

Architecture migration: monolithic COTS ecommerce (Magento) to microservices-based platform



The Problem

Multiple platforms for multiple sites, highly customized and monolithic code that demanded high maintenance costs. Did not allowed for agile change management and business continuity principles, necessary to allow the volume and velocity of the planned business growth

Additional challenge: how to accomplish this evolution while business is still growing on existing platform



The Solution

Zyzygy approached this project in a progressive fashion, by evolving the new architecture following the baseline design, implementing automation for CI/CD and creation of new services in a rapid manner, and seamlessly integrating some of the new microservices to be leveraged with added value by the legacy system until its sunset to gain value and validate the design hypothesis



The Value

Running both systems in parallel allowed the customer to contrast opportunities for improvement, validate business hypothesis by testing micro service in production and allowed engineering teams to test different options of persistence and homegrown vs native services, making the transformation more natural

CASE STUDY

Architecture migration to microservices platform

Solution Details

Design and implementation of a complete infrastructure framework for a modern business platform on Amazon AWS following security-first principles, focusing on full automation, and leveraging design patterns and state of the art tools for DevOps and Security architecture

Tools and Technologies:

AWS, EC2, EKS, ALB, S3, ElasticSearch, ElastiCache, RDS, DocumentDB, CloudFront, MSK, Jenkins, Ansible, Bash, Python, Docker, Helm, Kubernetes.

- * All the infrastructure is developed and deployed using IaC tools such as Terraform.
- * Created single region VPCs hosting multiple environments using multi-AZ.
- * Created multiple AWS EKS clusters.
- * Created multiple namespaces into k8s to separate environments.
- * Creation of CI/CD pipelines using Jenkins.
- * Created pod autoscaling and worker nodes autoscaling
- * Created a framework to generate helm charts on demand used on the CI/CD pipelines
- * Used consul to manage environment variables in conjunction with configmaps
- * Definition of multiple ALBs
- * Created multiple DBs, RDS-Aurora MySQL, ElastiCache-Redis and documentDB-MongoDB
- * Created multiple ElasticSearch clusters.
- * Created multiple MSK (Kafka) clusters.
- * Created multiple S3 buckets.
- * Configuration of CloudFront.
- * Defined multiple levels of monitoring and alerting using NewRelic infrastructure and APM.

Contextual Diagram of the Solution

CASE STUDY

Architecture migration to microservices platform

Solution Details

Design and implementation of a complete infrastructure framework for a modern business platform on Amazon AWS following security-first principles, focusing on full automation, and leveraging design patterns and state of the art tools for DevOps and Security architecture

Tools and Technologies:
 AWS, EC2, EKS, ALB, S3,
 ElasticSearch, Elaticache, RDS,
 DocumentDB, CloundFront, MSK,
 Jenkins, Ansible, Bash, Python,
 Docker, Helm, Kubernetes.

