Greenfield Deployment

Customer: A German multinational Finserv company

Summary

Case - Greenfield Deployment
Number of VM’s - 45
Number of applications - 5
Tools used & AWS Services used - Compute - EC2, EKS, ECR, Lambda, Shared storage - SFTP, EFS, Database - RDS Postgres, Advanced networking - Route53, route 53 resolver, custom DHCP, Security - AWS IAM, Active Directory, CloudTrail, AWS Config, IAAS CloudFormation, Other Services - Terraform, Jenkins with sonarQube, Nexus and Clair

The customer is regarded as global insurance giants of the financial sector. ABS is their consolidated insurance system that they are looking at migrating to AWS along with its supporting applications. They also wanted Powerup to create a Disaster Recovery facility on AWS and make the ABS insurance system available as a backup solution for one of their esteemed banking clients while also catering to a business continuity strategy, automation of applications and security & compliance.

About customer

The Customer is a German multinational, one of the leading integrated financial services company, headquartered in Munich. Their core business caters to offering products and services in insurance and asset management.

Problem statement / Objective

ABS is a monolithic application while the supporting applications are microservices-based. Hence, a microservice architecture which can seamlessly integrate with the customer’s core insurance module was needed.

They wanted Powerup to deploy their applications on production as well as on a secondary (Disaster Recovery) DR facility on AWS using a Continuous Integration (CI)/ Continuous Deployment (CD) pipeline. This was to serve as a Business Continuity solution for one of their esteemed banking clients.

For business continuity, the customer anticipated the Recovery Time Objective (RTO) of less than 4hr and Recovery Point Objective (RPO) of not more than 24 hours.

In addition to infrastructure deployment, all application deployments were requested to be
automated by the client. Being a financial services company, the customer is bound by multiple regulatory and compliance-related obligations for which Cloud Best Security practices were also to be instrumented.

Solution

Project Setup

AWS Landing Zone was set up with the following accounts - Organization Account, Production Account, Dev, Pre-Prod, Management, DR, Centralized Logging & Security Account.

The operational unit consisted of the customer business system (i.e., CISL (Core insurance layer), RAP (Rich Application), MFDB (Core application Database), CTRLM (Batch job automation)) and Non-ABS (Non-customer business system i.e., dispatcher).

All Logs will be centrally stored in the logging account. All management applications like Control-M, AD, Jenkins etc. will be deployed in the management account.

ABS application is deployed across multiple AZs and load balanced using AWS Application load balancer. Non-ABS applications are microservices-based and talk to the ABS running to process or fetch the required data based on request. Close to over 10 microservices are running on Docker within the EKS cluster.

Auto-scaling is enabled at the service level as well as EC2 level to scale out the microservices based on the load. The application uses Active Directory to authenticate.

Solution Highlights

Microservice - Amazon Elastic Kubernetes Service (EKS) backed the highly available, reliable and decoupled API services which are accessible only inside the customer’s private global shared network. Each module is segregated with the namespace.

Build Automation - Jenkins pipelines were used to build automation, Nexus tool to store artifacts and Clair for checking vulnerabilities. Build Artifact Vulnerability management was made easy with the aid of SonarQube.

Active passive Disaster recovery - Actively synchronised AWS Secure File Transfer Protocol (SFTP) is the active directory and private file storage space on the cloud.

Powerup methodically designed and tested a cross-account and cross-region disaster recovery strategy. At the time of live deployment, docker images are tagged (with versioning) and shipped to Amazon Elastic Container Register (ECR) DR account. Encrypted (Amazon Machine
AMIs and Relational Database Service (RDS) snapshots are passively shipped to DR account with a Recovery Point Objective (RPO) of 3 hours.

Custom coupled Lambda functions are used to generate, ship and eliminate encrypted AMIs and snapshots to DR accounts with no human intervention as a backup solution.

**Advanced strategy to ensure the best security** - Custom cloud formation template helped in monitoring AWS API calls made to Change or update configurations of IAM roles / SecurityGroups inbound or outbound rules/ EC2. Granular rules are followed in the AWS config for maintaining and remediating as per regulatory compliance.

**Network Service** – The customer’s network setup was the biggest issue faced. In AWS, the network was completely private, an environment without Internet Gateway (i.e., direct internet access) and Network Address Translation (NAT) because of which a custom Dynamic Host Configuration Protocol (DHCP) option set had to be used to cope up with an existing custom DNS server which was set up in the customer Shared Service account alongside a custom proxy setup for the internet. The most challenging part was registering the worker nodes to Master in EKS as some of the internal kubelet components were failing due to enterprise proxy and custom DNS servers. In order to fix this, AWS private link, route 53 resolver and Kubernetes configMap were fine-tuned effectively.

**Outcome/Result**

- The ABS insurance application was successfully deployed on AWS environment while meeting all security & high availability guidelines as per the stated compliance directives.
During load tests, the application was found to be able to handle 200 concurrent users successfully.

Microservices helped in ‘easier to build and maintain’ application programming interface (API) services. Flexibility and scalability of different API applications were also achieved.

The customer could maintain the lifecycle of identifying, investigating and prioritizing vulnerabilities in code as well as containers without any compromise.

The customer could now implement strong access and control measures and maintain an information security policy.

Tabletop run-through and DR scenario simulations ensured business continuity.