

This page introduces the concepts of Cloud Spanner instances, instance configurations, and nodes. It also describes the differences and tradeoffs between regional and multi-region instances. If you are not familiar with how replication works in Cloud Spanner, [read about replication \(/spanner/docs/replication\)](/spanner/docs/replication) first.

To use Cloud Spanner, you must first create a Cloud Spanner *instance* within your Google Cloud project. This instance is an allocation of resources that is used by Cloud Spanner databases created in that instance.

Instance creation includes two important choices: the *instance configuration* and the *node count*. These choices determine the location and amount of the instance's serving and storage resources. Your configuration choice is permanent for an instance, but you can change the node count later if needed.

An instance configuration defines the geographic placement and replication of the databases in that instance. When you create an instance, you must configure it as either *regional* (that is, all the resources are contained within a single Google Cloud [region \(/docs/geography-and-regions\)](/docs/geography-and-regions)) or *multi-region* (that is, the resources span more than one region). You make this choice by selecting an instance configuration, which determines where your data is stored for that instance. Regional and multi-region configurations are described in more detail below.

After you create an instance, you cannot change the configuration of that instance later.

When you create an instance, in addition to choosing where your data is stored, you must also choose the **node count**, or the number of nodes to allocate to that instance. Your choice of node

count determines the amount of serving and storage resources that are available to the databases in that instance.

Each node provides up to 2 TB of storage. The peak read and write throughput values that nodes can provide depend on the instance configuration, as well as on schema design and dataset characteristics. Refer to the [regional configuration performance](#) (#regional-performance) and [multi-region configuration performance](#) (#multi-region-performance) sections for details.

After you create an instance, you can add nodes to the instance later. In most cases, you can also remove nodes. There are a few cases in which you cannot remove nodes:

- Removing the nodes would require your instance to store more than 2 TB of data per node.
- Based on your historic usage patterns, Cloud Spanner has created a large number of [splits](#) (/spanner/docs/schema-and-data-model#database-splits) for your instance's data, and Cloud Spanner would not be able to manage the splits after removing the nodes.

You can use the [Cloud Console](https://console.cloud.google.com/spanner) (https://console.cloud.google.com/spanner), the [gcloud command-line tool](#) (/spanner/docs/gcloud-spanner), or the client libraries to change the number of nodes.

Cloud Spanner does not have a suspend mode. Cloud Spanner nodes are dedicated resources, and even when you are not running a workload, Cloud Spanner nodes frequently perform background work to optimize and protect your data.

Cloud Spanner does not automatically scale the number of nodes in your instance for you. You can set up notification for your instance with [Stackdriver Monitoring](#) (/spanner/docs/monitoring-stackdriver) to be alerted when your CPU or storage usage is above a threshold that you define.

If you need to scale up the serving and storage resources in your instance, add more nodes to that instance. Note that adding a node does not increase the number of replicas (which are fixed for a given configuration), but rather increases the resources each replica has in the instance. Adding nodes gives each replica more CPU and RAM, which increases the replica's throughput (that is, more reads and writes per second can occur). Effectively, the number of Cloud Spanner servers in each of the instance's replicas is the same as the node count of the instance. Thus, the total number of servers in a Cloud Spanner instance is the number of nodes the instance has multiplied by the number of replicas in the instance.

Google Cloud services are available in [locations](/about/locations) (/about/locations) across North America, South America, Europe, Asia, and Australia. If your users and services are located within a single region, choose a regional instance configuration for the lowest-latency reads and writes.

Cloud Spanner offers the following regional instance configurations:

Region Name	Region Description
<b>Americas</b>	
northamerica-northeast1	Montréal
southamerica-east1	São Paulo
us-central1	Iowa
us-east1	South Carolina
us-east4	Northern Virginia
us-west1	Oregon
us-west2	Los Angeles
<b>Europe</b>	
europa-north1	Finland
europa-west1	Belgium
europa-west2	London
europa-west3	Frankfurt
europa-west4	Netherlands
europa-west6	Zürich
<b>Asia Pacific</b>	
asia-south1	Mumbai
asia-east1	Taiwan

Region Name	Region Description
asia-east2	Hong Kong
asia-northeast1	Tokyo
asia-northeast2	Osaka
asia-southeast1	Singapore
australia-southeast1	Sydney

For any regional configuration, Cloud Spanner maintains 3 [read-write replicas](#) (/spanner/docs/replication#read-write), each within a different Google Cloud [zone](#) (/docs/geography-and-regions) in that region. Each read-write replica contains a full copy of your operational database that is able to serve read-write and read-only requests. Cloud Spanner uses replicas in different zones so that if a single-zone failure occurs, your database remains available.

Regional configurations contain exactly 3 read-write replicas. As described in [Cloud Spanner replication](#) (/spanner/docs/replication), every Cloud Spanner mutation requires a write quorum that's composed of a majority of voting replicas. Write quorums are formed from two out of the three replicas in regional configurations.

For optimal performance, follow these best practices:

- [Design a schema](#) (/spanner/docs/schema-design) that prevents hotspots and other performance issues.
- Place critical compute resources within the same region as your Cloud Spanner instance.
- Provision enough Cloud Spanner nodes to keep high priority total CPU utilization under 65%.

When you follow the [best practices described above](#) (#regional-best-practices), each Cloud Spanner node can provide up to 10,000 queries per second (QPS) of reads or 2,000 QPS of writes (writing single rows at 1 KB of data per row).

These performance numbers are estimates only. Cloud Spanner performance is highly dependent on workload, schema, and dataset characteristics.

As described above, Cloud Spanner regional configurations replicate data between multiple zones within a single region. However, if your application often needs to read data from multiple geographic locations (for example, to serve data to users in both North America and Asia), or if your writes originate from a different location than your reads (for example, if you have large write workloads in North America and large read workloads in Europe), then a regional configuration might not be optimal.

Multi-region configurations allow you to replicate the database's data not just in multiple zones, but in multiple zones across multiple regions, as defined by the instance configuration. These additional replicas enable you to read data with low latency from multiple locations close to or within the regions in the configuration. There are tradeoffs though, because in a multi-region configuration, the quorum (read-write) replicas are spread across more than one region. Hence, they can incur additional network latency when these replicas communicate with each other to vote on writes. In other words, multi-region configurations enable your application to achieve faster reads in more places at the cost of a small increase in write latency.

Configuration Name	Configuration Location	Default Leader Region	Additional Read-Write Region	Read-Only Regions
nam3	North America	us-east4	us-east1	None
nam6	North America	us-central1	us-east1	us-west1 us-west2
eur3	Europe	eu-west1	eu-west4	None

Configuration Name	Configuration Locations	Default Leader Region	Additional Read-Write Region	Read-Only Regions
nam-eur-asia1	North America, Europe, and Asia	us-centra11	us-centra12 (Oklahoma – <i>private Google Cloud region</i> )	europa-west1 asia-east1

Multi-region instances offer these primary benefits:

- **99.999% availability**, which is greater than the 99.99% availability that Cloud Spanner regional configurations provide.
- **Data distribution**: Cloud Spanner automatically replicates your data between regions with strong consistency guarantees. This allows your data to be stored where it's used, which can reduce latency and improve the user experience.
- **External consistency**: Even though Cloud Spanner replicates across geographically distant locations, you can still use Cloud Spanner as if it were a database running on a single machine. Transactions are guaranteed to be serializable, and the order of transactions within the database is the same as the order in which clients observe the transactions to have been committed. External consistency is a stronger guarantee than "strong consistency," which is offered by some other products. Read more about this property in [TrueTime and external consistency](#) (/spanner/docs/true-time-external-consistency).

Each multi-region configuration contains two regions that are designated as **read-write regions**, each of which contains two read-write replicas. One of these read-write regions is designated as the *default leader region*, which means that it contains your database's leader replicas. Cloud Spanner also places a witness replica in a third region called a **witness region**.

Each time a client issues a mutation to your database, a write quorum forms, consisting of one of the replicas from the default leader region and any two of the additional four voting replicas. (The quorum could be formed by replicas from two or three of the regions that make up your configuration, depending on which other replicas participate in the vote.) In addition to these 5 voting replicas, the configuration can also contain read-only replicas for serving low-latency reads. The regions that contain read-only replicas are called **read-only regions**.

In general, the voting regions in a multi-region configuration are placed geographically close—less than a thousand miles apart—to form a low-latency quorum that enables fast writes ([learn more](#)

(</spanner/docs/replication#aside-why-read-only-and-witness-replicas>). However, the regions are still far enough apart—typically, at least a few hundred miles—to avoid coordinated failures.

The next sections describe each of these region types in more detail and provide guidance for how to place your write and read workloads accordingly.

As described above, each multi-region configuration contains two read-write regions, each of which contains two [read-write replicas](/spanner/docs/replication#read-write) (</spanner/docs/replication#read-write>). One of these read-write regions is designated the default leader region where leader replicas are placed. Under normal conditions when all replicas are available, the default leader region contains the leaders and therefore is where writes are first processed. In the event of leadership failure, other replicas in the default leader region automatically assume leadership. In fact, leaders run health checks on themselves and can preemptively give up leadership if they detect they are unhealthy.

The second read-write region contains the additional replicas that are eligible to be leaders. In the unlikely event of the loss of the default leader region, new leader replicas are chosen from the second read-write region.

**int:** Place the bulk of your read and write workloads in the default leader region. In the event of the loss of a default , read and write workloads are fastest from the second read-write region.

Read-only regions contain [read-only replicas](/spanner/docs/replication#read-only) (</spanner/docs/replication#read-only>), which can serve low-latency reads to clients that are outside of the read-write regions.

**int:** Place additional read workloads in read-only regions.

A witness region contains a [witness replica](/spanner/docs/replication#witness) (</spanner/docs/replication#witness>), which is used to vote on writes. Witnesses become important in the rare event that the read-write regions become unavailable.

**int:** You can think of the witness region as a system-configured region for achieving write quorums.

For optimal performance, follow these best practices:

- [Design a schema](/spanner/docs/schema-design) (/spanner/docs/schema-design) that prevents hotspots and other performance issues.
- For optimal write latency, place compute resources for write-heavy workloads within or close to the default leader region.
- For optimal read performance outside of the default leader region, use staleness of at least 15 seconds.
- To avoid single-region dependency for your workloads, place critical compute resources in at least two regions.
- Provision enough Cloud Spanner nodes to keep high priority total CPU utilization under 45% in each region.

Each Cloud Spanner configuration has slightly different performance characteristics based on the replication topology.

When you follow the [best practices described above](#) (#multi-region-best-practices), each Cloud Spanner node can provide the following approximate performance:

Multi-Region Configuration	Approximate Peak Read (QPS per region)	Approximate Peak Writes (QPS total)
nam3	7,000	1,800
nam-eur-asia1	7,000	1,000

Note that read guidance is given per region (because reads can be served from anywhere), while write guidance is for the entire configuration. Write guidance assumes that you're writing single rows at 1 KB of data per row.

These performance numbers are estimates only. Cloud Spanner performance is highly dependent on workload, schema, and dataset characteristics.



Configuration	Availability	Latency	Cost	Data Locality
Regional	99.99%	Lower write latencies within region.	Lower cost; see <a href="/spanner/pricing">pricing</a> (/spanner/pricing).	Enables geographic data governance.
Multi-region	99.999%	Lower read latencies from multiple geographic regions.	Higher cost; see <a href="/spanner/pricing">pricing</a> (/spanner/pricing).	Distributes data across multiple regions within the configuration.

- [Create a Cloud Spanner instance](/spanner/docs/create-manage-instances#create-instance) (/spanner/docs/create-manage-instances#create-instance).
- Learn more about [Google Cloud regions and zones](/docs/geography-and-regions) (/docs/geography-and-regions).