THE GUIDE TO DEVOPS AND SECURITY TOOL CONSOLIDATION

Best practices, tips, and areas to start consolidating your tool sets for greater efficiency, security, and cost optimization.

JFrog

Table of contents:

Too Many Tools, Not Enough Value

Are you overwhelmed by too many tools? You're not alone. According to IDC, most companies use anywhere from six to 20 or more tools to track and monitor their software development life cycle (SDLC). With so many DevOps and security tools, and organizations giving dev teams the freedom to pick their own, tool sprawl occurs. The data shows, and it's no surprise, that DevOps and Security teams are now considering tool consolidation.

The process of tool consolidation combines redundant tools or chooses a single tool for multiple functions. Having too many DevOps and security tools can hinder workflows and processes, thereby reducing efficiency. Tool consolidation reduces the number of tools an organization uses to streamline workflows, standardize processes, and increase efficiency.

At the same time, DevOps teams are being tasked with integrating security into their tool stack. A recent survey by Gartner, Inc. found that 75% of organizations are pursuing security vendor consolidation, up from 29% in 2020. Security and risk management leaders are increasingly dissatisfied with the operational inefficiencies and the lack of integration of a heterogenous security stack," said John Watts, VP Analyst at Gartner. "As a result, they are consolidating the number of security vendors they use."

By consolidating multiple tools into a single source of truth, economies of scale can be achieved, which can help reduce capital and operational expenditure over time. Consolidating tools can also reduce overall vendor commitments and relationships, which can improve visibility and transparency into the organization's spending.

However, most single vendors won't deliver all of the features you want, at the caliber you need, so you'll need to identify which features and use cases are most important to you before you consolidate your toolsets. This will help you make sure you're consolidating appropriately, and that you're taking advantage of the capabilities offered by each of your tools.

In this ebook, we'll guide you through:

Part 1

Areas for tool consolidation in your DevOps toolchain

Part 1: Areas for tool consolidation in your DevOps toolchain

When thinking about tool consolidation, be sure to consider the various teams involved. Some teams may rely on the same tools you're trying to consolidate. Or perhaps those teams are also planning for consolidation, thus duplicating your efforts. So, be mindful when making tool consolidation decisions. Consider consolidating the following tools in your tool stack:

- **Artifact storage:** Organizations often store artifacts and components in multiple places, including: package-specific repositories and S3 buckets, Version Control Systems (VCS), and other options. This makes it hard to automate the process of tracking and updating the artifacts.
- **Container registries:** To avoid inconsistency and errors when working with multiple images across different environments, teams should centralize image management solutions.
- **Software security solutions:** Organizations today leverage multiple point security solutions that can leave gaps in coverage, provide redundant or complex results, and increase complexity in administration and management. Consolidating security tools can provide a simple pane-of glass view into the health of your supply chain and coverage over the entire SDLC.
- **Cloud/dynamic runtime deployment technologies:** Organizations rely on these technologies to quickly and easily deploy applications in the cloud or on-prem. By consolidating where these technologies/files are housed and maintained, teams can avoid managing multiple tools.
- **CI/CD tooling:** The use of CI/CD tools results in faster builds and deployments, which can lead to faster development, feedback cycles and application delivery. By consolidating CI/CD tooling, teams can more easily automate and monitor processes to quickly identify any issues and take corrective measures, reducing the risk of downtime and improving the reliability of applications.
- **Monitoring capabilities:** Use a single platform that can monitor multiple parts of your systems and processes to gauge development velocity. This will help streamline the monitoring process, provide insight into development gaps to improve velocity, and reduce the time spent managing multiple tools. Examples of such platforms include Datadog and Splunk.

Part 2

A five-step plan for consolidating your DevOps tools

Part 2: A five-step plan for consolidating your DevOps tools

With the influx of tools available, it can be difficult to keep track of all the changes that occur. To maintain optimal efficiency, it's essential to find a system to help manage the adoption, discarding, and updating of tools.

There are a few approaches you can take to finding the right areas of consolidation for your organization. So, how do you pick which tools to consolidate? Here's a five-step plan with best practices for tool consolidation with ideas for implementation.

1. **Create a DevOps and Security Tools Inventory**

   How to Implement: Begin by generating a comprehensive inventory of which tools are currently in use and how those tools are being used. Are they used for source code and version control management? Managing builds and dependencies? Or, are they deployment focused? To create your inventory, try the following:

   - ☐ Create an inventory of all the DevOps and application security tools currently in use. Include the names of the tools, the version numbers, and the purpose each tool is serving in your SDLC.
   - ☐ Identify any other viable purposes or additional functionality of your current tools you're not actively using. For example, many tools are now offering built-in security functionality.
   - ☐ Develop a process or cadence for regularly updating the inventory. This should include when new tools are added, when old tools are removed, and when existing tools are updated.
   - ☐ Create a system for assigning access controls to the inventory. This should include who can access the inventory and who has read-only access. Even if you keep your inventory on a simple spreadsheet, this recommendation still applies.
   - ☐ Use automation tools to keep the inventory up to date. This could include setting up scripts to automatically check for new versions of tools, or scheduling regular checks of the inventory.
   - ☐ Establish a way to monitor the usage of the DevOps tools. This could include tracking usage over time or setting up alerts when certain thresholds are passed.

FROG TIP: Creating an inventory is a recurring step. To keep your techstack optimized, continually review the tools in use and assess how they support or hinder DevOps processes.

2. **Identify what matters most in your tooling**

How To Implement: Analyze the inventory to determine which tools to consolidate or replace. Here's how you can identify areas of focus for tool consolidation:

- ☐ **Feature Sets:** Analyze the feature sets of the DevOps tools in the inventory to determine if any features overlap and can be consolidated into one tool. Also identify which features are key and where you want robust support vs "checking a box."
- ☐ **Maintenance Requirements:** Analyze the maintenance requirements for each tool to determine which can be consolidated or replaced. For example, does the vendor offer a SaaS or managed service approach? What is their support capacity? Are professional services available? DevOps teams should ask these questions to help better understand how to effectively allocate spend and improve . maintenance.
- ☐ **Cost:** DevOps teams should consider cost, but not solely. Before making any decisions, teams should identify the functional requirements of their operations and then do a cost analysis of actuals to determine if consolidation is the most cost-effective option. This cost analysis should take into account any potential cost savings from consolidating, such as reduced staff, infrastructure, and other related costs. Teams should also consider consolidating risks such as decreased flexibility.

FROG TIP: Not all tools can be consolidated. And consolidation often takes time to implement as old infrastructures might need to gradually sunset.

3. **Decide which focus areas to keep, combine, or phase out**

How To Implement: Now that you have a list of tools and key priorities, it's time to identify which areas are most important to either consolidate around or have a dedicated tool for. Consider evaluating the following focus areas that might support your consolidation efforts.

- ☐ **Automated Testing:** Identify gaps or overlaps in your automated testing. Review scope and types of tests to be conducted. Identify areas that aren't automated and review the scope and types of tests needed.
- ☐ **Source Control:** Source code and version control management is a natural area for organizations to consider consolidating tooling around if they haven't already. Especially with the rise of GitOps and the role it plays in enabling applications to run smoothly in production.
- ☐ **Monitoring and Logging:** The more visibility you have over your processes, infrastructure, and application, the better. But too many monitoring and logging tools generate data silos, which makes it difficult to analyze your data comprehensively when you need to.
- ☐ **Configuration Management:** Investigate which configuration management tools are being leveraged and how they're managed. Configuration management tools such as Chef, Puppet, and Ansible can be used to automate the deployment, configuration, and management of infrastructure and applications, but often introduce unique tools to manage each one versus provide single system to house and manage all of them.

- ☐ **Containerization:** There are multiple elements that go into delivering containerized applications into production — from building, securing, hosting, and running them — which each representing a potential tool to be leveraged by your software organization. If containerization is an important part of your application strategy, consider whether focusing consolidation efforts around this area makes sense.
- ☐ **Application Security:** Review what security tools you are using today, if you are facing any budgetary pressures, and how many tools and customizations are needed to deliver the comprehensive coverage required to secure your software supply chain. Consolidate application security tools to easily deliver consistent application security best practices and reduce spend on multiple tools. Consider the adage “Shift Left to Ship Right”.
- ☐ **Artifact Management:** Think about what your artifact management "end" is. For example, do you need to wrap your arms around third-party transitive dependencies and builds? Consolidating artifacts in a single repository manager means thinking about your current value stream management, curation, and archival capabilities, and whether your metadata is actionable.

FROG TIP: It's important to reiterate that some tools work better together than on their own. So, you might want to combine multiple tools to achieve the result youneed.

4. **Use vendors that bring tools together to speed up software updates:**

   How to implement: You can't consolidate everything, but an integrated approach can fill the gaps and connect tools, processes, and workflows. Research the vendor's roadmap to make sure it matches your needs. Consider whether the vendor offers tools with the following capabilities:

   - ☐ Extensive OOTB Integrations
   - ☐ Robust REST APIs paired with a CLI
   - ☐ Enables automation of processes
   - ☐ Available both on-premise and multiple clouds with consistent user experience
   - ☐ High-availability and disaster recovery support
   - ☐ Logging, monitoring, and analytics capabilities
   - ☐ Ensures security of the components, processes, and functionality
   - ☐ SSO and Access Controls support with robust role based access controls

FROG TIP: Make sure the tools you consolidate can work together, or seek out a platform you can use as the basis for your toolchain.

5. **Set up a plan to migrate to a consolidated set of DevSecOps tools**:

   How to Implement: Now that you've chosen a flexible vendor and decided which tools to consolidate, it's time to create a plan for migrating your datasets from each tool. Here's how to plan for migration:

   - ☐ Create a timeline and budget for the consolidation process.
   - ☐ Consider all policies related to each tool or process.
   - ☐ Execute the plan and monitor the process to ensure the successful consolidation of DevOps tools.
   - ☐ Train staff and users on the new consolidated DevOps tools.
   - ☐ Document the new consolidated DevOps toolchain and maintain it as part of your organization's IT infrastructure.
   - ☐ Develop and track a set of metrics that accurately reflect the business impact of your tool consolidation efforts.

FROG TIP: Schedule a meeting with Finance to discuss the budgetary impact of consolidation, including potential short-term spend increases and long-term savings.

Part 3

Determining ROI

Part 3: Determining ROI

Regarding tool consolidation in DevSecOps, one of the key considerations is ROI. One consideration when deciding whether to consolidate tooling is the cost vs. time benefit.

Did you know the Voice of the Customer report found?

90% of IT buyers want to work with fewer vendors

40% say there are too many pricing models

72% say pricing is too complex

Use this formula to calculate the cost of tool sprawl before making the decision to consolidate tooling:

License Cost + Infrastructure Cost + Maintenance Cost + Training Cost + Integration Cost

In addition to the formula to calculate the cost of tool sprawl, use this checklist to help you determine ROI for tool consolidation:

- ☐ By consolidating tools or moving to a new solution, will you gain or lose functionality that enables key use cases in your organization? How much work would be required to achieve feature parity using your existing tools and setup?
- ☐ Track metrics such as rework avoided and time invested in new features. These metrics can help provide a clear picture of how much time and money is being saved by consolidating tools.
- ☐ Consider the training required for employees to use new tools. While there may be an initial investment of time and money to train employees on new tools, this investment will pay off in the long run if it leads to increased efficiency and reduced number of toolsets to train future employees on.
  Consider these factors when deciding if tool consolidation is worth it for your organization.

FROG TIP: A flexible solution is more likely to provide a positive return than a rigid one.

Part 4

Common challenges when consolidating DevSecOps tools (and how to overcome them)

Part 4: Common challenges when consolidating DevSecOps tools (and how to overcome them)

When consolidating DevOps and security tools, consider these challenges and solutions when either implementing new tools or changing the infrastructure and architecture of others.

Challenge
Getting buy-in from all stakeholders

Solution
Include key stakeholders in the tool selection and/or consolidation process. Show them the value of the new solution you've chosen with the metrics you calculated using the ROI formula in Part 3.

Challenge
Consolidating and selecting the right tools

Solution
There are a lot of options available, so it's important to select the tools that'll best meet your needs. Use the five-step plan for consolidating your tools in Part 2 and consider the ROI using the formula in Part 3.

Challenge
Getting the right tools to work together

Solution
It's important to consider how well potential solutions connect and automate with other toolsets, as well as the flexibility of the solution. Reference the five-step plan for consolidating your tools in Part 2 to decide which tools you might combine, or which tools you might phase out based on your operational needs and/or budget.

Challenge
Training your team

Solution
Training the team on any new tools is essential for a successful transition. Without proper training, team members may not know how to use the tools effectively, and this can lead to increased build errors and delays in software development and distribution. By providing employees with the resources and training they need, you can set them up for success. See Part 2: A five-step plan for consolidating your DevSecOps tools.

Challenge
Managing up front expenses

Solution
For a time you'll need to maintain your existing tooling as well as your new or streamlined tooling. This can involve up-front costs for new solutions, and additional staffing hours to set up your new tools or handle migrations. Map out the long term savings and efficiency gains in order to justify the expenses and get budgetary approval using the ROI formula in Part 3.

Check out the adjacent subsection to learn more about real customer tool consolidations and their successes.

Three real-life examples of tool consolidations and their value

Here's how the JFrog Software Supply Chain Platform reduced spend and enhanced scalability, CI/DC workflows, and security posture for Hitachi Vantara, TomTom, and Yunex Traffic:

Hitachi Vantara

Problem: Multisite pipelines, siloed acquisitions, inconsistent security posture
Results: Consistent SSC practices, accelerated pipelines, shift-left security practices
Solutions: JFrog Artifactory, JFrog Xray, Amazon Web Services, Docker Hub
Value: Deliver consistent application of security practices, reduce spend on multiple tools

TOMTOM®

Problem: Build times, multiple platforms, unable to trace build context, or sustain CD cycle
Results: Multiple repositories, shortened build time, faster CD release
Solutions: Conan, JFrog Artifactory
Value: Refocus time away from managing multiple toolsets and meet budget requirements

YUNEX TRAFFIC

Problem: Multisite pipelines, siloed acquisitions, inconsistent security posture
Results: Consistent SSC practices, accelerated pipelines, shift-left security practices
Solutions: JFrog Artifactory, JFrog Xray, Amazon Web Services, Docker Hub
Value: Deliver consistent application of security practices, reduce spend on multiple tools

For more customer success stories, visit the JFrog Resource Center

Part 5

Consolidate with JFrog

Part 5: Consolidate with JFrog

JFrog's **Software Supply Chain Platform** offers a natural point of consolidation for many organizations. JFrog provides best-in-class solutions across three domains: DevOps, Security, and IoT all integrated onto a single platform. Organizations choosing JFrog can benefit from consolidating the following:

- **Package managers and package-specific tools:** Consolidate your component ecosystem into a well-governed, automated, and universal solution that serves as a single system of record for the entire company.
- **Container registry and deployment technologies:** Maintain full visibility and control of your applications through deployment. Incorporating your IaC and config files with your containers and build metadata in JFrog provides the most direct and traceable path of what is being delivered into production.
- **Software supply chain security:** Eliminate point security solutions in favor of an end-to-end application security solution that ensures accurate and continuous security across the supply chain.
- **Continuous integration:** Manage and orchestrate all of your pipelines in one place. JFrog delivers a native, enterprise-grade and security-focused CI solution that can also wrap around your existing workflows for "pipelines of pipelines."
- **Infrastructure, deployment, and platform automation:** Eliminate the need for additional third-party tools to manage your deployment and infrastructure technologies such as Terraform and Ansible. Create and manage custom workflows all from the JFrog Platform.
- **Edge and IoT management:** Benefit from a simplified approach to device updates that also includes remote diagnostics and monitoring, fleet management, and security monitoring.

With the JFrog Software Supply Chain Platform, your organization will benefit from the following:

Business Value	**DEEPER DEV + SEC + OPS INTEGRATION**
Benefit	✓ Deep platform, module, and application integrations on day 1 ✓ Shared database and unified data model ✓ Shared contextual information with integrated reporting ✓ Advanced automation and intelligent cross-domain policies ✓ Built-in high availability for non-stop operation ✓ A world-class 24/7 global support team

Business Value	**UNIFIED SECURITY, VISIBILITY, AND CONTROL**
Benefit	✓ Advanced SBOM-wide visibility ✓ Rich metadata capture and augmentation ✓ Rapid SBOM info extraction and contextualization ✓ Advanced reporting capabilities across orgs ✓ Insights and benchmarking drive data-driven optimization ✓ JFrog world-leading security researchers and engineers

Business Value	**FLEXIBLE, OPEN & EXPANDABLE BY DESIGN**
Benefit	✓ Extensive native package and file support (30+) ✓ Advanced multi-cloud capabilities ✓ Centralized or massively distributed (IoT) ✓ Partner pre-integrations out-of-the-box/Day 1 ✓ REST APIs for nearly everything paired with the open source JFrog CLI ✓ Start small or scale to managing millions of artifacts and builds on your timelines/budget

With JFrog, organizations can consolidate their DevOps and security tools to streamline their software delivery pipelines and automate their software releases.

Conclusion: Manage less, achieve more

As organizations look to cut costs while improving software supply chain security, tool consolidation is becoming a more urgent need. Consolidating DevOps tools reduces complexity, improves visibility and security, and enables teams to cross-collaborate creating a streamlined workflow that speeds software delivery and reduces operational costs.

The values and benefits are clear. Especially in today's rapidly moving world, the companies that can deploy and distribute software to market faster and with greater confidence will be the ones that survive and thrive.

About JFrog

JFrog is on a mission to create a world of software delivered without friction from developer to device. Driven by a “Liquid Software” vision, the JFrog Software Supply Chain Platform is a single system of record that powers organizations to build, manage, and distribute software quickly and securely, ensuring it is available, traceable, and tamper-proof. The integrated security features also help identify, protect, and remediate against threats and vulnerabilities. JFrog’s hybrid, universal, multi-cloud platform is available as both self-hosted and SaaS services across major cloud service providers. Millions of users and 7K+ customers worldwide, including a majority of the FORTUNE 100, depend on JFrog solutions to securely embrace digital transformation. Once you leap forward, you won’t go back.

www.jfrog.com

www.twitter.com/jfrog

www.facebook.com/artifrog/

www.linkedin.com/company/jfrog-ltd

The Liquid Software Company