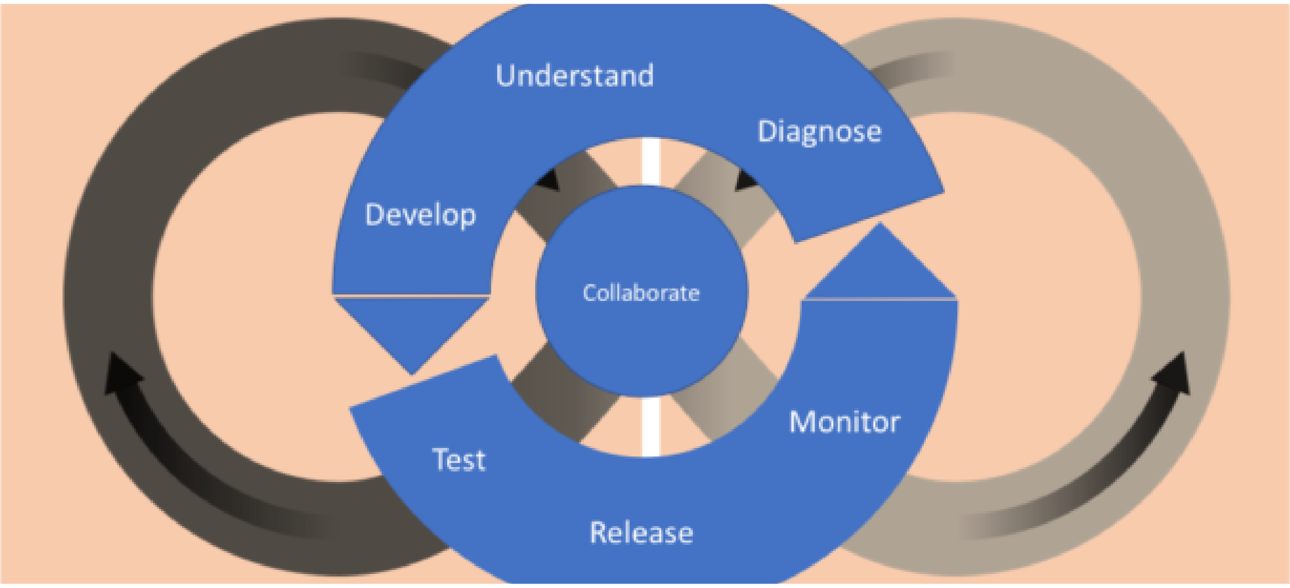
GIGAOM

Sector Roadmap: DevOps Automation Tools



By Jon Collins

Sector Roadmap: DevOps Automation Tools

11/16/2017

Table of Contents

1 Summary

Across all industries, enterprise organizations are looking to respond to rapidly-growing competitors who have appeared seemingly out of nowhere to pose a genuine threat. As they do so, attention inevitably falls on the DevOps practices used by such disruptive players in the market. Perhaps, goes the thinking, an enterprise can avoid being outmaneuvered and potentially achieve similar levels of growth if they change their own DevOps practices?

Such new approaches require rethinking process and lifecycle automation. While tools and platforms for software development and service management are not new, this sector roadmap looks specifically at the current landscape for DevOps automation tools, i.e. those designed or oriented to make DevOps workflows more efficient and effective.

This report covers major vendors, forward-looking solutions, and outsiders along with the primary use cases, to help decision makers and technology buyers set criteria as they sift through available options. It covers AWS, HPE, IBM, Microsoft, Puppet and Red Hat. Each offers a strong set of features, making them suitable for most usage scenarios described in this report. Key findings in our analysis include:

- The growth of DevOps practices and tools has been catalyzed by cloud platforms, open source, and Software as a Service (SaaS) models. These catalysts have reduced cost of entry and driven iterative development of DevOps tools.
- While DevOps automation tools can be used in other areas, DevOps practices could not exist without automation. Tools that automate the hand-off of software releases between development and operations are particularly important.
- Principles of DevOps can be applied without cloud as a target, but enterprises should be looking to cloud-based architectures if they are looking to achieve the same level of innovation and agility as start-ups.
- Given their disruptive nature, enterprise organizations looking to adopt DevOps practices and tools will require culture change and experience. IT decision makers will need to map tools against their own needs and maturity level.
- DevOps automation particularly suits situations in which requirements are fast-changing or poorly understood, proof of concept environments, and highly collaborative development scenarios (e.g. co-creation with customers).
- DevOps automation tools can be evaluated in terms of what they bring to the lifecycle: not only automation but also governance, collaboration, analytics, and integration features. Used

in the right way, automation tools go a long way to lowering barriers to entry into DevOps practices.

- Vendors place much emphasis on where "dev" meets "ops," but place less attention on where "ops" closes the loop with "dev." As tools evolve, it is hoped that the latter part of the cycle is treated.
- IBM, AWS and Microsoft offer DevOps automation as an on-ramp to their own infrastructure and platforms as a service. AWS only serves its own target environment, whereas other vendors support multiple target platforms, including OpenStack.
- DevOps practices will increase in both importance and maturity. Analytics and machine learning bring increased visibility and automation, while serverless infrastructure and more advanced orchestration models simplify deployment handoffs and increase platform autonomy.

Key:

- Number indicates company's relative strength across all vectors (note that higher is better)
- Size of ball indicates company's relative strength along individual vector

2 Introduction and Methodology

DevOps, the set of best practices around software development and operational service management, has moved from the domain of Web-scale startups into enterprise IT in recent years. While agile, just-in-time approaches have existed for decades in software development, enterprises now accept that the operational side of the house needs to become similarly dynamic. Simply put, how can you make weekly or even daily releases of software, when operations teams are geared up, staffed, and skilled around a minimum of six-monthly cycles?

DevOps implements governance frameworks and methodologies that respond to this question, supporting innovation and increasing speed of service delivery by:

1. Minimizing interfaces and overheads between development and operations teams and processes, avoiding expensive and unnecessary hand-offs.
2. Adopting agile concepts across both development and operations organizations, essentially re-organizing and re-skilling ops teams to expect, and cater for, frequent, repeated deliveries.
3. Allowing for engineers and decision makers to collaborate and focus on the work at hand, rather than duplicating or wasting unnecessary effort.

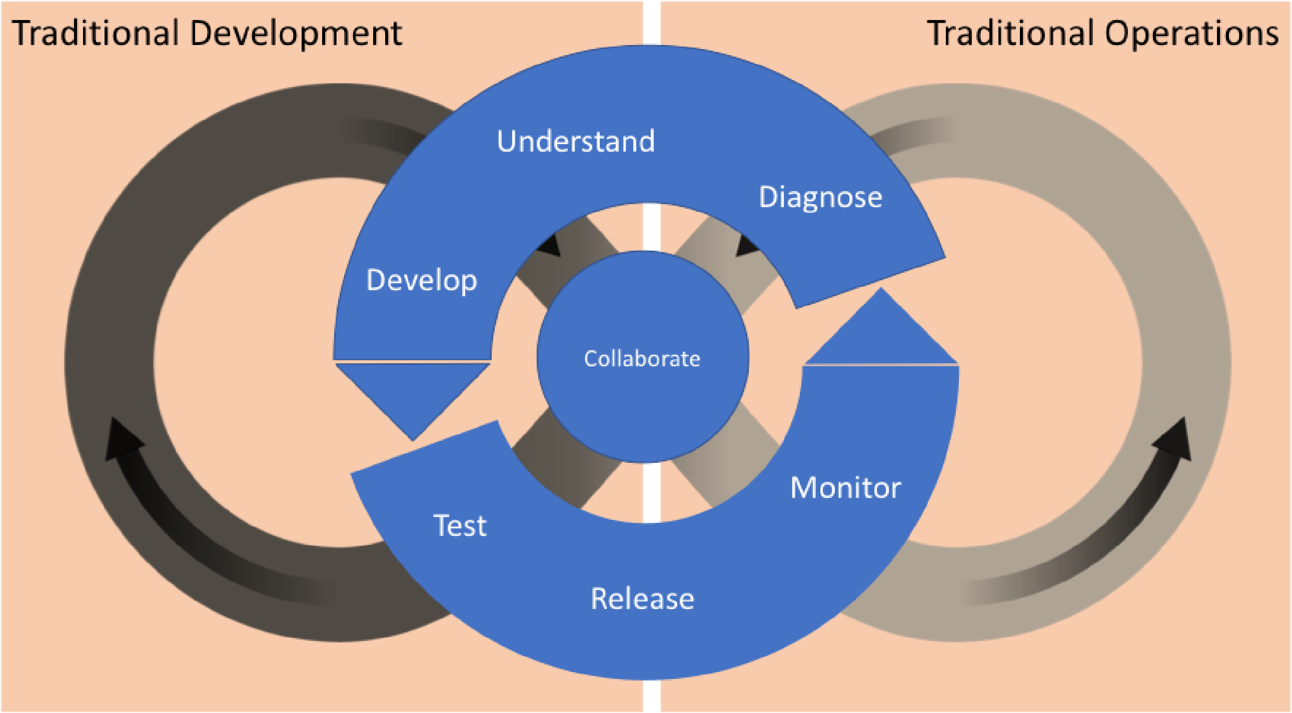


Figure 1. The eternal DevOps cycle

In Figure 1, we see how DevOps brings activities of development and operations into a single, repeating cycle involving the following activities:

- **Understand and plan** – requirements and problem management, user stories, change requests and issue reports, and aspects such as portfolio and financial management.
- **Develop** – day-to-day activities of creating, building, and integrating software packages.
- **Test and quality assurance –** test creation and execution, including security testing and post-deployment testing, e.g. blitz and failure testing, as well as management of test scripts and results.
- **Release and deploy** – movement of an application or package into the live environment, including provisioning, containerization, and configuration.
- **Monitor and manage** – IT service management and application performance management, event monitoring, fault identification, and diagnosis
- **Collaborate** – communication between stakeholders as they go about their own tasks, potentially in distributed teams. Also called "ChatOps" in DevOps circles.

While terminology may vary, success in DevOps boils down to making this closed cycle as smooth and efficient as possible. For an enterprise working with thousands of developers, even incremental improvements in efficiency equate to a significant sum.

Methodology

What makes a vendor offering "DevOps Automation" rather than just standard Application Lifecycle Management (ALM) or enterprise management tools? For our analysis, we have identified and assessed the relative importance of the following Disruption Vectors, marking out criteria that specifically aid DevOps automation:

- Lifecycle coverage
- Process automation
- Process governance
- Infrastructure automation
- Team collaboration

- Analytics and machine learning
- Third-party integration

The Disruption Vectors section of this report expands upon these and features a visualization, showing the relative importance of each. Gigaom's analysis process also assigns a score from one to five to each company for each vector. The combination of these scores, and the relative weighting and importance of the vectors, drives the company index across all vectors. The outcome is the Sector Roadmap chart found in the Company Analysis section. Decision makers and tech buyers can use this to understand where DevOps automation tools and their providers go above and beyond traditional ALM and IT service management solutions.

3 Usage Scenarios

An organization may be a traditional enterprise, with groups/teams at varying stages along the DevOps journey, or it could be a cloud-first or Web-scale organization formed without creating an arbitrary line between development and operational activities. Some companies may have a hybrid of each; for example, the traditional mutual bank Crédit Agricole in France chose to spin out a new subsidiary to create its financial app store. Equally, an organization may also choose to outsource some or all of its development and operations to a third party.

On this basis, we can consider DevOps automation usage from the perspective of the kinds of situation being tackled and the maturity of the organization.

Development and deployment situations

While DevOps may suit a range of software development and deployment scenarios, it is particularly appropriate for the following:

- **Fast-changing or poorly understood requirements.** Given the pace of change of technology and customer expectations, it is not always possible to define the “ideal” solution in advance. Creating and testing something small, potentially failing, and learning lessons, can be far more efficient than building something big from the outset.
- **Proof-of-concept environments.** An organization may need the flexibility to test models and optimize them prior to a larger roll-out. Enterprises can use proof-of-concept environments not only to test the viability of a package or app, but also to determine its suitability and gain experience of the DevOps approach.
- **Client collaboration or co-creation.** Organizations are increasingly looking to partner with clients, customers, and other users to determine requirements and test out ideas, involving new types of interaction such as co-creation and ideation workshops. Such approaches work alongside DevOps, enabling stakeholders to define and improve potential solutions with minimal overhead.
- **Systems in maintenance.** Applications and services that have been created in the traditional way can benefit from a DevOps approach, to automate and ease both ongoing development and management. The same principles can be applied to other areas, such as analytics and business reporting.

To understand whether DevOps is appropriate, organizations can ask themselves whether any of the above situations apply. If so, DevOps automation will also need to be part of the answer.

Organizational Maturity

An organization’s maturity in relation to its adoption of DevOps will have a bearing on the focus and level of automation to put in place. There are no hard rules, but:

- **Build and deployment optimization** are usually considered first, initially prioritizing continuous integration, then continuous provisioning and deployment.
- **Reporting and feedback** tend to come next, enabling teams to improve co-ordination and collaboration, increasing efficiency and visibility
- **Testing and quality control**, including security testing, runs in parallel to other activities. Fully automated and comprehensive testing may be less of a priority if direct budget is not available.

In practice, automation is both a symptom and a cause of DevOps maturity – more mature organizations will see the benefits of automation to deliver better software and achieve value faster. We would advise a governance-first approach that builds in testing and quality control from the outset.

4 Disruption Vectors

All scenarios in which development and operations work together more closely can be seen as beneficial, as any removal of barriers will increase efficiency, productivity, and so on. For DevOps automation to be seen as disruptive however, it needs to deliver the kinds of exponential acceleration of delivery cycle times that have had such an impact in start-ups and Web-scale organizations. In practice, this means minimal or seamless interface from development to operations, enabling rapid and repeated software delivery (called Continuous Deployment in DevOps circles). The result is more than just efficiency, as it enables repeated testing of ideas and therefore creates a basis for greatly accelerated innovation.

Cloud-based architectures are an important piece of this puzzle as such rapid deployment cycles would be very difficult without highly virtualized infrastructure resources (as delivered by public or private cloud infrastructures). So, while many principles of DevOps can be applied without cloud as a target, enterprises should be looking to cloud-based target architectures if they are looking to achieve the same level of innovation and agility as start-ups.

With this in mind and as shown in Figure 2, the vectors we have identified are as follows: each of these factors are geared towards delivering on the disruptive nature of rapid, continuous development, delivery and deployment with the resulting benefit to innovation. Technology decision makers can use this list as a starting point to conduct a more comprehensive evaluation of specific tools, according to their own needs:

- Lifecycle coverage
- Process automation
- Process governance
- Infrastructure automation
- Team collaboration
- Analytics and machine learning
- Third party integration

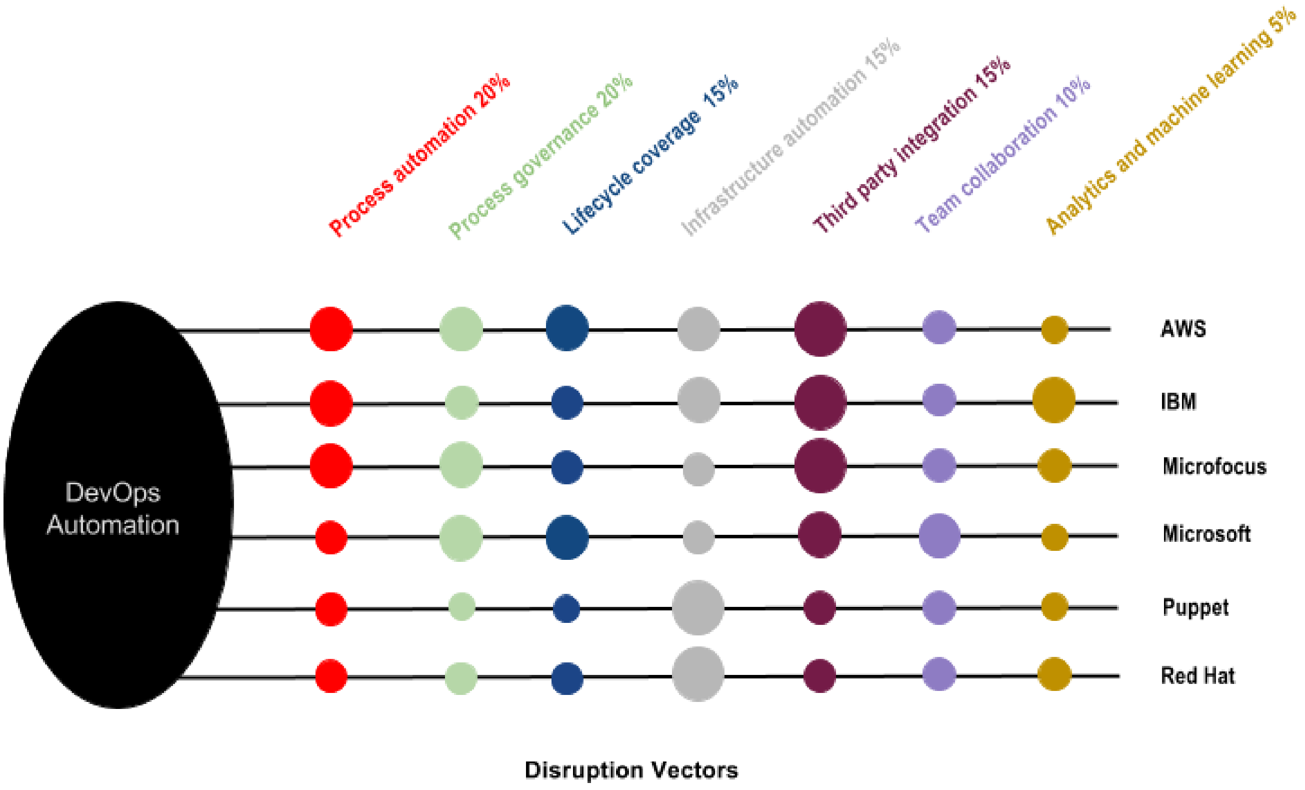


Figure 2. Disruption vectors and scores

We look at how these vectors apply to DevOps automation below. Note that one potential vector, covering whether a vendor operates a platform of its own, has been ruled out. Several larger vendors see this as a differentiator, not least because tools and platform work more closely to create additional efficiencies. While a suitable toolchain undoubtedly creates increased “platform stickiness,” this could be both a strength and a longer-term weakness for enterprise organizations.

Lifecycle coverage

DevOps automation tools are graded on how they cover the entire lifecycle of an application or service, with emphasis on both the hand-off to operations at deployment and the feedback loop back to development. While vendors with large portfolios of tools can claim to deliver across the lifecycle, this does not mean anything if tools do not integrate. Note that while vendors have tackled the link to operations head-on, closing the loop back to development remains a work in progress.

Process automation

This vector goes to the heart of what DevOps automation is all about: without process automation, it is unlikely that continuous development, integration, testing, or deployment cycles could take place multiple times a day. As with Infrastructure-as-Code (below) it is possible to codify process requirements, enabling an entire environment to be re-created. For example, an application in maintenance could incorporate the test automation scripts used when it was first tested.

Process governance

Contrary to the belief that lean development and operations approaches are an excuse for controlled chaos, successful DevOps requires highly structured, tightly controlled workflows. DevOps automation enables engineers to get on with the job, offering self-service access while ensuring that criteria such as role-based access, separation of concerns, data sovereignty, and security are met. Features such as portfolio or value management enable decision makers to allocate resources and ensure business outcomes are achieved.

Infrastructure automation

With today's infrastructure and networking platforms increasingly virtualized, it is altogether possible to define the hardware (server, storage and network) configuration upon which a software application needs to run, using an appropriate description language, and manage this information alongside code. So-called Infrastructure-as-Code enables an entire target environment to be "spun up" in a matter of minutes, and is a significant accelerator in how applications are created and deployed, therefore the potential for DevOps to disrupt.

Team collaboration

DevOps success is predicated on breaking down barriers between stakeholders – not only development and operations, but also testing and QA teams, contractors and partners, and internal users and external customers. Automated collaboration features or integrations become of prime importance, helping multi-functional teams and broader groups work together as closely as possible. In DevOps circles, the moniker ChatOps has been adopted to underline the significance of this criterion to accelerating innovation and delivery.

Analytics and machine learning

Software development and operations environments generate large quantities of data, alongside customer usage data and contextual information (e.g. customer numbers in a retail mall or social sentiment). DevOps automation tools can analyze data sources to help prioritize requirements and diagnose faults, as well as feeding process and infrastructure automation; for example, allowing deployment of low-risk features (confirmed by set criteria) without intervention. While few vendors are yet to integrate analytics, we expect to see this as a growth area in future.

Third party integration

To deliver on DevOps needs, automation tools need to integrate (via an Application Programming Interface (API)) with the plethora of software development tools now available and hook into (frequently cloud-based) infrastructure and platforms, enabling automated provisioning and orchestration of infrastructure resources. Given how all tools have some form of API, this criterion is worth more if a vendor's offering enables the integration of third-party tools across the lifecycle, as well as being able to integrate with other vendors.

5 Company Analysis

In this section, we consider several vendors who deliver on some or all the disruption vectors above. Each offers a strong set of features, making them suitable for most usage scenarios described earlier in this report. Listed alphabetically, these are:

- Amazon Web Services (AWS)
- Micro Focus (formerly HPE Software)
- IBM
- Microsoft
- Puppet
- Red Hat

Vendors have been evaluated based on their responses, on information to found on the Web, prior briefings, and industry research. Marks have been awarded based on whether a vendor offers a specific capability itself, as opposed to working with a third party. Of the above, AWS, HPE, IBM and Puppet have been interviewed directly for this report.

Amazon Web Services (AWS)

@awscloud | https://aws.amazon.com

Amazon.com's AWS subsidiary offers an extensive portfolio of tools and services built around its cloud platform. At the top level, AWS offers the CodeStar development environment, which incorporates four subservices:

- AWS CodeCommit – source code management
- AWS CodePipeline – automated build, test, and deployment
- AWS CodeBuild – enables build and test operations to be conducted in parallel
- AWS CodeDeploy – to deliver built software into the AWS live environment

AWS CodeStar provides an abstraction layer that allows developers to establish a toolchain, “with a few clicks in the AWS console.” The approach is template-based, while offering customization options and ensuring the whole lifecycle is covered. This also fits with the community-based experience sharing approach prevalent across development and operations in general, and DevOps in particular.

AWS CodeStar incorporates general collaboration features such as visibility on commits, as well as comprehensive role-based access, ensuring governance rules are kept. Functions integrate into the AWS service portfolio, enabling direct deployment into AWS Elastic Beanstalk, Amazon EC2 instances, SaaS offerings, or indeed the AWS Lambda serverless platform. Infrastructure characteristics such as elasticity, or regionalization for governance purposes, can be defined and deployed directly in an Infrastructure-as-Code way.

The AWS platform integrates with a wide variety of development tools, including the majority used in DevOps environments as well as testing, orchestration, and project management, available via the AWS marketplace. While supported by third parties, perhaps the only area of weakness in AWS’ own portfolio is in analytics use for DevOps. While AWS has made advances in machine learning and Web analytics, it has not progressed as far as some of its competitors in applying these ideas to DevOps.

Overall, AWS scores highly on most areas. On the topic of governance, an additional strength is AWS’ fully-managed service offering, which will increase in importance as DevOps matures and embraces IT service management practices such as ITIL. AWS CodeStar and its subservices only work with AWS targets, so it would not make sense to consider the former if the organization is not planning to use the latter; having said this, the comprehensive nature of automation tooling provided by AWS is a strong selling point for using the vendor’s cloud-based infrastructure and services.

Micro Focus (formerly HPE Software)

@MicroFocus | https://software.microfocus.com

Micro Focus (formerly HPE Software) released ALM Octane, a SaaS-based platform targeted at agile and DevOps environments in June 2016. While the company has a broad portfolio of tools supporting developers, operators, and senior managers, the vendor does not look to sell a full-lifecycle bundle, preferring to help organizations define their own lifecycles with appropriate and familiar DevOps tools. Alongside integration with the popular Jenkins platform and the support it offers to commonly-used open source tools, Micro Focus offers:

- Micro Focus Codar (previously HPE Software Codar) – a continuous deployment solution offering automation and release management for multi-tier applications.

- Micro Focus Release Control (previously Serena Release Control) – an enterprise release automation and management solution to automate software release, planning and tracking.

Micro Focus has adopted a governance-first approach to DevOps, building security and compliance into the pipeline based on tools such as Fortify and ArcSight, incorporating role-based access, and automating the creation of audit trails. Test automation and management was the bread and butter for HPE Software prior to its merger with Micro Focus, and continues to be on the critical path of DevOps best practice. This "governance by design" approach not only starts the lifecycle on the right foot; it also offers reassurance to organizations that see DevOps as a way of cutting corners, compared to traditional software development or service management approaches.

The company's data driven approach is a strength, lending itself to the use of analytics across the lifecycle. The future goal is to enable predictive and automated decision making based on, for example, analysis of usage trends, or by incorporating user experience data (via Micro Focus AppPulse) to set priorities on features. Once this is done, it is expected that Micro Focus will rank more highly on the analytics disruption vector. Micro Focus is also building ChatOps capabilities into each area of lifecycle, increasing its use of chat bots that can automatically respond to messages or instigate actions.

Micro Focus's go-to-market is services-led, with a focus on building automated pipelines for continuous delivery with the right kinds of feedback loops to close the DevOps cycle. The company also offers infrastructure deployment and management solutions for Hybrid IT environments, for IT Operations teams looking to build DevOps capabilities. Overall, while the organization does not offer tools to support the complete DevOps lifecycle, its emphasis on governance and testing and its increased use of analytics make it an attractive proposition. Note that effective September 1, 2017 HPE Software became part of Micro Focus.

IBM

@IBMDevOps | https://www.ibm.com/devops

IBM tends to work with established enterprises, and this perspective colors much of what IBM brings to the market. IBM offers on-ramps to DevOps approaches for organizations dealing with legacy infrastructures and organizational structures. Indeed, thinking through its IBM Cloud Garage Method, which emerged from IBM's own experiences of building for cloud and adopting DevOps practices, they answer questions such as how to work across teams that operate at different "cadences," which DevOps practices to prioritize, or how to manage APIs across multiple workload types.

IBM's decision to embrace open source in development and operational environments sets the scene for its DevOps automation approach. The company's Bluemix Continuous Delivery service incorporates the Eclipse Orion web-based development environment, Git Repositories and Issue Tracking, and a Delivery Pipeline, which in turn works with all the expected open source and third-party DevOps tools. IBM is largely agnostic about the tools to choose, though it offers a variety of templates for common toolchains and target environments.

IBM is also open about the deployment target; as well as its own Bluemix cloud platform, the company provides deployment capabilities for on-premises and across multiple clouds, including AWS, Azure, and Google with IBM UrbanCode. IBM works with containers such as Docker and with orchestration tools such as Kubernetes, available in both the Bluemix public cloud and IBM Cloud private.

A strength is how IBM brings cognitive analytics into the mix, from more standard insights about delivery speed, to diagnosis of potential risks (e.g. identifying a "hero programmer" who distorts productivity figures and creates the potential for bottlenecks), and looking to the future, integrating customer and other contextual data. The goal is to support automation of decision making across the DevOps cycle and therefore increase velocity while ensuring quality.

In summary, IBM offers a template-based portfolio and approach geared around giving enterprise organizations a starting point for the adoption of more efficient, effective DevOps practices and tools. While start-ups might not immediately see IBM as an appropriate partner, the vendor should not be ruled out, particularly as it sees the value of optimizing the whole loop through operations and back.

Microsoft

@MSCloud | https://www.visualstudio.com/vs/devops/

Microsoft's DevOps offering has evolved out of the vendor's portfolio for ALM and agile development, and as such it offers comprehensive support for the developer side of the lifecycle, across planning and source control, continuous development and integration, and indeed collaboration and information sharing. As such it does not have any specific components; rather, the whole environment lends itself to DevOps workflows. For example:

- Visual Studio Team Services integrates with tools such as Jenkins for build management, Ansible for automated deployment (see Red Hat), Selenium for testing, and Trello for collaboration.

- Azure portal can be configured to support continuous delivery to the Azure App Service hosting platform. App Service can also be used as a target environment from either Bit Bucket or GitHub.
- Application Insights and Azure Diagnostics monitoring and log data can provide enhanced insights across the DevOps cycle, triggering alerts and integrating with automation services to link back to development.
- Azure Automation enables configuration-as-code (based on PowerShell runbooks, increasingly working with third-party platforms such as Chef or Ansible) for continuous deployment, desired state conformance checks, and predefining monitoring criteria for applications in production.

The company sees the term "Cloud Cadence" at the heart of what it brings to the party: a tightly integrated solution delivery environment and stack. So, for example, Microsoft tools natively target Azure. The Microsoft environment is also multi-platform, supporting several development languages across Windows, Linux, and OSX desktops. Meanwhile, third-party options exist to deploy to other hosting platforms; for example, AWS offers a plug-in toolkit for Visual Studio.

The ensemble works well together, benefiting from facilities such as identity management to enable role-based access to specific services. Over time, Microsoft intends to broaden the level of integration, automation, and insight across its own environment, and to integrate with third-party offerings. Focus areas include policy-based management, predictive maintenance, auto-remediation, and security across applications and infrastructure. Overall, Microsoft offers a solid and well-governed platform of capabilities which will doubtless quickly evolve alongside its customers' DevOps needs as they mature.

Puppet

@puppetize | http://www.puppet.com/

Unlike many other vendors on this list, Puppet's initial premise was to solve problems for system administrators rather than developers. Puppet's goal was to release sysadmins from the tribulations of day-to-day operations so they could get on with more valuable activities. An innovator in this space, it was one of the first organizations to use the phrase "Infrastructure-as-Code," which it articulates in terms of "manifests" describing target infrastructure configurations. It does this through tools including:

- Puppet Enterprise – its flagship product that offers Web-based tools to manage manifests and auto-provision infrastructure, as well as monitor workflows and deployments.

- Puppet Forge – A repository of over 5,000 modules created by Puppet and the community, in four tiers from fully supported by Puppet, through partner-supported and Puppet-approved, to community contributed modules.
- Distelli, a continuous delivery automation platform that supports deployment to Kubernetes clusters and virtual machines.

Understandably, Puppet's primary success has come from Web-scale companies and startups who, by extension, could not have achieved such rapid innovation and growth without the rapid cycle times that DevOps in general, and infrastructure automation in particular, can bring. In recent years, Puppet has seen more traction from enterprise companies as they look to adopt DevOps and infrastructure automation practices, in specific areas at first, but growing more broadly and incorporating both legacy systems and new technologies, such as cloud and containers.

The question of how to scale DevOps enterprise-wide is therefore front and center, driving Puppet's strategy and customer engagement. This includes incorporating infrastructure and cloud discovery tools into the portfolio as an on-ramp to DevOps automation for container/cloud-based and legacy workloads alike. Puppet is agnostic about the environment it defines, making it a natural progression for the company to turn its hand to containers (and most recently, serverless approaches) as its customers look towards such approaches, but it is equally comfortable working in the mainframe environment.

For the future, the organization continues down the path of codifying both infrastructure and how it is architected; for example, enabling policies to be set around elasticity criteria that go beyond simple thresholds, or defining characteristics of how an application upgrade should take place. In addition, more emphasis will be placed on the DevOps governance. While it is in early days, serverless models are seen as the next frontier that will need to be codified and managed accordingly.

It is fair to say that while Infrastructure-as-Code tools such as Puppet can work outside of DevOps, DevOps could probably not exist without Infrastructure-as-Code. Puppet is not wed to any one target platform and can therefore exist more independently than vendors offering a target platform of their own. In evaluation terms, Puppet has not historically attempted to offer as broad a portfolio as others on this list but this is changing with the company's acquisition of Distelli, enabling the organization to respond to a broader set of DevOps use cases and increase its engagement with developers. As with Red Hat's Ansible below, for organizations beginning a journey to DevOps and who have a good handle on the software development side of the house, Puppet offers a place to start.

[1] Note that Satellite is currently backed by Puppet, but is moving to Ansible.

Red Hat

@redhatnews | redhat.com

Red Hat positions its DevOps automation approach at the point where development meets operations, ensuring that the handoff between dev and ops is as simple as possible. The vendor acquired Ansible in 2015, and has subsequently turned the Infrastructure-as-Code product (though Red Hat doesn't use this term) into a backbone for other Red Hat tools, including:

- Satellite – an infrastructure provisioning and management tool, supporting patching and lifecycle management of server resources
- CloudForms – a cloud management platform that can work across a hybrid cloud infrastructure environment
- Tower – an extensible automation and governance platform underpinning role-based management, monitoring, and reporting of Ansible jobs across the hybrid cloud environment
- Insights – a predictive analytics and machine learning capability, aimed at proactively identifying and preventing both configuration and security issues

Each tool integrates with the others, as well as third-party offerings: for example, if Insights identifies changes required for security or performance, it can directly generate Ansible playbooks that can then be pushed to a Tower installation; Ansible can provision assets requested via CloudForms; and so on. From a governance perspective, this enables better control over roles, as activities can take place without handing over root privileges for the target environment. Meanwhile Ansible Galaxy – a community hub for sharing Ansible information – enables sharing of templates and expertise. This help participants get started more quickly as they can adapt existing templates, and further reduces the need for third-party support.

Red Hat sees its Ansible-centric approach as a driver for DevOps success, acceptance, and growth, with the benefit that it enables a rethink of how infrastructure is treated; for example, developers can test an application in the same production-quality configuration as if it were live, repeatedly and even to the point of destruction. Red Hat prefers to leave software development lifecycle aspects to third parties, and it does place the same amount of emphasis on the point where operations re-engages with development. Like Puppet, Red Hat offers a starting point for organizations whose software development needs are met (or on track to be) and who are now looking to bridge the gap to operations.

6 Outlook and Key Takeaways

For all the fanfare and hype, DevOps is a natural and logical evolution of IT best practice. Cloud-first and Web-scale companies have leapfrogged traditional enterprises, leading to significantly greater innovation and market disruption. This would not have been possible without practices that put speed and collaboration before certainty and isolation. DevOps automation tools have both evolved from this revolution, and catalyzed it further.

All the same, it would be a mistake to think that the traditional roles have gone away. The developer mindset is one of problem solving, creation, and delivery, whereas the operational mindset prioritizes stability, security, and certainty. Maybe at some point in the future, infrastructure will become so automated that it can literally run itself, but this is unlikely; each wave of advances tends to be pushed to the limits by the latest wave of innovation, and so the cycle continues.

DevOps automation tools are a work in progress. At this point in their evolution, best-of-breed solutions are favored over commodity offerings; manual intervention is still required at most stages of the cycle, even if vendors work hard at developing pre-defined templates; standardized dashboards and metrics are still to come; and so on. However, enterprise organizations stand to gain a great deal by adopting and automating DevOps best practices and associated automation. The main benefit is that, in these uncertain times, results can be delivered faster, at lower cost.

We expect to see automation increasing, as agile practices are embraced across the operational side of the house. The dilemma: service management approaches such as ITIL remain the right answer even in an agile world, but are their own worst enemy as they create an excuse for inertia. Just as development drives operations, so operations will need to become a peer to development, setting priorities for innovation and driving the organization forward. In consequence, data will play an increasing role, enabling an increasing number of decisions to be automated while keeping project and service managers apprised. Some vendors are already looking at analytics and machine learning, and we expect this to become the norm rather than the exception.

Ultimately DevOps principles do not need further proof; Web-scale businesses have done so a thousand-fold. Rather, an enterprise needs to determine how to turn these principles into practice, using automation tools that enable innovation and deliver business value. Success may sometimes feel like a lottery, but not taking part is a guarantee of failure.