# Datalake for Fluent

This repo allows users to create a complete data lake in S3. It consists of a series of top level CloudFormation templates.

Template datalake.yaml at the current level is the parent template to be run in CloudFormation.

Templates under./datalake/ path are components to make nested stacks. They are never called directly, but are only called from the current level datalake.yaml templates

## Datalake Architecture

A screenshot of a social media post

Description automatically generated

## Datalake Infrastructure

Components:

[Datalake Infrastructure Stack](file:///Users/panchumarthi/Downloads/datalake.yaml)

* [KMS Customer Managed Key](file:///Users/panchumarthi/Downloads/datalake/kms.yaml)
* [S3 Raw Data Bucket](file:///Users/panchumarthi/Downloads/datalake/bucket.yaml)
* [S3 Refined Data Bucket](file:///Users/panchumarthi/Downloads/datalake/bucket.yaml)
* [S3 Transient Data Bucket](file:///Users/panchumarthi/Downloads/datalake/bucket.yaml)
* [S3 Glue Temporary Bucket](file:///Users/panchumarthi/Downloads/datalake/bucket.yaml)
* [S3 Athena Query Bucket](file:///Users/panchumarthi/Downloads/datalake/bucket.yaml)
* [Kinesis Data Stream](file:///Users/panchumarthi/Downloads/datalake/kinesis-datastream.yaml)
* [Kinesis Delivery Stream](file:///Users/panchumarthi/Downloads/datalake/kinesis-firehose.yaml)
  + Glue Database
    - Database in Glue is a Meta Store of Data Stored in S3
    - Database created for Zones are
      * Raw
      * Transient
      * Refined
  + Glue Table
    - Tables in Glue are a part of the Database that groups related data stored in S3 in a path
    - Tables created for the datalake are
      * Raw Table - to store CSV files generated out of Full Load DMS Task
      * CDC Table - to store Parquet Files streamed out of Kinesis Firehose Delivery
      * Transient Tables - Data Present in Transient Zone
      * Refined Tables - Data Present in Refined Zone
  + Log Group and Log Stream
    - Log Groups are logical grouping of logs to a particular AWS resource that generates it
    - Log Streams generated by an AWS Service that publishes logs to Amazon Cloudwatch
  + Kinesis Delivery Stream Policy
    - Allows Glue, S3, Lambda, Logs, Kinesis and KMS.
  + Delivery Stream Role
    - Role used by Kinesis Firehose Delivery Streams governed by the policy
  + Delivery Stream
    - Stream that collects data from Kinesis Data Streams and pushes to S3
* [Public VPC](file:///Users/panchumarthi/Downloads/datalake/default-vpc.yaml)
  + Internet Gateway and attach it to VPC
  + Public Subnets (2)
  + Public Route Tables for Public Subnets (2)
  + Security Group for inbound mysql from DB.
* [DMS Replication Instance](file:///Users/panchumarthi/Downloads/datalake/dmsreplicationinstance.yaml)
  + Replication Subnet Group
    - Replication Instance that uses a subnet group to deploy in a VPC
  + Replication Instance
    - Compute instance on which replication tasks run
* [DMS SNS Topic](file:///Users/panchumarthi/Downloads/datalake/dms-sns-topic.yaml)
  + SNS Topic
    - Topic for publishing notification messages
  + SNS Topic Policy
    - Permissions to trigger lambda
* [DB Secret](file:///Users/panchumarthi/Downloads/datalake/secret.yaml)
  + Secret to store credentials of the database
* [DMS Endpoint](file:///Users/panchumarthi/Downloads/datalake/dms-endpoint.yaml)
  + Source Endpoint
    - Connects to Source database
  + DMS Kinesis Policy
    - Allows DMS to put records to kinesis data stream
  + DMS Kinesis Role
    - Policy when DMS uses Kinesis as a Target
  + DMS Kinesis Target Endpoint
    - CDC are pushed to Kinesis data stream
  + DMS S3 Policy
    - Allow DMS to put records at Raw bucket
  + DMS S3 Role
    - DMS S3 Policy
  + DMS S3 Target Endpoint
    - Full Load are pushed to raw bucket
* [DMS Tasks](file:///Users/panchumarthi/Downloads/datalake/dms-tasks.yaml)
  + DMS CDC Task
    - DMS performs Change Data Capture (CDC) migration of source database
  + DMS Full Load Task
    - DMS Performs Full Load Migration of source database migration once
  + DMS Event Subscription
    - DMS publishes notifications to SNS for changes like configuration,event,failure
* [Glue IAM Roles and Policies](file:///Users/panchumarthi/Downloads/datalake/iam.yaml)
  + Refined Bucket Access Policy
  + Allow read/write/list to Refined bucket
  + Allow KMS encrypt/decrypt
  + Raw Bucket Access Policy
  + Allow read/write/list to Raw bucket
  + Allow KMS encrypt/decrypt
  + Transient Bucket Access Policy
  + Allow read/write/list to Transient bucket
  + Allow KMS encrypt/decrypt
  + Glue Temp Bucket Access Policy
  + Allow read/write/list to Glue Temp bucket
  + Allow KMS encrypt/decrypt
  + Athena Query Bucket Access Policy
  + Allow read/write/list to Athena Query bucket
  + Allow KMS encrypt/decrypt
  + KMS Policy
  + Allow KMS encrypt/decrypt
  + Cloudwatch Logging Policy
  + Allow Writing Logs to Log Bucket
  + Allow KMS encrypt/decrypt
  + Invoke Lambda Policy
  + Allow invoking lambda function for account
  + Refined Bucket Role
  + Refined Bucket Access Policy
  + Raw Bucket Role
  + Raw Bucket Access Policy
  + Glue Repo Bucket Role
  + Glue Repo Bucket Access Policy
  + Glue Temp Bucket Role
  + Glue Temp Bucket Access Policy
  + Glue Refined Data Crawler Role
  + Allow Glue Service Role
  + S3 Refined Data Bucket Policy
  + Glue Raw Data Crawler Role
  + Allow Glue Service Role
  + S3 Raw Data Bucket Policy
  + Glue Etl Job Role
  + Allow Glue Service Role
  + S3 Refined Bucket Policy
  + S3 Raw Bucket Policy
  + S3 Glue Repo Bucket Policy
  + S3 Glue Temp Buvket Policy
  + Cloudwatch Logging Policy
  + Invoke Lambda Policy
  + Allow Delete Object in all Raw Bucket
  + Allow Delete Object in all Refined Bucket
  + Allow Delete Object in all Glue Repo Bucket
  + Allow Delete Object in all Glue Temp Bucket
  + Cloudwatch Event Role
  + Allow Cloudwatch Events Role
  + Cloudwatch Logging Policy
  + Invoke Lambda Policy
* [Crawler to Scan RAW data tables](file:///Users/panchumarthi/Downloads/datalake/glue-crawler.yaml)
  + Creates a Glue Database
  + Creates a Glue Crawler
  + Creates a Glue Security Configuration for the crawler to enforce SSE with KMS Key
* [Crawler to Scan REFINED data tables](file:///Users/panchumarthi/Downloads/datalake/glue-crawler.yaml)
  + Creates a Glue Database
  + Creates a Glue Crawler
  + Creates a Glue Security Configuration for crawler to enforce SSE with KMS Key
* [Crawler to Scan Transient data tables](file:///Users/panchumarthi/Downloads/datalake/glue-crawler.yaml)
  + Creates a Glue Database
  + Creates a Glue Crawler
  + Creates a Glue Security Configuration for crawler to enforce SSE with KMS Key
* [Crawler to Scan Transient Updates data tables](file:///Users/panchumarthi/Downloads/datalake/glue-crawler.yaml)
  + Creates a Glue Database
  + Creates a Glue Crawler
  + Creates a Glue Security Configuration for crawler to enforce SSE with KMS Key
* [Crawler to Scan Refined Updates data tables](file:///Users/panchumarthi/Downloads/datalake/glue-crawler.yaml)
  + Creates a Glue Database
  + Creates a Glue Crawler
  + Creates a Glue Security Configuration for crawler to enforce SSE with KMS Key
* [Glue Jobs](file:///Users/panchumarthi/Downloads/datalake/glue-job.yaml)
  + Plain ETL Jobs for 10 tables of FIG database (10)
  + CDC job to capture the inserts of FIG database (1)
  + CDC job to capture the updates of FIG database (1)
  + Plain ETL jobs to transform data in transient zone to refined zone (10)
  + ETL job to transform the CDC updates data in transient zone to refined zone (3)
  + Workflow Failure job (1)
  + Refined Workflow Kickoff job (1)
* [Glue Workflows](file:///Users/panchumarthi/Downloads/datalake/glue-workflow.yaml)
  + Processing raw to transient workflow
  + Processing CDC - raw to transient workflow
  + Processing Transient to refined workflow
* [Glue Triggers](file:///Users/panchumarthi/Downloads/datalake.yaml)
  + Raw to Transient workflow kickoff trigger
  + Raw to Transient ETL jobs trigger
  + Raw workflow failure trigger
  + CDC workflow kickoff trigger
  + CDC inserts and updates job trigger
  + Transient crawler trigger
  + Transient - CDC updates Crawler trigger
  + CDC workflow failure trigger
  + Refined workflow kickoff job trigger
  + Refined ETL jobs trigger
  + Refined updates crawler trigger
  + Refined crawler trigger
  + Refined workflow failure trigger
* [FailureSNS](file:///Users/panchumarthi/Downloads/datalake/sns.yaml)
  + SNS for workflow failures
* [Lambda Function](file:///Users/panchumarthi/Downloads/datalake/lambdaNotification.yaml)
  + Starts the DMS CDC task and Processing raw to transient workflow when Full load DMS task succeeded
* [Athena Workgroup](file:///Users/panchumarthi/Downloads/datalake.yaml)

# Deploying to New Environment and Setup

1. Create new AWS Account
   1. Log into new AWS Account and create a non-root user with Administrator access
   2. Log in with new user
2. Clone Git Repo
   1. Note local location of repo - Ex: /home/user/fluent-data-lake
3. Open the /home/user/parameters.json file
   1. Give required parameters information as specified in the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description & Utilization** | **Allowed Values** | **Default Value** | **Required?** |
| CompanyName | S3 bucket naming conventions for the company the stuff is all for. |  | fluent | Optional |
| CodaS3BucketName | S3 bucket where the Quick Start templates and scripts are installed. |  | fluent-datalake-templates-dev-394780878318 | Optional |
| CodaS3KeyPrefix | S3 key prefix used to simulate a folder for your copy of Quick Start assets. |  | data-lake/ | Optional |
| ProjectName | Name of the project for tagging purposes. |  | datalake | Optional |
| Owner | Name of the Owner of the resources for tagging purposes. |  | Jacob Puthuparambil | Optional |
| Environment | On which environment the resources to be created. | dev/ qa/ staging/ production |  | Required |
| VpcCIDR | IP range (CIDR notation) for this VPC | '(().){3}/' | 10.192.0.0/16 | Optional |
| PublicSubnet1CIDR | IP range (CIDR notation) for this Public Subnet 1 | '(().){3}/' | 10.192.50.0/24 | Optional |
| PublicSubnet2CIDR | IP range (CIDR notation) for this Public Subnet 2 | '(().){3}/' | 10.192.40.0/24 | Optional |
| ReplicationInstanceClass | Instance type of Replication Instance. | dms.t2.micro dms.t2.small dms.t2.medium dms.t2.large dms.c4.large dms.c4.xlarge dms.c4.2xlarge dms.c4.4xlarge | dms.c4.2xlarge | Optional |
| EndpointBucketFolder | Prefix of raw bucket where full load data is populated |  | fig\_full\_load | Optional |
| Username | Username of Database |  | awsdatalake | Optional |
| Password | Password of Database |  |  | Required |
| Engine | Engine type of Database |  | mysql | Optional |
| Host | Host address of Database |  | 34.74.194.212 | Optional |
| Port | Port of the Database |  | 3306 | Optional |
| DBName | Name of the Database |  | FILU | Optional |
| RawInputDatabaseName | Name of Raw Zone Database |  | fluent\_dev\_filu\_db\_raw | Optional |
| TransientInputDatabaseName | Name of Transient Zone Database |  | fluent\_dev\_filu\_db\_transient | Optional |
| RefinedInputDatabaseName | Name of Refined Zone Database |  | fluent\_dev\_filu\_db\_refined | Optional |
| EmailID | Email ID for sns to notify glue workflow failure. |  |  | Required |

Note:  
 \* “Optional” parameter indicates that if the parameter value is not passed then default value will be used.

1. Create a S3 Bucket to store CloudFormation templates. Note: \* S3 bucket should be in the same region where Cloudformation stacks need to be deployed.
2. Create a secret that stores details of Source Database. The secret should contain:

|  |  |
| --- | --- |
| **Key** | **Value** |
| Username | awsdatalake |
| Password |  |
| Host | 34.74.194.212 |
| DBName | FILU |

1. Create a new user with programmatic access for deploying stacks using CircleCI. Note the Key values to store them as environment variables.
2. Open CircleCI dashboard and link fluent-datalake project.
3. Go to Project Settings> Environment Variables, add the following environment variables to the project

|  |  |
| --- | --- |
| Key | Description |
| AWS\_ACCESS\_KEY\_ID | Access Key value of the user you have created before. |
| AWS\_SECRET\_ACCESS\_KEY | Secret Key value of the user you have created before. |
| AWS\_DEFAULT\_REGION | Region where the stacks to be deployed. |
| AWS\_SECRET\_NAME | Name of AWS Secret that has the details of Database |
| AWS\_SECRET\_REGION | Region of AWS Secret that has the details of Database |
| STACK\_NAME | Name of Cloudformation Stack |
| SYNC\_BUCKET\_NAME | Name of the bucket where the Cloudformation templates need to be stored |

1. For every push to master branch, CircleCI will create/update the stack.