The Problem With Containerization

By Kaya Ismail | *May 30, 2019*

Follow 2,240 followers

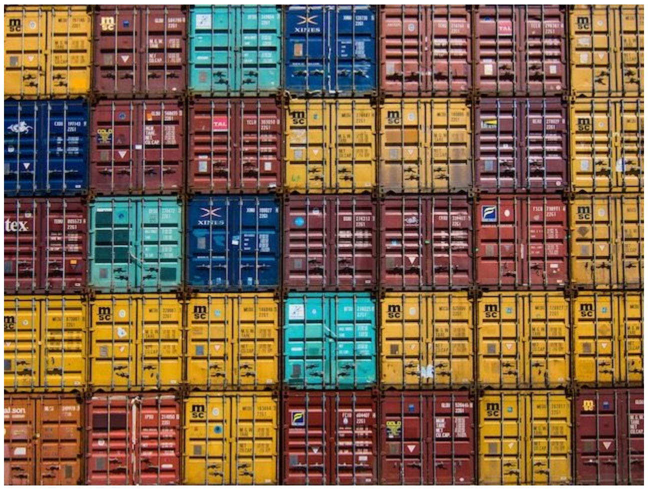

PHOTO: GUILLAUME BOLDUC

Containerization offers plenty of benefits — including potentially smaller server costs — which tempts brands to containerize applications and adopt technologies such as Docker and Kubernetes.

For companies with large, monolithic applications, the journey to containerization can be complicated, but that's not the most pressing issue. Containers may not offer the performance and cost-saving they promise, and could even expose your organization to security risks. Sixty percent of respondents in a recent study by Tripwire reported experiencing container security incidents.

Industry experts discuss containerization, the most significant drawbacks and ways to mitigate these potential issues.

What Is Containerization?

When it comes to the definition of containers, Brian Johnson, CEO and co-founder of DivvyCloud, said it best:

> "[Containers are] portable applications that are packaged up with all [the] required components [to deploy websites and applications], allowing the application to run on any system without any application-specific dependencies being installed."

Containers don't need to run a full operating system (OS) for each instance of an application as a virtual machine would. Multiple containers can share the same OS, which limits the need for numerous large virtual machines.

In addition, containerization makes it easier to manage the libraries and dependencies of an application because each image runs in isolation. Developers find containers easier to develop and deploy across multiple environments without running into configuration issues.

Related Article: **What You Need to Know About Containerization**

3 Drawbacks of Containerization

Although containers have increased the efficiency of development and deployment, DevOps teams should consider the disadvantages.

1. Performance

DevOps professionals often praise containerization for its performance benefits, but this is usually only seen with microservices architectures and not monolithic applications.

Davy Hua, the head of DevOps at Shiftleft, explained that "containers are usually kept small, from a few megabytes to a max of several hundred megabytes, which allows for quick deployment." He said that larger monolithic applications, especially ones with complicated inter-dependencies, can be in the gigabytes range. Monoliths make containers less than ideal for deployment and execution because they’re so much larger.

Furthermore, moving a monolith towards a microservices architecture can be an enormous task, and can create more complexity for your DevOps team without the proper tools to manage a multitude of containers.

Related Article: **The Benefits and Challenges of a Microservices Architecture**

2. Misconfigurations

Containerization is a relatively new technology, and for this reason, you may not have the IT expertise available to configure containers properly at your organization.

Nathaniel Quist, senior threat researcher at Palo Alto Networks, said that more often than you think, "IT teams fail to configure their containers properly, potentially leaving them exposed and creating significant security risks for their organization." He went on to describe misconfigurations such as "using default container names and leaving default service ports exposed to the public," which can leave organizations vulnerable to attackers.

To mitigate these risks, Quist suggested that "using the proper network policies or firewalls can prevent internal resources from being exposed to the public internet, while establishing basic authentication requirements for your containers ensure that even if these containers were to be found, attackers would not be able to access the data."

Related Article: **Container Security Woes Push Evolution in Two Directions**

3. Shared Infrastructure

While a shared infrastructure could mean better utilization of hardware resources for some organizations, it could leave you more vulnerable.

Glauber Costa, VP of field engineering at ScyllaDB, said, "Like other forms of shared infrastructure, containers are inherently vulnerable to so-called 'side-channel' attacks." Side-channel attacks occur based on information gained from the implementation of a system, which could be due to the misconfigurations mentioned earlier.

"Containers, in general, share the kernel with the host OS," Hua said, "and usually [have] full root access." He further clarified, "the vulnerability blast radius of a single container, or the host OS itself, can affect other neighboring containers of the same host." There's an enormous risk of vulnerabilities in one container spreading to the containers around it.

"It's easier to protect a few nodes," explained Costa, "where you're the only user [as opposed to] a massive, containerized horizontal cluster." He said it's often best to opt for isolated, single-tenant architectures on bare-metal instances to minimize your attack surface.

Weigh Your Options

The security risks from containers could lead to downtime or system outages. That's why Quist suggested that "investing in cloud security tools can alert organizations to risks within their current cloud infrastructure." If your software isn't available, your brand's reputation could take a hit and you're likely to have a significant economic impact as well.

Hua said, "alternatives to containers are bare metal and virtual machines through hardware virtualization. Each has its benefits and will be better than the other based on the specifics of each use case." Costa said you should take a "deliberative approach to using containers and virtual machines in the first place." He concluded, "You're taking a risk. Make sure you only take it where it's worth it." In the end, it's up to the individual organization to weigh the pros and cons of containerization.

0 Comments

0 Comments CMSWire.com 1 Login

Recommend Tweet Share Sort by Oldest

Start the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Be the first to comment.

Subscribe Add Disqus to your siteAdd DisqusAdd Disqus' Privacy PolicyPrivacy PolicyPrivacy

Featured Events

Jun 12 [CMSWire Webinar] 2019 State of the Customer Journey

Jun 17 Digital Workplace Experience Chicago 2019

Jun 26 CMSWire Tweet Jam:Building a Data-Driven Culture #DXchat

Jul 30 [CMSWire Webinar] Shed Light On the Darkest Corners of You DAM with AI

Nov 4 DX Summit Chicago 2019