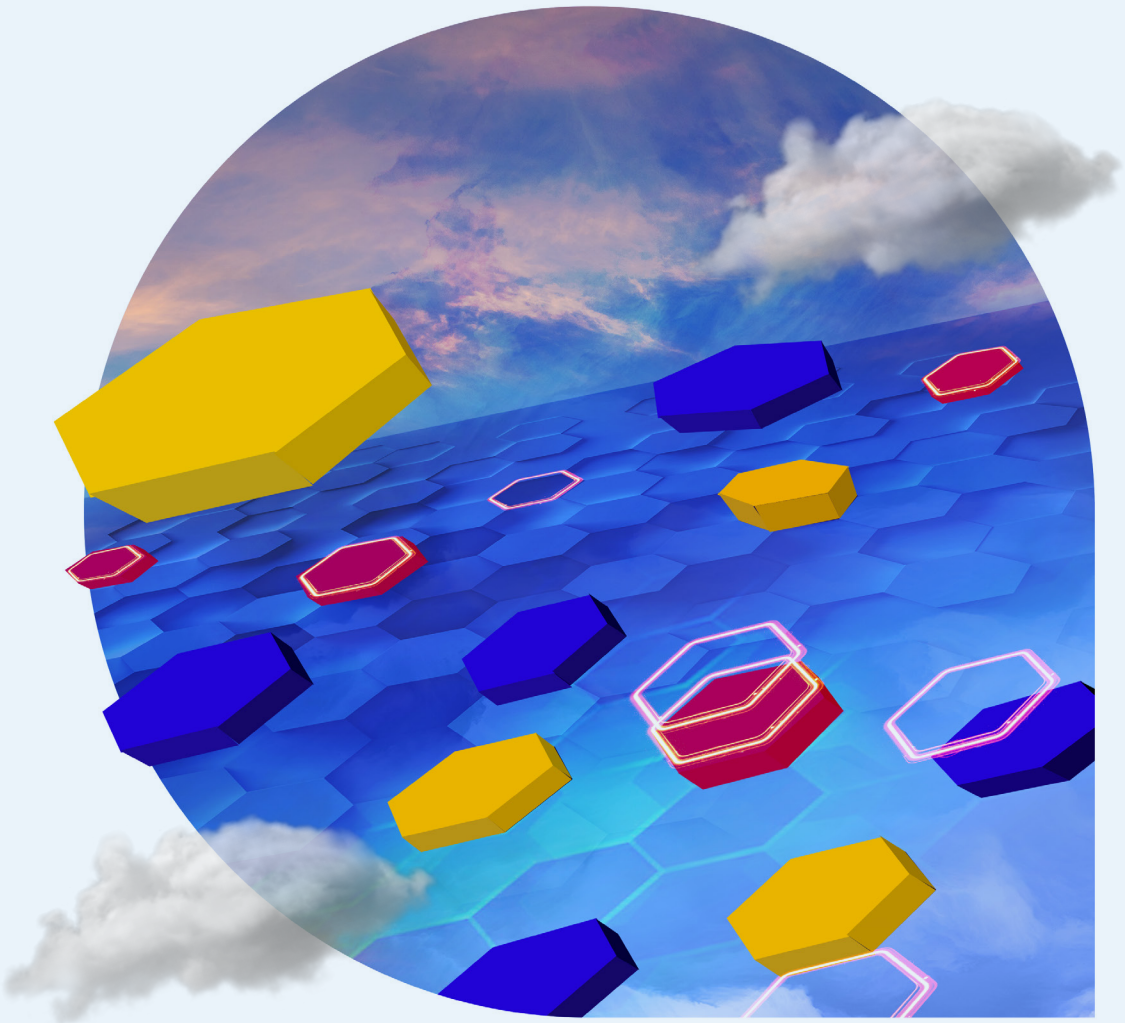
aqua | aws

Designing Zero Trust Security for AWS



Feb. 2023

What is Zero Trust?

Zero trust is a cybersecurity framework that focuses on securing assets and data as opposed to just establishing a perimeter (firewall) or layering defenses around the network. Before it was common to work outside of the network with cloud and edge devices, it was believed that firewalls were enough to secure the entire network and your data was protected. However, as more users began leveraging different devices, applications, and remote access to work, it became obvious that security had a gap.

In 2010, Forrester cyber security analyst, John Kindervag, recognized this and developed the Zero Trust architecture based on the simple concept that all users should automatically be untrusted. He applied this idea to organizational security practices by incorporating principles of identity verification and controlled system access.

It quickly became a cornerstone of all major platform, cloud, and cyber security providers solution offerings throughout the years. Then in 2020 NIST and NCSC researched many different Zero Trust strategies and implementation methodologies to create an official standardized Zero Trust Architecture (NIST SP 800-207).

Core Principles of Zero Trust Architecture

Never Trust, Always Verify Always authenticate and autorize user identity, location, device, data sources, services, or workload. Continuous verification means there are no trusted zones, devices, or users. Zero Trust treats everyone and everything as a potential threat.

Assume Breach Increase your security posture against potential threats, minimizing the impact if a breach does occur. Limit the "blast radius" or the extent and reach of potential damage incurred by a breach by segmenting access, reducing your attack surface, and monitoring your network in real time.

Least Priviliged Acess Limit access rights for any entity and only permits the minimum privileges necessary to perform its function. In other words, it prevents unnecessary access across the network, which leaves your netowrk vulnerable and creates a higher attack surface in case of a breach.

Zero Trust Challenges with Cloud Native

Despite the many benefits of cloud adoption, cloud native architectures significantly changed the risk management landscape for security teams with unique entry points and blind spots that traditional tools and processes can't detect.

Network-centric security approaches like firewalls, intrusion prevention systems, and intrusion detection systems were designed to keep threats from breaching the network perimeter. They do not account for the new addition of cloud solutions that enable access to corporate resources outside the safety of the network. Cloud native technologies, pipelines, infrastructure, and processes create new challenges for implementing zero trust architectures.

Key Components of Cloud Native Applications

Always assume the application is compromised at every stage from the first line of code to its behavior in production, taking into consideration the tools used to build it. By eliminating trust from the development process, zero trust security provides confidence that the applications being released remain protected and secure.

Elements of modern development process that security and development teams need to consider:

1 **Software Supply Chain** Everything that touches an application or its code, including all the tools used to build and deploy, known as CI/CD pipelines. Attackers prioritize exploiting these pipelines because they are often the weakest link and an effective target for attacks.

2 **Application** How an application is designed and executed opens the door for embedded vulnerabilities within written code or open-source packages.

3 **Images** The application will be invoked using container images that are stored in a designated repository. The images should be configured to match the application's needs without any unnecessary items such as secrets, or over-permissive rights.

4 **Host** The infrastructure used to run the application, whether it's a VM on a public cloud or a physical server. It should be secured and hardened according to industry standards such as CIS to prevent breaches.

5 **Cloud Workloads** There are a few types of cloud workloads such as containers, serverless and more. There is a need to ensure they are configured properly and behave according to the application they are running

Three Main Challenges Enforcing Zero Trust

1 Exponential Attack Surface

The move to cloud native changed the way applications were built. Microservices are being adopted so applications can communicate more efficiently over the network and support agile fast development. This change in application communication process became possible because containers and Kubernetes enable microservices to be deployed and orchestrated easily.

This innovative approach leads to new environments in which large numbers of containers from many types communicate with each other using APIs. Expanding the environment even further are dozens of servers, thousands of containers, and adoption of open source that potentially have vulnerabilities embedded within layered dependencies.

2 Visibility & Methodologies

Businesses adopt cloud native DevOps methodology to increase speed and agility of application delivery. New services and versions are more frequently (continuously) deployed, in smaller batches. In some cases, updates happen several times a day, as opposed to weeks or months for a major release. These deployments are carried out in a centralized way. Each team or each developer is responsible for specific point parts of a larger application, which means faster development velocity.

Despite these efficiencies developers can realize to build applications faster, the process change can be difficult for cross-functional teams working to define, implement, and enforce zero trust policies across the software pipeline.

Teams need visibility in order to define how to trust for cloud native applications and standardize consistent controls that continuously validate the processes. This helps determine if application access requests meet trust levels, as well as detect and prevent processes or access that is untrusted. Starting from when code is being written, account for dependencies, and extend all the way to when applications are running.

3 Tools

Dev tools are key to supporting zero trust practices and solving for visibility and complexity of an expanding architecture. Simply, old development tools used to support traditional network architectures don't provide an integrated, automated, and contextual way to asses health of open source components, proprietary code, infrastructure configurations, containers posture, and enforce policies within the tools themselves. They also lack integrations with SIEM tools which is critical for end-to-end observability and time-saving automation needed to enable zero trust at scale.

Designing AWS Zero Trust with Aqua

Aqua Security is an AWS partner that enables organizations to see and stop threats across every phase of the software development lifecycle (SDLC), from dev to cloud and back. AWS is a highly flexible identity aware cloud provider that provides many Zero Trust foundational elements, however security of the AWS applications and environment fall on the customer because of the shared responsibility model. Implementing Zero Trust with Aqua for AWS provides:

Protection for workloads running on Amazon EKS

Prevent unauthorized images from running in your EKS cluster. Enforce container immutability, network segmentation, and segregation of duties.

Security of Applications running on AWS Fargate containers

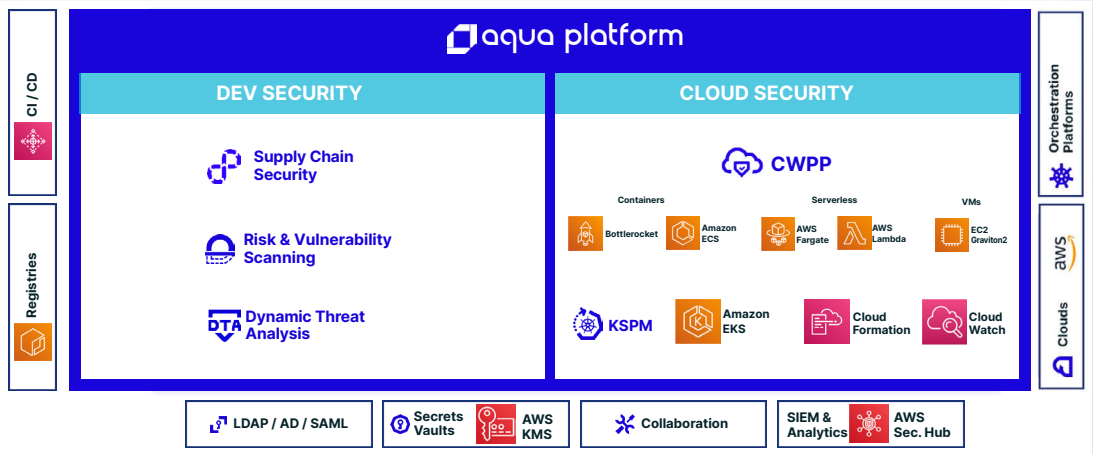
Ensure that workloads running on AWS Fargate, ECS, EKS and Graviton2 powered workloads are performing only their intended function by monitoring. Detect vulnerable or compromised containers with continuous verification and security scanning.

Extended security from Amazon ECR to Amazon ECS

Manage image vulnerabilities and ensure on only trusted images can be deployed. Implement policy mangement that automates use of acceptable containers that pass dynamic analysis and identifies anomalous behavior.

Protection of AWS Lambda Functions

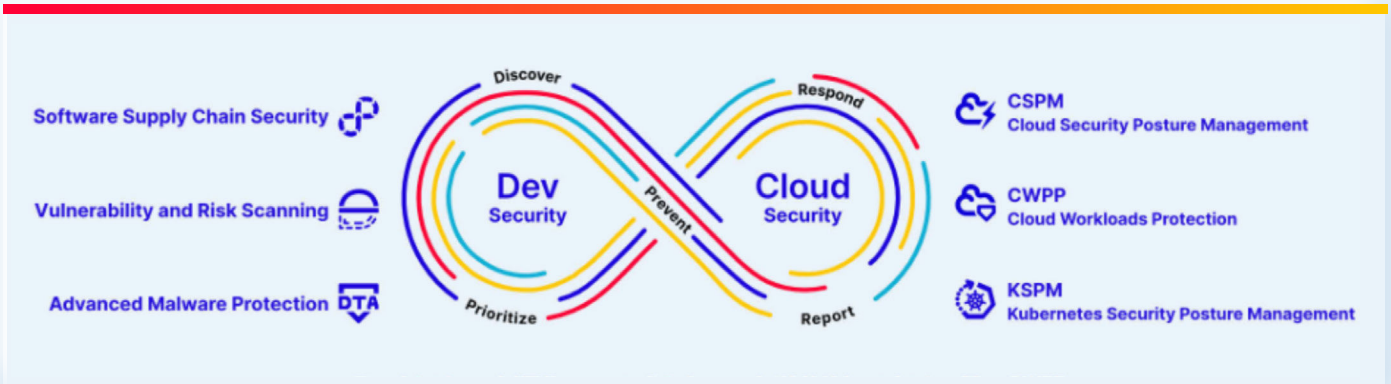
Limit risk of AWS Lambda functions by enforcing Zero Trust principles of least privileged access and identify over-provisioned permissions and roles, embedded credentials and keys, and vulnerabilities. Monitor functions at runtime, preventing code injection and malicious activity.



Aqua enables Zero Trust architectures by providing security of the entire application lifecycle from the first line of code to running in production, and takes into consideration the security of the tools used to build it. It supports the decentralization of business processes which enables the fundamental shift from perimeter defenses to a modern security posture.

Cross-functional teams including DevOps, Kubernetes administrators, site reliability engineers (SREs) and security engineers can eliminate point solutions and are united in a platform, gaining the visibility and communication needed to achieve Zero Trust architecture. Making it clear where to limit access permissions, continuously verify identity, and enforce policies across the cloud native application lifecycle.

The Aqua platform moves organizations away from implicit trust models of security by providing software bill of materials (SBOM) validation, DevOps and CI/CD pipeline integrity, vulnerability assessment, cloud security posture management, Kubernetes assurance and security, application scope least privilege and role-based access controls, logical micro-segmentation and runtime policies including drift prevention to stop untrusted activity.



Automate Enforcement of Zero Trust with Dev and Cloud Security

Dev Assurance Policies

Are critical for both security and DevOps teams to create a shared sense of trust around the guardrails for allowing artifacts into production. In an environment where implementing trusted code is critical across multiple pipelines and rapid code commits, using assurance policies to control the parameters of what should and shouldn't be allowed will lower the overall amount of noise. This will ultimately reduce the overall attack surface and making runtime policies more effective. Dev policies are best applied holistically across images, VMs, serverless functions, and orchestration tools such as Kubernetes.

Cloud Runtime Assurance Policies

Protect cloud native workloads across a mixed environment of VMs, containers, and functions, with purpose built controls for each. As such, runtime controls for containers should protect against any behavior that's not in line with the container's original intent. Enforcement happens with minimal management overhead and no impact to runtime.

Leverage One Platform to Continuously Verify All Components

Secure the software supply chain

Ensure that only known good software artifacts, images, and containers that are free from vulnerabilities and malware are promoted.

Ensure integrity of the CI/CD pipeline

Enforce developer least privilege access, code analysis, and misconfiguration checks. Validating the software bill of materials based on security checks.

Govern cross-functional team privileges and secrets management integration

Defining multi-team role-based access controls for system resources permissions, defining cross-cluster application scopes for least privilege access and automating injection secrets at run-time with no user interaction through vault integration.

Standardize control with assurance policies

Defining and consistently enforcing 'known good' assurance policies for supply chain integrity, container images, container configurations, Infrastructure as Code templates, Kubernetes clusters and pods, Kubernetes admission controller policies, and cloud account and hyperscaler least privilege and best practices settings.

Automate runtime security

Detecting and blocking untrusted activities by enforcing container immutability, enforcing logical network segmentation and blocking remote code execution, with monitoring and ongoing attack analysis

Enable response with visibility, risk insights, and monitoring

A single pane of glass that consolidates reporting, findings and insights on your unique compliance policies. Context aware insights drive fast risk assessment of images, configurations and Kubernetes clusters and incidents. Extend response and remediation with integration of SIEM, workflow, and observability tools. Granularly audit trails of all access activity through your repository of SBOMs, built for every artifact.

How to Apply Zero Trust Principles with Aqua

Dev Security

- **Ensure the integrity** of the pipeline and promote only validated SBOMs
- **Secure the software supply chain** through image scanning and malware detection ensures that code and container images are free from vulnerabilities.
- **Enforce and validate** developer access controls and least privilege access
- **Establish security gates** for malware and malicious before artifacts are promotedwith Dynamic Threat Analysis
- **Parse and evaluate** infrastructure as code templates and modules for misconfigurations, vulnerabilities,and secrets

Cloud Native Infrastructure

- **Scan cloud account settings** for misconfigurations, identify over-provisionedaccess and surface risky setting and vulnerability combinations
- **Scan Kubernetes clusters** for misconfigurations, scan Kubernetes hosts for malware and vulnerabilities and validate compliance with industry standard benchmarks
- **Enforce Kubernetes assurance policies** with validating and mutating admissioncontroller policies
- **Enforce image assurance policies** – allow only trusted images to run
- **Extend and enforce Kubernetes** role-based controls across clusters and applications
- **Define and enforce** cross-cluster application scopes for least privilege access

Cloud Security

- **Enforce runtime policies** based on workload behavior, unauthorized access,process or modification and detection of untrusted activity
- **Enforce container** immutability with Aqua Drift Prevention
- **Enforce assurance and compliance policies** for any software artifact that hasn’tbeen validated
- **Monitor container activity** for indicators of compromise using Aqua Cloud NativeDetection and Response to help mitigate issues quickly and minimize disruptionsPage13ConclusionConclusion

Aqua Security is the largest pure-play cloud native security company, providing customers the freedom to innovate and run their businesses with minimal friction. The Aqua Cloud Native Security Platform provides prevention, detection, and response automation across the entire application lifecycle to secure the build, secure cloud infrastructure and secure running workloads wherever they are deployed. Aqua customers are among the world’s largest enterprises in financial services, software, media, manufacturing and retail, with implementations across a broad range of cloud providers and modern technology stacks spanning containers, serverless functions, and cloud VMs.

Schedule demo ›