DevSecOps

A guide to developing and leveraging a culture of security

In collaboration with

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introduction

Organizational culture—shared values, mindsets, and behaviors that guide all employees—has everything to do with how successful an organization can be. Smaller teams within the business can adopt a specific culture that guides decisions on their focus area. A culture of security applies across a business from everyday best practices to developing modern applications in the cloud.

In this eBook, you'll discover how JFrog and AWS enable you to start creating a culture of security by combining your organization's own guiding principles and the DevOps philosophy of working. You'll explore how this approach impacts modernization and development strategies, and you'll learn how to build a pipeline of continuous integration and development that elevates every level of the business.

Why develop a culture of security?

A culture of security unites employees on a common path to business stability and modernization.

Organizations that are migrating to the cloud as a step toward modernizing must adopt an entirely new mindset around security and start to better leverage modern technologies and operational models, such as DevOps.

DevOps—bringing together formerly siloed development and operations teams—is a combination of cultural philosophies, practices, and tools that merges software development with information technology (IT) operations. DevOps enables companies to accelerate delivery of new application features and improved services to customers.

DevSecOps integrates security processes into the DevOps model. With DevSecOps, businesses can rapidly deliver secure and compliant application changes while running operations consistently with automation. This starts with developing operating *tenets* they can apply when shaping their vision for security as their business evolves.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How to develop tenets

When updating their security culture, successful organizations ensure that everyone understands the need for change and the path to reaching their common goal. Crucial steps in developing actionable tenets include:

- Working with all employees to identify the organization's core values that serve as the foundation for the tenets
- Establishing guidelines, expectations, and accountability for following the tenets—while empowering every team to follow them
- Garnering company-wide buy-in for the tenets and the new culture they support

At the highest level, each tenet should:

1. Be memorable.
2. Relay only a single idea.
3. Be specific to a program (e.g., security).
4. Guide, not proscribe.
5. Keep the business focused on the overall goal.

To guide their decisions and actions, businesses can apply the AWS-created tenets below. Organizations with an established culture of security with DevSecOps use the most common tenets to address:

Constant attacks
Build the understanding that the business is constantly under attack—both deliberately and accidentally—into every process.

Education
Prioritize security education for all employees. Stay abreast of developing threats, accept advice from security specialists, and seek to understand the organization's security policies and rules.

Hygiene
Evangelize company-wide that good security hygiene is part of doing things right. Do not share passwords or user accounts or expose personal information. Use secure coding practices.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Continuous improvement
After an error in protocol, take feedback to ensure it doesn't happen again.

Zero-defect approach
Do not accept any known vulnerabilities. Do not triage security defects and problems: fix every issue as soon as it arises. Scan code prior to moving it to production. Given that the vast majority of software contains segments of code obtained from open source, you need to confirm that all those code segments have no known vulnerabilities.

Reusable tools
Build and share security tools and processes—such as reusable logging and monitoring, enterprise-wide user provisioning, and standardized onboarding and offboarding processes for employees—across all IT systems.

Unified team
Ensure that all parts of the organization collaborate to strengthen security and enable resilient systems.

Testing
Rigorously test systems for vulnerabilities with automation—including failure scenarios and quality of response—both during development and production.

Threat modeling
Think as bad actors do to identify possible entries to attack, and then test to defend against them.

Peer reviews
Consider any possible defects and security vulnerabilities in the work and ensure peers always review the code.

Learn more about how tenets come into play on a cloud journey.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What does a culture of security look like?

In an established culture of security, an organization educates every employee in how to detect a potential threat, minimizes risk, and establishes a recovery plan. By acting proactively rather than reactively, the business is better positioned to protect themselves, their products and services, and their customers.

Below are just a few examples of what a culture of security looks like in action.

Employee-exposed passwords

Today
IT resets password(s), updates anti-virus software, and sends employee a link to reread the organization's security policy.

In a culture of security
Multi-factor authentication (MFA) is set by the organization; even though the password was exposed, the account is not compromised as a second factor is required to authenticate. The user will reset their own password and notify IT of the incident.

Software code integrity

Today
For the sake of speed, developers incorporate software code segments from multiple different public repositories into software package releases. This introduces risk—as the vulnerability and license compliance status may not have been checked—and jeopardizes the security of their binaries.

In a culture of security
The platform performs deep recursive scanning across the pipeline from your IDE to your production or edge devices. This enables you to verify security and compliance status of all the components and establishes software artifact integrity so organizations are confident in the application updates and releases it distributes.

Unauthorized internal access to data

Today
The organization defaults to granting employees full access to internal data.

In a culture of security
The organization establishes identity and access management practices (IAMs) that limit an employee's access to only need-to-know data.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Skunkwork cloud infrastructure projects

Today
The dev team cuts a ticket and gets a help desk request to provision a cloud instance for staging.

In a culture of security
The team uses a cloud formation template that includes security policies and governance, then provisions the cloud on that automated script.

Code typo or third-party code vulnerability

Today
After being informed of the software failure due to poorly tested code not catching a code typo or the incorporation of an unverified code segment from a third party, the business releases a patch that users must download from the website and manually install.

In a culture of security
The DevSecOps team undergoes threat detection and modeling during software development; if an error or vulnerability is determined, then minimize the time taken to fix vulnerabilities with enhanced vulnerability data detailing intuitive Step-by-Step developer remediation; then as a fix is required, the business automatically pushes it out to registered users.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Designing for security in the AWS Cloud

After identifying and evangelizing their tenets through the business, the next step is to align them with design principles that guide security in their cloud strategy. Below are design principles that can help strengthen workload security.

- Implement a strong identity foundation
- Enable traceability
- Apply security at all layers
- Automate security best practices
- Protect data in transit and at rest
- Restrict unauthorized access to data
- Prepare for security events

"The most popular misconception about moving to the cloud is that it's a set-and-forget proposition. Everything will run like clockwork, right? Instead of falling into this trap, [businesses] should be thinking about what happens on Day 2—the day after the last server has been decommissioned, and everything is fully running in the cloud—and what [their] cloud governance strategy will be."

Dr. James Bland
Global Technology Lead for DevOps,
Amazon Web Services (AWS)

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

An end-to-end pipeline of secure delivery and deployment

After an organization develops and rolls out their tenets and design principles, they're ready to set in motion their DevSecOps pipeline, which is critical to building a successful software factory that includes continuous:

- Integration (CI)
- Delivery and deployment (CD)
- Testing
- Logging and monitoring
- Auditing
- Governance
- Operations

Identifying vulnerabilities during the initial stages of the software development process can significantly help reduce the overall cost of developing application changes, but doing it in an automated fashion can accelerate the delivery of these changes as well.

Leveraging AWS and JFrog

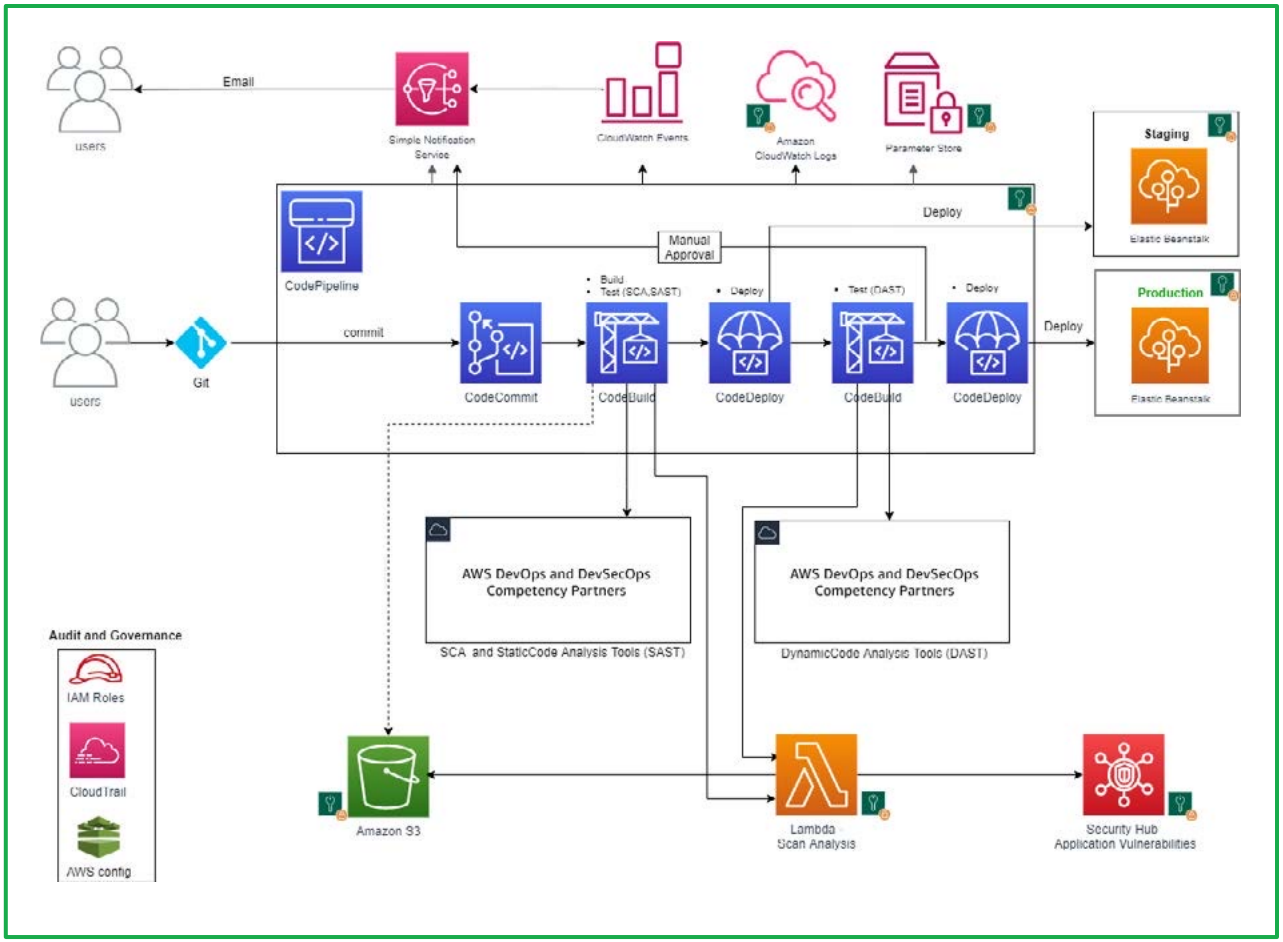
To identify security vulnerabilities at various stages, organizations can integrate various tools and services (both cloud and third-party) into their DevSecOps pipelines. The advantage of AWS native tools and partner integrations such as JFrog is the ability to template an organization's CI/CD pipeline as infrastructure and scale it in the cloud.

When hosted on AWS, JFrog can accelerate DevOps releases by using a more nimble and secure binary code and artifact management, packaging, and distribution environment. AWS CloudFormation templates further help with this by enabling enterprises to standardize deployment of JFrog DevOps environments, which speeds the time to implementation on AWS. By seamlessly supporting over 30 development environments, the JFrog Artifactory repository simplifies development operations and delivers a unified software engineering experience.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integrating various tools and aggregating the vulnerability and security findings from scratch can be a challenge. AWS and JFrog have the services and tools necessary to accelerate this objective and provide the ease and flexibility to build DevSecOps pipelines by integrating AWS and JFrog cloud-native and other third-party tools.

The AWS DevSecOps pipeline reference architecture illustrates these DevSecOps practices—including Software Composite Analysis (SCA), Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST)—and the aggregation of vulnerability findings into a single pane of glass.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migrate and modernize with confidence: The DevSecOps journey step by step

Start the DevSecOps journey with this step-by-step guide.

1: Undergo threat modeling

Define any threat vectors:

- What will move to the cloud in the next 18 months?
- How many points of entry are there?
- How does the business secure data in transit and data at rest?

Outcome:
An established point-in-time state of the state. It's important to note that the business will continually add variables during the transformation.

2: Upskill, enable, and empower all teams

Having an excellent security posture means having teams that are constantly on top of all threats across the infrastructure with a focus on continuing education. Security is a constantly moving target and a shared responsibility among all teams: developer, operations, security, and non-IT.

Outcome:
A detailed plan to upskill teams and shape the culture around collaboration to meet the organization's ever-changing security needs.

3: Implement a continuous security feedback loop across all stages of the delivery lifecycle

Establish and evangelize best practices around security coding standards, integrated security testing models for all pipelines, application security testing (AST), and vulnerability management.

Outcome:
Issue identification during code development and feedback loops, which helps accelerate remediation and reduce costs.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

4: Establish policies and governance

It's critical to ensure the business follows their policy and governance guardrails. Automate security policies to notify and remediate any violations or abnormalities.

Outcome:
Well-defined policy, governance, and automated remediation across the infrastructure and applications.

5: Gamify security and make it fun!

Consider implementing bug bounties for development, operations, and security teams. It's a fun way to drive education and collaboration and to incentivize a security mindset—and help meet education and upskilling goals.

Outcome:
An engaged, always-on security focus with an element of fun.

Shifting left is the right path

By shifting security compliance to the DevOps team as they compile code from multiple sources, the JFrog Xray scanning analysis tools have the ability to probe more than 30 package types stored in JFrog Artifactory and can see deep inside the binaries to check for code segment vulnerabilities and verify license compliance. The platform gives developers the opportunity to reconfirm the code artifacts with tools that are easy to use for streamlining the entire process.

> **"Some people didn't even realize that it's now running on AWS, and not running on-prem anymore."**
>
> **Caio Trevisan**
> Technology Director – Data and DevOps Platforms, Bendigo and Adelaide Bank

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CASE STUDY

Bendigo and Adelaide Bank

Situation

With roots going back to the 1850s, Bendigo and Adelaide Bank is one of Australia's biggest banks, with A$74b ($52b US) in deposits, and more than 4,800 employees helping over 2 million customers achieve their financial goals.

Even with the bank's long history, its DevOps Service team insists, "We work as a startup company." The team maintains a very mature DevOps pipeline and has committed to a four-year plan to transform the bank "to get at least 80% of our applications in the cloud" while maintaining compliance with rigorous banking regulations.

Challenges

To modernize, the DevOps Service team needed to migrate their JFrog Platform on-premises high-availability installation to self-managed Kubernetes clusters in a cloud service provider, with enhanced developer productivity and continued compliance while remaining compliant with regulations.

Solution

The bank's DevOps Services team chose AWS EKS to host the Artifactory repository and Xray security. Using the Helm charts made it "really easy to set up an instance of Artifactory and Xray in a Kubernetes environment with one command and a few values," said Caio Trevisan, Technology Director – Data and DevOps Platforms, Bendigo and Adelaide Bank. Within a single hour, they can spin up a test or production cloud environment for the JFrog Platform.

Results

Federating repositories between the on-prem and AWS installations, the bank was able to duplicate 1TB of accumulated packages, artifacts, and binaries data to the new cloud environment in AWS. This bidirectional mirroring capability in Artifactory, along with user token synchronization through JFrog Access Federation, enabled developer teams to seamlessly transition their repository use to the AWS environment with zero disruptions to daily operation.

Migration of all data and teams to the new AWS environment—including initial tests as well as rigorous internal compliance and governance procedures—was completed within 6 months.

Now fully in the cloud, production has accelerated, improving build times 30-40% and greatly reducing costs.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Build modern, build secure

With an established culture of security, organizations eliminate reactive processes that bog down a development pipeline with security vulnerabilities, patching, and rework. With a fully automated, DevSecOps-compliant pipeline that matches their tenets and design principles, businesses of any size can create modern applications that both fuel and drive today's digital world.

In addition to compliance and security, developers must also deal with limited bandwidth and network lag when seeking to quickly distribute critical software updates. JFrog helps control DevOps environments with out-of-the-box native and partner integrations that enable developers at enterprises to distribute trusted software releases from code to production.

Visit JFrog on aws marketplace

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.