON
            response = s3.get_object(Bucket=source_bucket, Key=file_key)
            file_data = response['Body'].read().decode('utf-8').strip()

            # Skip if Bronze file is empty
            if not file_data:
                print(f"Skipping empty Bronze file: {file_key}")
                continue

            # Try parsing JSON, skip if invalid
            try:
                raw_json = json.loads(file_data)
            except json.JSONDecodeError as e:
                print(f"Skipping invalid JSON in {file_key} -> {e}")
                continue

            # If it's a batch JSON array, flatten each weather object
            if isinstance(raw_json, list):
                combined_df = pd.DataFrame()
                for weather_entry in raw_json:
                    df = flatten_weather_json(weather_entry)
                    if not df.empty:
```

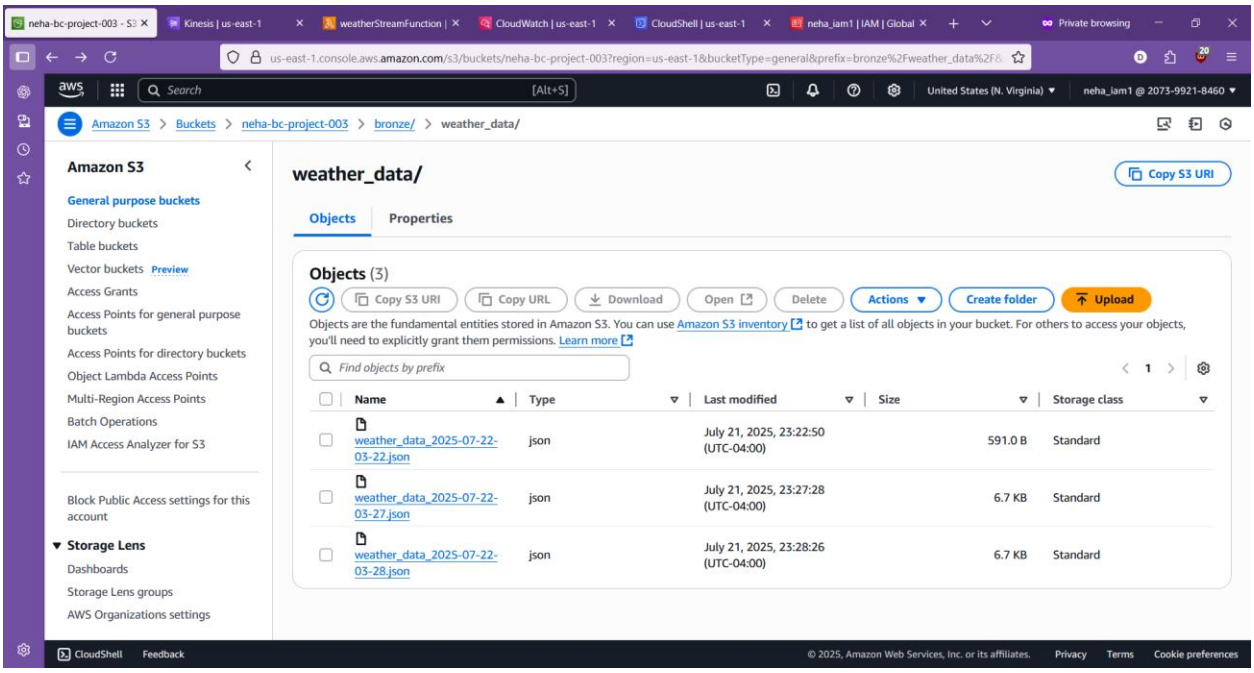
```
            combined_df = pd.concat([combined_df, df], ignore_index=True)
    else:
        df = flatten_weather_json(raw_json)
        combined_df = df if not df.empty else pd.DataFrame()

    # If no valid records, skip saving
    if combined_df.empty:
        print(f"No valid weather data found in {file_key}. Skipping upload.")
        continue

    # Convert to CSV in-memory
    out_buffer = io.StringIO()
    combined_df.to_csv(out_buffer, index=False)

    # Generate Silver-layer filename
    cleaned_filename = file_key.split('/')[-1].replace('.json', '.csv')
    cleaned_key = f"{silver_prefix}cleaned_{cleaned_filename}"

    # Upload cleaned CSV to Silver
    s3.put_object(
        Bucket=bucket,
        Key=cleaned_key,
        Body=out_buffer.getvalue(),
        ContentType='text/csv'
    )

    print(f"Cleaned file uploaded to: s3://{bucket}/{cleaned_key}")
    print(combined_df.head())
```

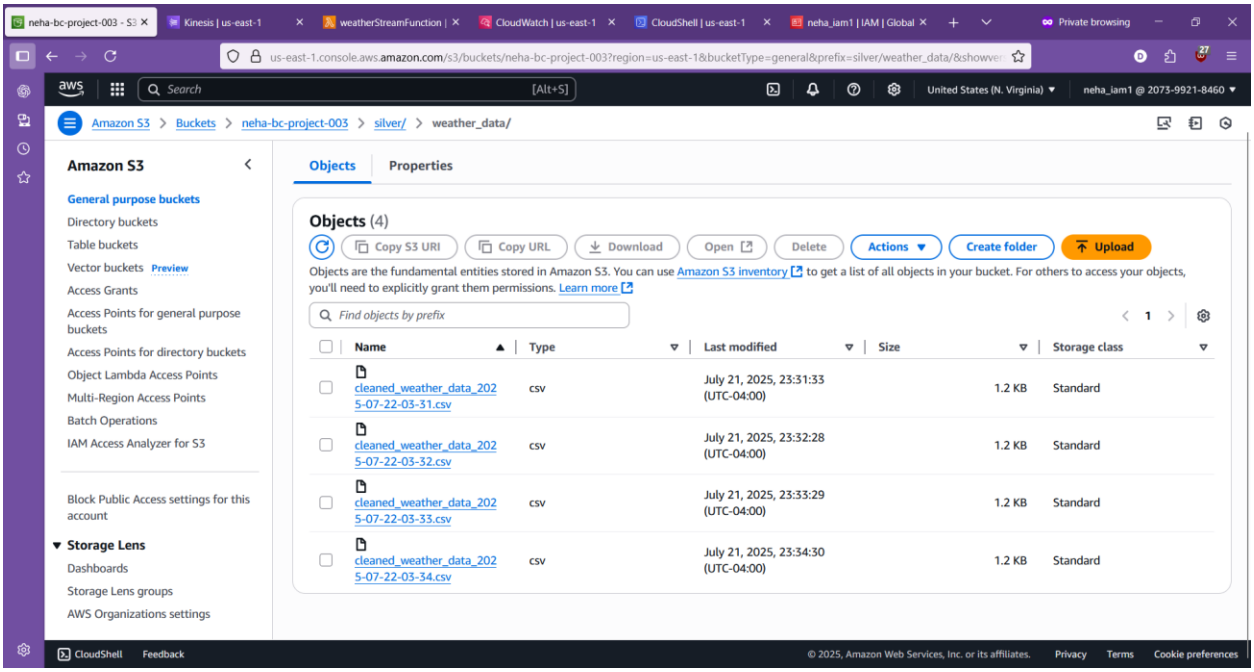
```
        return {
            'statusCode': 200,
            'body': json.dumps('File cleaned and uploaded successfully!')
        }

    except Exception as e:
        print(f"Error in handle_s3_event: {e}")
        return {"statusCode": 500, "body": str(e)}

def flatten_weather_json(raw_json):
    """Helper: Flattens a single weather JSON record into a DataFrame"""

    # Skip if record does not have weather data
    if not isinstance(raw_json, dict) or 'weather' not in raw_json:
        print("Skipping record without 'weather' key:", raw_json)
        return pd.DataFrame()

    df = pd.json_normalize(
        raw_json,
        record_path=['weather'],
        meta=[
            ['coord', 'lon'],
            ['coord', 'lat'],
            ['main', 'temp'],
            ['main', 'pressure'],
            ['main', 'humidity'],
            ['wind', 'speed'],
            ['sys', 'country'],
            ['sys', 'sunrise'],
```

```
        ['sys', 'sunset'],
        'name',
        'dt'
    ],
    errors='ignore'
)

# Rename columns
df.rename(columns={
    'coord.lon': 'lon',
    'coord.lat': 'lat',
    'main': 'weather',
    'description': 'weather_description',
    'main.temp': 'temp',
    'main.pressure': 'pressure',
    'main.humidity': 'humidity',
    'wind.speed': 'wind_speed',
    'sys.country': 'country_code',
    'sys.sunrise': 'sunrise',
    'sys.sunset': 'sunset',
    'name': 'city'
}, inplace=True)

if df.empty:
    return df

# Temperature in Celsius (OpenWeather returns Kelvin)
df['temp_celsius'] = df['temp'] - 273.15
```

```
# Convert sunrise/sunset to UTC datetime
df['sunrise_utc'] = pd.to_datetime(df['sunrise'], unit='s')
df['sunset_utc'] = pd.to_datetime(df['sunset'], unit='s')

# Map only US/Canada/Mexico
df['country'] = df['country_code'].map(
    lambda x: 'United States' if x == 'US' else
            'Canada' if x == 'CA' else
            'Mexico' if x == 'MX' else x
)

# Reorder columns
df = df[['lon', 'lat', 'weather', 'weather_description',
        'temp', 'temp_celsius', 'pressure', 'humidity',
        'wind_speed', 'country', 'country_code',
        'sunrise', 'sunset', 'city', 'dt']]

return df
```

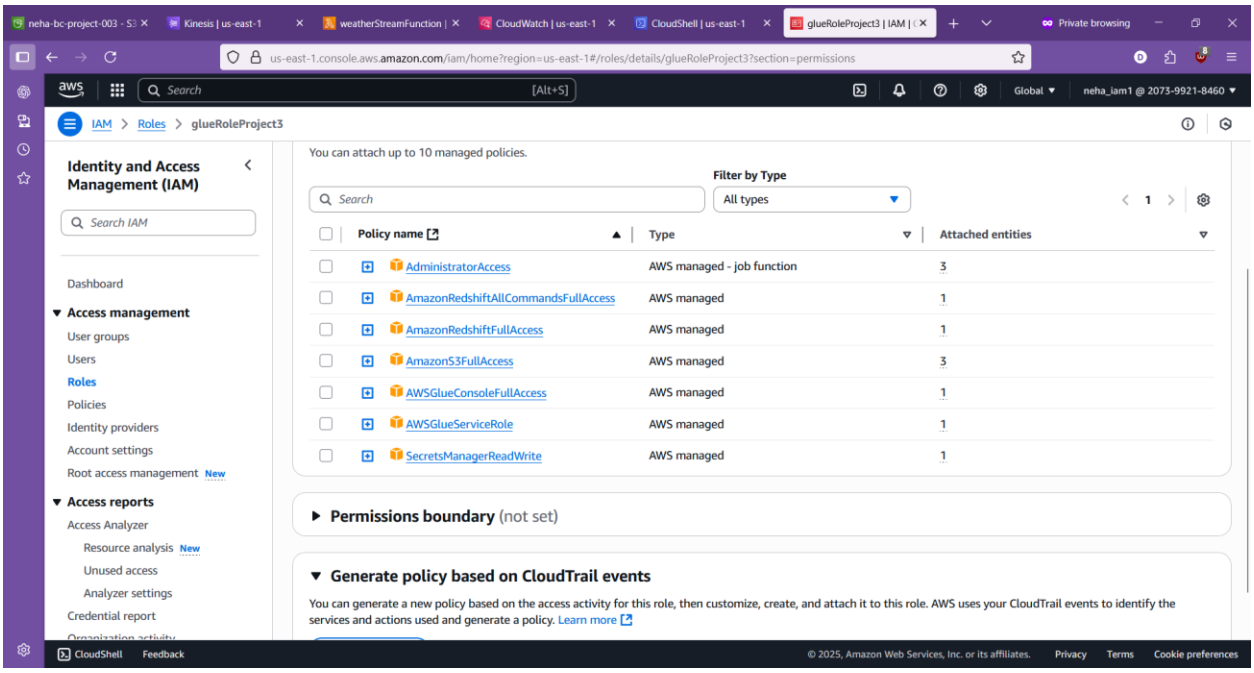
3. Check in S3 Bronze and Silver Layer folder, files are created.



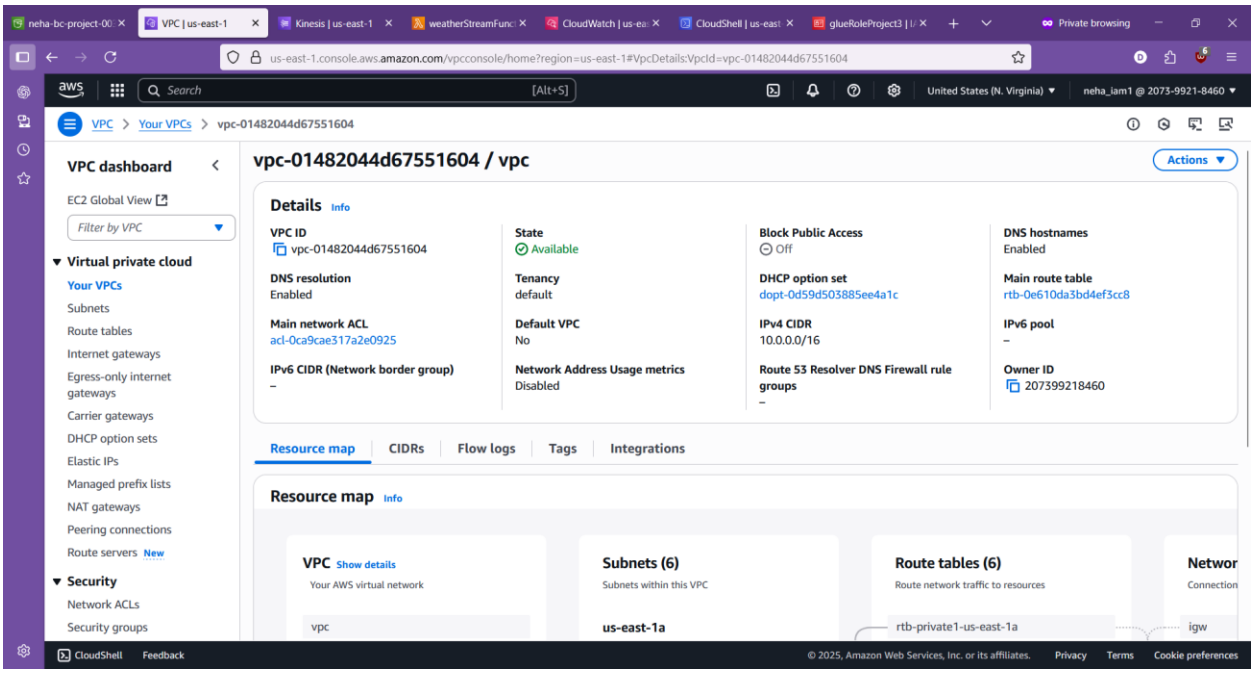


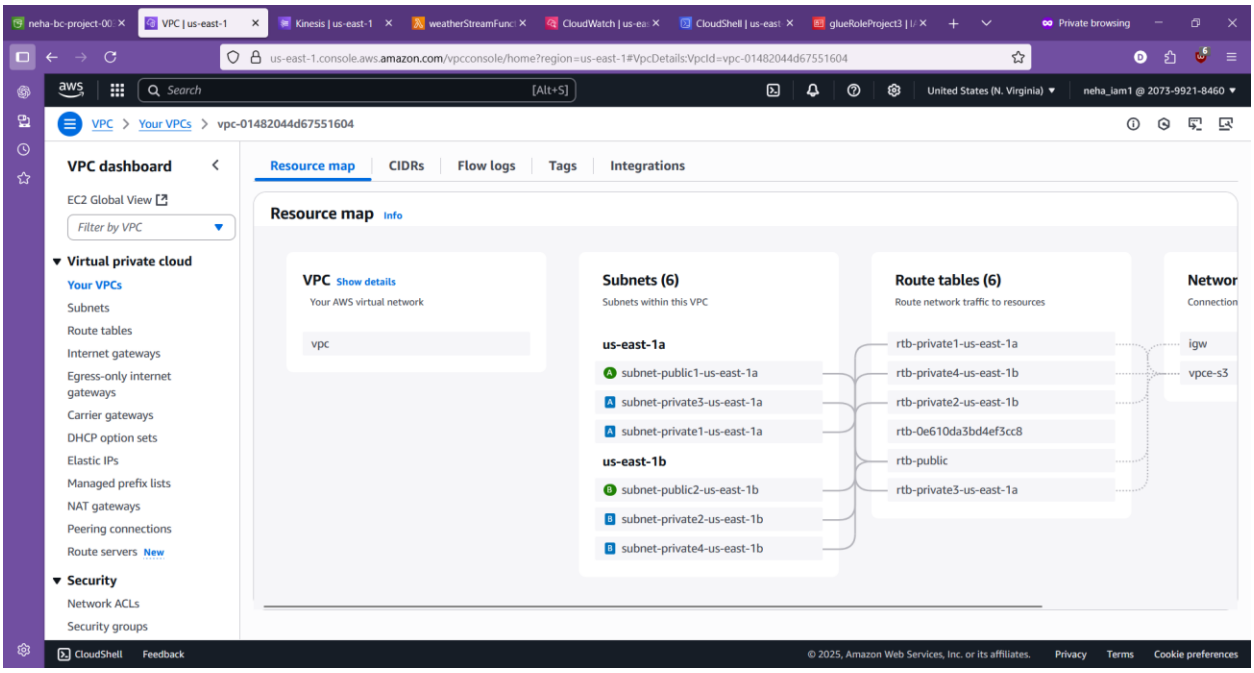
4. Connect to Redshift database

IAM role

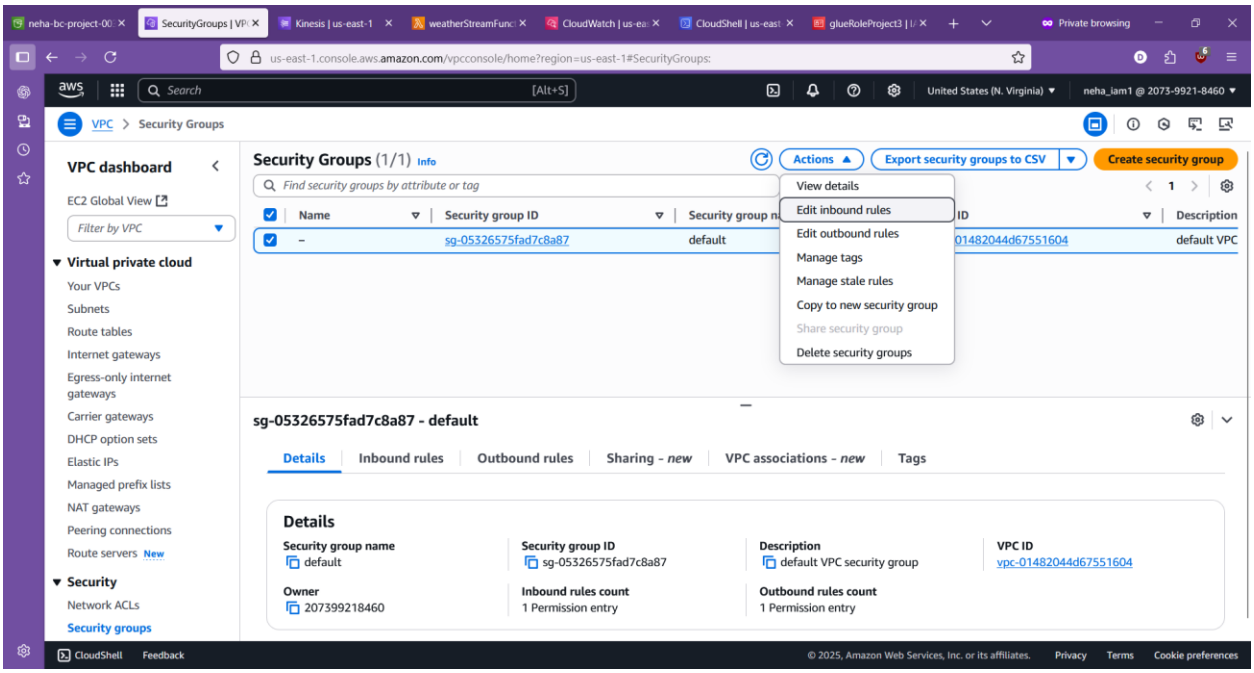

us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/details/glueRoleProject3?section=permissions
IAM > Roles > glueRoleProject3
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Access reports
Access Analyzer
Resource analysis New
Unused access
Analyzer settings
Credential report
You can attach up to 10 managed policies.
Filter by Type
All types
Policy name
Type
Attached entities
AdministratorAccess
AWS managed - job function
AmazonRedshiftAllCommandsFullAccess
AWS managed
AmazonRedshiftFullAccess
AWS managed
AmazonS3FullAccess
AWS managed
AWSGlueConsoleFullAccess
AWS managed
AWSGlueServiceRole
AWS managed
SecretsManagerReadWrite
AWS managed
Permissions boundary (not set)
Generate policy based on CloudTrail events
You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

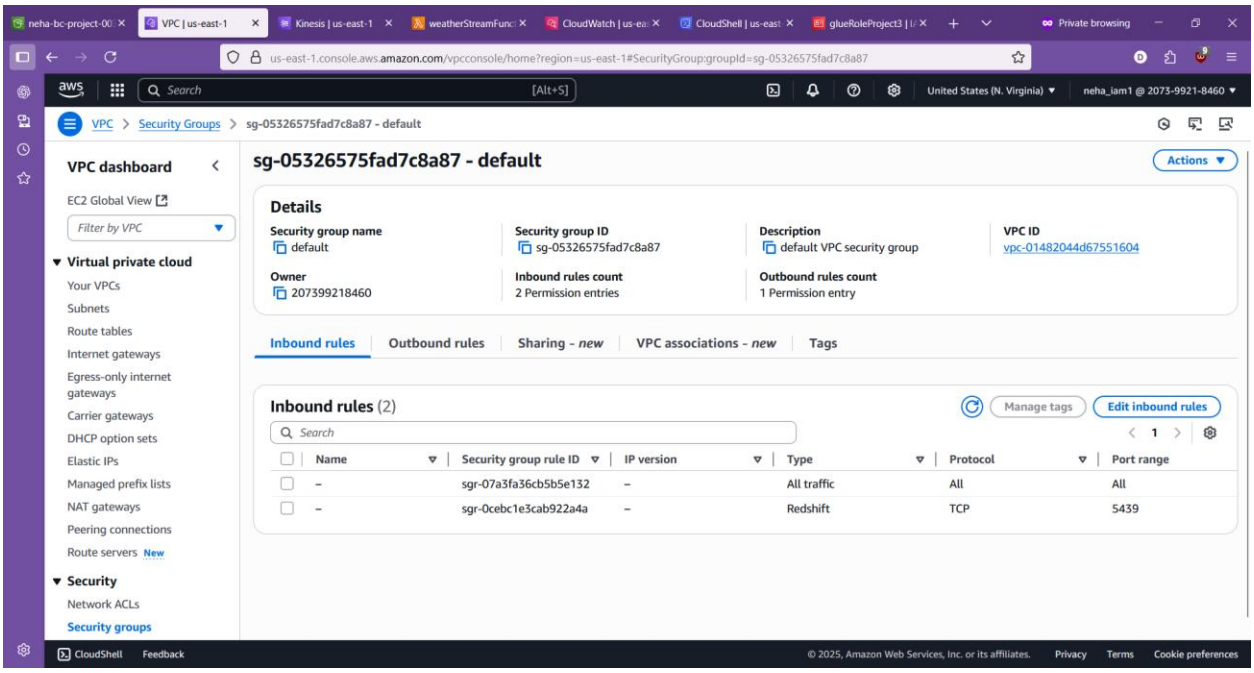
5. Create VPC


us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#VpcDetails:VpcId=vpc-01482044d67551604
VPC > Your VPCs > vpc-01482044d67551604
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route servers New
Security
Network ACLs
Security groups
vpc-01482044d67551604 / vpc
Actions
Details Info
VPC ID
vpc-01482044d67551604
State
Available
Block Public Access
Off
DNS hostnames
Enabled
DNS resolution
Enabled
Tenancy
default
DHCP option set
dopt-0d59d503885ee4a1c
Main route table
rtb-0e610da3bd4ef3cc8
Main network ACL
acl-0ca9cae317a2e0925
Default VPC
No
IPv4 CIDR
10.0.0.0/16
IPv6 pool
–
IPv6 CIDR (Network border group)
–
Network Address Usage metrics
Disabled
Route 53 Resolver DNS Firewall rule groups
–
Owner ID
207399218460
Resource map
CIDRs
Flow logs
Tags
Integrations
Resource map Info
VPC Show details
Your AWS virtual network
vpc
Subnets (6)
Subnets within this VPC
us-east-1a
Route tables (6)
Route network traffic to resources
rtb-private1-us-east-1a
igw
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

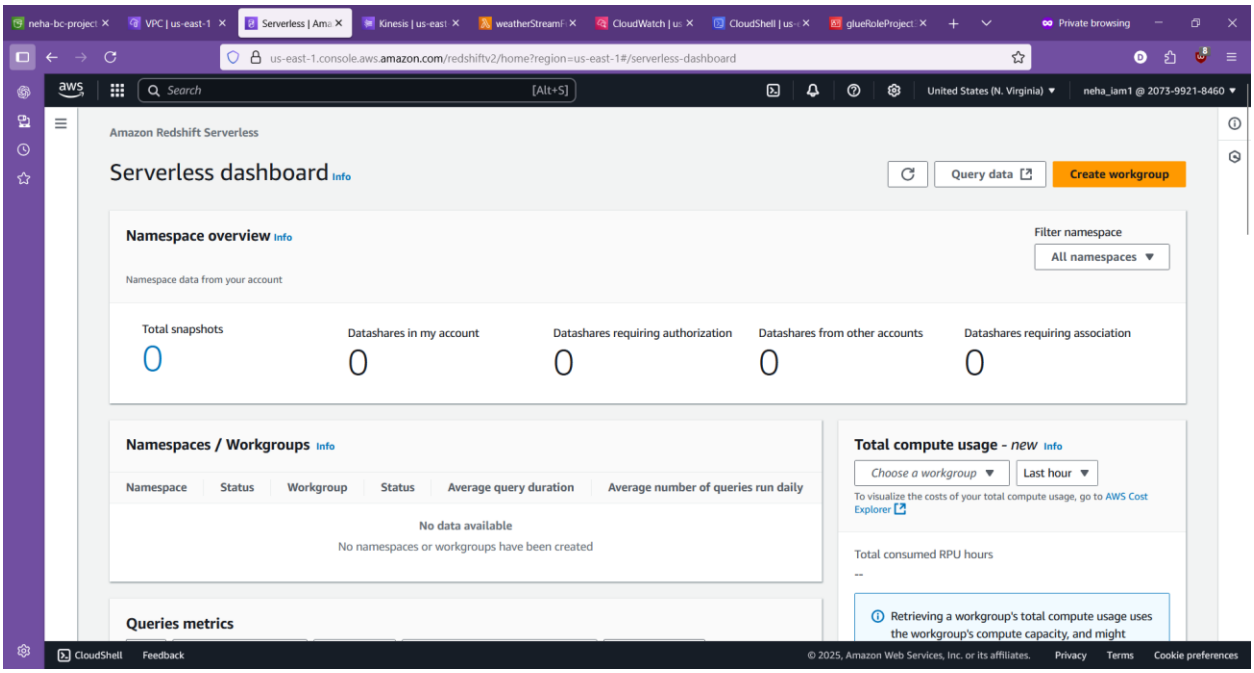


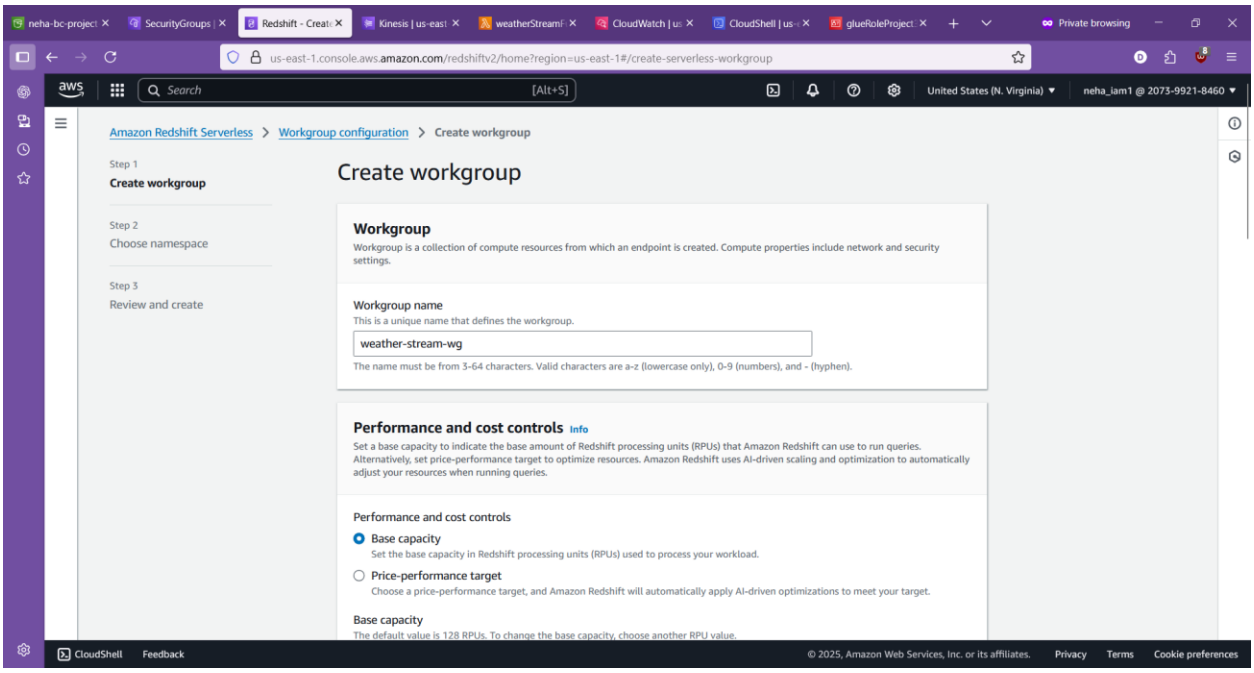
6. Configure Security group

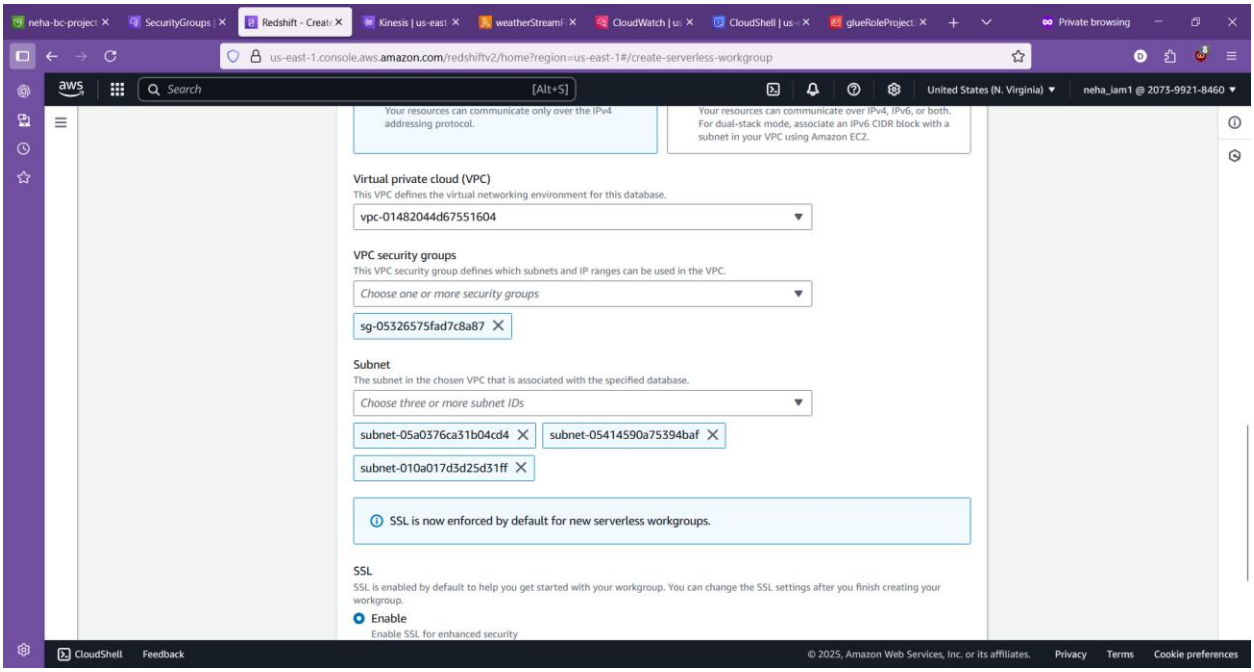


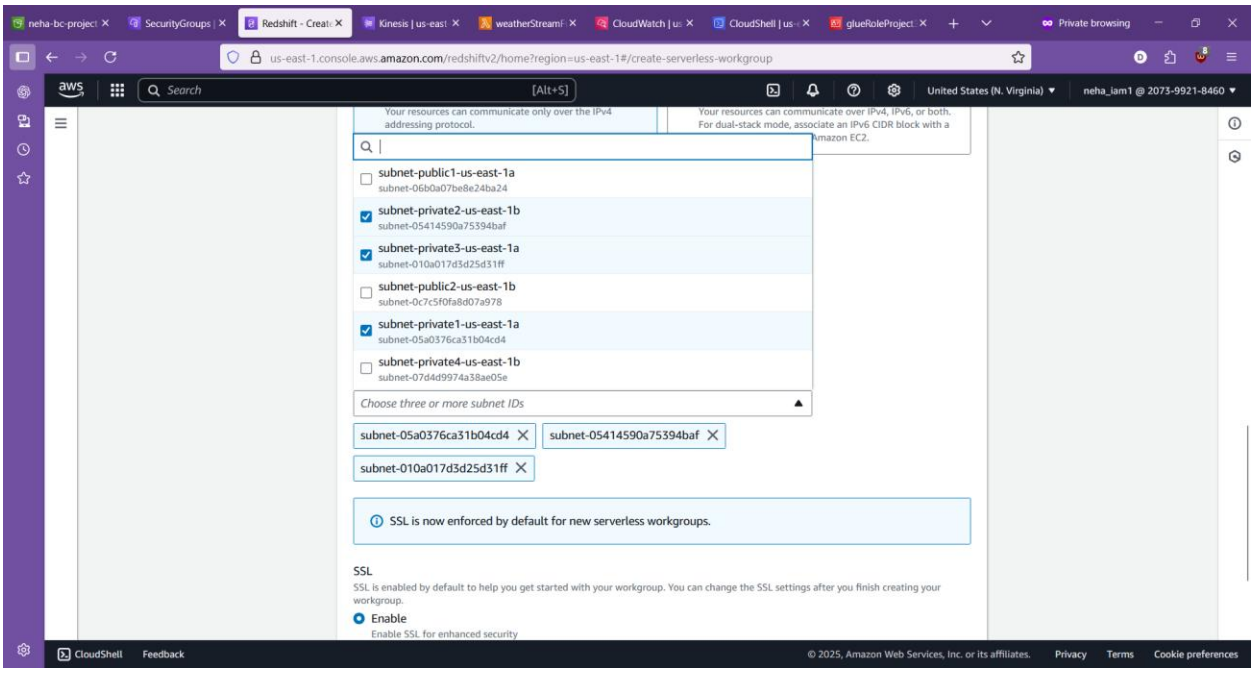


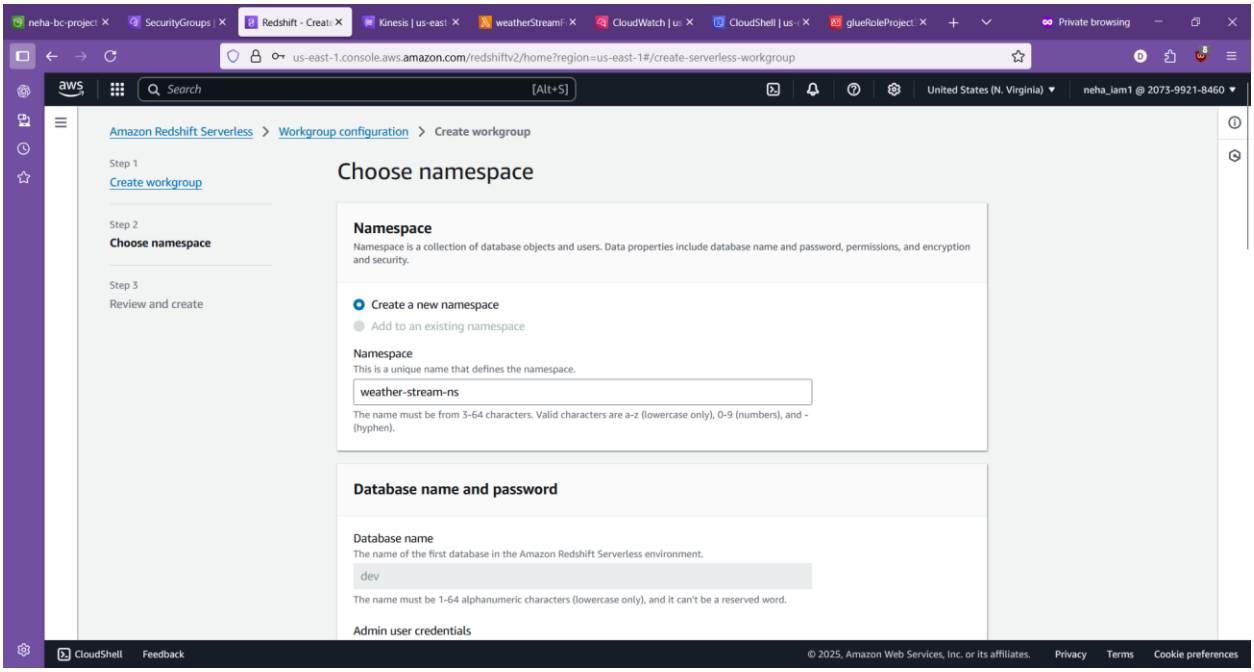
7. Create Redshift workgroup configuration

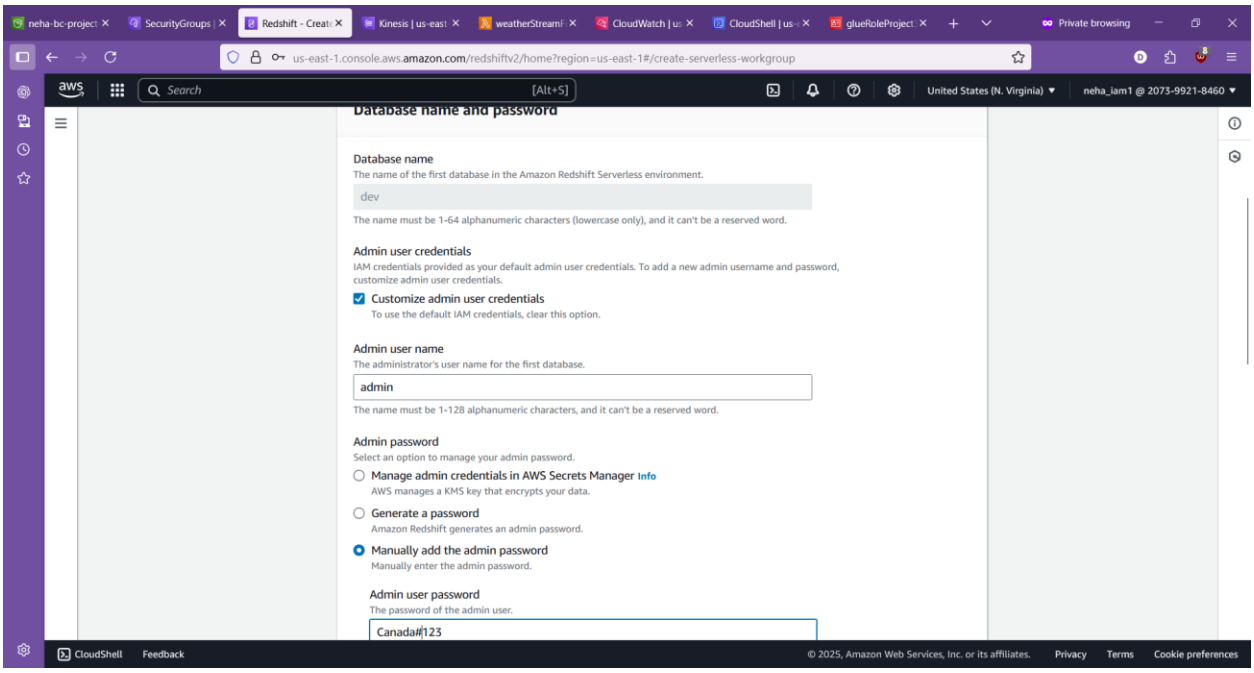


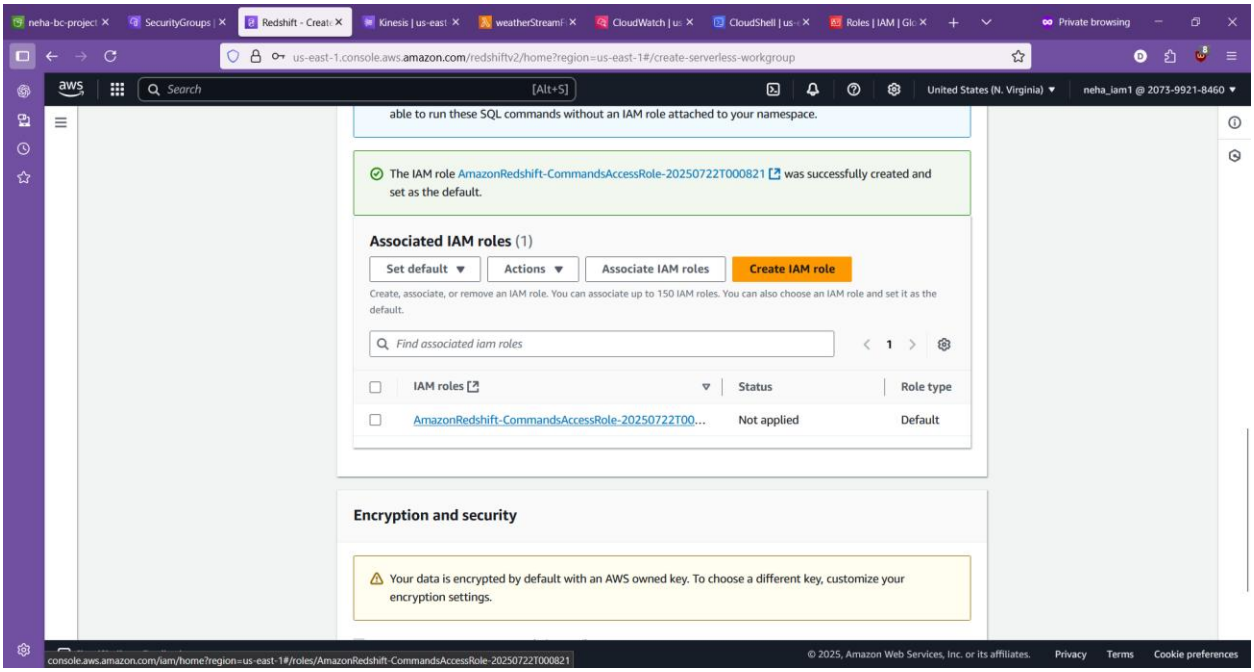

Private browsing
us-east-1.console.aws.amazon.com/redshiftv2/home?region=us-east-1#/create-serverless-workgroup
Search [Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
Amazon Redshift Serverless > Workgroup configuration > Create workgroup
Step 1
Create workgroup
Step 2
Choose namespace
Step 3
Review and create
Create workgroup
Workgroup
Workgroup is a collection of compute resources from which an endpoint is created. Compute properties include network and security settings.
Workgroup name
This is a unique name that defines the workgroup.
weather-stream-wg
The name must be from 3-64 characters. Valid characters are a-z (lowercase only), 0-9 (numbers), and - (hyphen).
Performance and cost controls Info
Set a base capacity to indicate the base amount of Redshift processing units (RPUs) that Amazon Redshift can use to run queries. Alternatively, set price-performance target to optimize resources. Amazon Redshift uses AI-driven scaling and optimization to automatically adjust your resources when running queries.
Performance and cost controls
Base capacity
Set the base capacity in Redshift processing units (RPUs) used to process your workload.
Price-performance target
Choose a price-performance target, and Amazon Redshift will automatically apply AI-driven optimizations to meet your target.
Base capacity
The default value is 128 RPUs. To change the base capacity, choose another RPU value.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

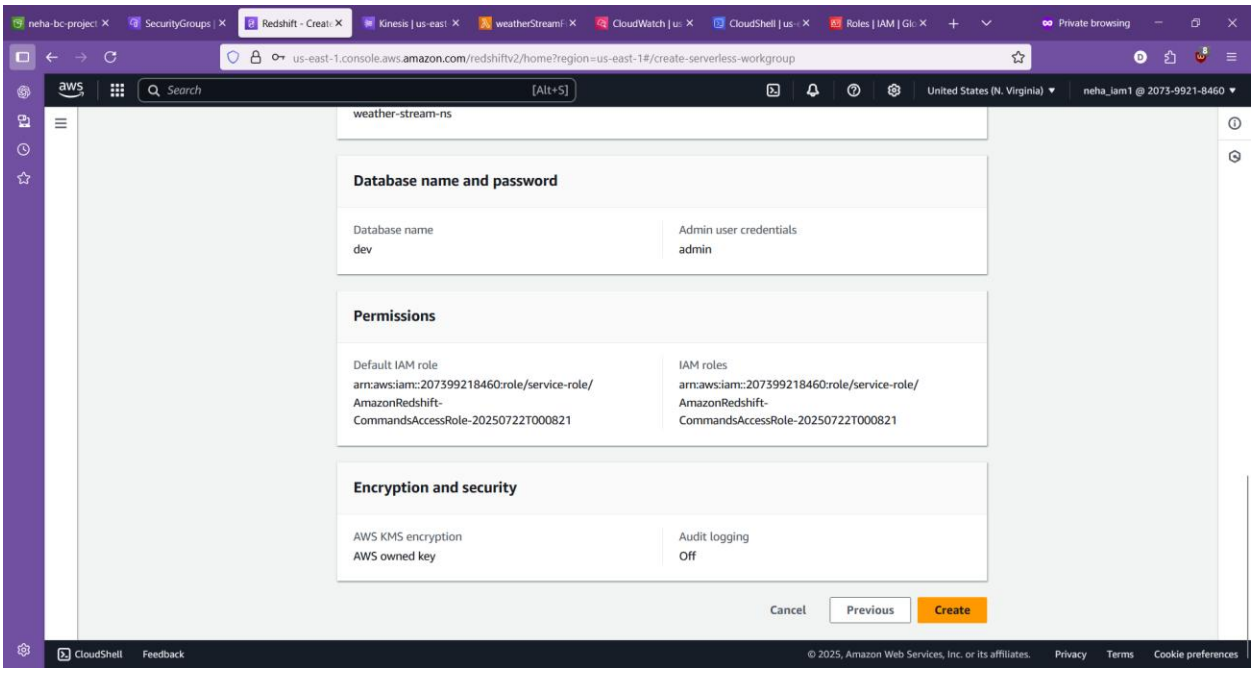

Private browsing
us-east-1.console.aws.amazon.com/redshiftv2/home?region=us-east-1#/create-serverless-workgroup
Search [Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
Your resources can communicate only over the IPv4 addressing protocol.
Your resources can communicate over IPv4, IPv6, or both. For dual-stack mode, associate an IPv6 CIDR block with a subnet in your VPC using Amazon EC2.
Virtual private cloud (VPC)
This VPC defines the virtual networking environment for this database.
vpc-01482044d67551604
VPC security groups
This VPC security group defines which subnets and IP ranges can be used in the VPC.
Choose one or more security groups
sg-05326575fad7c8a87
Subnet
The subnet in the chosen VPC that is associated with the specified database.
Choose three or more subnet IDs
subnet-05a0376ca31b04cd4
subnet-05414590a75394baf
subnet-010a017d3d25d31ff
SSL is now enforced by default for new serverless workgroups.
SSL
SSL is enabled by default to help you get started with your workgroup. You can change the SSL settings after you finish creating your workgroup.
Enable
Enable SSL for enhanced security
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

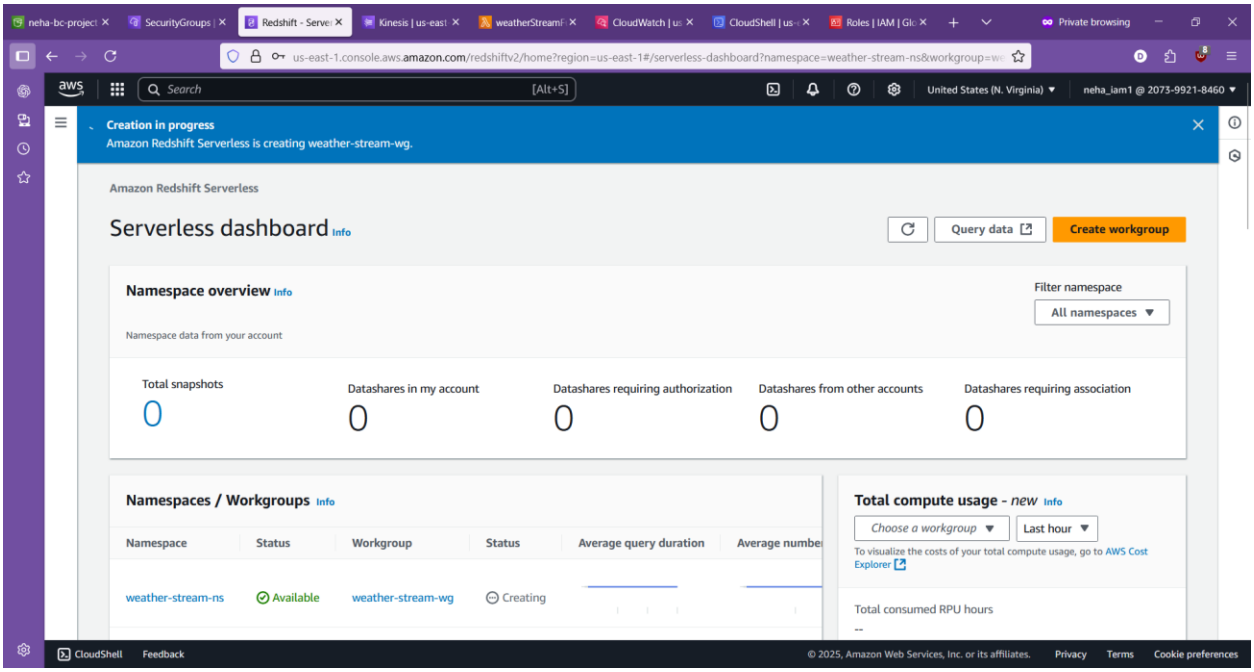

Your resources can communicate only over the IPv4 addressing protocol.
Your resources can communicate over IPv4, IPv6, or both. For dual-stack mode, associate an IPv6 CIDR block with a
Amazon EC2.
subnet-public1-us-east-1a
subnet-06b0a07be8e24ba24
subnet-private2-us-east-1b
subnet-05414590a75394baf
subnet-private3-us-east-1a
subnet-010a017d3d25d31ff
subnet-public2-us-east-1b
subnet-0c7c5f0fa8d07a978
subnet-private1-us-east-1a
subnet-05a0376ca31b04cd4
subnet-private4-us-east-1b
subnet-07d4d9974a38ae05e
Choose three or more subnet IDs
subnet-05a0376ca31b04cd4
subnet-05414590a75394baf
subnet-010a017d3d25d31ff
SSL is now enforced by default for new serverless workgroups.
SSL
SSL is enabled by default to help you get started with your workgroup. You can change the SSL settings after you finish creating your workgroup.
Enable
Enable SSL for enhanced security
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

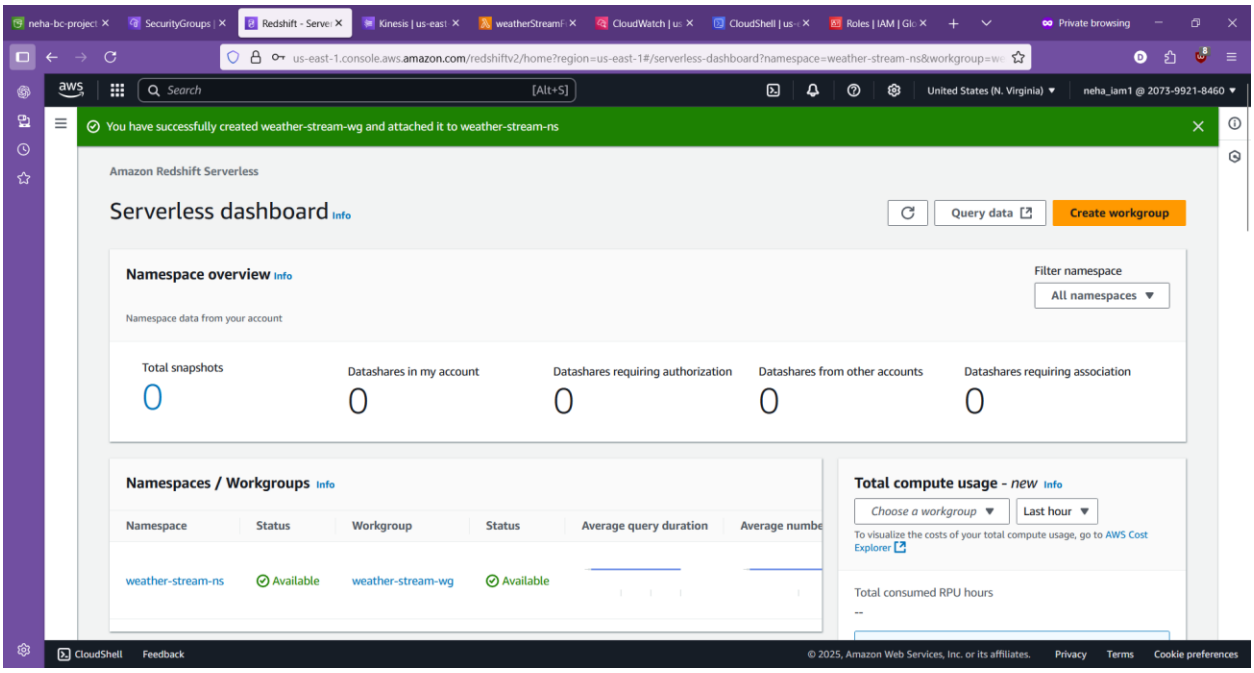

Amazon Redshift Serverless
Workgroup configuration
Create workgroup
Step 1
Create workgroup
Step 2
Choose namespace
Step 3
Review and create
Choose namespace
Namespace
Namespace is a collection of database objects and users. Data properties include database name and password, permissions, and encryption and security.
Create a new namespace
Add to an existing namespace
Namespace
This is a unique name that defines the namespace.
weather-stream-ns
The name must be from 3-64 characters. Valid characters are a-z (lowercase only), 0-9 (numbers), and - (hyphen).
Database name and password
Database name
The name of the first database in the Amazon Redshift Serverless environment.
dev
The name must be 1-64 alphanumeric characters (lowercase only), and it can't be a reserved word.
Admin user credentials
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Database name and password
Database name
The name of the first database in the Amazon Redshift Serverless environment.
dev
The name must be 1-64 alphanumeric characters (lowercase only), and it can't be a reserved word.
Admin user credentials
IAM credentials provided as your default admin user credentials. To add a new admin username and password, customize admin user credentials.
Customize admin user credentials
To use the default IAM credentials, clear this option.
Admin user name
The administrator's user name for the first database.
admin
The name must be 1-128 alphanumeric characters, and it can't be a reserved word.
Admin password
Select an option to manage your admin password.
Manage admin credentials in AWS Secrets Manager Info
AWS manages a KMS key that encrypts your data.
Generate a password
Amazon Redshift generates an admin password.
Manually add the admin password
Manually enter the admin password.
Admin user password
The password of the admin user.
Canada#123
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


able to run these SQL commands without an IAM role attached to your namespace.
The IAM role AmazonRedshift-CommandsAccessRole-20250722T000821 was successfully created and set as the default.
Associated IAM roles (1)
Set default
Actions
Associate IAM roles
Create IAM role
Create, associate, or remove an IAM role. You can associate up to 150 IAM roles. You can also choose an IAM role and set it as the default.
Find associated iam roles
IAM roles
Status
Role type
AmazonRedshift-CommandsAccessRole-20250722T00...
Not applied
Default
Encryption and security
Your data is encrypted by default with an AWS owned key. To choose a different key, customize your encryption settings.
console.aws.amazon.com/iam/home?region=us-east-1#/roles/AmazonRedshift-CommandsAccessRole-20250722T000821
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/redshiftv2/home?region=us-east-1#/create-serverless-workgroup
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
weather-stream-ns
Database name and password
Database name
dev
Admin user credentials
admin
Permissions
Default IAM role
arn:aws:iam::207399218460:role/service-role/AmazonRedshift-CommandsAccessRole-20250722T000821
IAM roles
arn:aws:iam::207399218460:role/service-role/AmazonRedshift-CommandsAccessRole-20250722T000821
Encryption and security
AWS KMS encryption
AWS owned key
Audit logging
Off
Cancel
Previous
Create
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

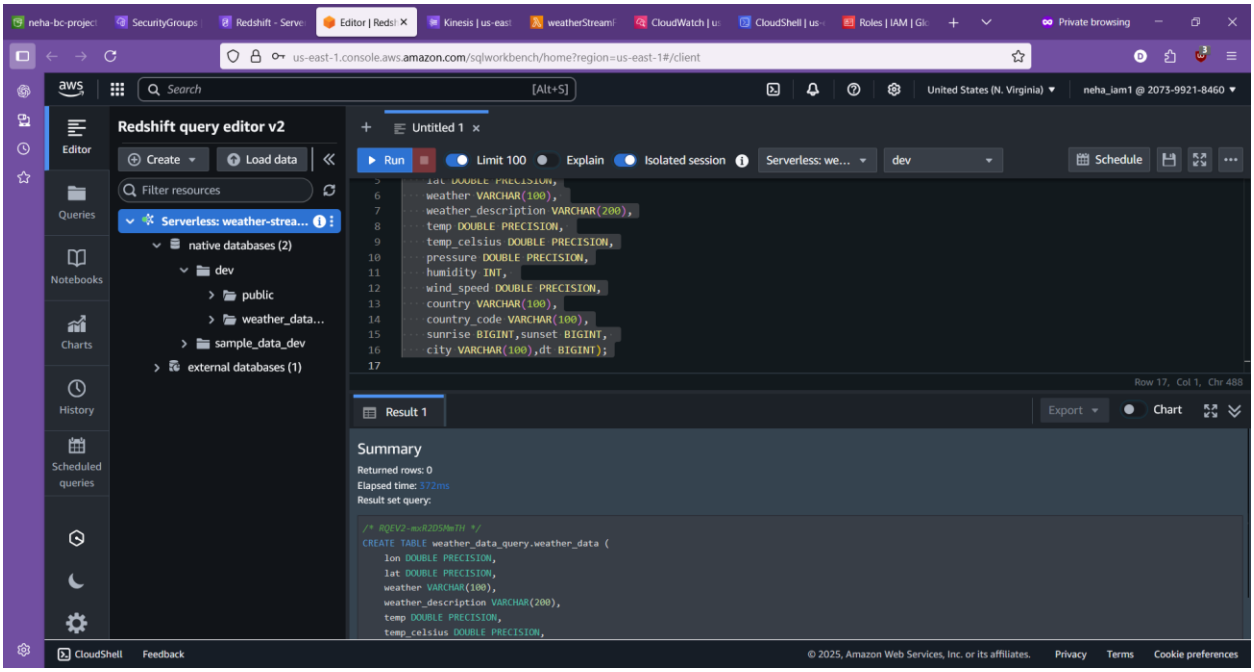

us-east-1.console.aws.amazon.com/redshiftv2/home?region=us-east-1#/serverless-dashboard?namespace=weather-stream-ns&workgroup=
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
Creation in progress
Amazon Redshift Serverless is creating weather-stream-wg.
Amazon Redshift Serverless
Serverless dashboard Info
Query data
Create workgroup
Namespace overview Info
Filter namespace
All namespaces
Namespace data from your account
Total snapshots
0
Datashares in my account
0
Datashares requiring authorization
0
Datashares from other accounts
0
Datashares requiring association
0
Namespaces / Workgroups Info
Namespace
Status
Workgroup
Status
Average query duration
Average numbe
weather-stream-ns
Available
weather-stream-wg
Creating
Total compute usage - new Info
Choose a workgroup
Last hour
To visualize the costs of your total compute usage, go to AWS Cost Explorer
Total consumed RPU hours
--
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

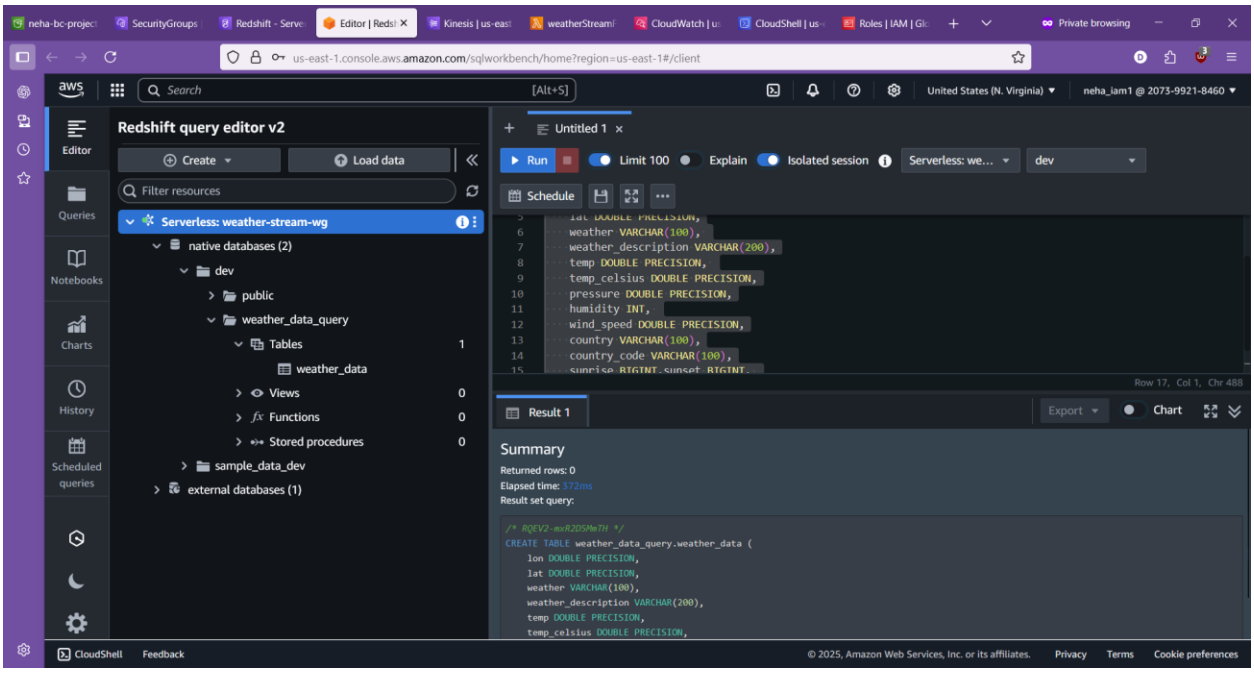


8. Query Data

CREATE TABLE weather_data_query.weather_data (lon DOUBLE PRECISION, lat DOUBLE PRECISION,weather VARCHAR(100), weather_description VARCHAR(200),temp DOUBLE PRECISION, temp_celsius DOUBLE PRECISION,pressure DOUBLE PRECISION,humidity INT, wind_speed DOUBLE PRECISION,country VARCHAR(100),country_code VARCHAR(100),sunrise BIGINT,sunset BIGINT,

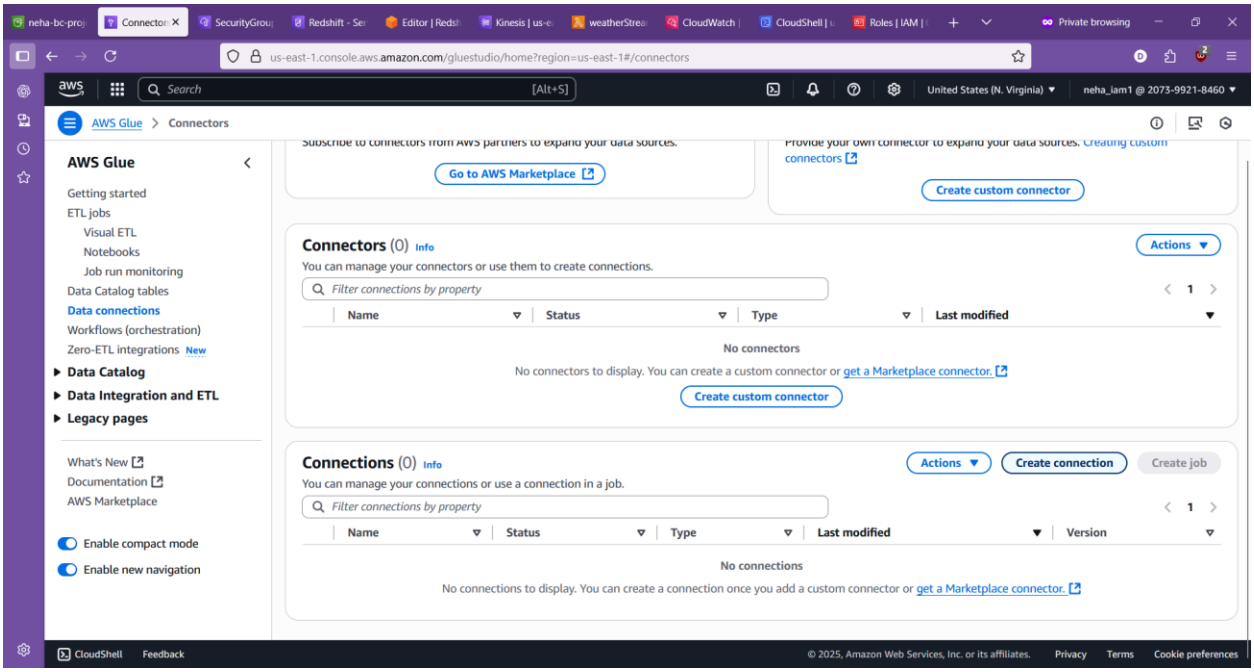
city VARCHAR(100),dt BIGINT);

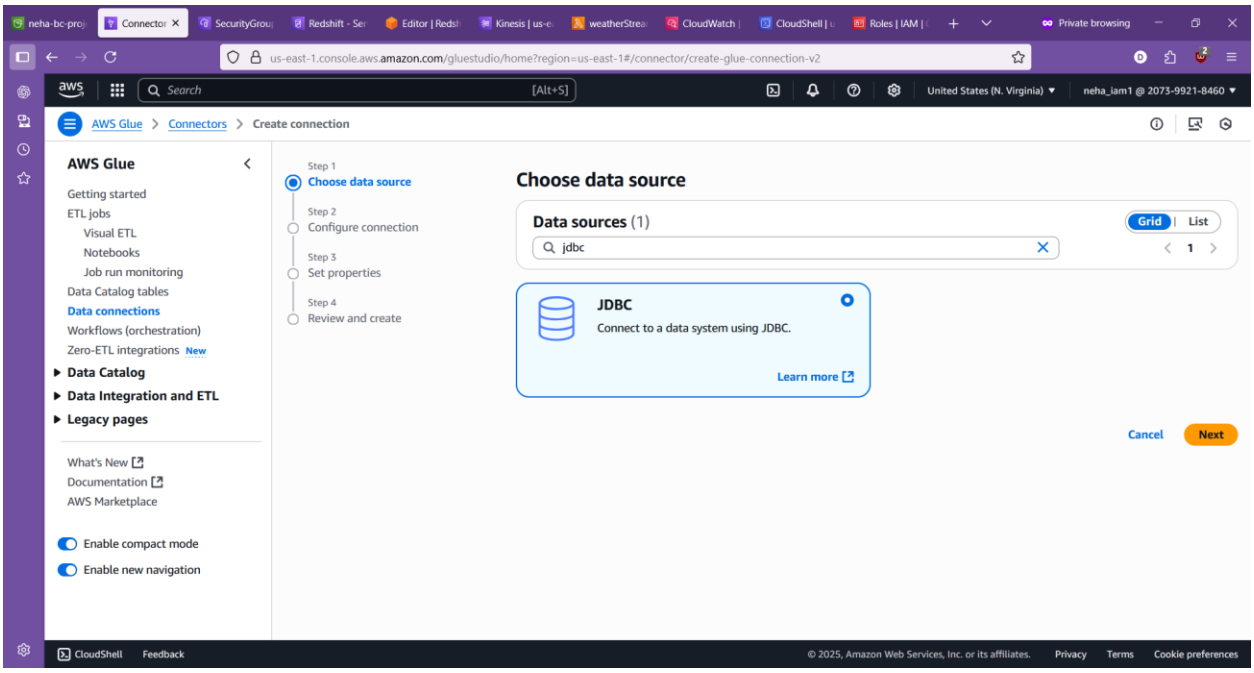


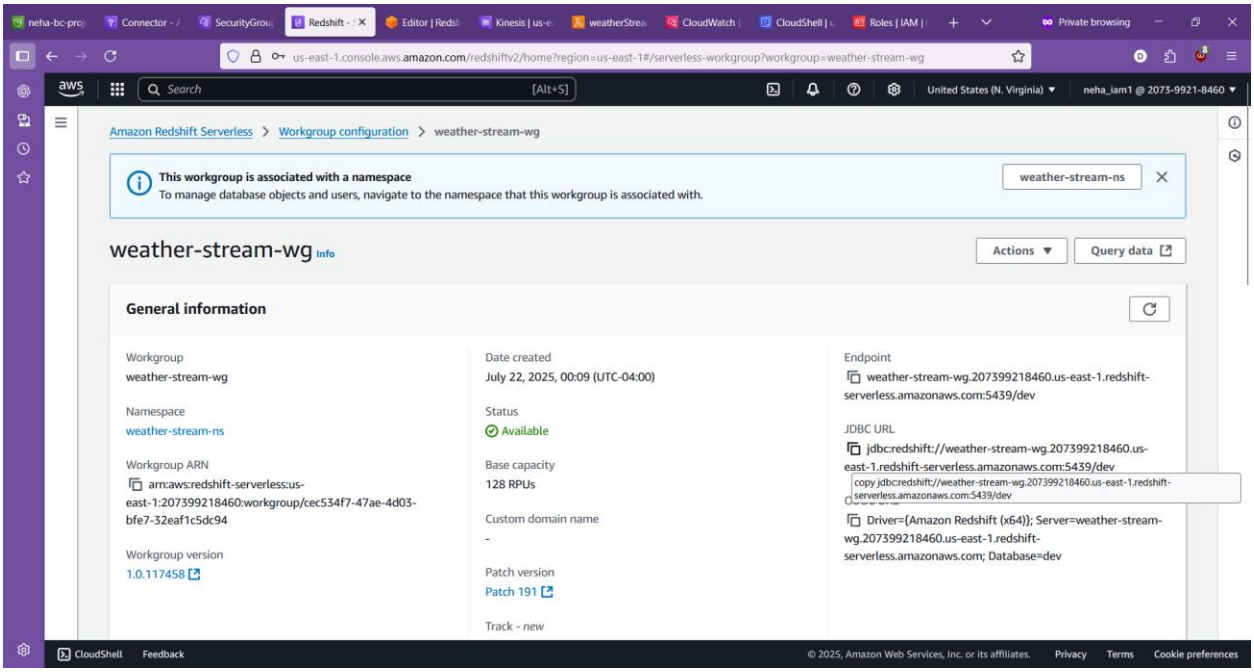


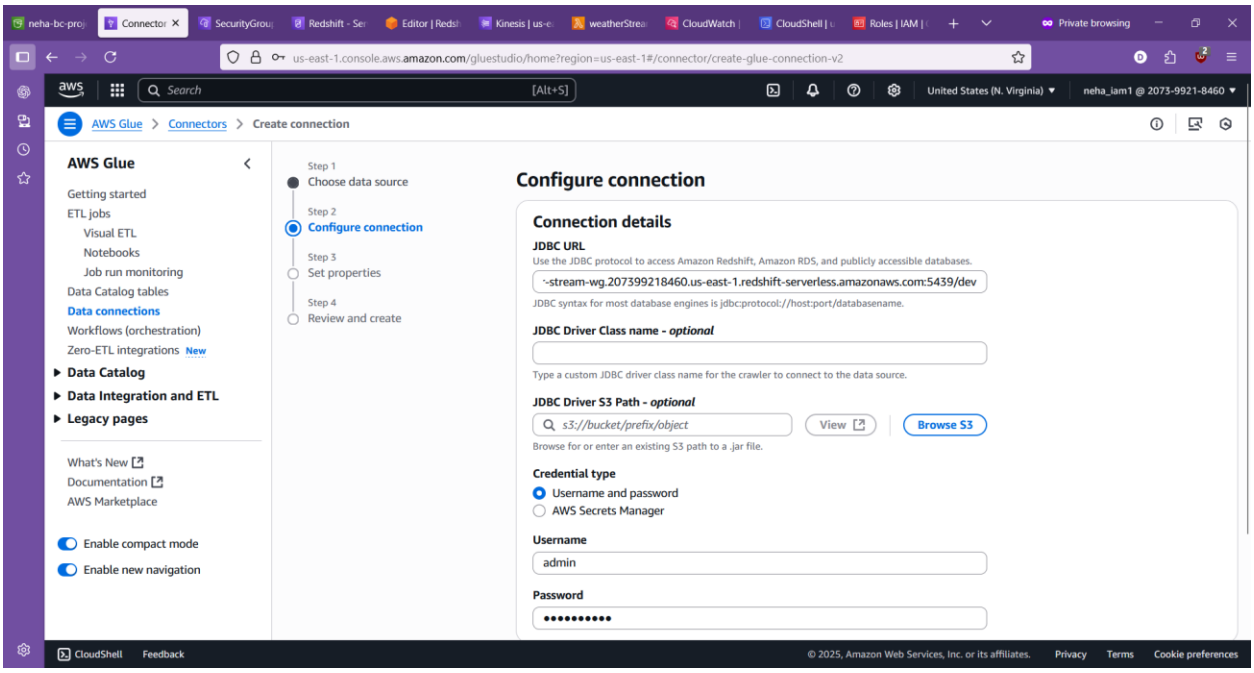
Gold Layer

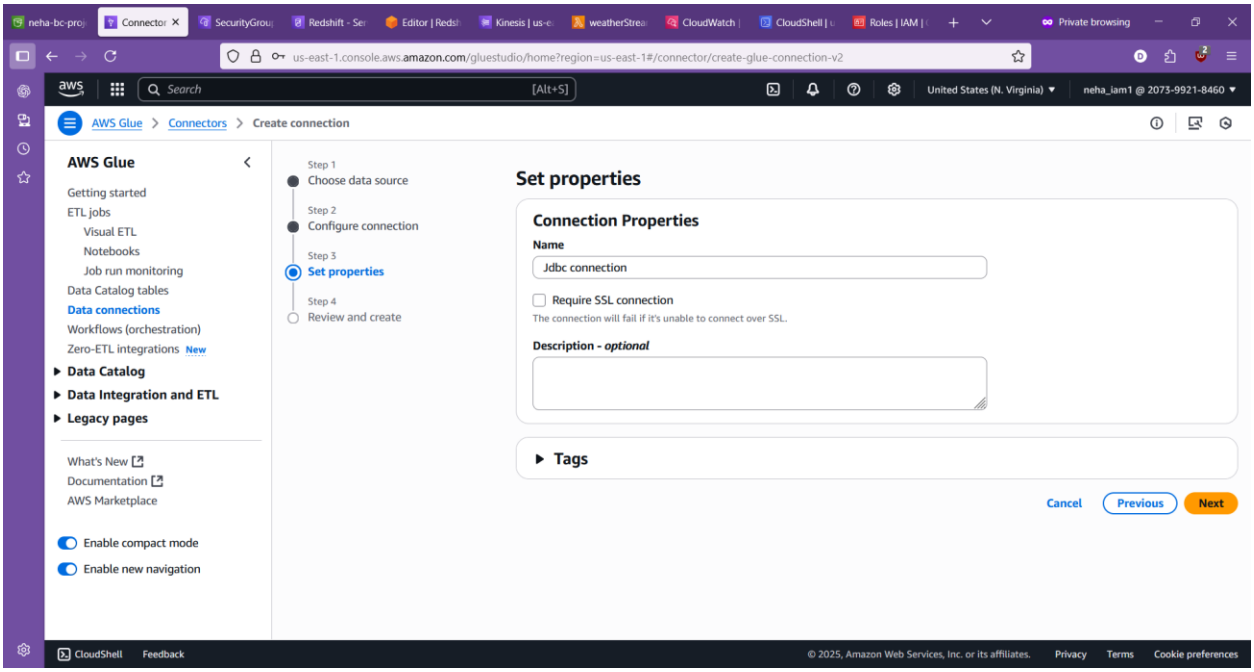
1. Create AWS Glue Connection

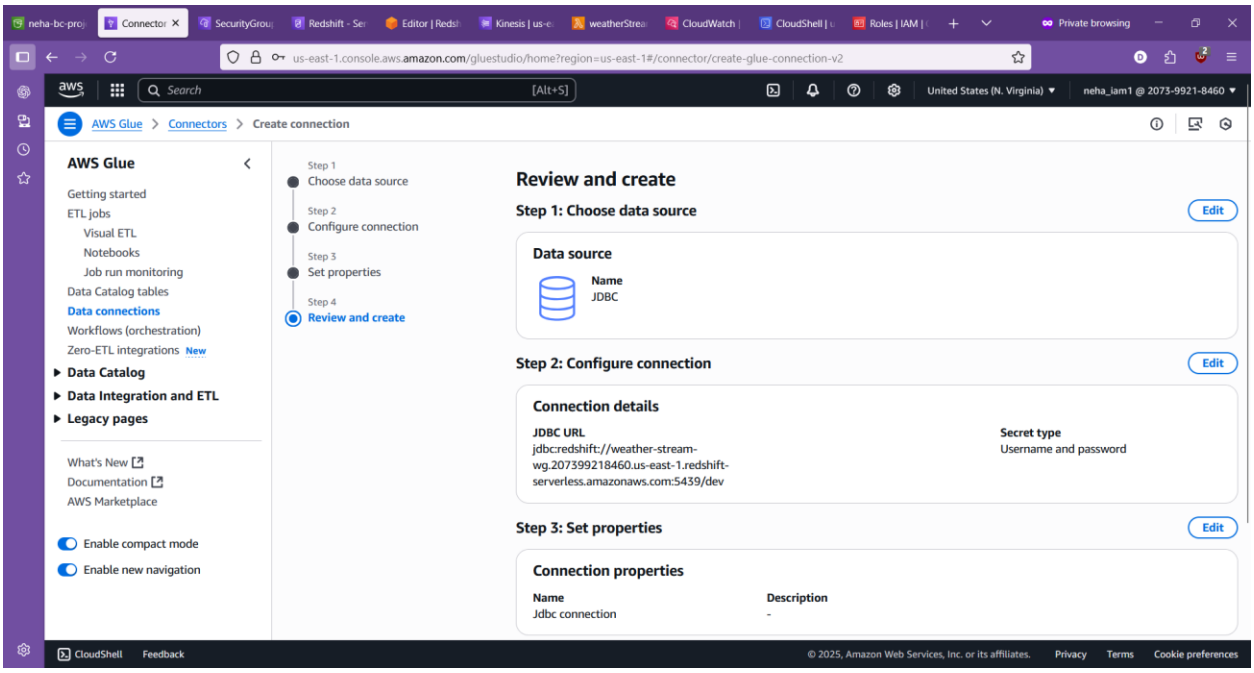


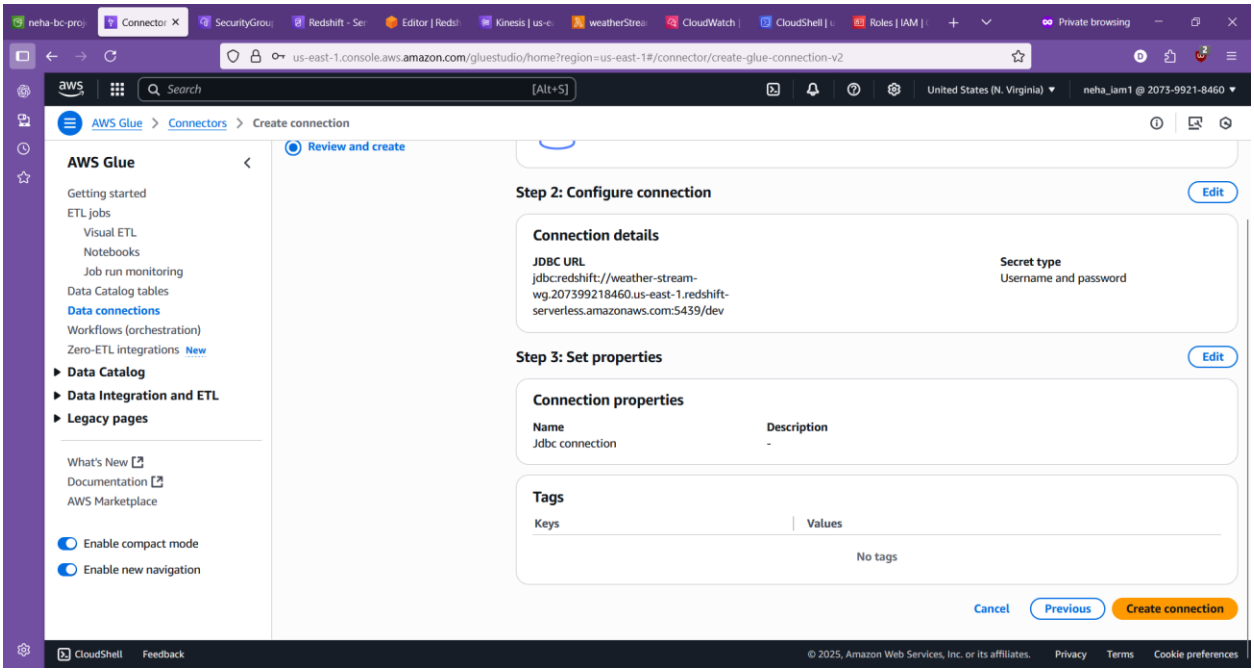

Connector
us-east-1.console.aws.amazon.com/gluestudio/home?region=us-east-1#/connector/create-glue-connection-v2
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
AWS Glue > Connectors > Create connection
AWS Glue
Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Zero-ETL integrations New
Data Catalog
Data Integration and ETL
Legacy pages
What's New
Documentation
AWS Marketplace
Enable compact mode
Enable new navigation
Step 1
Choose data source
Step 2
Configure connection
Step 3
Set properties
Step 4
Review and create
Choose data source
Data sources (1)
jdbc
Grid
List
JDBC
Connect to a data system using JDBC.
Learn more
Cancel
Next
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

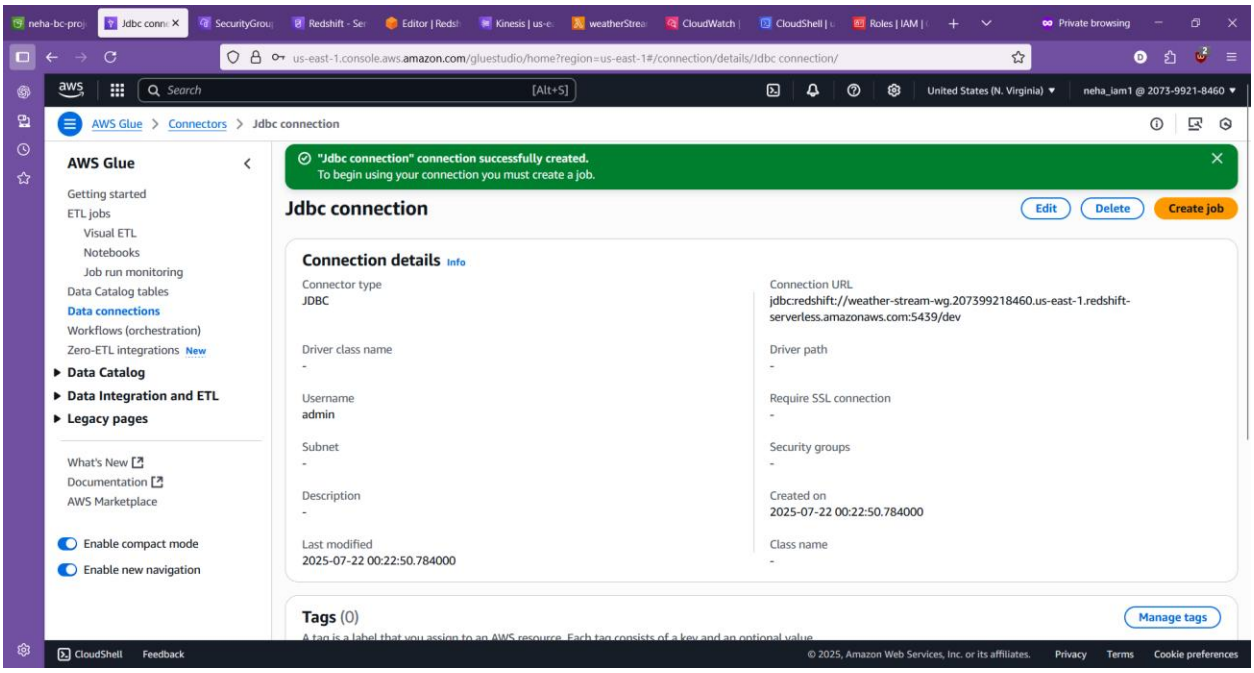

Redshift
us-east-1.console.aws.amazon.com/redshiftv2/home?region=us-east-1#/serverless-workgroup?workgroup=weather-stream-wg
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
Amazon Redshift Serverless > Workgroup configuration > weather-stream-wg
This workgroup is associated with a namespace
To manage database objects and users, navigate to the namespace that this workgroup is associated with.
weather-stream-ns
weather-stream-wg Info
Actions
Query data
General information
Workgroup
weather-stream-wg
Namespace
weather-stream-ns
Workgroup ARN
arn:aws:redshift-serverless:us-east-1:207399218460:workgroup/cec534f7-47ae-4d03-bfe7-32eaf1c5dc94
Workgroup version
1.0.117458
Date created
July 22, 2025, 00:09 (UTC-04:00)
Status
Available
Base capacity
128 RPUs
Custom domain name
-
Patch version
Patch 191
Track - new
Endpoint
weather-stream-wg.207399218460.us-east-1.redshift-serverless.amazonaws.com:5439/dev
JDBC URL
jdbc:redshift://weather-stream-wg.207399218460.us-east-1.redshift-serverless.amazonaws.com:5439/dev
copy jdbc:redshift://weather-stream-wg.207399218460.us-east-1.redshift-serverless.amazonaws.com:5439/dev
Driver={Amazon Redshift (x64)}; Server=weather-stream-wg.207399218460.us-east-1.redshift-serverless.amazonaws.com; Database=dev
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

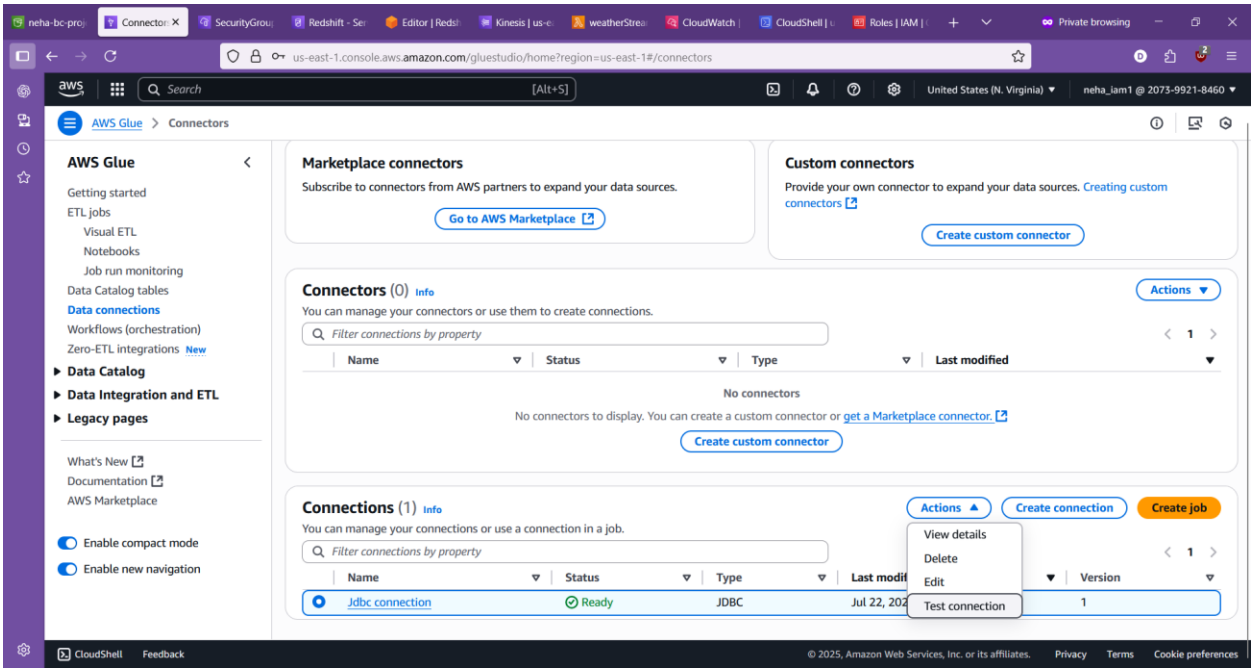

us-east-1.console.aws.amazon.com/gluestudio/home?region=us-east-1#/connector/create-glue-connection-v2
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
AWS Glue > Connectors > Create connection
AWS Glue
Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Zero-ETL integrations New
Data Catalog
Data Integration and ETL
Legacy pages
What's New
Documentation
AWS Marketplace
Enable compact mode
Enable new navigation
Step 1
Choose data source
Step 2
Configure connection
Step 3
Set properties
Step 4
Review and create
Configure connection
Connection details
JDBC URL
Use the JDBC protocol to access Amazon Redshift, Amazon RDS, and publicly accessible databases.
-stream-wg.207399218460.us-east-1.redshift-serverless.amazonaws.com:5439/dev
JDBC syntax for most database engines is jdbc:protocol://host:port/databasename.
JDBC Driver Class name - optional
Type a custom JDBC driver class name for the crawler to connect to the data source.
JDBC Driver S3 Path - optional
s3://bucket/prefix/object
View
Browse S3
Browse for or enter an existing S3 path to a .jar file.
Credential type
Username and password
AWS Secrets Manager
Username
admin
Password
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

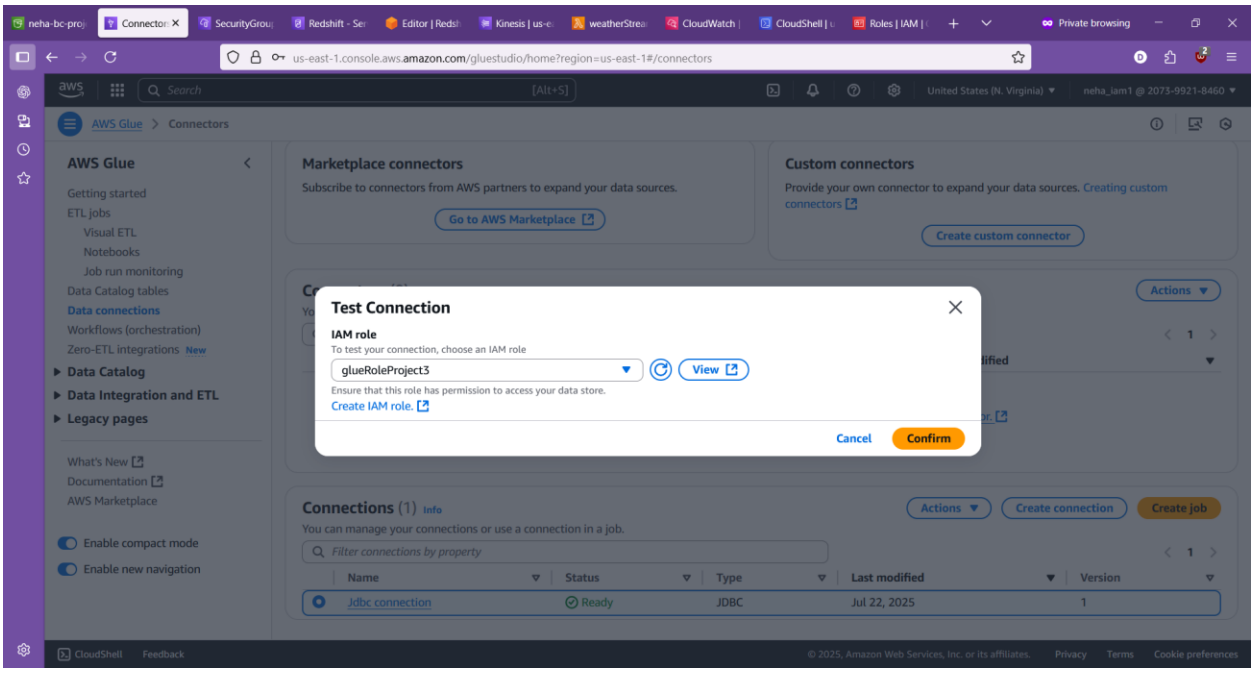

us-east-1.console.aws.amazon.com/gluestudio/home?region=us-east-1#/connector/create-glue-connection-v2
Search
[Alt+S]
United States (N. Virginia)
neha_iam1 @ 2073-9921-8460
AWS Glue > Connectors > Create connection
AWS Glue
Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Zero-ETL integrations New
Data Catalog
Data Integration and ETL
Legacy pages
What's New
Documentation
AWS Marketplace
Enable compact mode
Enable new navigation
Step 1
Choose data source
Step 2
Configure connection
Step 3
Set properties
Step 4
Review and create
Set properties
Connection Properties
Name
Jdbc connection
Require SSL connection
The connection will fail if it's unable to connect over SSL.
Description - optional
Tags
Cancel
Previous
Next
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

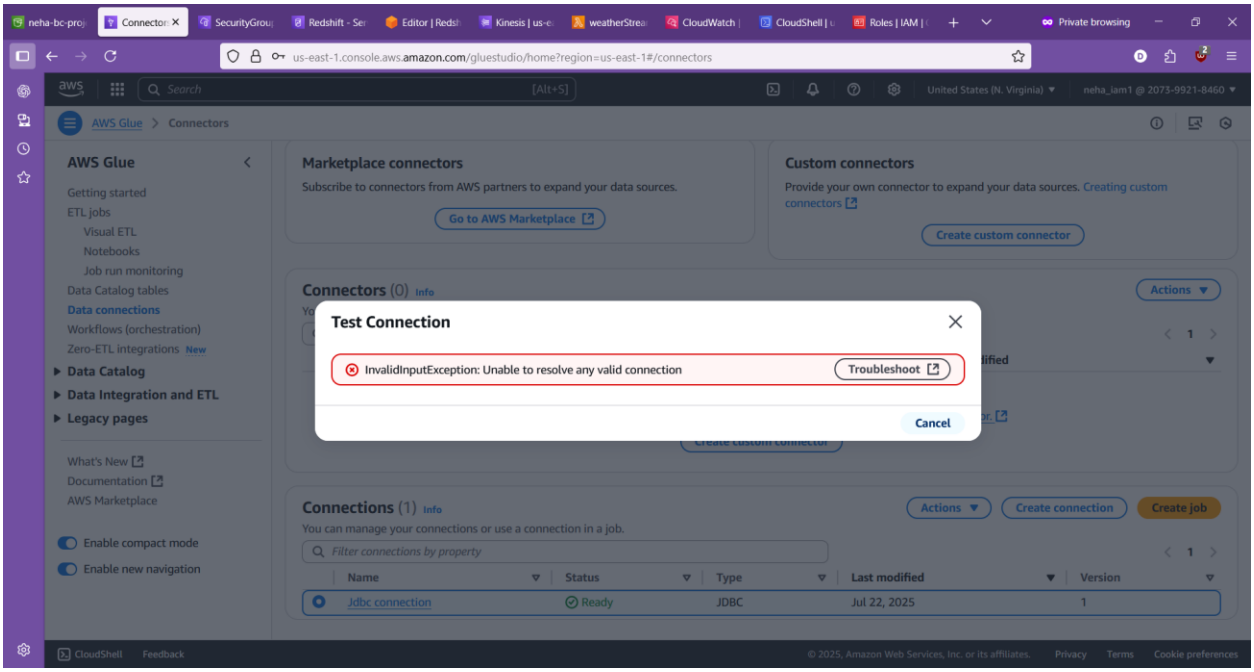

AWS Glue
Connectors
Create connection
Step 1
Choose data source
Step 2
Configure connection
Step 3
Set properties
Step 4
Review and create
Review and create
Step 1: Choose data source
Edit
Data source
Name
JDBC
Step 2: Configure connection
Edit
Connection details
JDBC URL
jdbc:redshift://weather-stream-wg.207399218460.us-east-1.redshift-serverless.amazonaws.com:5439/dev
Secret type
Username and password
Step 3: Set properties
Edit
Connection properties
Name
Jdbc connection
Description
-

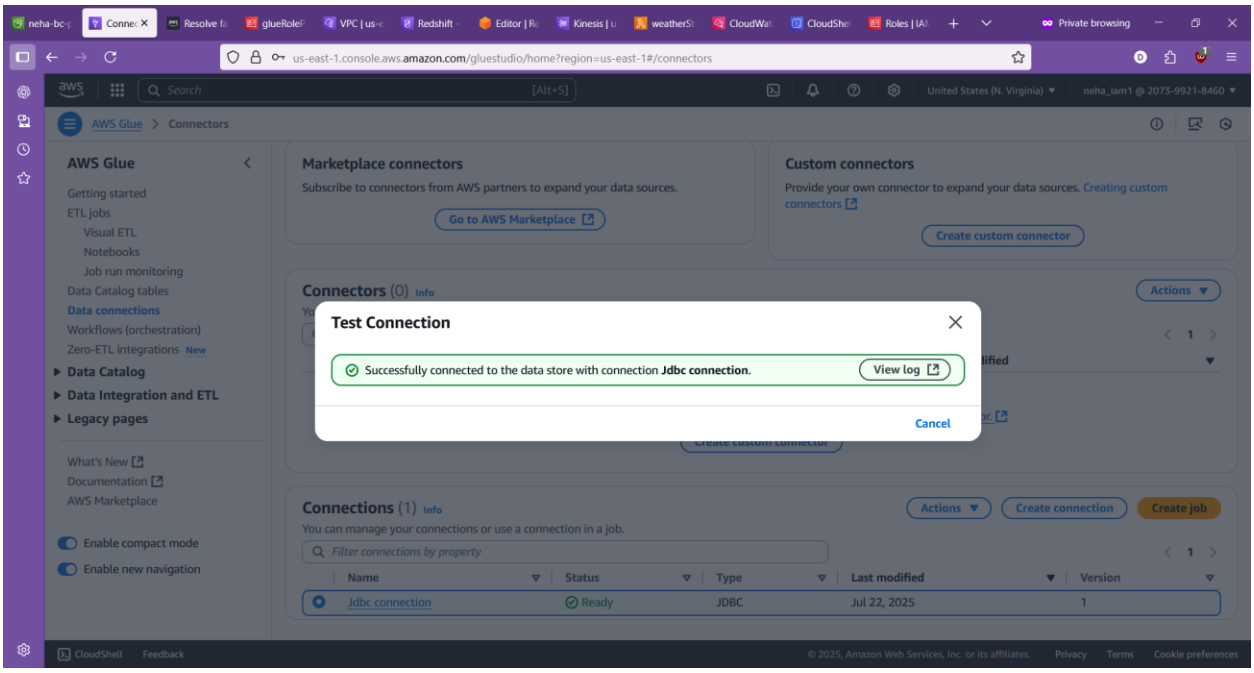

AWS Glue
Connectors
Create connection
Review and create
Step 2: Configure connection
Edit
Connection details
JDBC URL
jdbc:redshift://weather-stream-wg.207399218460.us-east-1.redshift-serverless.amazonaws.com:5439/dev
Secret type
Username and password
Step 3: Set properties
Edit
Connection properties
Name
Jdbc connection
Description
-
Tags
Keys
Values
No tags
Cancel
Previous
Create connection

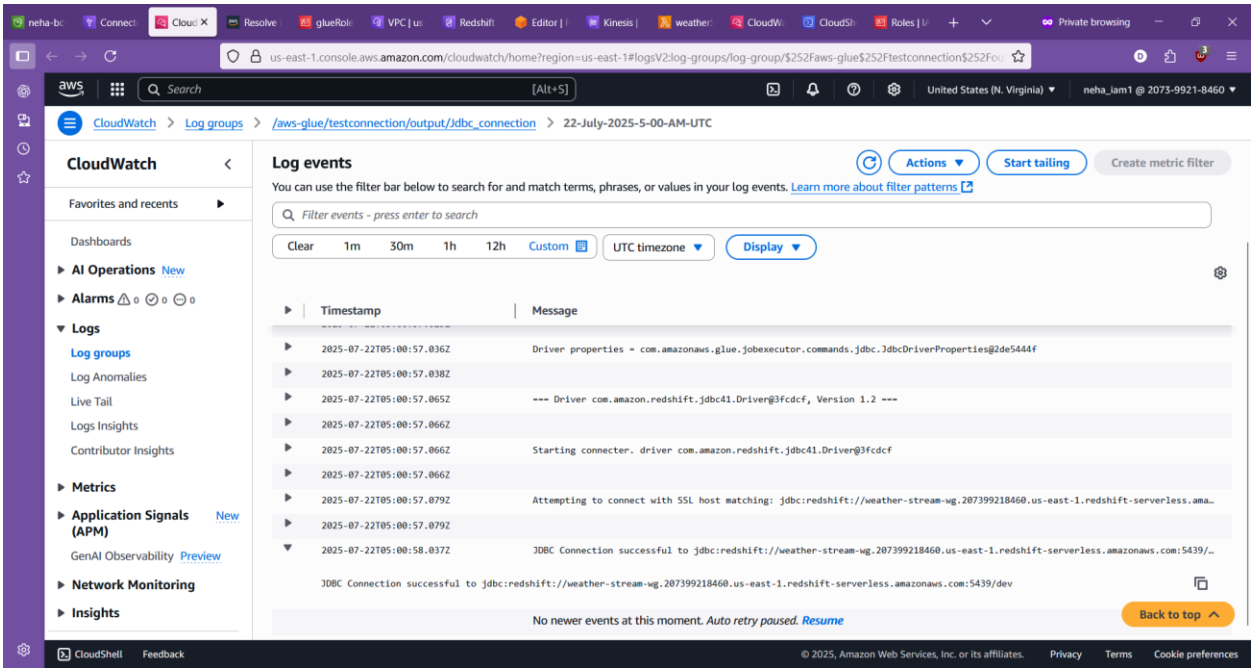

"Jdbc connection" connection successfully created.
To begin using your connection you must create a job.
Jdbc connection
Edit
Delete
Create job
Connection details Info
Connector type
JDBC
Connection URL
jdbc:redshift://weather-stream-wg.207399218460.us-east-1.redshift-serverless.amazonaws.com:5439/dev
Driver class name
-
Driver path
-
Username
admin
Require SSL connection
-
Subnet
-
Security groups
-
Description
-
Created on
2025-07-22 00:22:50.784000
Last modified
2025-07-22 00:22:50.784000
Class name
-
Tags (0)
Manage tags


Marketplace connectors
Subscribe to connectors from AWS partners to expand your data sources.
Go to AWS Marketplace
Custom connectors
Provide your own connector to expand your data sources. Creating custom connectors
Create custom connector
Connectors (0) Info
Actions
You can manage your connectors or use them to create connections.
Filter connections by property
Name
Status
Type
Last modified
No connectors
No connectors to display. You can create a custom connector or get a Marketplace connector.
Create custom connector
Connections (1) Info
Actions
Create connection
Create job
You can manage your connections or use a connection in a job.
View details
Delete
Edit
Test connection
Name
Status
Type
Version
Jdbc connection
Ready
JDBC
1

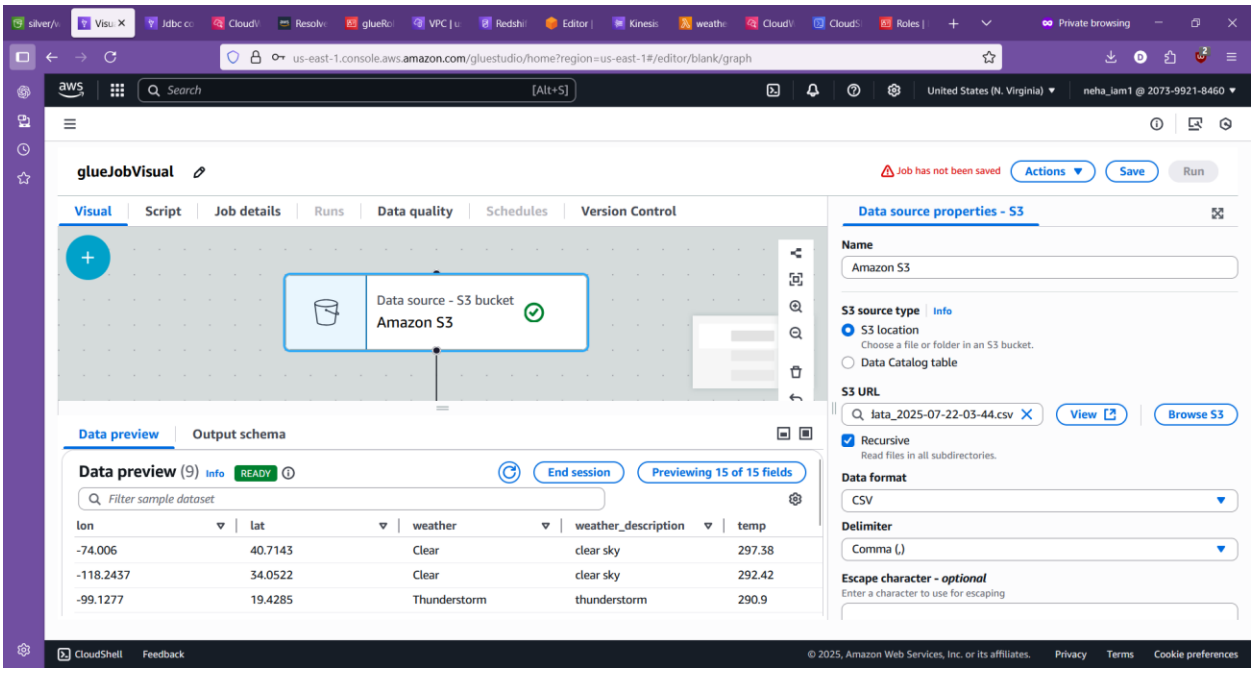

Test Connection
IAM role
To test your connection, choose an IAM role
glueRoleProject3
View
Ensure that this role has permission to access your data store.
Create IAM role.
Cancel
Confirm
Connections (1) Info
Name
Status
Type
Last modified
Version
Jdbc connection
Ready
JDBC
Jul 22, 2025
1

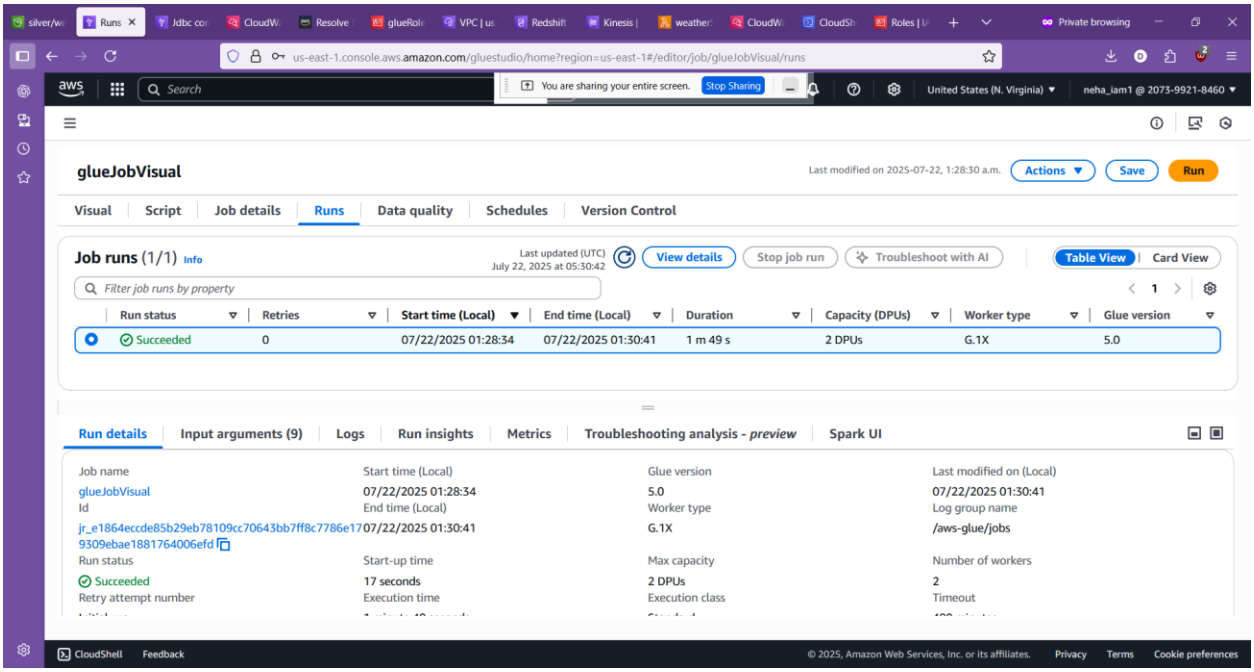

Test Connection
InvalidInputException: Unable to resolve any valid connection
Troubleshoot
Cancel
Connections (1) Info
Name
Status
Type
Last modified
Version
Jdbc connection
Ready
JDBC
Jul 22, 2025
1



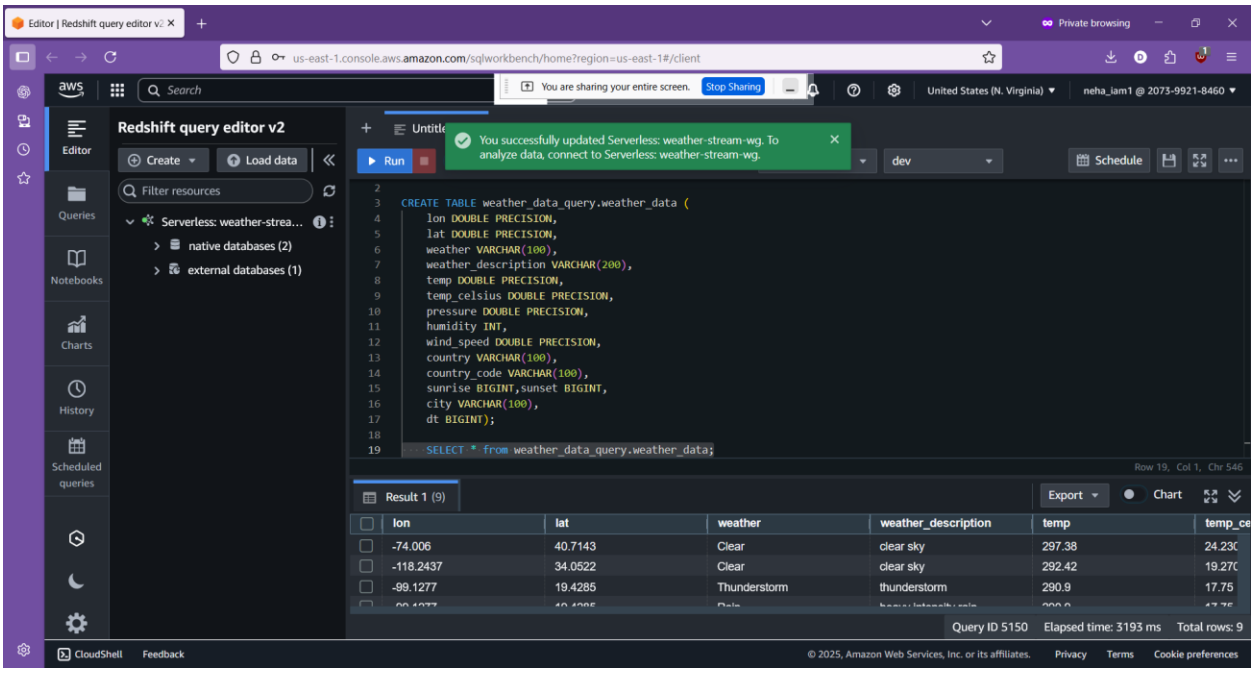


Create Glue Visual ETL job



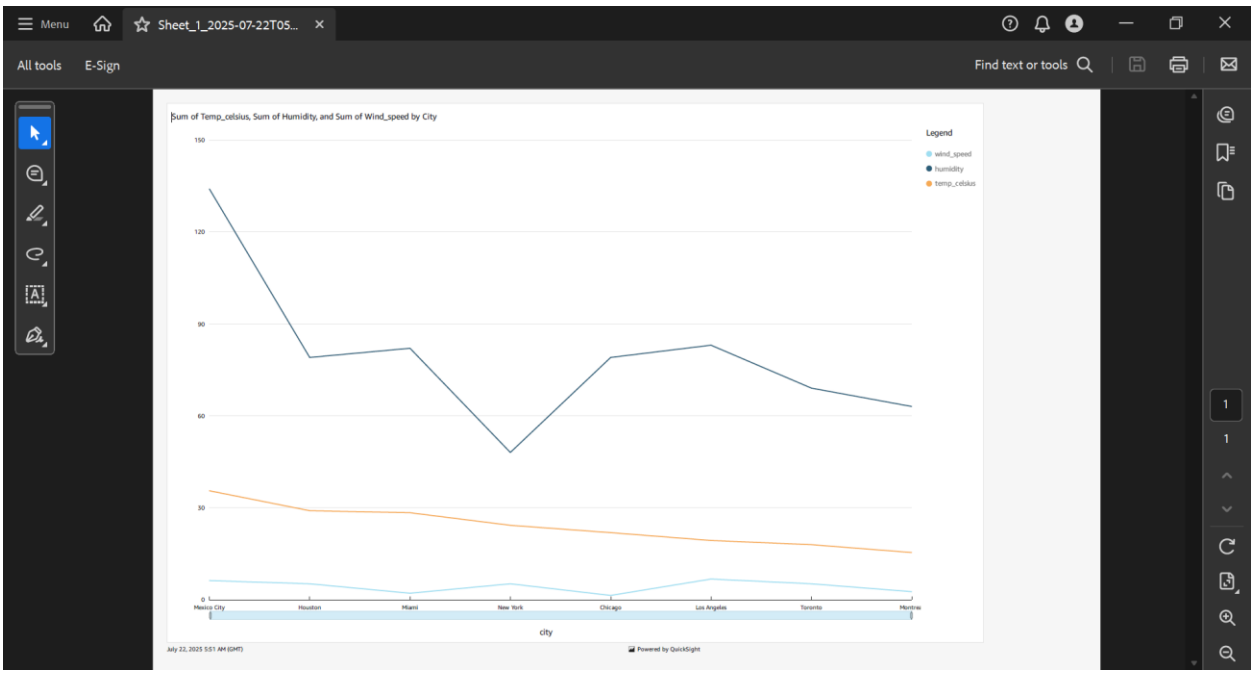


2. Check in Redshift table



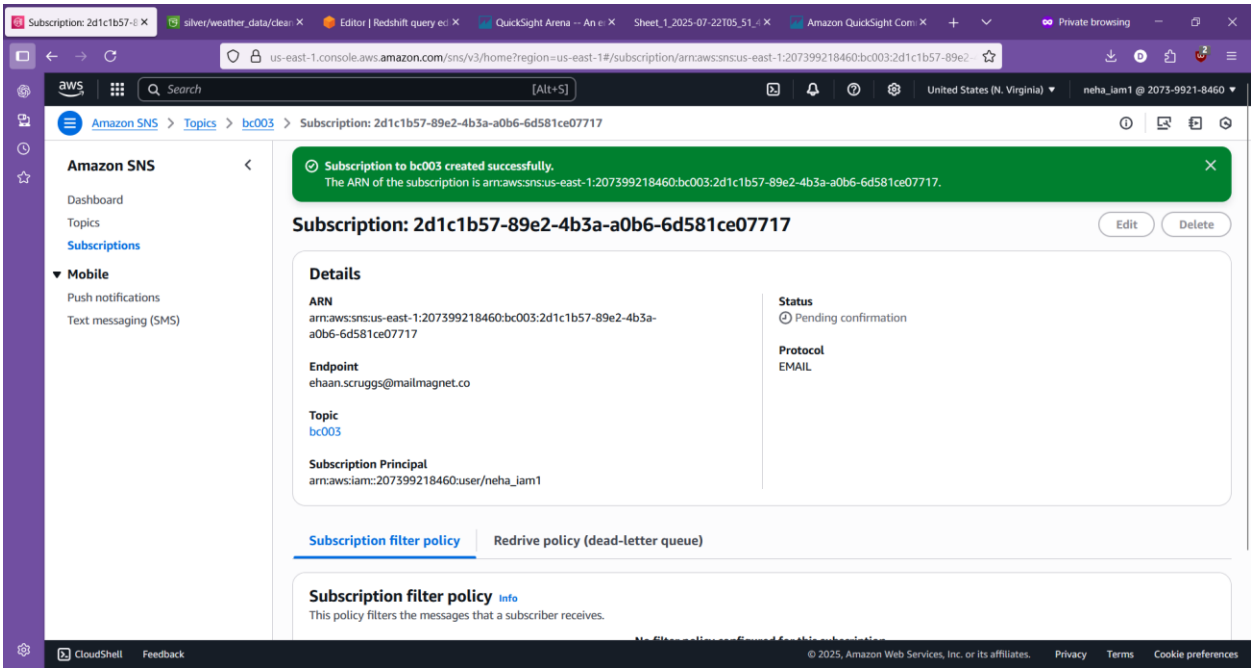
Visualization

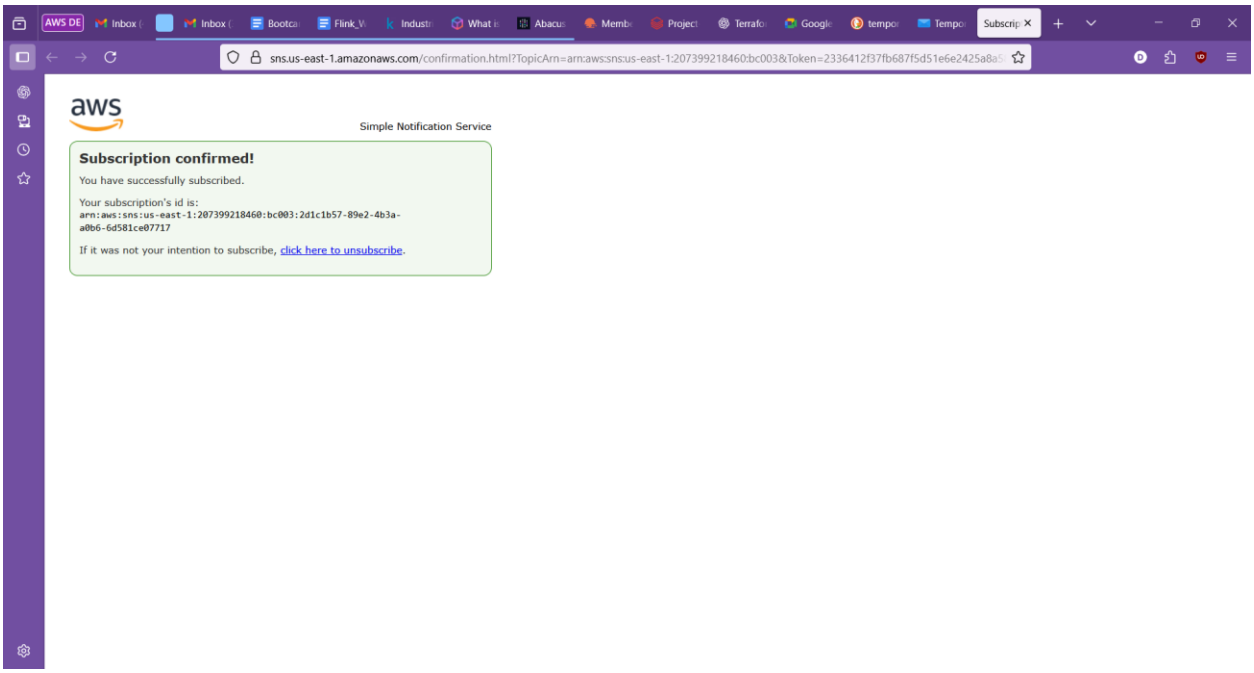
Sheet_1_2025-07-22T05_51_41.pdf



Advance analytics

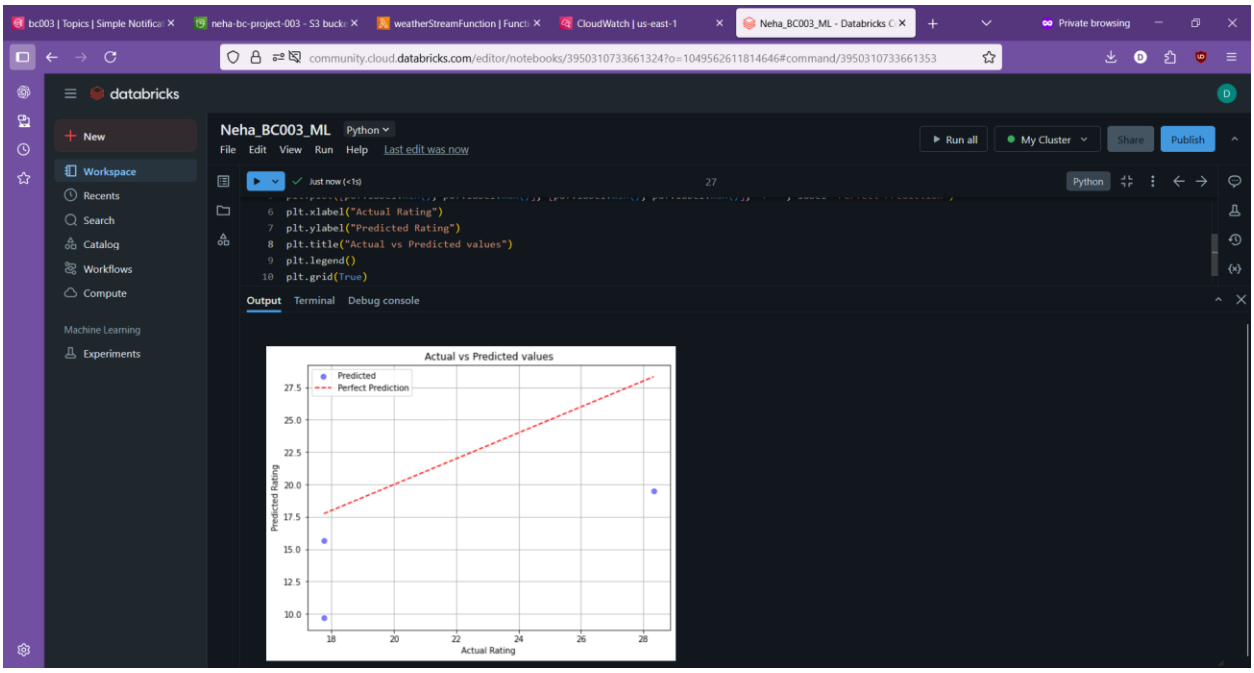
SNS topics:





Databricks:

https://community.cloud.databricks.com/editor/notebooks/3950310733661324?o=1049562611814646#command/3950310733661325


databricks
Neha_BC003_ML
Python
File Edit View Run Help Last edit was now
Run all
My Cluster
Share
Publish
New
Workspace
Recents
Search
Catalog
Workflows
Compute
Machine Learning
Experiments
Just now (<1s)
Python
plt.xlabel("Actual Rating")
plt.ylabel("Predicted Rating")
plt.title("Actual vs Predicted values")
plt.legend()
plt.grid(True)
Output Terminal Debug console
Actual vs Predicted values
Predicted
Perfect Prediction
Predicted Rating
Actual Rating