How to accelerate container adoption

LACEWORK

As organizations move fast to develop and deliver business-critical applications, they face significant challenges in aligning agility with security and compliance. Developing applications with containers has become an increasingly popular way to operationalize speed, but it must be able to do so without jeopardizing security and compliance.

Until recently, the notion of running production workloads in containers was considered heretical. But the success of Docker and Kubernetes changed that outlook as IT teams embraced a less cumbersome, more portable way to move applications from staging into production environments. Using a complete runtime structure, containers brought libraries, dependencies, configuration files, and other binaries into a single, comprehensive package.

The popularity of containers has been on an upward trend ever since, and it's clear they are now considered critical assets to organizations trying to deliver code at speed and scale. However, their agile structure is built for widespread access and integration across diverse resources, and that can leave security gaps.

"The popularity of containers has been on an upward trend ever since, and it's clear they are now considered critical assets to organizations trying to deliver code at speed and scale."

At its most basic form, a container is essentially a packaged set of code made up of application, operating system (OS), support, and config files, all wrapped into a single, read-only image. It's an efficient format for developers to manage and scale their needs on-demand, so they have the necessary capacity to deliver more when load demands increase, and downsize when the opposite occurs. Containers help organizations optimize the elastic benefits that cloud environments offer.

Most container architectures are formed around a client/server model, where the client initiates the creation of a container, and code interacts at the server level. From that standpoint, containers can be seen as a phenomenon in the evolution of application maturity, and developers coming from legacy models will still likely be comfortable operating within a containerized environment.

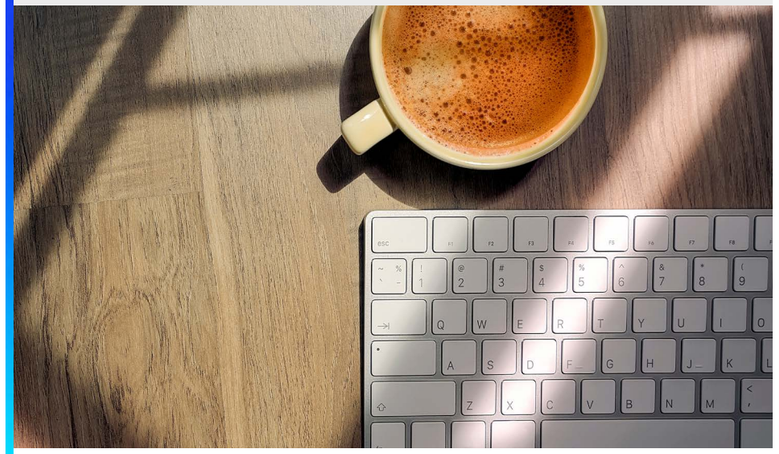
What distinguishes containers from other forms of development is that they are optimized for orchestration, whether that's among clusters of containers, or between applications. Orchestration gives them the flexibility to start and stop containers depending on demand. Additionally, the concept of the container image is critical to container concepts, as images are read only. Their job is to instruct a container server how to structure the container. They are stored in cloud repositories like AWS S3 buckets or Google Cloud Storage to be used as needed. This alone can lead to a great deal of risk—the wrong person accessing stored images means access to the control panel for applications and how they're developed.

> "Containers help organizations optimize the elastic benefits that cloud environments offer."

Addressing the specific security needs of containers

An environment might contain hundreds or thousands of containers and their respective images, each being made up of many different layers of files. Identifying security issues within a mountain of containerized applications is incredibly complicated and time consuming, and can end up dismantling all the advantages teams have realized by starting up their container infrastructure in the first place.

Consider a few key traits of containers that, while providing some advantages, also increase the risk surface across any part of the organization that a container is touching. It's in these elements of the container infrastructure that organizations need to employ a comprehensive security strategy:



Ephemerality

Because they are ephemeral by design, it's not wise to monitor a single container. Rather, IT teams need to identify clusters of microservices and containers to identify vulnerabilities. Traditional approaches look for changes to IP addresses and network access data, but these become easy to hide in single instances. IT teams need a comprehensive approach that identifies trends across entire container environments.

Configuration risk

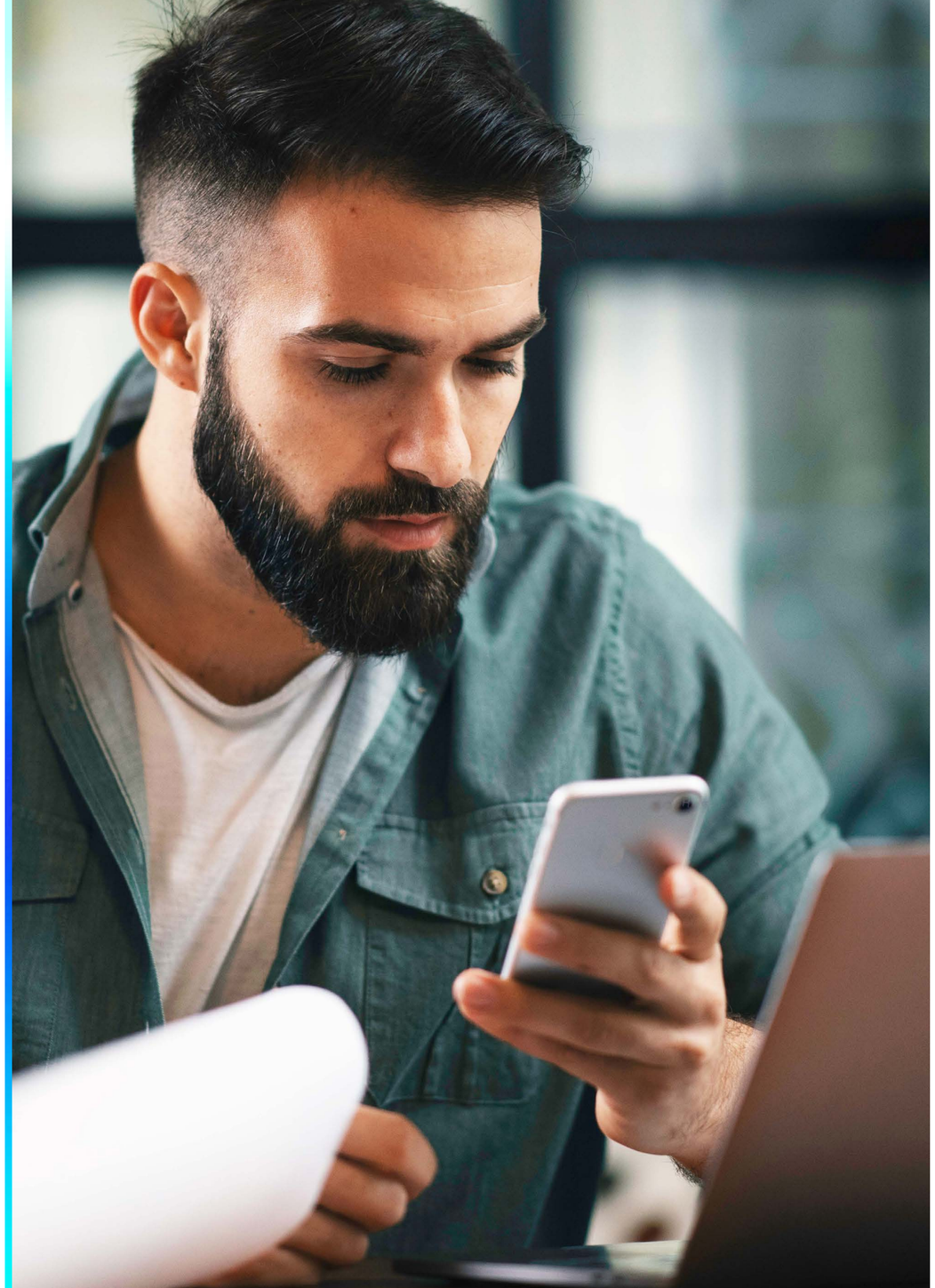
Containers are transient in nature—used for specific purposes, then discarded when no longer needed—but highly portable. They share the host OS they run on, which clearly has advantages of expedience. But it also can be prone to a different type of risk, which can come from a single misconfigured container that leaves gaps across the OS and increases risk.

Risk proliferation

Container images usually interact with and depend on other images. A vulnerability in any of these images could proliferate to thousands of containers, exposing massive troves of objects, services, and other data sources to risk.

Orchestration complexity

Orchestration software that automates container scheduling and deployment can be very complex and misconfigurations of this software can result in security vulnerabilities such as over provisioning of privileged access or an increased attack surface.



Balancing container advantages with container-specific risk

Containers are clearly capable of expediting the application development and delivery processes. But the impact is felt across the business as organizations can more quickly, and accurately, address business issues with the deployment of technology solutions.

But containers carry risk by exposing the environment to a broader array of threats. They increase the potential for visibility gaps as development becomes fragmented across many different containers. IT and security teams need to frame their container security approach:

OS vulnerability

As a matter of efficiency, containers share the host OS that they run on, and that means the host is inherently attached to the attack surface. Attacks that target vulnerabilities in the host OS will compromise every container that shares that OS. Because the OS is so broadly used across the application environment, a container-focused attack can spread rapidly, and it makes it difficult to isolate.

Rules-based approaches aren't enough

Since the modus operandi for containers is continuous change, rules are an ineffective approach to addressing security. Changing a single rule can take an IT team days to implement, whereas container changes, and the creation of security gaps are real-time and an always-on possibility.

Attack surface expansion

Orchestration platforms (Docker, Kubernetes) that automate container scheduling and deployment adds a layer of complexity. Misconfigurations at this layer can expose vulnerabilities, like over-provisioning of privileged access. The control plane and extensive use of APIs to deliver the actual compute service exposes application internals and adds to complexity. In microservice architectures, application-breakdown multiplies a small number of workloads by a factor of 10 or 100 and thus expands the attack surface.

LACEWORK

Container image vulnerability

Containers are typically built using images stored in publicly available repositories like GitHub. These images are usually dependent on other images, and when a vulnerability is detected in one, it can proliferate rapidly to any of (potentially) thousands of other containers.

Short lifespan challenges

Containers are spun up and tore down based on capacity demands. This ephemeral nature of containers renders perimeter and IP address-based security controls less effective. Moreover, overlay networks and IP address reuse obfuscate traceability. Logs and other evidence may be lost when containers are reset in response to a security incident. This renders forensic investigations difficult.

Over-permissioned access

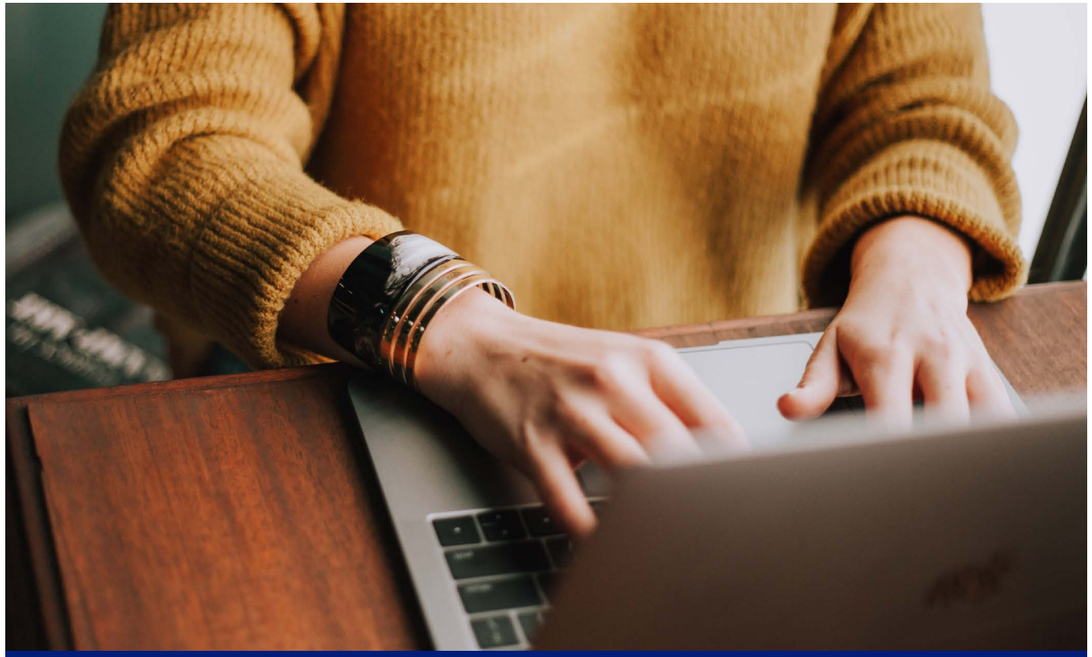
Containers are not agile. Access to the container is access to both container software and code. Visibility to who has access to what is within the container is not transparent. Unauthorized installs, abnormal login attempts/failures, key file changes are all examples of insider threats.

Change management

Orchestration tools have full permissions across all containers and can access their objects and services. Changes made to a single container replicate to all containers in a cluster. Replication of an error or vulnerability occurs rapidly, and as we've seen is difficult to isolate.

Network management

Containers and container clusters reside on a cloud network; cloud network security has the same dependency as on-premise security for patch and upgrade management. If the operating systems are not updated with current patches and supported versions, an exploit is feasible.



Containerized applications deployed in the cloud make it easier for organizations to give needed services to their customers more quickly. There continue to be exciting advances in technologies that analyze behavior in real-time to monitor for and alert on misconfigurations and other vulnerabilities in cloud-based container infrastructures. Coupling these technologies with anomaly analysis and evolved security best practices will create the necessary threat detection, protection, and response controls essential to keeping these dynamic clouds secure.

Utilizing a platform approach to container security

The Lacework platform employs the following container-specific approaches to ensure users have the necessary level of threat detection and remediation, and their general concepts should be part of any security-first organization:

Security risks introduced by containerized applications and their supporting services and infrastructure can be discovered and mitigated or remediated by applying old principles and proven techniques together with new and updated tools. Lacework was among the first cloud security vendors to highlight this need and its comprehensive cloud and container security platform resolves container-specific security challenges across the application lifecycle, and at every layer of the containerized stack.

Misconfiguration threat impact

Misconfigurations in container provisioning could result in a larger, more vulnerable surface area or allow untrusted access to trusted resources. Resources like the CIS Benchmark provide prescriptive security guidance for most operating systems and many applications including Docker and Kubernetes. Lacework supports the inclusion of these benchmarks as threat intelligence feeds for vulnerability scanning.

Visibility must deliver actionable insights

Orchestration and cloud administrative dashboards make it easy to manage all cloud and container assets from just one point, but a breach of these systems could lead to a full compromise of your entire cloud or containerized environment. Be sure to secure these dashboards with two factor authentication and regularly audit access. Keep an inventory of all your cloud subscriptions to ensure each cloud asset remains appropriately included in your security program.

The importance of trusted images

Containers are built from images stored on public or private repositories. Understand where your images are sourced and look for tools to scan these images for any vulnerabilities. Remember that one vulnerability in a single image will proliferate into every container based on that image.

Cloud-native security is not enough

Public cloud platforms offer native tools for containers and orchestration. Microsoft Azure, for example, offers fine-grain Identity and Access Management (IAM) to access Kubernetes resources, policy-driven communication paths between resources to secure Kubernetes workloads. There are also tools to define platform-specific security policies across multiple clusters and to track, validate and reconfigure nodes, pods, and other container images for compliance. These controls are again dependent on sufficient runtime visibility into configuration compliance, real-time anomaly detection and alerting, and traceability.

Real-time alerting of container vulnerability

Behavior-based anomaly detection exposes threats due to deviations from the normalized behavior of various cloud entities. Lacework continuously monitors traffic and events across containers and uses unsupervised machine learning to detect and alert abnormal behavior in real-time.

Evolving to meet the security needs of containerized environments

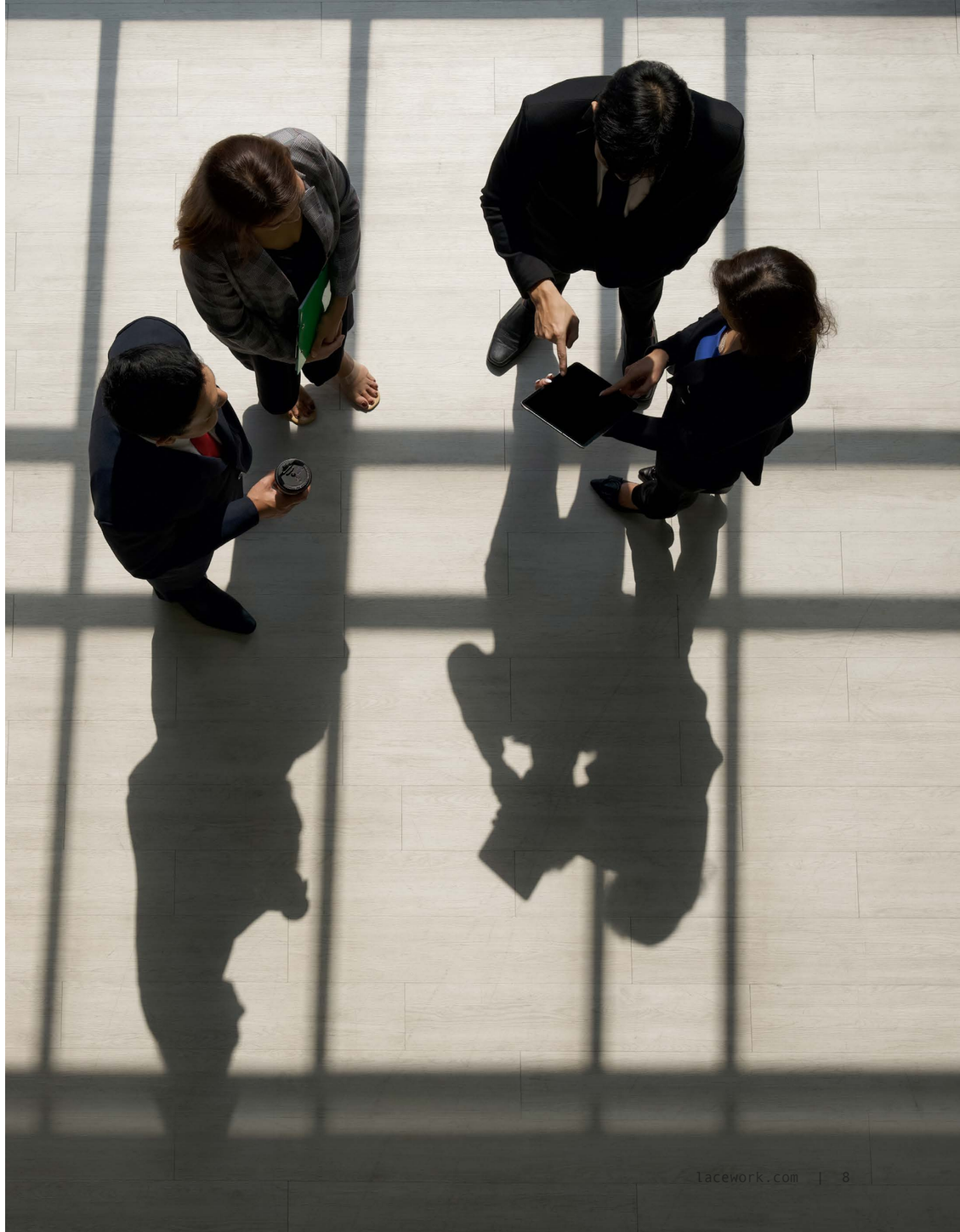
In addition to adapting existing security controls to cloud based container environments, new tools and techniques will still be required to meet all the security challenges in today's agile environments.

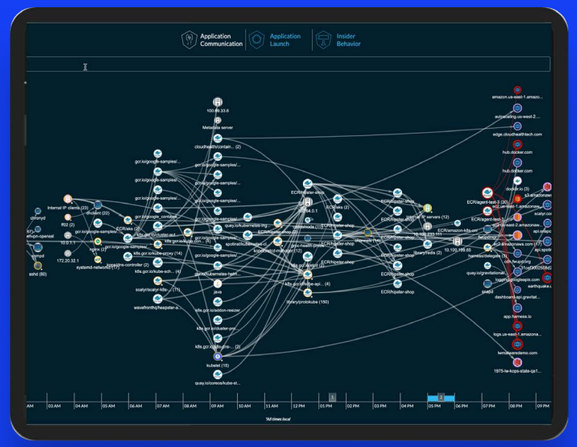
As microservice architectures evolve, new security best practices must be developed. Using immutable objects will reduce an attacker's ability to gain a foothold or tamper with the service. As an example, every microservice should be threat modeled and its surface area managed for the specific function it provides (i.e., an internal service should not be accessible from the internet).

After a breach, compromised containers can be shut down and quarantined while clean replacements are easily brought online. New tools that correlate container activity to users and processes will be necessary to ensure traceability across a security incident and appropriate evidence and other bread crumbs are preserved to help investigators.

Conclusion

Containerized architectures deployed in the cloud are essential for organizations to deliver products and services with speed and agility. Adopting the necessary security to protect containers must be applied quickly with a comprehensive approach that supports rapid development and deployment, and is still adaptable while container needs change. Lacework provides a comprehensive container security solution that allows organizations to apply real-time threat and anomaly detection, process-level security visibility, and host and configuration compliance controls with immediacy, and with the ability to evolve as the enterprise's needs change. This platform-based, comprehensive approach gives organizations a container security infrastructure that can be employed immediately to meet the changing needs of today's threat landscape.





Ready to chat?

Request a demo

Lacework delivers security and compliance for the cloud generation. The Polygraph® Data Platform is cloud-native and offered as-a-Service, delivering build-time to run-time threat detection, behavioral anomaly detection, and cloud compliance across multi-cloud environments, workloads, containers, and Kubernetes. Trusted by enterprise customers worldwide, Lacework significantly drives down costs and risk, while removing the burden of unnecessary toil, rule writing, and inaccurate alerts. Lacework is based in San Jose, California, and backed by Sutter Hill Ventures, Liberty Global Ventures, Spike Ventures, the Webb Investment Network (WIN), and AME Cloud Ventures.

Get started at www.lacework.com

LACEWORK®