radware

FROM MONOLITHIC SCRIPTING TO DEVOPS-GRADE AUTOMATION

4 REASONS WHY APPLICATION DELIVERY IS CRITICAL TO DEVOPS

The move to continuous integration/continuous deployment (CI/CD) and agile development methodologies means application development now requires network IT operation teams (NetOps) and security operation teams (SecOps) to work together, thereby putting increased stress on developers and engineers.

Add heterogenous computing environments to the mix, and it's no wonder that the role of application delivery and security within DevOps has never been greater. Selecting an application delivery solution that maximizes automation and integrates into your existing DevOps environment is critical.

Here are four reasons why application delivery is critical to supporting DevOps:

1. PRODUCTION-READY AUTOMATION

CI/CD and agile development methodologies require IT experts from different domains (i.e., application, storage, network security, application delivery) to deploy new application services and their associated policies. Traditionally, they've had to invest in long, tedious provisioning and programming tasks that are often repetitive and require specific domain expertise to execute.

In a DevOps-driven world, this is no longer possible. IT requires workflow automation so that even non-technical experts can develop and craft application delivery and security workflows. According to Radware research, over half (53%) of organizations don't integrate application protection into CI/CD processes.

This is why IT operations requires an evolution, transitioning from simple scripts to standard-based, production-ready modules. The goal? Enable anybody from the aforementioned teams to deploy and manage services regardless of the domain expertise he or she possesses.

2. AUTOMATED ADC AND SECURITY

The marketplace for application delivery and security solutions are as diverse as their capabilities. Selecting one that integrates with your application provisioning and deployment solution, such as Ansible, is critical to enabling end-to-end automation.

Traditionally, deploying a new application delivery service has meant writing and managing a monolithic, gargantuan script to automate the workflow. Each application deployment requires a different workflow, and thus, a different script. DevOps requires these monolithic codes to be broken into discrete operations to speed time to development and deployment.

Underlying application delivery and security services that support these applications are no exception to this rule. An enterprise-grade ADC should provide pre-defined/production ready modules and integration with DevOp orchestration tools to allow non-technical users to spin up and deploy application and security services.

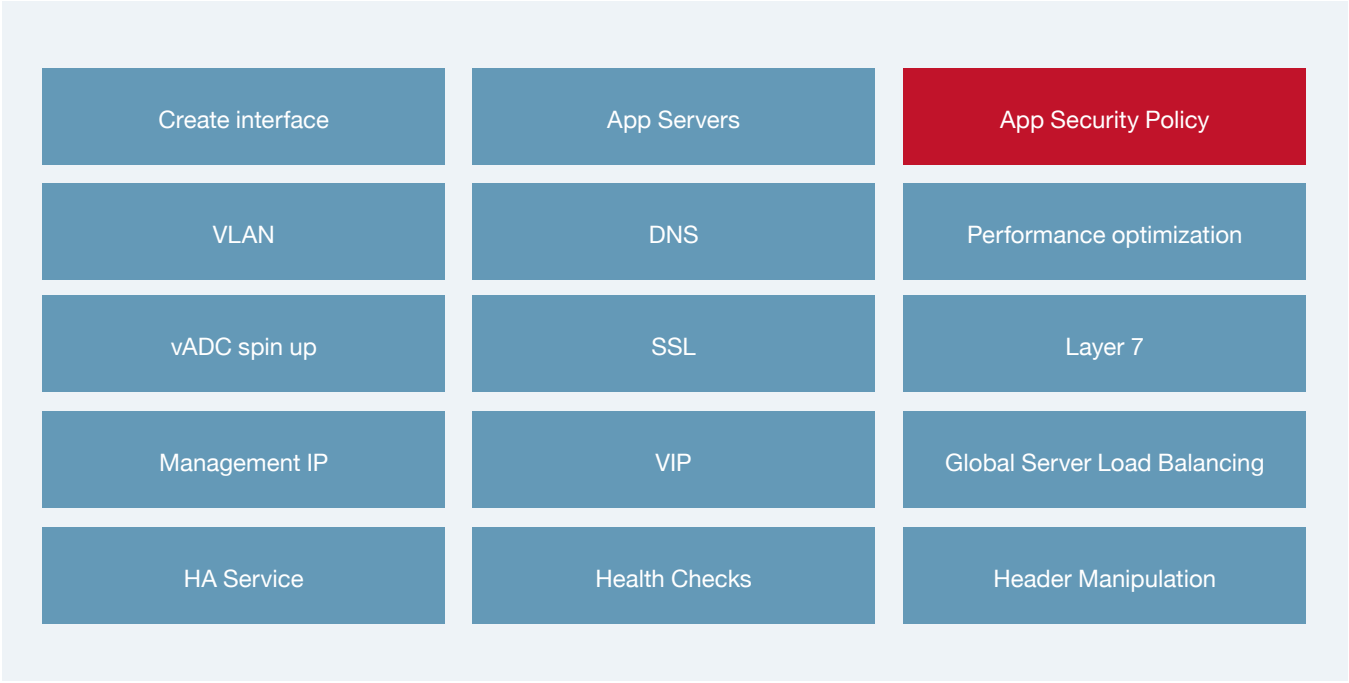

Figure 1: Examples of various operations comprising the application delivery service

Traditionally, if an error occurs, a review and QA of the code would be required. In a module-based world, errors become more transparent and segmented. For example, if only the application security policy is incorrect (see Figure 1), the problem can be identified, fixed and ONLY the security policy rerun, versus the entire script-based workflow.

3. REUSE AND REPURPOSE

Production-ready modules provide the ability to reuse and repurpose. If a particular module, such as the aforementioned application security module, requires update, SecOps can make that update, create a subversion if necessary, and apply it to other application workflows. Changes are isolated to the specific module and don't impact other aspects of the workflow.

Lastly, an application owner, versus an ADC expert, can make these changes to the application security module, or other modules. For example, while application servers and security policies can reside with the application owner, VLAN operations might be the responsibility of a network administrator. Now, a single person can manage this normally disparate responsibilities, thereby saving time to market.

This modularized approach also mean switching from staging to production, and back, seamlessly. Changes to workflow parameters made within a staging environment can be pushed to production following staging. These capabilities support CI/CD processes by minimizing human errors and reducing downtime.

4. ENTERPRISE-GRADE ADC CAPABILITIES

Lastly, any application delivery solution must combine ease of use with advanced application delivery capabilities, such as optimizing Layer 7 load balancing deploying and managing application security policies. This further enhances the capabilities of any DevOps orchestration tool.

Learn More About What Multi-Cloud Application Delivery Should Encompass

About Radware

Radware® (NASDAQ: RDWR) is a global leader of cybersecurity and application delivery solutions for physical, cloud and software-defined data centers. Its award-winning solutions portfolio secures the digital experience by providing infrastructure, application and corporate IT protection and availability services to enterprises globally. Radware's solutions empower more than 12,500 enterprise and carrier customers worldwide to adapt quickly to market challenges, maintain business continuity and achieve maximum productivity while keeping costs down. For more information, please visit www.radware.com.

Radware encourages you to join our community and follow us on: Radware Blog, LinkedIn, Facebook, Twitter, SlideShare, YouTube, Radware Connect app for iPhone® and our security center DDoSWarriors.com that provides a comprehensive analysis of DDoS attack tools, trends and threats.