

CASE STUDY: Continuous Integration and Deployment pipeline



The Problem

A collection of ecommerce properties running on separate COTS platforms, inherited through acquisitions and implemented by different digital agencies. Lack of a controlled and traceable development process, with no clear demarcation of responsibilities nor release management governance.



The Solution

Zyzygy was tasked with optimizing this ecosystem in a progressive fashion, first by consolidating most properties into a single ecommerce system, for a later migration to a microservices-based platform. Implementation of Infrastructure as a Code, Continuous Integration and deployment pipelines and organized release management



The Value

Multi environment on AWS public cloud, with complete development and release governance, making possible a bi-weekly release cadence with no production downtimes. Increase of 100 % features per release capacity and 90 % reduction of escape defects. Ability to accelerate the development of new platform by 2 months

01 Continuous Integration and Delivery Pipeline

Infrastructure as a Code

Security first principles:

- single region VPCs hosting multiple environments using multi-AZ.

- multiple VPN connections to customer's datacenter

Performance and Reliability Assurance

- Blue-green deployments

- Autoscaling policies based on CPU utilization.

- Definition of multiple ALBs.

Distributed Architecture

- Created multiple DBs, RDS (Aurora MySQL) and ElastiCache (Redis)

- Created multiple ElasticSearch clusters

- Microservices Factory

02

Integrated Automated Testing

Tests at each level, from unit testing to user acceptance integrated into the pipeline based on events and schedule

Automatic audit of coverage and vulnerabilities through SonarQube

03

Multiple Environments for each stakeholder

Development: Used as a first integration environment - constant changes

QA: gatekeeping by Quality team, to accelerate early testing during sprints

STAGE: gatekeeping by customer, in coordination with QA team. UAT and Pre production

PROD: release control system, stable and with full observability

04

Observability, Governance, Integrity, Security and Traceability

Integration of NewRelic and Kibana with multiple levels of APM and Infrastructure monitoring and alerting

Robust release planning process with traceable changes and deployment and changes SOPs

CASE STUDY

Continuous Integration and Deployment

Solution Details

Design and implementation of a complete CI + CD pipeline on Amazon AWS following security-first principles, focusing on full automation, and leveraging modern and state of the art tools for DevOps and Security architecture

Tools and Technologies:

AWS, EC2, ALB, S3, ElasticSearch, ElastiCache, RDS, Jenkins, Ansible, Bash, Python, Docker, Kubernetes, NewRelic, Kibana, DocumentDB.

Contextual Diagram of the Solution

CASE STUDY

Continuous Integration and Deployment

Solution Details

Design and implementation of a complete CI + CD pipeline on Amazon AWS following security-first principles, focusing on full automation, and leveraging modern and state of the art tools for DevOps and Security architecture

Tools and Technologies:

AWS, EC2, ALB, S3, ElasticSearch, Elasticache, RDS, Jenkins, Ansible, Bash, Python, Docker, Kubernetes, NewRelic, Kibana, DocumentDB.

