



How JFrog Supports the AWS Well-Architected Framework

AWS Well-Architected helps cloud architects build secure, high-performing, resilient, and efficient infrastructure for a variety of applications and workloads. This guide explains how JFrog supports the operational excellence and security pillars.

[Learn more about AWS Well-Architected >](#)

Code	Topic	Common Anti-patterns	Relevant JFrog Solutions	Explanation
OPS05-BP04	Use build and deployment management systems Desired outcome: Your build and deployment management systems support your organization's continuous integration continuous delivery (CI/CD) system that provide capabilities for automating safe rollouts with the correct configurations.	After compiling your code on your development system, you copy the executable onto your production systems and it fails to start. The local log files indicate that it has failed due to missing dependencies.	- JFrog Release Lifecycle Management - Release Bundle Promotion - Release Bundle Distribution	JFrog's Release Lifecycle Management helps in ensuring that all build dependencies are shipped together with your production builds in Release Bundles. This include all associated artifacts which you want to ship together, regardless of technology type. Evidence-based release bundle promotion ensures transparency and traceability in releases.
		You successfully build your application with new features in your development environment and provide the code to quality assurance (QA). It fails QA because it is missing static assets.	- JFrog CLI for Developers - using the JFrog CLI to connect dev environments to JFrog Artifactory - JFrog integrations with all major build systems - GitHub Actions, Gitlab, Jenkins, Azure Pipelines, TeamCity, Bamboo	Connecting CI systems to Artifactory ensures that all dependencies are available to the build process in the same manner as they are to the development environment.
SEC04-BP03	Correlate and enrich security alerts	Your organization funnels all security finding and alert data to standard locations, but requires investigators to perform manual correlation and enrichment.	- JFrog Xray integration with Atlassian Jira - JFrog Xray integration with AWS Security Hub - JFrog Advanced Security - Contextual Analysis	- JFrog Xray's Jira integration allows users to automatically create Jira tickets, emails, or webhooks-based notifications for each discovered vulnerability, with detailed information about the CVE and links back to the binary and the context of the vulnerability. - Similarly, JFrog integrates with AWS Security Hub to enable AWS customers to view all automated security and license alerts in the Security Hub. - With JFrog Advanced Security, the task of correlating code to vulnerabilities is automated. The JFrog Research team provides detailed information with the Contextual Analysis feature of JFrog Advanced Security to ensure end-users understand the scope and remediation of vulnerabilities.
		You rely solely on the intelligence of threat detection systems to report on findings and establish criticality.	- JFrog Research - enriched vulnerability information	While CVE databases provide excellent information, the JFrog Research team adds valuable context and focuses on understanding code use and applicability, thus changing the criticality of vulnerabilities based on developer secure coding practices.

Code	Topic	Common Anti-patterns	Relevant JFrog Solutions	Explanation
SEC06-BP01	Perform vulnerability management	Not having a vulnerability management program.	- JFrog Advanced Security - SAST scanning - JFrog Xray integration with Atlassian Jira - JFrog Xray integration with AWS Security Hub - JFrog Advanced Security - Shift-left security - IDE plugins	- JFrog Advanced Security offers the ability to understand, prioritize, and manage vulnerabilities. Our SAST scanning capabilities allow deeper understanding and faster resolution of CVEs. - Further, our native integration with Jira allows you to easily assign and track vulnerabilities in an environment familiar to developers. - JFrog integrates with AWS Security Hub to enable AWS customers to view all automated security and license alerts in the Security Hub. - Lastly, JFrog Advanced Security enables developers to leverage shift-left security capabilities by running scans and viewing results from within the IDEs of their choice, in the context of their code. This enables them to ship secure code and fix vulnerabilities right away without needing any more resources or guidance from security teams.
		Using software that has passed its vendor-provided end of life (EOL) date.	- JFrog Xray - Operational Risk policies	With JFrog Xray, users can create policies that prevent EOL Open Source dependencies from being included in builds.
		Deploying code into production before analyzing it for security issues.	- JFrog Xray - integrations with CI systems - Frogbot PR scanning and repo auto-scanning - JFrog Advanced Security with SAST scanning	- JFrog Xray's tight integration with CI systems means every build automatically gets scanned. - FrogBot automatically scans all PRs and periodically scans repos and creates PRs with vulnerability fixes. - JFrog Advanced Security enables developers to run SAST scanning within their IDEs, giving them the results of our analysis within the context of their code. This enables a full shift-left approach to shipping secure code and encourages for a security-first mindset in development teams.
SEC06-BP04	Validate software integrity	Not signing your own software, code, or libraries, even when only used in your own deployments.	- JFrog Distribution - Release Bundles	Release Lifecycle Management with Release Bundles enables customers to sign releases with GPG encryption, allowing for secure and trusted deployment.
SEC08-BP03	Automate data at rest protection	Not enforcing public access settings for storage services.	- JFrog Advanced Security - IAC Exposure scanning - S3 Bucket exposure scanning	JFrog Advanced Security scans terraform plans and state files to look for security issues and excessive permissions. It lays a special focus on S3 Bucket permissions to ensure customer data at rest is not open to unauthorized users.

Code	Topic	Common Anti-patterns	Relevant JFrog Solutions	Explanation
SEC11-BP01	Train for application security	Leaving all security decisions to the security team.	- JFrog Advanced Security - Shift-left security - IDE plugins - JFrog CLI	JFrog Advanced Security aims to shift the power and knowledge of security left to the developers. With Contextual Analysis and IDE tools that leverage the all powerful JFrog CLI, developers can understand how their code is insecure and be informed on remediation steps from within their favorite IDEs.
		Engaging in the security review process too late.	- JFrog Xray - regular security review and policy application as part of CI cycle - JFrog Advanced Security - SAST scanning - JFrog Curation - Org-level automated security	JFrog Xray promotes a security process at the build level. JFrog Advanced Security enables developers to scan code on their development machines and understand security issues even before they ship the code. JFrog Curation enables organizations to set-and-forget policies that ensure developers can only download open source dependencies that pass set security standards.
SEC11-BP04	Manual code reviews	Not using automation to assist or orchestrate code reviews.	JFrog FrogBot	FrogBot performs vulnerability scanning as part of code reviews and posts the results in the code review for the developer and reviewers to analyze.
SEC11-BP05	Centralize services for packages and dependencies	Pulling packages from arbitrary repositories on the internet.	JFrog Artifactory	Artifactory is the ideal proxy and cache for getting packages from the Internet. Developers can depend on the stability and speed of the platform to store and manage packages at scale.
		Not testing new packages before making them available to builders.	JFrog Curation	With JFrog Curation, security folk can get information on new packages and take informed decisions. They can also audit logs to understand who is using which packages and form their Org-level policies based on trends they observe.
SEC11-BP06	Deploy software programmatically	Benefits of this best practice: -Increased confidence in the software release process. - Increased release cadence due to lower change risk. - Automatic rollback capability for unexpected events during deployment. - Ability to cryptographically prove that the software that was tested is the software deployed.	- JFrog Artifactory - JFrog Release Lifecycle Management - JFrog Connect - JFrog Evidence	- JFrog Artifactory is the Single Source of Truth for your CI and CD. - JFrog Release Lifecycle Management and Release Bundles ensure that you can cryptographically sign software and pass it through evidence based build promotion stages. - With JFrog Connect, you can deploy IoT software with confidence, so that if there's any issue during deployment, the software is rolled back to the previous version. - JFrog Release Lifecycle Management now includes Evidence collection, which users to easily capture proof of software maturity as it undergoes testing. This makes JFrog the single source of truth for attestation and simplifies auditing and compliance.

Code	Topic	Common Anti-patterns	Relevant JFrog Solutions	Explanation
SEC11-BP07	Regularly assess security properties of the pipelines	Not regularly reviewing the permissions associated with your CI/CD infrastructure.	JFrog Advanced Security with IAC Exposure Scanning	JFrog Advanced Security enables customers to scan their terraform plans and state files for permissions issues, security issues, and threat analysis.
		Use of long-term or hardcoded credentials.	- JFrog Advanced Security with Advanced Secrets Detection - HashiCorp Vault Secrets Plugin for JFrog Artifactory - JFrog Platform OIDC Connect for GitHub Actions	- JFrog Advanced Security scans your source code and binaries for potentially leaked secrets such as passwords, tokens, and API keys. We enforce build failure, webhook-based notification, and Jira ticket creation to ensure customers are aware of such secrets. - The HashiCorp Vault plugin created and maintained by JFrog allows users to connect the Vault with JFrog Artifactory and provision access tokens dynamically. This drastically limits the accidental exposure window of Artifactory tokens and is useful for CI servers.- Similarly, the JFrog Platform includes OIDC Connect setup for GitHub Actions provides the capability to create temporary limited-scope access tokens which survive for the length of the CI run. This ensures you never have to hardcode access tokens in your pipelines.
SEC11-BP08	Build a program that embeds security ownership in workload teams	Leaving all security design decisions to a security team.	JFrog Advanced Security	JFrog Advanced Security puts the power of secure code in the hands of the developers. From scanning their source code before they ship, to finding CVE information and remediation steps easily in their IDEs, developers can take charge of the security of their code. Contextual Analysis allows for developers to explore how their code is secure and CVEs affecting used dependencies are applicable or not.
		Not addressing security requirements early enough in the development process.	- JFrog CLI - JFrog IDE plugins - JFrog Curation	- The JFrog Platform has an ecosystem of API-first solutions to ensure developers are able to understand and implement security early on in the development process. - The JFrog CLI and the IDE plugins are excellent tools for developers to run local security scans, to infer scan results, and to ensure the code they ship is secure. - JFrog Curation enables organizations to define policies that ward off security issues at the time of ingestion of a new dependency or version. This ensures that even before development starts using that dependency, it passes muster.
		Not obtaining feedback from builders and security people on the operation of the program.	- JFrog Curation - JFrog Research - JFrog integration with GitHub Copilot	- JFrog Curation's Dry Run approach is a great starting point for security conversations. Security personnel can work with developers to monitor open source dependency usage and threat surfaces. The resultant Curation policies at the organization-level are more secure due to the constant feedback loop from both sides - developers who use these dependencies on a daily basis, and security teams responsible for ensuring only highly secure code is shipped. - The JFrog Research team is a resource for every customer as they act as a continuous threat monitor. The Research team updates the JFrog Xray Database daily with new CVEs, contextual analysis, code scanners and code samples to empower customers and inform them about vulnerabilities, resolution, and mitigation techniques. Whenever the JFrog Research team finds a vulnerability not yet disclosed to the market, it files the vulnerability in the Xray database with an internal ID. This protects our customers often weeks or months before the market finds out about the vulnerability. - The JFrog integration with GitHub Copilot enables customers to connect with the JFrog Platform like never before, asking security questions in natural language and getting tailored responses based on their organization's vulnerability risk tolerance.

Visit jfrog.com/cloud to learn more

Start a free trial of JFrog on AWS

