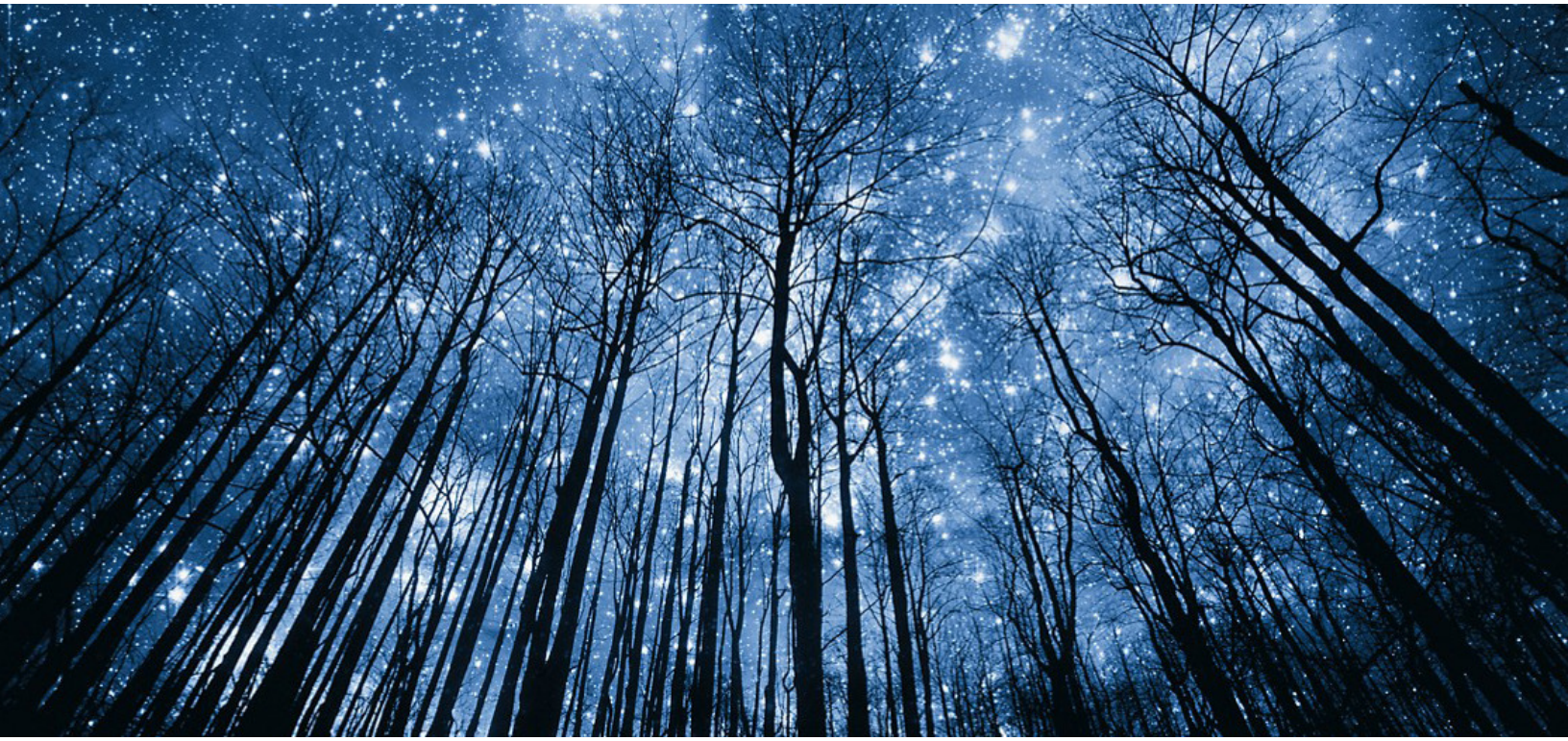
DELL Technologies

CONTAINER SECURITY FOR CONTINUOUS INTEGRATED PROTECTION



Divya R
Specialist 2, Inside Product
Dell Technologies
Divya.r1@dell.com

Anirudh Sandur
Specialist 2, Inside Product
Dell Technologies
Anirudh.sandur@dell.com

DELL Technologies
Proven Professional

Knowledge Sharing Article

The Dell Technologies Proven Professional Certification program validates a wide range of skills and competencies across multiple technologies and products.

From Associate, entry-level courses to Expert-level, experience-based exams, all professionals in or looking to begin a career in IT benefit from industry-leading training and certification paths from one of the world's most trusted technology partners.

Proven Professional certifications include:

- Cloud
- Converged/Hyperconverged Infrastructure
- Data Protection
- Data Science
- Networking
- Security
- Servers
- Storage
- Enterprise Architect

Courses are offered to meet different learning styles and schedules, including self-paced On Demand, remote-based Virtual Instructor-Led and in-person Classrooms.

Whether you are an experienced IT professional or just getting started, Dell Technologies Proven Professional certifications are designed to clearly signal proficiency to colleagues and employers.

Learn more at www.dell.com/certification

Abstract

This Knowledge Sharing article serves as a keystone document for anyone seeking to gain an overview of container architecture and provides a starting point to appropriately investigate the steps or best practices to mitigate risk and vulnerabilities associated with transitioning into a container architecture. This article will also explore the benefits of container security and a few use cases of VMware Carbon Black to illustrate the practices to be considered to automate DevOps with Full Lifecycle Container Security based on the environment.

Table of Contents

Introduction

Today, almost all applications are containerized. They provide Wi-Fi in coffee shops, process your purchase at the stores, enable online banking service at your fingertips. You can now transfer money

through applications from wherever you are, place delivery orders instead of going to a store and purchasing it, consult your doctor through mobile applications and upload the medicine receipt and book delivery online – all of these are possible only because they are running on containers.

With the real workloads in the environment come real situations. Your business is your data and running your critical applications in containers has elevated its importance, compelling leaders to make business decisions. If you are evaluating a cloud provider and your security lead says, “Cloud 1 provider offers managed images; for Cloud 2 provider, we’d be on our own,” how important is this kind of conversation to you? Should it influence your decision?

The goal of this article is to teach you the fundamentals of container and its security and prepare you with the required basic knowledge that would help you to keep your business up and running safely.

Containers

2.1 Introduction to Containers

The word ‘container’ may sound familiar and is, in fact, a highly trending word in IT.

Generally, container means something used to cover a multitude of things. A container is a way of packaging a given application’s code and dependencies so that the application will run easily in any computing environment. This solves the common problem of portability – or, more precisely, the lack thereof. Applications are built and tested using specific language, runtime, package, and library versions. When Developer A hands off work to Developer B, who merges it for testing and into production, inconsistencies between these environments can cause the application to break. (Operating system versions, for example, can be hard to keep coordinated between development and production; OS and application upgrades risk accidentally pushing incompatible changes.) Also, if you want to run multiple applications on the same host, these applications may require incompatible OS versions.

Containers solve the portability problem by isolating the application and its dependencies so they can be moved seamlessly between machines. A process running in a container lives isolated from the underlying environment. You control what it can see and what resources it can access. This helps you use resources more efficiently and not worry about the underlying infrastructure. However, while a container can be considered a boundary, it is a boundary with limitations. Just like VMs (Virtual Machines), containers can still be compromised through various attacks, or left vulnerable through misconfigurations or unpatched components. In “Running Kubernetes Securely,” we will talk more about the threats caused by common mistakes, but a compromised or misconfigured container can lead to unauthorized access to your workloads and your computer resources, and even the potential to recreate your application (and its data) somewhere else.

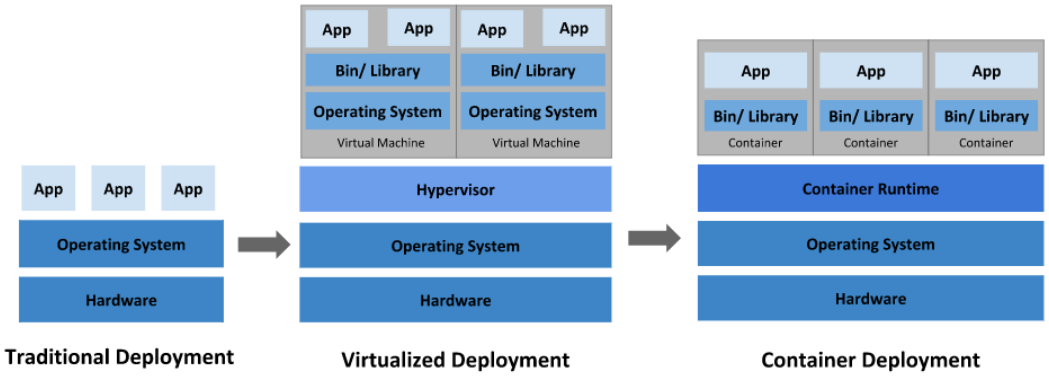


Figure 1: Timeline of deployment models
(Source: kubernetes.io/docs)

Container Security

Introduction to container security

As organizations transition to microservices and container architectures, the job of security teams become challenging because they need to develop container security solutions that facilitate these kinds of architectures. Container security must be continuous and integrated within the architecture.

Container Security is the process of using various security tools and different policies to protect the applications running on the containers and performance including infrastructure, software supply chain, system libraries, and runtime against cyber security threats.

Container Security mitigates risks throughout the entire environment, from the applications they support to the infrastructure they rely on. When implementing container network security solutions make sure they are integrated with the underlying container orchestration for context awareness of the application.

3.2 Importance of Container Security and its benefits

3.2.1 Why is Container Security important?

While containers offer few of the built-in security advantages such as application isolation, etc. If the organization is failing to recognize and plan certain security measures for the containers it would increase the security risks for organizations and would be prey for the cyber-attack.

There is significant increase in number of organizations adopting and transforming to container architecture. Containers have been adopted in production environments and has made them a primary target for cyber-attackers. Any one vulnerable or compromised container is more than enough to potentially become a point of entry for the attackers to access an organization's broader network. In the modern technology world where data is everything, these threats underscore the importance of container security. Traditional network security solutions offer no protection against these modern malicious attacks.

3.2.2 Benefits of Container Security

Container security is trending in the IT market and there is significant use of it by most of the organizations. Also, various key players of the IT world are acknowledging the importance of investing in app container security across their platforms.

Container security is concerned with all aspects of securing a containerized app and its underlying infrastructure. This produces a couple of benefits to strengthen the security of production as well:

- It can become a multiplier for improving IT security overall in the environment.
- Requiring continuous security monitoring across development, test, and production environments, also known as DevSecOps, can improve overall security in the environment. For example, by introducing automated scanning in Continuous Integration pipeline.

3.3 Best practices to consider while securing a container

3.3.1 How to secure a container?

The National Institute of Standards and Technology (NIST) published its Application Container Security Guide, which provides several basic approaches to doing so. Here are three key considerations from NIST's report:

Use container-specific host operating systems: NIST recommends using container-specific host OS, which are built-in with reduced features, to mitigate risk and attacks.

Segment containers by purpose and risk profile. Container platforms do a fair job of isolating the containers (from the built-in OS). However, greater defense can be achieved by grouping the containers based on their "purpose, sensitivity, and threat posture" and deploying them on separate host OS so that they have the isolation feature based on their requirement.

Use container-specific vulnerability management and runtime security tools. Traditional vulnerability scanning and management tools have certain drawbacks when it comes to containers. They are not designed to work specifically on the containers, which can lead to inaccurate reporting and a potential spot for security breach of images. Similarly, ensuring security at runtime is a key facet of container deployments and operations. Traditional, perimeter-oriented tools were not built with containers as a focus and cannot properly protect the environment.

It is recommended to opt for hardware-based built-in modules, such as the Trusted Platform Module (TPM), for another layer of security and it is suitable for containers and cloud-native application development.

3.3.2 Container Security essentials?

There are three important aspects of container security:

- Configuration
- Automation
- Container security solutions

Configuration: Many container, orchestration and cloud platforms offer security capabilities and controls. However, it must be correctly configured and should not be left to default settings. This configuration includes critical settings such as access, isolation, and networking.

Automation: Because of the architecture of containerized applications and their underlying infrastructure, security needs vulnerability check and anomaly detection which is a challenging task when done manually. Therefore, automation is key for many container security features and tools, in the same way that container orchestration also helps automate many operational tasks involved in running containers.

Container security solutions: Certain security teams integrate certain new security tools and support to their production that are built specifically for containerized environments. All those security tools are focused on various aspects of the cloud-native ecosystem such as CI tools, container runtime security and sometimes end-to-end environment also.

3.3 Common Container Security mistakes to avoid

There are several common mistakes when it comes to securing containers and its environment.

- **Forgetting basic security hygiene**. Containers are a relatively recent technology and new security approaches are required. But that does not mean abandoning certain security fundamentals. For example, keeping your system software's patched and updated, whether those are operating systems or container runtimes or other software.

- **Failure to configure and harden your tools and environments**. Many platforms come with significant security capabilities with built-in container and orchestration tools. However, to leverage all the benefits, configuration must be done accurately based on the environments.

- **Ignoring to monitor, log, and test.** When containers run in production, many of them neglect an applications health and do not monitor which leads to failure of noticing high impact alerts or events. Ensuring that the environment has the proper monitoring, logging, and testing tools and dummy environments will minimize unknown vulnerabilities and other drawbacks.

- **Not securing all phases of the CI/CD pipeline**. Another potential threat in container security strategy is ignoring certain elements of the software continuous delivery and integrated pipeline.

Prioritizing security as early as possible in the supply chain is beneficial and helps in numerous ways.

VMware Carbon Black Container

A security product from VMware, Carbon Black Container (CBC) enables continuous visibility, security, and compliance for the full lifecycle of containers and Kubernetes applications from development to production. Using CBC in the environment can automate DevSecOps with Full Lifecycle Container Security that bridges the developer and enhances security hygiene for DevOps.

Benefits of VMware Carbon Black Container:

- Complete Visibility into Container Security Posture
- Automate and Customize Compliance Policy to enforce secure configurations.
- Scan Container Images for blind spots/drawbacks from Development to Production
- Governance and Enforcement of policies from build to deployment to detect vulnerabilities and prevent them from being deployed to production environment.

References

Google Cloud – The Fundamentals of container security Document: https://inthecloud.withgoogle.com/anthos-security/Dl-cd.html

Docker Page: https://www.docker.com/resources/what-container/

VMware Carbon Black Container: https://www.vmware.com/products/carbon-black-cloud-container.html

What is Container Security? https://www.mcafee.com/enterprise/en-in/security-awareness/cloud/what-is-a-container.

VMware Container Security: https://www.vmware.com/topics/glossary/content/container-security.html