Blackfridge is a pioneering fintech company that aims to provide a fully collateralised GBP stablecoin for institutional clients.

Industry: Finance
Headquarters: UK
Employees: 500+

Challenge

- The most important and essential thing is security.
- Delay in time to market increases the cost of delivering software – need for DevOps method
- Increase agility.
- Application performance measurement, maintaining stability and meeting operational SLA can be made easy by using DevOps.

There were also optimizations in the development process that could be made to enable new features to be released, faster. The development team had a desire to adopt a new methodology to foster good DevOps practices.

Solution

- Design and Implementation of the AWS Landing zone to meet Blackfridge's standards and requirements. Cloud native security tools will keep the infrastructure more secure than the on-premises.
- Provisioning of AWS Core Infrastructure components - AWS accounts, networks, compliance configuration, core IAM identities and roles to meet Blackfridge's workload hosting requirements.
- Network connectivity to Blackfridge's on-premises network based on Blackfridge's security policies and segregation of accounts.
- Ingram Micro Cloud coded the deployment of the infrastructure stack in Terraform to give Blackfridge the capability of deploying the infrastructure in an automated fashion in minutes.
- Finally, Ingram Micro Cloud assisted Blackfridge in establishing code deployment pipelines using Ansible playbooks for orchestration

Results

- Scalability
- Automation
- Better Collaboration
- Streamlined release of feature and updates

Implementing the solution

DevOps Architecture consists of the following components:

Pipeline for Application deployment in UAT environment:

- AWS CodeCommit consist of three branches i.e develop, release and master.
- The developers develop the codes in develop branch and, commit and merge it to release branch.
- The CloudWatch detects the changes and pull the source code and put it into an artifact bucket. CodeBuild downloads the source code into the build environment and then uses the build specification (buildspec), as defined in the build project, or included directly in the source code. CodeBuild compiles the source code, runs unit tests, and produces artifacts that are ready to deploy.
- After performing all the unit test, the CodeBuild upload the code to an artifact bucket, CodeDeploy download the code and deploy it to Docker swarm container in UAT-VPC for testing by the QA/Testing department. After it passes all the checkpoints, then the final code undergoes through the second pipeline

Pipeline for Application deployment in PROD environment:

- The developers take the final codes in release branch and, commit and merge it to master branch.
- The CloudWatch detects the changes and pull the source code and put it into an artifact bucket. CodeBuild downloads the source code into the build environment and then uses the build specification (buildspec), as defined in the build project or included directly in the source code. CodeBuild compiles the source code, runs unit tests, and produces artifacts that are ready to deploy.
- After performing all the unit test, the CodeBuild upload the code to an artifact bucket, CodeDeploy download the code and deploy it to Docker swarm container in PROD-VPC in the production environment where it is accessed by the end users.

AWS Stack: AWS Org, AWS Control Tower, SSO, AWS Service Catalog, VPC, IAM, EC2, CodePipeline, Control Tower, CloudFormation, ELB, S3, Transit Gateway
Tech Stack: PHP, Magento, Apache, Drupal, MySQL