Shifting Security Left and Protecting Right

Get Started
Introduction

Those familiar with DevOps know that it is the implementation of automation within the development lifecycle to help unite both development and IT operations teams. However, some may not realize that adding security into the mix is the only way to ensure that security is embedded into the entire application lifecycle. With a DevSecOps approach, users can automate security from the build phase all the way through deployment and optimization.

This culture of automation allows DevSecOps teams to not only build more efficient processes, but also to break down silos between development, IT operations and security. With this increased collaboration, security tools and procedures become more effective, ultimately enhancing the security posture of the organization. Furthermore, synergistic relationships between security and development teams can help them complete the software development lifecycle (SDLC) five business days faster than those with negative relationships.¹

In addition, DevSecOps is vital when it comes to building cloud native applications as these applications are inherently more scalable than their legacy counterparts due to a higher resistance to human error. Traditional security practices—especially those involving human intervention—often don’t address the complexities of cloud native applications. As a result, cloud-first companies quickly adopt the use of containers running as microservices to reap significant value and scale company growth.

The ephemeral and immutable nature of containers, however, brings both challenges and advantages for security. This highlights DevSecOps practices as a valued solution to the tedium associated with compliance and auditing, and suggests that the main catalyst for success is a cohesive relationship between development, security, and IT operations.

This ebook will provide a better understanding of the challenges that affect today’s market. We’ll explore how tooling leads to success, the value of automation to the continuous integration and continuous delivery (CI/CD) pipeline, and how VMware’s full security stack differentiates itself among competing container security solutions that tend to focus on narrow use cases.

By 2025, more than 85 percent of global organizations will be running containerized applications in production.²

² Gartner, Inc. “Best Practices for Running Containers and Kubernetes in Production.” Arun Chandrasekaran. August 4, 2020. GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved.
Challenges in deploying Kubernetes

As the Kubernetes market continues to mature, there are still a number of challenges that the industry must overcome on its way to growth and broader customer adoption (Figure 1). According to VMware’s State of Kubernetes 2021 report, the top three security concerns are (Figure 2):

1. Applying policy consistently across clusters and teams
2. Securing container images in a CI/CD pipeline
3. Securing containerized workloads at runtime

While policy consistency stood out as the top concern from the report, it is important to note that policy standardization needs attention from development through production.
Securing container images in a CI/CD pipeline and securing containerized workloads at runtime, as the report’s second and third concern (Figure 2), are directly related to Kubernetes, modern development practices, and the needs of developers and DevOps teams. These concerns have an outsized effect on a company’s overall security posture as it pushes containerized applications to production. Securing containerized workloads at runtime was the number two concern for IT operators (22 percent) and the number one concern for both infrastructure staff (23 percent) and architects (23 percent).  

In this ecosystem, development teams are learning that they need more than just DevOps to execute ideas to production. They are also discovering the need to embed security in the CI/CD pipeline from an early stage of code to the runtime state for efficacy.
Recognized benefits

While containers and cloud native applications come with their fair share of challenges, there are obvious benefits that cannot be ignored. Some of these benefits include:

- **Isolation** – By using microservices architecture, organizations can innovate more, expedite their route to market, and use less resources in the process. This benefits security teams because containerized applications can easily be built within isolated environments. Sensitive information within that app is restricted to that unique instance.

- **Elasticity** – Organizations can scale up or scale down depending on their specific needs at that time.

- **Flexibility** – New containers can easily be created, stopped or patched. If a container has critical vulnerabilities, it is easier to remediate that vulnerability than it would be with legacy infrastructure.

- **Efficiency** – Utilizing automation, these fixes to container vulnerabilities can be remediated faster than ever before.

Another key benefit of utilizing containers for cloud native applications is that everything can be managed in an automated and orchestrated manner. With this type of structure in place, development teams can incorporate security from an early stage of code through the build of container image phase, to running the container on Kubernetes as pods. In other words, every system can be part of the automation process to achieve the DevSecOps goals of shipping code swiftly, reliably and securely to production.
How to apply security that spans the entire application lifecycle?

Comprehensive security that spans the entire application lifecycle at the speed of DevOps requires tightly orchestrated and automated solutions that should:

- Support business continuity
- Emphasize transparency and open communication between teams
- Remove complexity
- Provide streamlined access management and onboarding
- Ensure compliance through security policy automation

By 2025, nearly two-thirds of enterprises will be prolific software producers with code deployed daily, more than 90 percent of apps will be cloud native, and there will be 1.6 times more developers than today.  

Shifting security left and protecting right

Here’s how VMware approaches DevSecOps to provide development teams with the full security stack. To understand how security should be applied throughout the application lifecycle, let’s look at the various security requirements for each phase.

CI/CD Pipeline

Code + Build + Store + Prep + Deploy + Run + Continuous Scanning

**Code**
The first step is to code securely. To do so, developers need building blocks that are both secured and trusted, and should work with their security counterpart to patch and test reported vulnerabilities. VMware Tanzu® development tools provide regular updates for these born-secure building blocks to better protect your data and apps from day one.

**Business outcomes**
- Secure software development
- Streamlined integration
- Comprehensive authentication
Shifting security left and protecting right

Here’s how VMware approaches DevSecOps to provide development teams with the full security stack. To understand how security should be applied throughout the application lifecycle, let’s look at the various security requirements for each phase.

CI/CD Pipeline

<table>
<thead>
<tr>
<th>Code</th>
<th>Build</th>
<th>Store</th>
<th>Prep</th>
<th>Deploy</th>
<th>Run</th>
</tr>
</thead>
</table>

Build

When developers create a container image that contains a core OS, application dependencies, application runtime and code, they need to build that image securely and based around the latest dependencies according to the development language used. While Tanzu Build Service™ manages this securely, there can still be various vulnerabilities in the time between when they pull the opinionated dependencies to when they build the container image. Consequently, before the container image is pushed into the container registry, it is necessary to scan that image for vulnerabilities.

Business outcomes

- Automated container builds
- Centralized image build system
- Customized compliance policies
Shifting security left and protecting right

Here’s how VMware approaches DevSecOps to provide development teams with the full security stack. To understand how security should be applied throughout the application lifecycle, let’s look at the various security requirements for each phase.

**CI/CD Pipeline**

<table>
<thead>
<tr>
<th>Code</th>
<th>Build</th>
<th>Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prep</td>
<td>Deploy</td>
<td>Run</td>
</tr>
</tbody>
</table>

**Store**

Not all images used in a developer’s environment will be built by the in-house development team, as most of the surrounding back-end services for the app—such as message queuing and database—will be based on off-the-shelf products with accompanying images. Because of this, it’s necessary for developers to scan these off-the-shelf images stored locally on the local image registry. Fortunately, with VMware, developers can pull opinionated dependencies securely with VMware Tanzu and scan for vulnerabilities in the container image with VMware Carbon Black Cloud Container™.

**Business outcomes**

- Container image scanning
- Vulnerability management
- Container visibility
- Risk prioritization
Shifting security left and protecting right

Here’s how VMware approaches DevSecOps to provide development teams with the full security stack. To understand how security should be applied throughout the application lifecycle, let’s look at the various security requirements for each phase.

CI/CD Pipeline

<table>
<thead>
<tr>
<th>Code</th>
<th>Build</th>
<th>Store</th>
<th>Prep</th>
<th>Deploy</th>
<th>Run</th>
</tr>
</thead>
</table>

| Continuous Scanning |

Prep

As we move into the continuous deployment portion of the pipeline, the manifest (aka the configuration file) is created at the Prep stage to declare to Kubernetes how the workload should run. Configuration errors in the Kubernetes manifest, however, can unintentionally allow the container to run with escalation privileges or host access, exposing it to threats. Therefore, developers need to validate that the manifest file complies with security policies based on Kubernetes security best practices and CIS benchmarks. If the file is configured according to the security policy, developers can proceed to deploy and provision the container in PROD.

Business outcomes

- Secure application building blocks
- Verify images
- Audit container images
Here's how VMware approaches DevSecOps to provide development teams with the full security stack. To understand how security should be applied throughout the application lifecycle, let’s look at the various security requirements for each phase.

**CI/CD Pipeline**

- **Code**
- **Build**
- **Store**
- **Prep**
- **Deploy**
- **Run**

**Deploy**

After the configuration file is prepped and ready to be deployed, container images need to be scanned in the Deploy phase. Using the scan results in tandem with an admission controller allows the rejection of the deployment of an application if the container images suffer from known vulnerabilities or are not compliant with the established security policy.

Misconfigurations pose the highest risk to workload environments. To mitigate these risks, developers must enforce Kubernetes security best practices and CIS benchmarks that define what standard configurations should be modified to promote a stronger security posture. To enforce these security standards, developers can segment their environment by building scopes, such as DEV, PROD and Application, aggregating different clusters or namespaces to ensure similar assets are uniformly protected. In other words, these scopes give developers the flexibility to control the level of restriction at each stage.

**Business outcomes**

- Security of the full container lifecycle
- Speed of delivery
- Visibility into workload health
Here’s how VMware approaches DevSecOps to provide development teams with the full security stack. To understand how security should be applied throughout the application lifecycle, let’s look at the various security requirements for each phase.

CI/CD Pipeline

Run
Runtime security addresses all aspects of running a workload in production. Developers use open source to build containers and cloud native applications because of the speed and flexibility it offers. As a result, open source tools are widely adopted and commonly used by developers building cloud native applications. Open source tools are great for agility, but SecOps teams need to ensure that these applications still meet the high level of security needed after they are in production. Just like how images and third-party registries need to be scanned for vulnerabilities during the Build phase, these images also need to be scanned during the Run phase.

Business outcomes
• Secure your apps in runtime across clouds
• Identify malicious network activity
• Secure egress connections
Extending security to the underlying technology stack

Each component of the CI/CD pipeline runs on Kubernetes. So, what’s ensuring the secured workload and pipeline? Well, developers need to secure the infrastructure—the Kubernetes orchestration layer, the Linux/Windows OS, and the hypervisor/public cloud—that the workload and pipeline run on. VMware Carbon Black Cloud™ delivers comprehensive built-in security for a holistic DevSecOps, from workloads and pipeline to the infrastructure they run on.

CI/CD Pipeline

- Code
- Build
- Store
- Prep
- Deploy
- Run

Continuous Scanning

Infrastructure

- Kubernetes / Container Runtime
- Linux / Windows OS
- Hypervisor / Public Cloud
Summary

VMware Carbon Black Cloud Container enables enterprise-grade container security at the speed of DevOps by providing continuous visibility, security and compliance for containerized applications from build to runtime, in any on-premises or public cloud environment. This solution provides security teams with visibility and the ability to enforce compliance while integrating seamlessly into existing DevOps processes to avoid adding operational complexity. The combined power of the VMware Tanzu portfolio and VMware Carbon Black Cloud enable true application modernization with security from code to production. With VMware, organizations can reduce risk, maintain compliance, and simplify security for Kubernetes environments at scale.