

The State of DevOps Tools

A Primer

DevOps tools are continuously evolving.
Here are the essentials you need to know now.

by Don McVittie
and Alan Shimel



DevOps practices are rapidly changing how enterprises and software producers bring their applications and digital services to market. Those watching the DevOps market have seen it grow, with new processes and toolsets to support software delivery and infrastructure management. The market is changing so quickly that anyone who takes their eyes off it for a moment finds the tools available, and their capabilities, have changed.

That's the first reason why the DevOps tools market is so fascinating to watch. The second reason is its rapid growth. According to a recent report from by KBV Research, the global DevOps market will hit \$8.8 billion by 2023, growing at a compound annual rate of 18 percent. That far outpaces the growth of the broader IT market. And, according to Forrester Research, 50 percent of organizations have implemented DevOps, reaching what Forrester calls "Escape Velocity."

This report, designed to be a primer on the current state of the DevOps tools market, isn't meant to be a definitive guide to every DevOps tool available. We hope to set the DevOps toolset baseline and clear confusion by providing an overview of the tool categories that currently are widely used in DevOps organizations, as well as where they fit in the DevOps environment. The report also provides an overview of tools that are either coming of age or worthy of consideration, even if not as evenly deployed — yet.

Contents

Source Code Management.....4	
GitHub.....6	
BitBucket.....7	
Perforce Helix.....8	
Artifact Repositories10	
Sonatype Nexus Repository12	
JFrog Artifactory.....13	
Continuous Integration	
Servers.....14	
Jenkins.....16	
CircleCI.....18	
GitLab	18
Travis CI.....19	
Continuous Delivery20	
IBM UrbanCode.....22	
Atlassian Bamboo.....23	
CloudBees Jenkins Solutions..24	
CodeShip.....26	
Application Release	
Automation28	
OpenMake Software.....30	
XebiaLabs.....33	
Puppet.....34	
Electric Cloud.....35	
CA Automic	36
Infrastructure Automation....38	
Chef	40
AWS Opsworks.....41	
SaltStack.....42	
Terraform.....43	
Puppet.....44	
Ansible.....45	
Continuous Delivery	
Management46	
Plutora Platform.....48	
Log Management50	
Splunk.....52	
SumoLogic.....53	
Scalyr.....53	
Monitoring54	
ExtraHop	56
Datadog	56
SignalFX.....57	
Alerting58	
VictorOps.....60	
PagerDuty.....61	
OpsGenie.....61	
ChatOps62	
Slack.....64	
HipChat	64
Flock.....65	
Database Configuration	
Automation Tools66	
DBMaestro.....68	
Datical	69
RedGate.....69	
Container Management.....70	
Kubernetes.....72	
Mesos and DC/OS.....73	
Docker Enterprise.....74	
Serverless76	
Conclusion.....79	

Source Code Management

A logical place to start a review of DevOps tools is with code repositories, as they are storing the code that makes up all of the applications deployed. Increasingly, they also store the configurations those applications require. Most organizations and developers use some sort of repository to store source code. At a very minimum, code repositories are a central place where teams can share and work on a common code base. The repositories also provide version control. What's more, today's repositories also have APIs and integration, triggering the building and testing of the project simply by checking in the code.

In an era of “low code” development and component and modular-based application development, the majority of application development today contains a smaller amount of new “original code.” Most code in applications come from pre-existing libraries, modules, components and artifacts. For this reason, some of the leading code repositories are actually storing open source components, artifacts and libraries.



GitHub

Git is a no-cost source code management version control system designed for projects both large and small. Git can be used on-premises or as a hosted service, one of the strengths of the system. Many organizations choose to do both, storing some projects on-premises and others within the online service.

As a commercial remote Git server offering, GitHub provides both free and premium services. And with a community of more than 26 million developers and 72 million repositories worldwide, GitHub is a de facto standard. Many other code repositories support Git integration and formats.

A significant advantage to using GitHub is its relatively large user base and near-ubiquity. As a result, integrations are plentiful.

Bitbucket

BitBucket

BitBucket is Atlassian's hosted, collaborative version control service for projects that use Git or Mercurial. As part of the Atlassian stack, BitBucket offers integration with Atlassian offerings such as Jira and Trello. The Atlassian marketplace also features a variety of other integrations. Although they may find BitBucket doesn't have quite as many integrations as GitHub, developers still likely would find the integration they need.

BitBucket's strengths include superior scalability and team collaboration. Recent additions to BitBucket expand on basic repository capabilities and edge toward continuous integration and other functional areas. One example of increased functionality is BitBucket Pipelines integration to support continuous integration and continuous delivery.

Like GitHub, Atlassian provides tiered free and premium services. The important difference between BitBucket and GitHub is in the free usage tier. BitBucket repositories are free for teams up to five. GitHub, in contrast, does not charge for public projects and has no limits on users if the repository is public and/or open source, but charges for private projects regardless of the number of users.



Helix**Core**



Helix**4Git**

Perforce Helix

If you can't beat them, join them. Perforce has been in the repository business for more than 15 years. After years of competing against Git, Perforce made the strategic decision to support Git with the goal of providing enterprise teams additional value.

Following a number of acquisitions, Perforce has expanded the capabilities of its Helix product into the software development lifecycle market. For those already using Git, the Helix4Git is available with Helix TeamHub enterprise.



Software at the speed of ideas.

Transform the software that fuels your business with CloudBees Jenkins Solutions.

Your business has amazing, world-changing ideas - the only thing standing in your way is the time, energy and processes it takes to turn code into finished product, to transform ideas into impact. CloudBees® - the only secure, scalable and supported Jenkins®-based DevOps platform - lets you focus on the ideas you want to bring to life, not the challenge of building, testing and deploying them, so you can make an impact sooner.

Get enterprise DevOps automation from CloudBees and allow your team to focus on building great software!



DevOps at scale:

Your business is critical. CloudBees takes the frustration of having to wait for build systems, making them ready and available.



Build and Deploy in Minutes:

Jumpstart your DevOps efforts by sharing templates across teams and deploy effortlessly.



End to End Visibility:

Get better at predicting what features will ship by the drop date. Capture metrics across the pipeline to identify bottlenecks.



Enterprise-grade security:

CloudBees provides a vetted version of Jenkins so you can lock down sensitive projects and ensure compliance.

To learn more about CloudBees Jenkins Solutions, visit: www.cloudbees.com.





Artifact **Repositories**

Open source software, like all software, changes with each version. Most of these changes evolve the software and enhance its capabilities, performance or security. Sometimes, however, changes can introduce errors and security defects. This is where artifact repositories come in. Artifact repositories provide organizations an authoritative source for current and approved versions of a module, library or other artifact. Artifact repositories also enable organizations to revert back to the last known trusted and approved version when undesirable software is entered into a system.

Artifact repositories support both internal and externally developed artifacts so teams have consistent access to all necessary modules, libraries and other artifacts.



Sonatype Nexus Repository

Sonatype Nexus Repository specializes in open source — especially Java-based and Maven-built — artifact and component storage and management. When it comes to security, the Sonatype Nexus Repository has a strong heritage and helps ensure an organization is using the latest and vulnerability-free versions.



JFrog Artifactory

Artifactory is a widely used software artifact repository. Much of its popularity is due to its ability to integrate with many toolsets in the DevOps and continuous integration and delivery stacks. Artifactory is integrated with a wide selection of DevOps tools, making them applicable to a variety of toolchains.

JFrog has its repository roots in an RDBMS format, but has grown beyond those roots. Sonatype has its roots in the Maven repository format but has also grown beyond those roots. Today, both support an array of repository formats, and users increasingly want specialized repositories that meet the needs of their other tools, so Docker repository support, or Maven repository support, based upon the rest of the architecture. Organizations should pay special attention during any evaluation to see which better fits the organizations' specific environment.

The Essentials:

A person wearing a dark grey suit jacket and a light blue striped shirt is shown from the chest up. They are holding a large, glowing cube composed of many smaller, metallic-looking cubes. The cube is illuminated from within, with a bright yellow-orange glow at its core, fading to white on the edges. The person's hand is visible, supporting the bottom right corner of the cube.

Continuous Integration Servers

Continuous integration (CI) servers are the point of entry for many new DevOps organizations. As an extension of agile, CI servers enable automated build and test, along with varying levels of notifications fundamental to keeping agile efforts on track. The open source project Jenkins® tracks active installations, for example, and (at press time) boasts 153,000 active Jenkins installs. While Jenkins is by far the most dominant CI/CD server in terms of market share, the number of active installations and users (estimated at more than 1.5 million) are indicative of a healthy market.

Additionally, many CI tools have added continuous delivery (CD) capabilities. The differences between continuous integration, continuous delivery and application release automation (ARA) continue to blur.



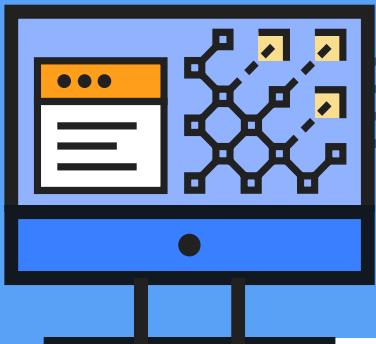
Jenkins

Jenkins

If there is one tool that the vast majority of DevOps teams have deployed, it is Jenkins. There are other tools, but Jenkins is the “Got DevOps?” standard. If Jenkins has a problem, it is one of abundance. The Jenkins plugin category is an embarrassment of riches: As many as 1,500 different plugins are available for Jenkins.

Certainly, Jenkins has taken steps to make plugin selection easier for its users, but the first thing every new user asks is, “Which plugins do I need?” Short answer: As few as possible. Especially at first. Start with a source code management (SCM) repository plugin (GitHub, BitBucket or Local Git, for example) and another plugin for builds (Gradle or Maven, for example). More can be added later, but starting light keeps the build/test environment simple and manageable, and doesn’t load down Jenkins with unnecessary plugins.

Jenkins 2.0 also includes Jenkins Pipeline and a friendlier interface (Blue Ocean). Managed Jenkins and other enterprise features are available from CloudBees Jenkins Enterprise.



Visually create CD pipelines in minutes & automatically deploy apps — all with one tool.

With Puppet Pipelines, you can standardize your software delivery pipelines from development to deployment — across servers, VMs and containers, on any cloud and on-premises.

[Meet Puppet Pipelines >](#)

[Request a demo >](#)



Learn more at puppet.com



CircleCI

CircleCI could be a good choice for those looking for a quick, inexpensive CI tool to get started with. Its emphasis on easy setup and ease of use makes it a simple choice for those starting out. Like other offerings, it is available as both a cloud instance and a local installation. It doesn't have the breadth of integrations or plugins as others, but if simplicity, lean and fast uptake are important, CircleCI is worth considering.



GitLab

GitLab

The open source version of GitLab has more than 80,000 active instances, while the GitLab Enterprise Edition has 500-plus paying enterprise customers. GitLab is well-documented, easy-to-use and extensible. Users have complained it suffers from poor alerting and notifications, but for those enterprise teams seeking an affordable option to Jenkins, GitLab is worth a look.



Travis CI

Travis CI

Travis CI is representative of hosted CI services. Because it is hosted and has fewer configuration steps, it is generally simpler to set up than Jenkins. However, the edge gained from simplicity is lost as projects grow and configuration files become more complex. Open Source projects often use Travis CI because it is free for such projects, and users don't have to maintain the hardware and software that Jenkins requires.

Choosing between a hosted or on-premises system often varies from project to project and even team to team. A team that is small and pressed for time might appreciate Travis CI because it replaces one server from the build chain with a service, and those just starting out might appreciate the simple setup while they decide what best suits the team's longer-term goals.



Continuous Delivery

It has been said that “if DevOps is the symptom, continuous delivery is the cure.” Automation and agile gave rise to continuous integration; the next logical step is continuous delivery. Many of the CI tools previously mentioned are also in continuous delivery. Additionally, many CD vendors are also in the application release automation market.

Where does continuous delivery fit in the CI/CD continuum? Continuous delivery is the extension of agile to build and test processes, continuous delivery is delivering software more often through use of DevOps methods. The composite of both is the core of application lifecycle management (ALM) in DevOps.



IBM UrbanCode

UrbanCode was an early DevOps favorite. After IBM acquired UrbanCode Inc, some of its small-company and open source aura faded. To IBM's credit, the company continued providing new UrbanCode functionality. Today, UrbanCode Deploy and UrbanCode Release together offer widely used CD and application release automation toolsets that integrate with Jenkins for continuous integration.



Atlassian Bamboo

Another tool within the Atlassian stack is Bamboo. Bamboo is deeply integrated with Atlassian products; it doesn't have as many integrations outside of the Atlassian stack as competitors. Outside of its own product stack, Atlassian has focused on integrations that are more popular, so many environments will find what they need. Organizations using or considering BitBucket, Jira and HipChat should seriously consider Bamboo.



CloudBees Jenkins Solutions

Jenkins, an open source project, is the most widely adopted continuous integration solution in the market, enabling developers to automate the integration and validation of code changes. Its automation capabilities and number of integrations also make it a popular solution for orchestrating continuous delivery (CD). However, as adoption of CD in an organization grows so do the number of Jenkins instances, and it becomes more difficult to manage, secure and support Jenkins for business critical applications.

This is where CloudBees® comes into the picture, their Jenkins solutions help enterprises implement and scale CD faster. Jenkins and CD pipelines can be centrally managed to ensure teams can adopt CD quickly and flexibly while still adhering to company process and security standards. CloudBees Jenkins Solutions extend the Jenkins user interface to enable onboarding of teams and projects with just a few clicks.

Jenkins has +1500 plugins enabling integration to virtually every DevOps tool, however some users find the sheer number of integrations to present a compatibility challenge. CloudBees solves this problem by rigorously testing plugins for quality and interoperability, making adopting new functionality and upgrading much easier.

As continuous delivery becomes increasingly business critical, it is important that the CD tools are resilient and well supported. In fact, many organizations have begun to apply production service level agreements to the CD tools. CloudBees Jenkins Solutions support this by providing 24/7 for Jenkins and CD, so you can reliably deliver better software faster.

Recently, CloudBees extended its solution offerings with CloudBees DevOptics®, providing a holistic view of the entire DevOps process through Value Streams.



CODESHIP

CodeShip

Codeship® by CloudBees helps engineering teams to test and deliver their software in the cloud, fast. No matter the architecture or application type, Codeship has you covered.

Codeship's products offer a turnkey, ready to go CI/CD environment that adapts to your teams' needs and helps you set up Continuous Delivery for your cloud applications. From simple, standardized workflows to very complex applications and requirements.

Additionally, Codeship offers excellent support for the container ecosystem helping you test, build and deploy your Cloud Native/containerized stack by being able to leverage the environment configuration you already have in place.

For software delivery, these are the most common questions you hear:

PLUTORA.

When will the release go live?

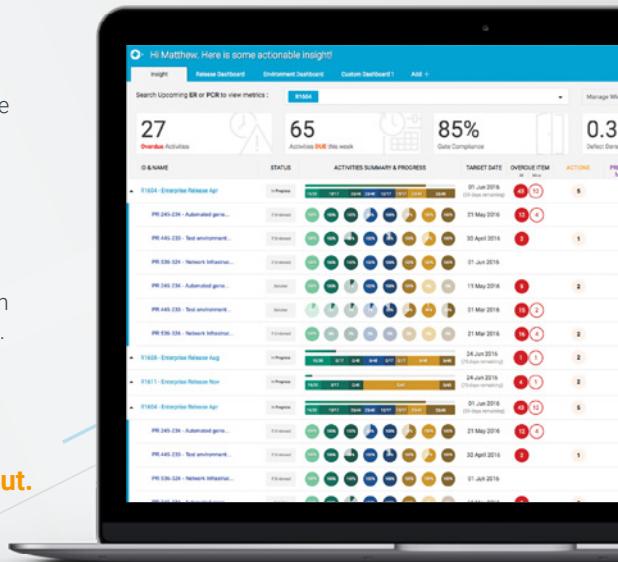
How much risk is involved with this release?

What is the quality of this release?

These simple questions are hard to answer without looking beyond simple feature burndown charts, test results, and release calendars. Your job depends upon knowing the answers.

Plutora gives you complete visibility into the entire software delivery process, improving team collaboration and efficiency not previously possible. You can ensure predictability while continuously improving the process.

**Your business is software.
Become a Software Juggernaut.**



Plutora integrates with your current toolchains from Continuous Delivery to Waterfall

service^{now}

CHEF

IBM

ca
technologies

git

Microsoft

puppet

Jira Software

Jenkins

Learn more at
plutora.com →



Application Release Automation

Application release automation (ARA) combines the processes of continuous integration and deployment with deployment automation and release coordination. Ideally, the toolsets will be part of a chain that runs throughout application build, test and deployment. Once ready, applications and application updates can hit production by simply pushing a button. Gartner estimated the ARA market to be at \$228.2 million USD in 2016, with a compound annual growth rate of 20 percent through 2020. The research firm expects 50 percent of enterprises to use application release automation by 2020, up from 15 percent today.



OpenMake Software

OpenMake offers DeployHub Pro, a continuous delivery solution based upon the open source DeployHub product. With agentless deploys, interfaces to the most common DevOps tools and multi-platform support, it has much to offer the modern enterprise. OpenMake also offers Meister, an automated build tool that integrates with an array of development tools, including a massive selection of compilers. Meister claims to improve compile times by 95 percent, which is impressive in environments that have large code bases.

The strength of DeployHub is definitely its open source roots and agentless operations. The ability to grab the source and see what is going on, combined with the ability to deploy applications to a wealth of targets (including Cisco routers), make it a promising choice for complex environments that lean toward open source solutions. Support via the open source project is actually pretty solid, with some of the best OSS documentation we've seen combined with an active community responding to requests.

The strengths of Meister are just-in-time build dependencies that shorten the amount of time spent rebuilding objects and parallel processing for builds. A good secondary strength is server pooling for Jenkins builds.

The ability to have a single central build tool for hundreds of different compilers on a variety of targets is useful to an enterprise that has grown organically. It is rare that the current standard dev language is the only language in use, so broad compiler support is key.

Between the two, build and delivery of applications are covered with a combination of open source and commercial applications. OpenMake is worth evaluating for organizations with a complex programming language environment and a desire for an agentless deploy system.

XebiaLabs

XebiaLabs

XebiaLabs provides two ARA tools: XL Release, for managing development, and XL Deploy, for managing operations. Building the application and installing to the environment are effectively managed by both products. Like others here, XebiaLabs bridges the gap between build and deployment tools. Users can map what resources are required for a deployment and target that specific deployment to a predefined infrastructure. The ARA processes are repeatable and reduce manual intervention and associated errors.



DeployHub®

01

02

03

04

Versioned deployments from
database to infrastructure

The future of
Software Releases

Open Source | Agentless | Built-in Versioning
Continuous Deployments | Packaging | Kubernetes Support
Jenkins, Ansible, GitHub, Jira Ready

Try a Sandbox
DeployHub.org



Puppet Pipelines for Applications

Puppet Pipelines for Applications simplifies application delivery, providing continuous delivery and release automation from every commit to every deployment. Puppet's ARA tool automatically builds and automates deployments with full control to ensure that your development and release teams are consistently shipping software and delivering on time.

You can also integrate with automatic notifications to Slack, HipChat, and email and allow role-based access control for more governance. Perhaps the greatest feature of Pipelines for Applications is that it works with any cloud or on-premise application and you can visualize your software delivery pipelines all in a single place.



Electric Cloud

Electric Cloud provides ARA tools that are comparable to the other tools listed here, with one feature some will find useful. Electric Cloud's Electric Flow offers the ability to customize the user interface for business users, making it easy for those who are not so technical to monitor their projects.

Business users who want regular status updates may want to consider the Electric Flow dashboard, which ties progress to their internally defined business goals (as opposed to agile task completion or story completion). There are many other features to consider as well.

While many ARA tools look quite similar (and there are a lot of tools to consider), identifying the differentiators among each is worth the time.



Automic™

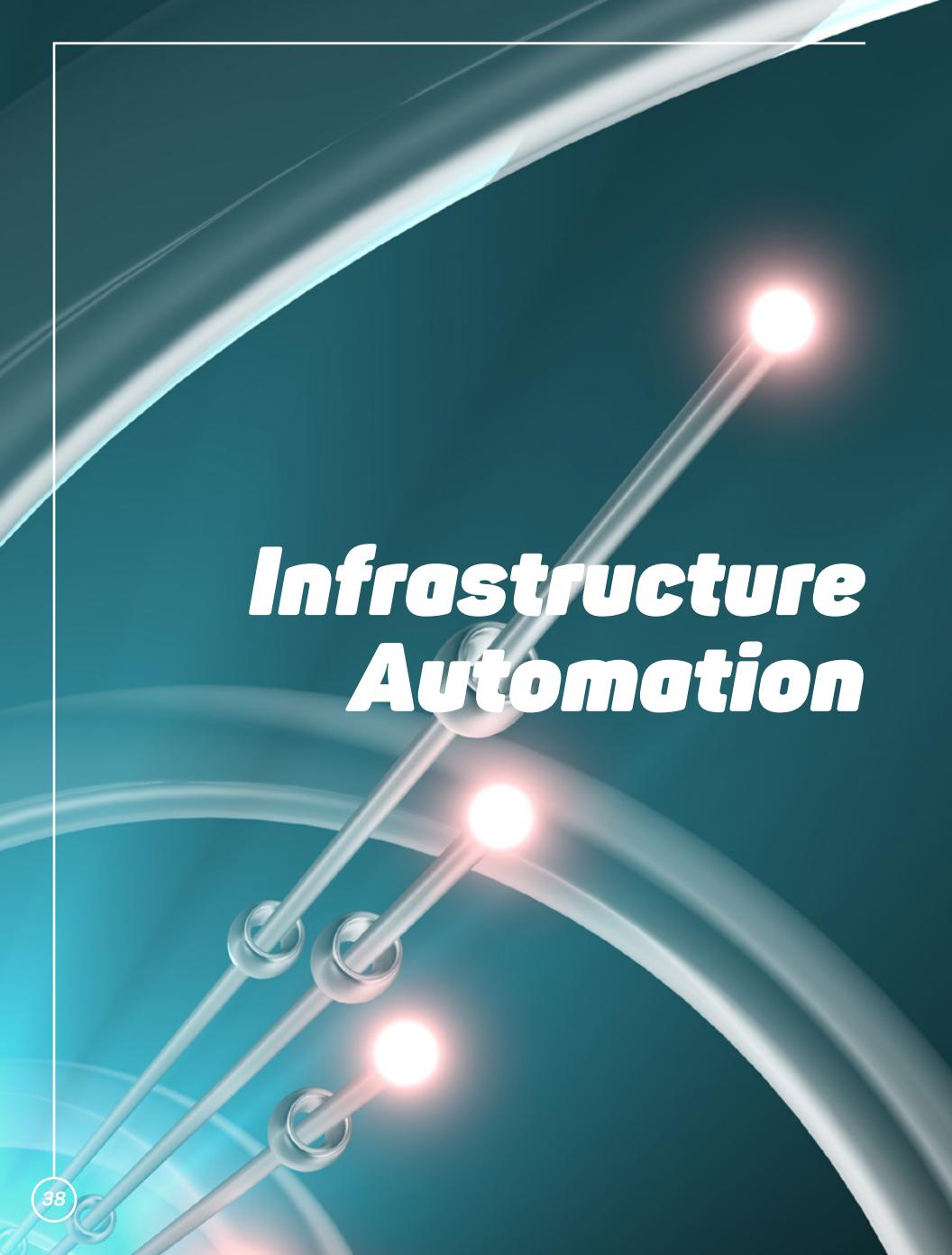
CA Automic

Like all ARA tools, Automic sits between your CI/CD processes and your deployment stack, melding them into a complete DevOps toolchain. And like every other ARA tool, Automic supports the standards that are the current cost-of-entry into the ARA market. It also supports a growing array of other tools, which is one of its strengths. Automic had good and growing support for extant tools before being purchased by CA, and working with CA has increased its access to CA tools. Automic puts out what it calls “Action Packs” that support new tools and platforms at a pretty impressive rate.

As of this writing, Automic supports deployment to every environment from the mainframe through serverless, integrating with tools that are common in the enterprise. Considering the work required to get any ARA tool integrated into the build/test/deploy process, support for the DevOps tools used in a given environment is critical as it means fewer custom interfaces have to be written.

PM and line-of-business sponsors like the rollup reporting environment that Automic offers, while operations staff likes the breadth of tool support. About the only common complaint involving the product is its lack of support for a variety of scripting languages for customization – a complaint hardly unique to Automic in the DevOps world.

Available plug-ins for Automic are included in the Automic Marketplace. Automic is well on its way to providing out-of-the-box support for the majority of Dev/DevOps toolsets, and those it doesn't support can be integrated via its scripting language and/or APIs.

The background features a complex network of translucent, glowing blue and white lines forming a grid-like structure against a dark teal gradient background. Several glowing red spheres, representing nodes, are connected by these lines. One node is located at the top right, another near the center, one below it, and two more along a diagonal line towards the bottom left.

Infrastructure Automation

Infrastructure automation is the process that automates, largely through scripting, the installation, configuration and management of servers. It's often called infrastructure-as-code because infrastructure automation is the ability to define infrastructure and workload configurations as sets of scripts that can be applied to either a single server instance or thousands of servers. DevOps has made infrastructure-as-code tools popular and has helped to mature these implementations. For example, launching an application with a web/API server and a database server with all of the networking and supporting applications could be scripted and run as needed.



Chef

Chef is the elder statesman of infrastructure-as-a-service (IaaS) automation. Using Chef, machines can be defined in scripts, which can be used to ensure the configuration of those machines are exactly the same every time. This is useful not just for systems operations but also for deployment, because Chef makes deployment failures easier to debug. If the Chef script works one day but doesn't work the next day, a quick look at what changed in the script usually will resolve the issue.



AWS Opsworks

Large cloud vendors follow an API-first development methodology. This means that APIs are available for tasks ranging from “launch a server” to “create a route.” The existence of APIs for all infrastructure in a cloud makes the cloud particularly well-suited for DevOps infrastructure management.

AWS Opsworks originally was a private-labeled Chef implementation for Amazon Web Services. It was built to use Chef and APIs to get AWS-tested or -hosted cloud environments online and running quickly. While most infrastructure automation tools support the major cloud providers, Opsworks is optimized specifically to deploy and manipulate AWS objects. Out of the box, Opsworks includes Chef functions for AWS-specific deployment needs, relieving teams from having to cobble something together.

Recently Puppet announced enhanced functionality with Opsworks. It remains to be seen whether Opsworks will evolve into a standalone tool independent of its Chef roots.



SaltStack

Similar in design to Chef, SaltStack Enterprise is more focused on compliance and security for the enterprise. As with most infrastructure orchestration and automation tools, SaltStack supports application installation and automation and can manage the entire deployment stack regardless of the target infrastructure. If managing the entire deployment stack is necessary, but security and compliance are priorities, SaltStack can answer both needs more readily than competitors that have focused more on infrastructure and application deployment than on security and compliance.



Terraform

Terraform is designed to automate the entirety of the data center. While it has the same aim as SaltStack Enterprise, it approaches the task differently. Terraform is designed to manage the tools that would normally automate the data center. To automate a task, Terraform typically issues a request to the tool that specializes in that task. For example, Chef may be called to manage servers, while Puppet might be called to install and configure the applications on those servers. It takes a lot to orchestrate every bit of infrastructure, and Terraform is a complex system. But for those that have the time and intend to have a highly portable, highly flexible data center, this is a tool worth considering.



Puppet

If there was ever a ubiquitous tool for DevOps software deployment, it is Puppet. Even if an organization isn't "doing DevOps," it probably is using Puppet somewhere. Puppet is very good at getting applications deployed to pre-built infrastructure. For those just getting started with deployment automation, Puppet is a good place to start. But it is not just for beginners: According to the company, more than 3 in 4 Fortune 100 companies use Puppet for application deployment.

Puppet automates the installation and configuration of applications on target servers. By writing scripts that define and install application prerequisites, setting the server variables the application requires and writing configuration scripts for applications and daemons, a team can then simply rerun the script to reinstall. A large user community has already scripted many application installations. For a large number of software applications — particularly in the open source realm — users merely need to download the relevant puppet script files and change what is written to the configuration files to reflect their project.



Ansible

Ansible is different from Puppet in that it is agentless. Ansible can be set up and running without installing software on each machine. This speeds time-to-value for Ansible users. Also, Ansible uses the human-readable data-serialization language YAML to define installations. Different users find this to be either a plus or a minus when compared to Puppet Script, and since writing scripts is a big part of both tools' usage, it is worth determining the organization's preference.

Also since its acquisition by Red Hat, Ansible now offers tighter integrations to the Red Hat Enterprise Linux (RHEL) stack that could make it a favorable selection.

For organizations that have yet to adopt deployment automation, it is worth the effort to look at all the deployment automation tools and determine which will work best for their enterprise. Many of these automation tools are also capable of providing either a complete CI, CD and ARA pipeline or integrations with popular tools in that area.



Continuous Delivery Management

Enterprise IT environments are amazingly complex. Even before DevOps came along, most enterprises had several development languages, platforms, reporting tools, test environments and more. DevOps introduced a new set of tools – some that span this complexity, some that increase it. For organizations with dozens of application development teams, there are likely just as many architectures for build/test/deploy. And often they are interdependent, as applications require integration with other projects or platforms.



Plutora Platform

Plutora is one of a few organizations that sees the problem of continuous delivery management and attempts to address it. It offers a software-as-service (SaaS) based continuous delivery management platform, with four different tools that all aim to help an organization with one goal: to make the complex world of enterprise development more manageable.

Plutora Release brings operational visibility into complex, multisystem releases. Plutora release works to open up visibility of the delivery pipeline from inception to deployment. Beyond defining the release with its associated features, environments and calendar, it allows for development phases (eg. sprints) and associated gating criteria in order to move to the next phase. For example, gates can mix both manual and automated criteria to ensure quality goals, documentation/regulatory requirements, and code coverage requirements. The idea is to offer a unified platform for managing software delivery across disparate systems and teams.

Plutora Environments aims to make provisioning of test environments easier. Allocation of test environments for complex systems can be the bottleneck that holds up the entire build stack. Plutora Environments aims to ease this by offering a booking system that coordinates management and configuration of pre-production test environments.

Plutora Deploy attempts to address deployments that can span not just systems and platforms, but geographies also. Plutora Deploy helps users create a “go live” plan with both automated and manual steps to ensure the entire system rolls out as expected, when expected.

Plutora Test is a complete test management solution that doubles as a centralized clearinghouse of test metrics and information. It readily synchronizes data with Jira and a suite of automation tools gathering the results across the environments. Real-time traceability matrices enable test teams to influence testing activities from unit test created in development all the way to exploratory testing done on staging and production.

Taken as a whole, the Plutora Platform is a department-spanning management toolset. However, it’s likely a given organization will start with a single tool before considering the others. For large enterprises, particularly ones that span geographies, Plutora Platform useful to bring control and visibility to all Test/Release/Deploy functions. The benefit to DevOps is integration with DevOps tools and reducing the extra complexity those DevOps tools introduce.

Note that Plutora is 100 percent SaaS. This means accessibility from multiple locations to the same data, but for some organizations might be a cause for concern with the information these tools will require. Suitability to environment is always a key thing to check, and Plutora is best aimed at complex environments that are comfortable with SaaS solutions.

Log Management

DevOps teams rely on log management to better run their environments and organizations. Without a log management tool, it can be difficult — if not almost impossible — to manage and analyze the volume of logs in most environments. Discovering and identifying trouble spots through either automation or search is essential. There are many tools available that help with log management, monitoring and alerting. According to Markets and Markets, the APM space is expected to grow 12.86 percent, compounded annually, to reach \$4.98 billion in 2019.



Splunk

When it comes to log management and analysis, Splunk is the gold standard. Splunk can aggregate tremendous volumes of logs and offers advanced log search and analysis capabilities. Splunk's log searching and automated analysis assists organizations in reducing the amount of time in finding what went wrong, enabling them to remedy problems more swiftly. Splunk historically has been on-premises, but the company also has a hosted option called Splunk Cloud.

One of DevOps' objectives is to shorten IT task turnaround times. Searching throughout an enterprise for logs to correlate events and determine root cause can be, in a word, painful. It's also time-consuming. Splunk and tools like it aim to make such tasks faster and easier.



SumoLogic

SumoLogic is a cloud-hosted log manager. Similar to Splunk in concept, SumoLogic runs on someone else's servers, which may be a plus or a minus depending on the organization. SumoLogic is a cloud-hosted service, but it also provides for local usage.



Scalyr

Scalyr also looks and feels like Splunk. Its pricing is generally more predictable and less costly than Splunk's pricing, and Scalyr uses that as a reason to consider its offering. Scalyr also points out its performance and unlimited log growth as additional reasons to consider its technology.



Monitoring

Monitoring from a DevOps perspective encompasses not only application performance but also can include user interaction, infrastructure capacity and delivery. Leading tools in this category have stretched the envelope in UX design to give DevOps teams visibility into a constantly scaling environment. The best of the best in this category can give a birds-eye view of expansive deployments, while allowing for quick drill-down to the gritty details.



ExtraHop

ExtraHop provides visibility into communications across an infrastructure and helps determine bottlenecks. ExtraHop can reconstruct failed data flows to find the source of a problem and creates a picture of the infrastructure taken from a perspective many enterprises don't have the ability to see: data flows across the network. For quality assurance and test teams, ExtraHop captures what is actually happening in the environment to augment the test results. ExtraHop is particularly suitable for complex or high-performance environments.



DATADOG

Datadog

Datadog monitors all aspects of the software infrastructure so the DevOps team sees what's happening with specific components. With roll-up results and built-in application performance management, Datadog helps to determine what is happening within a complex environment. While ExtraHop analyzes networking streams to offer performance and problem-resolution information, Datadog offers similar information but from the perspective of what is happening inside of the software.



SignalFX

SignalFX is a tool similar in many ways to Datadog, with an agent that collects local data and reports that data to SignalFX for processing and presentation in a dashboard. Where SignalFX is different is in custom data – the data a given application or team deems important but isn't normally collected. That data is sent directly to SignalFX, whereas in Datadog it is sent to the agent, and bundled along with the other collected data. For environments with intermittent connection issues, Datadog might be a better fit. For environments where a lot of different custom data is collected, SignalFX may be a better choice, simply because the tool reports data directly to the hosting servers with no middleman. This makes it easier to track lost data and identify incorrect custom data.



Alerting

No matter the monitoring tools and processes, a tree falling in a forest that no one hears fall doesn't help DevOps teams deliver. The DevOps market has seen several alerting platforms evolve that free the admin from the NAC, but allow for teamwide 24/7 communication and response. The hallmark of these tools is a wide integration footprint with as many monitoring tools as possible, as well as the widest possible means to alert, communicate and remediate incidents.



VictorOps

VictorOps hopes to change the way DevOps team members interact with systems and users. The current iteration aims to help improve incident visibility and cross-team communications by providing clear, understandable information to those working outside their discipline, such as a developer troubleshooting network issues.

VictorOps enables both broad and targeted communications of status and issues. Providing more information to team members who otherwise might not be aware of event details can aid problem resolution significantly.



PagerDuty

PagerDuty grew from an automated paging system to providing full incident response, including artificial intelligence-aided automated response. It can take considerable effort to correctly configure support team members, call schedules, escalations and other related aspects of incident response, but once configuration is completed, PagerDuty helps streamline incident response.



OpsGenie

OpsGenie is a less costly alternative to PagerDuty and VictorOps. While their feature sets are similar, each has differences that could matter considerably to some environments. It's worth looking at all three before making a decision.



ChatOps

ChatOps originally described the use of APIs to embed chat tools into systems alerting. Vendors identified customer need and some started integrating chat capabilities into the DevOps toolchain to facilitate systems management and troubleshooting. Today, using ChatOps, it is possible to check in code changes and inform teams of the results, helping resolve problems quickly.



Slack

Slack is the ChatOps standard. Teams originally integrated Slack into their DevOps practices as a way to offer quick communications, and later for systems notifications. Today, Slack has more than 900 integrations, including many of the tools discussed in this report. Slack provides rapid notifications, the ability to save select conversations to cloud-based storage, automation of DevOps tools based on channel activity and more. Slack's API makes incremental integration creation a straightforward process.



HipChat

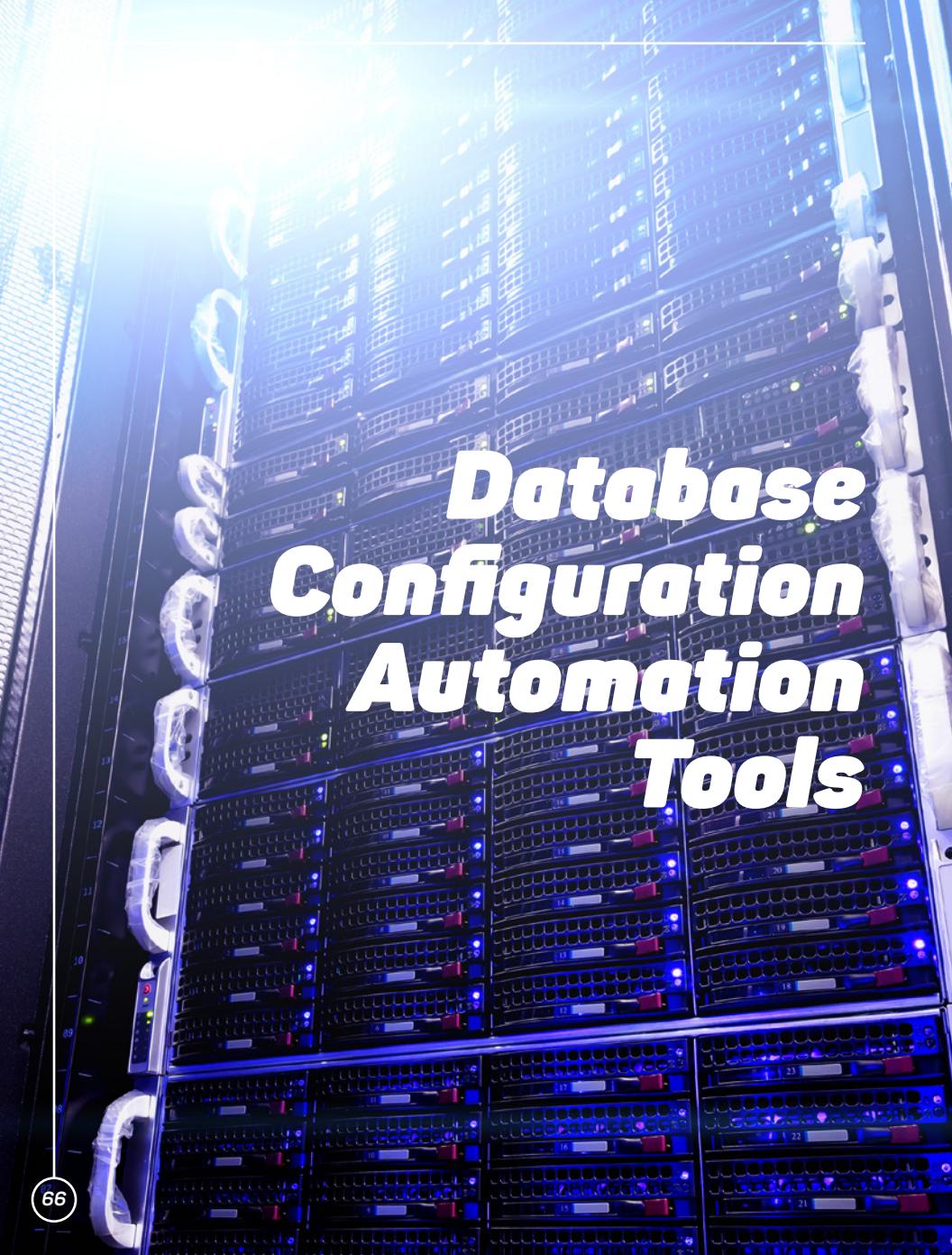
HipChat is the communications channel underlying the Atlassian platform. As such, organizations heavily standardized on Atlassian prefer HipChat over other ChatOps offerings. HipChat offers advanced capabilities within the Atlassian tool stack, and third-party integrations and an API for customization are available.



Flock

Flock is a less-expensive Slack alternative. However, it does not have the number of users or the depth of integrations. With an API, theoretically anything can be integrated with Flock; it just takes time.

The tools mentioned here are currently battling it out when it comes to features and cost. Research is warranted, as the market is rapidly evolving.



Database Configuration Automation Tools

DevOps organizations have been slow to add database support and management into their toolchains. But if code is changing, it is almost guaranteed that databases are, too. The vendors below help fold the database into the DevOps practice. After all, sooner or later, the database needs to be included in DevOps simply because most apps rely heavily on data. But databases are not terribly mobile, which makes targeting databases to new platforms more difficult.

Database configuration and automation tools are relevant to DevOps database management because agile applications still need help getting to existing data. And, in some use cases, the database actually is mobile, or a new empty instance can be launched on demand without loss of application functionality.



DBMaestro

DBMaestro's twin products, Database Source Control and Database Release Automation, combine to offer CI and CD functionality to the database. By creating a source-controlled single point of authority, and then a tool to build and release the database defined by that single point of authority, DBMaestro brings DevOps to the database.

There are features that emulate the merge functionality of source control management offerings, which helps avoid the loss of new columns or attributes when two competing changes are made.

DBMaestro is worth considering when moving the database to a more agile footing — or straight into DevOps.



Datical

Datical enables database as code, automating the database change and deploy process — the same way other codebases are managed. In environments where new instances of the database are launched or where datasets are mobile, defining the database as code standardizes data stores across instances and even architectures.



RedGate

RedGate provides a selection of tools that facilitate SQL Server deployment, test, monitoring and troubleshooting processes. Although it applies specifically to SQL Server, RedGate's toolset is comprehensive. Companies with Microsoft data architectures might want to consider RedGate.

A close-up photograph of several glass bottles. In the foreground, a single red glass bottle stands vertically, its vibrant color contrasting sharply with the surrounding clear bottles. Behind it, several clear glass bottles are arranged horizontally, their shapes slightly blurred by a shallow depth of field. The lighting highlights the smooth surfaces and curved forms of the glass.

Container Management

Containers have enjoyed rapid adoption across enterprises and industries. Containers are quick and easy to spin up, use fewer resources than a fully virtualized machine and, when combined with DevOps automation, can be scaled quickly to meet the needs of the application. Still, when a given application or service grows in complexity, more than just containers is needed. Infrastructure management, scalability and the ability to manage groups of containers uniformly are all necessary so the application can be created with dependencies and the container environment is useful. Container management tools address these needs.

Thus far, vendors have offered two approaches to container management systems: one that makes a selection of servers appear as a single server that can run many containers (clustering) and another that makes each server into a “pod” that will run related containers.



Kubernetes

Kubernetes is the most used container management tool in the enterprise. Kubernetes manages individual machines as entities that can be assigned multiple containers. Simply adding machines increases scale, and each machine manages internal tasks such as routing and scheduling.

With Kubernetes, a collection of containers can be run as services (servers, databases, etc.) and can be called upon when needed. Kubernetes manages routing between machines or groups of machines, and auto-scaling adds resiliency to the infrastructure by scaling instances as user-defined thresholds are met.



MESOS



DC/OS

Mesos and DC/OS

Mesos and its commercial descendant DC/OS take a different approach than Kubernetes to container management. Servers are joined to the Mesos cluster, and the cluster is presented as one large computer with jobs distributed across underlying systems. Recently, DC/OS has added Kubernetes support, which means that a Kubernetes instance (or three) could be running on a DC/OS cluster. This enables different versions of Kubernetes to run as needed, but introduces another layer of management that staff will have to maintain. The tradeoff associated with managing another layer of software might be worthwhile for organizations that need to run multiple, concurrent versions of Kubernetes.



Docker Enterprise (Swarm)

Docker Enterprise Edition is the container management tool offered by popular container vendor Docker. Originally a competitor to Kubernetes, Docker Enterprise has been positioned as more secure by default than Kubernetes. Eventually, as Kubernetes and its commercial variants took over the enterprise container management space, Docker began offering Kubernetes as an option for deploying containers, along with Docker's security and management.

Because Docker containers are the standard for system level images, Docker Enterprise will see some amount of success. The additional security protections offered are not well-matched by other competitors, but as we've seen across DevOps, that can change quickly.

Docker Enterprise Edition is worth a look.

The Modern Software Factory.

The blueprint for digital transformation.

It's where mobile-to-mainframe agility, automation, insights and security come together to help your business compete in the rapidly changing application economy.

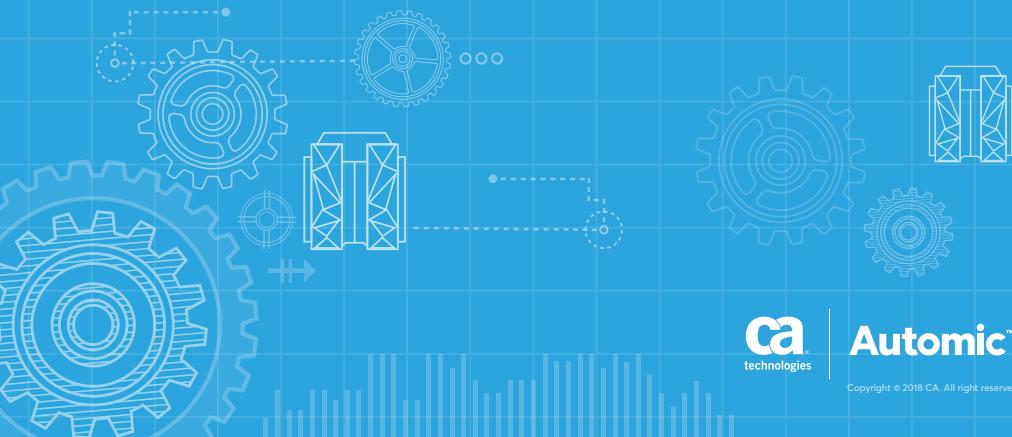
It's not a place you go. It's what you build with end-to-end tools and guidance from CA Technologies.

It's how you continuously deliver better apps, faster and at scale—to meet ever-growing consumer demand.



**Modern Software Factory:
The Answer - Video**

To learn more visit automic.com



Automic™

Copyright © 2018 CA. All right reserved.

Serverless

Serverless grew out of the microservices movement. In microservices, an application is broken into smaller pieces, each of which rely heavily upon internal APIs to communicate. Serverless takes this concept of independent pieces tied together by an API one step further, making each logical function into an API-interfaced independent entity. By breaking the code up in this manner, serverless then services requests to a given API on demand. If no requests are incoming, nothing is running. If many requests are incoming, many instances of that particular API may be servicing requests concurrently. The appealing element of serverless is also how it gets its name: The API is not tied to a given server. And while it will be spun up when it is run to respond to a given request, where it runs is not important to the completion of the API.

Serverless is still new and it is questionable whether it will find broad use in enterprise applications. Already, though, there are good use cases for serverless, including cloud-bursting for spikes in application usage. When bottleneck code is placed in a serverless environment, scaling of that code happens automatically. Since use cases exist that make sense for serverless, it will likely find a home in the agile and DevOps toolboxes.



APACHE
OpenWhisk™



Since serverless technology is still young, there are not yet prominent tools. Indeed, each serverless implementation is unique – which means there is no portability between serverless vendors. All of the toolsets currently provide functionality such as application firewalls, triggered invocations, REST APIs and the ability to connect to back-end services. Still, one useful – perhaps determining – differentiator is where serverless functions can run. At some organizations, some functions need to be in-house. That means the ability to deploy serverless internally or externally must be considered if certain applications can't go to a public cloud. Current vendors include Google CloudFunctions (currently in beta), OpenWhisk and AWS Lambda.

Conclusion

The maturation of DevOps guarantees new tools will continue to emerge. We've offered a cross-section of useful tools for enterprise DevOps teams. It is by no means an exhaustive study, but rather a good jumping-off point. Of course, an organization's mileage may vary depending on its own needs. While exact toolchains for given project teams or enterprises will vary, the tools from each of these categories are all essential to automating the application life cycle and DevOps success.



DevOps.com

Where the world meets DevOps



<http://www.devops.com>



<https://twitter.com/devopsdotcom>



<https://www.facebook.com/devopscom>