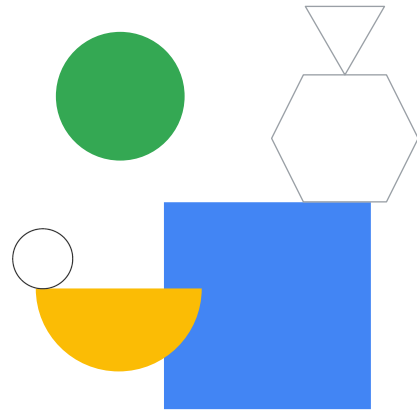# Application Migration with Anthos

Welcome to Application Migration with Anthos.

Learning objectives

Understand	Learn
Understand the path to modernization and its benefits so that you can choose the right tools and services for your use cases.	Learn to assess and discover which workloads are best suited for containerization and migration in both a manual and automated way.
Migrate	Optimize
Use the Migrate for Anthos and GKE solution to migrate stateless and stateful workloads from VMs to containers and Kubernetes.	Optimize migrated workloads so that you can maintain, improve, and continue developing them using the same methods as with green field software.

Google Cloud

In this module, we:

- Understand the path to modernization and its benefits, so that you can choose the right tools and services for your use cases.

- Learn to assess and discover which workloads are best suited for containerization and migration in both a manual and automated way.

- Use the Migrate for Anthos and GKE solution to migrate stateless and stateful workloads from VMs to containers and Kubernetes.

- Optimize migrated workloads so that you can maintain, improve and continue developing them using the same methods as with green field software.

Today's agenda

01	Modernization vision
02	Workload discovery and migration assessment
03	Migrating applications to containers

Google Cloud

Here is our agenda for the module.

Today's agenda

Google Cloud

Let's start by describing the modernization vision and the ways to achieve it.
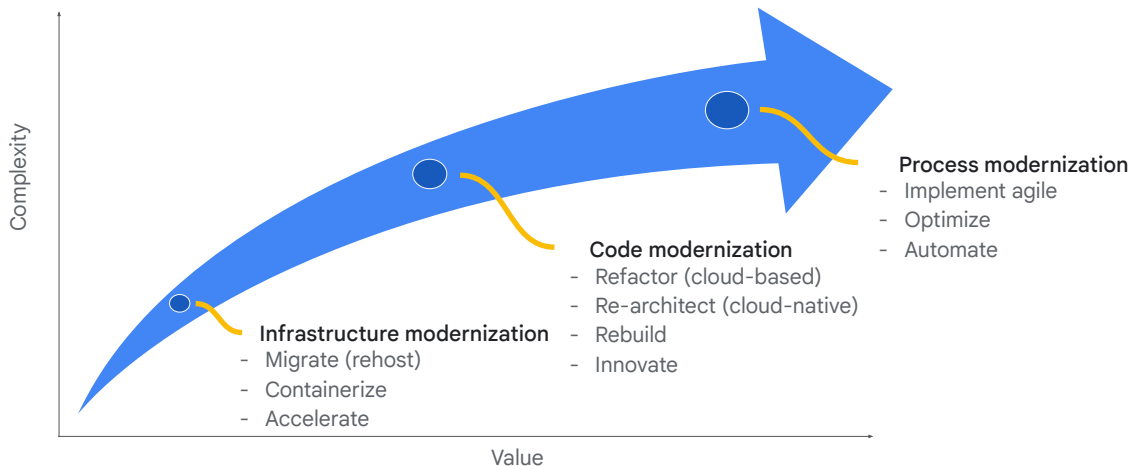
Modernization is a journey. There is a common vision, but the path is long and difficult.

When planning for application modernization, you must consider three ideas:

- Infrastructure modernization refers to implementing software-defined compute, storage, and networking resources that can scale on demand.

- Code modernization consists of using cloud-native concepts such as containers, microservices, and API-driven automation to refactor legacy workloads.

- Process modernization is the introduction of DevOps methodologies, including continuous integration and continuous delivery, or CI/CD, of new software. We talk about this in the next module and focus on infrastructure modernization.

Gains from achieving a modern infrastructure

CIO reasons for pursuing infrastructure and modernization
100 points allocated across:

|	
Agility and time to market	28
Quality of services or reliability	27
Cost efficiency	20
Security and risk reduction	19
Others (such as employee satisfaction and talent retention)	6
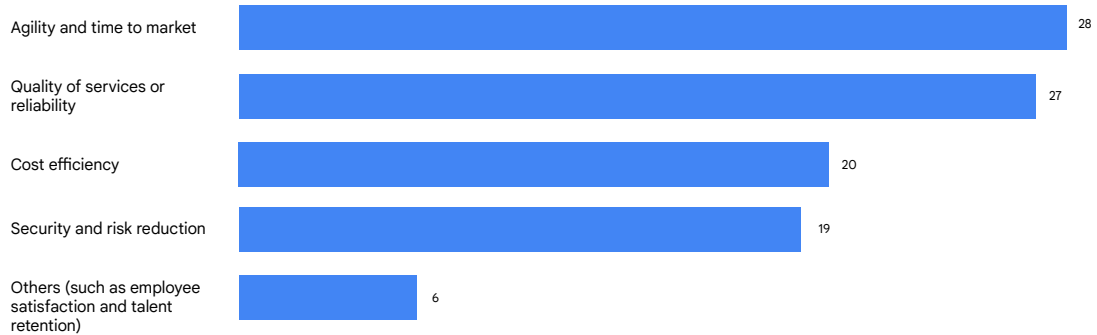
Source: McKinsey expert interviews (*N*=52)

Google Cloud

When looking at workloads modernization, CIOs seem to agree that there are many benefits. Organizations can:

- Move faster on development and time to market.

- Improve reliability and service quality.

- Cut cost while increasing security and reducing risk.

- Consider other factors such as employee satisfaction.

Migration strategies and destinations
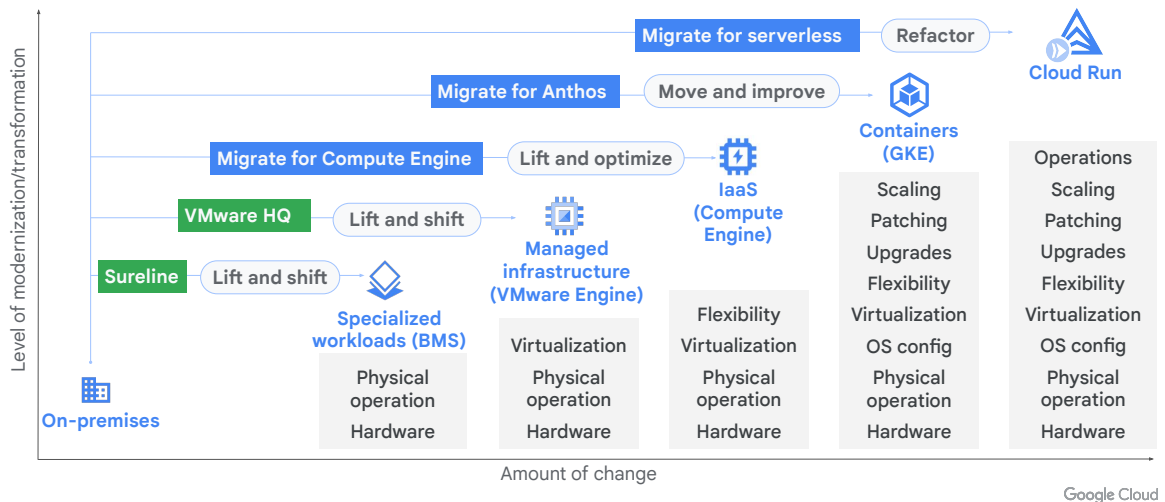
The goal when migrating enterprise workloads is to be most transparent and least disruptive with minimized downtime. The common approach until now has been to lift and shift, that is, to migrate workloads as they are.

To accomplish that, different strategies have been developed. If you have specialized workloads that require direct, low-level access to the server, or you run third-party virtualization software not supported in Google Cloud, you can use bare metal servers in Google Cloud. You basically offload the infrastructure costs and move from capital expenses to the operational expenses of managing that infrastructure.

If you are currently working with VMware and want to easily lift and shift your applications to Google Cloud without changes to your apps, tools, or processes, you can run in a dedicated VMware software-defined data center (SDDC) in Google Cloud. This migration offloads the hardware, but you keep paying the VMware licenses and focus on speed instead of cost optimization.

The third option is to lift and shift, but also optimize and rightsize. You can do so with Migrate for Compute Engine, which has been used to migrate tens of thousands of VMs to date. It supports migrating workloads from VMware installations using vCenter, and migrations from AWS and Azure clouds. By migrating the applications to Compute Engine, you can stop paying for any VMware licenses. But also, you can

further reduce costs by rightsizing your instances and making sure they are not overprovisioned, working with autoscalers to dynamically grow or shrink the number of VMs depending on user load, or benefit from other pricing models such as Spot VMs or commitments.

However, companies still don't get all the benefits of the cloud at this stage. You still have the same number of operating system images to manage, you might still not be getting all the efficiencies of running infrastructure dynamically because you have not increased the density, that is, one application per VM, and you might still have a disconnect on how you are managing your greenfield cloud-native applications in relation to the brownfield applications you migrated over.

So when companies come to the cloud, they also realize it's a good time to modernize. They start looking at Kubernetes to be infrastructure-agnostic and receive all the benefits of containers, such as application density, developer productivity, and agility. And they say, "I wish we could do the same for all the legacy workloads that we have on-premises," and start wanting to modernize those workloads. However, modernizing their workloads, that is, breaking up their monolithic applications and containerizing services, can be a very long and expensive process. Especially when companies can use Migrate for Anthos and GKE, a solution that migrates from VMs into containers that you can then run on Kubernetes, GKE, Anthos clusters, or even "container as a service" offerings like Cloud Run.

Now, it's not a binary, all-or-nothing decision: it depends on the type of workloads that you have. You can also choose to refactor certain applications and run them directly in Cloud Run.

Focus on migration to containers
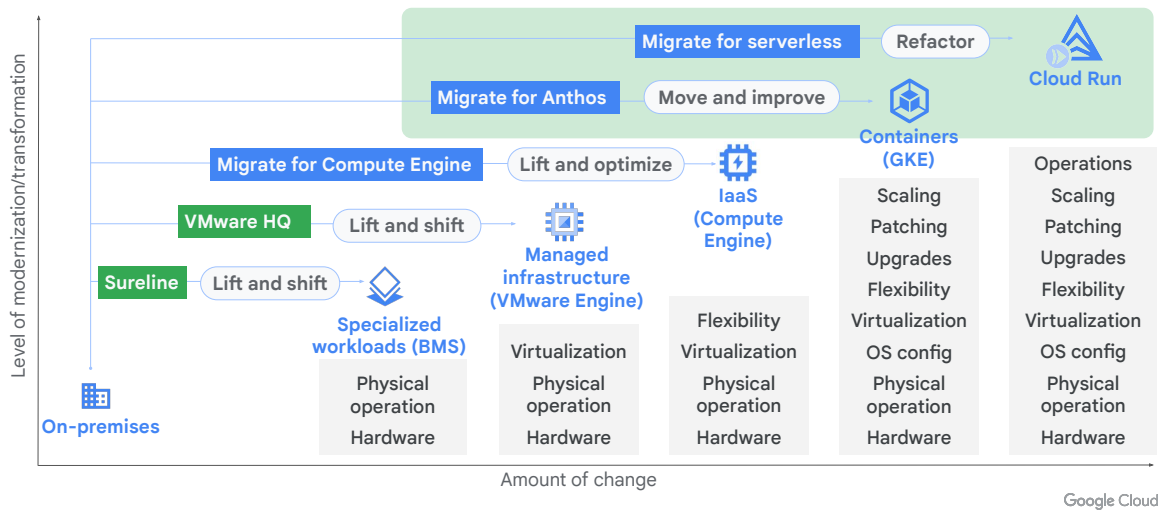
In this module, we focus on migrations to containers.

Business drivers for containerization and Kubernetes

- **Infrastructure savings**
  - Higher density (60-80%): Multiple containers in a single node
  - OS updates and kernel patches

- **Operations productivity**: Reduced downtime and IT management
  - Automated and coded "desired state" management
  - Autoscaling, self-healing, monitoring, security, etc.

- **Developer productivity**: Faster time to market, agility
  - Modern app lifecycle management with CI/CD pipelining
  - Fast deployment, "Write once, run everywhere"

Google Cloud

There are many business drivers for containerization and Kubernetes. The most visible benefit is the realization of infrastructure savings by increasing application density. Many enterprise applications, although necessary and key in many areas, are not high traffic but still must always be running, available, and up to date. These are perfect candidates to be placed in a shared infrastructure instead of in their own compute resources.

Increasing operations productivity is sometimes undervalued because it's not so visible at first glance. However, having a unique way to manage all applications in your organization and leveraging a modern platform for managing, autoscaling, self-healing, and securing them reduces overall complexity and errors. It also reduces the number of people involved in keeping software running.

Finally, there is an increased speed from idea to market because developers can focus on building the business logic and use modern, out-of-the-box, zero-touch CI/CD pipelines and the Kubernetes platform to deploy and run their applications.

Manual containerization is difficult

Complex apps	Multi-tier applications with many dependencies
Data gravity	Large volumes of data (files, databases)
Legacy apps	Outdated development models, VM-based
Risk	What if apps don't run (well) in cloud?

Google Cloud

But moving to the cloud and modernizing legacy workloads can be difficult, expensive, and time consuming. Enterprise applications portfolio can have many problems:

- Complicated, multi-tier applications might have many dependencies.

- Large chunks of different types of data can prolong the moving process.

- Many legacy applications might be 10 to 15 years old with outdated development models and long-gone developers, so their context has disappeared.

- Some applications won't work properly in the cloud.

Migrate for Anthos and GKE automates containerization

- Eliminates app rewrite development costs.
- Generates application-specific CI/CD artifacts such as the Docker image, the Dockerfile, and the deployment yaml.
- Refactors storage into a Kubernetes-supported PVC.
- Leaves out VM-related files and components that are not essential for the application in a Kubernetes environment.
- Ported applications can run on any Anthos or GKE cluster independent of the Migrate for Anthos deployment.
- Supported sources: VMware, AWS, Azure, Compute Engine
- Processing environments: GKE, Anthos on AWS, Anthos on VMware
- Supported Workloads OS types: Linux and Windows distributions
- Available user interfaces: Google Cloud Console, Google Cloud CLI, API
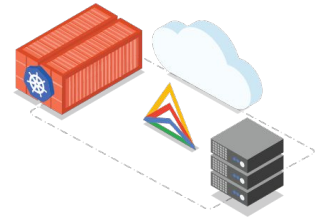
Google Cloud

Migrate for Anthos and GKE automates containerization and eliminates app rewrite development costs.

It also generates application-specific CI/CD artifacts such as the Docker image, the Dockerfile, and the deployment yaml.

Source VMs might have different file systems and logical volume managers that are not necessarily standard or suitable for maintenance by Kubernetes. So as part of the modernization, transform the storage layout into a standard Kubernetes PersistentVolumeClaim. You can use any ready Kubernetes or Anthos storage provider, and you can decide which storage is currently best to serve your volume need regardless of decisions that were made years ago when the VM was originally structured.

Migrate for Anthos and GKE also omits VM-related files and components that are not essential for the application in a Kubernetes environment. For example, the files and components that keep the VM updated, monitored, managed, and patched don't need to be copied over.

Ported applications can run on any Anthos or GKE cluster independent of the Migrate for Anthos deployment.

Migrate for Anthos and GKE supports a variety of sources, including VMware, AWS, Azure, and Compute Engine, and different processing environments, such as GKE,

Anthos on AWS, and Anthos on VMware.

Both Linux and Windows operating systems are supported.

Migrations can be managed via the Google Cloud Console, Google Cloud CLI, APIs.

Today's agenda

Google Cloud

Now that we understand the reasons for modernizing workloads, let's discuss the kind of workloads suited for containerization with Migrate for Anthos.

Migrate for Anthos and GKE supports specific OS and GKE versions

OS	Compute Engine	VMware	AWS	Azure
CentOS	6.0, 7.0, 7.0 UEFI, 8.0	6.7, 6.9, 7.6	6.10, 7.5, 7.6	6.5-7.6, 7.0 LVM
Debian	7.0, 8.0, 9.0, 10.0	9.4, 9.6	6.0, 7.0, 8.0, 9.0, 10.0	8.0, 9.0, 10.0
RHEL	6.0, 7.0, 7.0 UEFI, 7.4 SAP, 7.6 SAP, 8.0	6.5, 7.5, 7.6, 8.3	5.10, 5.11, 6.5-7.0, 7.1-7.7, 8.0, 8.1	6.7-6.10, 7.2-7.4, 7.5 LVM, 7.5 RAW, 7.6 LVM, 7.6 RAW, 7.4-7.7 SAP, 8.0
SUSE	12, 12 SP3 SAP, 12 SP4 SAP, 15, 15 SAP, 15 SP1 SAP	12 SP2, 12 SP3, 12 SP4, 15	11 SP4, 11 SP4 SAP, 12 SP2 SAP, 12 SP3, 12 SP3 SAP, 12 SP4, 12 SP4 SAP, 15 SAP, 15 SP1, 15 SP1 SAP	11 SP4 SAP, 11 SP4, 12 SP1, 12 SP1 SAP, 12 SP2, 12 SP2 SAP, 12 SP3, 12 SP3 SAP, 12 SP4, 12 SP4 SAP, 15, 15 SAP, 15 SP1, 15 SP1 SAP
Ubuntu	12 LTS, 14 LTS, 16 LTS, 16 LTS minimal, 18 LTS, 18 LT..., 18 LTS UE..., 19.04 min...	12.04.5 LTS, 14.04 LTS,	12 LTS, 14 LTS, 16 LTS, 18 LTS, 19.04	12.04 LTS, 14.04 LTS, 16.04 LTS, 18.04 LTS, 19

OS	Compute Engine
Windows	• Microsoft Windows Server 2008R2 or higher.
	• Microsoft IIS 7 or higher web applications.
	• ASP.NET and .NET Framework version 3.5 or higher.
	After migration, the resulting container will run Windows Server version 1909.

1.13.5-gke.10 and later
(GKE version for processing cluster)

Ubuntu
COS (ext2/3/4)
(GKE node OS for processing cluster)

Google Cloud

As we mentioned before, Migrate for Anthos and GKE automates the containerization and execution of your workloads, and it supports specific OS and GKE versions. These include CentOS, Debian, RHEL, SUSE, Ubuntu, and Windows operating systems. The GKE cluster used to process the migration must use Ubuntu Container-Optimized OS and must be version 1.13 or later.

Now, let's discuss specific types of application to containerize: What do you think are the best application candidates to migrate?

Good workload fit for Migrate for Anthos

Good fit

Application architectures:
- Web/application servers
- Stateless web frontends
- Business logic middleware (such as Tomcat, J2EE, COTS)
- Multi-VM, multi-tier stacks (such as LAMP, WordPress)
- Small/medium-sized databases (such as MySQL, PostgreSQL)

Characteristics:
- Low duty-cycle and bursty workloads
- Development, testing, and training lab environments
- Always-on, low-load services
- Resilience to restarts and scaleout
- Compatible licensing (not tied to OS, hardware, hypervisor)
- Compatible third-party vendor support contracts

Google Cloud

In general, if containerized usage is supported and a path to containerization without much complexity is available, the workload is a good candidate for Migrate for Anthos. Some workloads are especially good candidates. For example:

- Web/application servers

- Stateless web frontends

- Business logic middleware (such as Tomcat, J2EE, COTS)

- Multi-VM, multi-tier stacks (such as LAMP, WordPress)

- Small/medium-sized databases (such as MySQL, PostgreSQL)

These applications are an especially good fit if they share the following characteristics:

- Low duty-cycle and bursty workloads

- Development, testing, and training lab environments

- Always-on, low-load services

- Resilient to restarts and scaleout

- Compatible licensing (not tied to OS, hardware, hypervisor)

- Compatible third-party vendor support contracts

What do you think are the worst application candidates to migrate?

Bad workload fit for Migrate for Anthos

Bad fit

- VMs with special kernel drivers (such as kernel mode NFS)
- Dependencies on specific hardware
- Software with licenses tied to certain hardware ID registration
- Always-on database VMs that don't fit Cloud SQL
- HW-specific license constraints (such as per CPU)
- 32-bit OS
- File servers
- VM-based workloads that would require whole-node capacity, such as high-performance, high-memory databases (SAP HANA)
- Single-instance workloads that rely on Compute Engine live migration to meet high availability requirements

Google Cloud

Some workloads are especially bad candidates for migration. For example:

- VMs with special kernel drivers (such as kernel mode NFS)

- Dependencies on specific hardware

- Software with licenses tied to certain hardware ID registration

- Always-on database VMs that don't fit Cloud SQL

- HW-specific license constraints (such as per CPU)

- 32-bit OS

- File servers

- VM-based workloads that would require whole-node capacity, such as high-performance, high-memory databases (SAP HANA)

- Single-instance workloads that rely on Compute Engine live migration to meet high availability requirements

For some of the workloads, we recommend investigating whether they are a fit for Migrate for Compute Engine instead.

Potential workload fit for Migrate for Anthos

Potential fit	**Database servers:** ● High SLA requirements for HA, DR, and performance are best migrated to managed DB services. ● Non-production, such as test or dev environments with a low SLA requirement, could be better candidates for Migrate for Anthos.
TBD	These workloads would require more exploration. We recommend engaging the Migrate for Anthos product team for assistance on evaluating the engagement. ● SAP NetWeaver Application Servers

Google Cloud

For other workloads, consider SLA requirements for high availability, disaster recovery, and performance. In most cases when you migrate a database, a low SLA is best serviced by moving to a managed database service. If these are not production apps (for example, the app is in test or training), it might not matter whether the database has highly reliable services and is a candidate for Migrate for Anthos.

Some workloads would require more exploration. We recommend engaging the Migrate for Anthos product team for assistance with evaluating the engagement for products such as the SAP NetWeaver Application Servers.

Automating workload discovery and assessment with StratoZone and the fit assessment tool

StratoZone
- Workload discovery
- Network dependencies
- Migration roadmap
- Opportunities for modernization
- Cloud rightsizing
- Business case

Fit assessment tool
- Workload qualification
- Standalone HTML outputs
- Integration with Google Cloud Console

Google Cloud

Going through all enterprise workloads manually can be a tedious process. Google Cloud provides solutions for automating the workload discovery and assessment with StratoZone and the fit assessment tool.

StratoZone, which was acquired by Google in 2020, provides a fast, data-driven assessment of your current infrastructure and workloads to accelerate the start of your cloud journey. It provides a workload discovery solution that encompasses both applications and network dependencies, together with a migration roadmap with all the opportunities for modernization, including cloud rightsizing reports for comparing the costs in Google Cloud and making your business case.

Although StratoZone focuses on general discovery and can be used to do lift and shift with Migrate for Compute Engine, the fit assessment tool focuses on qualifying the workload fit to be converted into containers. This tool can generate reports both as a standalone HTML output and as integrations with the Google Cloud Console.
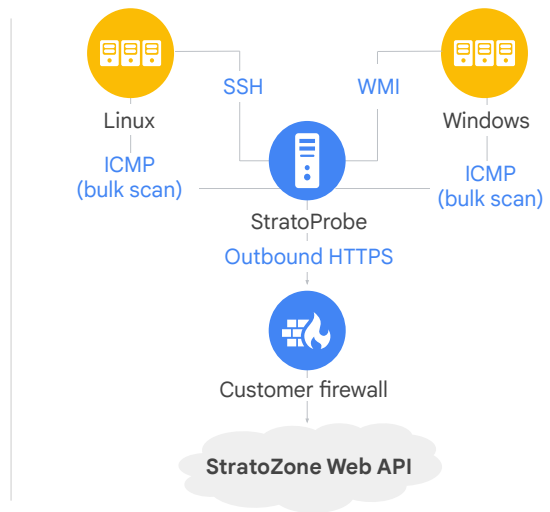
StrataZone collects data

Data that is collected:

- Machine data (capacity, model, hardware)
- Machine performance (CPU, memory, storage)
- Installed application names
- Network traffic data (port, type, direction)
- Processes, services, and related utilization

Data that is not collected:

- User information or profiles
- Data underlying any applications
- Data in your databases or file storage
- Application configuration files
- Data subject to PCI or HIPAA regulations

- All data is **encrypted** in transit and at rest.
- Customers can **anonymize** any data required.

Linux · SSH · WMI · Windows
ICMP (bulk scan) · StratoProbe · ICMP (bulk scan)
Outbound HTTPS
Customer firewall
**StratoZone Web API**

Google Cloud

StratoZone collects data related to machines, performance, capacities, network, and utilization. It uses this information to qualify the workloads and to suggest which Google Cloud resources can be used if you do a lift and shift to Compute Engine VMs.

However, StratoZone DOES NOT COLLECT data from inside applications, databases, or user data except for the username and email of the admin who runs the collector AND the username associated with running processes.

StratoZone does not collect any user information and is compliant with many standards such as PII, PCI, and HIPAA.

Fit assessment tool verifies fitness for containerization

- Helps customers discover their inventory and assess fit for modernization.
- Uses a lightweight CLI that runs on admin desktop.
- Runs completely **disconnected** from the internet.
- Main features:
  - Acquires VMs **inventory** with vSphere integrations.
  - Obtains single workload information from VMs running on Compute Engine, AWS, and Azure.
  - Collects guest-level data via script running on each VM.
  - Supports Linux and Windows VMs.
- Collected data is also used to **automate** parts of the migration process.
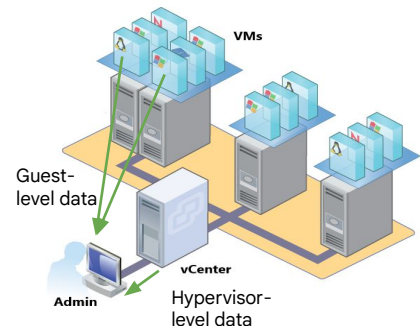
Google Cloud

The fit assessment tool verifies fitness for containerization.

It also helps customers discover their inventory, but it especially focuses on assessing workload fitness for modernization.

It uses a lightweight CLI running on admin desktop and runs completely disconnected from the internet.

Some features this tool offers include the possibility to:

- Acquire VMs inventory with vSphere integrations.

- Obtain single workload information from VMs running on Compute Engine, AWS, and Azure.

- Collects guest-level data via script running on each VM.

These features are available for both Linux and Windows VMs

Finally, all the collected data can also be used to automate parts of the migration process.

Steps to assess a workload with the fit assessment tool

1️⃣ Use ssh to connect to the VM, and download the tool:

```
curl -O
https://anthos-migrate-release.storage.googleapis.com/v1.10.2/linux/amd64/mfit-linux-
collect.sh
curl -O
https://anthos-migrate-release.storage.googleapis.com/v1.10.2/linux/amd64/mfit
```

2️⃣ Collect machine information in a downloadable tar file:

```
./mfit-linux-collect.sh
```

3️⃣ Create an assessment report in an HTML, JSON, or CSV file:

```
./mfit assess sample m4a-collect.tar --format json > monolith-mfit-report.json
```

Google Cloud

Download the fit assessment tool and either use ssh or collect data remotely if you are on vSphere.

Then collect information for your machine, which gathers a detailed report of all basic VM information, such as OS and cores, and detailed information about the workloads that run on it. This can be downloaded as a zip or a tar file.

From this downloadable file, you can run an assessment and export a report in an HTML, JSON, or CSV file, depending on how you want to analyze it.

Fit assessment reports available in CSV, JSON, and HTML

Fit assessment generates reports in CSV, JSON, and HTML. JSON files can be viewed in the Google Cloud Console for a richer user experience.
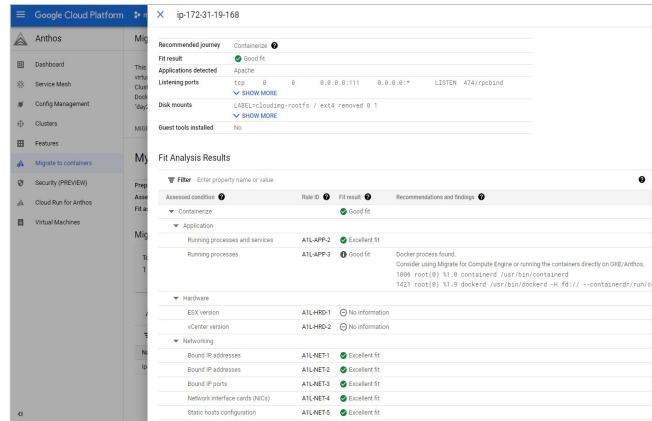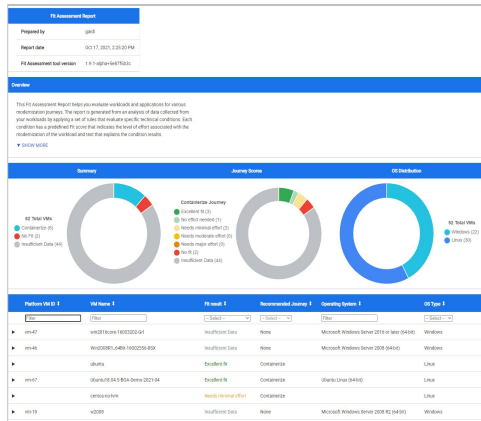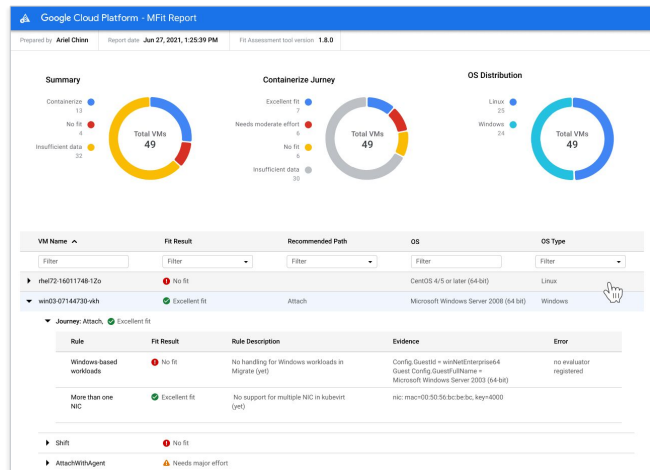
Fit assessment reports qualify your workload for migration

- Possible results:
  - An excellent fit
  - A good fit with some findings that might require attention
  - A fit with minimal, moderate, or major work before migrating
  - No fit or insufficient data
- Platforms:
  - Compute Engine VM
  - GKE or Anthos clusters
  - Tomcat
  - Cloud Run

Google Cloud

The fit assessment reports qualify your workload for migration and specify whether you have:

- An excellent fit

- A good fit with some findings that might require attention

- A fit, with minimal, moderate, or major work before migrating

- No fit or insufficient data.

Fit assessment is also performed per platform. Some workloads might be ideal for migration to Compute Engine, but might not be ready for Cloud Run, for example, if they use stateful workloads. Also, a stateful workload might need minimal changes to work on Anthos and GKE, such as setting up PersistentVolumes to work with the underlying hardware and PersistentVolumeClaims to work with the storage component. Additionally, there is a Tomcat option component. Migrate for Anthos optimizes the migration of Tomcat servers; it disregards other services that run on the VM and therefore optimizes the migration for this type of server.

Today's agenda

Google Cloud

Now that we have clarified which workloads to migrate, let's discuss the migration with Migrate for Anthos.

Migration setup and planning

Discovery and assessment

- What applications do I have running?
- What are the best candidates for migration?

STRATOZONE*

Fit report

Setup and planning

- Where will I run my migrated applications?
- How can I scale my migration process?

Migration plan (yaml)

Repeat at-scale

Migrate and modernize

- How do I perform an actual migration?
- What if I need to modify or customize a migrated application?

* Container image
* Docker file
* Deployment yaml
* Data volumes

Optimize

- How do I handle the ongoing maintenance or upgrades of a migrated application?

Google Cloud

After you complete the discovery and assessment phase with StratoZone and the fit assessment tool, it's time to discuss the setup and plan how to run your applications.

1. Decide on the source of the migration

- Determine the operating system of your VMs:
  - Linux
  - Windows
- Determine the origin:
  - VMware
  - AWS
  - Azure
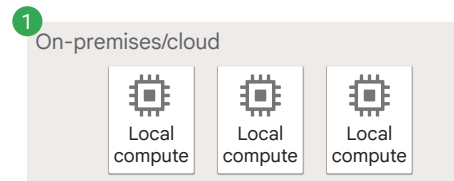
Let's walk through the migration process steps. The first step is to decide on the source of the migration; that is, determining whether you are going to migrate Linux or Windows VMs, and the source provider, such as VMware, AWS, Azure, or Compute Engine, where the VMs that you want to migrate and run as containers on GKE or Anthos are located.

2. Create a processing cluster in VMware, AWS, or Google Cloud
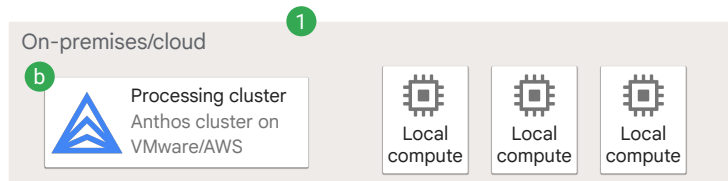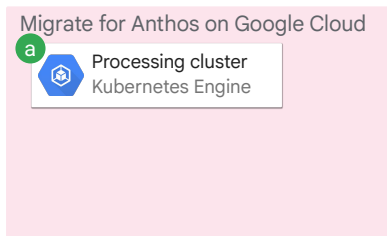
If your migration source is VMware or AWS, you can use a local Anthos cluster in the same location to migrate your workloads. If your destination is Google Cloud or you are migrating from other sources, you must set up a GKE cluster in Google Cloud to process the migration.

2. GKE processing clusters use Migrate for Compute Engine
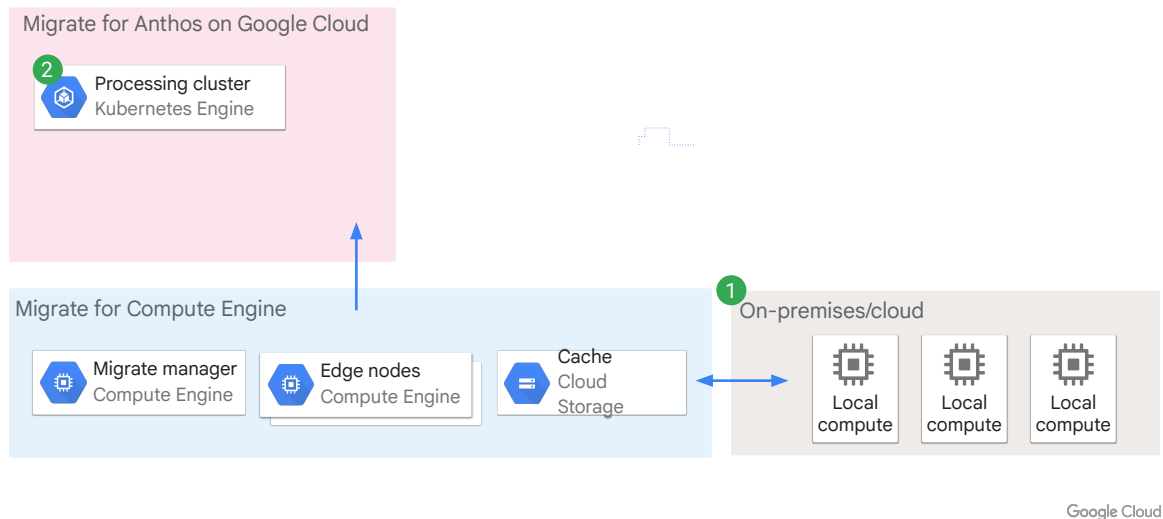
For VMware, AWS, and Azure, when the target is Google Cloud, there is a dependency on installing Migrate for Compute Engine to facilitate the transfer of workloads into Google Cloud.

Migrate for Compute Engine components:

1. The Migrate for Compute Engine Manager is a Google Cloud VM deployed into a Google Cloud project. This VM will orchestrate the migration operations and provide a web UI.

2. The Migrate for Compute Engine backend is a virtual appliance deployed into your vSphere environment. It handles the data center part of streaming data.
    a. It communicates with Google Cloud API endpoints and Google Cloud's operations suite.

3. The Migrate for Compute Engine vCenter Plugin provides management functionality within the vCenter UI.

4. The Cloud Extension handles the data streaming on the Google Cloud side.
    a. It's composed of a couple of edge node VMs (two VMs in two zones for HA) that serve as a bridge between the migrated VMs and the storage caching tier (using iscsi).
    b. Edge nodes communicate with Cloud Storage and Google Cloud's operations suite.

3. Add a migration source

Migrate for Anthos on Google Cloud

(2) Processing cluster
Kubernetes Engine

SC
(3)

```
migctl source create local-vmware local-vmware-src \
  --vc '1.2.3.4' \
  --username 'admin'
```

Migrate for Compute Engine

Migrate manager
Compute Engine

Edge nodes
Compute Engine

Cache
Cloud Storage

(1) On-premises/cloud

Local compute

Local compute

Local compute
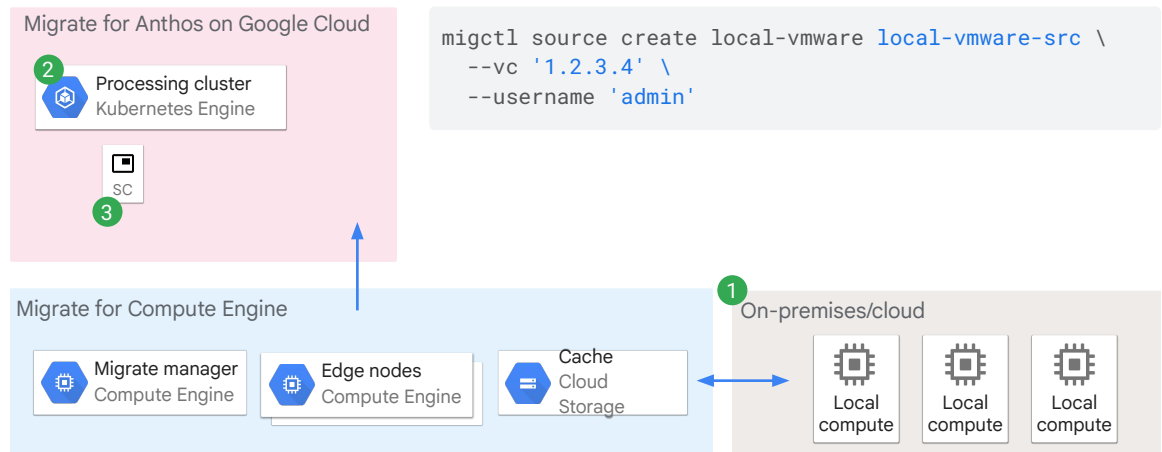
Google Cloud

Define the migration source you're migrating from by running the migctl source create command or by using the Google Cloud Console. This process adds the details that are needed to migrate from the source you specify: VMware, AWS, Azure, Compute Engine, or on-premises.
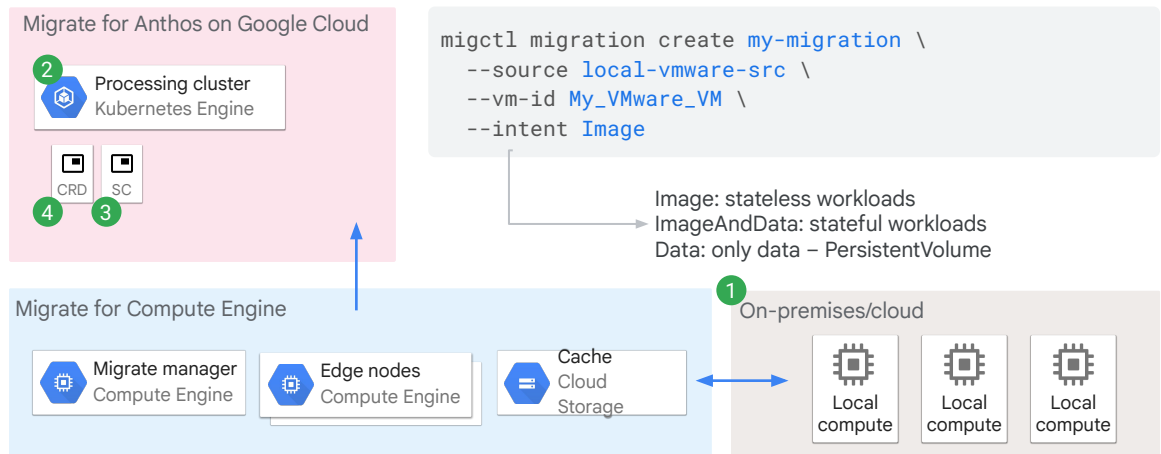
4. Create a migration plan


```
migctl migration create my-migration \
  --source local-vmware-src \
  --vm-id My_VMware_VM \
  --intent Image
```

Image: stateless workloads
ImageAndData: stateful workloads
Data: only data – PersistentVolume

Migrate for Anthos on Google Cloud

2 Processing cluster
Kubernetes Engine

CRD   SC
4   3

Migrate for Compute Engine

Migrate manager
Compute Engine

Edge nodes
Compute Engine

Cache
Cloud
Storage

1 On-premises/cloud

Local compute   Local compute   Local compute

Google Cloud

A migration plan is the central object with which you perform migration actions and monitor migration activities and status with migctl and the Google Cloud Console. The migration object is implemented as a Kubernetes Custom Resource Definition, or CRD, and is contained along with additional resources such as a Kubernetes PersistentVolumeClaim in the migration plan.

When you create a migration, you specify an intent flag value based on the nature of your workload. The flag's value determines the contents of your migration plan, which in turn guides the migration. There are three options available:

- Image: use for stateless workloads.

- ImageAndData: use for stateful workloads where the application and user mode system are extracted to a container image, and data is extracted to a persistent volume.

- Data: use with stateful workloads where only the data portion of the migration plan is extracted into a persistent volume. If you use the same source VM and migration plan to repeatedly execute with this intent value, the result is to synchronize delta changes since the previous data sync to the target persistent disk.

This process takes a couple of minutes to complete and can be checked by using the provided command: gcloud container clusters get-credentials.

Migration and modernization phase

Repeat at scale

Discovery and assessment

- What applications do I have running?
- What are the best candidates for migration?

STRATOZONE®

Fit report

Setup and planning

- Where will I run my migrated applications?
- How can I scale my migration process?

Migration plan (yaml)

Migrate and modernize

- How do I perform an actual migration?
- What if I need to modify or customize a migrated application?

* Container image
* Docker file
* Deployment yaml
* Data volumes

Optimize

- How do I handle the ongoing maintenance or upgrades of a migrated application?

Google Cloud

After the setup is complete, the next step is to customize and execute the migration.

5. Customizing your migration plan


```
migctl migration get my-migration
```
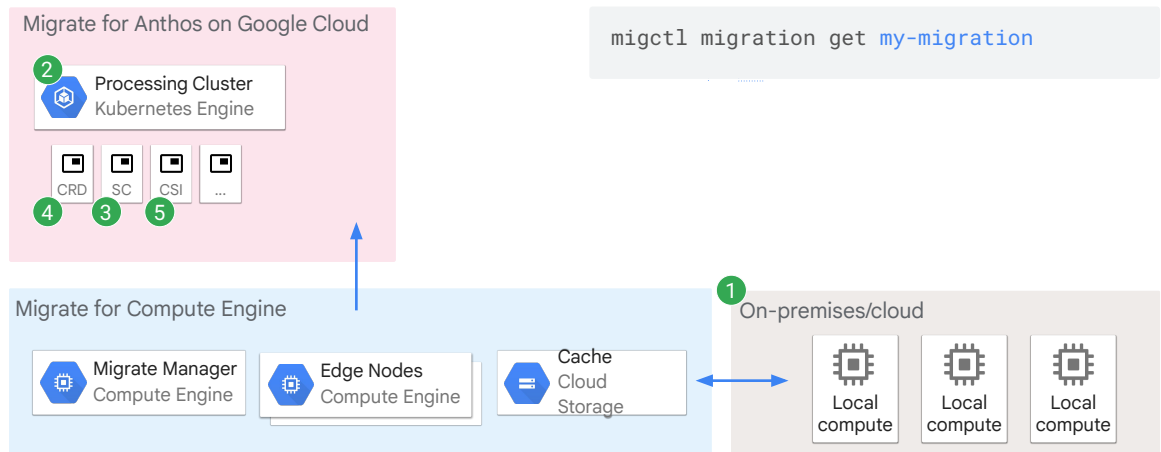
Use the migctl migration get command to obtain the migration files so that you can review and customize them before proceeding to execution. The details of your migration plan are used to extract the workload's container artifacts from the source VM, and also to generate Kubernetes deployment files that you can use to deploy the container image to other clusters, such as a production cluster.

Configure the services and data needed from the original VM and the destination namespace in the Kubernetes cluster or the persistent storage to be used.

6. Generate container artifacts
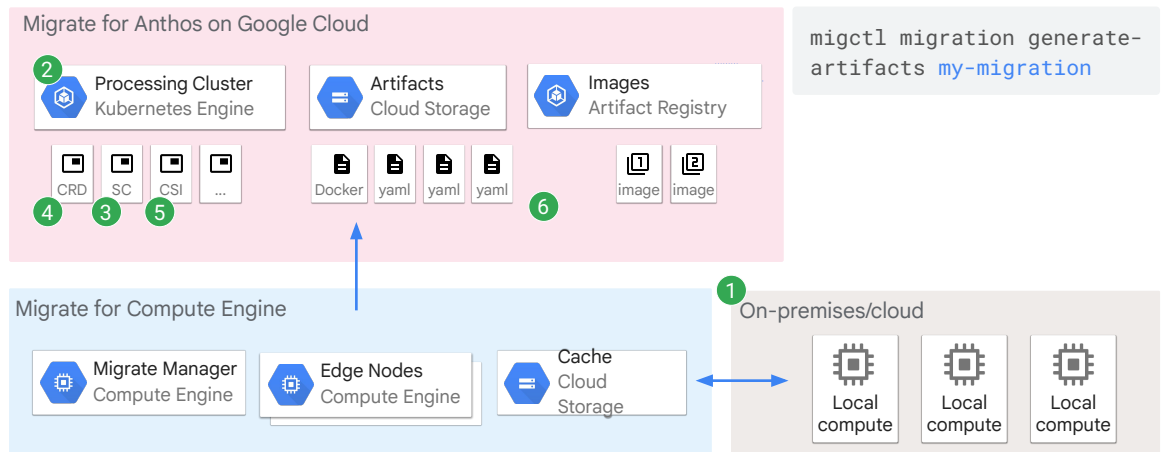
Use the migctl migration generate-artifacts command to:

- Extract workload artifacts from the VM into a container image and data volume (for stateful workloads).

- Copy files and directories that represent the VM to the Container Registry as Docker images.

- Generate migration deployment artifacts you can use for production deployments to another GKE cluster. These are copied into a Cloud Storage bucket as an intermediate location. You can later download these files.

7. Deploy your workload to a GKE or Anthos cluster

Migrate for Anthos

Processing Cluster	Artifacts	Images
Kubernetes Engine	Cloud Storage	Artifact Registry

CRD SC CSI ...   Docker yaml yaml yaml   image image

Production project

Production cluster
Kubernetes Engine

app

```
kubectl apply -f app.yaml
```

Migrate for Compute Engine

Migrate Manager	Edge Nodes	Cache
Compute Engine	Compute Engine	Cloud Storage

On-premises/cloud

Local compute Local compute Local compute

Google Cloud

After the artifacts are generated, you can:

- Test the image and data volume (for stateful workloads) on the processing cluster.
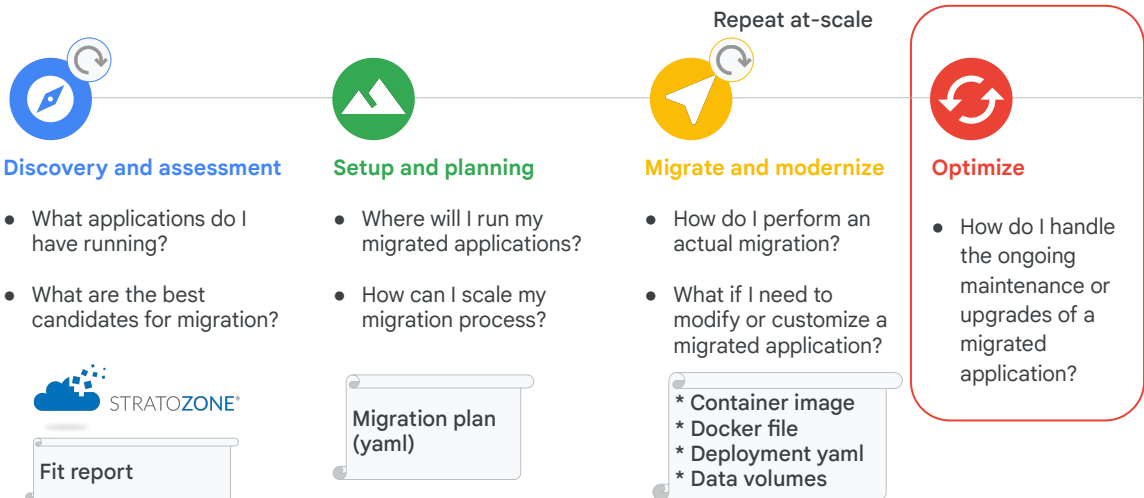
- If the results are good, deploy the image and data volume (for stateful workloads) to production cluster.

Optimization and maintenance phase

Repeat at-scale

Discovery and assessment

- What applications do I have running?
- What are the best candidates for migration?

STRATOZONE®

Fit report

Setup and planning

- Where will I run my migrated applications?
- How can I scale my migration process?

Migration plan (yaml)

Migrate and modernize

- How do I perform an actual migration?
- What if I need to modify or customize a migrated application?

* Container image
* Docker file
* Deployment yaml
* Data volumes

Optimize

- How do I handle the ongoing maintenance or upgrades of a migrated application?

Google Cloud

In this phase, you can use Istio to add access policies, encryption, and authentication and use Cloud Logging and Cloud Monitoring to monitor and log, all by changing configuration instead of rebuilding your applications. You can also use tools like Cloud Build to integrate with a CI/CD pipeline in order to implement day-2 maintenance procedures, such as software package and version updates.

How do you handle the ongoing maintenance or upgrades of a migrated application?

The container artifacts you create with the migctl migration generate-artifacts command are meant for both deployment of the migrated workload in the target cluster and for day-2 maintenance operations, including applying application and user-mode OS software updates, security patches, editing embedded configurations, adding or replacing files, and updating the Migrate for Anthos and GKE runtime software.

Optimization and maintenance phase

```
# Runnable layer provided by Migrate for Anthos
FROM anthos-migrate.gcr.io/v2k-run-embedded:v1.6.0 as migrate-for-anthos-runtime

# If you want to update parts of the image, add your commands here. For example:
# RUN apt-get update
# RUN apt-get install -y \
#   package1=version

# Image containing data captured from the source VM
FROM gcr.io/myproject/myworkload-non-runnable-base:v1.0.0 as source-content

COPY --from=migrate-for-anthos-runtime / /

# Migrate for Anthos image includes entrypoint
ENTRYPOINT [ "/.v2k.go" ]
```

Google Cloud

Such maintenance operations leverage the generated Dockerfile and the captured system image layer. When these are combined with the Migrate for Anthos and GKE runtime layer, these files can be built into an executable container image.

In the generated Dockerfile, the code will look similar to the code on the slide: this piece is the layer that comes from the captured image. We call it the non-runnable image because it contains the extracted system. That by itself cannot run in a container. It requires the other piece, the runnable layer, which is a Migrate for Anthos component that is deployed and defines the entry point for the container to execute. This allows deploying containers from images in any GKE or Anthos environment independent from the Migrate for Anthos and GKE software. In the non-runnable layer, you can apply any changes you need to your original system. If the base image is Ubuntu, you can use apt-get; if it's SUSE, you can use Zipper; or if it's Red Hat you can use Yang. Then you can perform any copies or executional scripts you need. This means that anything you're doing in regular native containers or you were doing on these VMs on-premises, you can do on the image level and not the instance level.

Finally, when new versions of Migrate for Anthos and GKE software are released, you can update that software version in deployed workload images. Such updates might include new functionality, enhancements, or bug fixes. To update the Migrate for Anthos and GKE software layer, edit the Dockerfile and change the version tag to the

updated version you want to apply.

Updating the source code for your application

```
mvn -f src/ledgermonolith/ package
```

```
# Image containing data captured from the source VM
FROM anthos-migrate.gcr.io/v2k-run-embedded:v1.6.0 as migrate-for-anthos-runtime

# Image containing data captured from the source VM
FROM gcr.io/myproject/myworkload-non-runnable-base:v1.0.0 as source-content
COPY --from=migrate-for-anthos-runtime / /

# Migrate for Anthos image includes entrypoint
ENTRYPOINT [ "/.v2k.go" ]

# Add new application binaries to update the application code
COPY target/ledgermonolith-1.0.jar /opt/monolith/ledgermonolith.jar
```

Additionally, you can also change your source code. After you make the desired updates, you have to create the application binaries—in this case, it is a new jar file because it's a java application—and add them at the end of the Dockerfile.

Updating the source code for your application

1 Build your container:

```
docker build . -t gcr.io/$PROJECT_ID/app:newest --no-cache
```

2 Push the new container to the container registry:

```
docker push gcr.io/$PROJECT_ID/app:newest
```

3 Update the Kubernetes application:

```
kubectl set image StatefulSet/app app=gcr.io/$PROJECT_ID/app:newest
```

Google Cloud

After you update the Dockerfile, you will need to build a new workload container image version, push it to the container registry, and modify the existing Kubernetes applications to get them updated.

Deploy the migrated container on serverless Cloud Run or GKE Autopilot

```
gcloud run deploy my-runtime
    --image gcr.io/PROJECT_NAME/IMAGE_NAME:LABEL \
    --region REGION \
    --platform managed \
    --set-env-vars=HC_V2K_SERVICE_MANAGER=true \
    --port PORT
```
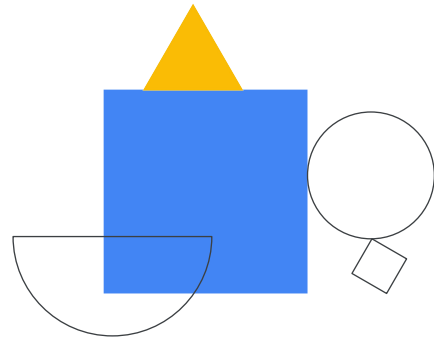
You can also replatform your container and run it in serverless offerings such as GKE Autopilot or Cloud Run, with minor changes. For example, deploying a migrated container to Cloud Run only requires the environment variable HC_V2K_SERVICE_MANAGER set to true.

Lab intro 🕐 60 min

Migrating Workloads to
Containers with Migrate for
Anthos

Google Cloud

Objectives:

- Discover and assess the workloads for migration to containers with the migration fit tool.
- Migrate and deploy the workloads into a GKE cluster using Migrate for Anthos and GKE.
- Optimize the deployment.

Questions and answers

1. Migrate for Anthos just shoves a VM into a container.

1   True

2   False

1. Migrate for Anthos just shoves a VM into a container.

| 1 | True | 2 | False |

False. M4A extracts the application user mode and data out of a VM and creates CI/CD artifacts that support image based management and day-2 operations.

2. What are the benefits of Migrate for Anthos?

1 Application density	2 Unified deployment model
3 Unified monitoring model	4 All of the above

2. What are the benefits of Migrate for Anthos?

1	Application density	2	Unified deployment model
3	Unified monitoring model	4	All of the above

3. Which applications are a good fit for Migrate for Anthos? (select 2)

1 — SAP HANA	2 — WordPress
3 — Tomcat	4 — Kernel-mode NFS apps

3. Which applications are a good fit for Migrate for Anthos? (select 2)

1 SAP HANA	**2** WordPress
3 Tomcat	**4** Kernel-mode NFS apps