IMPROVING RELEASE RELIABILITY WITH ARGO ROLLOUTS ACROSS 10K CUSTOMER SERVICES

Adobe

Carlos Sanchez / csanchez.org / @csanchez

Principal Scientist / Adobe Experience Manager Cloud Service

ADOBE EXPERIENCE MANAGER

Content Management System

Used by many Fortune 100 companies

An existing distributed Java OSGi application

Using OSS components from The ASF

Customers can add their own code

AEM ON KUBERNETES

Running on Azure

35+ clusters and growing

Multiple regions: US, Europe, Australia, Singapore, Japan, India, more coming

AEM ENVIRONMENTS

- Customers can have multiple AEM environments that they can self-serve
- Each customer: 3+ Kubernetes namespaces (dev, stage, prod environments)
- Each environment is a micro-monolith ™

SCALE

17k+ environments

100k+ `Deployments`

6k+ namespaces

Already doing progressive rollouts at the environment level

CHALLENGES

How to avoid issues in production

deploying Adobe code / customer code

For 17k+ unique services

CHALLENGES

Full end to end testing is expensive

Does not cover all cases and does not scale

If a few environments fail it requires analysis

- is it an AEM release issue?
- is it a customer code issue?
- is it a temporary issue?

CHALLENGES

It is time consuming

Releases can get delayed

Issues can impact 100% of one environment traffic

SOLUTION: ARGO ROLLOUTS

Canary deployments with automatic rollback

Based on real world traffic and error metrics

Using existing metrics from Prometheus

THE GOOD

Automatic roll back on high error rates

Non blocking rollouts across environments and async investigation

Only a percentage of traffic is affected temporarily

THE GOOD

More frequent releases

Validate with real traffic

More velocity

THE BAD

Migration requires orchestration to avoid downtime

Even using `workloadRef`

A problem with 1000s of services

THE UGLY

Need good metrics

Metrics need to account for `canary`/`stable` labels

What happens with environments with low traffic?

`Rollout` requires changing runbooks, tooling and training

Progressive Delivery is a great idea

Argo Rollouts is a great implementation

Some things to iron out and prepare for

csanchez.org / @csanchez