aqua

Container Security Checklist

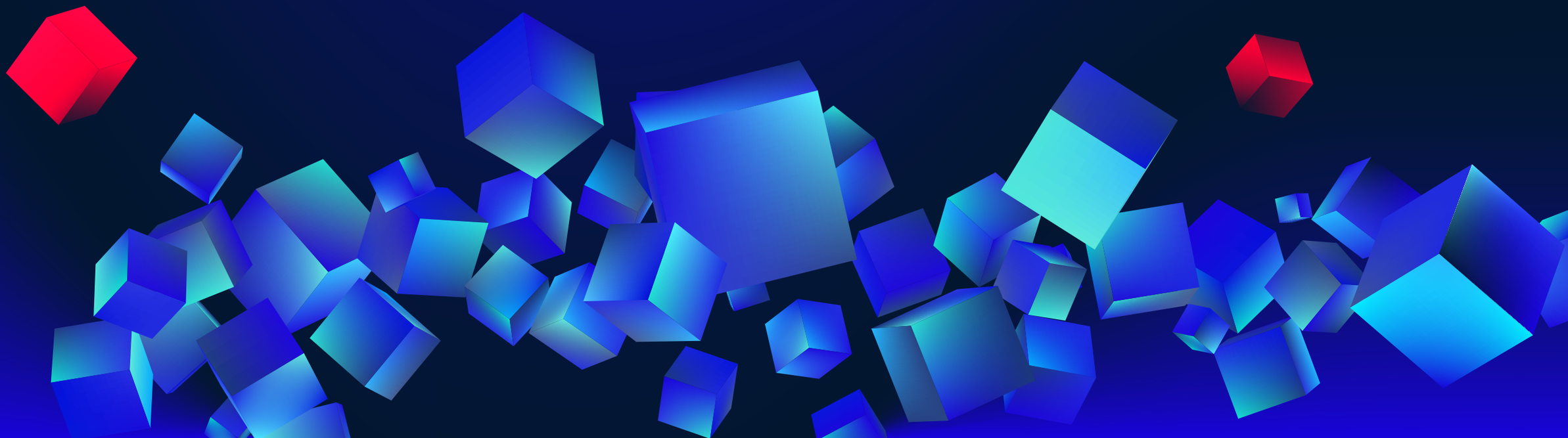
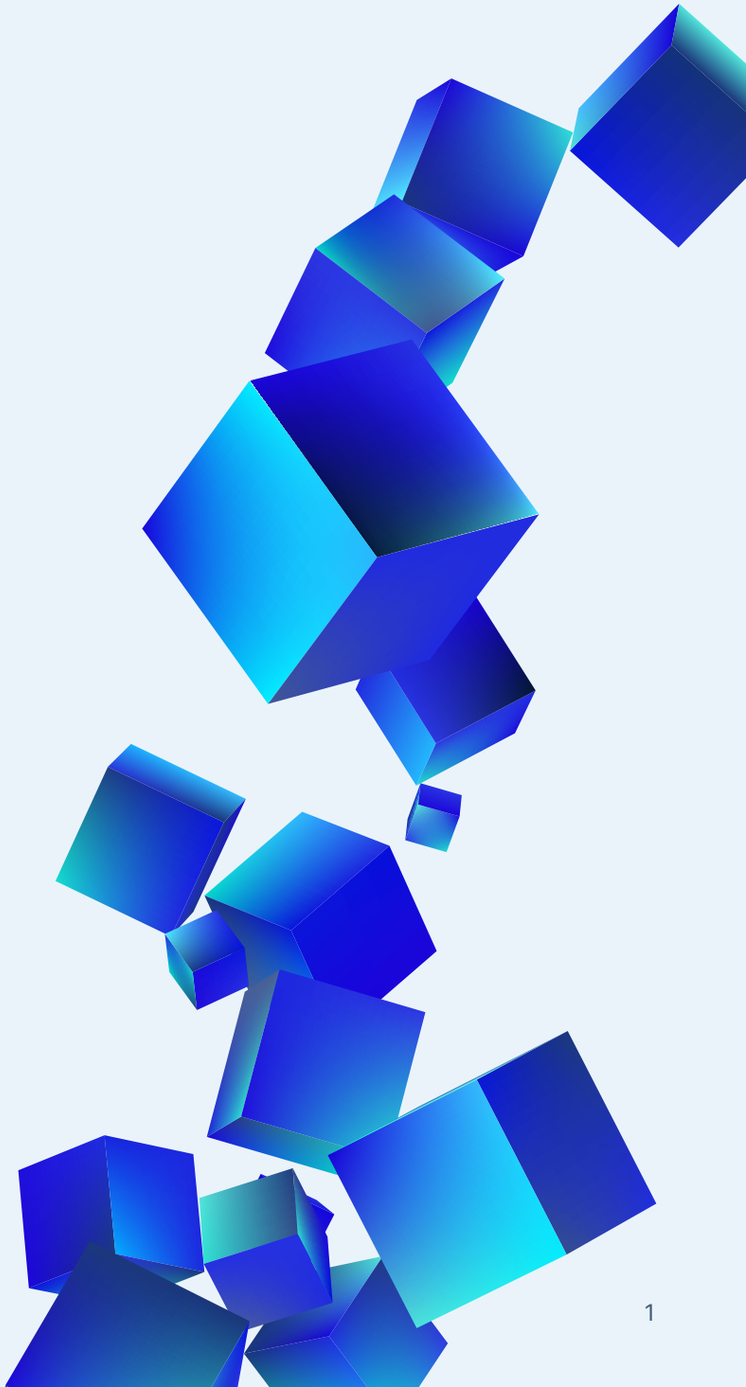


Table Of Contents

Introduction

In the dynamic world of modern application development, containerization is a game-changer, enabling rapid deployment and scalability. However, as the dynamic nature of containers continues to grow, so do the security challenges they present. Ensuring the security of your containerized environment is not just a technical necessity but a strategic imperative.

This checklist examines the critical aspects of container security, offering a comprehensive guide to reinforce your defenses. Fromsecuring the images that form the foundation of your containers to establishing robust incident response protocols, these best practicesare designed to help you navigate the complexities of container security with confidence and precision.



Build Securely

Continually scanning images and using trusted sources ensures that your container environment is free from vulnerabilities, leading to safer and more reliable deployments. Protecting your registry and orchestration platforms with strict access controls and monitoring reduces security risks and enhances overall system integrity. Managing secrets and restricting access to only necessary containers ensures that sensitive data remains secure, providing a strong foundation for your containerized applications.

Secure Container Images

Strengthen Kubernetes security: Harden Kubernetes environments by implementing access controls, securing pod communications, and using network policies to prevent lateral movement of attacks.

Limit privileges: Use allow-list techniques to control access and minimize the risks associated with over-privileged accounts within orchestration platforms.

Harden the host: Ensure that the operating system hosting the containers is secured with appropriate access controls and is continually monitored for vulnerabilities.

Secure the Host and Orchestration Platform

Scan continually: Regularly scan container images for vulnerabilities before and after deployment to identify and mitigate risks promptly.

Use trusted sources: Ensure that images are sourced from trusted, verified repositories to avoid the introduction of malicious code.

Maintain a bill of materials: Document all software components within container images to track and manage security vulnerabilities effectively.

Manage Secrets and Sensitive Data

Avoid hardcoding secrets: Use dedicated secrets-management tools to securely store sensitive data like API keys and credentials, ensuring that they are not hardcoded into images.

Implement access controls: Limit access to secrets to only those containers and services that absolutely need them.

Secure the Registry

Secure the registry: Protect the container registry by enforcing strict access controls and monitoring for unauthorized changes or vulnerability escalations.

Monitor continually: Implement continual monitoring to detect vulnerabilities in stored images and lock down the registry server to prevent unauthorized access.

Deploy with Confidence

Real-time monitoring with centralized logging and time-stamped data collection enables quick detection and response to potential threats, reducing the risk of incidents. Consistently enforcing security policies across all containers ensures a secure and compliant environment, giving you peace of mind during deployment. Automating compliance further reduces the risk of human error, ensuring that your containerized applications are always protected and ready for production.

Monitor in Real Time

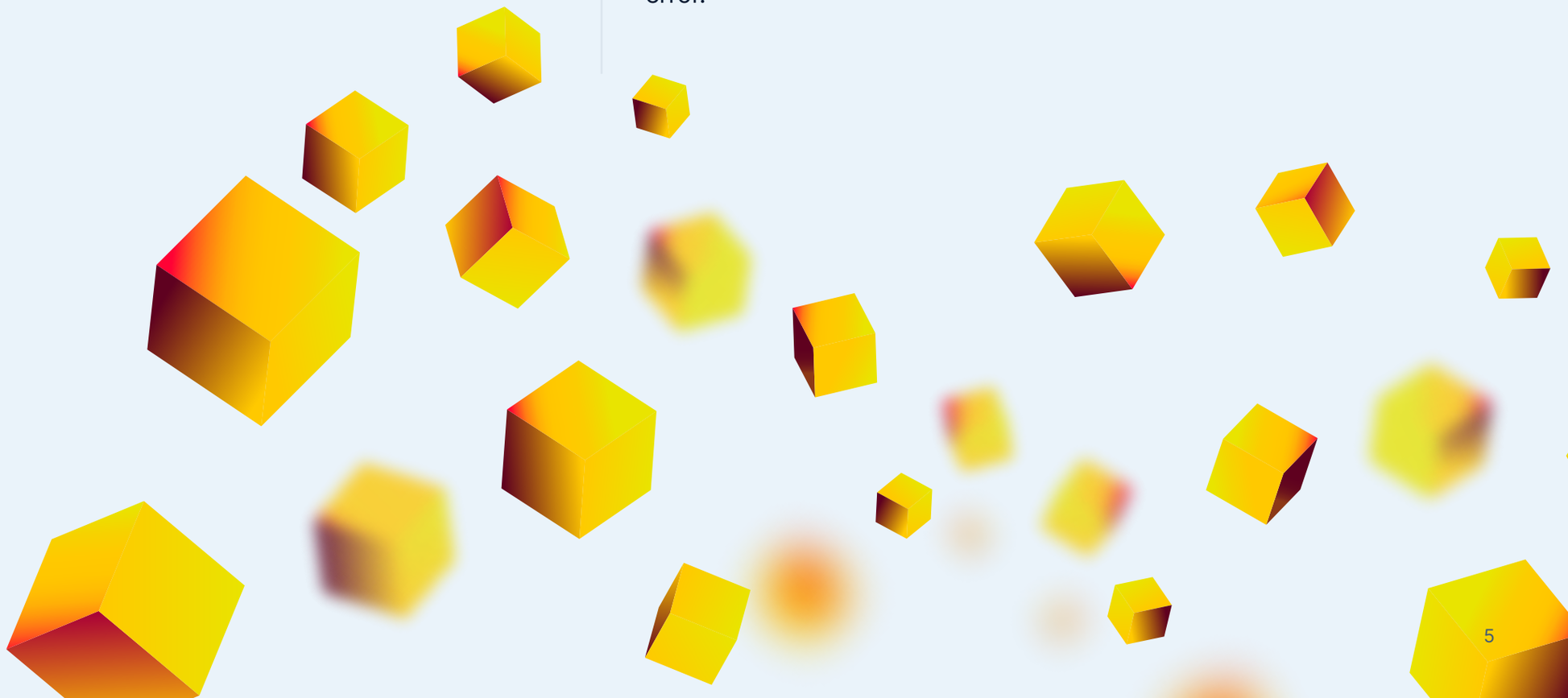
Centralize logging: Implement centralized logging to capture container activity across all environments, enabling efficient tracking and incident response.

Monitor time-stamped data: Use monitoring tools that support time-stamped data collection to accurately trace actions and identify potential security breaches.

Enforce Assurance Policies

Define and enforce policies: Establish security policies and assurance gates that define what is allowed in the container environment and use automated tools to enforce these policies consistently across all containers.

Automate compliance: Automate the enforcement of security policies to ensure continuous compliance, reducing the risk of human error.



Run Securely

Protect containers in runtime with prevention controls to lock down the environment and with real-time behavioral detection to catch threats that others miss. Respond quickly by using rich runtime context to prioritize risks and running regular incident response drills for effective handling of security incidents. Secure hybrid and multi-cloud environments with a unified security framework and extend protections to serverless components, ensuring consistent security across all cloud native workloads.

Protect In Runtime

Protect in real time: Identify threats by using behavior- and signature-based detection methods to uncover known and unknown threats before they reach production.

Ensure workload immutability: Establish baselines for container behavior and monitor for any deviations that might indicate a security threat or compromise.

Reduce the attack surface: Eliminate potential entry points for threats by hardening the runtime environment, ensuring restricted access, and preventing any lateral movement or escalation within or between workloads.

Respond Quickly

Respond with context: Leverage rich runtime context to focus only on the highest-priority risks to make better remediation decisions, respond fast, and save time.

Run simulations: Conduct regular incident response drills to ensure that your team can handle real-world container security incidents effectively.

Secure Hybrid and Multi-Cloud Environments

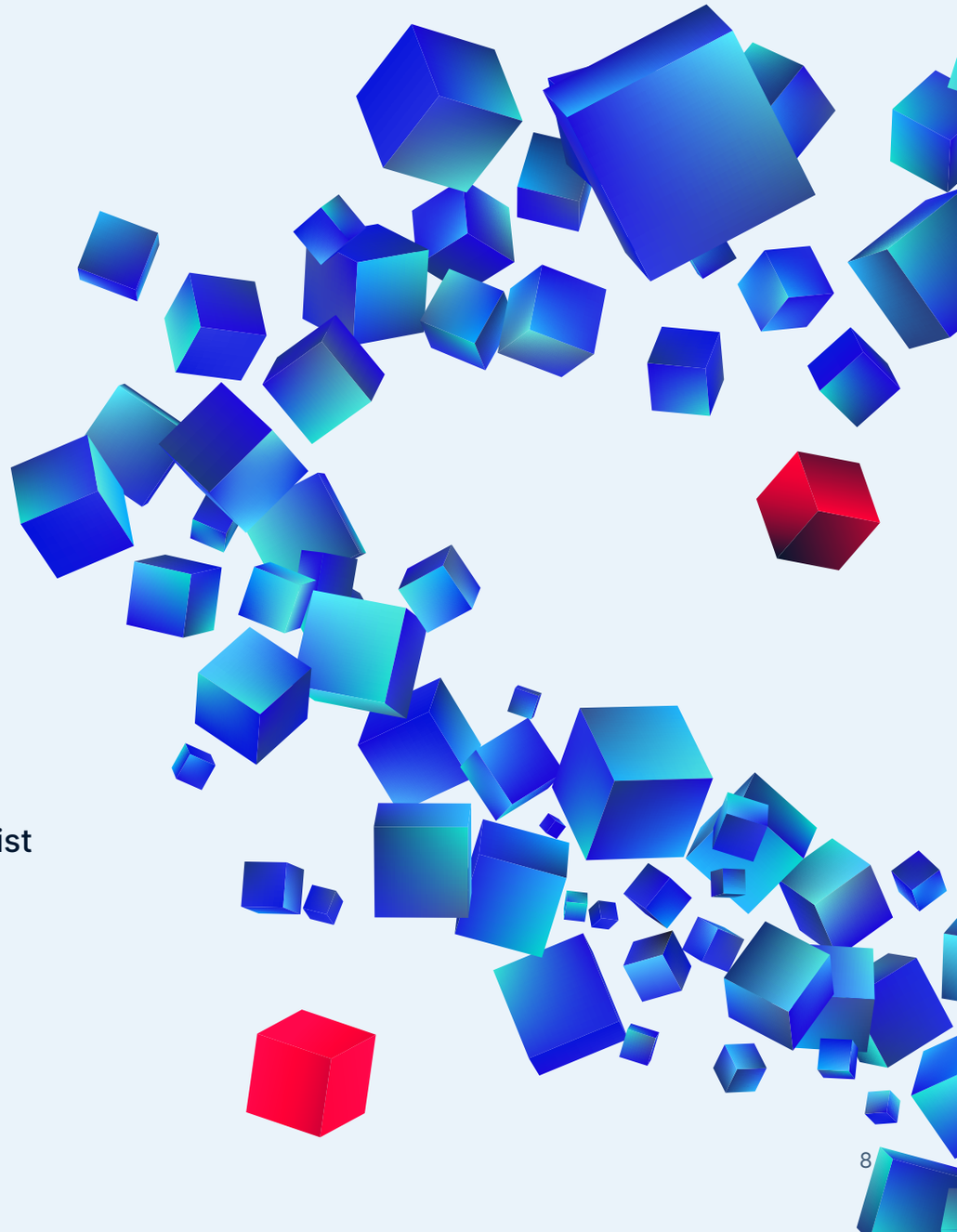
Define and enforce policies: Establish security policies and assurance gates that define what is allowed in the container environment and use automated tools to enforce these policies consistently across all containers.

Automate compliance: Automate the enforcement of security policies to ensure continuous compliance, reducing the risk of human error.

Conclusion

Ensuring comprehensive security for containerized applications requires a multi-layered approach covering the entire life cycle, from development to production. Implementing the best practices outlined in this checklist can significantly enhance your security posture.

Aqua Container Security is a leading solution that provides full life cycle security for containerized applications, regardless of deployment location. Aqua's solution identifies common security flaws early in the build phase, integrates acceptance gates within the CI/CD pipeline to reduce the attack surface, and offers real-time protection in production. By adopting the best practices in this checklist and leveraging Aqua's robust container security capabilities, you can ensure a strong security posture for your containerized applications, safeguarding them against potential threats and vulnerabilities.



aqua

Aqua Security is the pioneer in securing containerized cloud native applications from development to production. Aqua›s full lifecycle solution prevents attacks by enforcing pre-deployment hygiene and mitigates attacks in real time in production, reducing mean time to repair and overall business risk. The Aqua Platform, a Cloud Native Application Protection Platform (CNAPP), integrates security from Code to Cloud, combining the power of agent and agentless technology into a single solution. With enterprise scale that doesn't slow development pipelines, Aqua secures your future in the cloud. Founded in 2015, Aqua is headquartered in Boston, MA and Ramat Gan, IL protecting over 500 of the world's largest enterprises. For more information, visit https://www.aquasec.com.

Schedule demo ›