digital.ai™

# DEVOPS, PIPELINES, TOOLCHAINS, AND APPLICATION RELEASE ORCHESTRATION:

# A Short Primer

## OVERVIEW

In the last few years, companies in all industries have woken up to the reality that in order to stay competitive, they need to excel at delivering a good experience to customers online and in apps. And to deliver user-facing software, many moving parts need to work in harmony, work quickly, and most importantly, not break. But what are those moving parts, and how do they fit together? Here's a primer explaining key concepts surrounding software delivery, from DevOps and Continuous Delivery to pipelines, toolchains, and Application Release Automation/Orchestration.

## WHAT IS DEVOPS?

DevOps is a software engineering practice that brings Development and Operations teams and activities closer together to achieve shorter development cycles and more frequent, more reliable software deployments. DevOps breaks down traditional corporate barriers so that Development and Operations teams are no longer "siloed." These groups often work together as a single team that takes responsibility for bringing an application from the developer's laptop all the way to live, successful deployment in production. And if something breaks, they're all on the hook to fix it.

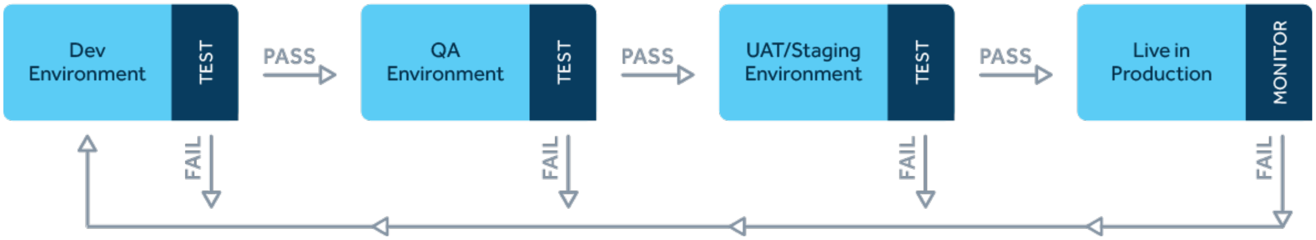
Ultimately, DevOps is about delivering products, features, and services to customers and to the market faster, by improving team collaboration and employing automation throughout the application lifecycle. DevOps practices have rapidly gained wide adoption as a way to optimize software delivery in modern environments.

## UNDERSTANDING CONTINUOUS DELIVERY AND AUTOMATION

Automation of release tasks is a core requirement for DevOps teams. One important thing that automation enables is Continuous Delivery (CD), an approach that focuses on producing software features in short cycles, so any problems that are introduced can be found and fixed quickly. Continuous Delivery principles help Development teams ensure their code is always in a releasable state, which accelerates software delivery and improves overall software quality.

Implementing CD means incorporating automated testing into the development cycle, then moving the latest version of an application through a series of release activities and a progression of different test environments—the software delivery pipeline, also known as the Continuous Delivery pipeline or the release pipeline. These test environments are increasingly similar to what things will look like in production. As the application progresses successfully towards release to customers, it moves from the developer's machine to a test environment, then on to a user acceptance or staging environment, and finally into production itself.

## Software Delivery Stages and Environments

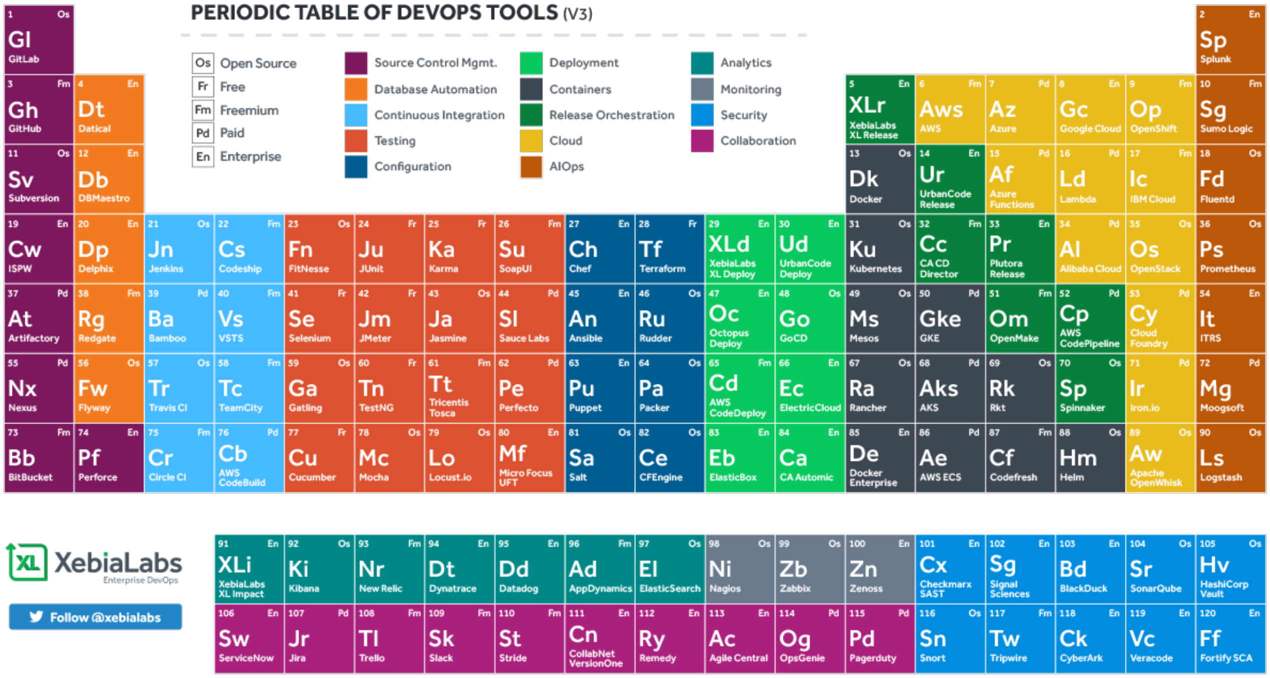


Automatically promoting software changes through a delivery pipeline and flagging failures when they happen are vital aspects of DevOps and Continuous Delivery. Automation significantly reduces the overhead of releasing applications, and it brings down the error rate too. By constantly running tests and promoting only validated releases to the next stage, automation helps ensure that deployment to a live production environment will succeed. It also ensures execution of consistent, repeatable processes. And with today's variety of software environments and expected pace of delivery, there are simply too many steps and too many tests to do them manually!

## THE DEVOPS TOOLCHAIN

In fact, executing all of the functions needed to release an application requires a multitude of specialized tools. Effective DevOps teams carefully build a toolchain of products that carry out their necessary release steps, often providing specialized automation for each type of step. Teams need tools that continuously integrate code into shared repositories as soon as it's written; tools that immediately test these code changes in myriad ways; tools that automatically push new features and changes along the pipeline; tools that manage the environments in the pipeline to avoid configuration drift; and tools that execute deployments in a repeatable, reliable way.

The Periodic Table of DevOps below illustrates the many tools that can go into creating a release pipeline.



PERIODIC TABLE OF DEVOPS TOOLS (V3)

Os Open Source
Fr Free
Fm Freemium
Pd Paid
En Enterprise

🟪 Source Control Mgmt.
🟧 Database Automation
🟦 Continuous Integration
🟥 Testing
🟦 Configuration
🟩 Deployment
⬛ Containers
🟩 Release Orchestration
🟨 Cloud
🟫 AIOps
🟦 Analytics
⬛ Monitoring
🟦 Security
🟪 Collaboration

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 Os<br>Gl<br>GitLab | | | | | | | | | | | | | | | | | 2 En<br>Sp<br>Splunk |
| 3 Fm<br>Gh<br>GitHub | 4 En<br>Dt<br>Datical | | | | | | | | | | | 5 En<br>XLr<br>XebiaLabs XL Release | 6 Fm<br>Aws<br>AWS | 7 Pd<br>Az<br>Azure | 8 En<br>Gc<br>Google Cloud | 9 Fm<br>Op<br>OpenShift | 10 Fm<br>Sg<br>Sumo Logic |
| 11 Os<br>Sv<br>Subversion | 12 En<br>Db<br>DBMaestro | | | | | | | | | | | 13 Os<br>Dk<br>Docker | 14 En<br>Ur<br>UrbanCode Release | 15 Pd<br>Af<br>Azure Functions | 16 Pd<br>Ld<br>Lambda | 17 Fm<br>Ic<br>IBM Cloud | 18 Os<br>Fd<br>Fluentd |
| 19 En<br>Cw<br>ISPW | 20 En<br>Dp<br>Delphix | 21 Os<br>Jn<br>Jenkins | 22 Fm<br>Cs<br>Codeship | 23 Os<br>Fn<br>FitNesse | 24 Fr<br>Ju<br>JUnit | 25 Fr<br>Ka<br>Karma | 26 Fm<br>Su<br>SoapUI | 27 En<br>Ch<br>Chef | 28 Fr<br>Tf<br>Terraform | 29 En<br>XLd<br>XebiaLabs XL Deploy | 30 En<br>Ud<br>UrbanCode Deploy | 31 Os<br>Ku<br>Kubernetes | 32 Fm<br>Cc<br>CA CD Director | 33 En<br>Pr<br>Plutora Release | 34 Pd<br>Al<br>Alibaba Cloud | 35 Os<br>Os<br>OpenStack | 36 Os<br>Ps<br>Prometheus |
| 37 Pd<br>At<br>Artifactory | 38 Fm<br>Rg<br>Redgate | 39 Pd<br>Ba<br>Bamboo | 40 Fm<br>Vs<br>VSTS | 41 Fr<br>Se<br>Selenium | 42 Fr<br>Jm<br>JMeter | 43 Os<br>Ja<br>Jasmine | 44 Pd<br>Sl<br>Sauce Labs | 45 En<br>An<br>Ansible | 46 Os<br>Ru<br>Rudder | 47 En<br>Oc<br>Octopus Deploy | 48 Os<br>Go<br>GoCD | 49 Os<br>Ms<br>Mesos | 50 Pd<br>Gke<br>GKE | 51 Fm<br>Om<br>OpenMake | 52 Pd<br>Cp<br>AWS CodePipeline | 53 Pd<br>Cy<br>Cloud Foundry | 54 En<br>It<br>ITRS |
| 55 Pd<br>Nx<br>Nexus | 56 Os<br>Fw<br>Flyway | 57 Os<br>Tr<br>Travis CI | 58 Fm<br>Tc<br>TeamCity | 59 Os<br>Ga<br>Gatling | 60 Fr<br>Tn<br>TestNG | 61 Fm<br>Tt<br>Tricentis Tosca | 62 Pd<br>Pe<br>Perfecto | 63 En<br>Pu<br>Puppet | 64 Os<br>Pa<br>Packer | 65 Fm<br>Cd<br>AWS CodeDeploy | 66 En<br>Ec<br>ElectricCloud | 67 Os<br>Ra<br>Rancher | 68 Pd<br>Aks<br>AKS | 69 Os<br>Rk<br>Rkt | 70 Os<br>Sp<br>Spinnaker | 71 Pd<br>Ir<br>Iron.io | 72 Pd<br>Mg<br>Moogsoft |
| 73 Fm<br>Bb<br>BitBucket | 74 En<br>Pf<br>Perforce | 75 Fm<br>Cr<br>Circle CI | 76 Pd<br>Cb<br>AWS CodeBuild | 77 Fr<br>Cu<br>Cucumber | 78 Os<br>Mc<br>Mocha | 79 Os<br>Lo<br>Locust.io | 80 En<br>Mf<br>Micro Focus UFT | 81 Os<br>Sa<br>Salt | 82 Os<br>Ce<br>CFEngine | 83 En<br>Eb<br>ElasticBox | 84 En<br>Ca<br>CA Automic | 85 En<br>De<br>Docker Enterprise | 86 Pd<br>Ae<br>AWS ECS | 87 Fm<br>Cf<br>Codefresh | 88 Os<br>Hm<br>Helm | 89 Os<br>Aw<br>Apache OpenWhisk | 90 Os<br>Ls<br>Logstash |
| | | | 91 En<br>XLi<br>XebiaLabs XL Impact | 92 Os<br>Ki<br>Kibana | 93 Fm<br>Nr<br>New Relic | 94 En<br>Dt<br>Dynatrace | 95 En<br>Dd<br>Datadog | 96 Fm<br>Ad<br>AppDynamics | 97 Os<br>El<br>ElasticSearch | 98 Os<br>Ni<br>Nagios | 99 Os<br>Zb<br>Zabbix | 100 En<br>Zn<br>Zenoss | 101 En<br>Cx<br>Checkmarx SAST | 102 En<br>Sg<br>Signal Sciences | 103 En<br>Bd<br>BlackDuck | 104 Os<br>Sr<br>SonarQube | 105 Os<br>Hv<br>HashiCorp Vault |
| | | | 106 En<br>Sw<br>ServiceNow | 107 Pd<br>Jr<br>Jira | 108 Fm<br>Tl<br>Trello | 109 Fm<br>Sk<br>Slack | 110 Fm<br>St<br>Stride | 111 En<br>Cn<br>CollabNet VersionOne | 112 En<br>Ry<br>Remedy | 113 En<br>Ac<br>Agile Central | 114 Pd<br>Og<br>OpsGenie | 115 Pd<br>Pd<br>Pagerduty | 116 Os<br>Sn<br>Snort | 117 Fm<br>Tw<br>Tripwire | 118 En<br>Ck<br>CyberArk | 119 En<br>Vc<br>Veracode | 120 En<br>Ff<br>Fortify SCA |

XebiaLabs Enterprise DevOps

Follow @xebialabs

But with so many tools and activities, how do you keep everything coordinated and in sync?

## MANAGE YOUR PIPELINE WITH APPLICATION RELEASE ORCHESTRATION

With so many tools involved in the release process, teams can't just automate... they need to orchestrate and automate their automation! They need automated ways to manage their complete DevOps toolchain and promote applications through the release pipeline. Application Release Orchestration (ARO) fills that role, overseeing the myriad of tools in the toolchain and orchestrating the overall release pipeline, from planning to production. Think of a symphony conductor keeping her orchestra in sync and playing music in beautiful harmony, and you'll get the right idea.

**Application Release Orchestration orchestrates the build, test, provisioning, configuration management, and deployment tools in your software delivery pipeline to automate the entire release process, from the initial idea of a feature to the ultimate delivery of that feature to the user.**

Application Release Orchestration also automates and orchestrates the many interdependent steps involved in deploying applications to various environments in the pipeline, and it encompasses automatically advancing applications across these environments, from development to testing to staging to production.

## STREAMLINING DEVOPS AND CONTINUOUS DELIVERY PIPELINES WITH ARO

ARO tools are crucial for the release orchestration and deployment automation functions in the DevOps release pipeline—they make sure the right steps happen at the right time using the right settings and tools... and when something fails, they adjust the process accordingly. ARO provides a foundation to unify all of the tools in a Continuous Delivery pipeline, so releases are organized, integrated and streamlined.

The right ARO solution enables teams to make applications environment-independent and to model deployments for maximum reusability and scalability. Along the way, it should support standardization of processes and tasks across the enterprise and deliver the instant auditability that today's businesses require... whether you're using legacy environments, migrating to modern infrastructure in the cloud, containerizing your apps, or taking advantage of a hybrid mix.

With ARO, you can improve collaboration across teams and ensure that everyone has the visibility they need to keep the pipeline flowing. ARO solutions also collect important metrics about completed releases, so you can identify bottlenecks and areas for improvement.

ARO should span your entire real-world software delivery pipeline and provide real-time status information to everyone who is involved in the release process, from developers and operations staff to release managers and compliance officers.

## Effective App Migration Requires a DevOps Framework Designed Using Other Types of Tools for Release Orchestration

For some enterprises introducing release orchestration, the path of least resistance is to repurpose existing tools, such as continuous integration (for example, Jenkins, Circle CI, or TravisCI) and provisioning (like Puppet, Chef, Ansible, and Salt), to try to orchestrate releases. These tools are often already in the pipeline, serve a specific purpose and may have basic functionality to enforce the workflow of release steps. However, using tools that are not designed for pipeline release orchestration tends to have pitfalls.

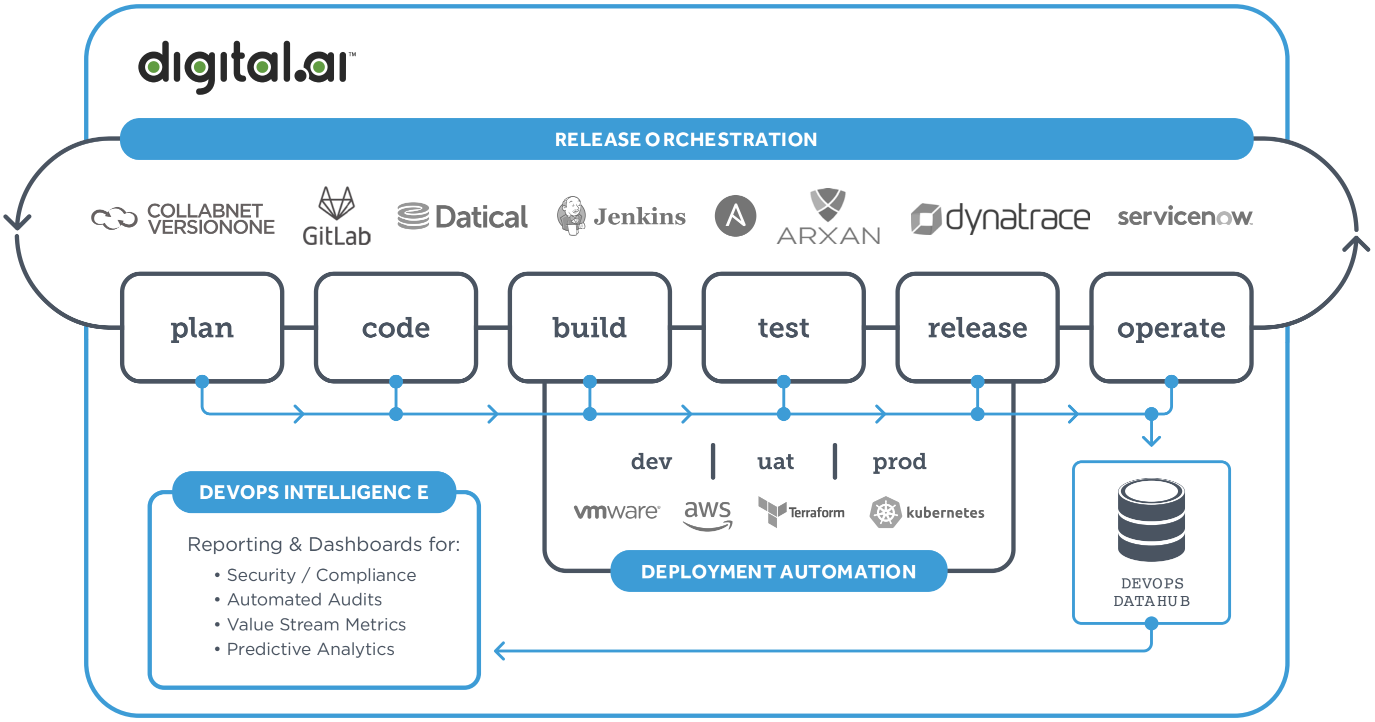
- ✓ Scripting doesn't scale and is difficult to maintain. Extensive scripting is often required to do release orchestration with continuous integration and provisioning tools. But scripts don't model the process and must be manually changed in all places when anything changes.
- ✓ Repurposed technical tools lack visualization, intelligence and reporting capabilities. These insights are necessary in order to truly understand and continuously improve the release process.
- ✓ These tools don't work across the business. They have technical interfaces that don't support the needs of business users, and they don't provide the necessary business- focused views and metrics.
- ✓ They typically lack enterprise-level compliance and security features. Existing tools are not designed to capture audit trails, standardize processes and enforce compliance. And they lack crucial capabilities like role-based access control.

## APPLICATION RELEASE ORCHESTRATION WITH DIGITAL.AI

Digital.ai (formerly XebiaLabs) is an Application Release Orchestration and Automation solution that models releases and deployments so you can easily standardize processes and spread DevOps innovations across the organization.

The Digital.ai Value Stream Platform orchestrates and automates your DevOps release pipelines, managing dependencies and ensuring that all of the tools in your DevOps toolchain are coordinated for optimal results.

- ✓ **Comprehensive ARO solution, including automated application deployment, advanced release orchestration and DevOps Intelligence**
- ✓ **Helps teams do DevOps on any architecture, from mainframe, to hybrid clouds, to containers, to a mixture of many**
- ✓ **Orchestrates and optimizes release flow for all of the tools in your software delivery pipeline, e.g. Jenkins, Puppet, Chef, Ansible, Docker, Jira and hundreds of others**
- ✓ **Top-ranked DevOps technology, designed for enterprises who need to deliver software quickly, with high quality and full compliance: model-based, agentless, dual-mode, and easy to use**



**Designed to meet the unique needs of companies doing Continuous Delivery at enterprise scale, Digital.ai is the most powerful, end-to-end DevOps software you'll find anywhere.**

## About Digital.ai

Digital.ai enables enterprises to focus on outcomes instead of outputs, create greater business value faster, and deliver secure digital experiences their customers trust. The Digital.ai Value Stream Platform seamlessly integrates all the disparate tools and processes across the various value streams, uses data and AI/ML to create connective tissue between them, and provides the real-time, contextual insights required to drive and sustain successful digital transformation. With Digital.ai, enterprises have the visibility they've been seeking to deliver value, drive growth, increase profitability, reduce security risk, and improve customer experience. **Learn more at digital.ai**