

Articles

- DevOps Principles
- DevOps Frameworks
- ▼ DevOps Tools
 - Overview
 - **Considerations for your DevOps toolchain**
 - DevOps Monitoring
 - DevOps Pipeline
 - DevSecOps Tools
 - Test Automation

Tutorials

- Automation
- Testing
- Security
- Observability
- ▼ Feature Flags

Considerations for your DevOps toolchain

Should you choose an all-in-one or customizable DevOps toolchain?



ROBERT KROHN

Head of Engineering, DevOps

EDITORIAL CONTRIBUTION: CHANDLER HARRIS

A DevOps toolchain is a collection of tools, often from a variety of vendors, that operate as an integrated unit to design, build, test, manage, measure, and operate software and systems. It enables development and operations teams to collaborate across the entire product lifecycle and tackles key DevOps fundamentals including continuous integration, continuous delivery, automation, and collaboration.

From agile to DevOps

When agile principles gained widespread adoption, they revolutionized the way we create products. We embraced small, cross-functional teams and one to two week sprints that produced artifacts ready for production. We had tight feedback cycles and continuous improvement. We delivered products faster and with fewer headaches.

It's amazing how quickly things change. Now with cloud, SaaS, and always-on services, the development lifecycle is much faster. It's common for multiple development and testing stages to happen simultaneously. While agile typically had one to two week sprints, teams in today's cloud native environments iterate and deploy multiple times a day. Workflows and codebases evolve constantly. Teams use **feature flags**, progressive roll-outs and A/B tests to ensure clean, quality code deploys continuously.

This has spawned an evolution from agile to DevOps, a set of practices that works to automate and integrate the processes between software development and IT teams, allowing them to build, test, and release software faster and more reliably. DevOps teams are measured by how quickly they push code into production. They are defined by the number of iterations shipped each day and how long it takes for a code change to go from testing through deployment and into production. Typical projects on my team see up to 20 changes per day. Some larger projects see up to 100 changes per day.

This accelerated workflow is enabled by, and relies on, new tools that allow teams to collaborate across development, testing, and deployment. In particular, a DevOps toolchain helps teams tackle each stage of the development lifecycle, very, very quickly.

What is a DevOps toolchain?

A DevOps toolchain includes the tools and technology that enable development and operations teams to collaborate across the entire software lifecycle. It tackles key DevOps fundamentals including continuous integration, continuous delivery, automation, and collaboration.



Since DevOps is a cultural shift where development and operations work as an integrated unit, there isn't a single tool that enables DevOps principles and practices.

Since DevOps is a cultural shift where development and operations work as an integrated unit, there isn't a single tool that enables DevOps principles and practices. Instead, a DevOps toolchain is a collection of tools, often from a variety of vendors, that operate as an integrated unit to design, build, test, manage, measure, and operate software and systems. Often, organizations and teams need to experiment with different combinations of tools to find the right toolchain.

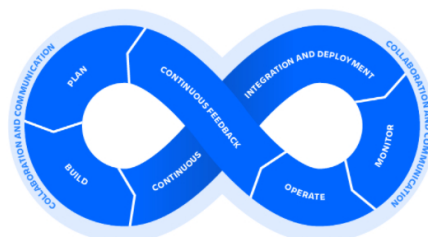
If you look across a sophisticated DevOps product, the DevOps toolchain should be able to tackle different parts of the development lifecycle very quickly and offer multiple lenses for different users. It should include development tools that address each phase of the development lifecycle, including continuous integration and delivery, test automation, and high-speed deployment. On the operations side of DevOps, tools should include functionality that helps with monitoring and incident management. Plus, tools should bridge development and operations by providing continuous feedback and logging.



RELATED MATERIAL

Get started for free[Learn more →](#)

RELATED MATERIAL

Explore integrations[Learn more →](#)

When you consider the left-hand side of the infinity loop as the product side and the right-hand side as the operation side, the product manager who pushes a new feature into production is interested in seeing how the project breaks up into tasks and user stories. The developer on the left side of the project needs to see how to move the feature into production including project tickets, users stories, and dependencies. If developers adhere to the DevOps principle of “you build it, you run it”, they are also interested in incident remediation.

Moving to the operations side of the life cycle, the site reliability engineer needs to understand the services that can be measured and monitored, so if there's a problem, it can be fixed. If you don't have a toolchain that ties all these processes together, you have a messy, uncorrelated, chaotic environment. If you have a well-integrated toolchain, you can get better context into what is going on.

Options for Building Your DevOps Toolchain

When deciding the right DevOps toolchain, it's important first to understand fundamental DevOps best practices and how tools assist these practices. Then, establish a common tools strategy that allows teams to collaborate across development, testing, and deployment.

When organizations adopt DevOps they are usually faced with two choices -- an “all-in-one” DevOps toolchain or a customized DevOps toolchain. Choosing the right configuration is important since it shapes the DevOps processes of a team.

All-in-one DevOps toolchain

An all-in-one DevOps solution provides a complete solution that may not integrate with other third-party tools. This can be useful for companies or groups just beginning their DevOps journey, or if a team wants to start a project quickly. The downside of this type of toolchain is that most established teams already have a set of tools they use and prefer, which may not integrate with a complete solution. Plus, such a comprehensive toolchain can suffer from the “jack of all trades, master of none” syndrome. One tool simply can't evolve to fast-changing markets. Finally, more often than not companies need to integrate legacy tools into a DevOps toolchain and an all-in-one toolchain can limit this.

Customizable DevOps toolchain

The other approach is to use a DevOps toolchain that can be customized for a team's needs with different tools. This allows teams to bring the existing tools they know and love into the wider DevOps toolchain. For example, a team can use Jira for planning and workflow tracking, Kubernetes to provision individual development environments, Github for collaborative coding, Jenkins for continuous integration, and more. Organizations can customize their workflows by teams and/or by project.

Integration is essential for these types of toolchains. If the different tools don't integrate, team members spend unnecessary time switching between screens, logging in to multiple places, and it may be challenging to share information between tools. That's not a great experience for developers or anyone trying to understand what's going on. If you're responding to an incident, you don't have time to look at a manual and try to find key information about a new tool.

In conclusion...

DevOps is about breaking down silos and making the development life cycle faster, highly automated, and seamlessly collaborative -- whether it's cross-functional or within one team. Choosing the right DevOps tools that work together is first and foremost about taking a hard look at your current software development and IT operations process and deciding where you need to improve.

Read more about [tools for each phase of the DevOps lifecycle](#).



ROBERT KROHN

Robert Krohn is the Head Of Engineering, DevOps at Atlassian. He has more than 20 years of software engineering, management, and executive experience spanning software, hardware, and operations, across diverse product segments. He is passionate about building world-class engineering teams and market-leading products. He loves to ride bikes, sail his Laser, and snow ski. He lives in Belvedere, California with his wife Nicole, kids, dogs, and cat.

SHARE THIS ARTICLE



NEXT TOPIC

[DevOps Engineer](#) →

Recommended reading

Bookmark these resources to learn about types of DevOps teams, or for ongoing updates about DevOps at Atlassian.





DevOps community

[Learn more →](#)



Simulation workshop

[Learn more →](#)



Get started for free

[Learn more →](#)

Sign up for our DevOps newsletter

Email address

[Sign up](#)

ATlassian

PRODUCTS

Jira Software
Jira Align
Jira Service Management
Confluence
Trello
Bitbucket

[View all products](#)

RESOURCES

Technical Support
Purchasing & licensing
Atlassian Community
Knowledge base
Marketplace
My Account

[Create support ticket](#)

EXPAND & LEARN

Partners
Training & Certification
Documentation
Developer Resources
Purchasing FAQ
Enterprise services

[View all resources](#)

ABOUT ATLASSIAN

Company
Careers
Events
Blogs
Investor Relations
Trust & Security

[Contact us](#)

 English ▼

[Privacy policy](#)

[Terms](#)

[Impressum](#)

Copyright © 2021 Atlassian

