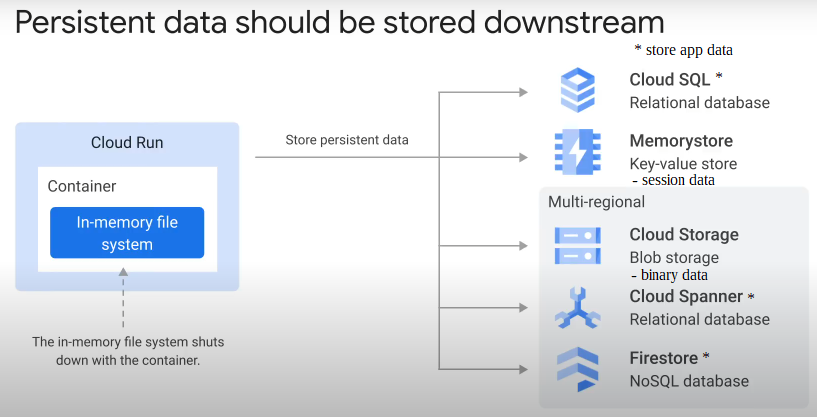
Application Development with Cloud Run

### Persisting Data

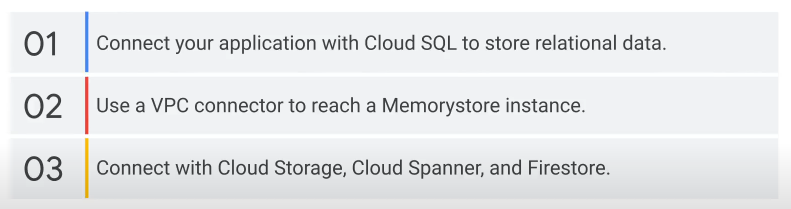
#### Introduction

Cloud Run service can scale, so backend storage must be designed to scale too.

A container in Cloud Run service has a small in-memory file system. However, does not persist beyond lifetime of container. Disappears when container shuts down. Need to persist data in downstream system.



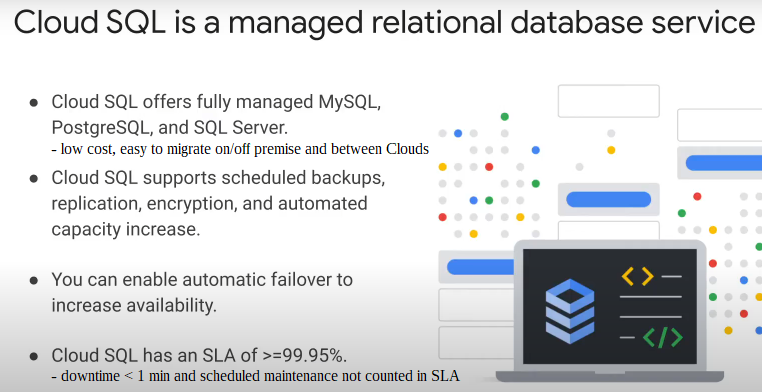
To learn in module:

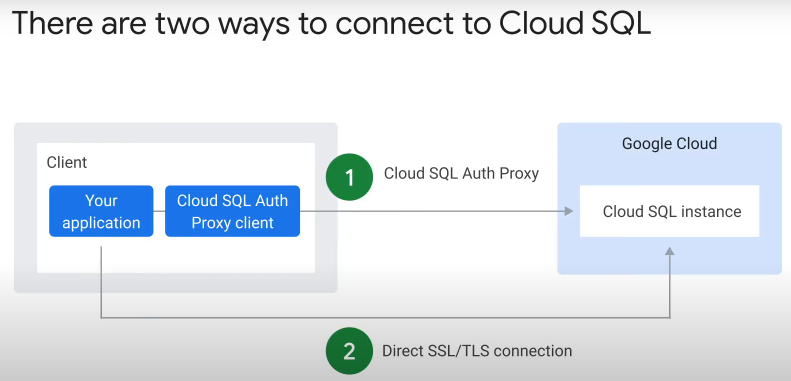


#### Connecting to Cloud SQL

https://cloud.google.com/sql/docs/introduction

https://cloud.google.com/sql/docs/key-terms

Connect Cloud Run container to Cloud SQL:

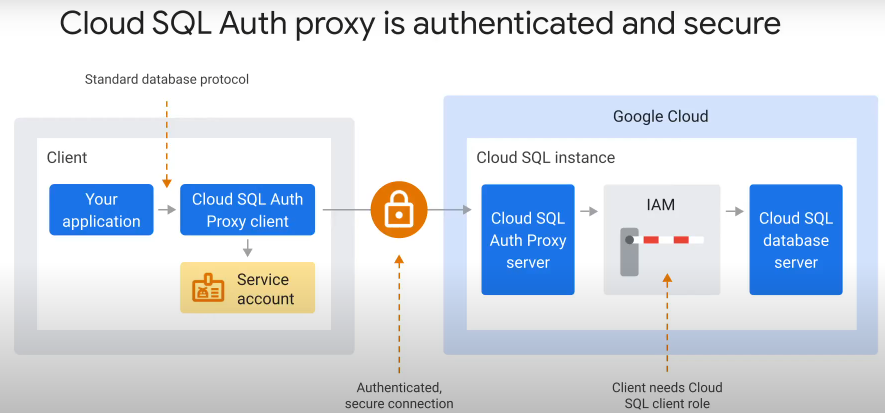
Cloud SQL Auth Proxy - separate programs to install on client and host sides

- handles authentication and encryption

Direct SSL/TLS connection – need to set up and self-manage client certificate

3rd way – unauthenticated connection, not recommended.

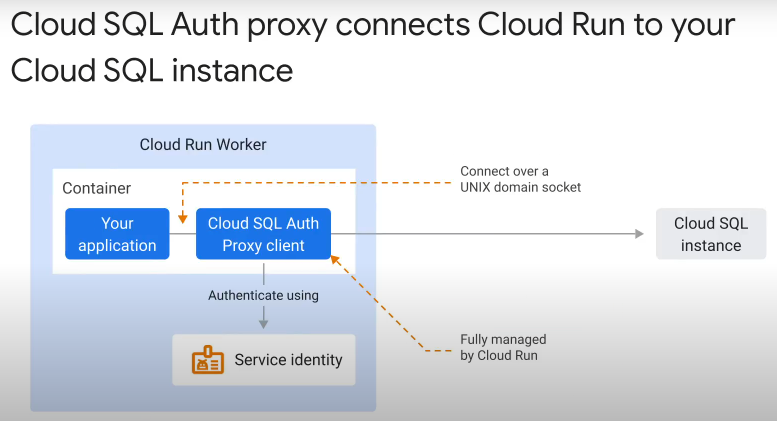
How application connects to Cloud SQL:

1. application connects locally to Cloud SQL Auth Proxy client using standard database protocol
2. client forwards connections to Cloud SQL Auth Proxy server.

* adds access token using Service Account
* connection encrypted using SSL/TLS
* connects to public IP (default) of Proxy server

1. Cloud SQL Auth Proxy server verifies access token and retrieves Service Account identity.

* checks authorization using IAM
* Service Account identity must be bound to Cloud SQL client role. IAM policy bound to resource (Cloud SQL instance)

How to set up client side:

- Cloud SQL Auth Proxy is most convenient way to connect to a Cloud SQL instance for Cloud Run.

- 3 things to do to connect Cloud Run worker to a Cloud SQL instance in same Google Cloud project:

https://cloud.google.com/sql/docs/mysql/connect-run

1. configure Cloud Run service to connect to Cloud SQL instance

* config change > new Cloud Run revision.

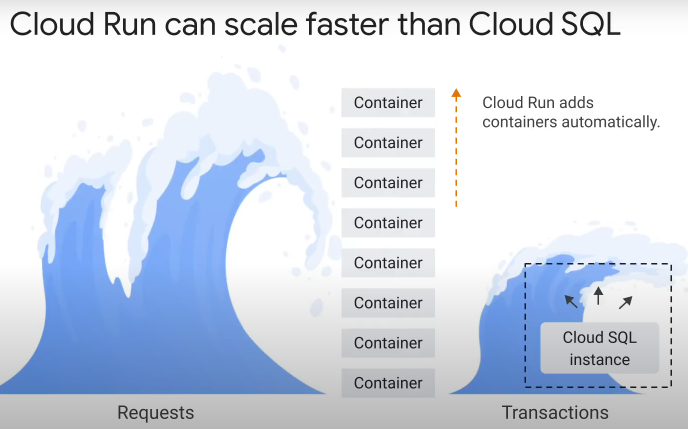
2. from application, connect to Cloud SQL instance’s UNIX domain socket

* accessed via container filesystem at path /cloudsql/INSTANCE\_CONNECTION\_NAME
* standard way to connect to databases

3. configure service account of Cloud Run service. Service account identity must have “Cloud SQL client” role for permissions needed.

Application in Cloud Run can connect directly to Cloud SQL instance using **private IP** address on port 3306, using TCP.

* Through VPC Access Connector

Scaling:

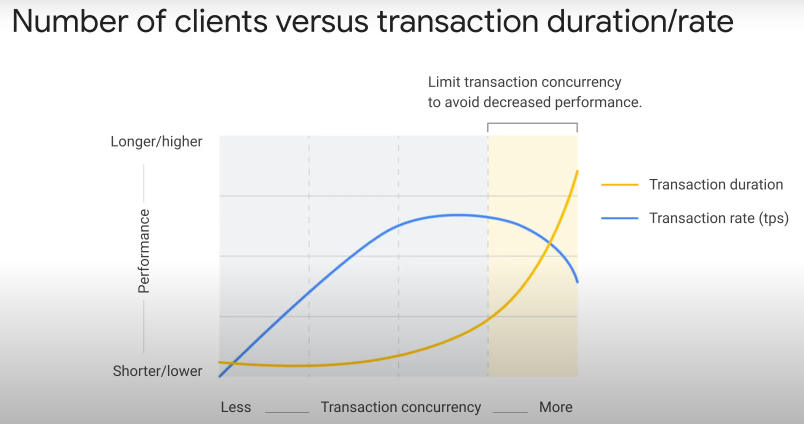
Cloud Run is regional, quota limit of 1000 instances (1vCPU/2GB memory)

If warm start or startup time short, Cloud Run can scale from 0 rapidly within 10s.

Clodu SQL is also **regional**.

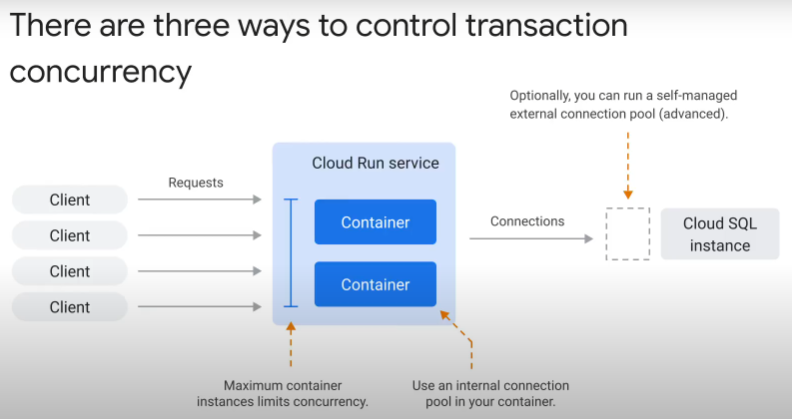
Cloud SQL max 1,000 instances per project (default).

However, Cloud SQL needs a failover to scale => slower than Cloud Run => may not be able to handle surge in requests.

Number of connections at given time (concurrency) vs Speed (rate) of transaction:

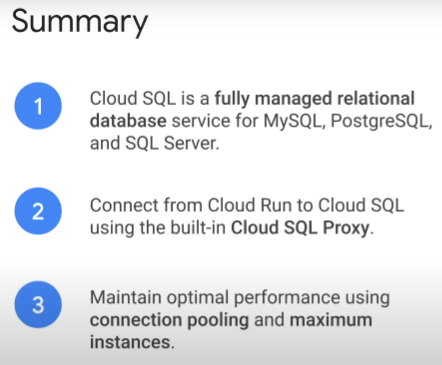
* many concurrent transactions > resource contention > transaction duration up, transaction rate (number) down > queries take longer, i.e. poorer performance
* Surge in requests > Cloud Run scales rapidly by adding containers > surge in queries to database > transactional concurrency
* Solution: manage number of concurrent transctions.
  1. Use connection pooling.
  + https://cloud.google.com/sql/docs/mysql/manage-connections
  + fixed number of connections in the pool, so upper limit on number of concurrent transactions
  1. Limit max number of container instances for Cloud Run

How to limit transaction concurrency:

* Cloud Run service:
  + add connection pool on each individual container
    - external connection pool hosted on a Compute Engine VM (advanced scenario)
  + max instance setting > limit max # of containers > limit max # of requests that can be handled at any time > request queueing when containers at max capacity > some requests timeout / rejected
* use load testing to simulate high-traffic situations > find right numbers for connection pool size and max instance count.
* closely monitor health of Cloud SQL instance in production

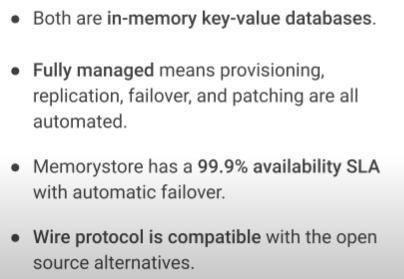
Summary:



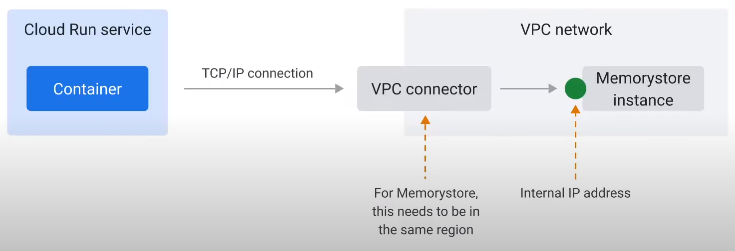


#### Connecting to MemoryStore

Fully managed Redis or Memcached:

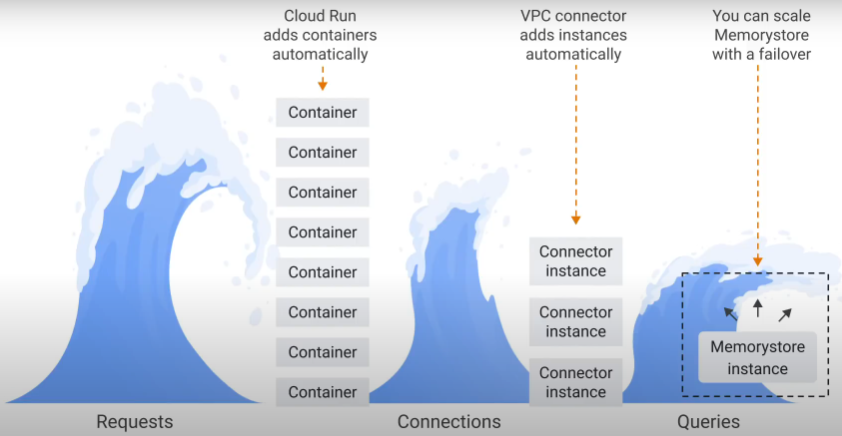


MemoryStore instances **only available within VPC at private IP address**:

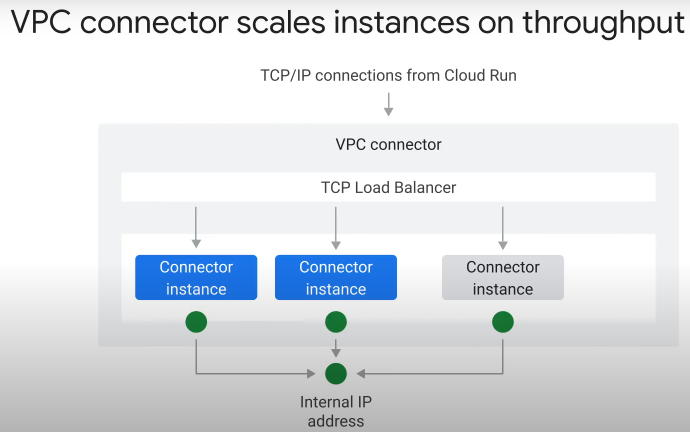
* Cloud Run not part of VPC
* need VPC Access Connector to connect to internal IP address within VPC
* VPC Connector **MUST** be in same region as MemoryStore instance
* Cloud Run service MUST be in same region as VPC Connector and MemoryStore instance
* MemoryStore not accessible directly from internet (security)
* IAM provides granular role-based access control

Auto-scaling:

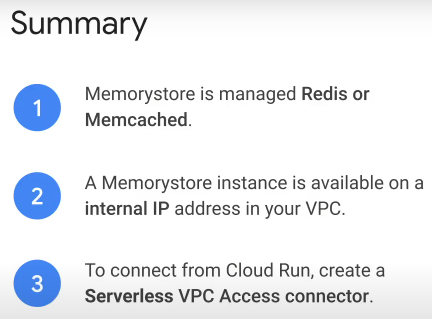
* Cloud Run (regional) can scale up to 1000 containers (1vCPU/2GB memory) in <10s
* VPC Connectors => auto-scale **up (only)** VM instances (not serverless)
* MemoryStore => scale up with failover when need more capacity



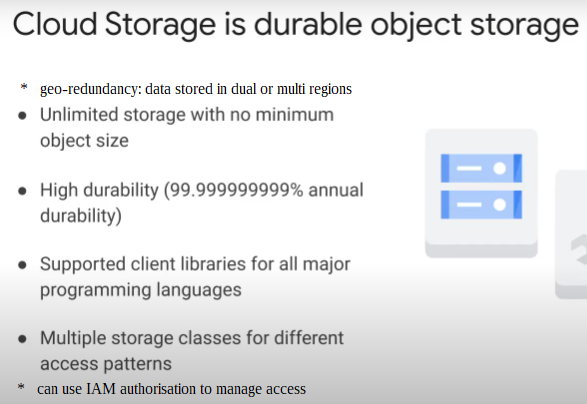
VPC Connector scales up VM instances when traffic increase:

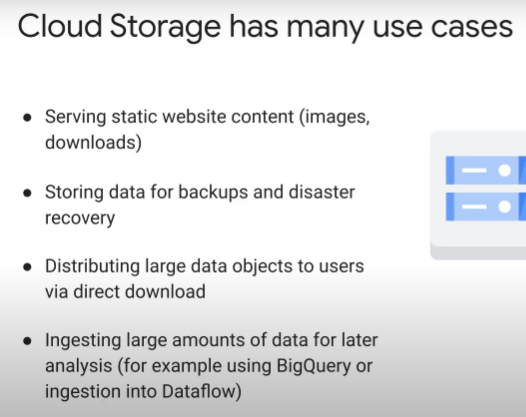


* VPC connector is a group of Virtual Machines (connector instances).
  + They forward traffic from Cloud Run to VPC network.
* If more bandwidth/throughput needed, VPC connector adds instances.
  1. You can choose machine type.
     + More memory > higher network throughput > but more expensive
     + but note that Redis is single-threaded, so there is limit how much network throughput you can increase before saturating a single Redis server instance
  2. It never remove instances to avoid breaking connections
     + scale up, but never scale down
       - not severless
     + watch expense – each connector instance added is a constant flat fee
       - can be pricy after throughput spike
* default setting for VPC connector works for most organizations.
  + Most likely no need to customize, change scaling settings
  + except large heavy user of MemoryStore

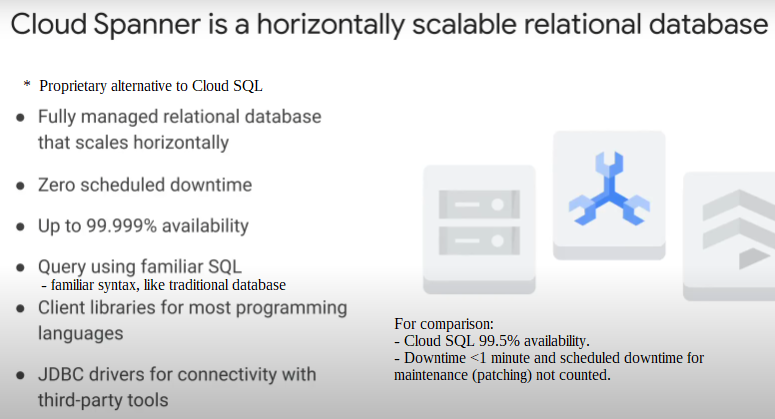


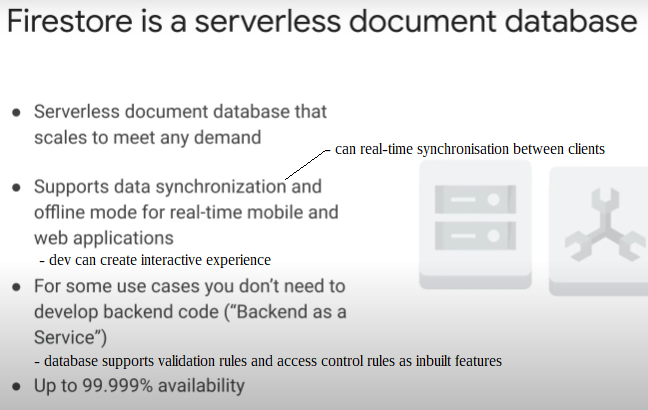
#### Cloud Storage, Cloud Spanner, Firestore

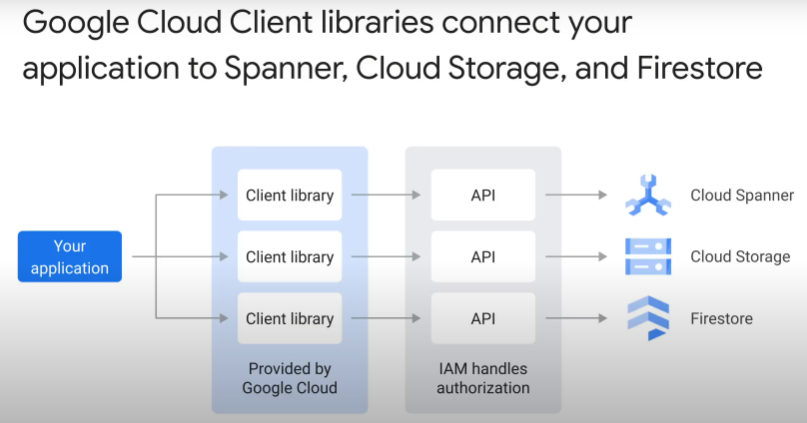




Cloud Spanner:

Firestore:



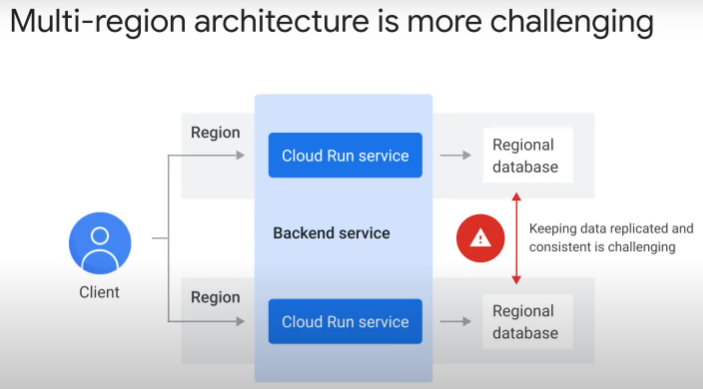


#### Multi-region data persistence

Google Cloud is regions connected by software-defined network.

* A region is a defined geographical location – a city
* A zone is a data center.

Cloud Run service is regional, i.e. all containers in one service are in single region.



* Cloud Run service can be deployed with global HTTP(S) load balancer
* will then deploy identical Cloud Run services in multiple regions
  + client traffic will be routed to Cloud Run in region nearest to them
* note that some databases are regional
  + e.g. Cloud SQL and MemoryStore
  + difficult for data to be consistent, especially if regions far apart
* consider multi-regional data stores
  + data is replicated to multiple regions; stored and accessible from >1 region.
  + within each region, data is replicated across multiple zones
  + but multi-region ≠ entire world
  + e.g. Cloud Storage, Cloud Spanner, Firestore

Benefits of mult-regional data stores:

