

Continuous delivery for cloud-based digital services

Continuous delivery automates the production path in AWS, Azure or GCP cloud. Code is tested and automated throughout the deployment pipeline from initial requirement to live. Enabling responsiveness to change, user feedback and new security vulnerabilities in a rapid, repeatable way. From 'big-bang' to smaller incremental releases, with zero downtime.

Organisations that currently struggle to respond to the pace of change in today's fast-moving market – or who seek to reduce the time to market for new features and products – will particularly benefit from a move to continuous delivery as they transition to cloud-based technologies and services.

However, the approach is recommended for any organisation seeking to maximise the benefits of digital delivery. Continuous delivery can be transformative for an organisation, enabling rapid response to changing situations and user needs. It reduces the risk of deploying to the cloud by releasing small changes more frequently that are easy to roll back in a repeatable and reliable process. It also provides opportunities for organisations to automate and streamline their internal release processes.

By removing waste in this way, the cost of releasing features and the time to market can be reduced for new digital products and services. Our continuous delivery approach has successfully met Government Digital Service standards in the deployment of multiple services to the cloud.

Features

- Migration of environments to cloud infrastructure
- Can deliver a combined approach of cloud and existing infrastructure
- Cloud expertise and partnerships with GCP, AWS and Azure
- Server automation expertise in Ansible, Puppet, Terraform and Chef
- Continuous integration expertise in GitHub, Jenkins, TeamCity and Travis
- TDD and BDD expertise in JUnit, NUnit and xUnit frameworks
- Expertise in Java, Scala, NET, Ruby, Python, Go and Docker
- Automated functional frameworks, including Selenium and Webdriver
- Support from EE's Continuous Delivery and DevOps teams
- Delivered independently or collaboratively with an existing client team

Benefits

- Improves time to market for new features and functionality
- Decreases risk for deployment (zero downtime)
- Greater responsiveness to change, for example, mitigating new security vulnerabilities
- Automated configuration of all environments across the deployment pipeline
- No single points of failure in the cloud infrastructure
- Faster roll-back of changes
- Resilient set-up to minimise cost
- Auto scaling of new infrastructure resources, as required
- Built-in disaster recovery can respond within seconds
- Automated security patches included in the EE Continuous Delivery service

Our approach to continuous delivery

Equal Experts works with client organisations to automate their path to production by ensuring that all code is fully tested and automated throughout the deployment process – from initial requirement or feature to live. This ensures the organisation can respond to changes, user feedback, market conditions, and new security vulnerabilities in a rapid, repeatable way.

To ensure that continuous delivery is safe, secure, efficient and has the requisite level of buy-in throughout the organisation, Equal Experts uses its 10-step continuous delivery approach at the start of a new or existing initiative. Conducted in close collaboration with the client organisation, our approach ensures we quickly get to an agreement on the customer objectives and our proposed solution. These steps are:

1. Understand deployment objectives, including desired release cycles and deployment confidence levels. For example, once everything has been tested, is the organisation comfortable releasing to live?
2. Analyse the current path to production to understand the “As Is” build pipeline and deployment processes (including use of existing or desired tools), and identify current pain points.
3. Investigate potential blockages in the current process, including considering adequate automated test coverage, commitment of code to source control, code packaging, automation of deployment processes and architecture constraints.

4. Investigate the test approach and automation – including unit testing, static analysis, integration testing, smoke testing, performance testing, security testing, packaging, deployment scripting, versioning, and source control.
5. Investigate ways to reduce unnecessary waiting time between steps in the process via automation.
6. Investigate moving from large releases to incremental ones – from complex, disruptive ‘big bang’ releases to smaller incremental releases, with reduced risk.
7. Investigate deployment confidence via the use of like-environments and automated tests.
8. Establish the development iteration duration – is the current one sufficient to provide adequate feedback?
9. Identify redundant or duplicate tasks throughout the deployment process.
10. Define a “To Be” vision for automated deployment to the cloud within the organisation – including pragmatic recommendations for improvement and implementation.

Following this initial analysis and planning, we will work with client delivery teams to implement the agreed “To Be” vision for automated deployment and continuous delivery within the organisation, ensuring code is tested and automated throughout the deployment pipeline (from initial requirement to live). We aim for the following practices to start demonstrating tangible benefits as soon as possible:

- Continuous improvement in the deployment process and practices
- Reductions in cycle times for new features and increased throughput
- Improved quality of fit-for-purpose software
- Automated build and deployment with zero downtime releases of new features into production, where possible

Continuous delivery ensures cost-effective responsiveness to change, user feedback, market conditions, and new security vulnerabilities in a rapid, repeatable way, and allows frequent, incremental code releases – with zero downtime.

We seek to encourage teams to do as many small releases as possible, for the following reasons:

- Smaller releases mean fewer changes in each release, which translates to fewer opportunities for error, reduced time to identify any issues and consequently less risk
- Practice makes perfect; if things are hard, they should be repeated more often, to make them simpler and more reliable

- It is easier to roll back small changes than big ones, should issues occur
- We want users to benefit from added value as quickly as possible
- We want to maximise opportunities for frequent feedback

Skills & Technologies

These are some of the skills and technologies that we use to implement Continuous Delivery:

Server Configuration Management: Puppet, Ansible, Chef, Terraform, CloudFormation, SaltStack	Virtualisation/ Containers: Docker, VMware, Openstack, Rancher.	PaaS: Cloud Foundry, Heroku	Continuous Integration: Jenkins, TeamCity, Travis, AppVea, Bamboo, GoCD.
Build Scripts and Automation: Maven, Gradle, MSBuild, Python, Ruby	Test Driven Development and Behaviour Driven Development: JUnit, NUnit, DBUnit, HTTPUnit, Cucumber,	Databases: MongoDB, Neo4J, AWS Relational Database Services,	Automated Functional Testing: Selenium, WebDriver, White (for .NET thick client), Appium
Languages: Java, .Net, Ruby, Python, Scala, Go	Cloud: AWS, Azure, Google Cloud Platform, UKCloud	Logging, Monitoring, and Notification: Elasticsearch, Logstash, Kibana, Graphite, Fluentd, Kibana, Prometheus, Grafana. Sensu, Nagios, StackDriver	Orchestration: Kubernetes, GKE, ECS, KOPS

Examples of continuous delivery in action

Equal Experts has worked with many different organisations, including HMRC, to implement continuous delivery services. The path to production is one of the first things we

Making Software. Better.

The information contained in this document is confidential and proprietary. Copyright © 2022-6 Equal Experts. All rights reserved.

look at on any engagement, with a view to determining how it can be achieved in an automated fashion.

At HMRC, Equal Experts has been the key programme delivery partner for the Multi-channel Digital Tax Platform from its outset, at the start of 2013. The aim of the platform is to move approximately 600M+ transactions a year, across 50M+ customers, from traditional channels to digital. Integrated transactional services provide seamless end-to-end experiences for business and individual customers, with self-service applications 'bundled' together to provide services based around life events (eg. a customer setting up a business), and address tax risks.

The sheer scale of the programme – with so many services to provide, so many customers to look after – means the value of continuous delivery has been obvious to all those involved. Our approach at HMRC means services and tools can be tested and deployed independently within a continuous delivery environment, without impacting other production services. Teams are able to release multiple times each week across each of their digital services. We practice zero downtime deployments driven from Jenkins and can rollback using the same mechanism.

To successfully create this continuous delivery environment, a dedicated platform team of WebOps engineers has iteratively built the Platform as a Service, across multiple cloud providers (to provide resilience and disaster recovery). We have provided build, test and live environments that meet stringent security and data compliance regulations, and the monitoring and deployment pipelines that allow an automated, repeatable path to production. This allows over 60 feature teams, spread across multiple locations in different delivery centres, to iteratively release new features.

This example – using best engineering practices to identify and eliminate inefficiencies, and laying the foundations for smoother, more frequent delivery – shows how our approach to continuous delivery can unlock the real potential of delivery teams. It is an ideal way to put real, measurable improvements in customers' hands, on a more frequent basis.