Google Cloud Whitepaper

Anthos under the hood

The technologies that will transform enterprise applications

Google Cloud

Table of Contents

Introduction

Anthos: A modern platform for managing applications in today's hybrid and multi-cloud world

We introduced Anthos in 2018 to meet our customers wherever they were in their application modernization journey. Today, we're excited to see customers in diverse industries using Anthos to transform their business-critical application portfolios. Anthos was the first modern application platform to provide a consistent development and operations experience across different environments—multiple clouds, on-premises and edge. Since it became generally available, organizations are turning to Anthos to bring the benefits of cloud, containers and service mesh to their applications.

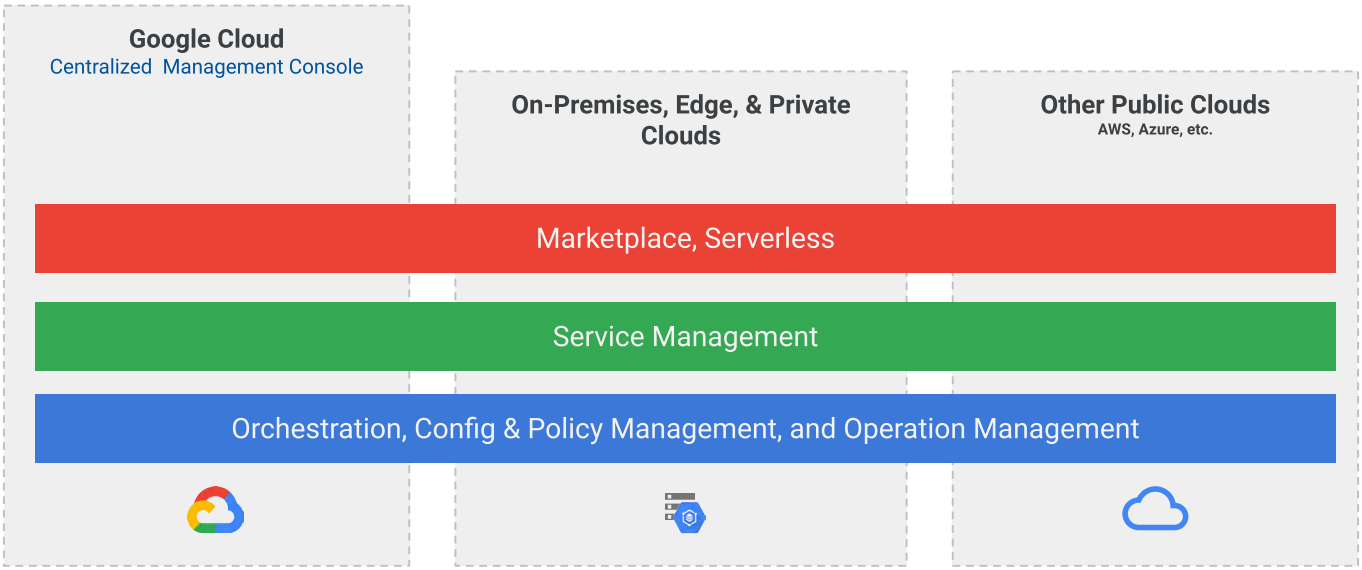
Anthos is how our customers want to deploy, manage and optimize all their applications—legacy as well as cloud native. With Anthos, your developers can write an application once, then run it anywhere. Anthos runs on an infrastructure layer that's been abstracted, and applications that run on it have access to high-value services that let them run efficiently and securely, without fear of lock-in or needless complexity. Anthos saves money by optimizing your infrastructure costs and reducing management overhead—wherever your applications may be. Better yet, Anthos provides this pathway to incrementally modernize existing applications without rewriting them, so you can realize immediate operational cost savings.

Most organizations want to adopt a multi-cloud approach out of a desire to avoid vendor lock-in or to take advantage of best-of-breed solutions, but managing applications across different clouds is easier said than done. To address these challenges, Anthos offers a unified model for computing, networking, and even service management across clouds and on-premises, managed through a centralized control plane. Anthos reduces business risk by simplifying multi-cloud management and makes it a reality for enterprise customers.

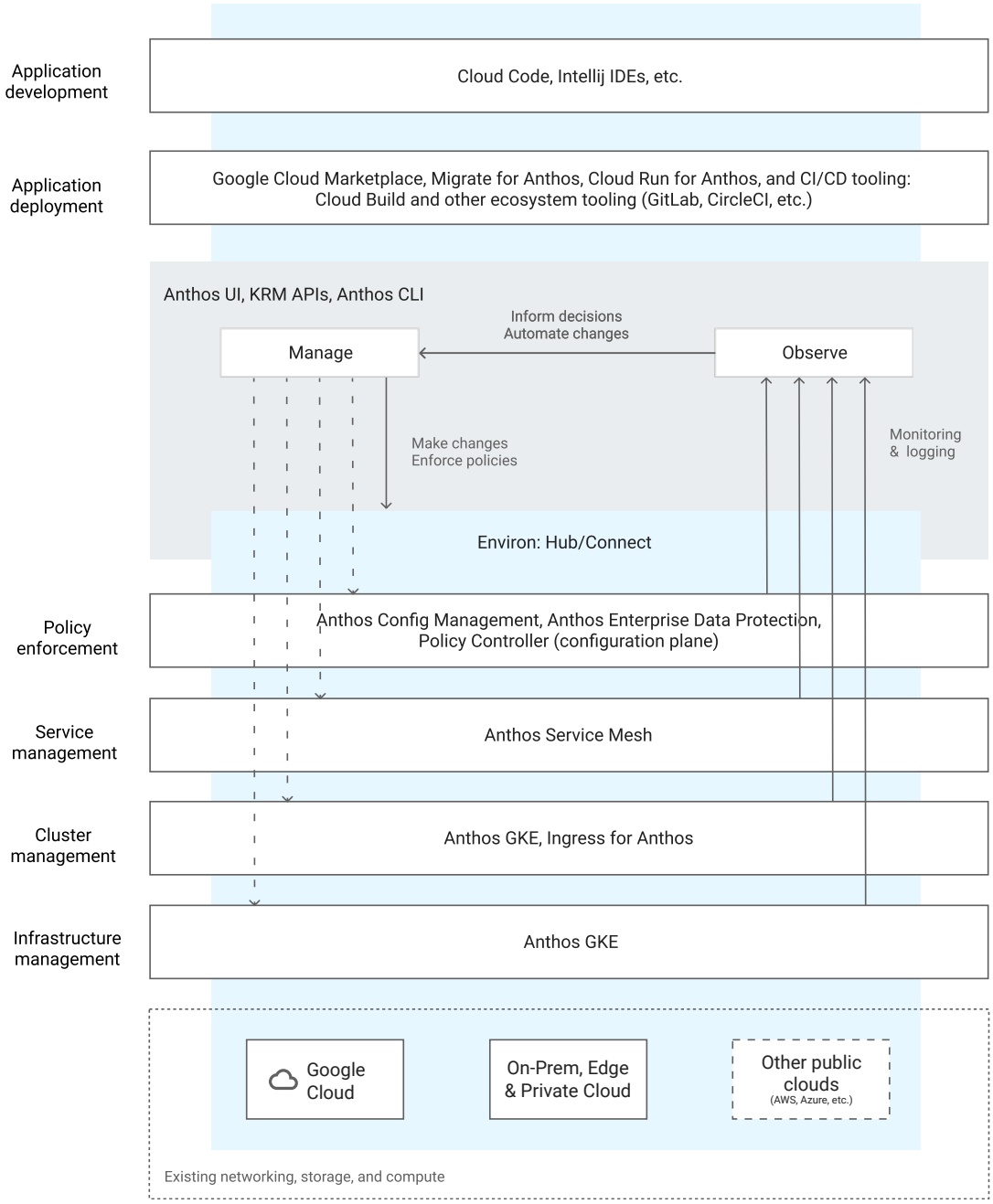
Customers tell us Anthos is how they want to deploy, manage and optimize all their applications—legacy as well as cloud native.

The Anthos platform consists of several key services: infrastructure management, container management and orchestration, service management and policy enforcement. Developers can also use integrated services to develop and deploy applications to Anthos environments, and operators can manage Anthos with the same tools they use to manage applications in other parts of Google Cloud.

Unified management with Anthos



The following diagram shows the Anthos components and their interactions in a typical enterprise environment.

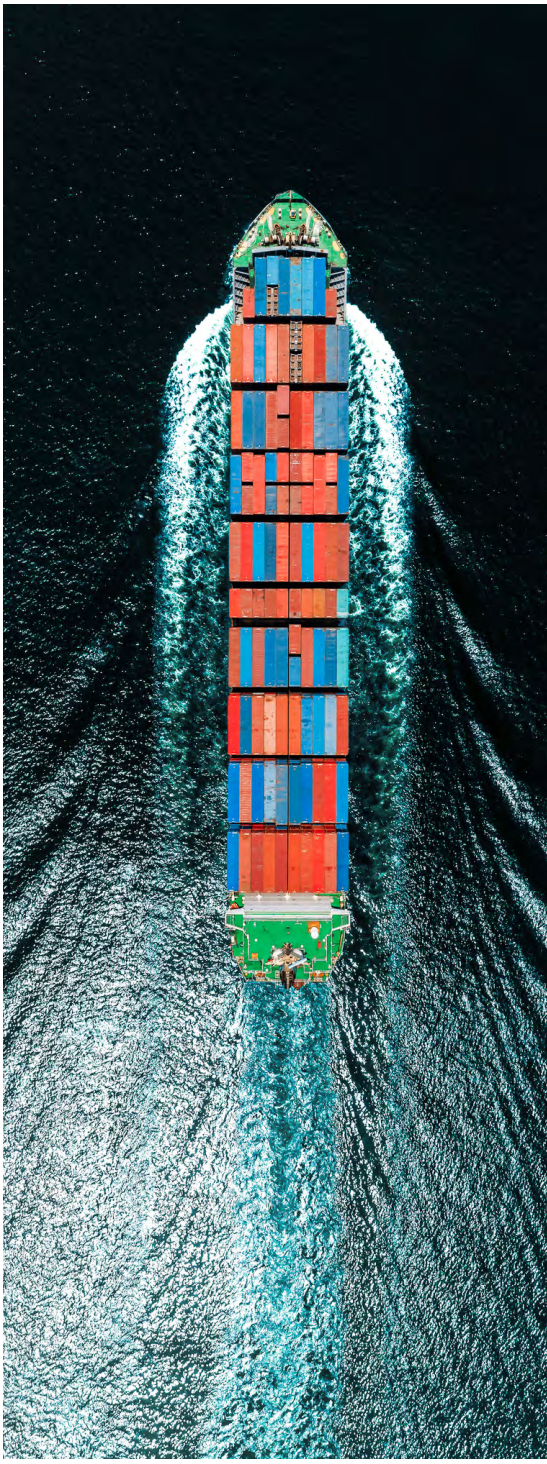


In this whitepaper, we'll provide an overview of each layer of the Anthos platform and how to leverage its features, so your organization can move forward with its goals of modernizing applications for the cloud era. Happy reading!

Chapter 1

Container orchestration and management with Anthos GKE

By Jay Smith and Sandeep Parikh



From Gmail to YouTube to Search, everything at Google runs in containers. We have been running containerized workloads in production over the past decade. In fact, our internal cluster management system, Borg, was so successful that we took what we learned and put it into the open source project Kubernetes, helping you benefit from our experience. Also known as "k8s," Kubernetes automates container orchestration, improving reliability and reducing the time and resources you need to spend on DevOps—not to mention the stress related to these tasks.

Google Cloud's managed Kubernetes distribution is of course Google Kubernetes Engine (GKE), and it's this same distribution that's the cornerstone of the Anthos platform.

Anthos GKE, part of Anthos, lets you take advantage of Kubernetes and cloud technology in your data center and in the cloud. You get a full GKE experience with quick, managed, and simple installs as well as upgrades validated by Google.

Under the hood, Anthos GKE is the same GKE that runs on Google Cloud. You can keep your environments in sync with the same Kubernetes version, OS, runtime, and add-ons between Anthos GKE deployed in your on-premises environment and in the cloud. You can also monitor, manage, and enforce policies across all your GKE clusters, whether in the cloud or on-prem, from the Google Cloud Console.

In addition to container orchestration, Anthos GKE offers differentiated security and networking capabilities out of the box. Let's take a deeper look.

Enterprise security capabilities

Whether deployed to the cloud or on-premises, Anthos GKE is built with enterprise security in mind. Anthos GKE builds upon and enhances Kubernetes with a number of critical security features needed to run mission-critical applications.

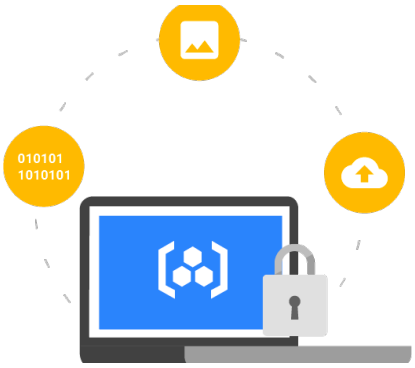
Anthos GKE on Google Cloud

Control plane security: In Anthos GKE, the Kubernetes master components are managed and maintained by Google. You can protect the Kubernetes API server by using master authorized networks and private clusters, which allow you to assign a private IP address to the master and disable access on the public IP address.

Node security: Anthos GKE deploys your workloads on Compute Engine instances in your Google Cloud project. Each instance uses Google's Container-Optimized OS as the operating system on which to run Kubernetes and its components. By default, Container-Optimized OS implements a locked-down firewall, a read-only filesystem, and limited user accounts (with root disabled). For stronger isolation in multi-tenant deployment scenarios, you can enable GKE Sandbox on your cluster to isolate untrusted workloads in sandboxes on the node.

Network security: Anthos GKE leverages a powerful software-defined network that enables simple Pod-to-Pod communications within a Kubernetes cluster, and within the cluster's VPC. Using Network Policies you can lock down the ingress and egress connections created to and from the Pods in a namespace. As part of your creation of clusters and/or namespaces, you can apply the default to deny traffic to both ingress and egress of every Pod, to ensure that all new workloads added to the cluster explicitly authorize the traffic they require. Finally, you can apply filtering to incoming load-balanced traffic for services that require external access, by supplying whitelisted CIDR IP ranges.

Workload security: When running workloads using Anthos GKE, individual Pods and containers can be configured with limited privileges using Kubernetes Security Context, and administrators can enforce cluster-wide privilege limits using a PodSecurity Policy. Additionally, Anthos GKE applies default Docker AppArmor security policies to all Kubernetes Pods. Finally, Workload Identity for Anthos GKE aligns Kubernetes service accounts with Google Cloud service account permissions, for managing access to Google Cloud resources from individual Pods. Google Cloud service accounts are centrally managed using Cloud Identity and Access Management (IAM).



Audit logging: Audit logging provides a way for administrators to retain, query, process, and alert on events that occur in your Anthos GKE environments. Administrators can use the logged information to do forensic analysis, real-time alerting, or to catalog how a fleet of Anthos GKE clusters are being used and by whom. By default, Anthos GKE logs Admin Activity logs. You also have the option to log Data Access events, depending on the types of operations you are interested in inspecting.

Anthos GKE deployed on-prem

Node security: Anthos GKE deployed on-prem runs your workloads in VMware instances, which are attached to your clusters as nodes. Google uses an optimized version of Ubuntu Linux to run the control plane and nodes in Anthos GKE. This distribution has been optimized for the cloud to use modern standards such as automatic security updates to the kernel, enhanced packages, and limited user access.

Logging: Kubernetes auditing provides a way for administrators to retain, query, process, and alert on events that occur in your Anthos GKE environments. Administrators can use the logged information to do forensic analysis, real-time alerting, or to catalog how a fleet of GKE clusters is being used and by whom.

By default, Anthos GKE logs admin activity. You can also log data access events, depending on the types of operations you want to inspect. The Connect Agent only talks to the local API server running on-premises, and each cluster should have its own set of audit logs. All actions that users perform from the UI through Connect are logged by that cluster.

Encryption: Google Cloud Key Management Service (Cloud KMS) is a cloud-hosted key management service that lets you manage cryptographic keys for your services. You can generate, use, rotate, and destroy AES256, RSA 2048, RSA 3072, RSA 4096, EC P256, and EC P384 cryptographic keys. Cloud KMS is integrated with Cloud IAM and Cloud Audit Logs so that you can manage permissions on individual keys and monitor how these are used. You can use Cloud KMS to protect secrets and other sensitive data that you need to store.

Connectivity: If your Anthos GKE deployed on-prem clusters and workloads securely connect to Google Cloud services through Cloud VPN, you can use Cloud KMS for key management. Otherwise, you can use alternatives such as Kubernetes Secrets, HashiCorp Vault, or a hardware security module.

Anthos GKE leverages a powerful software-defined network that enables simple Pod-to-Pod communications within a Kubernetes cluster, and within the cluster's VPC

Platform networking integrations

Whether deployed to cloud or on-premises, Anthos GKE is configured out of the box with a series of networking improvements that extend Kubernetes networking primitives with environment-specific integrations.

Anthos GKE on Google Cloud

Anthos GKE deployments running in Google Cloud are able to leverage Google's advanced software defined network (SDN), which enables packet routing and forwarding for Pods, Services, and nodes across different zones in the same regional cluster. Anthos GKE is also capable of dynamically configuring IP filtering rules, routing tables, and firewall rules on each Kubernetes node.

Every Anthos GKE node has an IP address assigned from the cluster's Virtual Private Cloud (VPC) network. This node IP provides connectivity from control components like kube-proxy and the kubelet to the Kubernetes API server. This IP address is the node's connection to the rest of the cluster. Each node has a pool of IP addresses that is assigned to Pods running on that node. Each Pod has a single IP address assigned from the IP range of its node. This IP address is shared by all containers running within the Pod, and connects them to other Pods running in the cluster. Each Service has an IP address, called the ClusterIP, assigned from the cluster's VPC network. You can optionally customize the VPC network when you create the cluster.

Anthos GKE provides three different types of Cloud Load Balancing load balancers to control access and to spread incoming traffic across your cluster as evenly as possible. You can configure one Service to use multiple types of load balancers simultaneously.

Anthos GKE provides three different types of load balancers to control access and to spread incoming traffic across your cluster as evenly as possible.

- External load balancers manage traffic coming from outside the cluster and outside your Google Cloud VPC. They use forwarding rules associated with the Google Cloud network to route traffic to a Kubernetes node.
- Internal load balancers manage traffic coming from within the same VPC network. Like external load balancers, they use forwarding rules associated with the Google Cloud network to route traffic to a Kubernetes node.
- HTTP(S) load balancers are specialized external load balancers used for HTTP(S) traffic. They use an Ingress resource rather than a forwarding rule to route traffic to a Kubernetes node.

When traffic reaches an Anthos GKE node, it is handled the same way, regardless of the type of load balancer. The load balancer is not aware of which nodes in the cluster are running Pods for its Service. Instead, it balances traffic across all nodes in the cluster, even those not running a relevant Pod. In a regional cluster, the load is spread across all nodes in all zones of the cluster's region. When traffic is routed to a node, the node routes the traffic to a Pod, which may be running on the same node or a different node.

Anthos GKE deployed on-prem

Anthos GKE deployed on-premises uses an *Island Mode* configuration in which Pods can directly talk to each other within a cluster, but cannot be reached from outside the cluster. This configuration forms an "island" within the network that is not connected to the external network. Clusters form a full node-to-node mesh across the cluster nodes, allowing a Pod to reach other Pods within the cluster directly.

All egress traffic from the Pod to targets outside the cluster originates from the Pod using the host node's IP address. Anthos GKE deployed on-prem includes an L7 load balancer with an Envoy-based ingress controller that handles Ingress object rules for ClusterIP Services deployed within the cluster. The ingress controller itself is exposed as a NodePort Service in the cluster.

Anthos GKE deployed on-premises uses an *Island Mode* configuration in which Pods can directly talk to each other within a cluster, but cannot be reached from outside the cluster.

Anthos GKE deployed on-prem includes a built-in L4 load balancer as well as provides support for external F5 Networks L3/L4 load balancers. The installation configures a virtual IP address (VIP) (with port 80 and 443) on the load balancer. The VIP points to the ports in the NodePort Service for the ingress controller. This is how external clients can access services in the cluster. When using an external F5 load balancer, user clusters running Services set to type LoadBalancer must configure the loadBalancerIP field to use the above-configured VIP.

As an alternative to using the built-in or F5 load balancer integrations, you can enable manual load balancing mode. If you choose to use manual load balancing, you cannot run Services of type LoadBalancer. Instead, you can create Services of type NodePort and manually configure your load balancer to use them as backends. Also, you can expose Services to outside clients by using an Ingress object.

Anthos GKE meets the needs of enterprise workloads

Anthos GKE includes a number of networking and security enhancements to support mission-critical enterprise workloads. Anthos GKE builds upon Kubernetes primitives and integrates them with Google Cloud systems, such as Cloud IAM. Whether deployed in the cloud or on-prem, Anthos GKE benefits from Google's expertise at running large-scale distributed systems. Add to that industry-first four-way autoscaling, a financially backed SLA, and flexible release channels, and Anthos GKE provides the foundation you need to build highly reliable services.

Chapter 2

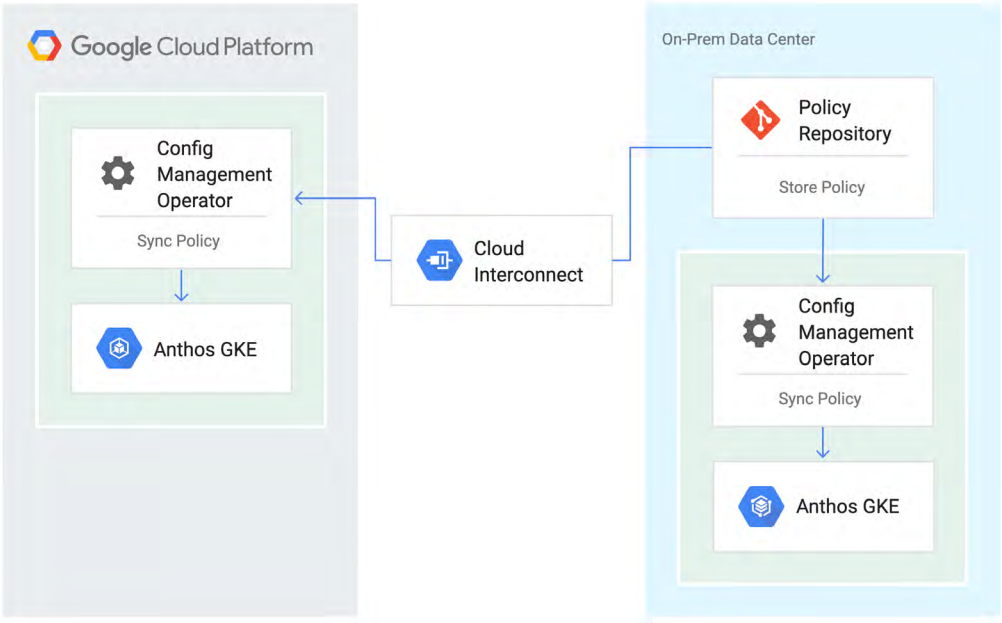
Set your policy with Anthos Config Management

By Sandeep Parikh

As the number of Anthos GKE clusters begins to grow, delivering individual configuration changes for each deployment leads to overhead and management challenges. Anthos Config Management is a core component of the Anthos stack and provides platform, service, and security operators with a single, unified approach to multi-cluster management that spans both on-premises and cloud environments. Specifically, Anthos Config Management allows operators to create and apply common configurations and policies at scale across all of your Anthos GKE clusters. Let's take a deeper look at how we architected Anthos Config Management, and the benefits that it brings to the table.

Anthos Config Management architecture

Anthos Config Management builds on modern software tools and practices by leveraging a central Git repository to manage access-control policies like role-based access control (RBAC), resource quotas, and namespaces across environments. As configuration and policy changes are committed to a Git repository, Anthos Config Management evaluates those changes and rolls them out to the Anthos GKE clusters in your deployments. Anthos Config Management follows Kubernetes best practices, favoring declarative approaches over imperative operations, and continuously monitors cluster state and applies the desired state as defined in Git.



As shown in the above diagram, Anthos Config Management is deployed as a custom controller in each of your Anthos GKE clusters and includes three key components:

1. An importer that reads from a central Git repository
2. A component to synchronize and hydrate stored configuration data into Kubernetes objects
3. A component that monitors for drift between stored and active cluster configurations and reconciles them as needed.

The central Git configuration and policy repository can be hosted on-premises, in Google Cloud, or using any hosted Git provider (for example, GitLab or Github)—the only requirement is that the importer component have network connectivity to the Git repository.

How Anthos Config Management helps

Anthos Config Management allows cluster management to follow modern software development practices, making cluster configuration and policy changes auditable, revertable, and versionable. It also modernizes configuration management practices by incorporating several key Kubernetes-native features and capabilities. Building on Kubernetes primitives, Anthos Config Management can flexibly apply different configurations to groups of clusters or namespaces—for example, you can apply different quota levels to staging vs. production resources.

Enforce IT governance

Beyond configuration, Anthos Config Management also includes support for defining and enforcing custom rules not covered by native Kubernetes objects. The Anthos Config Management policy controller mechanism allows you to create guardrails that correspond to your organization's unique security, compliance, and governance requirements. These guardrails allow you to inspect updates to your Anthos infrastructure and reject changes that don't comply with your

The Anthos Config Management policy controller mechanism allows you to create guardrails that correspond to your organization's unique security, compliance, and governance requirements.

unique policies. For example, your organization may require that applications use specific network configurations or storage mechanisms. With the Anthos Config Management policy controller, you can help new teams get up and running quickly, knowing that their applications are following compliance best practices.

The Anthos Config Management policy controller is deployed as an additional component in your Anthos GKE clusters, and is based on the Open Policy Agent Gatekeeper project. The Anthos Config Management policy controller operates as a Kubernetes admission controller webhook, allowing it to transparently integrate with the Kubernetes API and evaluate Kubernetes objects as they are admitted into the cluster. It enforces guardrails by rejecting admission or auditing objects that violate predefined constraints, giving security operators a centralized way to monitor compliance. Guardrails are created using two objects: the constraint's rules are defined in a cluster-scoped template, and the enforcement of that constraint is scoped to a particular Kubernetes namespace and/or specific types of Kubernetes objects. For example, a guardrail can be created that enforces specific labeling guidelines for all Pods in the "default" namespace.

Unify resource management

Many cloud-native development teams work with a mix of configuration systems, APIs, and tools to manage their infrastructure. This mix is often difficult to understand, leading to reduced velocity and expensive mistakes. In addition to enforcing IT governance with its policy controller, Anthos Config Management includes support for Config Connector, which provides a method to configure many Google Cloud services and resources, such as Compute Engine VMs or Cloud Pub/Sub messaging, using Kubernetes tooling and APIs.

Config Connector provides a collection of Kubernetes Custom Resource Definitions (CRDs) and associated controllers. Config Connector creates Google Cloud resources when you configure and apply custom Objects to your cluster. Config Connector is a Kubernetes-native system, meaning it can leverage common

Kubernetes best practices such as storing and using sensitive data with Secrets, managing runtime configuration using ConfigMap objects, and following standard role-based access control approaches. Additionally, when combined with the Anthos Config Management policy controller, IT governance rules can be applied to the creation and ongoing management of Config Connector-driven resources.

As a core component of the Anthos stack, Anthos Config Management provides a modern approach to configuration and policy management and allows platform, service, and security operators to leverage a unified approach to managing multiple clusters in hybrid deployments. Anthos Config Management is capable of managing configuration and policies for Anthos GKE, as well as Anthos Service Mesh and Cloud Run for Anthos.

Chapter 3

Monitor and manage your services with Anthos Service Mesh

By Tony Pujals and Sandeep Parikh

An increasingly popular approach to application modernization is to decompose large “monolithic” applications—those written as a single logical executable—into independent microservices deployments, where functionality is decomposed into smaller, independent services that communicate via APIs.

But as these microservices deployments proliferate, there is often a sharp increase in the effort required to operate and scale those deployments. Anthos Service Mesh provides a suite of tools that help monitor and manage services of all shapes and sizes, whether running in cloud, hybrid, or multi-cloud environments.

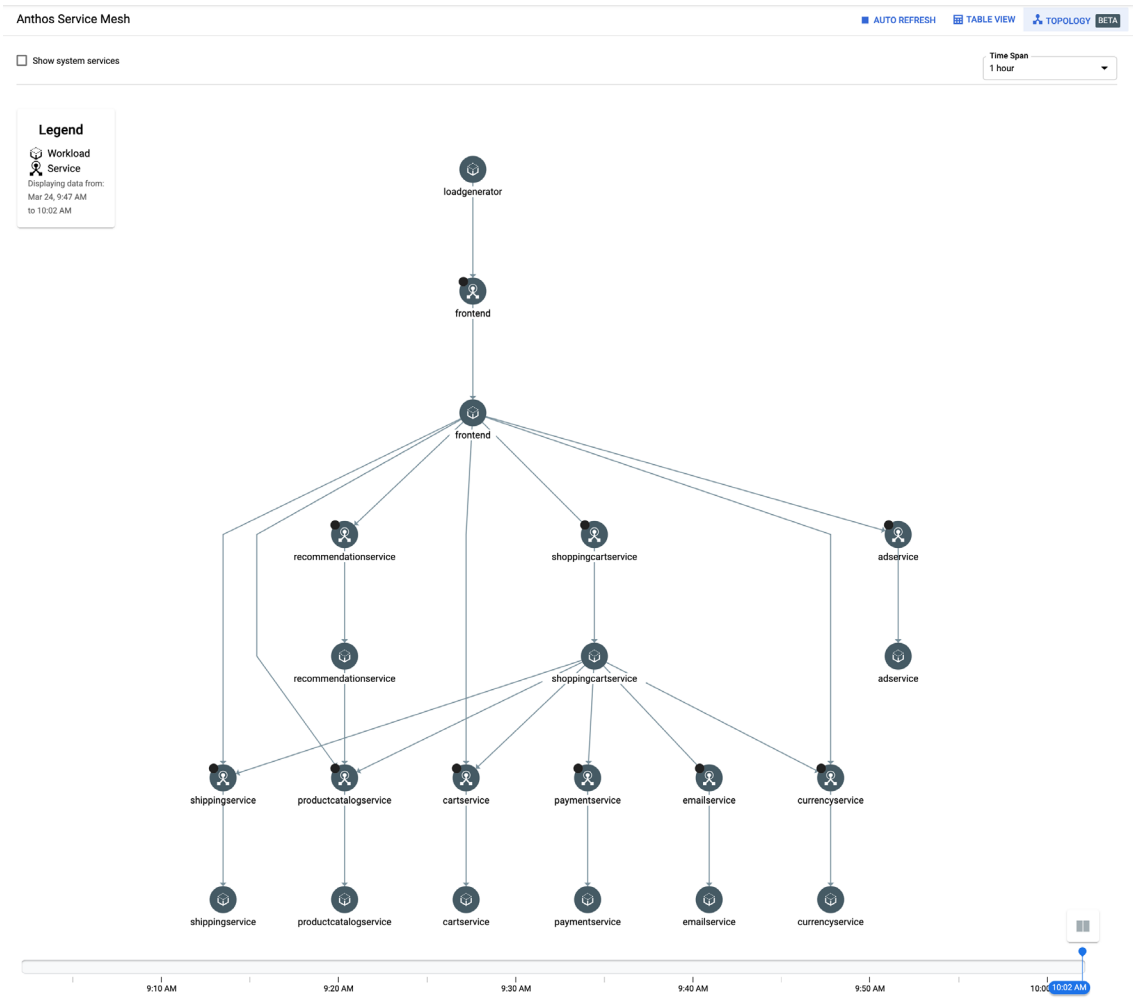
Anthos Service Mesh architecture

Anthos Service Mesh leverages the APIs and core components from Istio, a highly configurable and open-source service mesh platform, and builds upon them with fully managed service health, operational agility, and security mechanisms.

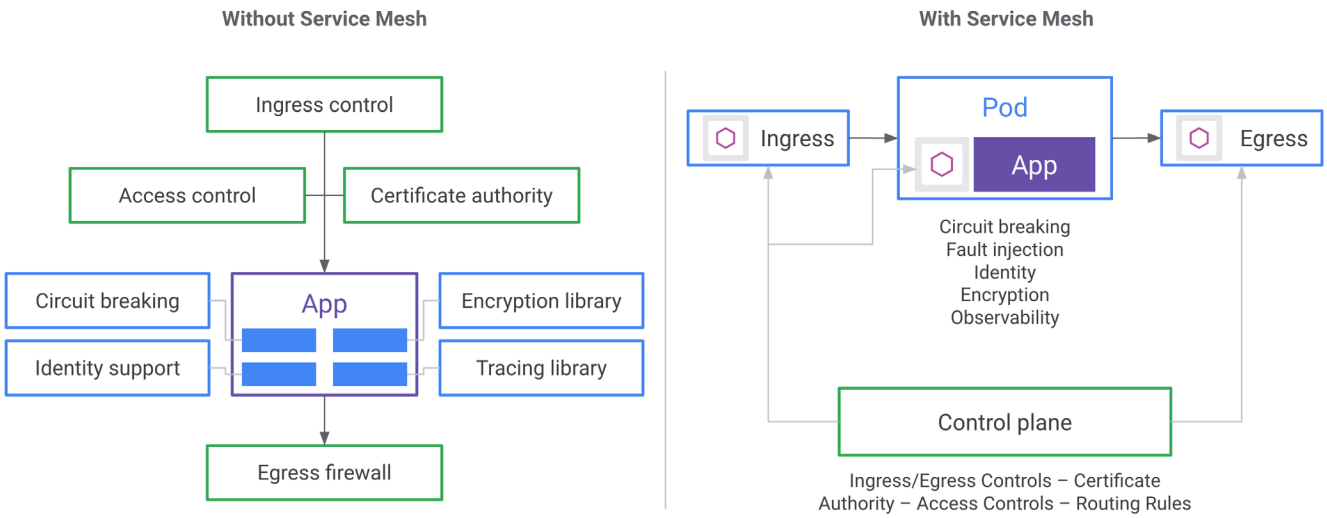
Like other service mesh platforms, Anthos Service Mesh relies on two main components: a data plane and a control plane. In Anthos Service Mesh deployments, the data plane is deployed as a set of distributed proxies that mediate all inbound and outbound network traffic between individual services. The proxies themselves are configured using a centralized control plane and an open API. This approach allows for broader automation of common networking tasks, like implementing traffic splitting or steering between services, or

enabling service-to-service authentication and encryption. When using Anthos Service Mesh, the control plane is operated as a fully managed offering outside of Anthos GKE clusters, simplifying management overhead and ensuring the highest possible availability.

The Anthos Service Mesh managed control plane, meanwhile, consists of three components. The first component, Traffic Director, Google Cloud's fully managed service mesh traffic control plane, is responsible for translating Istio API objects into configuration information for the distributed proxies, as well directing service mesh ingress and egress traffic. The second, Managed CA, is a centralized certificate authority responsible for providing SSL certificates to each of the distributed proxies, authentication information, and distributing secrets. The final component is Google Cloud's operations tooling (formerly Stackdriver), which provides a managed ingestion point for observability and telemetry, specifically monitoring, tracing, and logging data generated by each of the proxies. Additionally, this tooling powers the Anthos Service Mesh Observability dashboard, which allows service operators to visually inspect their services and service dependencies, as well as implement SRE best practices for monitoring SLIs and establishing SLOs.



Prior to adopting service mesh technologies, application teams were required to build in support for common management, networking, or security functionality, such as service-to-service authentication/authorization controls, encrypting service communications, or fine-grained traffic management. With service meshes, and Istio in particular, that work is offloaded from applications to the distributed proxies that form the data plane.



Anthos Service Mesh leverages the Istio APIs to deliver the following features for services deployed to Anthos GKE, or mixed deployments with container- and VM-based services (cloud or on-prem).

How Anthos Service Mesh helps

Taken together, the tools and technologies that make up Anthos Service Mesh deliver significant operational benefits to Anthos environments, with minimal additional overhead.

Uniform observability

As previously mentioned, the distributed proxy data plane in Anthos Service Mesh is responsible for mediating all inbound and outbound communication from every deployed service. While it proxies communications, the proxy reports service-to-service communication back to the control plane so it can generate a service dependency graph. The proxy also inspects traffic and inserts headers to facilitate distributed tracing. Finally, it captures and reports service logs and service-level metrics (i.e., latency, errors, availability). This is all accomplished without requiring deployed applications to integrate custom approaches for metrics, tracing, or logging data capture. For applications

or environments with custom observability integrations, the data captured by the proxies can also be forwarded to third-party observability systems.

Operational agility

Anthos Service Mesh provides fine-grained controls for managing the flow of inter-mesh (north-south) and intra-mesh (east-west) traffic. Traffic controls can also be integrated with ingress and egress traffic mechanisms, giving developers and service operators complete control over how traffic moves into, out of, and across their service mesh. These controls allow developers and operators to implement traffic controls such as:

- Traffic splitting across differing service versions for canary or A/B testing
- HTTP header-based traffic steering between individual services or versions
- Circuit breaking to prevent cascading failures
- Fault injection to help build resilient and fault-tolerant deployments

Traffic controls are implemented using Istio API objects and the distributed proxies are configured using Traffic Director. By default, each proxy maintains information about all possible inbound and outbound routes, along with routable upstream hosts and services within the service mesh. Traffic Director periodically delivers configuration data to each of the proxies to ensure that they are up to date and aware of other services in the mesh.

Policy-driven security

The goal of Anthos Service Mesh is to deliver turnkey security controls to services, reducing the need for custom service-specific integrations. Anthos Service Mesh provides the underlying secure communications channel, and manages authentication, authorization, and encryption of service communication. With Anthos Service Mesh, service communications are secured by default, letting you enforce

The goal of Anthos Service Mesh is to deliver turnkey security controls to services, reducing the need for custom service-specific integrations.

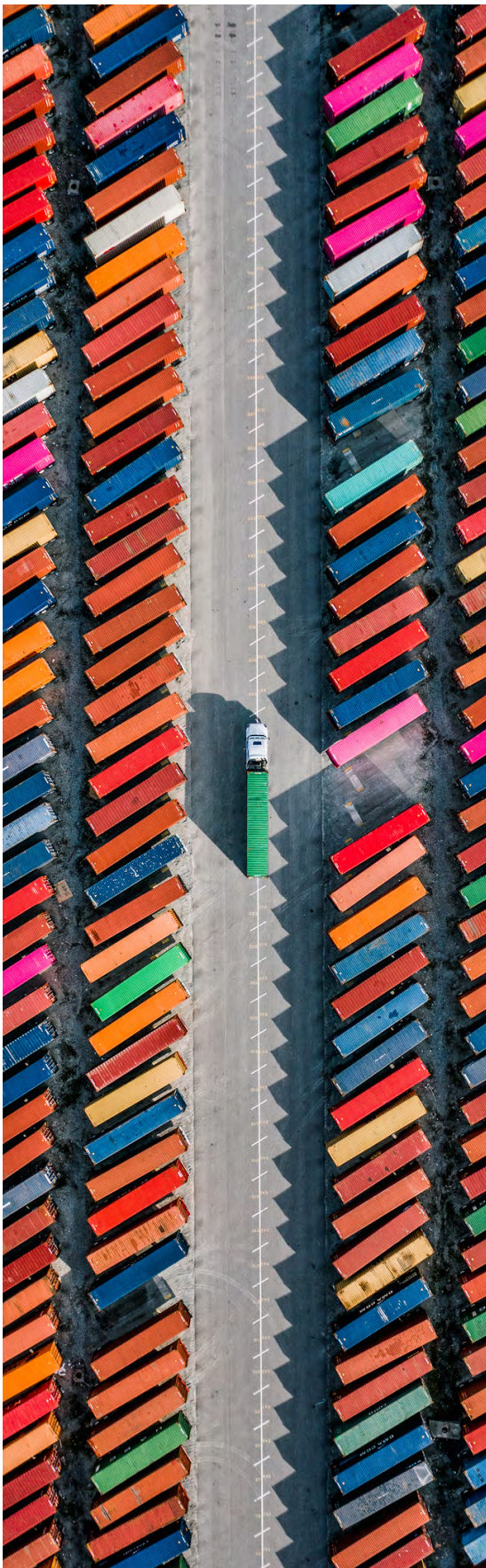
policies consistently across diverse protocols and runtimes. As with traffic and observability features, the distributed proxies are responsible for implementing each of Anthos Service Mesh's security features.

In Anthos Service Mesh, identity is a fundamental component—services exchange identity credentials to mutually authenticate. On the client side, the server's identity is checked against the secure naming information to see if it is authorized to run the service. On the server side, the server determines what information the client can access based on authorization policies, and audits service usage. This identity model provides significant flexibility and granularity for Anthos Service Mesh to understand end-user, single-service, or multi-service identities.

Combined with service identity, the Anthos Service Mesh managed CA provides a trusted root certificate authority that distributes the keys and certificates, required to operate Anthos Service Mesh's public key infrastructure, to each proxy. Managed CA is also responsible for rotating keys and certificates for each proxy prior to expiration. Using this infrastructure, services in Anthos Service Mesh are capable of using transport or origin authentication.

Origin authentication allows end-user (or device) identity to be verified by mesh-based services, and can be enabled at the individual-request level. Transport authentication is for service-to-service communication, which is tunneled through a mutually authenticated TLS connection between proxies to create an encrypted channel between services. For deployments where some services may not be part of the service mesh, mTLS authentication can be rolled out in an incremental fashion to provide a gradual onboarding process.

Anthos Service Mesh also provides access control mechanisms that are controlled using fine-grained authorization policies. These policies support mesh-, namespace-, and workload-level access control for services within the mesh. Each proxy enforces access controls prior to traffic hitting its intended destination service. When authorization policies are created, they contain details about which services are affected, and rules that define who (a list of sources) is allowed to do what (a list of operations), under which conditions. Sources are based



on service or end-user identity. The list of operations contains allowed HTTP methods and URL paths. The available conditions that can be incorporated into authorization policies support inspecting individual request metadata, or source/destination networking attributes (e.g., IP address, ports).

In short, Anthos Service Mesh is a critical component of the overall Anthos platform, delivering functionality that complements the foundational Kubernetes container orchestration layer, and giving you the agility, observability and security capabilities you need to build modern apps based on distributed, containerized microservices.

Chapter 4

Serverless anywhere with Cloud Run for Anthos

By Ahmet Alp Balkan

When people talk about serverless, they're often referring to functions-as-a-service or AWS Lambda-style workloads. At Google Cloud, we think of serverless more broadly: workloads that run on managed infrastructure where developers self-serve their needs. For example, Google App Engine has been bringing the serverless platform as a service experience to developers all around the world for well over a decade.

With the new Cloud Run platform, we bring the serverless experience to containers. Any HTTP application that runs in a container image can be run on Google's fully managed infrastructure. Google handles wiring up networking, autoscaling, domain names, TLS and many other aspects for you, so you don't have to worry about setting up and managing things like virtual machines, clusters, or load balancers.

Serverless experiences like Cloud Run empower development teams to serve their own needs in an agile way. For example, with a self-serve model, a data science team can deploy their machine learning prediction models and autoscale them without being a burden to their operations or platform teams.

In this section, we'll look at how Cloud Run for Anthos allows you to have high-level developer experiences and insulate your development teams from the underlying infrastructure while easing operations for running services for your operations teams. But first, let's take a look at some of the challenges of running modern apps on Kubernetes.

Cloud Run for Anthos allows you to have high-level developer experiences and insulate your development teams from the underlying infrastructure while easing operations for running services for your operations teams.

The challenge of operating microservices on Kubernetes

Kubernetes does an excellent job of operating a set of workloads on a set of machines with a declarative, state-driven model. However, Kubernetes doesn't know a lot about the specific needs of an application.

For example, as far as Kubernetes is concerned, a microservice application is just a container with some TCP ports. Therefore, Kubernetes falls short of going the last mile of making it easy to operate modern applications that follow the twelve-factor app methodology. Kubernetes, for instance, doesn't natively support "versions of an application," and so doesn't natively offer high-level features like canary deployments or clean rollbacks.

Similarly, Kubernetes networking and load balancing between services isn't aware of application-layer (Layer 7) networking protocols like HTTP or gRPC, so it can't split traffic between versions, enforce traffic policies, or autoscale applications based on request metrics.

Autoscaling microservices-based applications is also a non-trivial task on Kubernetes. Spiky traffic patterns can cause containers to fall over and drop requests, because its Horizontal Pod Autoscaler (HPA) acts only on a running average of CPU and memory metrics that are the result of the traffic pattern, thus do not reliably prevent the container from crashing in case of a traffic spike. Nor does Kubernetes offer a way to buffer requests until the requests can be served by an available container.

This is where serverless platforms shine: Give them your unit of deployment (function, application or container image) and the infrastructure runs and scales the application for you.

This is where serverless platforms shine: Give them your unit of deployment and the infrastructure runs and scales the application for you.

Serverless on Kubernetes: Open and extensible

Cloud Run for Anthos brings the serverless container experience to your Anthos clusters. Cloud Run for Anthos offers a high-level platform experience on top of Kubernetes clusters and building blocks, allowing your platform teams to build custom platforms on Kubernetes.

Cloud Run for Anthos is built with Knative, an open-source operator for Kubernetes that brings serverless application serving and eventing capabilities to your cluster. Knative was originally created by Google, with contributions from over 50 different companies.

Knative adds the missing high-level pieces to your Kubernetes clusters while being compatible with other Anthos components and Kubernetes tooling that you might choose to adopt. At the end of the day, Knative workloads are still Kubernetes workloads.

How Cloud Run for Anthos simplifies operations

For platform teams that want to offer developers additional tools to test, deploy and run applications, Knative provides a simple way to provide this enhanced developer experience on top of existing Kubernetes clusters.

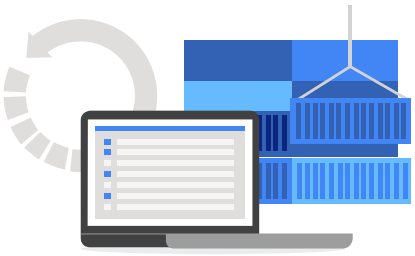
Let's explore the benefits that Cloud Run for Anthos brings to your Kubernetes environments.

Easy migration from Kubernetes Deployments

To run a microservice on Kubernetes, you need to configure Deployment, Service and HorizontalPodAutoscaler objects for a load balancer and autoscaling. In addition, you cannot easily change or roll these configurations back if your application is already serving traffic.

With Cloud Run for Anthos, you don't need to configure these features up front, and rollback is easy. And migrating to Cloud Run for Anthos is simple: Just take your Kubernetes Deployment, change a few lines of code, discard the HPA and Kubernetes Service objects, and you have a Knative Service manifest describing a microservices application that's autoscaled and load balanced.

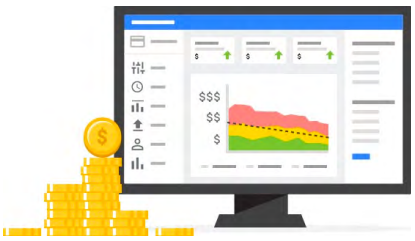
Autoscaling



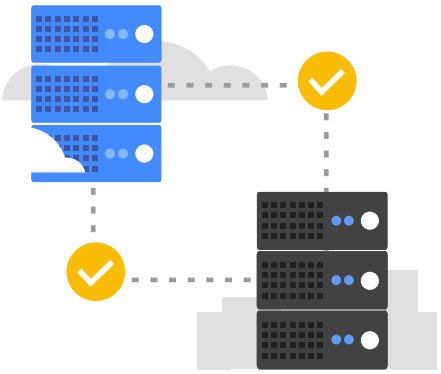
As described earlier, the Kubernetes HPA works based on CPU and memory metrics, which are often too slow and delayed to react to the needs of spiky microservices. A sudden traffic spike may cause your application containers on Kubernetes to crash (and drop the requests) as they would be overloaded while trying to serve the high volume of traffic.

Cloud Run for Anthos offers three high-level out-of-the-box autoscaling primitives for your applications that don't exist in Kubernetes natively:

- **Rapid, request-based autoscaling**: By default, all Knative applications come with an autoscaler that monitors request metrics. This lets Cloud Run for Anthos handle spiky traffic patterns smoothly.
- **Concurrency controls**: Knative lets you enforce concurrency limits (i.e., maximum in-flight requests per container), which ensures that a container does not become overloaded and crash. The requests are buffered until more containers are added to handle the spiky traffic.
- **Scale-to-zero**: If an application doesn't get any requests for a while—or is entirely inactive—Cloud Run scales it down to zero, to reduce its footprint on your cluster. Then, the first request that comes to the application waits until a container is created to handle the request. Alternatively, you can turn off scale-to-zero to prevent cold starts.



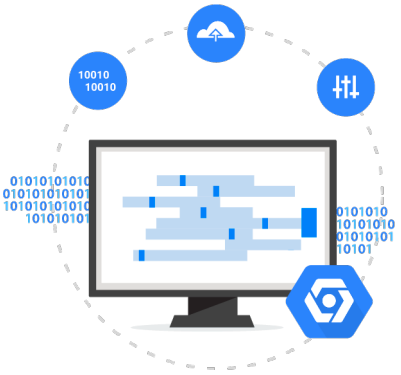
Networking



By default, Kubernetes does not have the notion of application-layer networking protocols such as HTTP or gRPC. This often causes traffic to be load balanced between replicas of a deployment unevenly, as Kubernetes only supports TCP load balancing.

Cloud Run for Anthos has built-in load balancing capabilities and policies for traffic splitting between multiple versions of an application. And because it has a deep understanding of requests, it can buffer each request during autoscaling and collect request-level metrics from applications out of the box.

Cloud Run for Anthos is compatible with Anthos Service Mesh, and any application deployed on Knative will take advantage of the capabilities of the service mesh, such as traffic policies, mutual TLS and telemetry.



Releases and rollouts

Cloud Run for Anthos also supports the notion of the Knative API's Revisions, which describe new versions or different configurations of your application. For example, changing the container image or increasing memory limits causes new revisions to be created.

Revisions are immutable, hence they allow you to cleanly roll back to a previous configuration. Knative also has the concept of pinning to container <image:tag> references, so that when you re-push a new image to an existing tag, your Knative application will not get the new image. This ensures that your Knative deployments are reproducible and can be rolled back cleanly.

Further, you can perform canary deployments by splitting traffic to an application using a high-level traffic configuration, e.g., send 90% traffic to Rev1, and 10% to Rev2. By sending only a small percentage of traffic to a new version of your application, you can test the impact of new features or changes to your users gradually.

Monitoring

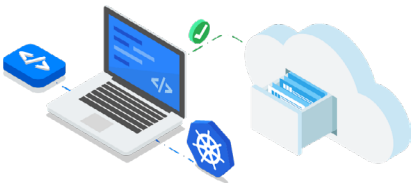
As mentioned in the Networking section, Cloud Run for Anthos is aware of all requests that come to your applications. Therefore, it can observe and record golden star metrics such as latency, error rate and requests per second.

These metrics are automatically collected and sent to Google Cloud monitoring and operations tools, without any configuration, and they can help you troubleshoot and observe discrepancies between different versions of your application. For example, you can track and measure and observe an internal service-level objective (SLO) such as 99.5% request latency for a service, and see how it behaves on different revisions of your application.

Cloud Run for Anthos has built-in load balancing capabilities and policies for traffic splitting between multiple versions of an application.

Cloud Run for Anthos use cases

Cloud Run for Anthos is suitable for running stateless applications, as it doesn't support stateful applications or sticky sessions. Some examples of applications that run great on Cloud Run for Anthos include:



- Microservices, web front ends, API gateways, API middleware.
- Event handlers, ETL: where the application reads an event payload pushed to it and processes it.
- Machine learning model prediction: such as TensorFlow serving containers.

Overall, Cloud Run for Anthos simplifies the operations and management burden for applications in service-oriented architecture (SOA) or microservices architecture.

Chapter 5

Developing applications for Anthos

By Sandeep Parikh

Developing applications for Kubernetes often means having to integrate different toolchains and build systems up front, along with security and deployment mechanisms. Integrating these systems can be complicated. Further, the resulting systems can be complex and brittle, prone to breaking. To build applications on Anthos, Google Cloud offers a well-integrated suite of tools, helping to accelerate the development workflow.

Coding applications for Kubernetes

Developing applications starts with code that goes through a constant cycle of writing, running, and debugging. To facilitate the development of modern Kubernetes-based applications, Google created Cloud Code, a set of tools to help you write, run, and debug. Cloud Code is available as extensions to popular integrated development environments (IDEs) like Visual Studio Code and IntelliJ, providing built-in support for rapid iteration, debugging, and running applications in development and production Kubernetes environments.

In fact, Cloud Code supports the full development cycle for Kubernetes-based applications, from creating a cluster for development and testing to running finished applications—again, all from the developer's preferred IDE. It also includes run-ready samples, out-of-the-box configuration snippets, and a tailored debugging experience. For local development, Cloud Code uses tools like Skaffold, Jib, and kubectl to automate development tasks and provide continuous feedback. And for production deployments, Cloud Code provides pre-built Skaffold profiles and leverages Kustomize for managing environment-specific resources. Finally, when debugging applications, Cloud Code lets developers use IDE-native debugging tools and examine application logs, whether applications are running locally or remotely.

Cloud Code supports the full cycle of developing Kubernetes applications, from creating a cluster for development and testing to running finished applications.

Creating build artifacts

In addition to helping developers create Kubernetes-based applications with Cloud Code, Anthos provides direct integration for building applications and packaging them into container images using Cloud Build for Anthos, a system that executes software builds in your Anthos environments. Cloud Build can import source code from Google Cloud Storage, Cloud Source Repositories, GitHub, or GitLab, execute a build to your specifications, and produce artifacts such as Docker containers or Java archives.

Cloud Build executes your build as a series of build steps, where each build step is run in a Docker container. A build step can do anything that can be done from a container, irrespective of the environment. To perform build workflows, you can either use the supported build steps provided by Cloud Build or write custom build steps. Container images built using Cloud Build are stored in Google Container Registry, while other artifacts such as binaries can be stored in Google Cloud Storage buckets or any third-party repository. Cloud Build requests can be executed manually or triggered automatically, based on source changes.

Securing your software

To help you develop and run applications securely, Google Cloud provides tools to incorporate security best practices at every stage of the development process. Prior to submitting your build requests, you can configure the following access and management mechanisms provided by Cloud Build:

- Cloud Identity and Access Management (IAM) controls the management permissions for creating, viewing, and canceling builds requests
- Cloud Audit Logs, for post-hoc analysis of build requests

- Cloud Key Management Service, to allow the use of encrypted resources in build requests

At build time, you can also use Cloud Build with Binary Authorization—a security control that ensures only trusted container images are deployed on Anthos GKE clusters. Binary Authorization works by integrating attestation into the container image building process where images are digitally signed based on their unique digest. Binary Authorization can also be extended to support other use cases, such as build verification or vulnerability scanning. In build verification scenarios, Binary Authorization can verify that the container image was built by a specific build system. For vulnerability scanning scenarios, Binary Authorization can also integrate with Container Analysis to ensure that any identified vulnerabilities are addressed prior to signing the container images. At deploy time, the Binary Authorization enforcer uses an attestor to verify the digital signature. In this way, only container images with verified attestations are allowed to deploy.

Anthos provides direct integration for building applications and packaging them into container images using Cloud Build for Anthos.

Running at scale

When it comes to making developers more productive, being able to quickly deploy and run applications at scale is critically important. While there are lots of tools to help package and deploy applications for Kubernetes (such as Helm), they require custom integrations with third-party tools. As discussed above, Cloud Code makes it easy to go from running locally, testing remotely, and finally deploying to production Anthos environments. Using profiles for each environment, Cloud Code provides built-in tools to orchestrate secure application deployment via Cloud Build workflows. It also lets you create profiles that support building container images using Cloud Build, which are then deployed to Anthos GKE using Cloud Run (see Chapter 4 for more information about Cloud Run). All told, this combination of Cloud Code, Cloud Build, and Cloud Run represents an integrated suite of tools to accelerate the modern application development workflow for your Anthos environment.

Chapter 6

Architecting a secure software platform with Anthos and GitOps

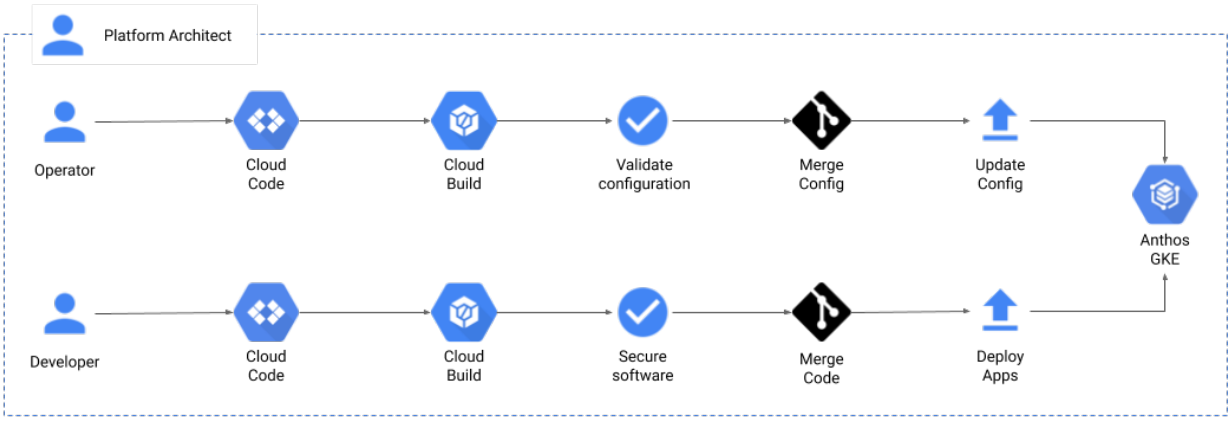
By Sandeep Parikh

Today's platform architects have a common goal: to build a modern and secure platform on which to run their organization's applications. To be successful, that platform must provide appropriate levels of abstractions for two groups of users: developers and operators. Both groups need the ability to ship application or configuration updates quickly and efficiently, while adhering to their organization's governance and operational requirements. It's up to the platform architect to provide those teams the tools they need to perform their core tasks, while ensuring that they can meet an acceptable security posture and service level objectives (SLOs).

The good news is that, out of the box, Anthos provides the building blocks that a platform architect needs to build just that kind of secure software platform. Anthos also lends itself well to GitOps, which applies version control to infrastructure and governance rules using standard Git best practices for managing and merging in configuration changes.

For platform architects, the first thing to do is to leverage Anthos Config Management as a mechanism to define and deliver infrastructure and governance as code. Using the GitOps model, operators can use Anthos Config Management to apply version control to infrastructure and governance rules. In production deployments, platform architects can integrate Anthos Config Management with Cloud Build to help operators ensure that configuration changes are validated against existing policies, so as not to introduce changes that break infrastructure or governance.

Developers building and delivering applications also need the GitOps application development model to be integrated with their Anthos environment. For enterprise software deployments, platform architects can integrate Git repositories with Cloud Build to manage deployments to staging environments and promote them to production environments. Cloud Build also integrates with Anthos Config Management to validate application/deployment changes against existing policies, ensuring that any governance violations are detected at build time. With these integrations in place, developers can deploy software without violating existing guardrails and operate in a self-service manner.



For both operators and developers, this integrated continuous integration (CI) approach shifts the burden of following security guardrails and governance best practices to the “left”—any breaking changes are identified well in advance of deployment, mitigating risk to production environments.

Anthos also provides security systems that enforce deploy-time security policies for container images. With Binary Authorization, platform architects can design a system that can automatically and digitally check each component of the software supply chain, ensuring the quality and integrity of software before an application is deployed to production environments (see ‘Securing your software’ in Chaper 5).

In summary, you can architect a modern and secure software delivery platform using Anthos and by leveraging GitOps best practices for operators and developers, integrating security guardrail and governance checks with CI tools, and using tools like Binary Authorization to ensure that safe container images are deployed to production.

Chapter 7

Anthos integrated services

By Arun Ananthampalayam

There's a whole lot more to Anthos than the core components listed above. Here's a quick look at how Anthos fits with other Google Cloud offerings.

Integrating with the Google Cloud portfolio

Over time, Google Cloud's goal is to make Anthos work seamlessly with the entire product suite. Here's a sampling of other current integrations not mentioned earlier.

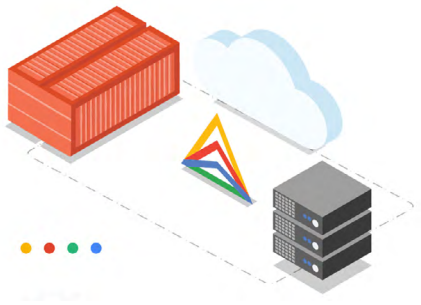
Operations tools

Google Cloud offers a complete operations suite designed to monitor, troubleshoot, and improve cloud infrastructure, software, and application performance. Formerly known as Stackdriver, these tools let you efficiently build and run workloads, so your applications perform well and stay available. Specifically, these tools let you:

- Collect signals across Google Cloud internal and external apps, platforms, and services
- Analyze and monitor your operational telemetry
- Set up appropriate performance and availability indicators
- Use built-in observability to troubleshoot and improve your applications
- Automate ops using both out-of-the-box tools and tools customized through programmatic interfaces

Google Cloud offers a complete operations suite designed to monitor, troubleshoot, and improve cloud infrastructure, software, and application performance.

Google Cloud Marketplace for Anthos



Develop faster and run anywhere with enterprise-ready containerized applications featuring prebuilt deployment templates and consolidated billing. Kubernetes applications are enterprise-ready containerized solutions with prebuilt deployment templates, featuring portability, simplified licensing, and consolidated billing. They can be run on Anthos, in the cloud, on-premises, or on Kubernetes clusters hosted in other environments. These are not just container images, but open-source, Google-built, and commercial applications that increase developer productivity, available now on Google Cloud Marketplace.

Migrate for Anthos

Use Migrate for Anthos to move and convert workloads directly into containers in Google Kubernetes Engine (GKE). Target workloads can include physical servers and VMs running on-premises, in Compute Engine, or in other clouds, giving you the flexibility to transform your existing infrastructure with ease.

Complete Kubernetes API support

Anthos GKE supports the standard Kubernetes API and also comes with additional CRDs such as the Cluster API. Users will be able to have the same open-source Kubernetes experience with a few additional APIs to help manage and scale at an enterprise level.

Chapter 8

The Anthos partner ecosystem

By Nima Badiey and Arun Ananthampalayam

Many of our customers have existing software and infrastructure investments, yet still want the freedom to invest in their cloud future with Anthos. We're working closely with our ecosystem of partners to offer innovative solutions that leverage Google Cloud's industry-leading open-source technologies. We have hardware, software and system integration partners ready to help customers leverage Anthos right out of the gate.

Consulting, MSP and system integration partners

Consulting, managed service provider and system integration partners can configure, install, manage and operate Anthos. System integrators including Accenture, Arctiq, Atos, Cognizant, Deloitte, HCL, IGNW, SADA, SoftServe, Wipro, WWT, and others can also help you modernize and extend your applications through services and solutions to help you to incorporate Anthos into your environment.

Google Cloud and partners are committed to meeting customers where they are, and providing them with the ability to run key workloads and applications in the environment best suited for their business.

Channel services

Customers can acquire and install Anthos through their preferred channel and reseller partners if they wish to maintain their existing unified billing and service options.

Embedded and integrated solutions

Google works closely with select partners to embed and integrate key technologies for a "batteries included" experience for customers. Anthos ships with the Canonical Ubuntu operating system included, and is preconfigured to work with F5 BIG-IP load balancers for on-premises installations. Installation guidance for customers that want to configure their prefered partner solutions such as Citrix load balancers is also available.

Hardware infrastructure and platform partners

Customers can leverage their existing on-premises hardware solutions and data center infrastructure with partners who have validated Anthos on their solution stack. Partners such as Cisco, Dell EMC, HPE, Intel, and Lenovo have committed to delivering Anthos on their own hyperconverged infrastructure. By validating Anthos on their solution stacks, our mutual customers can choose hardware based on their storage, memory, and performance needs.

Hardware platforms partners have published reference architectures and can support customers running Anthos on their solutions, including:

- Atos (BullSequana)
- Cisco (HyperFlex)
- Dell EMC (VxFlex integrated rack)
- HPE (SimpliVity, Nimble Storage, ProLiant, Synergy, 3PAR)
- Intel (Select Solutions)
- Lenovo (ThinkAgile VX)
- NetApp HCI

In addition, HPE GreenLake is an authorized reseller and distribution of Anthos-certified products and is fully supported by HPE.

Google Cloud and partners are committed to meeting customers where they are, and providing them with the ability to run key workloads and applications in the environment best suited for their business.

Google Cloud Marketplace for software and SaaS solutions

Customers can use their preferred Kubernetes- and Istio-ready software, or select from the growing list of open-source and commercially supported software and SaaS solutions available in the Google Cloud Marketplace.

Googles Cloud Marketplace is the canonical resource to discover, deploy, and use Anthos Ready Kubernetes applications for security and identity, data, analytics, developer tools, IT operations and more. Kubernetes applications are enterprise-ready containerized solutions with pre-built deployment templates, featuring capabilities such as portability, simplified licensing, and consolidated billing. They can be deployed now on Anthos, in the cloud, and on-premises—or in the future, on Kubernetes clusters hosted in other environments.These are not just container images, but partner-built commercial solutions.

Anthos Ready partner solutions

Anthos Ready designates partner solutions that meet Google Cloud's qualification requirements and have been verified to work with Anthos to meet the infrastructure and application development needs of enterprise customers. Partner solutions that complete and maintain the applicable qualification requirements are awarded the "Works with Anthos" badge to identify compatible infrastructure. The Anthos Ready program includes Storage, Network, Platform, Security & Identity, Data & Analytics, Developer Tools and IT Ops solutions from partners. The program will be extended to include other solution categories in the future.

Anthos Ready partners have met multiple criteria, such as:

- Demonstrated core Kubernetes functionality including dynamic provisioning of software and services via open and portable Kubernetes-native APIs
- A proven ability to automatically manage services across clusters, including scaling

Anthos Ready designates partner solutions that meet Google Cloud's qualification requirements and have been verified to work with Anthos to meet the infrastructure and application development needs of enterprise customers.

- A simplified deployment experience following Kubernetes practices

For example: Anthos Ready Storage recognizes partner solutions that have met a core set of requirements to run optimally with Anthos running on-premises, and helps organizations select storage solutions that can be deployed with Anthos. Presently, specific storage solutions from Dell, HPE, NetApp, Portworx, Pure Storage, and Robin.io have been qualified as Anthos Ready.

Modernize applications anywhere

Customers tell us that what Anthos can do is so transformational, that they want us to extend Anthos to more kinds of environments and applications. Why limit modern application deployment, management, and control to new applications? We agree and we're working hard to help bring Anthos to every application running everywhere. Until then, you can learn more about how Anthos can positively impact your bottom line by reading the latest Total Economic Impact report written by Forrester Research.

Next steps

To learn more or to get started, visit https://cloud.google.com/anthos

Google Cloud