SOLUTION BRIEF

Fortinet Container Security

Executive Summary

The adoption of cloud-native technologies to deliver new products and services has enabled organizations to rapidly transform key areas of their business. These technologies include the use of containers in microservices architectures, which have streamlined the way applications are built, tested, deployed, and redeployed.

Container technology enables developers with new ways to design and develop applications in a more modular and resilient manner. This is accomplished by separating different logical functions of applications into separate containers, which makes it easier to manage and scale. Further, by breaking applications out into different logical functions, namely, microservices, container technology enables enhanced portability of microservices and applications across different public and private cloud environments.

Despite its advantages, traditional security tools lack the in-depth capabilities to secure container workloads. Container security requires visibility and protection during all stages of a container life cycle.

The Fortinet container security strategy offers multiple solutions that address the entire life cycle of container-based applications, providing comprehensive protection against threats associated with the different threat vectors to which container-based applications are exposed.

The Benefits of Container Technology

Many organizations have adopted the use of containers to accelerate application deployments. It is estimated that by 2022, over 75% of global organizations will be running containerized applications in production, and up to 15% of enterprise applications will run in a container environment by 2024, up from less than 5% in 2020.[1]

The Key Attributes of Container Security

- Container-aware security
- Container-enabled security
- Container-integrated security
- Container registry security
- Shift-left security: Embeds security into software development life cycle

Teams need to be able to rapidly develop modular applications in containers—on-premises or in the cloud. Security should be as portable and consistent as the applications themselves.

Some of the primary drivers contributing to the accelerated adoption of containers include:

- **Simplicity and Portability:** Containers are lightweight, self-contained software application bundles, independent from the host operating system, which allows them to be run consistently across any platform or cloud.
- **Agility:** Containers allow for faster application development and deployment cycles.
- **Increased Efficiency:** Unlike virtual machines (VMs), containers are simpler, lighter software stacks that require less startup time, allowing for higher resource efficiencies, resulting in lower costs.

Containers break up applications into smaller functional objects with autonomous functionality.

Each object is maintained separately (including different versions and bug fixes) and scaled separately, as different pieces may require different levels of performance. These objects are typically referred to as services—or increasingly, microservices.

In server virtualization environments, VMs have a set of metadata attributes that are visible at the hypervisor or virtual infrastructure levels—commonly referred to as "tags"—to manage repositories of images. Similarly, containers also have associated metadata attributes that are typically called "labels."

Currently, the most common container-format standard is Docker, which has both an open-source and a commercial implementation. To build an application using container technologies, an organization requires multiple, interdependent services interacting with each other—typically called a POD. Building applications requires multiple containers, and while they may or may not be grouped into PODs, they all must be interconnected for the application to properly operate.

Containers are connected by a service/application composition, which is normally performed as part of the orchestration process of bringing up a containerized application. This orchestration process dynamically assigns addresses for the different services and offers a service/name resolution capability for different services used to resolve each other. This is where Kubernetes enters the picture.

Kubernetes has emerged to become the most common container orchestration system, with the ability and instrumentation necessary to describe application compositions, service dependencies, service-scale requirements, service-availability requirements, and more. Kubernetes also provides the tools needed to independently manage the life cycle, scalability, availability, and performance of different services without interrupting the availability of an entire application.

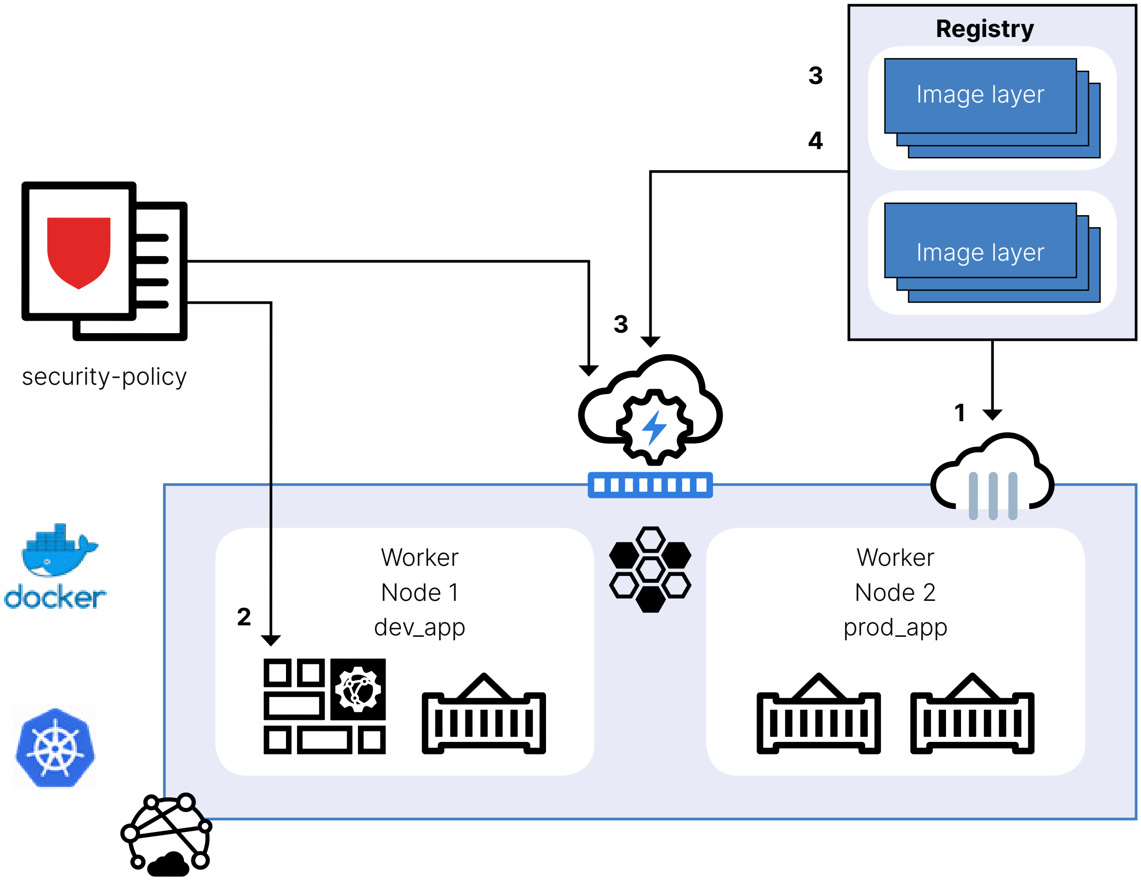

Figure 1: Protecting application containers throughout the application life cycle.

Security for the Entire Application Container Life Cycle With Fortinet

Containers have quickly grown in popularity since they changed how businesses deploy and use applications. And their usage will continue to rise, with 86% of IT leaders prioritizing them for more applications.[2] But security is also one of the biggest areas of concern with adoption, as evidenced by a recent survey in which 94% of organizations reported that they have experienced at least one security incident in their Kubernetes and container environments in the past 12 months.[3]

Securing containerized applications is challenging due to their dynamic nature. Traditional security tools lack the in-depth capabilities to secure container workloads.

The Fortinet container security solution solves these requirements in the following ways:

1. **Container-aware security:** The **FortiGate Next-Generation Firewall (NGFW)** effectively connects to the container management layer and learns the labels of different containers. Security policies are label-aware and can use these labels to describe objects in security policies. This solution is primarily relevant for securing traffic in and out of the container infrastructure—namely, north-south security. FortiGate NGFWs offer Fabric Connectors that interface with major container orchestration systems to leverage metadata as security policy objects, including native Kubernetes, AWS EKS, GCP GKE, Azure AKS, and OCI OKE. When traffic leaves the boundaries of a containerized environment, it crosses a FortiGate NGFW that enforces the policy based on the container role. FortiGate also scans ingress and egress container traffic for vulnerabilities and file-based threats using an intrusion prevention system and advanced malware protection via FortiSandbox integrations.

2. **Container-enabled security:** Organizations can also leverage the **FortiWeb Web Application Firewall** as a container image that can be bundled within an application chain. Since it is very typical to create microservices for web service-based applications, the ability to couple web application and application programming interface (API) protection with microservice-based applications offers significant benefits to organizations building these applications. Developers can roll out security controls alongside their application development life cycle and port application security along with the other application services throughout the application life cycle to different environments. FortiWeb is currently offered as a native Docker container as well as an AWS EKS marketplace offering. FortiWeb also integrates with FortiSandbox, adding an additional layer of zero-day threat protection for incoming and outgoing traffic.

3. **Container-integrated security: FortiCWP Container Guardian** embeds security through the software development life cycle by preventing the propagation of known vulnerabilities into the build process. Integrations with developer toolchains automate and build the continuous integration/continuous deployment (CI/CD) pipeline. FortiCWP Container Guardian drives security governance using CIS Kubernetes benchmark policies to prevent unsafe workloads from getting deployed and continues to monitor the risk posture to identify evolving vulnerabilities.

   When containers are running, much of the container-based internal application traffic occurs within the container host and is not visible in the network infrastructure. To ensure that each traffic flow is inspected, traffic flow between services must be modified within the application or a mechanism of service insertion. FortiCWP Container Guardian provides insights into the components in each Kubernetes cluster, as well as the traffic connections between containers.

4. **Container registry security:** Container images are typically stored in public repositories known as registries, and there are few restrictions on the publication of new container images to them. This often leads to container images that are intentionally or mistakenly seeded with malicious code that is easily "pulled" from the registry by application developers. This situation introduces unnecessary risk to the application development process. **FortiCWP Container Guardian** scans for vulnerabilities in container images before they get integrated into the build cycle. **FortiSandbox** offers the APIs and integration capabilities to specifically address the needs of container-based application developers. This helps them mitigate the potential risks introduced through their agile development methodologies.

Enabling a Secure, Holistic Container Strategy

Container technologies are rapidly becoming more popular as an application infrastructure and development technology. However, the risks they introduce are inadequately addressed by using traditional security tools. The ability to source a comprehensive container security solution that is compatible with a broad range of container orchestration systems is essential for organizations deploying any application on any container infrastructure in both public or private cloud environments.

Container security solutions from Fortinet fully address the expanded attack surface, enabling security to be integrated in the container application life cycle and allowing organizations to deliver more secure applications.

[1] Michael Warrilow, "Forecast Analysis: Container Management (Software and Services), Worldwide," Gartner, May 29, 2020.

[2] "Cloud Container Adoption In The Enterprise," Forrester, June 2020.

[3] "State of Kubernetes Security Report," Red Hat, 2021.

November 2, 2021 7:03 AM
357646-A-0-EN