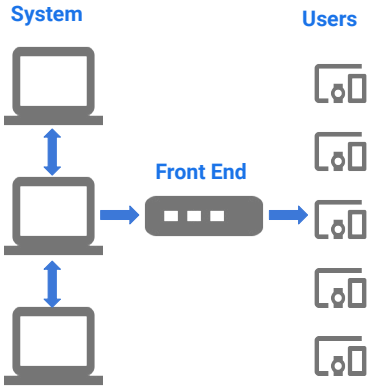
Distributed systems housed on-premises is the traditional approach but it lacks flexibility and agility

- Increasing capacity means buying more servers.
- Lead time for new capacity could be up to a year or more.
- Upgrades are expensive.
- The practical life of a server is short.
- Products and services may be constrained by the architecture.



Google Cloud

Now that you understand containers, let's take that understanding a step further and talk about using them in a modern hybrid cloud and multi-cloud architecture.

But before we do that, however, let's have a quick look at a typical on-premises distributed systems architecture, which is how businesses traditionally met their enterprise computing needs before cloud computing.

As you may know, most enterprise-scale applications are designed as distributed systems, spreading the computing workload required to provide services over two or more networked servers. Over the past few years, containers have become a popular way to break these workloads down into "microservices" so they can be more easily maintained and expanded.

Traditionally, these enterprise systems (and their workloads, containerized or not) have been housed on-premises, which means they are housed on a set of high-capacity servers running somewhere within the company's network, or within a company-owned data center.

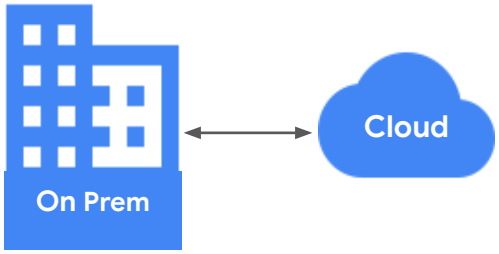
When an application's computing needs begin to outstrip its available computing resources, a company using on-premises systems would need to requisition more (or more powerful) servers, install them on the company network (after any necessary

network changes or expansion), configure the new servers, and finally load the application and its dependencies onto the new servers, before resource bottlenecks could be resolved.

The time required to complete an on-premises upgrade of this kind could be anywhere from several months to one or more *years*. It may also be quite costly, especially when you consider the useful lifespan of the average server is only 3-5 years.

Modern distributed systems allow a more agile approach to managing your compute resources

- Move only some of your compute workloads to the cloud if you wish.
- Migrate these workloads at your own pace.
- Quickly take advantage of the cloud's flexibility, scalability and lower computing costs.
- Add specialized services to your compute resources stack.



Google Cloud

But what if you need more computing power *now*, not *months* from now?

What if your company wants to begin to relocate some workloads away from on-premises, to the cloud, to take advantage of lower costs and higher availability, but is unwilling (or unable) to move the entire enterprise application from the on-premises network?

What if you want to use specialized products and services that are only available in the cloud?

This is where a modern hybrid or multi-cloud architecture can help. It allows you to:

- Keep parts of your systems infrastructure on-premises, while moving other parts to the cloud, creating an environment that is uniquely suited to your company's needs.
- Move only specific workloads to the cloud at your own pace, because a full scale migration is not required for it to work.
- Take advantage of the flexibility, scalability and lower computing costs offered by cloud services for running the workloads you decide to migrate.
- Add specialized services such as machine learning, content caching, data analysis, long-term storage, and IoT to your computing resources toolkit.

Anthos is Google's modern solution for hybrid and multi-cloud systems and services management

- Kubernetes and GKE On-Prem create the foundation.
- On-premises and cloud environments stay in sync.
- A rich set of tools is provided for:
  - Managing services on-premises and in the cloud.
  - Monitoring systems and services.
  - Migrating applications from VMs into your clusters.
  - Maintaining consistent policies across all clusters, whether on-premises or in the cloud.

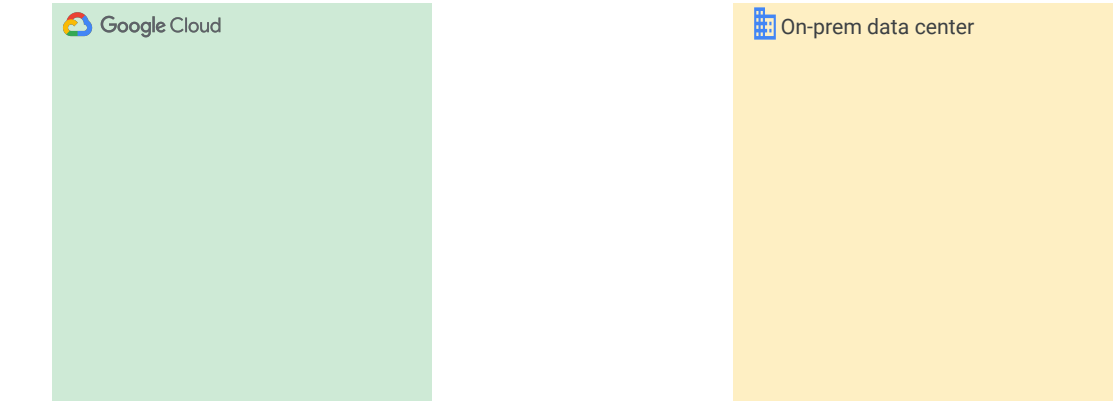
Google Cloud

You may have heard a lot of discussions recently concerning the adoption of "hybrid" architecture for powering distributed systems and services. You may have even heard discussion of Google's answer to modern hybrid and multi-cloud distributed systems and services management, called "Anthos."

But, what is Anthos?

Anthos is a hybrid and multi-cloud solution powered by the latest innovations in distributed systems and service management software from Google. The Anthos framework rests on Kubernetes and GKE On-Prem, which provides the foundation for an architecture that is fully integrated, with centralized management through a central control plane that supports policy-based application lifecycle delivery across hybrid and multiple cloud environments. Anthos also provides a rich set of tools for monitoring and maintaining the consistency of your applications across all of your network, whether on-premises, in the cloud, or in multiple clouds.

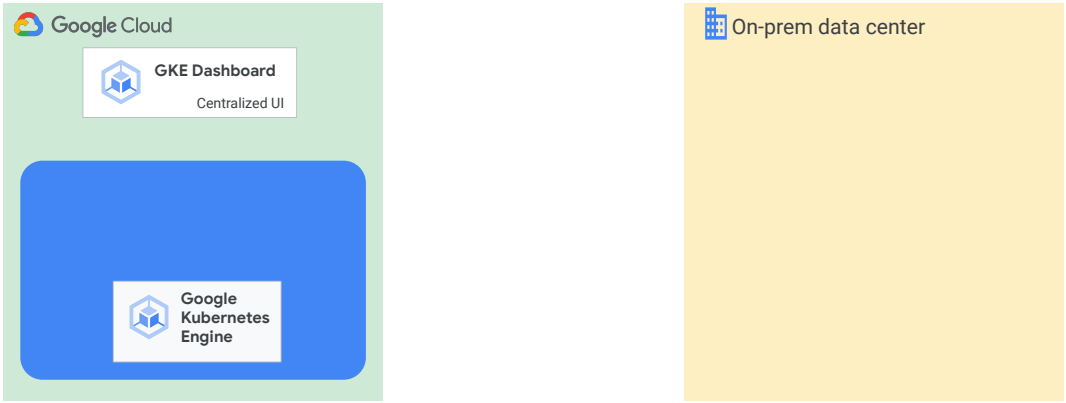
Building a modern hybrid infrastructure, step by step



Google Cloud

Let’s take a deeper look at this framework, as we build a modern hybrid infrastructure stack, step by step, with Anthos.

GKE for production ready apps



Google Cloud

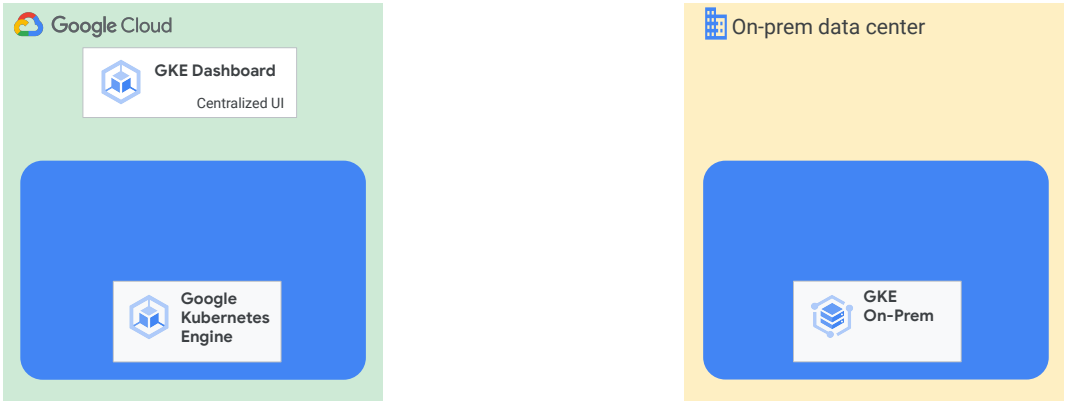
First, let's look at Google Kubernetes Engine on the cloud side of our hybrid network.

Google Kubernetes Engine:

- Is a managed, production-ready environment for deploying containerized applications.
- Operates seamlessly with high availability and an SLA.
- Runs Certified Kubernetes, ensuring portability across clouds and on-premises.
- Includes auto node repair, auto upgrade, auto scaling.

- Uses regional clusters for high availability with multiple control planes, node storage replication across multiple zones (as of Oct. 2019 the number of zones is 3.)

GKE On-Prem is turn-key production-grade Kubernetes



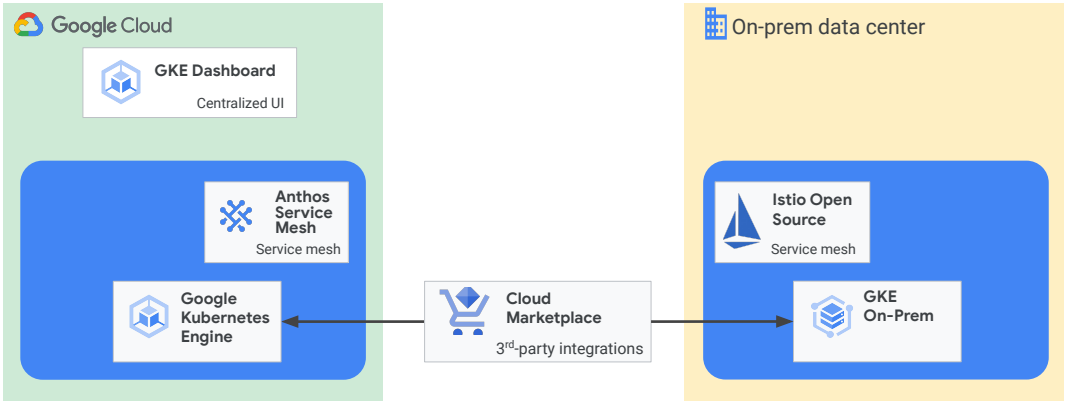
Google Cloud

Its counterpart on the on-premises side of our hybrid network is GKE On-Prem.

GKE On-Prem:

- Is a turn-key, production-grade, conformant version of Kubernetes with a best-practice configuration already pre-loaded.
- Provides an easy upgrade path to the latest Kubernetes releases that have been validated and tested by Google.
- Provides access to Container services on Google Cloud, such as Cloud Build, Container Registry, Cloud Audit Logs, and more.
- Integrates with Istio, Knative, and Cloud Marketplace solutions.
- Ensures a consistent Kubernetes version and experience across cloud and on-premises environments.

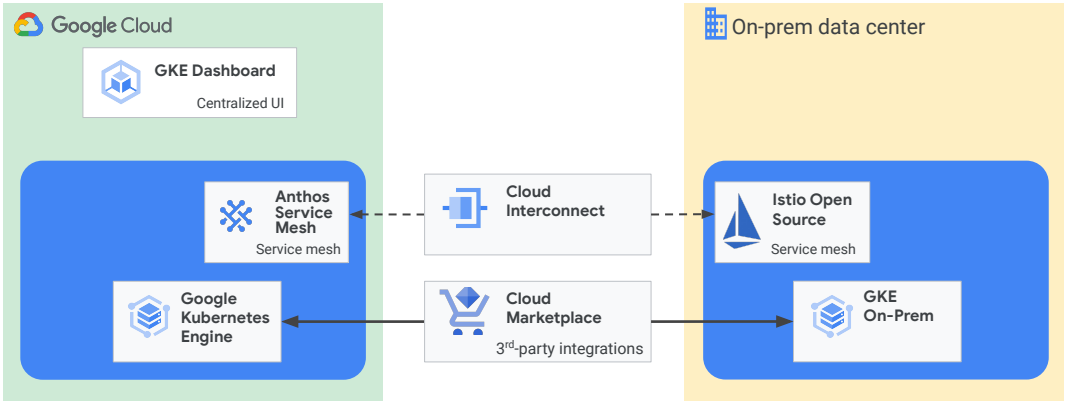
Marketplace applications are available to all clusters



Google Cloud

As mentioned, both Google Kubernetes Engine and GKE On-Prem integrate with Marketplace, so that all of the clusters in your network, whether on-premises or in the cloud, have access to the same repository of containerized applications. This allows you to use the same configurations on both sides of the network, reducing time spent developing applications (write once, replicate anywhere) and maintaining conformity between your clusters.

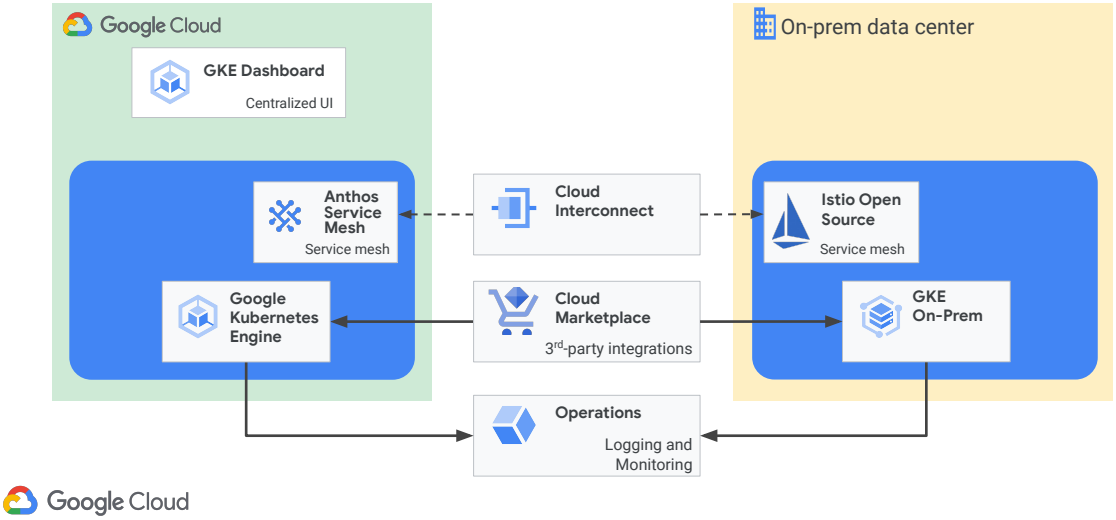
Service Meshes make apps more secure & observable



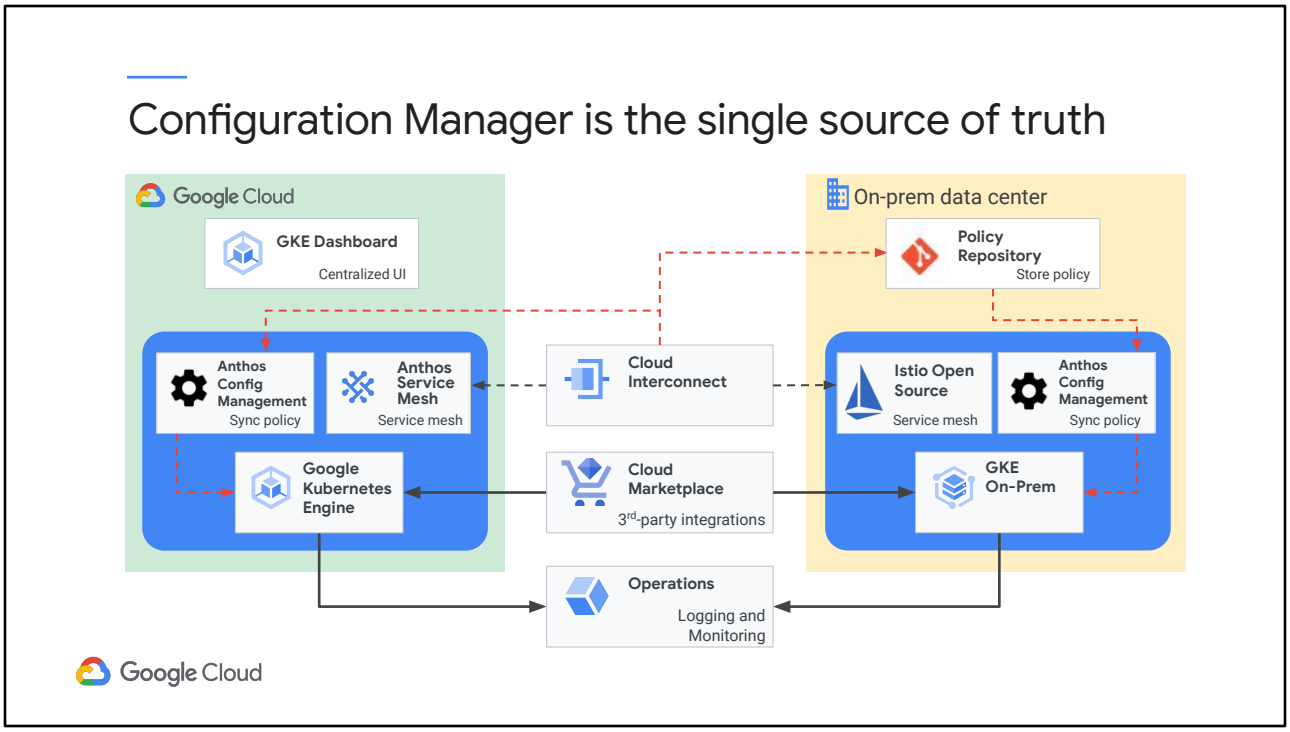
Google Cloud

Enterprise applications may use hundreds of microservices to handle computing workloads. Keeping track of all of these services and monitoring their health can quickly become a challenge. Anthos and Istio Open Source service meshes take all of the guesswork out of managing and securing your microservices. These service mesh layers communicate across the hybrid network using Cloud Interconnect, as shown, to sync and pass their data.

Cloud Logging and Cloud Monitoring watch all sides



Cloud Logging and Cloud Monitoring are the built-in logging and monitoring solutions for Google Cloud. Google Cloud's operations suite offers a fully managed logging, metrics collection, monitoring, dashboarding, and alerting solution that watches all sides of your hybrid or multi-cloud network. It's the ideal solution for customers wanting a single, easy to configure, powerful cloud-based observability solution that also gives you a single pane of glass dashboard to monitor all of your environments.



Lastly, Anthos Configuration Management provides a single source of truth for your clusters configuration. That source of truth is kept in the Policy Repository, which is actually a git repository (in this illustration this repository happened to be located on premises, but it can also be hosted in the cloud.) The Anthos Configuration Management agents use the Policy Repository to enforce configurations locally in each environment, managing the complexity of owning clusters across environments.

Anthos Configuration Management also provides administrators and developers the ability to deploy code changes with a single repository commit, and the option to implement configuration inheritance by using Namespaces.

GKE and Amazon ECS have similar service models

	GKE	Amazon EKS	Amazon ECS
Cluster nodes	Compute Engine instances	Amazon EC2 instances	
Supported daemons	Docker or rkt		Docker
Node agent	Kubelet		Amazon ECS Agent
Container group	Pod		Task
Deployment sizing service	Replication Controller	Service	
Command line tool	kubectl or gcloud	Amazon EKS CLI or kubectl	Amazon ECS CLI
Portability	Runs wherever Kubernetes runs	Runs only on AWS	

Google Cloud

Google Kubernetes Engine and Amazon Elastic Container Service serve the same purpose in different ways. Let's look at how certain features differ.

- Both use a cluster of virtual machines to run and manage containers. GKE uses Compute Engine instances, while Amazon EKS and Amazon ECS uses EC2 instances.
- While Amazon ECS supports Docker, GKE and Amazon EKS supports both Docker and Rocket format containers.
- The node agent for GKE and for Amazon EKS is Kubelet, which is an open source agent used with Kubernetes. Amazon ECS uses a proprietary Amazon ECS Agent.
- GKE and Amazon EKS both group containers by pods, which is how Kubernetes runs applications. Amazon ECS uses a specific group called a task.
- Sizing can be done on GKE using a Replication Controller, while Amazon EKS and Amazon ECS uses a Service.
- Running a command in GKE can be done through a combination of the open source kubeclt command line and gcloud. Amazon EKS can use its own CLI or kubectl. Amazon ECS uses the Amazon ECS CLI.
- Finally, one of the biggest advantages to GKE is portability. Kubernetes can run on a variety of clouds as well as physical hardware and virtual machines. Amazon EKS and Amazon ECS can only run on AWS.