Name of the Customer: One of the top medical equipment company

AWS Account ID (production)- 666227710600

Problem Statement:

mosaIQ is a data lake platform built on top of AWS stack and stores structure and unstructured data in raw format for Rapid discovery of actionable insights to improve patient care and business outcomes while maintaining security and regulatory compliance.

mosaIQ platform allows the analytics, engineers, and data scientist team to respond with more timely information to support better business decisions.

mosaIQ has been implemented on the AWS Cloud Platform. This platform can provide capabilities to the wider business team communities to better support the business needs.

About the Customer

The customer is a California-based medical equipment company. It primarily provides cloud-connectable medical devices for the treatment of sleep apnea (such as CPAP devices and masks), chronic obstructive pulmonary disease (COPD), and other respiratory conditions. It also provides software to out-of-hospital care agencies to streamline transitions of care into and between these care settings for seniors and their care providers (i.e. home health, hospice, skilled nursing facilities, life plan communities, senior living centers, and private duty).

The customer employs more than 7,500 employees worldwide as of October 2019. [2] The company operates in more than 120 countries worldwide and has manufacturing facilities in Australia, France, Singapore and the United States. It achieved revenues of US$2.6 billion in the fiscal year 2019.

Customer Challenge

High-level requirements

- myAir data is properly D-Identified and routed to the correct accounts (PHI, NON-PHI)
- myAir data is available and accessible in end location in the expected format

Detail requirements

- Maintain History for all Updates from before Merge layer
- Merge layer is Latest updated dataset
- Data Download and sharing should not be done.

Software development requirements

- The development workflow is documented and followed
- Unit tests are developed and integrated into CI/CD for all code
- All code should be able to run locally in some manner
- The code Review process is documented and followed
- Terraform deployment for all Resources and code artifacts
- Follow the customer naming convention

Why AWS

Amazon Web Services provides on-demand cloud computing platforms and APIs which are reliable, scalable on a metered pay-as-you-go basis. It provides the necessary features which exactly fits the customer's requirement to run their business.

The customer has applications running in their on-premise Data Centres and they decided to go with AWS since it provides low-cost migration services of the existing infrastructure to the AWS Cloud.

Along with that, AWS provides resources to aid in expansion and utilize more of their services for a long term basis with an appropriate Savings plan or reserving the capacity.

AWS is much more secure than a company hosting its website or storage. It currently has dozens of data centres across the globe which are continuously monitored and strictly maintained. The diversification of the data centres ensures that a disaster striking one region doesn't cause a permanent data loss worldwide.

Why the Customer Chose the Partner

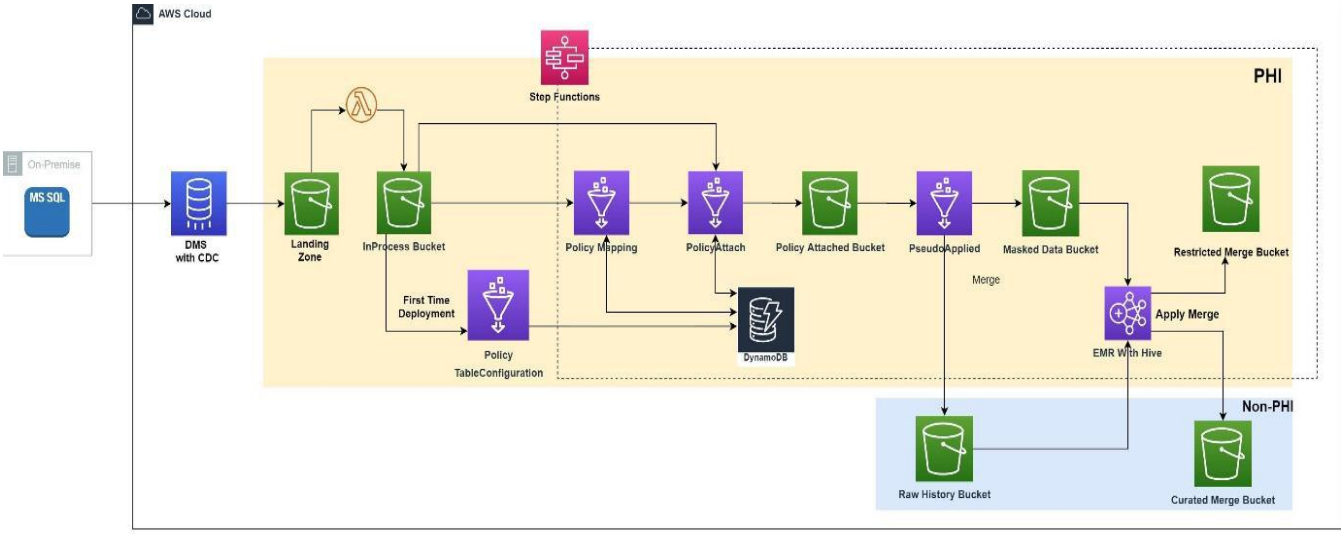
Powerup is a cloud technology company with product & services in cloud migrations, managed cloud and AI solutions. A Premier Consulting Partner of AWS, Powerup provides product-led consulting solutions to enterprise customers on the public cloud platforms.

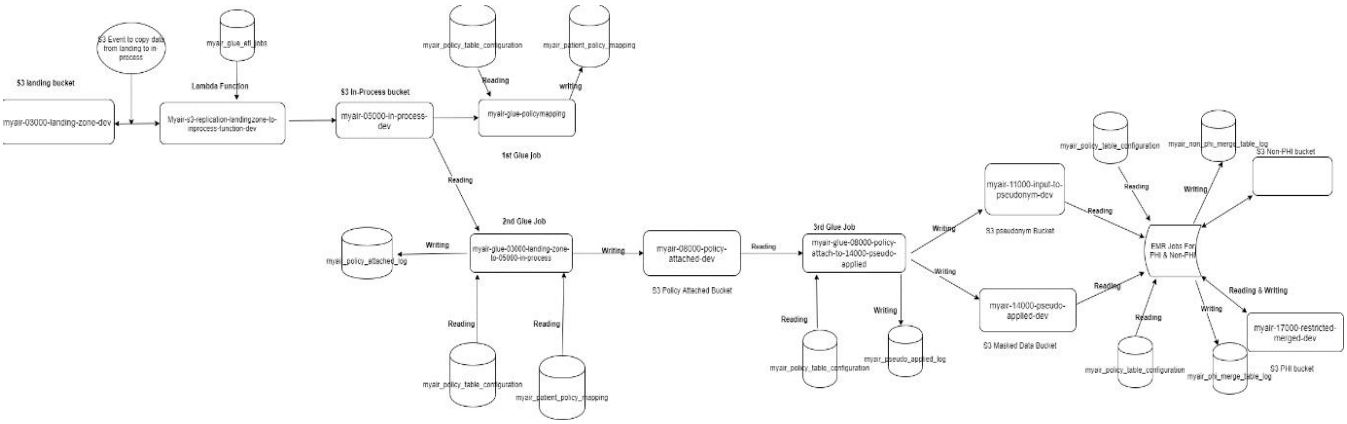
Powerup has gained immense popularity over the last few years and successfully executed over 150 projects in cloud transformation. The customer was convinced that Powerup would be the best option who could deliver the services and fulfil their business requirement. It stood as a market leader in the Cloud Migration sector and managed to setup a strong customer base through its successfully executed projects. This encouraged the Customer to move ahead with Powerup for assisting in the Cloud Migration Program and managing its services both in Cloud and on-premise.

Technologies used

AWS S3, Lambda, AWS Glue, AWS EMR, AWS DynamoDB, AWS Step Function, AWS CloudFormation, AWS DMS + CDC

Architecture Diagram

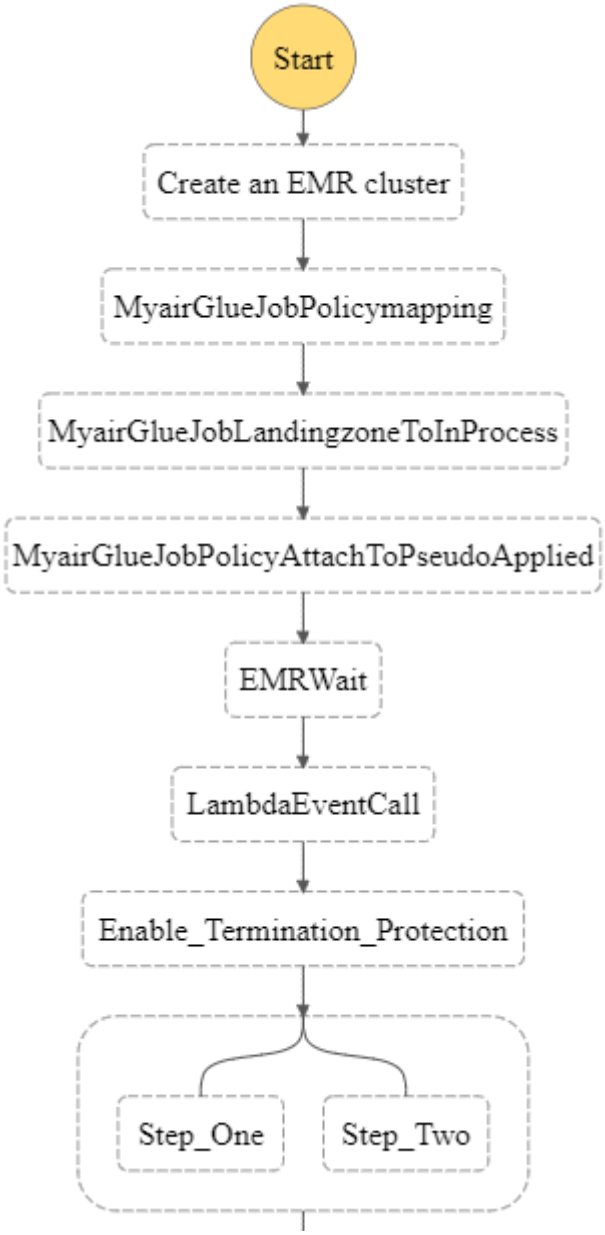


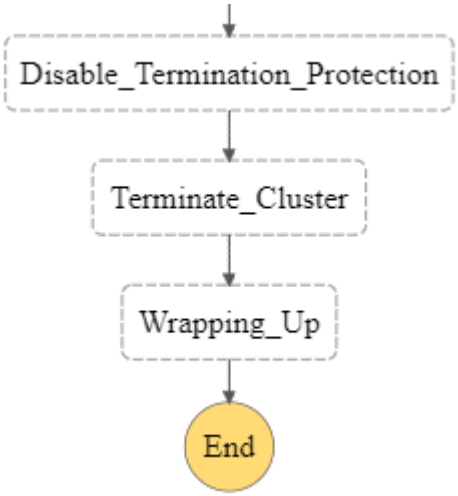


DMS will keep pushing the change data into S3 landing bucket, with all file inside single folder per table.

- An S3 event-based lambda function has will be setup to copy the file from landing bucket to in-process bucket with date and hour folder structure. The lambda will read the date and hour value from the dynamo DB table which will get updated as soon as the first glue job start and any new file will go to the new date and hour folder: myair-05000-in-process-dev/Active/Account/p_date=2018-01-01/p_hour=12
- The first glue job will be used to build the policy table and write it to DynamoDB table. The table will have patientid, deviceid and allowenhanceanalytic as main columns, which will be used to define the policy across table.
- Second glue job policy-attached will read the data from the in-process bucket and add two extra columns license_agreement_accepted_flag and license_agreement_policy_date. mark the records which flag 1 and 0 to segregate the patients into phi and non-phi
- Third glue job Pseudo-applied, will read the data from policy attached bucket and apply the masking to defined columns and write it to the pseudo applied bucket.

lStep function Design considerations





1. IAM role for Step function
2. The service role created for the Step function should have permission only to the Glue Job, EMR job and SNS topics.
3. There must be an error handling mechanism and notifications for each task and step involved
4. Handling inputs and output for the Step function is easier in the Step function
5. Use Timeouts to Avoid Stuck Executions
6. Since we have long-running jobs always use **Standard** Workflows

ASSUMPTIONS

- Stated (02-April-2020) The Stepfuntion Can not call Cross Account StepFunction/EMR/GLUE/Lambda jobs
- EMR will be an async call which will be checked by lambda function if the EMR is ready
- Suggested to use Lambda with an async call (https://docs.aws.amazon.com/lambda/latest/dg/API_Invoke.html#API_Invoke_RequestParameters)
- StepFunction will Run in PHI Account
- cloud watch event CodePipeline will trigger this Step-function
- EMR cluster to be in PHI account

Steps:

1. Start with EMR async call
2. call all glue jobs, with stepwise, All Glue to have SNS for failure and success.
3. call the waiter for 5 min to wait pause the stepfunction
4. call lambda function to check the EMR creation
5. Enable termination protection for EMR
6. Than trigger EMR steps parallelly
7. Disable Termination Protection
8. Terminate EMR
9. Wrapup

Lessons Learned

- To Setup cross-account Step Function Orchestration across Glue and EMR.
- To setup 2 step job with single EMR cluster setup.
- Setup DMS CDC data having transactionID populated as NULL, and to handle that.