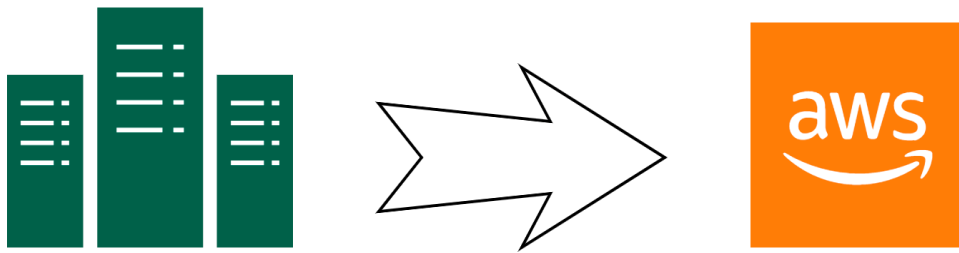


Migrating an On-Prem Application to AWS



Proof of Concept (PoC)

Version: 1.1

Date: 03-July-2022

Author: POC2 Team

Table of Contents

1	OVERVIEW	3
1.1	DOCUMENT PURPOSE	3
2.	PROJECT OR SOLUTION INFORMATION.....	4
2.1	OVERVIEW OF THE PROJECT	4
2.2	OBJECTIVE AND PURPOSE OF THIS POC	4
2.3	SCOPE OF WORK	4
2.4	OUT OF SCOPE	7
2.5	SUCCESS CRITERIA.....	7
2.6	PARTICIPANTS, ROLES AND STRUCTURE.....	7
2.7	RISK MITIGATIONS	9
2.8	MIGRATION CONSULTANT COST	10
2.9	TIMELINES	11
2.10	ASSUMPTIONS	11
3.	SOLUTION SUMMARY	12
3.1	APPLICATION DETAILS	12
3.2	CLOUD STRATEGY OVERVIEW	12
3.3	APPLICATION MIGRATION PROCESS.....	13
4.	PROPOSED DELIVERY	14
4.1	OVERVIEW	14
4.2	ARCHITECTURE OVERVIEW	14
4.3	ARCHITECTURE DIAGRAM	15
4.4	CONNECTIVITY REQUIREMENTS	17
4.5	IMPLEMENTATION PLAN	17
5.	RESULTS.....	17
6.	CONCLUSION	17
7.	TERMINOLOGY AND REFERENCES	17
8.	SUPPORTED DOCUMENTS	18
9.	TEMPLATE VERSION HISTORY	18
10.	CONTACT DETAILS	18

1 Overview

1.1 Document Purpose

This document presents the proposal for migrating an on-premises application to AWS cloud as per the client's requirement. The application is a pricing application that receives real-time stock prices from exchanges through a data vendor. Once the stock price goes through, the proxy hits Feeder's interface application.

The feeder and pricing application will run 24/6 and any maintenance activities will be conducted on Sunday. The pricing application and feeder application will be moved to the cloud and will continue to receive data from the same data vendor. The cloud hosted applications will benefit from high availability, scalability, security, reliability and reduced admin overhead.

This document will cover the cloud strategy for migration, migration plan, cost, time and effort and the scope of managed services provided by the consultant for the period of two years.

2. Project or Solution Information

Migrating On-prem applications to public cloud platform (AWS)

2.1 Overview of the project

Currently users are accessing **pricing application** which is running on-prem and this project is to migrate this workload into AWS.

2.2 Objective and purpose of this POC

The key objective of the PoC is to design the solution for migrating on-prem application to AWS. This migration project is going to be cost effective and least administration overhead (Infrastructure administration).

2.3 Scope of work

Sr. No	Tasks	POC2 Team	Client	Timeline
1	Prerequisites: <ul style="list-style-type: none"> Identifying SPOC from both Teams Access to AWS account On-Prem Infrastructure analysis and Application discovery. 	RA	CI	Week 1
PROD SETUP				
2	Configuring Networking for PROD: <ul style="list-style-type: none"> Configure/validate PROD VPC & subnets for relevant access and connectivity for the Pricing and Feeder app deployment. Configure the Routes Table, Internet gateway, Transit Gateway, VPN connections as required. Create ACL and Security groups with required rules. 	RA	CI	Week 1
3	Load Balancer Setup for Prod: <ul style="list-style-type: none"> Create required target groups and add the targets as Instances 	RA	CI	Week 1-2

	<ul style="list-style-type: none"> Create the External and Internal ALB and associate the respective target group for the Prod environment Open the respective port and listeners as per the old setup Validate the URL 			
4	<p>Creation of Feeder Application:</p> <ul style="list-style-type: none"> Create EC2 instances and configure the feeder application on it Configure Auto Scaling for the EC2 instances with scaling policy based on load in multi AZ. 	RA	CI	Week 1 & 2
5	<p>Creation of the App tier of Pricing App:</p> <ul style="list-style-type: none"> Create EC2 instances and configure the Pricing application on it Configure Auto Scaling for the EC2 instances with scaling policy based on load in multi AZ. 	RA	CI	Week 2
6	<p>Database Deployment for PROD:</p> <ul style="list-style-type: none"> Create MS-SQL DB on EC2 Machine Take a backup of existing Database and restore the DB Test app functionality with database 	RA	CI	Week 2
7	<p>Establish connection to Data Vendor and Proxy:</p> <ul style="list-style-type: none"> Test VPN connection with Data vendor and proxy server 	RA	CI	Week 2
8	<p>Connectivity Check between all layers:</p> <ul style="list-style-type: none"> Check the connectivity between the Web,App and DB layer Verify the functionality of the application. 	RA	CI	Week 2
9	<p>Cutover:</p> <ul style="list-style-type: none"> Plan cutover activities for Applications and MS-SQL databases. Create pre-cutover, cutover and postover plan Execute the cutover on planned date 	RA	CI	Week 3
10	<p>Monitoring:</p> <ul style="list-style-type: none"> Installing Cloudwatch monitoring agents to Monitor EC2 Instance resources utilization 	RA	CI	Week 3

	<ul style="list-style-type: none"> Configuring alerts for VMs on basis of CPU, Disk and RAM usage Configuring alerts for service uptime issues Send Cloudwatch logs to S3 bucket for storage Enable Cloud Trail logging (if required) 			
11	Final Testing & Validation	CI	RACI	Week 3
DR(Disaster Recovery) SETUP				
12	Initial DR Setup: <ul style="list-style-type: none"> Creating Infrastructure, Networking, Application, Database and Monitoring setup as similar to Primary Zone 	RA	CI	Week 3-4
13	Failover with Primary <ul style="list-style-type: none"> Creating Route53 Failover policy with Primary setup as Primary and DR as Failover setup Performing Failover Tests and Validation between Primary and DR site 	RA	CI	Week 4
14	Final Validation and Handoff <ul style="list-style-type: none"> Final End to end validation of setup with Client Team and Sign off 	CI	RACI	Week 4

RACI Definitions

Responsible	R	Responsible for performing the task
Accountable	A	Accountable for making the business decision or delegating specific tasks to other teams
Consulted	C	Consulted for inputs and feedback; however, agreement or action on input is not required
Informed	I	Informed of the final result, task completion, and/or deliverable distribution

2.4 Out of Scope

- Any issues related with applications functionality
- Adding new plugins, software or functionality not already included
- Maintaining historical data of pricing application
- Connectivity to on-prem server/storage post migration
- Sign-off from different Team for any future rollback
- Support for any future platform migration plan
- Implementing third-party logging and monitoring solution

2.5 Success criteria

- Users will be able to login into the application without any issues
- Data validation can be checked after the migrations
- Network latency of the application will be considered very low (improved network latency compared with on-prem)
- Continuous managed services for 2 years
- Successful Terraform script execution

2.6 Participants, Roles and Structure

Name	Title	Responsibilities	Team
Sanyam Surana	Project Manager	-Management of the complete migration lifecycle -Create and meet deadlines -Keep project under scope and budget -Make sure there is clear communication b/w application and migration teams	Project Management
Samia Shafique	Solution Architect	-Discover and asses the on-prem app -Decide and implement migration strategy -Architect the design for application on AWS	Migration and Managed Services Management
Logeswaran Vinyagam	Presales consultant	-Present client with proposal and relevant documents -Point of contact for further enquiries -Assist in architectural solution	Migration and Managed Services Management
Amiya Pradhan	Senior cloud engineer	-Implement migration plan -Create and deploy app infrastructure in the cloud -Provide managed services	Migration

Sameer Khanna	Senior Cloud Engineer	<ul style="list-style-type: none"> -Ensuring High Quality Maintainance of AWS Environment -Troubleshooting any Potential issues with Infrastructure and AWS Environment -Priority Resolution of Incidents based on severity -Ensuring Regular Cadences, Reporting and Team Mangement 	Managed Services
Nitin Jagwani	Senior Cloud Engineer	<ul style="list-style-type: none"> -Ensuring High Quality Maintainance of AWS Environment -Troubleshooting any Potential issues with Infrastructure and AWS Environment -Priority Resolution of Incidents based on severity -Ensuring Regular Cadences, Reporting and Team Mangement 	Managed Services
Srinivas Raju	Senior Cloud DBA	<ul style="list-style-type: none"> -Implement migration plan -Create and deploy app infrastructure in the cloud -Provide managed services 	Migration
Teddy Mpoyi	Cloud Engineer	<ul style="list-style-type: none"> -Implement migration plan -Create and deploy app infrastructure in the cloud -Provide managed services 	Migration
Sudhakar Dhavala	Cloud Engineer	<ul style="list-style-type: none"> -Implement migration plan -Create and deploy app infrastructure in the cloud -Provide managed services 	Migration
Tapender Kamboj	Cloud Engineer	<ul style="list-style-type: none"> -Ensuring High Quality Maintainance of AWS Environment -Troubleshooting any Potential issues with Infrastructure and AWS Environment -Priority Resolution of Incidents based on severity 	Managed Services
Francis Yeo	Cloud Engineer	<ul style="list-style-type: none"> -Ensuring High Quality Maintainance of AWS Environment -Troubleshooting any Potential issues with Infrastructure and AWS Environment -Priority Resolution of Incidents based on severity 	Managed Services
Subhadip Chanda	Cloud DBA Engineer	<ul style="list-style-type: none"> -Ensuring High Quality Maintainance of AWS Environment -Troubleshooting any Potential issues with Infrastructure and AWS Environment -Priority Resolution of Incidents based on severity 	Managed Services

2.7 Risk Mitigations

Theme	Risk	Mitigation
Security	Insecure REST API calls can be a security issue	Encryption of data both in transit and at rest should be done at application level. AWS Key Management service is already used which can further be utilized for encrypting REST API as well.
	Data Security and Protection	Training to be provided to the team handling or responsible for data. Services like KMS, Cloudtrail, Guard duty are used for the same. Encryption algorithm AWS AES-256 can be used here to encrypt the data.
	Misuse of IAM	Using the concept of minimum access should always be considered
Performance between Pricing Application and Feeder Application (Internal)	Added latency since VPN connectivity is shared and unpredictable	Direct Connect can be considered in this case. This will increase cost but will have consistent and higher network performance.
Performance for user accessing Pricing application Web service (External)	Average round trip traffic is more than 120 seconds resulting in high latency. Users from different countries complain that the page's response is very slow, and the website is taking time to load.	AWS CloudFront is used to deliver content through edge location minimizing latency.
Data	Data being lost during migration	Frequent backup of data would be taken before and during migration
Application/Patching	Internal application dependency can cause issues in application when moved to cloud	Application developer assigned/available on contact
	Patching of server/application can cause downtime. Security Patching and Security Policies of newly provisioned application instances on AWS cloud are weak. Lack of visibility on the security posture of applications running on AWS cloud. No patching process exists. Vulnerabilities could exist due to inadequate configuration if software updates across all infrastructure and software components.	Patching would be done only on the agreed day in a week. The customer and client would be informed for such downtime. IAC would also help in this process which will enable it to configure the server freshly. Also, AWS Systems Manager could also be considered.
Network Security	Potential Risks around Cross-site Scripting	AWS WAF is used
	Crawler BOTS and Attack BOTS from hackers extracting information from application leading to DDOS in both on premise and cloud	AWS Shield Standard is used
Secrets / Password Protection	Security credentials are exposed on the application. Hackers can query database using certain search fields in the website	Use AWS Secrets Manager is used
Secrets Protection on Client	Users / Applications can gain unauthorized access on the Pricing application. Privileged Accounts are not properly managed increasing the risk of security incidents.	AWS IAM roles can be used to control the access
Encryption Keys Generation / Storage / LifeCycle / Rotation	No one to manage and generate Encryption keys. The generated keys are no of industry recognized level. Customer is not aware of where the	AWS Managed KMS is used here. Automatic key rotation can also be enabled here which rotates the key every year (only valid for customer managed keys).

	keys are stored. Plaintext access to the keys are not restricted. Timelines when the keys should be rotated are unknown.	
Monitoring and Logging Mechanism	No monitoring and logging present for this pricing and feeder application, Unauthorized access or changes not detected until business gets impacted. No access logs making it difficult to establish evidence of malicious patterns.	AWS Cloud Watch, Cloud Trail, Guard Duty is being used to help detect patterns within and across applications. With the help of application developers the same can be used in pricing applications as well.

2.8 Migration Consultancy Cost

Resource	Number (N)	Hourly Rate (R)	Number of Weeks	Total Cost(Number * Rate * Total Hours)
Solutions Architect	1	\$65	2	5200
Presales Consultant	1	\$50	4(2hrs/week)	400
Senior Cloud Engineer	1	\$40	4	6400
Senior Cloud DBA	1	\$40	2	3200
Cloud Engineer	1	\$30	4	4800
Total Cost for 4 weeks =				\$20,000

	No Of Hours per Resource			
Phase	Solution Architect	Sr Cloud Engineer	Sr Cloud DBA	Cloud Engineer
Discovery & Planning	5	20	10	20
Migration Architecture	20	20	10	20
Infrastructure Setup	10	40	20	40
Migration Plan	15	20	10	20
End to End Validation	10	30	20	40
People Skills	20	30	10	20
Total No Of Hours	80	160	80	160

Managed Services

Resource	Number (N)	Hourly Rate (R)	Based on Severity of Ticket	Total Cost(Number * Rate * Total Hours)
Solutions Architect	1	\$65	Ad Hoc Basis	–
Presales Consultant	1	\$50	Ad Hoc Basis	–
Senior Cloud Engineer	1	\$40	–	320,000
Cloud DBA Engineer	1	\$40	–	160,000
Cloud Engineer	2	\$30	–	24,000
Total Cost for 2 Years = 500(approx working days) * Rate/hr * 8 Hours				\$720,000

Additionally, the First 50 P1 tickets will be done Free of charge by the partner.

2.9 Timelines

Sprint	Week	Description
Sprint 1	Week 1	Application Discovery and Assessment
	Week 2	Design and prepare for Migration (LLD)
Sprint 2	Week 1	Setting up the infrastructure in AWS using Terraform
	Week 2	Validate and cutover

2.10 Assumptions

ID	Statement
1	The on-prem app is hosted on couple of servers
2	The connection between the on-prem app and AWS is going to be established through a site-to-site VPN
3	The client has requested to use a CIDR range of 10.0.0.0/24 for the VPC
4	The client has security and networking teams to manage the on-prem application
5	Access through AWS console will be restricted

6	The consultant will be providing cost optimisation, monitoring services and security enhancement solutions in the managed services
7	Client owns the responsible on cost of load balancing in case huge unpredictable traffic
8	Customers have to pay for the exchanges or whatever real-time prices they are going to be fetching from the data provider via Feeder app

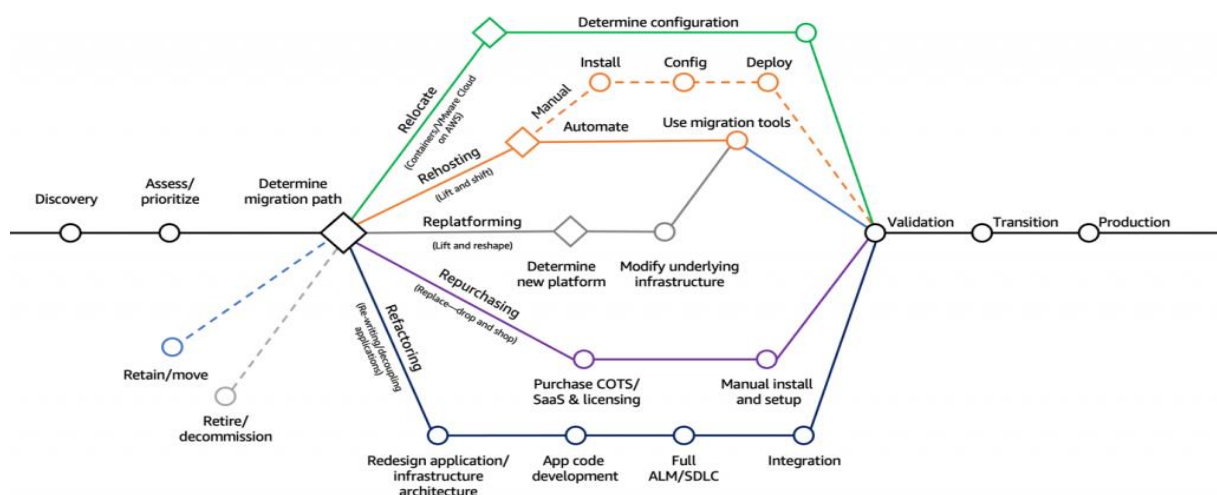
3. Solution Summary

3.1 Application details

Currently user accessing the application from on-prem servers and application hosted on the same servers. Users accessing pricing application to get the real time data for the financial instruments (Ex : Stock price of NASDAQ) and this application is integrated with internal feeder application which will get the real time data from outside company data provider using proxy.

3.2 Cloud strategy overview

When considering the move to AWS there are typically seven cloud strategies that are taken into consideration. To elucidate why we chose a specific strategy out of the seven we will highlight them all below.



1. **Rehost (lift and shift)** – Move an application to the cloud without making any changes to take advantage of cloud capabilities.

2. **Replatform (lift and reshape)** – Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities.
3. **Refactor/re-architect** – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability.
4. **Repurchase (drop and shop)** – Switch to a different product, typically by moving from a traditional license to a SaaS model
5. **Relocate (hypervisor-level lift and shift)** – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. This migration scenario is specific to VMware Cloud on AWS.
6. **Retain (revisit)** – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there's no business justification for migrating them.
7. **Retire** – Decommission or remove applications that are no longer needed in your source environment.

After careful analysis and evaluation of the pricing application our migration team decided to go ahead with the replatforming migration strategy. Because it fits the given criteria mentioned by the client.

3.3 Application migration process



The key steps of this migration process are:

1. **Application Discovery** where relevant data and information about the application is collected.
2. **Assess** the application how ready it is to move to the cloud, choose a suitable cloud migration strategy and create a cloud design.
3. **Build** by implementing the design and integrate with dependencies identified during the assessment phase and with native AWS services.
4. **Validate** the build by performing testing on the target architecture and addressing any gaps or issues that are identified

5. **Cut over** to the application in the cloud by following a well-defined cutover plan (date, time, tasks, and their respective owners) as agreed to by the migration and application teams

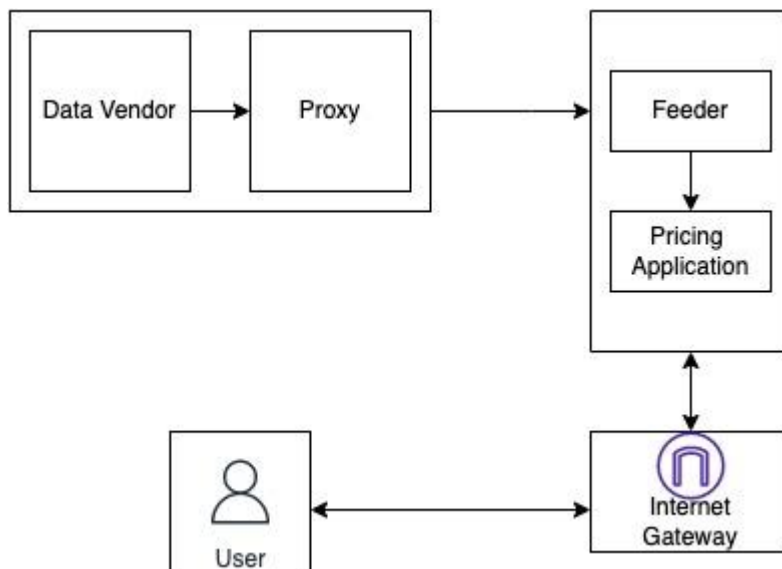
4. Proposed Delivery

4.1 Overview

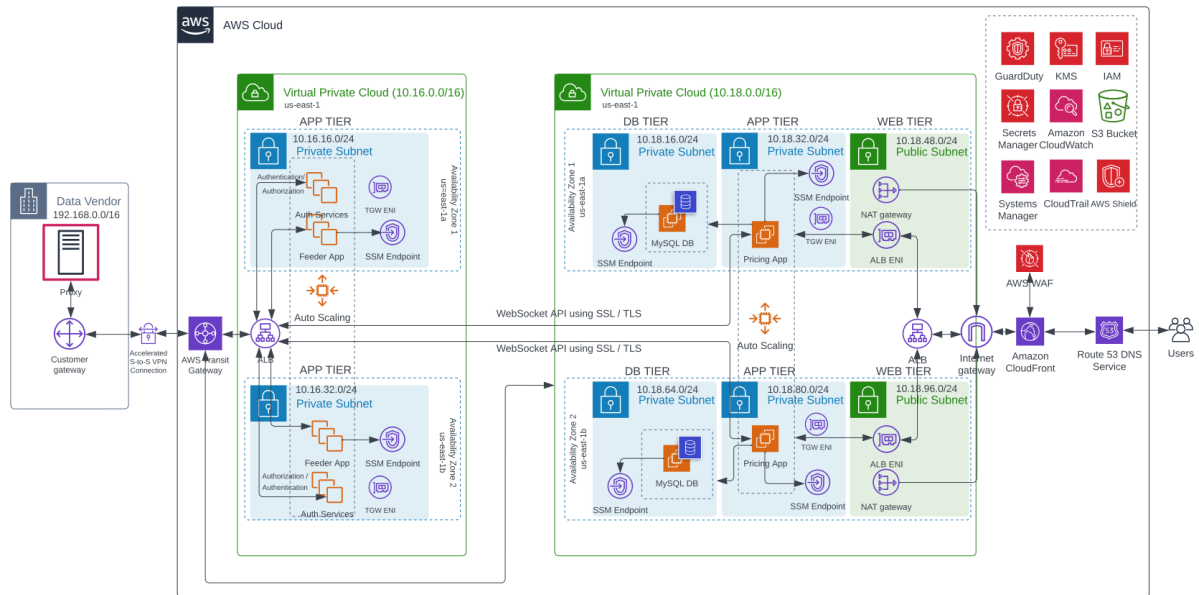
- Utilise the compute, networking and database services of AWS and also automation
- Provisioning/destroy resources achieved by using Terraform
- Managed services for 2 years

4.2 Architecture overview

Current architecture of the existing application is as shown below.

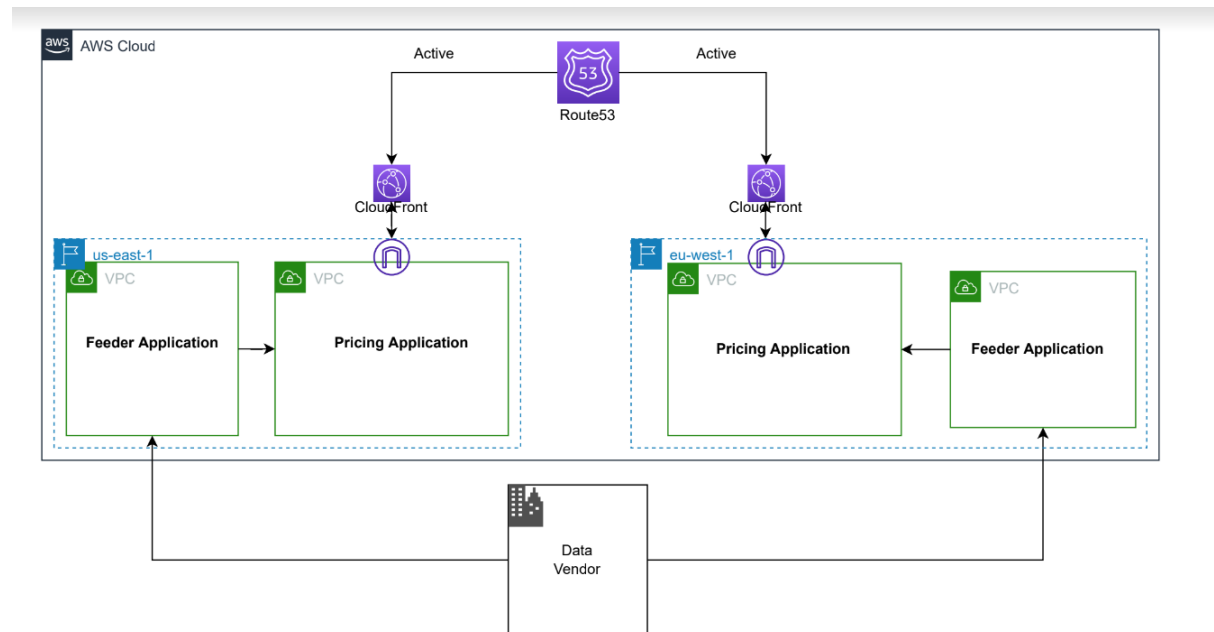


4.3 Architecture diagram



- The application would be deployed in US-EAST-1 (Production)
- 3-tier architecture is used and application/web servers hosted in public subnets and database hosted on private subnets
- The DR setup will be done in an active - active configuration since the availability requirement is 24/6 and it cannot afford downtime.
- Regional failover setup in EU-WEST-1 (Check with customer if any regulatory requirements for hosting outside the country/region)

DR Diagram



4.4 Connectivity requirements

Source	Destination	Protocol/Encryption	Authentication	Port	Environment_Type

4.5 Implementation plan

We will be using Terraform cloud for deploying infrastructure with the following benefits:

- Many Terraform Cloud features are free for small teams
- Stable, remote environment and securely stores states and secrets
- Offers private registry for sharing modules and providers and some other benefits

5.Results

PoC	Environment	Description	Expected Result	Actual Result	Severity

6. Conclusion

To be updated after completion of PoC

7.Terminalogy and References

VPC – Virtual Private cloud

RDS – AWS Relational Database services

VPN – Virtual Private network

TGW – Transit Gateway

IGW – Internet Gateway

KMS – Key management system

S3 – Simple storage service

8.Supported Documents

Document Title	URL / File

9.Template version history

Versions

Version	Date	Author	Description of change
1.0	19-June-2022	POC Team	Initial draft
1.1	03-July-2022	POC Team	Review for approval

Reviewed By

Name	Title	Date
Joe & Team	Head	19-June-2022
Joe & Team	Head	03-July-2022

Approval

Name	Title	Date
Joe	Head	03-July-2022

10.Contact Details

Name	Title	Email address	Mobile Number
Sanyam Surana	Project Manager	Sanyam.surana@poc.com	XXX-XXX-XXXX
Samia Shafique	Solution Architect	Samia.shafique@poc.com	XXX-XXX-XXXX
Logeswaran Vinyagam	Presales consultant	Logeswaran.vinayagam@poc.com	XXX-XXX-XXXX
Amiya Pradhan	Senior cloud engineer	Amiya.pradhan@poc.com	XXX-XXX-XXXX
Sameer Khanna	Senior Cloud Engineer	Sameer.khanna@poc.com	XXX-XXX-XXXX
Nitin Jagwani	Senior Cloud Engineer	Nitin.jagwani@poc.com	XXX-XXX-XXXX
Srinivas Raju	Senior Cloud DBA	Srinivas.raju@poc.com	XXX-XXX-XXXX
Subhadip Chanda	Cloud DBA Engineer	Subhadip.chanda@poc.com	XXX-XXX-XXXX
Teddy Mpoyi	Cloud Engineer	Teddy.mpoyi@poc.com	XXX-XXX-XXXX
Sudhakar Dhavala	Cloud Engineer	Sudhakar.dhavala@poc.com	XXX-XXX-XXXX
Tapender Kamboj	Cloud Engineer	Tapender.kambok@poc.com	XXX-XXX-XXXX
Francis Yeo	Cloud Engineer	Franics.yeo@poc.com	XXX-XXX-XXXX