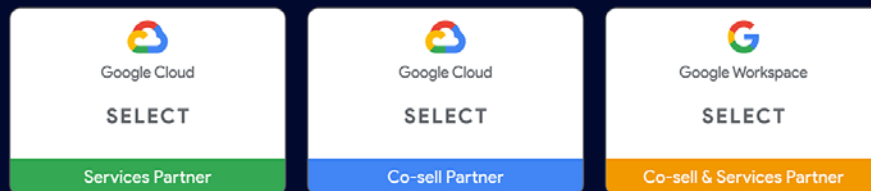


The Ultimate Google Cloud Migration Checklist



The most successful organizations are constantly seeking ways to innovate, scale, and increase efficiency. Cloud computing is a cornerstone of modern business transformation.

Google Cloud has emerged as a leader in the cloud space. It's growing faster than any other cloud platform and has the largest global network of any public cloud provider. With its robust infrastructure, cutting-edge AI/ML capabilities, and unparalleled flexibility, Google Cloud can unlock significant benefits for growing businesses.

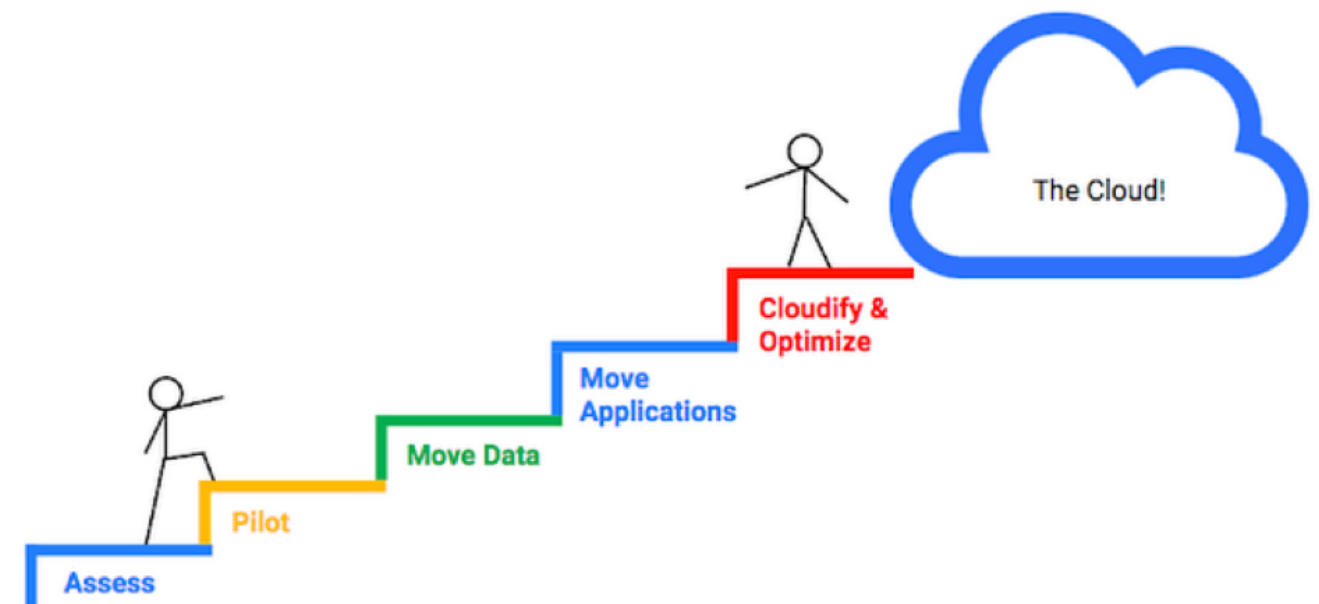
But migrating can be complicated. Without a well-structured plan and thorough understanding of the steps involved, the migration journey can be daunting and risky.

This checklist aims to simplify the migration journey.

Migration Checklist

These are key questions you should ask yourself before starting your migration:

- ✓ *How do you choose serverless product locations?*
- ✓ *What are your cost drivers?*
- ✓ *Did you check the main quotas and limits on each platform?*
- ✓ *How do your applications connect to your databases?*
- ✓ *How do you migrate your data?*
- ✓ *How do your applications integrate with other cloud services?*
- ✓ *How do you build and deploy your applications?*
- ✓ *How do you manage business continuity for your applications?*
- ✓ *How do you manage security?*
- ✓ *What compliance standards are you using?*
- ✓ *How do you handle observability?*



A Sequential Approach To Cloud Migration

Let's dive a little deeper into each question.

> How do you choose serverless product locations?

All serverless products are regional.
Need to run your apps in specific locations?
Check the location availability of your
Google Cloud serverless products.

Some products, like Cloud Storage
or Firestore, support multi-region
configurations, which you can use
for higher availability.

**You can reduce
latency by placing
all your resources
near each other.**



Google's continually growing network of global cloud infrastructure.

> What are your cost drivers?

You should always consider cost implications when tackling a migration.

Google Cloud offers a 90-day, \$300 credit trial (which isn't specifically for serverless products).

There's also a free tier for serverless products like Cloud Functions and Cloud Run. This tier never expires!

The serverless pricing model is mainly pay-as-you-go.

The usage calculation depends on the products you use.

Check out these pages for the latest Google Cloud pricing details:

- > [Google Cloud pricing page](#)
- > [Google Cloud pricing calculator](#)
- > [Google Cloud free tier information](#)

> Did you check the main quotas and limits on each platform?

Each platform has quotas and limits to prevent misconfiguration from consuming all available resources – but many of them aren't hard quotas. You can submit service requests in both clouds to raise the limits.

For detailed information, refer to > [Google's Cloud Quotas documentation](#).

> How do your applications connect to your databases?

Planning to use a NoSQL database like Firestore? Consider using Cloud SDK, which lets you connect to the database directly! It also provides Cloud Client Libraries for various popular programming language and a Google Cloud Command Line Interface (gcloud CLI) to help users perform many common cloud tasks.

Serverless services are transient. If you connect directly, there could be a lot of connections to your backend databases. SQL databases need many resources to support many connections.

The harsh reality is: There's no good connection-pooling mechanism in serverless. Your backend database could get overwhelmed. With Google Cloud, a solution is to use the CLOUD SQL Auth Proxy, which provides:

- **Secure connections** by automatically encrypting traffic to and from your database
- **Easier connection** authorization by using IAM permissions
- **IAM database authentication** by automatically refreshing OAuth 2.0 access tokens

> How do you migrate your data?

If you have existing data in AWS, you can migrate it to Google Cloud.

The migration typically has two parts:

1. TAKE A SNAPSHOT OF THE EXISTING DATA.

Then, import it to Google Cloud. You may have to change the schema and indexes or do some data transformation along the way.

2. CAPTURE THE DATA CHANGES IN AWS AND DUPLICATE THEM IN GOOGLE CLOUD.

This minimizes disruptions to your users during the migration. You'll typically change data streams and use a pipeline to transform and replicate your data.

For more guidance on your data migration, see
> [Google's S3 to Cloud Storage migration guide](#).

> How do your applications integrate with other cloud services?

Serverless applications are often integrated closely with other cloud services. Integration typically uses synchronous or asynchronous invocations to the other services.

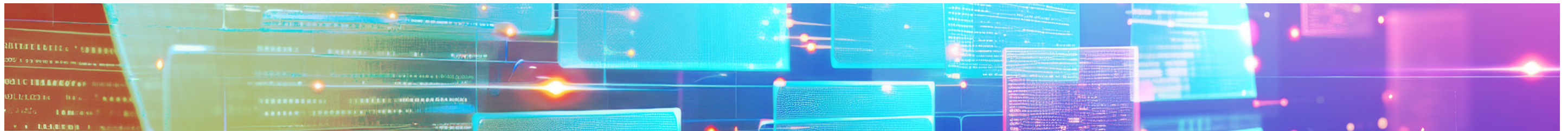
Conveniently, Google Cloud serverless products (like Cloud Run and Cloud Functions) provide native HTTPS endpoints. That means you don't have to use an API gateway or load balancer to call the services synchronously!

Asynchronous invocation is best done through events with an event-driven approach.
> [Cloud Events](#) are adopted by Google Cloud serverless products, which provide a standard for the event specification.

To orchestrate microservices in an event-driven application, you have two options:

1. USE AN ORCHESTRATOR LIKE CLOUD WORKFLOWS (FOR CENTRALLY ORCHESTRATED SERVICES).

2. FOLLOW THE CHOREOGRAPHY PATTERN USING PUB/SUB OR EVENTARC.



> **How do you build and deploy your applications?**

Serverless applications typically use the microservice architecture pattern. This comes with its own benefits and challenges.

BENEFITS OF MICROSERVICES	CHALLENGES OF MICROSERVICES
Individual services can be faster to develop, and easier to understand and maintain.	Distributed systems cause more complexity in terms of communication and handling partial failures or unavailability.
Microservices can be organized around business boundaries as opposed to a product's technical capabilities.	Microservice applications are more complex to test in local dev environments than monolithic applications.
Each microservice can be written in the language that's most effective for it, speeding up application development and reducing code complexity.	Since transactions need to be managed across different microservices, multiple databases need to be updated and data may become inconsistent if failures occur.
Since capabilities are decoupled, teams can release their microservice on independent release cycles.	Microservices architecture involves more points of failure, requires monitoring of more services, and increases latency.

On Google Cloud, you can use Cloud Code IDE plugins or run Cloud Functions and Cloud Run instances locally for development and testing. For automation, you can use Terraform, CDK for Terraform, and Pulumi along with your CI/CD tooling.



> How do you manage business continuity for your applications?

Any service in the cloud could fail, and humans make mistakes too. Therefore, you need a business continuity plan and a process to execute it.

Here are some best practices to support your business continuity in the cloud:

UNDERSTAND THE DEPENDENCIES OF YOUR APPLICATIONS.

Assess the impact if any of them fail. If there's a problem, use rate limiting or the circuit breaker pattern to limit the affected area.

MONITOR YOUR APPLICATIONS AND PROACTIVELY GENERATE RELEVANT ALERTS BEFORE PROBLEMS HAPPEN.

Ensure the redundancy of your notification channels in case of failure.

EXTERNALIZE YOUR APPLICATION STATE.

Store the state and other data in a reliable, highly available data store. Make frequent backups of your data store and exercise your disaster restore (DR) plan.

IF AVAILABLE, TAKE ADVANTAGE OF MULTI-REGION SERVICE OFFERINGS.

Cloud Run services can be deployed to multiple regions. Multi-region deployments increase your availability and decrease disruption if there's a regional outage.



> How do you manage security?

When it comes to keeping your data safe and sound in Google Cloud, there are a few key essentials to consider.

ENCRYPTION

By default, all data at rest in Google Cloud uses Google-managed default encryption. Google Cloud handles and manages this default encryption without any action on your part!

Have specific compliance or regulatory requirements to protect your data? Try using customer-managed encryption keys (CMEK) with > [Cloud Functions](#) or > [Cloud Run](#).

ACCESS CONTROLS

Google Cloud uses Identity Access Management (IAM) to control access to resources. For example, the IAM role 'roles/cloudfunctions.invoker' can be granted to a user or service account to invoke an HTTP Cloud Function.

For application authentication and authorization, Google Identity with > [Identity-Aware Proxy \(IAP\)](#) or > [Firebase Auth](#) can be used.

Remember:

IAM systems aren't identical. AWS, Azure, and Google Cloud use different frameworks. You may not be able to translate directly from a feature in one service to a feature in the other.

See

> [Google's comparison guide](#) for more details.

VPC SERVICE CONTROLS

VPC Service Controls (VPC-SC) lets you set up a service perimeter and create a data transfer boundary. You can check Google's list of > [VPC-SC supported products](#) to ensure it can protect your services.

SERVERLESS VPC ACCESS

Serverless VPC Access makes it possible to connect directly to your Virtual Private Cloud network from serverless environments like Cloud Run, App Engine, or Cloud Functions.

> What compliance standards are you using?

Your compliance responsibility is determined by:

- The type of data you have (e.g. PII, ePHI, card holder data, etc.)
- Your company's compliance requirements and objectives
- Applicable laws and regulations as related to aforementioned data

If you need to be compliant with programs like HIPAA, PCI DSS, or GDPR, see the following document:

> [Google Cloud Compliance Resource Center](#)

> How do you handle observability?

Observability is key to effectively monitor, troubleshoot, and optimize complex systems – especially microservices architectures and cloud-native applications. Make sure you contemplate the following considerations when migrating to Google Cloud.

LOGGING

On Google Cloud, all API calls for serverless products are captured in the audit logs of Cloud Logging.

MONITORING

You can find metrics for all Google Cloud products on the > [Cloud Monitoring metrics page](#).

Additionally, you can create logs-based metrics to capture metrics from logs. This gives you more flexibility when monitoring your system and establishing service level indicators and objectives (SLIs/SLOs).

DISTRIBUTED TRACING

Tracing is critical for performance insights about complex serverless applications, which typically have stateless and ephemeral microservices. On Google Cloud, use > [Cloud Trace with OpenTelemetry](#) to trace serverless applications.

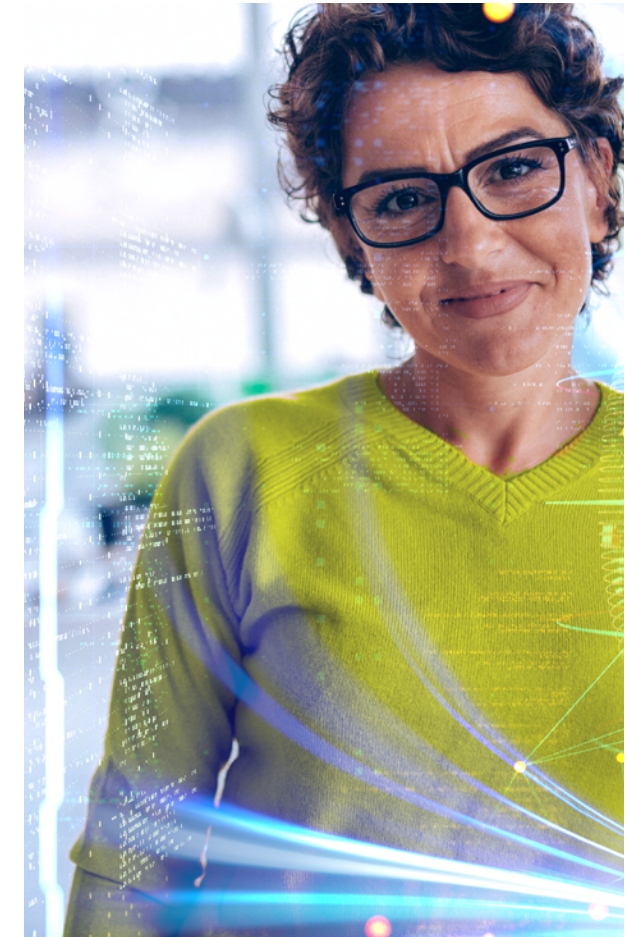


These are only a few of the key considerations you need to think about when migrating to Google Cloud. From security to performance to integration and more, there's no shortage of factors to consider.

This checklist should make the process easier. But if you want to worry less and feel more confident about your migration, consider working with a highly certified partner like RapidScale. We have 60+ Google Cloud certifications, including:

Full-stack Google Cloud Certified:

- ✓ Professional Cloud Architect
- ✓ Professional Cloud Developer
- ✓ Professional Cloud Security Engineer
- ✓ Professional Google Workspace Administrator
- ✓ Professional Machine Learning Engineer
- ✓ Professional Cloud Network Engineer
- ✓ Professional DevOps Engineer
- ✓ Professional Data Engineer



Ready to unlock the full potential of your business with Google Cloud?

Talk to one of our experts today to get a personalized consultation and discover how Google Cloud can accelerate your digital transformation journey. Talk to an expert today: meet@rapidscale.com or www.rapidscale.com

