Multi-Cloud Data Pipeline: ServiceNow API to Power BI Analytics

Project Overview

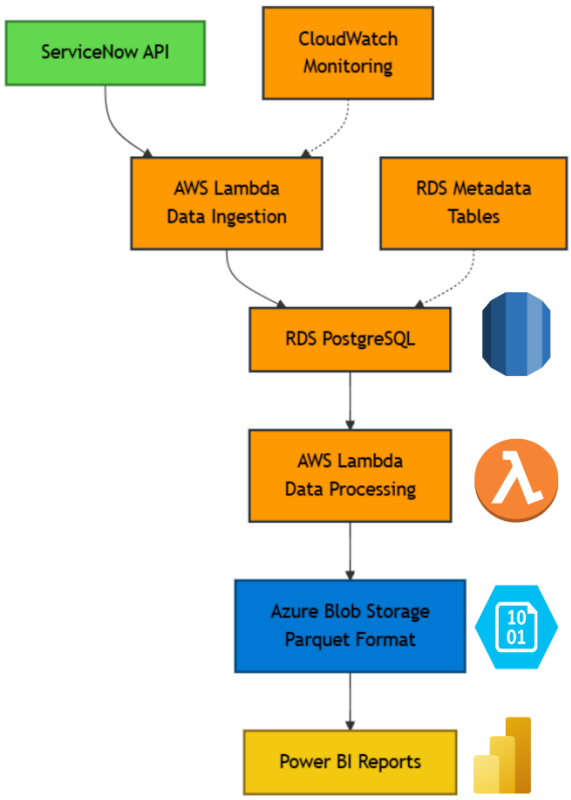
This project demonstrates a production-ready multi-cloud data pipeline that extracts IT service management data from ServiceNow API, processes it through AWS Lambda and RDS, transfers to Azure Blob Storage in optimized Parquet format, and delivers insights through Power BI dashboards. The pipeline showcases cross-cloud integration, cost optimization, and automated reporting capabilities.

Business Problem

An enterprise IT organization needs to:

- Integrate real-time ServiceNow incident, change, and asset data for operational insights
- Store historical ITSM data for trend analysis and predictive maintenance
- Transfer processed data to Azure for existing Power BI infrastructure
- Automate daily/weekly IT operations reports
- Optimize costs across AWS and Azure services
- Ensure 99.9% data accuracy and reliability for critical business decisions

Data Pipeline Architecture Overview



Anupama Godara
+91 9717298623
anupama.godara4@gmail.com

Technical Implementation

1. AWS Lambda - ServiceNow API Data Ingestion

Function Purpose: Extracts incident, change request, and configuration item data from ServiceNow REST API with error handling and rate limiting.

Key Features:

- Scheduled execution via CloudWatch Events (every 15 minutes for incidents, hourly for changes)
- OAuth 2.0 authentication with ServiceNow
- Incremental data extraction using sys_updated_on timestamps
- Built-in retry logic with exponential backoff
- Data validation and cleansing
- CloudWatch logging and metrics

2. AWS Lambda - RDS to Azure Blob Transfer

Function Purpose: Processes and transforms ServiceNow data from RDS PostgreSQL, converts to optimized Parquet format, and transfers to Azure Blob Storage for Power BI consumption.

Key Features:

- Triggered by CloudWatch Events (daily at 2 AM UTC for full sync)
- Pandas-based data transformation and aggregation
- Parquet format optimization with compression
- Azure Blob Storage integration with SAS token authentication
- Data partitioning by date for efficient Power BI queries
- Error handling with retry mechanism for Azure connectivity issues

Anupama Godara
+91 9717298623
anupama.godara4@gmail.com

Key Technical Components:

```
# Main function structure
def lambda_handler(event, context):
    process_type = event.get('process_type', 'incremental')
    if process_type == 'full':
        process_full_sync()
    else:
        process_incremental_sync()

# Data extraction from RDS
def extract_data_from_rds(data_type, date_filter=None):
    conn = psycopg2.connect(host=RDS_ENDPOINT, database=RDS_DATABASE)
    query = f"SELECT * FROM {data_type} WHERE DATE(updated_at) >=
'{date_filter}'"
    df = pd.read_sql_query(query, conn)
    return df

# Azure Blob Storage upload
def upload_to_azure_blob(df, data_type, date_partition):
    blob_service_client =
BlobServiceClient.from_connection_string(CONNECTION_STRING)
    blob_name =
f"servicenow/{data_type}/year={year}/month={month}/{data_type}_{date_partition}.
parquet"

    parquet_buffer = BytesIO()
    df.to_parquet(parquet_buffer, engine='pyarrow', compression='snappy')

    blob_client.upload_blob(parquet_buffer, overwrite=True)
```

Environment Variables:

- RDS_ENDPOINT: RDS PostgreSQL endpoint
- RDS_DATABASE: Database name
- RDS_USERNAME: Database username
- RDS_PASSWORD: Database password
- AZURE_STORAGE_CONNECTION_STRING: Azure Storage connection string
- AZURE_CONTAINER_NAME: Azure Blob container name

Required Python Packages (in deployment package):

- pandas
- pyarrow
- azure-storage-blob
- psycopg2-binary

Anupama Godara
+91 9717298623
anupama.godara4@gmail.com

IAM Permissions Required:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rds:DescribeDBInstances",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*"
        }
    ]
}
```

Data Quality Features:

- Null value handling and data type optimization
- Calculated columns for Power BI analytics
- Data validation before Azure upload
- Transfer logging for audit trail

Anupama Godara
+91 9717298623
anupama.godara4@gmail.com