RIT | Rochester Institute of Technology of Dubai

PROF 741 Enterprise Infrastructure



AWS Lab: Performing Real-Time Data Analysis with Kinesis

Prepared by:

Walid Elsaid Shehata- AWS PS Solution Architect &

Dr. Ioannis Karamitsos- RIT Instructor

Overview

In this lab, you will build a streamlined data pipeline that collects, stores, and analyzes stock market data. You'll learn how to collect streaming data, store it efficiently, and perform post-collection analysis.

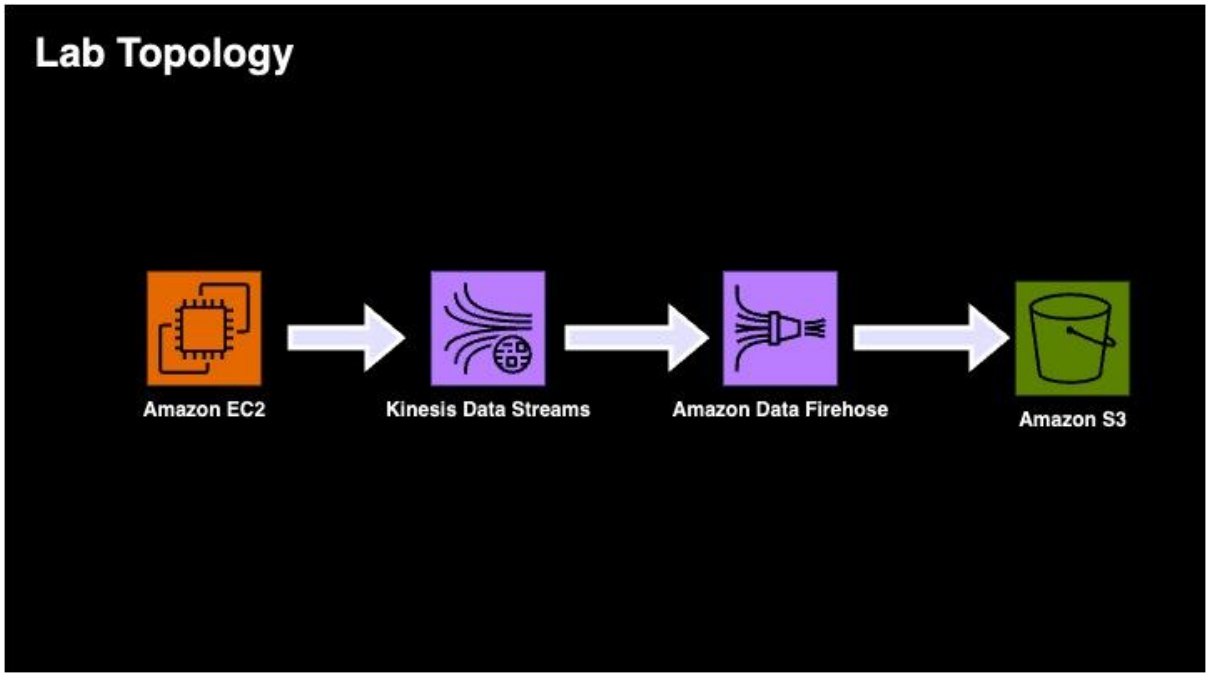
Lab Objectives

- Set up a real-time data processing architecture using AWS services
- Configure Kinesis Data Streams to ingest streaming data
- Store processed data in S3 using Amazon Data Firehose
- Analyze collected data using Python scripts

Prerequisites

- AWS user account
- Basic understanding of Python programming
- Familiarity with data analysis concepts

Architecture Overview



Part 1: Setting Up IAM Roles and Policies

Step 1: Create IAM Role for EC2

1. Navigate to the IAM console in the AWS Management Console
2. Click on "Roles" in the left navigation pane
3. Click "Create role"
4. Select "AWS service" as the trusted entity
5. Choose "EC2" as the service that will use this role
6. Click "Next: Permissions"
7. Search for and select the following policies:
    - AmazonKinesisFullAccess
    - AmazonS3FullAccess
8. Click "Next"
9. Name the role "KinesisLabEC2Role-Student*XY*" (XY are your initials, Ex: If your name is Peter Jhonson then the role is KinesisLabEC2Role-StudentPJ)
10. (Optional) Add tags if desired
11. Click "Create role"

Step 2: Create IAM Role for Kinesis Firehose

1. Navigate back to "Roles" in the IAM console
2. Click "Create role"
3. Select "AWS service" as the trusted entity
4. Choose "Kinesis" from the service list
5. Select "Kinesis Firehose" as the use case
6. Click "Next: Permissions"
7. Search for and select the following policies:
    - AmazonS3FullAccess
    - AmazonKinesisFullAccess
8. Click "Next"
9. Name the role "KinesisLabFirehoseRole-Student*XY*" (XY are your initials, Ex: If your name is Peter Jhonson then the role is KinesisLabFirehoseRole-StudentPJ)
10. (Optional) Add tags if desired
11. Click "Create role"

Part 2: Setting Up S3 Bucket

1. Navigate to the S3 console in the AWS Management Console
2. Click "Create bucket"
3. Enter a unique bucket name: "stock-data-lab-rit-*your-initials*"
4. Keep all other default settings
5. Click "Create bucket"
6. Once created, click on the bucket name to open it
7. Create a folder inside the bucket:
    - Click "Create folder" and name it "stock-data"

Part 3: Setting Up Kinesis Data Stream

1. Navigate to the Kinesis console in the AWS Management Console
2. Click on "Data streams" in the left navigation pane
3. Click "Create data stream"
4. Enter "StockDataStream" for the stream name
5. Select "On-demand" for capacity mode
6. Click "Create data stream"
7. Wait for the stream status to become "Active" (may take a minute)

Part 4: Setting Up Amazon Data Firehose

1. Navigate to the Kinesis console
2. Click on "Amazon Data Firehose" in the left navigation pane
3. Click "Create Firehose stream"
4. For source, select "Amazon Kinesis Data Streams" and choose previously created stream “StockDataStream"
5. For destination, select "Amazon S3"
6. Enter "AvgPriceFirehose" for the delivery stream name
7. Under Destination Settings, select your previously created S3 bucket
8. Enter "stock-data/" for the S3 prefix
9. Enter "errors" for the S3 error prefix
10. For S3 buffer conditions, set:
    - Buffer size: 1 MB
    - Buffer interval: 60 seconds
11. For S3 compression, select "Disabled"
12. For S3 encryption, select "Use the encryption setting of the S3 bucket"
13. Under Advanced Settings, in service access, select the "KinesisLabFirehoseRole" you created earlier
14. Click "Next"
15. Review the settings and click "Create Firehose stream"

Part 5: Setting Up EC2 Instance

1. Navigate to the EC2 console in the AWS Management Console
2. Click "Launch Instance"
3. Enter "KinesisLabInstance-rit-*YourInitialsXY*" for the name
4. Select "Amazon Linux 2023" for the AMI
5. Select "t2.micro" for the instance type
6. Under "Key pair", create a new one (The .pem file should be saved, as this can be used to have remote access to the Instance -if needed)
7. Under "Network settings", select "Create security group"
8. Make sure "Allow SSH traffic" is checked
9. Under "Advanced details", select the "KinesisLabEC2Role" for the IAM instance profile
10. Keep all other settings at their defaults

11. Click "Launch instance"
12. Wait for the instance to initialize (Status checks: 2/2 passed) [You might need to hit the refresh button]
13. Select the instance and click "Connect"
14. Choose "EC2 Instance Connect" and click "Connect"

Part 6: Generate Sample Stock Data

1. In the EC2 terminal, run the following commands to install required packages:

```
sudo yum update -y
sudo yum install -y python3 python3-pip
pip3 install boto3 pandas matplotlib
```

2. Create a Python script to generate stock data:

```
nano stock_data_generator.py
```

3. Paste the following code:

```
import json
import random
import time
import boto3
from datetime import datetime

# Initialize Kinesis client
kinesis_client = boto3.client('kinesis', region_name='us-east-1')  # Change to your region if different

# Stock symbols to generate data for
stocks = ['AAPL', 'MSFT', 'AMZN', 'GOOGL', 'META', 'TSLA', 'NVDA', 'PYPL', 'NFLX', 'INTC']

def generate_stock_data():
    """Generate random stock market data"""
    stock = random.choice(stocks)
    price = random.uniform(100, 1000)
    change_percent = random.uniform(-5, 5)
    volume = random.randint(1000, 10000)
    timestamp = datetime.now().isoformat()

    return {
        'stock': stock,
        'price': round(price, 2),
        'change_percent': round(change_percent, 2),
        'volume': volume,
```

```
        'timestamp': timestamp
    }

def send_to_kinesis(data):
    """Send data to Kinesis Data Stream"""
    kinesis_client.put_record(
        StreamName='StockDataStream',
        Data=json.dumps(data),
        PartitionKey=data['stock']
    )
    print(f"Sent: {data}")

# Main loop to generate and send data
def main():
    while True:
        stock_data = generate_stock_data()
        send_to_kinesis(stock_data)
        # Sleep for a random time between 0.1 and 0.5 seconds
        time.sleep(random.uniform(0.1, 0.5))

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        print("Data generation stopped")
```

4. Save the file and exit

5. Run the script:

```
python3 stock_data_generator.py
```

6. Let it run for a few minutes to generate streaming data. You should see output showing the data being sent to Kinesis.

7. Keep this script running for the remainder of the lab. You might want to open a new terminal window for subsequent steps.

Part 7: Analyze the collected Data

1. Wait at least 5 minutes after stopping the generator to ensure all data has been delivered to S3 (Firehose uses buffering)
2. Create a Python Script for analyzing/visualizing data captured on S3.

Part 8: Clean Up Resources

When you've completed the lab, follow these steps to clean up all resources to avoid unnecessary charges:

1. Stop any running scripts on the EC2 instance (Ctrl+C)

2. Delete the Kinesis Firehose Delivery Stream:

   - Navigate to Kinesis in the AWS Management Console

   - Click on "Delivery streams"

   - Select "StockDataFirehose"

   - Click "Delete" and confirm

3. Delete the Kinesis Data Stream:

   - Navigate to Kinesis in the AWS Management Console

   - Click on "Data streams"

   - Select "StockDataStream"

   - Click "Delete" and confirm

4. Terminate the EC2 Instance:

   - Navigate to EC2 in the AWS Management Console

   - Select your instance

   - Click "Instance state" -> "Terminate instance"

   - Confirm the termination

5. Delete the S3 Bucket:

   - Navigate to S3 in the AWS Management Console

   - Select your bucket

   - Click "Delete"

   - Type the bucket name to confirm deletion

6. Delete the IAM Roles:

- Navigate to IAM in the AWS Management Console

- Click on "Roles"

- Select "KinesisLabEC2Role"

- Click "Delete" and confirm

- Repeat for "KinesisLabFirehoseRole"

Conclusion

In this lab, you've built a streamlined data pipeline using AWS Kinesis services. You've learned how to:

- Generate and ingest streaming data with Kinesis Data Streams

- Deliver streaming data to S3 with Kinesis Data Firehose

- Perform batch analysis on collected data

These skills are directly applicable to real-world scenarios such as monitoring financial markets, analyzing IoT sensor data, tracking application metrics, or processing click streams for web analytics.

The data pipeline pattern you've implemented is a fundamental architecture used in many big data and analytics applications, providing a solid foundation for more advanced data engineering concepts.