

Self-Service CI/CD Pipelines to support a diverse application portfolio



Executive Summary

The client is one of the largest financial analytics organizations looking to design and build a continuous integration and continuous deployment (CI/CD) framework for Customer's website and economics applications.

Key Challenge/Problem Statement

The customer wants to replace their on-premises, diverse CI/CD pipelines with cloud-native, standardized CI/CD pipeline patterns to reduce variation in tooling and allow their teams and tools to scale better as the business grows. These pipelines will eventually be available to application teams via a self-service mechanism like AWS Service Catalog, so they must be easily deliverable via such a solution. Given the opportunity to innovate as they transform their DevOps pipelines, the customer also wants their new pipelines to deploy only immutable infrastructure using a blue/green strategy, in contrast to their current deployment processes, which often update long-lived resources in-place.

Proposed Solution and Architecture

The customer's core request was that Vertical Relevance simplify and standardize the existing pipelines, so VR began by surveying the current applications and pipelines and distilling these into three paradigmatic use cases targeting three distinct platforms: AWS ECS, AWS Lambda, and Windows EC2s configured with Puppet.

Three "baseline" pipelines were built to cover these three major use cases. The pipelines were as alike as possible without sacrificing flexibility; this would help meet the customer's need for standardization while also allowing easy adoption by teams with different needs.

VR built each pipeline with standard components from AWS Developer Tools:

- CodePipeline for pipeline automation and notifications.
- CodeBuild to allow for multiple build runtimes and other kinds of variation between applications.
- CodeDeploy to facilitate blue/green deployment for each type of target infrastructure. CloudWatch would capture logs from each step in the pipeline as well as the infrastructure and applications.


The broad steps in each pipeline were essentially the same:

1. The pipeline is triggered when a change to the GitHub repository.
2. CodePipeline clones the "develop" branch of the GitHub repository.
3. The pipeline initiates the CodeBuild project associated with the application type. The name of this project is different depending on which pipeline is being run.
4. CodeBuild uses the buildspec.yml file from the application repository to build the application. Teams customize this file to suit their build (and testing) needs. This stage must publish some artifact(s), including any container image, application binaries, etc. that downstream steps, including CodeDeploy, depend on.
5. When the CodeBuild step completes, the built application artifacts are passed back to the pipeline for deployment by the later CodeDeploy step, either via CodePipeline's artifacts S3 bucket, or via a reference to a Lambda source code object in S3, or a reference to artifacts in the Artifactory artifact store, depending on the type of pipeline.

About the Client

The customer is one of the largest financial analytics organizations in the world.

They have lines of business in financial intelligence and analytics tools, software services, and consulting that produce an annual revenue of over \$3.5 billion.

- 
6. A CloudFormation deployment action is initiated in CodePipeline to create the application's infrastructure resources.
 7. The artifacts communicated to CodePipeline in step (5) are deployed by CodeDeploy in a blue/green style.

Vertical Relevance (VR) used an Infrastructure as Code (IaC) approach for infrastructure management to promote reusability, auditability, and extensibility for cloud solutions, so for this engagement, VR provisioned all pipeline resources using CloudFormation templates stored in GitHub, the customer's version control and collaboration platform.

Implementing Blue/Green Deployments

Blue/Green Deployments to Windows EC2s

The Windows EC2 pipeline seems relatively simple on the surface, not differing much from a typical blue/green EC2-targeted deployment using the CodeDeploy agent, but this engagement presented a few unique challenges due to constraints around the new (green group) EC2 bootstrapping process and the time taken to provision Windows EC2 instances.

Blue/Green Deployments to ECS

Implementing the Windows EC2 pipeline type presented challenges when creating new instances, but since containers are inherently immutable infrastructure and can usually be provisioned in less than a second, the blue/green step was naturally simpler and faster.

Blue/Green Deployments to Lambda

VR leveraged the first-class support for blue/green Lambda deployments for the Lambda pipeline, which kept this pipeline simpler than even the ECS pipeline. One obvious benefit of this pipeline type is the ease with which pre- and post-cutover checks could be performed via built-in CodeDeploy hooks. These hooks are represented in the following diagram via the arrows labeled "Invoke before traffic shift" and "Invoke after traffic shift."

Results

In the end, the Customer had a way to standardized self-service CI/CD pipelines across their enterprise without ignoring the differing needs of their many application teams. The customer's CI/CD infrastructure can now move to the AWS cloud alongside the infrastructure it supports, reducing heterogeneity in tools and required skillsets, which increased scalability of the customer's CI/CD practices and DevOps teams.

Summary

By engaging with Vertical Relevance, the customer moved closer to providing standardized self-service CI/CD pipelines across their enterprise without ignoring the differing needs of their many application teams. Furthermore, the customer's CI/CD infrastructure can now move to the AWS cloud alongside the infrastructure it supports, reducing heterogeneity in tools and required skillsets, which will pay off in the form of increased scalability of the customer's CI/CD practices and DevOps teams.

About Vertical Relevance

Vertical Relevance is a Financial Services focused (Wealth Management, Asset Management, Banking, Insurance) consulting firm helping with the design & delivery of effective transformation programs across people, process, & systems. With 10+ years of AWS & 20+ years of Financial Services experience, we understand the business needs & build solutions to meet sales, marketing, & compliance goals.

