# 

Case study for AWS Infrastructure automation.

Industry: Advertisement

**Geography/Regions served**: Global  
**Engagement size:**

Peak Team size: 7

Duration: 20 Days

Business Problem: Manual process, Lack of Visibility, Collaboration, Traceability and Reports, prone to human error, time consuming.

Business Solution: Implemented cloud provisioning automation.

Business Impact: Achieved Visibility, Collaboration, Traceability and Reports, reduced time.

Tools and Technologies:

Terraform, CloudFormation, Docker

About Client

It is advertisement publishing company. It offer the simplest way to reach people and grow revenue with in-feed, native advertising on Facebook, Twitter, Instagram and Pinterest.

Business Problem

* Complete project environment is created manually on AWS
* It consumes 4-5 business hours
* Person should have good AWS knowledge to create complete stack
* Highly prone to Human Errors.
* IAM policies, VPC-subnet security were missing.
* Versioning of state was not possible
* While destroying whole stack manually, some resources might miss and may costs unnecessarily.

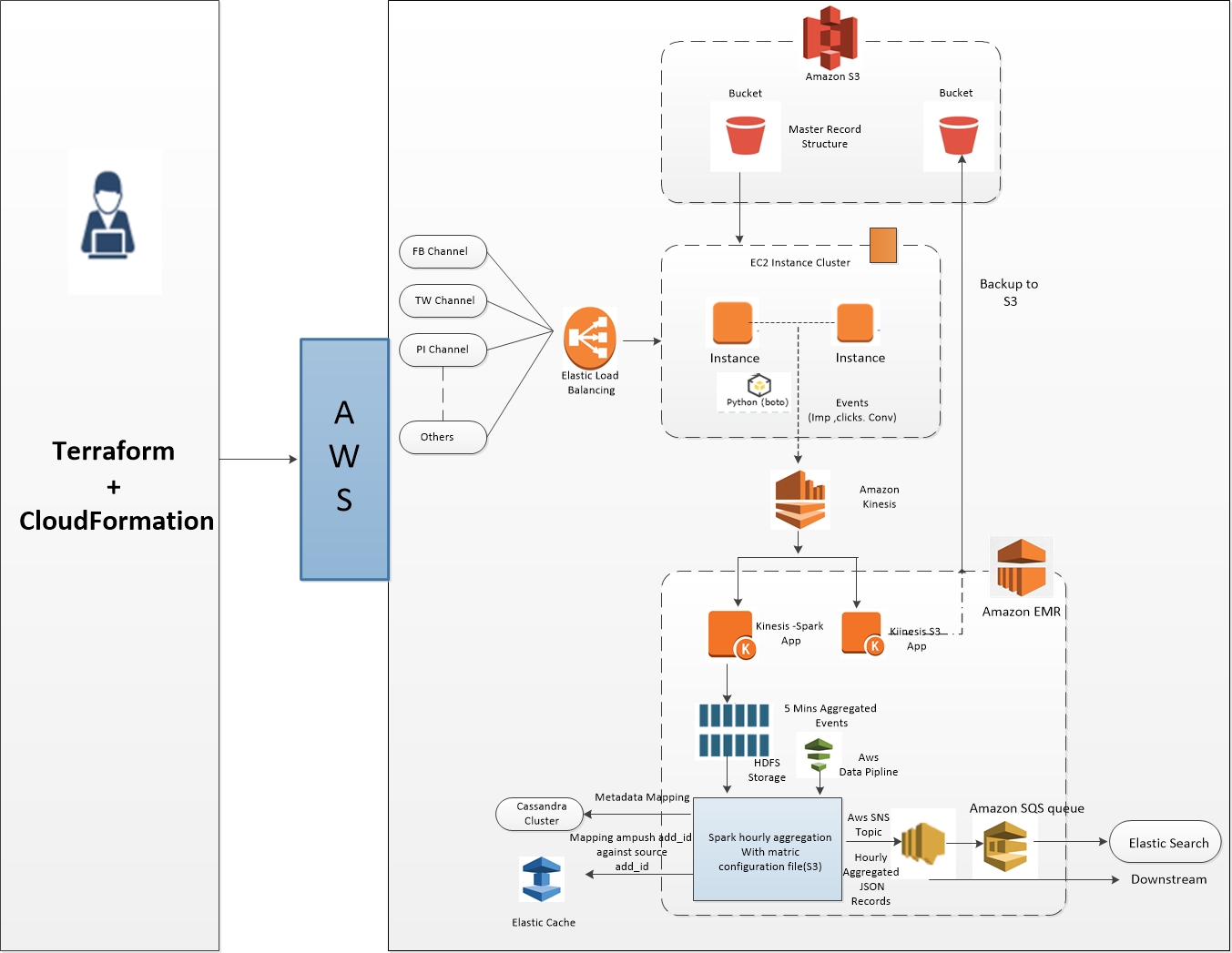
Business Solution

* We suggested Hashicorp’s Terraform and AWS cloudformation tools for automating AWS environment provisioning.
* As Terraform & Cloudformation both can’t not provisioned 100% AWS environment, we suggested to use them in combination. So that we can make 100% automation.
* We also Dockerized KCL app used in Kenesis service and used with ECS service and autoscalling group to maximize log-processing throughput .

Tools and technologies

* Terraform : Cloud provisioning tool by Hashicorp
* CloudFormation : AWS provisioning tool by AWS
* Docker: Containerization

Provisioned AWS Environment



Provisioned AWS Services

* EC2
* ECS
* Auto-scaling Group/policies
* Cloudwatch
* VPC (Subnet, routing table, ACL, gateway)
* IAM (roles, policies )
* SNS
* SQS
* Kenisis Stream
* S3
* Cassandra Cluster
* Elasticache
* EMR Cluster
* Data-Pipeline

Business Impact

* Automated provisioning of all AWS resources.
* Can save current state of AWS environment.
* Environment versioning is possible as it is infrastructure as code.
* Time reduced to 20-25 min.
* Complete environment creation and destruction are one click processes.
* Now anyone can provision complete environment without expert AWS knowledge.