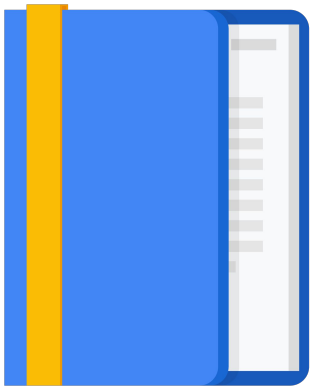
Agenda

Solution Overview

Traditional Database Architectures

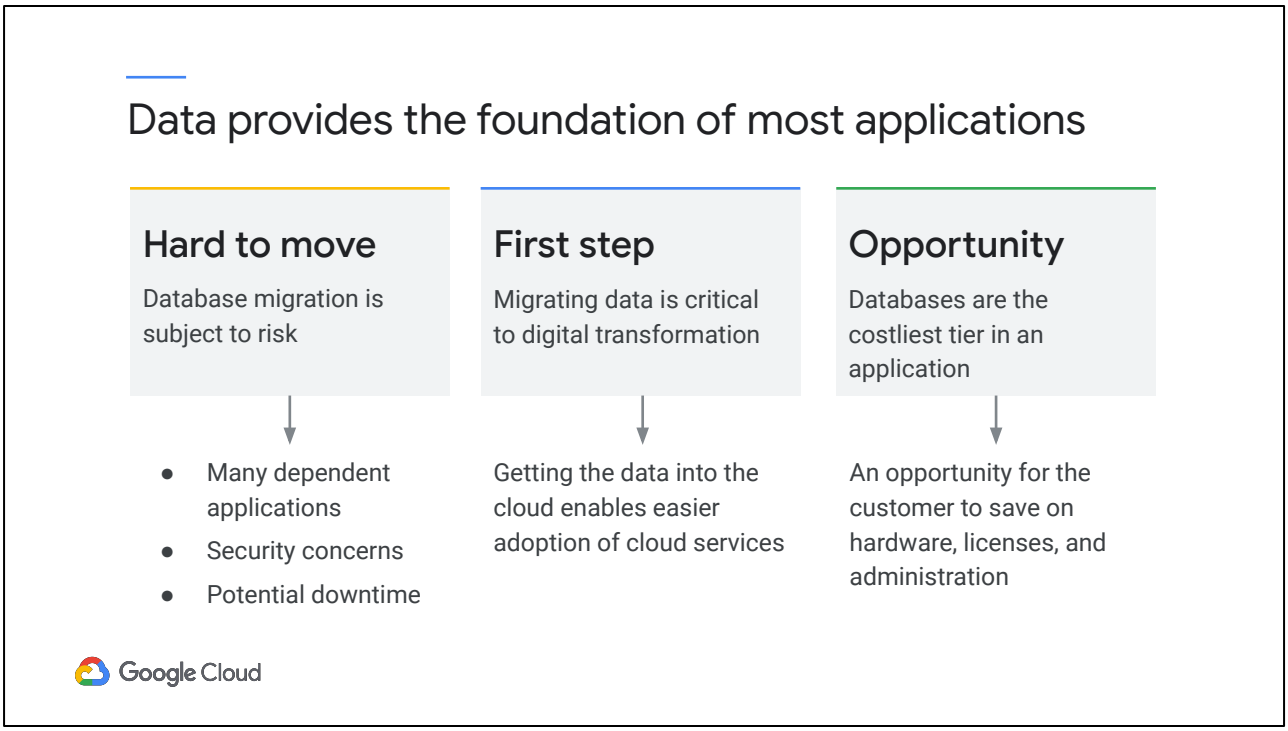
Optimizing Databases for the Cloud

Architecting Scalable and Highly Available Databases



Google Cloud

Let's get started with an overview of database migration problems and some solutions provided, not only by Google Cloud, but other providers as well.



- Databases can be the hardest part of an application to move. Often, many dependent applications rely on the database for their data. And those applications may be constantly adding new data to the database. Moving the database can break those connections.

  There are also security concerns. When hackers are attacking your systems, they are usually after your data. You need to design and architect your system in a way that protects the database, but allows dependent applications to have access.

  For some applications, defining a maintenance window and bringing the database down for that window of time is acceptable. For mission-critical applications though, downtime needs to be minimized and sometimes avoided completely. This complicates your migration project significantly. Automation, database replication, and extensive testing will be required to ensure that when you “flip the switch” to move from the old database to the new one, everything will continue to work seamlessly.

- Because the database is so important to the operation of its dependent applications, moving it to the cloud is often a critical first step when migrating applications. After the database is moved and running, moving other applications is comparatively easy.

  Customers often want to take advantage of the many services Google

provides as part of a digital transformation. When you’re running in the cloud, you can more easily take advantage of Google's advanced machine learning and big data processing services, for example.

- Moving the database can also be a big cost-saving opportunity for customers. Between hardware, maintenance, licensing, and administration, the database tier is likely the most expensive part of an application. Moving to Google Cloud can help reduce all of those costs. When their database is in the cloud, customers can work to optimize their databases and move to even cheaper, completely managed data services like BigQuery, Firestore, and Spanner.

Database is Hard to Move

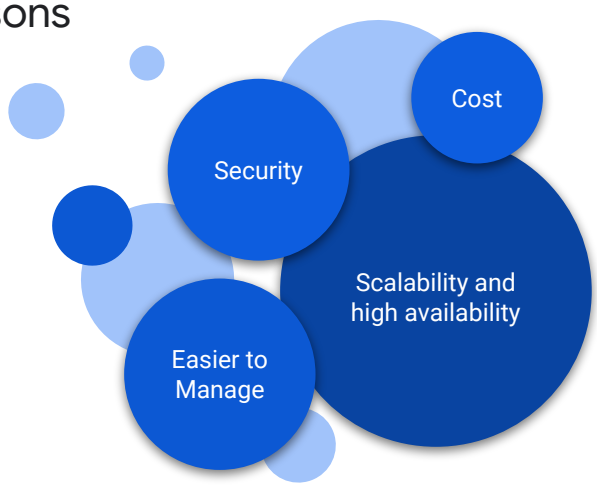
- We require
  1. Automation
  2. Database Replication
  3. Extensive Testing
- to ensure: when “flip the switch” to move from old to new DB,
  - everything will continue to work seamlessly

Moving Database can be a Big Cost-saving

- Save on
  1. hardware
  2. maintenance
  3. licensing
  4. administration

- When their database is in the cloud, customers can work to
  - optimize their databases
  - move to cheaper, completely managed services like BigQuery, Firestore, Spanner

Customers choose to move databases to the cloud for many reasons



Google Cloud

There are many great reasons customers want to move their databases and applications to Google Cloud.

- Google has data centers in regions all over the world, and each region is divided into multiple zones. You can deploy your database to multiple zones or even multiple regions for greater scalability and fault tolerance.
- Google will manage all the hardware for you, which simplifies the management of your databases. If you use a managed database service like Cloud SQL or Spanner, Google does practically all the maintenance for you.
- Sometimes people think moving to the cloud is less secure. On the contrary, Google's security is unmatched. If you know what you are doing, moving to Google Cloud can in fact enhance the security of your applications and data.
- And of course, decreasing the total cost of ownership of running your applications is often a primary driver for moving to Google Cloud.

Scalability vs Highly Available (HA)

- **`Scalable System`**
  - continues to work even as the number of users and the amount of data grow

- **`Highly Available System`**
  - continue to work even when there is a failure

Durability

- **In materials**
  - the ability to remain serviceable in the surrounding environment during the useful life without damage or unexpected maintenance

- **In databases**
  - once a transaction completes successfully,
    - its effects cannot be altered without running a compensating transaction
  - changes made by successful transaction survive subsequent failures of the system

Fault Tolerance vs High Availability (1/2)

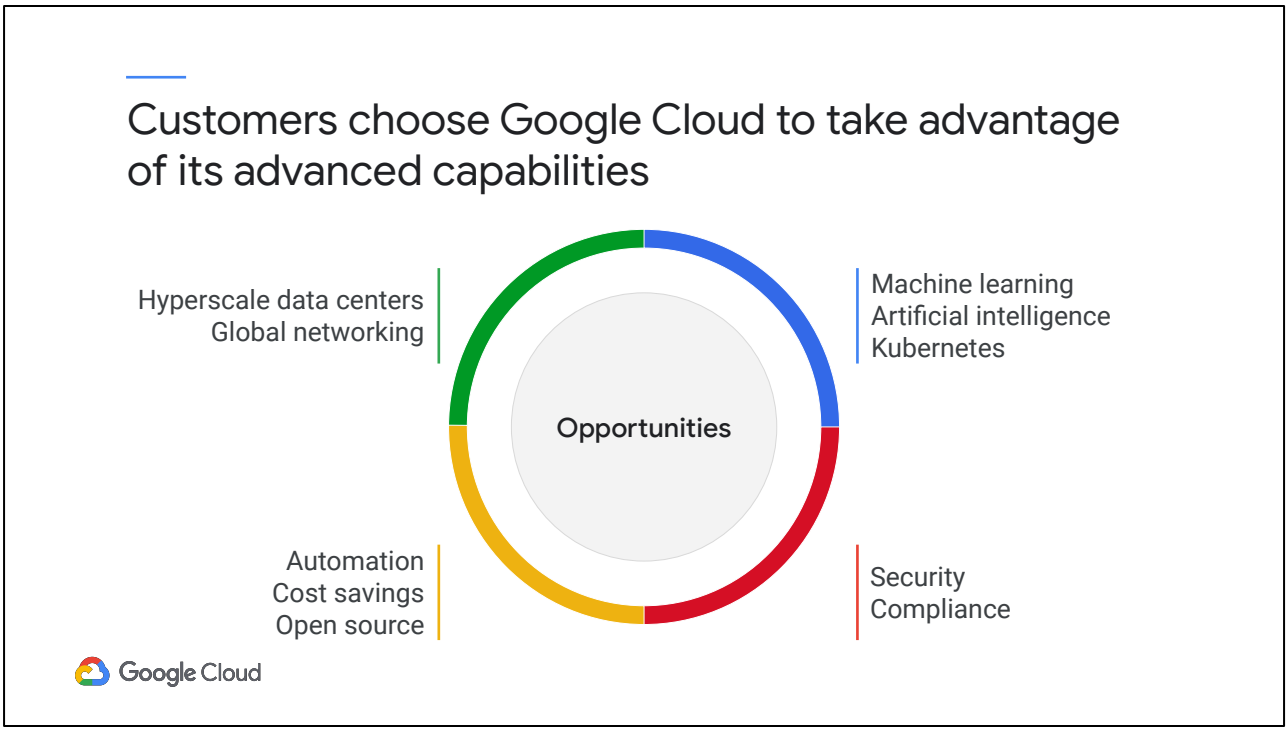
- **Fault Tolerant Environment**
  - relies on **specialized hardware** to detect **hardware fault**
  - instantaneously switch to a **redundant hardware** component
  - has no service interruption but a significantly higher cost
  - does not address **software failures**

- **Highly Available Environment**
  - services are restored rapidly (often in less than a minute) but not instantaneous
  - has a minimal service interruption
  - are excellent solution for apps that must be restored quickly
  - combines **software** with **hardware** to minimize downtime

Fault Tolerance vs High Availability (2/2)

- Many sites
  - can absorb a small amount of downtime with **HA**
  - rather than pay the much higher cost of providing **Fault Tolerance**

- Some industries
  - have apps so time-critical and cannot withstand even a few seconds of downtime
- Many other industries
  - can withstand small periods of time when their **database** is unavailable



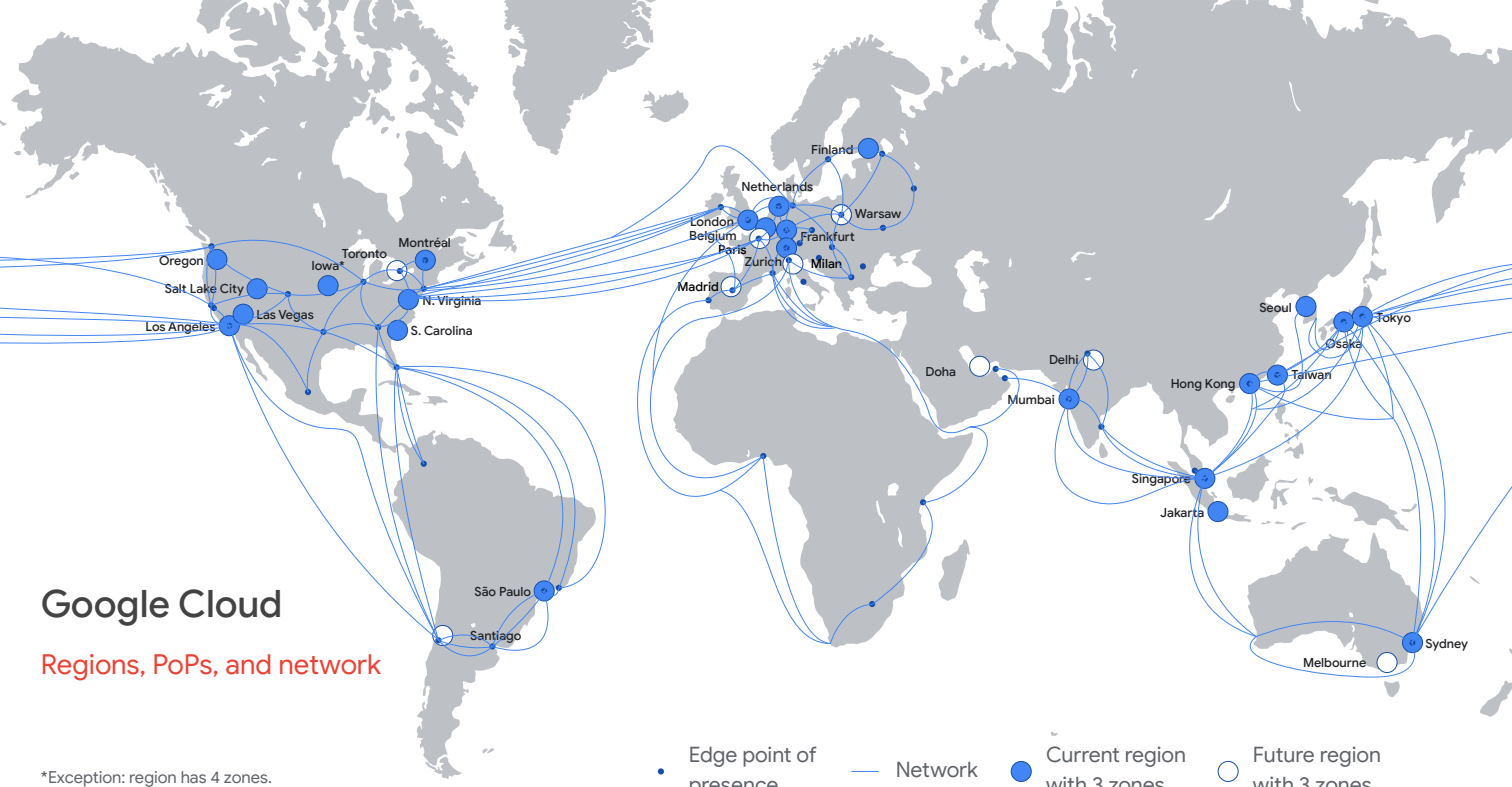
Google Cloud offers advanced capabilities that only the largest organizations could replicate in their own data centers.

- There is practically unlimited compute power and storage provided by Google's many hyperscale data centers located all around the globe. The data centers are connected by Google's fast and reliable global network.
- All resources in Google Cloud can be automated for easier management and cost savings. Google also embraces open-source technologies and is a big contributor to the open-source community. Google uses a custom Linux distribution for its own servers. Some very popular open-source technologies like TensorFlow and Kubernetes originated at Google.
- Google is a leader in advanced technologies, which is often why customers choose Google over other cloud providers. Some customers want to add machine learning capabilities to their applications using TensorFlow or use one of Google's artificial intelligence APIs. Many customers want to simplify their data center management using Kubernetes.
- Enhanced security is also an important factor when customers move to the cloud. Because Google Cloud is already certified by many government and industry compliance standards, running on Google Cloud can make compliance easier for many customers.

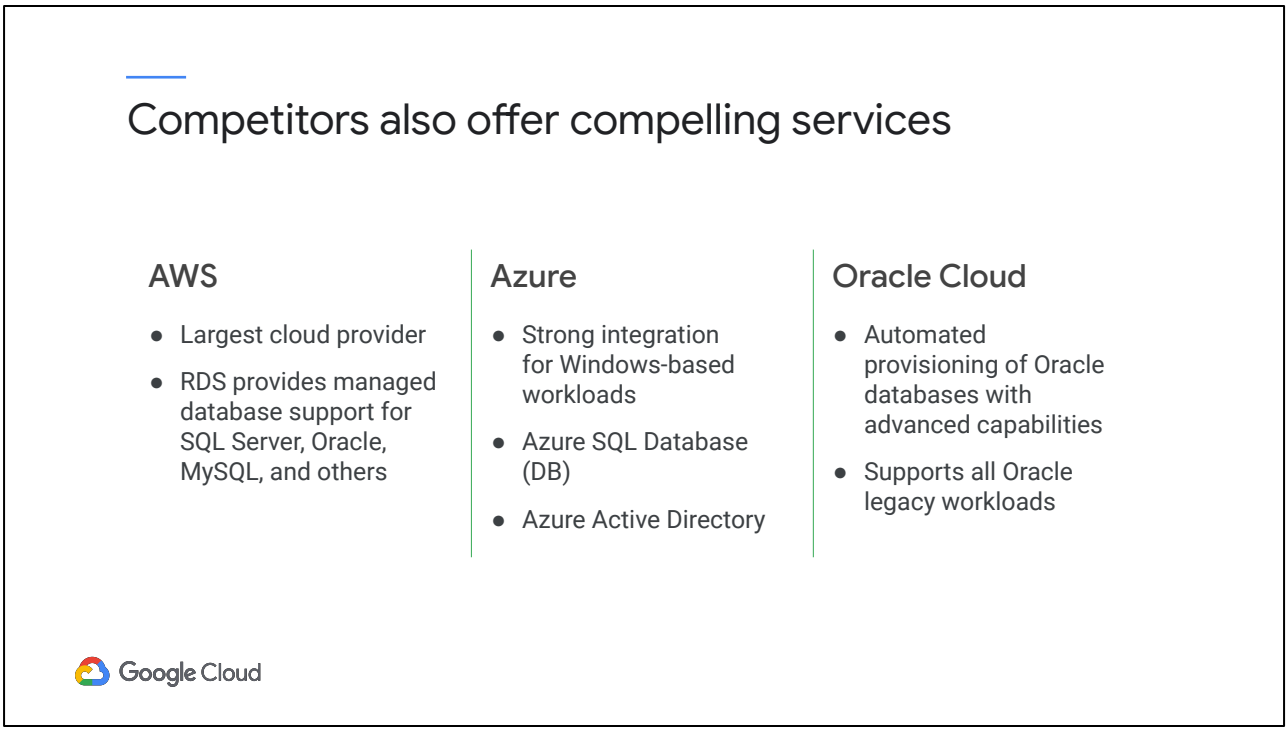
Google Cloud

- Resources (including databases) can be automated for
  - easier management
  - cost savings

- Google
  - embraces open-source technologies
  - is a big contributor to the open-source community
  - uses a custom Linux distribution for its own servers
  - created some very popular open-source technologies like TensorFlow and K8s



*Exception: region has 4 zones.



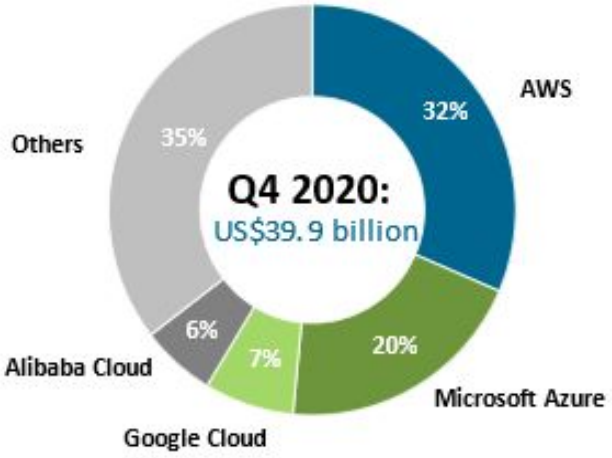
Other cloud providers, competing with Google, also offer compelling platforms and services.

- **Amazon Web Services** is one of the largest cloud providers in the world. They offer a managed database solution called RDS. RDS supports SQL Server, Oracle, MySQL, and other databases. Amazon also has a customized version of MySQL called Aurora that is optimized to run on their cloud. RDS has many advanced features like automated backups and automatic replication for high availability.

- For customers who rely heavily on Windows, **Microsoft Azure**, as you would expect, provides very strong integration with Microsoft products and tools. Azure SQL Database provides SQL Server as a service. There is also a cloud-based Azure Active Directory service. Azure also provides support for open-source technologies. You can run Linux or Windows virtual machines and many other databases like MySQL and MongoDB. Azure also supports Kubernetes through Azure Kubernetes Service.

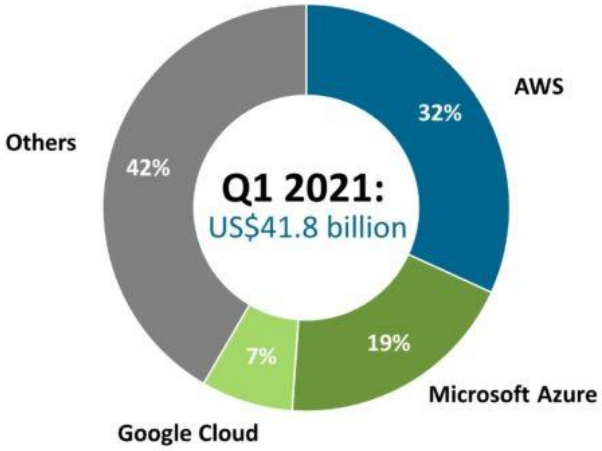
- **Oracle** also provides their own cloud for those customers who rely on Oracle databases. Oracle Cloud automates the provisioning of Oracle databases. It differs from the managed service provided by AWS RDS in that it supports all Oracle features, which RDS does not, and it supports all Oracle versions.

Cloud Providers Market-share



Source: Canalys estimates, February 2021

canalys



Source: Canalys estimates, April 2021

canalys

Amazon Web Services (AWS) 1/2

- **`Amazon Relational Database Service (Amazon RDS)`**
  - web service to facilitate: set up & operate & scale a relational database in the cloud
  - provides cost-efficient, resizeable capacity for industry-standard relational DB
  - manages common database administration tasks (managed service)
  - supports **Oracle** & **SQL Server** & **MySQL** & **MariaDB** & **PostgreSQL**
  - `has many advanced features like`
    - `automated` **`backups`**
    - `automatic replication for` **`HA`**

Amazon Web Services (AWS) 2/2

- **`Amazon Aurora`**
  - part of Amazon RDS
  - fully-managed relational database engine that's built for the cloud
  - **customized versions** of **MySQL** or **PostgreSQL** that is optimized to run on AWS

Microsoft Azure

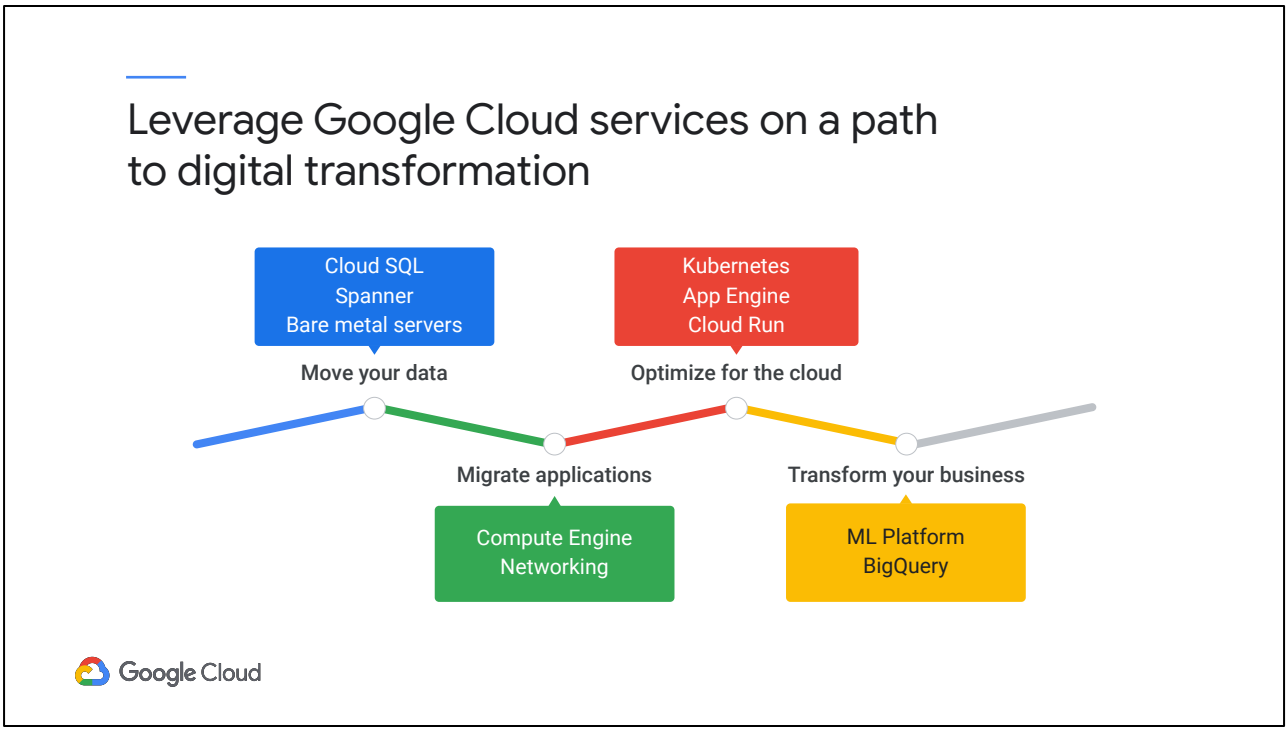
- **Azure SQL Database**
  - SQL Server as a service

- **Azure Active Directory (Azure AD)**
  - cloud-based Active Directory, which:
    - stores info about objects on the network
    - makes this info easy for admins and users to find and use

- Provides support for **open-source** technologies
  - can run many databases like **MySQL** and **MongoDB**
  - can run **Linux** or **Windows** VMs
  - supports **Kubernetes** through **Azure Kubernetes Service (AKS)**

Oracle Cloud

- `differs from AWS RDS in`
  - supports all Oracle features
  - supports all Oracle versions



In summary, customers on a path to digital transformation want to leverage Google Cloud's hyperscale data centers, global reach, and advanced services. That path starts with moving your data and databases, and then moving your applications. After you're in the cloud, you can start optimizing for the cloud and ultimately transform your business with machine learning and artificial intelligence.

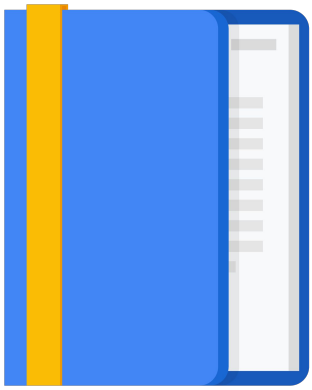
Agenda

Solution Overview

Traditional Database Architectures

Optimizing Databases for the Cloud

Architecting Scalable and Highly Available Databases

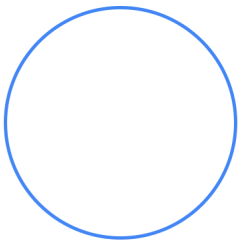


Google Cloud

In this section, you learn about the different database architectures you will encounter. In one sense, this is a history lesson on how database architectures have evolved. However, in large organizations, you might find a mix of database architectures, depending on who created the databases and what they were designed for.

Client-server databases have been a standard since the 1980s

- Highly normalized.
- Business logic provided by the database:
  - Constraints and relationships
  - Stored procedures
  - Triggers
- Client applications connect directly to the database.



Google Cloud

Client-Server databases have been a standard for many years. Client-server databases tend to be highly normalized. Also, business logic tends to be provided by the database server itself, rather than by the client. Business rules are implemented using constraints on fields and relationships between tables, as well as stored procedures and triggers. Clients connect directly to the database.

Clients in a client-server architecture were intended to be as thin as possible. The work of managing transactions and enforcing rules was the domain of the database server where this logic was centrally located and shared by the clients.

Client-server databases are fast and secure and are the preferred architecture for many DBAs.