Anthos: An Opportunity to Modernize Application Security

Anthos Whitepaper
Sept 2019

Google Cloud

Google Cloud

Table of Contents

Anthos: An opportunity to modernize application security

Anthos is a modern app management platform from Google Cloud that aims to deliver a consistent development, operations, and security experience across cloud environments. Anthos is designed for enterprise organizations that want to accelerate the development and deployment of dynamic apps and who value service automation, cost governance, and security controls.

Anthos enables you to build apps once and run anywhere. With Anthos, as an enterprise organization you can modernize apps in place, automate at scale, and manage consistently across hybrid- and multi-cloud environments.

With Anthos, you can modernize your approach to app security. As much as possible, Anthos aims to provide security by default and can help you automate the following security operations:

- Enforcing consistent policy across environments.
- Isolating workloads with different risk profiles.
- Deploying only trusted workloads.

Security challenges of modern apps

Modern apps differ from traditional apps in three key attributes: microservices architecture, declarative configuration, and high degree of automation. To learn more about app modernization, see Application modernization and the decoupling of infrastructure services and teams.

However, without a way to enforce consistency and manage workloads that span environments, your organization can face security challenges when modernizing their apps.

The following are three key security challenges that emerge with modern apps:

- Applying consistent policies across heterogeneous environments.
- Securing the software supply chain.
- Securing multi-tenant environments across a shared platform.

Consistent policies

Microservices are dynamic, ephemeral, and you can distribute them across many hosts, clusters, or even clouds. As your services are deployed, shutdown, and redeployed, it's difficult to maintain security policies that are consistent--a problem known as cluster sprawl.

Automated, declarative policies that you deploy using configuration management tooling help you to ensure consistency across diverse environments. Namespaces provide logical group abstractions across services and let your policy admins set up guardrails for tenant-specific environments. You can use inheritance to delegate organization-level policies to tenants.

In this new model, access control and policy enforcement must be declarative and automated to conform to controls that meet your organization's security, risk, governance, and compliance requirements.

Software supply chain security

Continuous integration and continuous development deployment (CI/CD) practices for modern apps help your teams ship faster and scale more easily. You can build containers from scratch inside CI/CD systems. However, the ease and flexibility with which you can build containers, and the resulting workflow where containers are replaced frequently to add new functionality or patch vulnerabilities creates new security challenges.

Each container build starts with a base operating system (OS). A frequent challenge is the proliferation of base OSs, often with varying patch levels. You might have provenance of code issues because your developers can download new components and libraries and incorporate them into builds.

While a developer-centric build and deploy model enables rapid deployment, security practices such as vulnerability scanning and patching are hard to enforce, leading to a lack of standardization which undermines strong governance.

This new model requires a means for automating the process of deploying only trusted workloads.

Security of multi-tenant environments across a shared platform

In a container architecture, multiple containers run on the same host, and share the same set of machine resources. This architecture delivers portability and resource efficiency benefits, but introduces a different threat model—workloads running in the cluster can have different risk profiles and need to be treated accordingly, while sharing the same host. For example, you can't apply traditional IP-based authorization in this new architecture. The mechanisms that separate containers such as control groups (cgroups) and namespaces, and security modules such as AppArmor can be adequate for some workloads, but inadequate for the risk profile of others.

This new model requires new security controls that address network policies, workload isolation, and service authentication.

Anthos: Modernizing Security for Hybrid and Multi-Cloud

By using Anthos, you can enforce consistent policy across environments, deploy only what you trust, and isolate workloads with different risk profiles.

Enforcing consistent policies across environments

Anthos helps your organization enforce consistent policy across clusters through Anthos Config Management, a centralized, declarative configuration manager that works consistently across on-premises and multi-cloud deployments. You can use Anthos Config Management to create a common configuration for all your admin policies and apply it to all your clusters, wherever they're deployed.

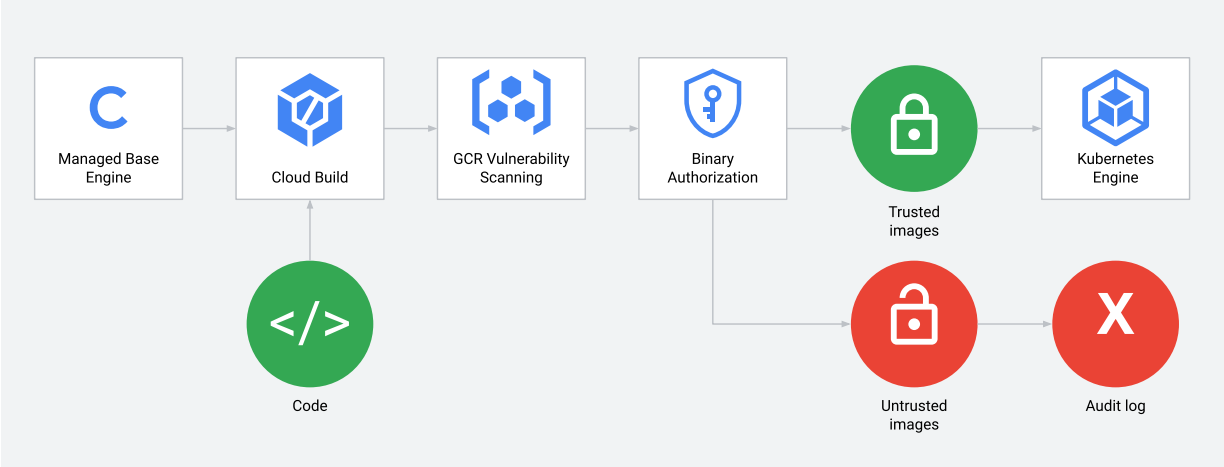
A central Git repository hosts a common configuration, which can cover access-control policies, resource quotas, and namespaces. Anthos Config Management evaluates each commit to the repository and then rolls out configuration changes to all clusters, so that the state you want is quickly reflected. A built-in validator checks for misconfigurations submitted to the repository to prevent the pushing of bad configurations. ACM prevents configuration drift with continuous monitoring of each cluster's state by using the declarative model to apply policies that enforce compliance.

By automating and scaling policy creation, rollout, audit, and enforcement continuously across all Anthos environments, your developers can execute as fast as the business requires while staying within the guardrails put in place by security.

Deploying only trusted workloads

Regardless of environment, you need to know that the container images you deploy are trusted. Arbitrary public container images can include unpatched vulnerabilities, or even embedded, malicious code that can expose your enterprise to preventable attacks.

The following diagram illustrates how your enterprise organization can modernize your app security approach.



Source

You build container images by taking a minimal OS base image and adding the packages, libraries, and binaries that you need for your app. You can reduce the surface of attack of these container images by doing the following:

- Use images purpose-built for containers.
- Make sure that images are up-to-date with the latest available patches
- Use deploy-time checks for supply chain integrity.

Anthos helps guard against vulnerable container images with a defense-in-depth approach. GCP provides managed base images that you can use to build container images. Managed base images are built reproducibly and are patched automatically when patches are available upstream. GCP also provides distroless images, a more minimal alternative to managed base images. When you build container images with distroless images as their base image, it contains only your app and its runtime dependencies, greatly reducing the potential attack surface.

When you use Anthos, you benefit from the native vulnerability scanning capabilities of Container Registry. Container Registry vulnerability scanning looks for known vulnerabilities (based off the Common Vulnerability and Exposures (CVE) database). Having knowledge of image vulnerabilities prior to deployment, enables your developers and operators to prevent patchable and potentially high-risk images from being deployed into production. The scanning results show the severity (based off the Common Vulnerability Scoring System (CVSS) score), availability of a fix, and the name of the package that contains the vulnerability.

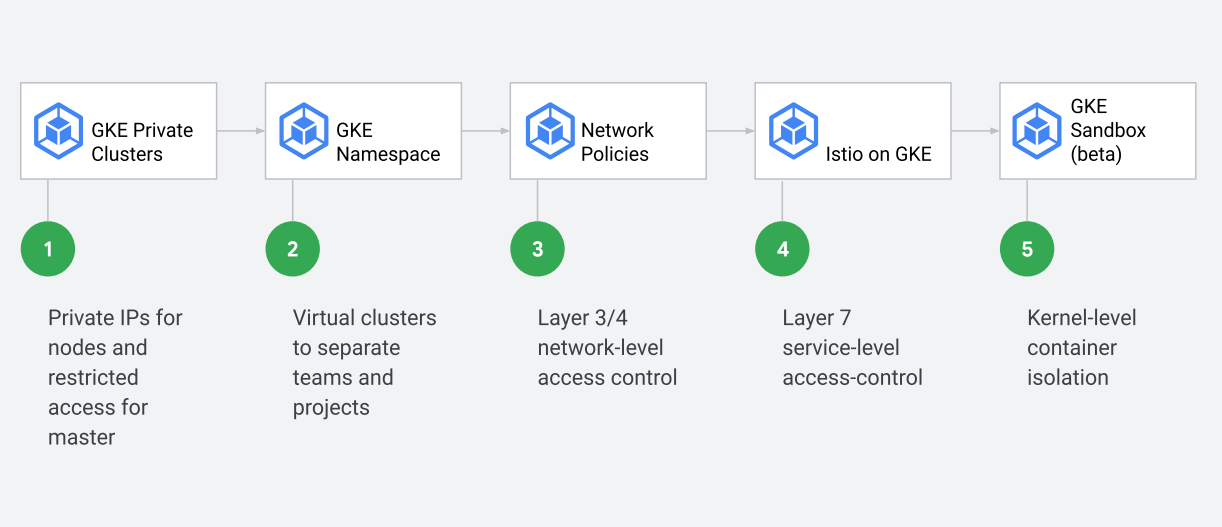
By using Anthos, you can also benefit from Binary Authorization, a deploy-time control that lets your organization define the requirements for a container image that you want to deploy, and stop deployment if an image doesn't meet your requirements. While each organization has different definitions of what constitutes a trusted image, common requirements are vulnerability scanning, verification of a legitimate build, and review by the quality assurance (QA) team. Binary Authorization integrates into many popular CI/CD tools, and uses cryptographic attestations to verify requirements are met prior to deploying an image.

With tools such as managed base images, Container Registry vulnerability scanning, and Binary Authorization, you can "shift security left" by building defined security checks into the development process, and making security a part of the app lifecycle.

Isolating workloads with different risk profiles

To gain resource efficiency, containers with different risk profiles can share the same host kernel or cluster of machine nodes. You need to isolate and segment your apps with different risk profiles running on this shared infrastructure so that only authorized services can communicate with each other and access intended resources.

The following diagram illustrates how Anthos delivers a full suite of security capabilities to isolate apps at multiple levels, including host, cluster, network, and service.



Source

- At the cluster level, you can deploy GKE private clusters, node clusters that aren't exposed to the public internet because they don't have public IP addresses. Access to the master is also restricted. You can authorize external networks to access the master.
- Within the cluster, you can separate teams and projects using namespaces, which have different Kubernetes identities provisioned and you can assign quotas for memory and CPU usage.
- At the network level, you can control access to the workloads and pods by using GKE network policies at layer 3 and 4. From a multi-tenancy perspective, this helps you to ensure that pods from different apps or different tenants aren't able to communicate.
- At the app level, you can use Anthos Service Mesh for service-to-service and end-user-to-service authorization at different levels of granularity, including namespace level, service level, and method level. (For additional reading on zero trust security and user access controls, see BeyondCorp.)
- At the host level, GKE Sandbox provides an extra layer of isolation between the host kernel and containers for sensitive or untrusted workloads. GKE Sandbox limits the host kernel surface area accessible to the app while still giving the app the ability to perform the system operations it needs. Anthos facilitates this modernization, enabling enterprises to evolve to a platform where automated security operations that deploy only trusted workloads, isolate workloads with different risk profiles, and enforce consistent policies across heterogeneous environments.

Conclusion

Modern app architectures have optimized how your enterprise can deliver new features to customers. While this modern app architecture presents a different set of security challenges, you can embrace this change to simultaneously modernize and automate your app security workflows.

Anthos facilitates this modernization, enabling you to move to a platform where automated security operations deploy only trusted workloads, isolate workloads with different risk profiles, and enforce consistent policies across heterogeneous environments.

Anthos facilitates this modernization, enabling enterprises to evolve to a platform where automated security operations that deploy only trusted workloads, isolate workloads with different risk profiles, and enforce consistent policies across heterogeneous environments.