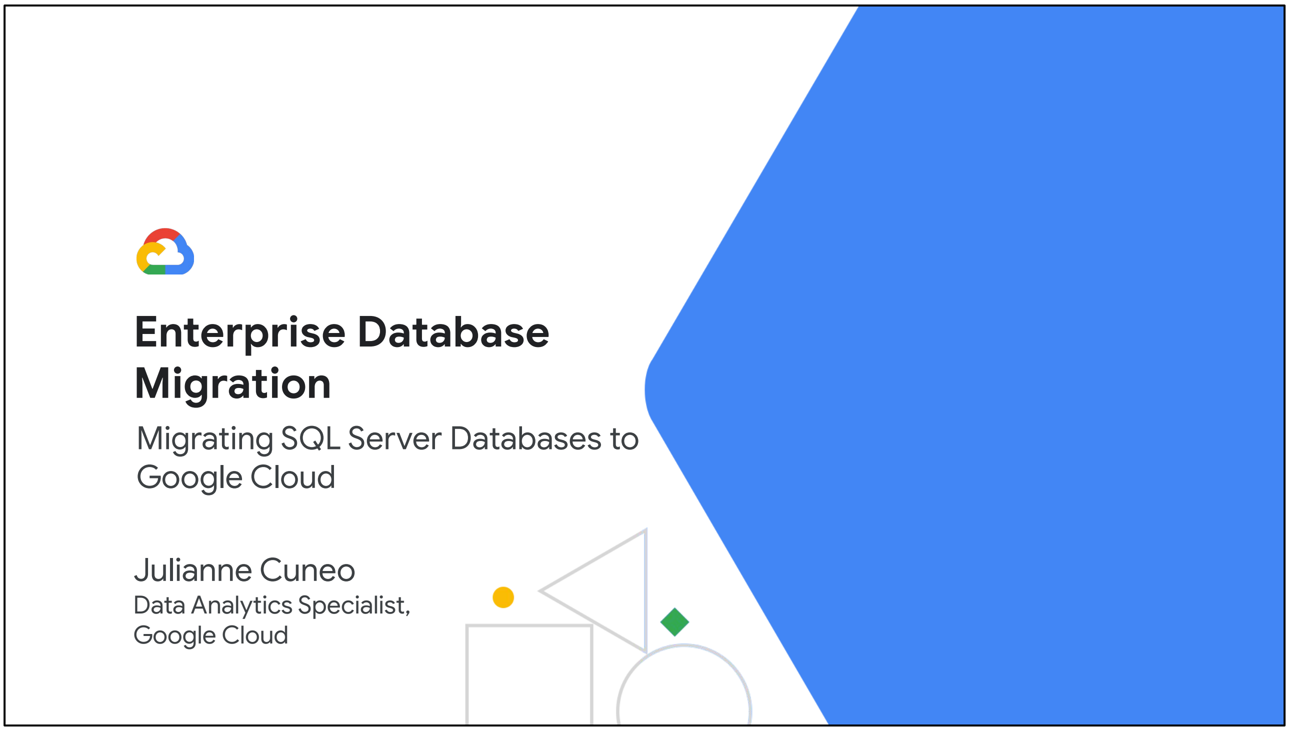
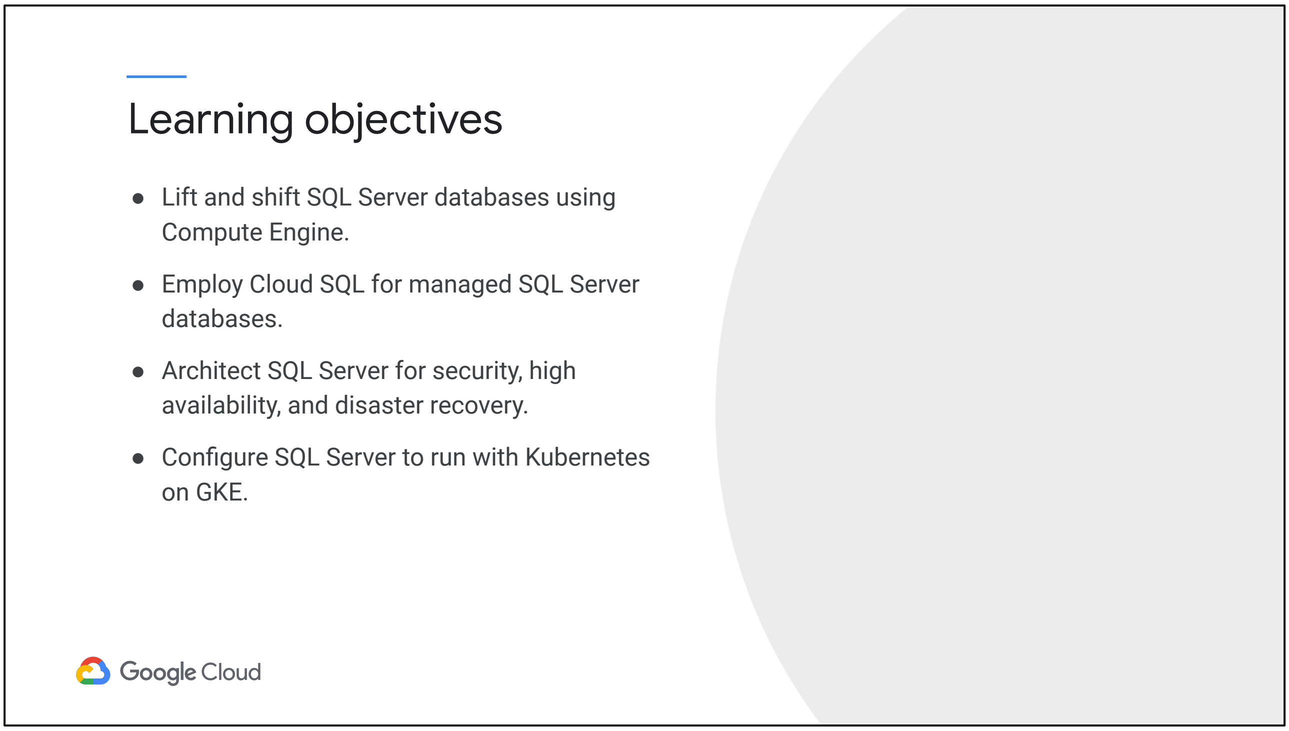


Enterprise Database Migration

Migrating SQL Server Databases to Google Cloud

Julianne Cuneo
Data Analytics Specialist,
Google Cloud

Hello everyone, Julianne here. There are several options available for migrating a SQL Server database to Google Cloud. So Let’s explore those in some detail.



Learning objectives

- Lift and shift SQL Server databases using Compute Engine.
- Employ Cloud SQL for managed SQL Server databases.
- Architect SQL Server for security, high availability, and disaster recovery.
- Configure SQL Server to run with Kubernetes on GKE.

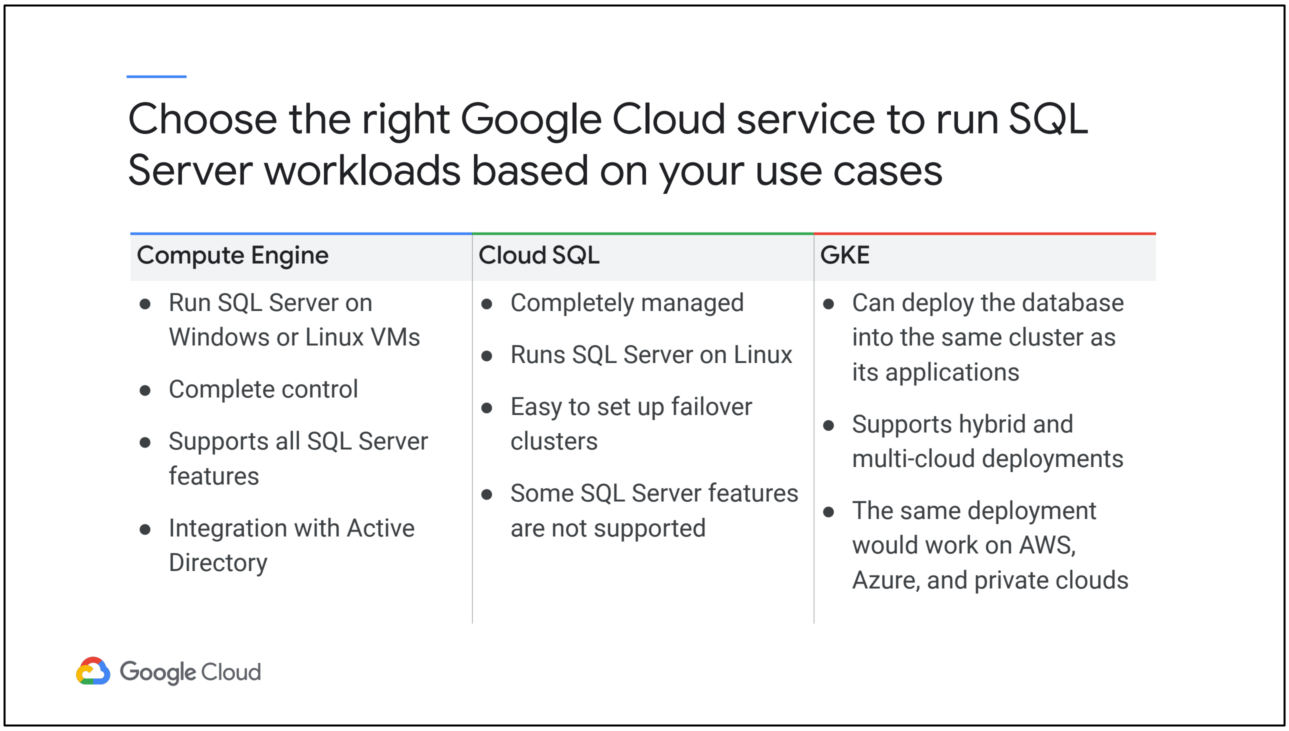
Google Cloud

A lift and shift strategy involves picking up and moving virtual machines into Google Cloud. You could employ this strategy with your SQL Server databases by using Compute Engine.

Alternatively, you could use a managed version of SQL Server provided by Cloud SQL.

In any case, you also need to architect your SQL Server databases for security, high availability, and disaster recovery.

You can also run SQL Server in a Kubernetes environment using Google Kubernetes Engine.



Choose the right Google Cloud service to run SQL Server workloads based on your use cases

Compute Engine	Cloud SQL	GKE
• Run SQL Server on Windows or Linux VMs • Complete control • Supports all SQL Server features • Integration with Active Directory	• Completely managed • Runs SQL Server on Linux • Easy to set up failover clusters • Some SQL Server features are not supported	• Can deploy the database into the same cluster as its applications • Supports hybrid and multi-cloud deployments • The same deployment would work on AWS, Azure, and private clouds

Google Cloud

Choose the right Google Cloud service to run SQL Server workloads based on your use cases. Each of the three services offer advantages over the others.

- **Compute Engine** offers the most straightforward and familiar option with minimal changes from your existing SQL Server installation. You basically build a custom SQL Server with the exact features you want, just as you would on an on-premises machine. But it also means DBAs have to do all the same work they would on a local server.

- **Cloud SQL** offers all the advantages of a managed service and removes some of the administrative burden from the DBAs; however, not all SQL Server features are supported, primarily because it's running on Linux. So it's critical to ensure that none of the unsupported features are important to you before you choose this option.

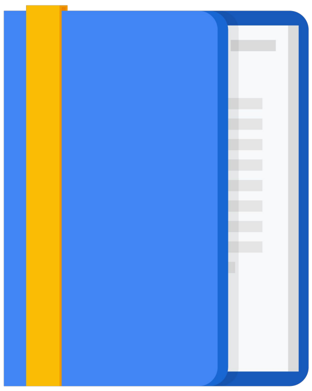
- **GKE** offers flexibility in deploying the database into the same cluster as your applications, as well as the ability to create hybrid deployments and portability to other cloud environments.

Agenda

SQL Server on Compute Engine

SQL Server on Cloud SQL

SQL Server on GKE

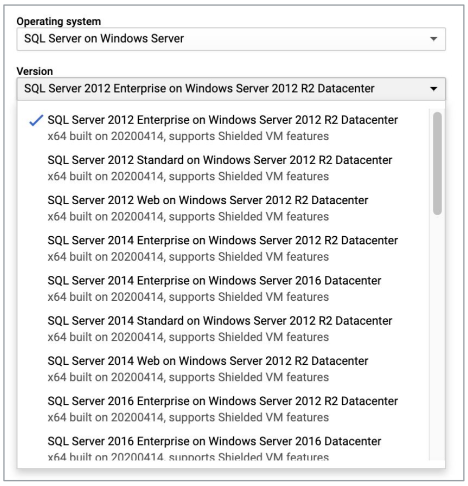


Google Cloud

SQL Server on Compute Engine is basically just relocating the server from a local machine to a Compute Engine VM. This is often the easiest way to do the migration itself, but may not be the best long-term option.

Google provides pre-built SQL Server images

- Versions included: 2012 through 2019
- Editions included:
  - Express
  - Web
  - Standard
  - Enterprise
- See: https://cloud.google.com/compute/docs/instances/windows#sql_server



Operating system
SQL Server on Windows Server

Version
SQL Server 2012 Enterprise on Windows Server 2012 R2 Datacenter

✓ SQL Server 2012 Enterprise on Windows Server 2012 R2 Datacenter
x64 built on 20200414, supports Shielded VM features

SQL Server 2012 Standard on Windows Server 2012 R2 Datacenter
x64 built on 20200414, supports Shielded VM features

SQL Server 2012 Web on Windows Server 2012 R2 Datacenter
x64 built on 20200414, supports Shielded VM features

SQL Server 2014 Enterprise on Windows Server 2012 R2 Datacenter
x64 built on 20200414, supports Shielded VM features

SQL Server 2014 Enterprise on Windows Server 2016 Datacenter
x64 built on 20200414, supports Shielded VM features

SQL Server 2014 Standard on Windows Server 2012 R2 Datacenter
x64 built on 20200414, supports Shielded VM features

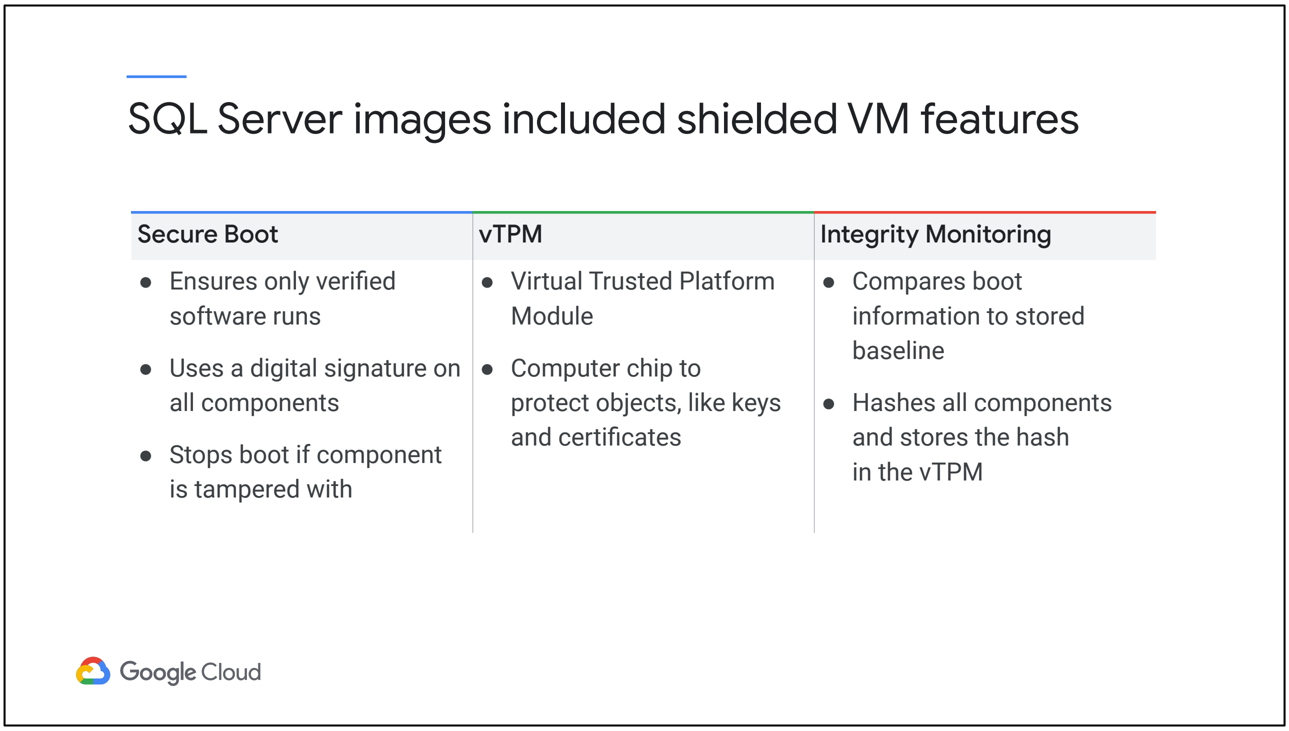
SQL Server 2014 Web on Windows Server 2012 R2 Datacenter
x64 built on 20200414, supports Shielded VM features

SQL Server 2016 Enterprise on Windows Server 2012 R2 Datacenter
x64 built on 20200414, supports Shielded VM features

SQL Server 2016 Enterprise on Windows Server 2016 Datacenter
x64 built on 20200414, supports Shielded VM features

Google Cloud

When you create the Compute Engine VM, you could start with a bare Windows image and install your own SQL Server databse onto it, but Google offers pre-built versions of SQL Server in many different versions, which simplifies the whole process of creating the VM.



SQL Server images included shielded VM features

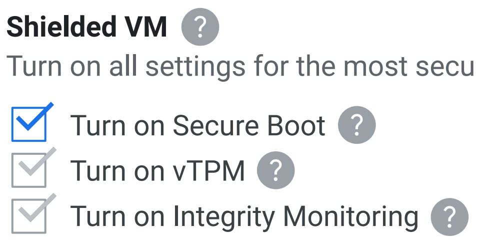
Secure Boot	vTPM	Integrity Monitoring
• Ensures only verified software runs • Uses a digital signature on all components • Stops boot if component is tampered with	• Virtual Trusted Platform Module • Computer chip to protect objects, like keys and certificates	• Compares boot information to stored baseline • Hashes all components and stores the hash in the vTPM

Google Cloud

Using the Google-provided images also offers benefits beyond convenience during the build process. They provide a higher level of security against attempts to hack your server by providing mechanisms to detect and prevent unauthorized changes to the software and to protect keys and certificates.

- **Secure boot** ensures that only verified software runs on your VM. It uses digital signatures to verify software components and stops the boot process if something has been tampered with.
- Shielded VMs use a custom chip called a Virtual Trusted Platform Module or **vTPM** to protect digital signatures.
- And **Integrity monitoring** compares boot information to a stored baseline hash that is stored in the vTPM.

Enable Shielded VM features when configuring the virtual machine



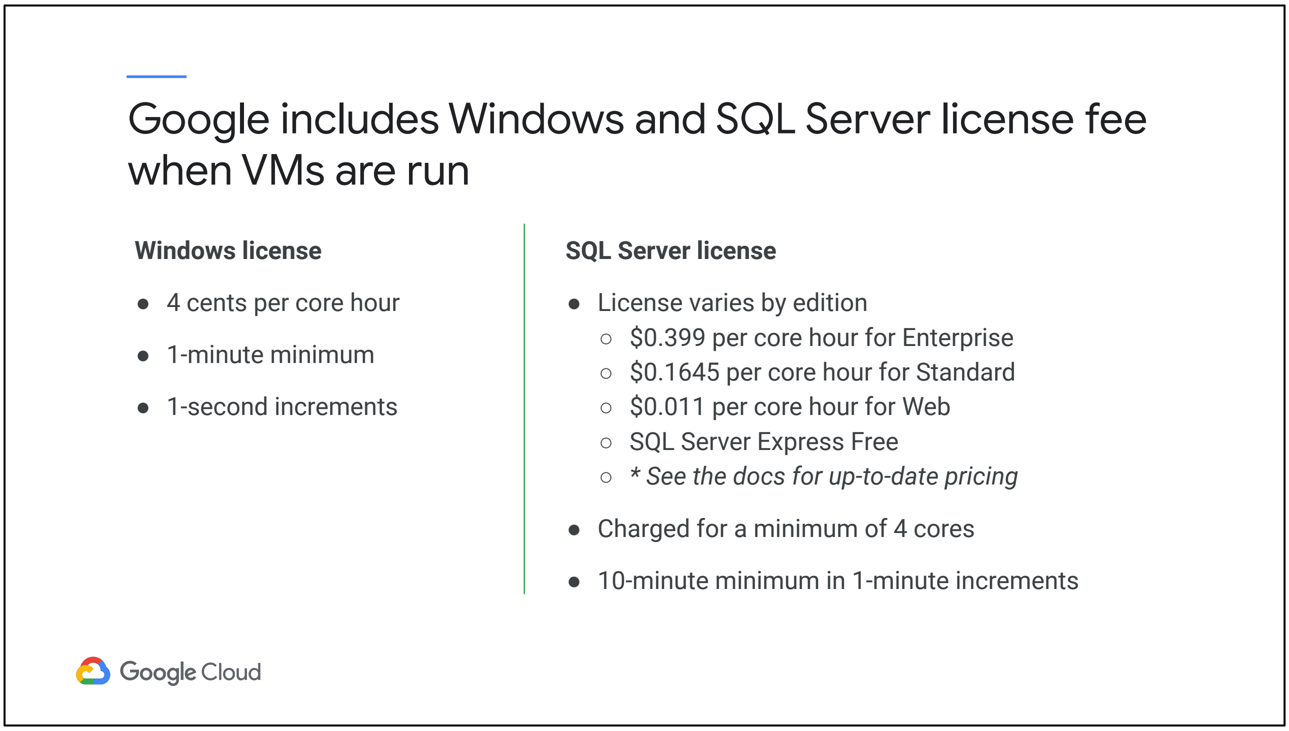
Shielded VM

Turn on all settings for the most secure configuration.

- [✓] Turn on Secure Boot
- [✓] Turn on vTPM
- [✓] Turn on Integrity Monitoring

Google Cloud

The Shielded VM features are available when you configure the VM, and you can enable just the ones you want.



Google includes Windows and SQL Server license fee when VMs are run

Windows license

- 4 cents per core hour
- 1-minute minimum
- 1-second increments

SQL Server license

- License varies by edition
  - $0.399 per core hour for Enterprise
  - $0.1645 per core hour for Standard
  - $0.011 per core hour for Web
  - SQL Server Express Free
  - ** See the docs for up-to-date pricing*
- Charged for a minimum of 4 cores
- 10-minute minimum in 1-minute increments

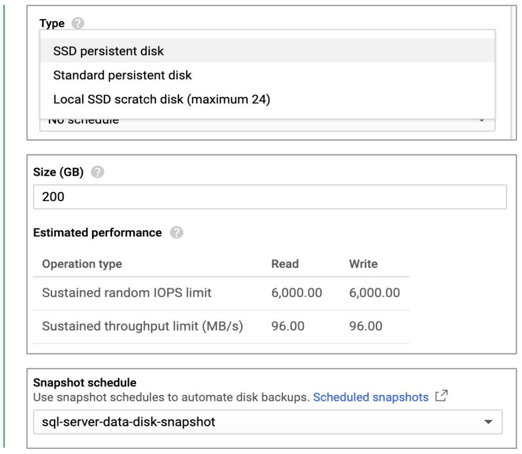
Google Cloud

Another advantage of using the pre-built Google images is that the license cost of using Windows and SQL Server is built into the running costs of the machine. The pricing varies depending on which edition of SQL Server you choose and how many cores you configure the VM for. Still, it is often a lot easier to manage than manually paying for your own SQL Server license.

However, if you already pay for licenses, there is a Microsoft License Mobility for Google Cloud program. You can See the documentation for more information.

Add additional disks to store SQL Server data when creating the virtual machine

- Choose SSD or Standard disks:
  - SSD is faster.
  - Standard is cheaper.
- Disks can be 10 GB to 64 TB:
  - Larger disks are faster.
- Can set up a snapshot schedule for automated backups.



Type

SSD persistent disk

Standard persistent disk

Local SSD scratch disk (maximum 24)

No schedule

Size (GB)

200

Estimated performance

Operation type	Read	Write
Sustained random IOPS limit	6,000.00	6,000.00
Sustained throughput limit (MB/s)	96.00	96.00

Snapshot schedule

Use snapshot schedules to automate disk backups. Scheduled snapshots

sql-server-data-disk-snapshot

Google Cloud

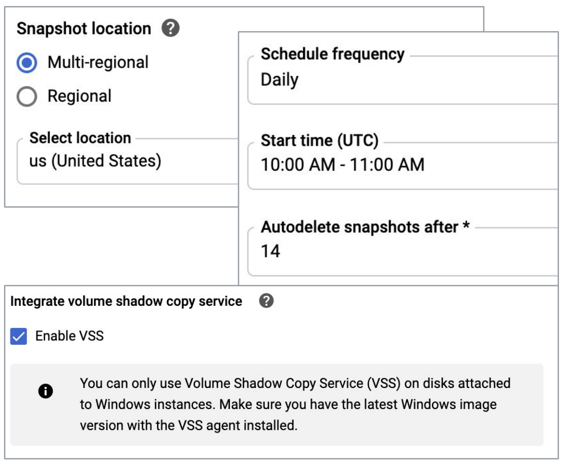
When configuring the VM, you also need to allocate storage space. This is done by adding data disks. As usual, they can be SSDs for faster performance or standard disks for cost savings. You can configure the size to meet your needs, but remember that you pay for what you allocate, not what you use.

Disks can be very large: up to 64 terabytes.

And You can also schedule automatic disk snapshots to make disk backups, but remember that this is different from a SQL Server backup.

Use Snapshot Schedules to automatically backup disks

- Choose the regions you want the snapshots to be stored in.
- Schedule snapshots during off-peak hours.
- For Windows machines, enable Volume Shadow Copy Service:
  - Safely creates snapshots without needing machines to be off.

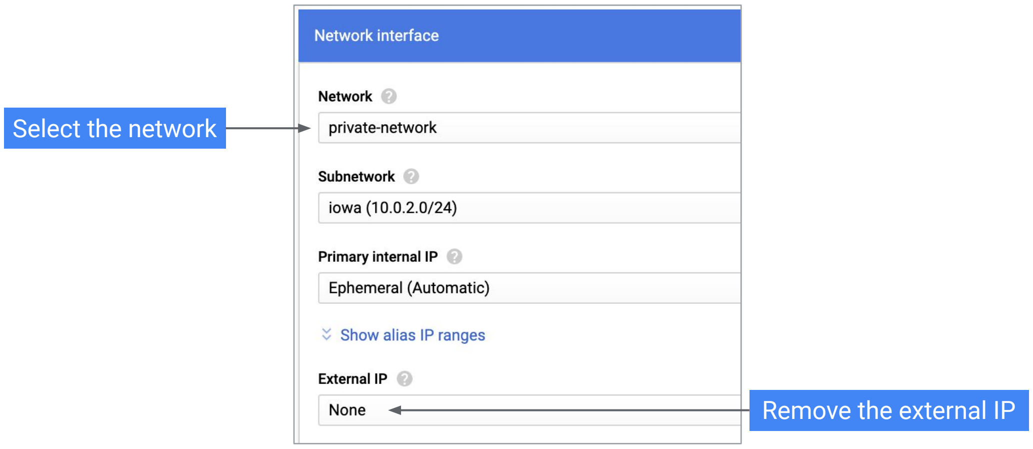

Snapshot location
[✓ Multi-regional] Regional
Select location: us (United States)
Schedule frequency: Daily
Start time (UTC): 10:00 AM - 11:00 AM
Autodelete snapshots after *: 14
Integrate volume shadow copy service
☑ Enable VSS
You can only use Volume Shadow Copy Service (VSS) on disks attached to Windows instances. Make sure you have the latest Windows image version with the VSS agent installed.

Google Cloud

Snapshot backups can be configured to be stored in one or multiple regions and on any schedule that suits you. It's best to schedule the snapshots at off-peak hours.

Also, if you're running a Windows instance, you can choose the Volume Shadow Copy Service to allow snapshots without shutting down the SQL Server service.

Configure the VM network interface to provide secure access to the SQL Server



Network interface

Network: private-network

Subnetwork: iowa (10.0.2.0/24)

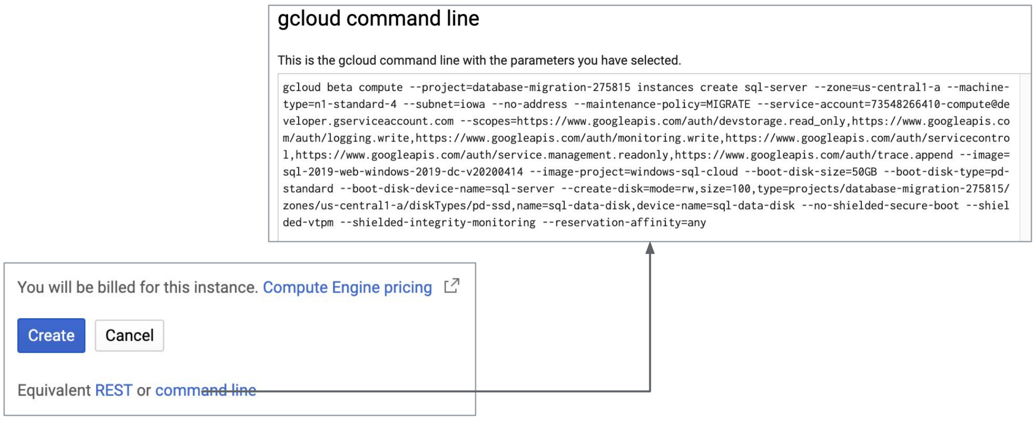
Primary internal IP: Ephemeral (Automatic)

External IP: None

Google Cloud

Ultimately, you will have a SQL Server VM that clients will need to connect to. So in order to reach the server, you need to configure the networking. Typically, you don’t want an external IP address, but instead want to make a private network for the clients to access the server. This provides more security.

Use the Console to configure the machine, but automate its creation using code



gcloud command line

This is the gcloud command line with the parameters you have selected.

```
gcloud beta compute --project=database-migration-275815 instances create sql-server --zone=us-central1-a --machine-
type=n1-standard-4 --subnet=iowa --no-address --maintenance-policy=MIGRATE --service-account=73548266410-compute@de
veloper.gserviceaccount.com --scopes=https://www.googleapis.com/auth/devstorage.read_only,https://www.googleapis.co
m/auth/logging.write,https://www.googleapis.com/auth/monitoring.write,https://www.googleapis.com/auth/servicecontro
l,https://www.googleapis.com/auth/service.management.readonly,https://www.googleapis.com/auth/trace.append --image=
sql-2019-web-windows-2019-dc-v20200414 --image-project=windows-sql-cloud --boot-disk-size=50GB --boot-disk-type=pd-
standard --boot-disk-device-name=sql-server --create-disk=mode=rw,size=100,type=projects/database-migration-275815/
zones/us-central1-a/diskTypes/pd-ssd,name=sql-data-disk,device-name=sql-data-disk --no-shielded-secure-boot --shiel
ded-vtpm --shielded-integrity-monitoring --reservation-affinity=any
```

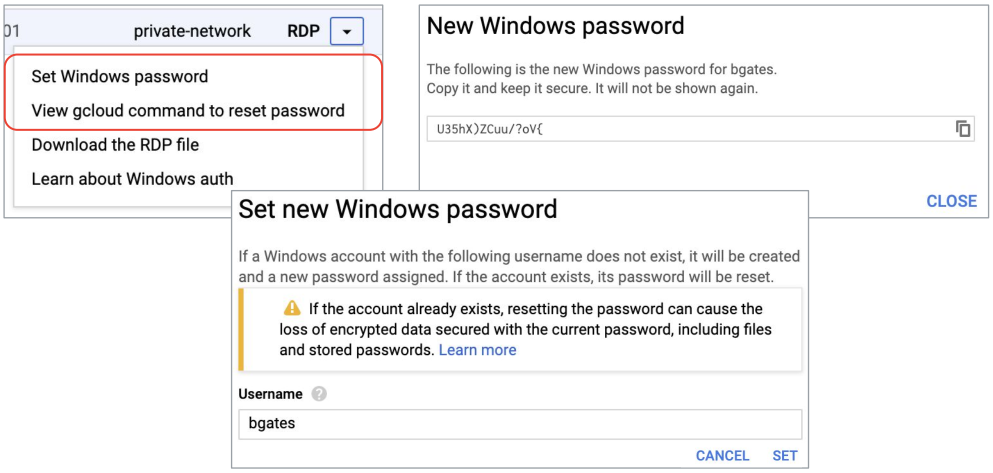
You will be billed for this instance. Compute Engine pricing

Equivalent REST or command line

Google Cloud

As usual, you can use the console to enter the options to configure the VM, but it's a good idea to copy the generated script to do the actual VM creation. As much as possible, automate your migration process so it is repeatable and testable. You can write shell scripts using the command line commands. As you saw earlier, you could also use Terraform.

Setting a Windows username and password to log in


01 private-network RDP

Set Windows password

View gcloud command to reset password

Download the RDP file

Learn about Windows auth

New Windows password

The following is the new Windows password for bgates.
Copy it and keep it secure. It will not be shown again.

```
U35hX)ZCuu/?oV{
```

CLOSE

Set new Windows password

If a Windows account with the following username does not exist, it will be created and a new password assigned. If the account exists, its password will be reset.

⚠ If the account already exists, resetting the password can cause the loss of encrypted data secured with the current password, including files and stored passwords. Learn more

Username

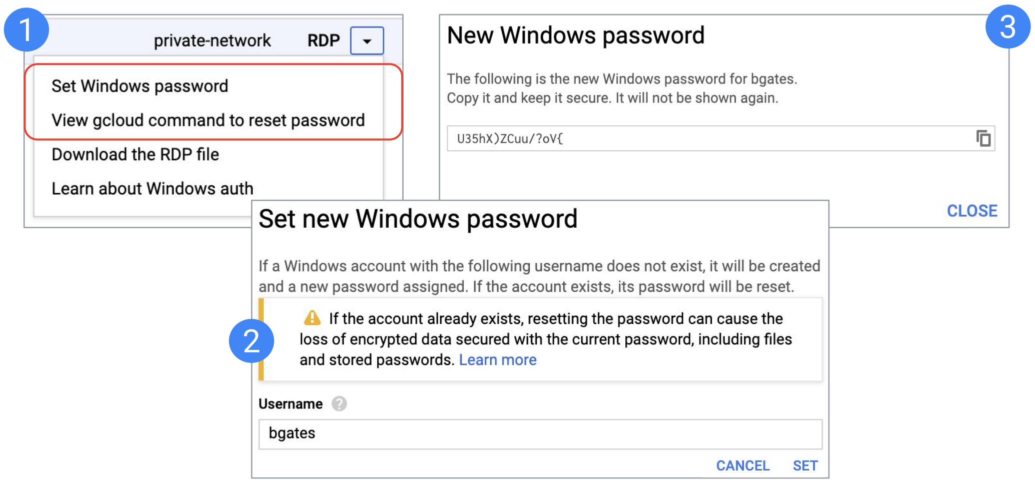
bgates

CANCEL SET

Google Cloud

To administer a Windows server, you will use RDP. In order to RDP into the server, you will need to set a Windows username and password.

Setting a Windows username and password to log in



New Windows password

The following is the new Windows password for bgates.
Copy it and keep it secure. It will not be shown again.

```
U35hX)ZCuu/?oV{
```

Set new Windows password

If a Windows account with the following username does not exist, it will be created and a new password assigned. If the account exists, its password will be reset.

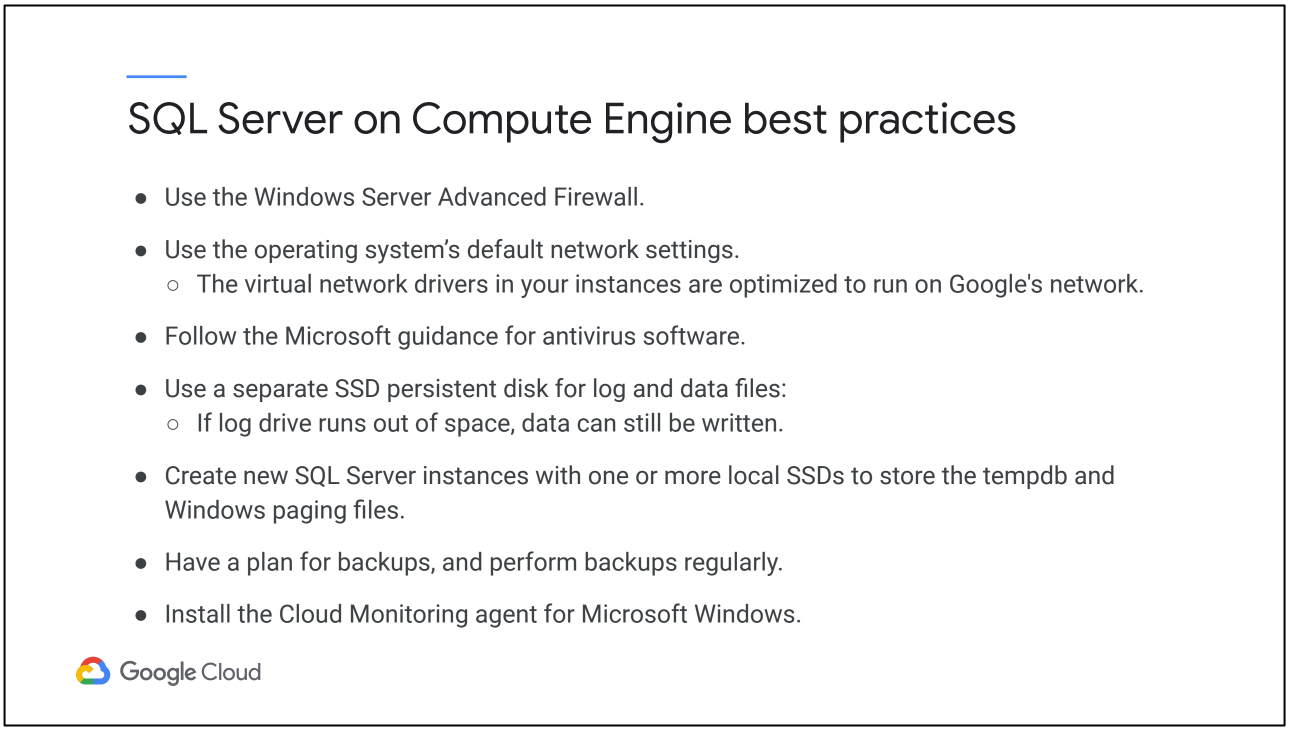
⚠ If the account already exists, resetting the password can cause the loss of encrypted data secured with the current password, including files and stored passwords. Learn more

Username: bgates

Google Cloud

1. This can be done from the console or the CLI.
2. Specify any username you like,
3. and a strong password will be generated.

As with all usernames and passwords, you will need to save or remember these. You can always change the password after logging in.



SQL Server on Compute Engine best practices

- Use the Windows Server Advanced Firewall.
- Use the operating system's default network settings.
  - The virtual network drivers in your instances are optimized to run on Google's network.
- Follow the Microsoft guidance for antivirus software.
- Use a separate SSD persistent disk for log and data files:
  - If log drive runs out of space, data can still be written.
- Create new SQL Server instances with one or more local SSDs to store the tempdb and Windows paging files.
- Have a plan for backups, and perform backups regularly.
- Install the Cloud Monitoring agent for Microsoft Windows.

Google Cloud

There are many best practice recommendations, but they are all pretty straightforward.

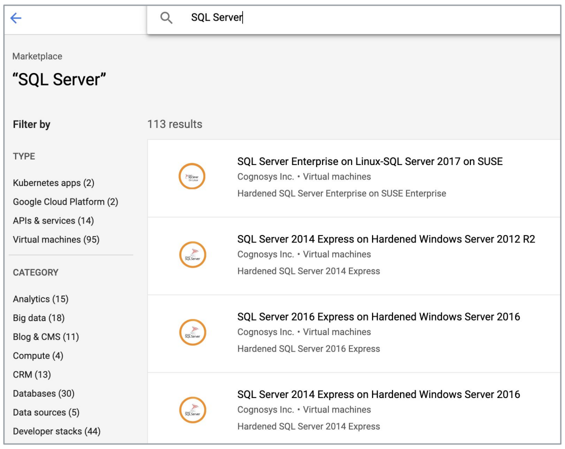
You want to follow the typical best practices for creating a VM, which include using a firewall and the default network settings, as well as installing anti-virus software.

Additionally, you would configure the SQL Server portion in a way that's similar to how you would on a local machine. Namely, you would want separate disks for the log and data files, as well as for tempdb and the Windows paging files.

Using SSDs is more expensive but offers better performance. If you can't use SSDs for all the drives, it's probably most important to use them for the tempdb and paging files first.

Also, although you have the server in the cloud and have turned on snapshots, you can't skip normal SQL Server maintenance. You still need to set up regular database and log backups and schedule them with SQL Server Agent.

You can also search Marketplace for SQL and find pre-configured VMs on both Windows and Linux



SQL Server

Marketplace

"SQL Server"

Filter by

113 results

TYPE

Kubernetes apps (2)
Google Cloud Platform (2)
APIs & services (14)
Virtual machines (95)

CATEGORY

Analytics (15)
Big data (18)
Blog & CMS (11)
Compute (4)
CRM (13)
Databases (30)
Data sources (5)
Developer stacks (44)

SQL Server Enterprise on Linux-SQL Server 2017 on SUSE
Cognosys Inc. • Virtual machines
Hardened SQL Server Enterprise on SUSE Enterprise

SQL Server 2014 Express on Hardened Windows Server 2012 R2
Cognosys Inc. • Virtual machines
Hardened SQL Server 2014 Express

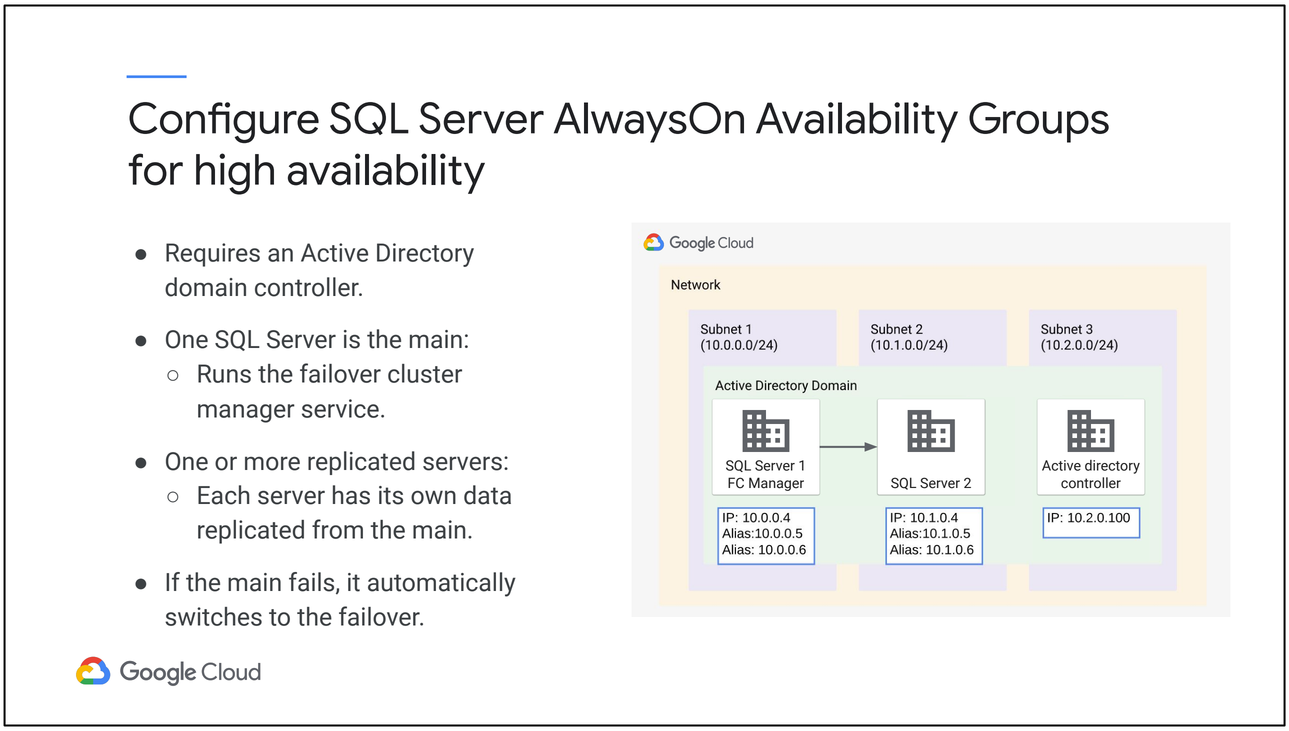
SQL Server 2016 Express on Hardened Windows Server 2016
Cognosys Inc. • Virtual machines
Hardened SQL Server 2016 Express

SQL Server 2014 Express on Hardened Windows Server 2016
Cognosys Inc. • Virtual machines
Hardened SQL Server 2014 Express

Google Cloud

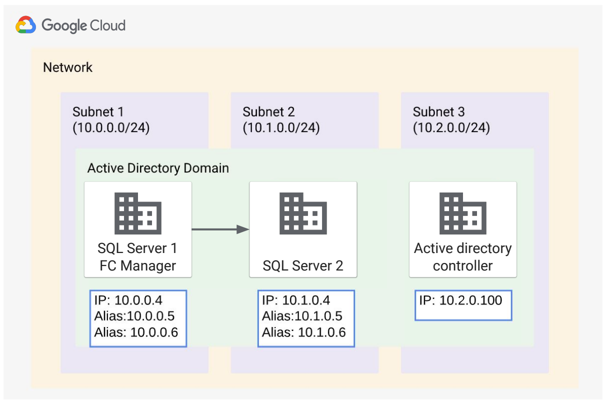
As an added convenience, instead of just using a pre-made Google image to build your own machine, you could search the Marketplace and find fully pre-configured machines based on Windows or Linux.

Base your decisions on how much time you want to invest in building your image and how much custom configuration you need.



Configure SQL Server AlwaysOn Availability Groups for high availability

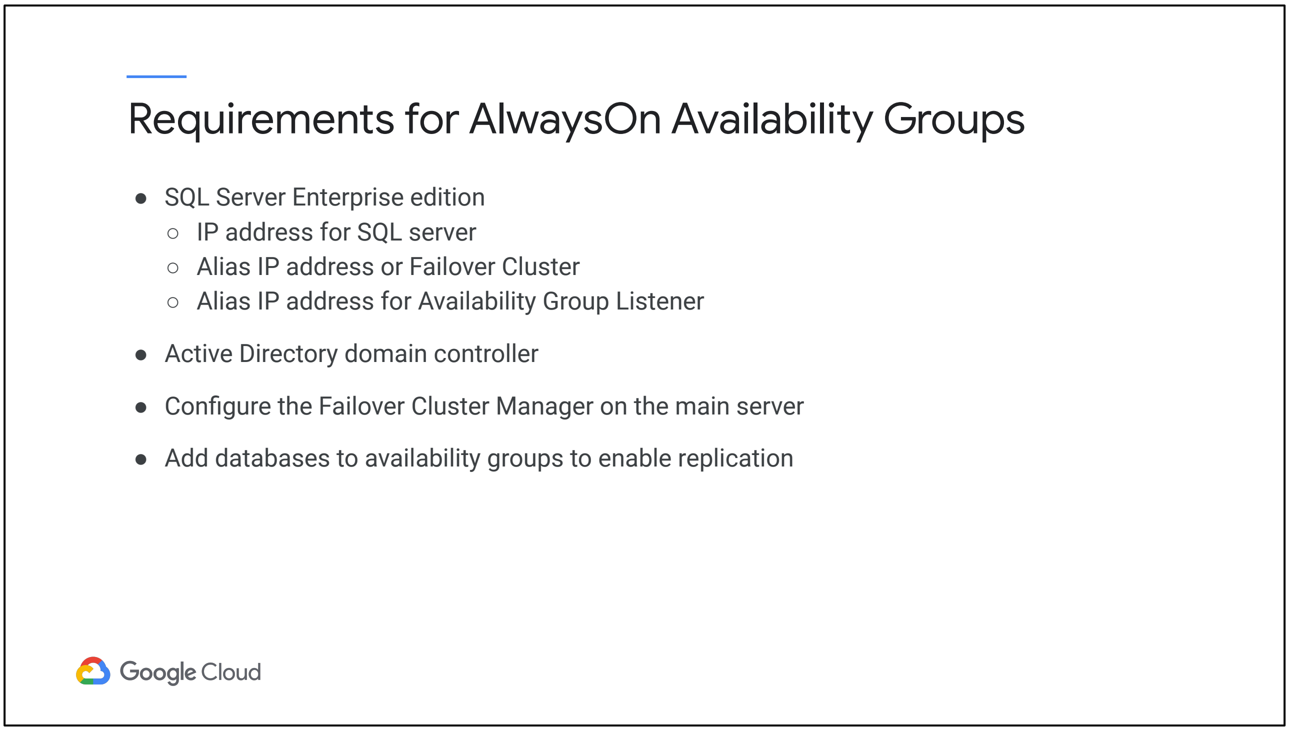
- Requires an Active Directory domain controller.
- One SQL Server is the main:
  - Runs the failover cluster manager service.
- One or more replicated servers:
  - Each server has its own data replicated from the main.
- If the main fails, it automatically switches to the failover.



Google Cloud

If you're worried about the main SQL Server instance going down and need a solution that guarantees the server is always available, you can use the SQL Server AlwaysOn Availability Groups.

- This requires an Active Directory domain controller.
- One server would be designated as the main server.
- Then one or more replicated servers would be in a standby mode, constantly keeping their own copy of the data in sync with the main.
- If the main fails, one of the standbys would take over as the main server and continue the workload uninterrupted. Eventually, you can fix the main, resync the data, and switch back to it.



Requirements for AlwaysOn Availability Groups

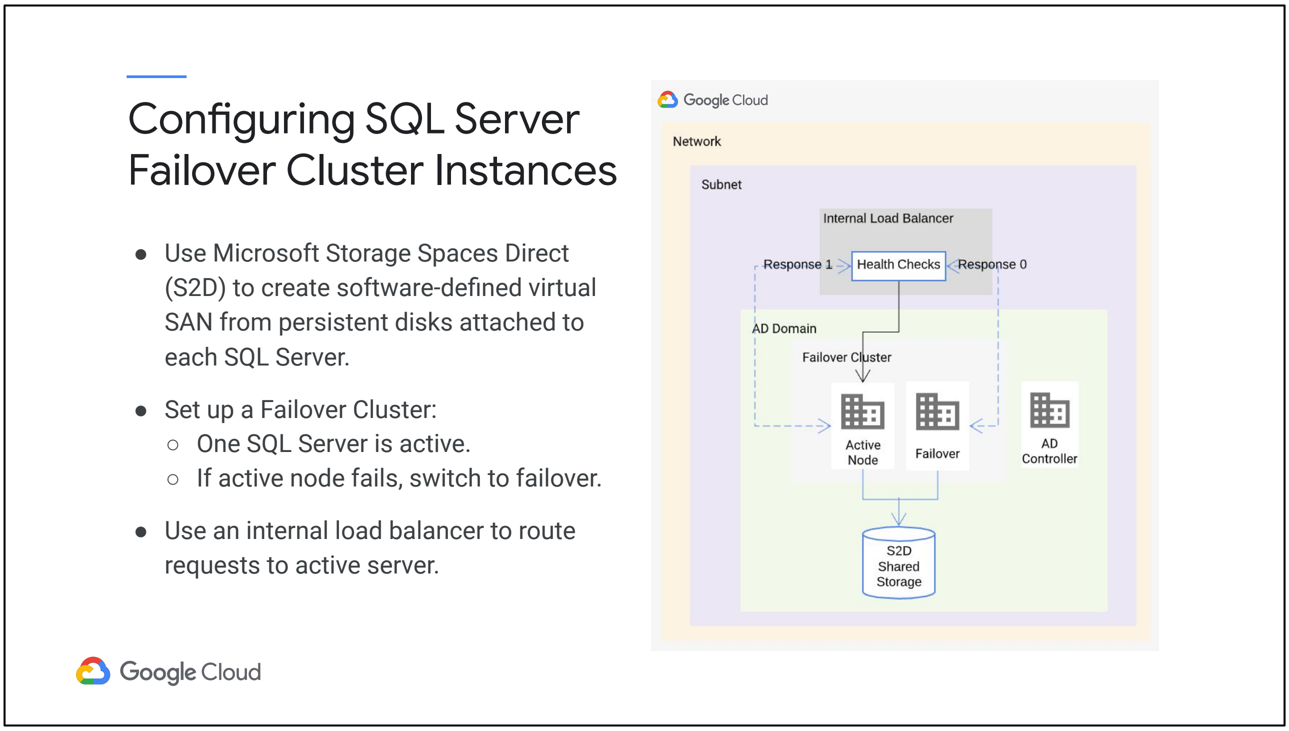
- SQL Server Enterprise edition
  - IP address for SQL server
  - Alias IP address or Failover Cluster
  - Alias IP address for Availability Group Listener
- Active Directory domain controller
- Configure the Failover Cluster Manager on the main server
- Add databases to availability groups to enable replication

AlwaysOn Availability requires the more expensive Enterprise edition of SQL Server, so if you need this feature, make sure you consider price when choosing the image to build your VM.

Configuring all of this is exactly as it would be for an on-premises version of SQL Server.

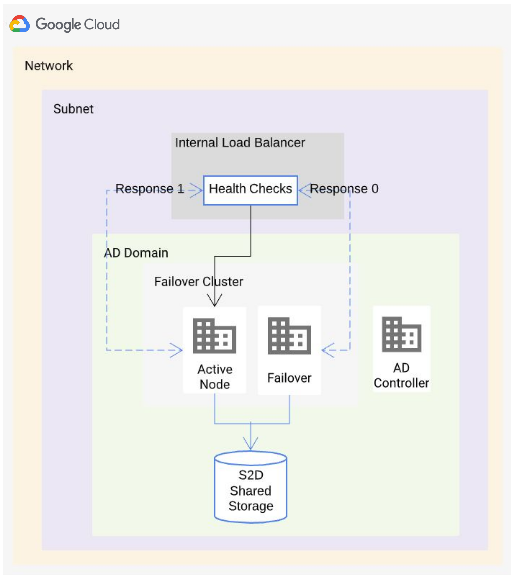
You set up the main and the Active Directory, configure the Failover Cluster Manager on the main server, and add databases to the availability groups to enable replication.

The Google documentation contains a good tutorial on setting this up so please refer to that documentation for more details.



Configuring SQL Server Failover Cluster Instances

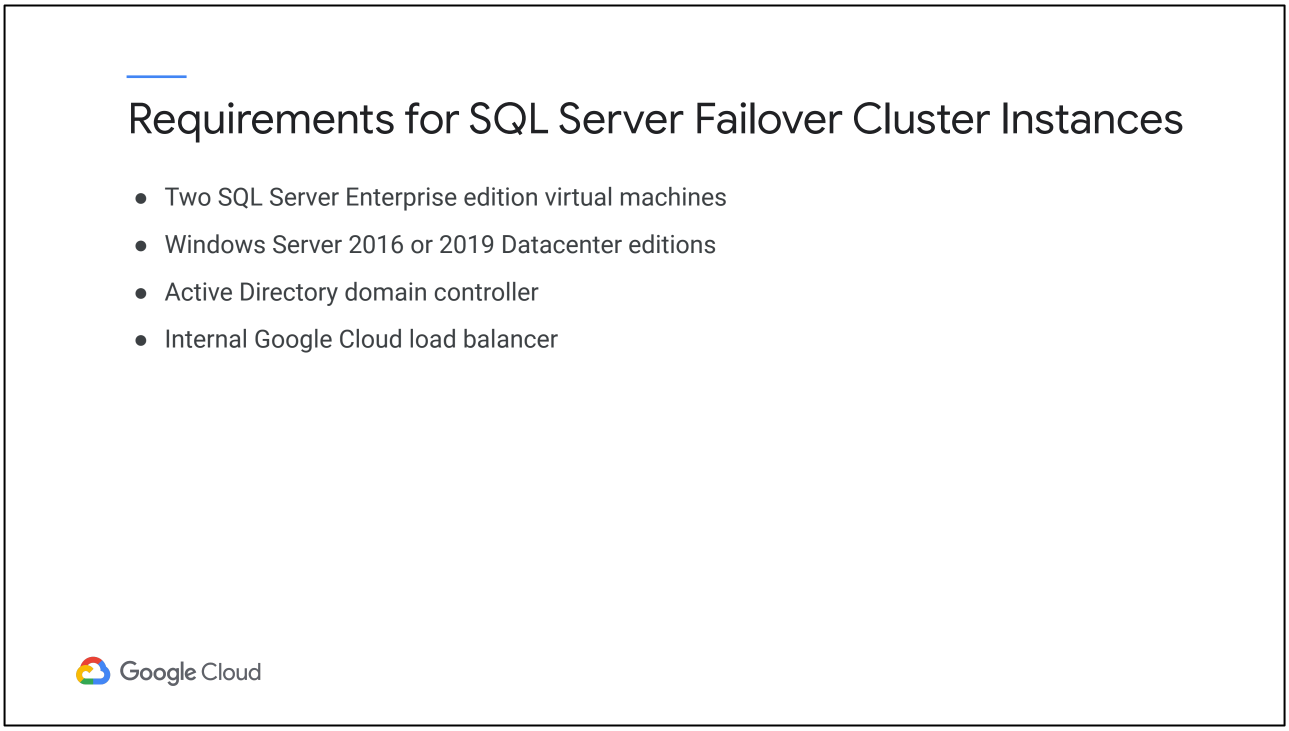
- Use Microsoft Storage Spaces Direct (S2D) to create software-defined virtual SAN from persistent disks attached to each SQL Server.
- Set up a Failover Cluster:
  - One SQL Server is active.
  - If active node fails, switch to failover.
- Use an internal load balancer to route requests to active server.



Google Cloud

Another slightly different option for high availability is SQL Server Failover Cluster Instances.

- Using this option, you set up a single shared storage device. This would be like a virtual SAN setup using Microsoft Storage Spaces Direct.
- Then you have a main instance of SQL Server that uses that virtual SAN, but if it fails, a standby node would become active and attach to the shared SAN to continue the work uninterrupted.
- A load balancer is used to send requests to the active server.



Requirements for SQL Server Failover Cluster Instances

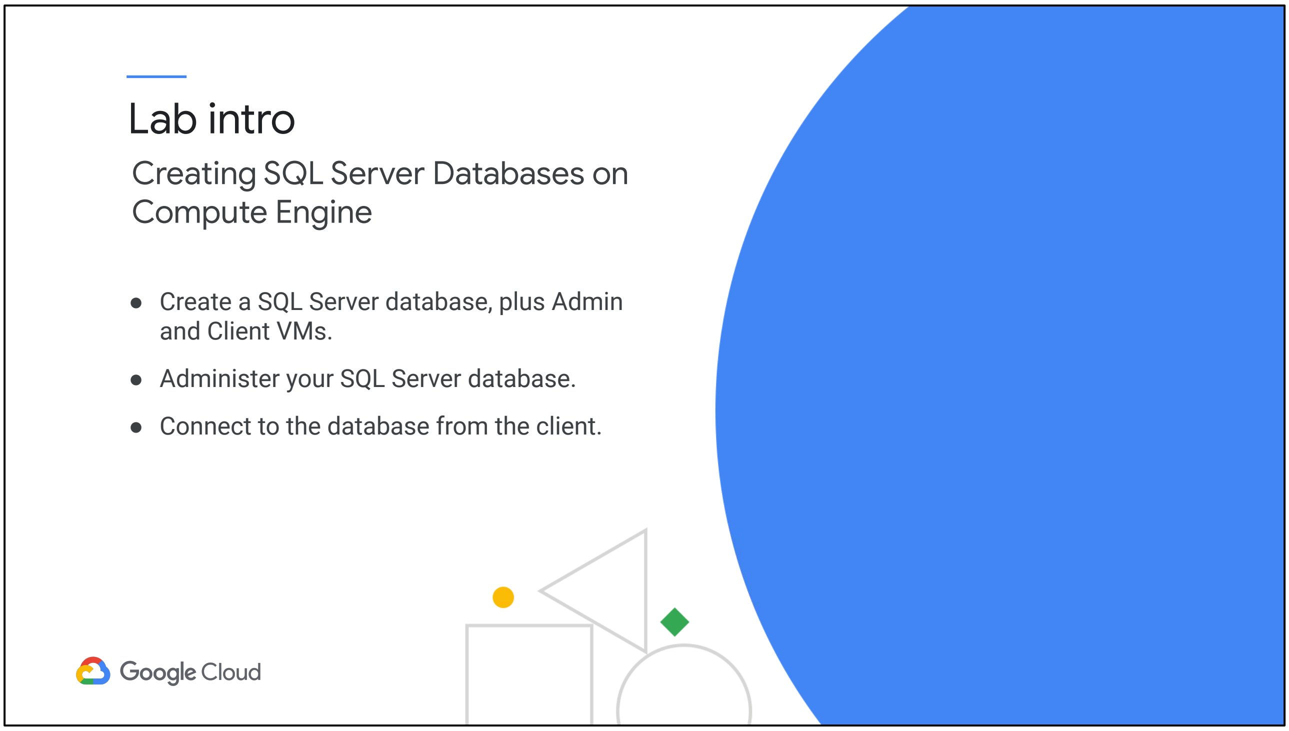
- Two SQL Server Enterprise edition virtual machines
- Windows Server 2016 or 2019 Datacenter editions
- Active Directory domain controller
- Internal Google Cloud load balancer

Google Cloud

This option also requires the Enterprise Edition of SQL Server, as well as the Datacenter versions of Windows Server. Also, you need an Active Directory domain controller and an internal Google Cloud load balancer.

Choosing between Availability Groups and Failover Cluster Instances is a tricky choice full of pros and cons for each. Microsoft offers both of these choices to provide flexibility in meeting your needs.

As with Availability Groups, there is a good tutorial in the Google Cloud documentation for setting up failover cluster instances.



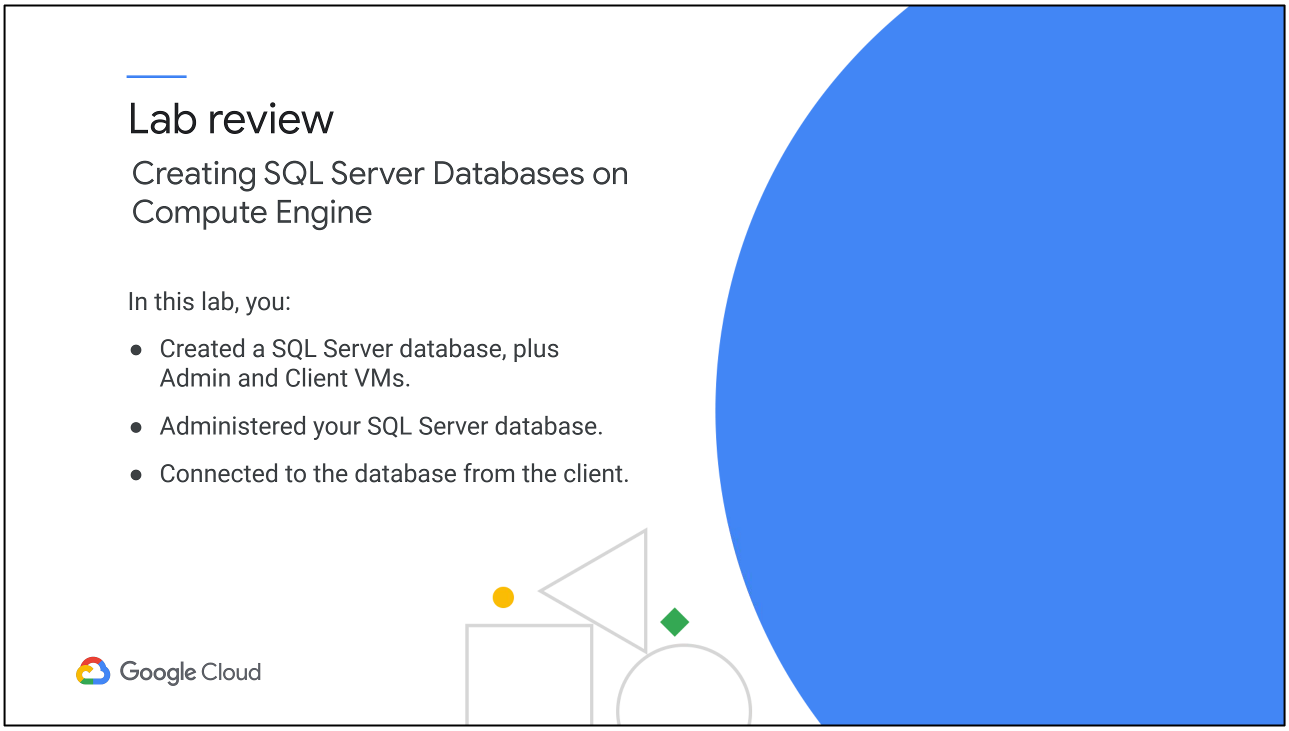
Lab intro

Creating SQL Server Databases on Compute Engine

- Create a SQL Server database, plus Admin and Client VMs.
- Administer your SQL Server database.
- Connect to the database from the client.

Google Cloud

In this lab, you create a SQL Server VM and admin and client VMs. You then do some basic administration and make a connection from a client to the database.



Lab review

Creating SQL Server Databases on Compute Engine

In this lab, you:

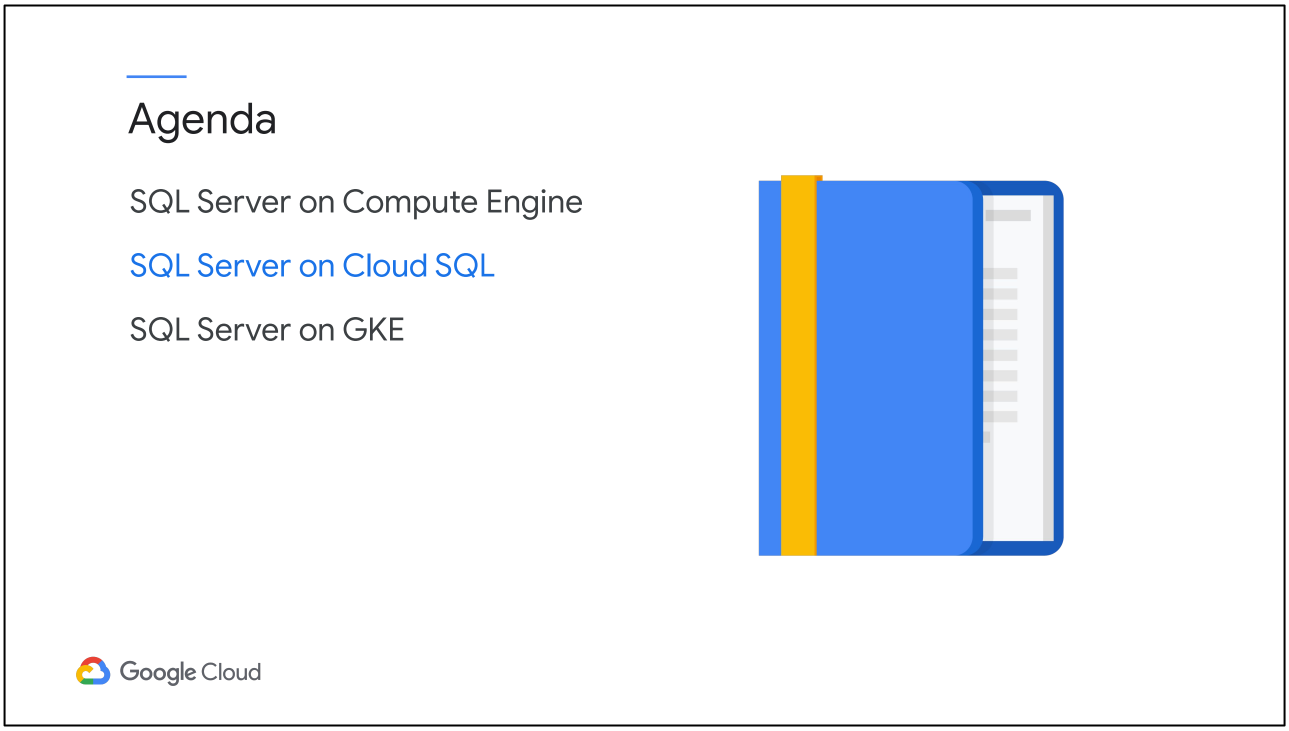
- Created a SQL Server database, plus Admin and Client VMs.
- Administered your SQL Server database.
- Connected to the database from the client.

Google Cloud

In this lab, you:
Created a SQL Server database, plus Admin and Client VMs.
Administered your SQL Server database.
And connected to the database from the client.

Compute Engine allows you to deploy SQL Server databases with no limitations. Using Compute Engine, you can configure your SQL Server databases just as you would on-premises.

It is ideal for when you want to migrate using a lift-and-shift approach.

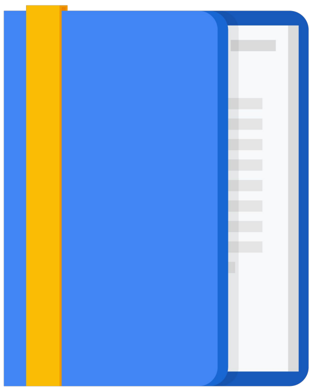


Agenda

SQL Server on Compute Engine

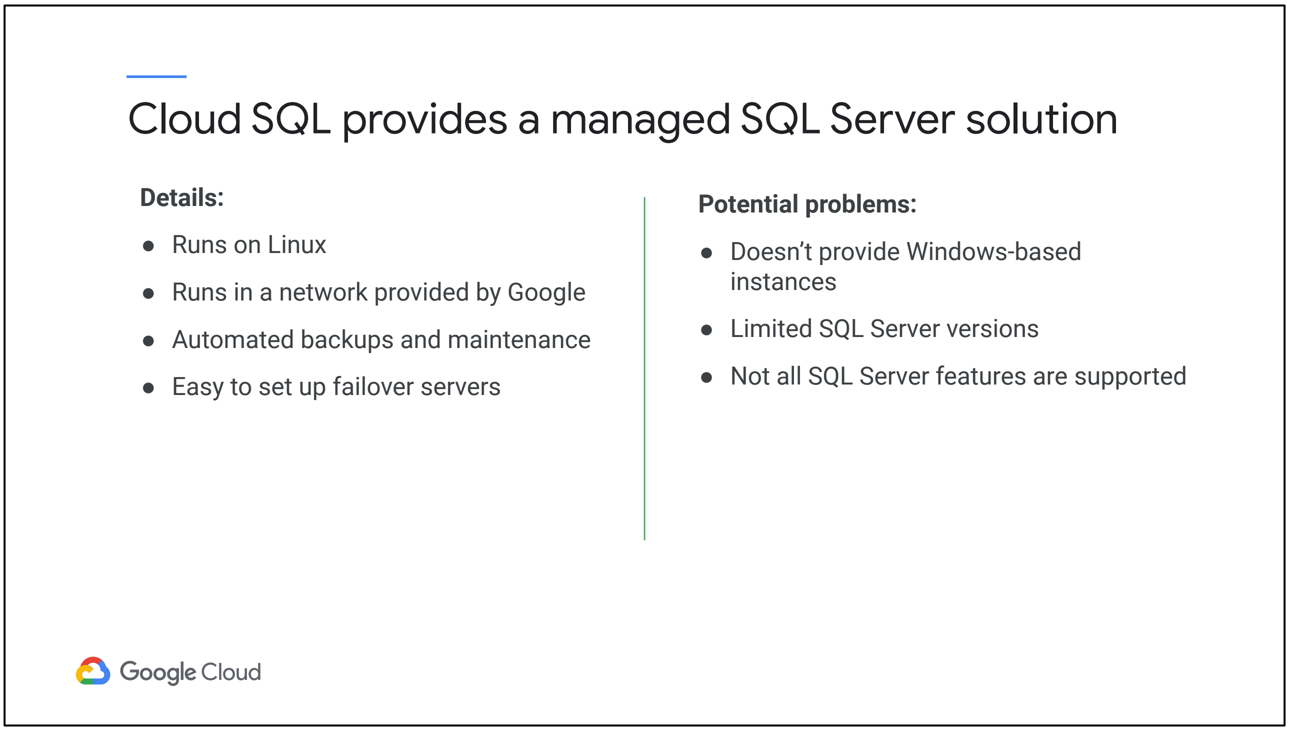
SQL Server on Cloud SQL

SQL Server on GKE



Google Cloud

SQL Server on Cloud SQL offers an easier alternative to fully building out a VM. It comes with all the benefits of using a fully managed service compared to your own VM, but as always, you have to make some trade-offs for that easier option.



Cloud SQL provides a managed SQL Server solution

Details:

- Runs on Linux
- Runs in a network provided by Google
- Automated backups and maintenance
- Easy to set up failover servers

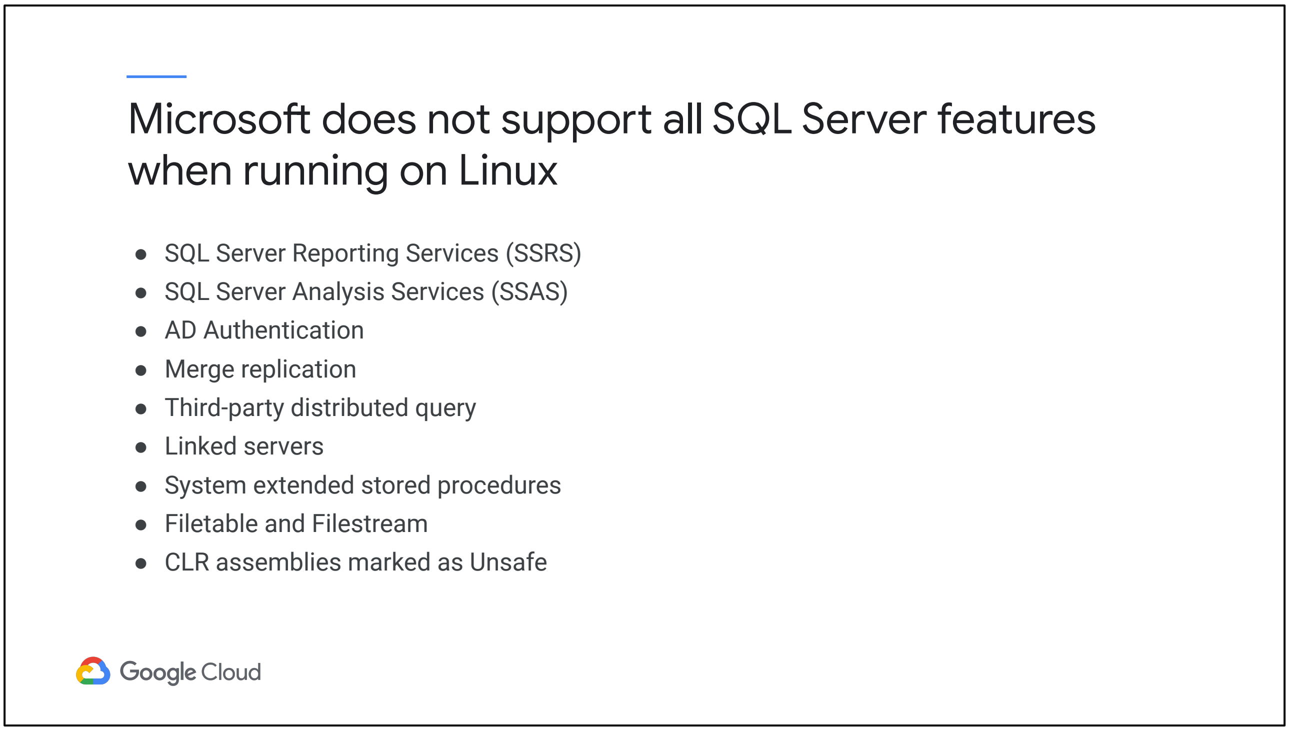
Potential problems:

- Doesn't provide Windows-based instances
- Limited SQL Server versions
- Not all SQL Server features are supported

Google Cloud

The first notable difference is that the server runs on Linux, not Windows. Google handles all the networking and backup and maintenance for you, and also makes it easy to set up failover servers.

The downside of not having a Windows-based instance is that not all versions of SQL Server are supported. Also, some of the features that rely on Windows are not supported.



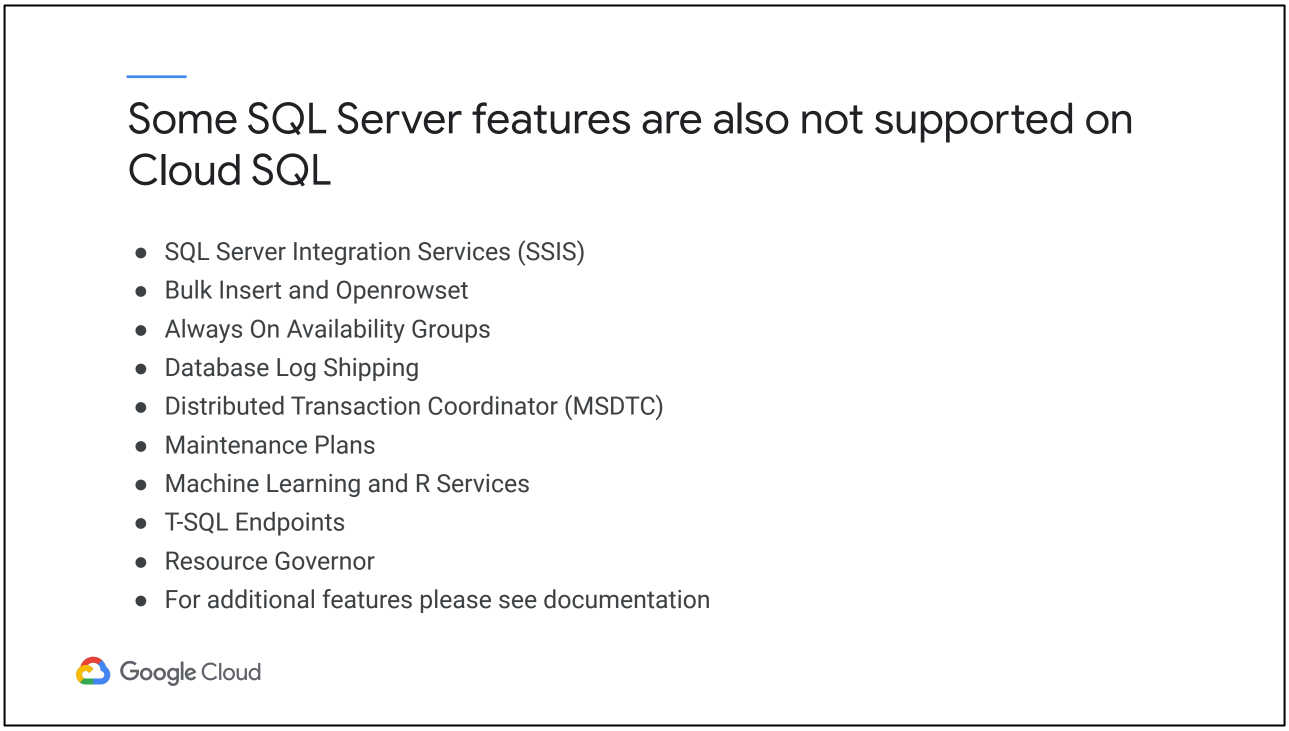
Microsoft does not support all SQL Server features when running on Linux

- SQL Server Reporting Services (SSRS)
- SQL Server Analysis Services (SSAS)
- AD Authentication
- Merge replication
- Third-party distributed query
- Linked servers
- System extended stored procedures
- Filetable and Filestream
- CLR assemblies marked as Unsafe

Google Cloud

Until Microsoft's recent association with Linux, SQL Server ran exclusively on Windows, so many features heavily relied on Windows features. Uncoupling SQL Server from Windows means that some of those features won't work on a Linux instance. If you need any of these features, you either have to use a Compute Engine VM or re-engineer your application to use an alternative that does work on a Linux instance.

This slide lists the features of SQL Server that Microsoft currently doesn't support on Linux.



Some SQL Server features are also not supported on Cloud SQL

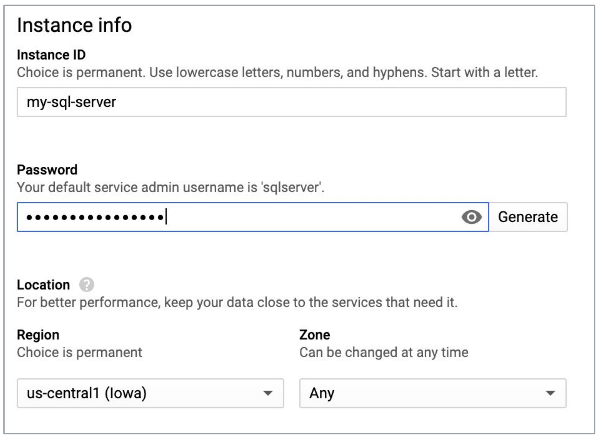
- SQL Server Integration Services (SSIS)
- Bulk Insert and Openrowset
- Always On Availability Groups
- Database Log Shipping
- Distributed Transaction Coordinator (MSDTC)
- Maintenance Plans
- Machine Learning and R Services
- T-SQL Endpoints
- Resource Governor
- For additional features please see documentation

Google Cloud

The previous slide notes the features that do not run on a Linux instance. So even if you built your own VM on Linux, those are not supported. In addition, running a Cloud SQL instance involves an additional list of features that are not supported. Some of these just don’t make sense in a Cloud SQL environment, and others are simply not supported for technical reasons.

This slide lists features of SQL Server that are not supported by Cloud SQL. These lists may seem long. However, many features are not widely used, and you can often use alternatives or workarounds.

Create a Cloud SQL for SQL Server instance in minutes using the Console


Instance info

Instance ID
Choice is permanent. Use lowercase letters, numbers, and hyphens. Start with a letter.
my-sql-server

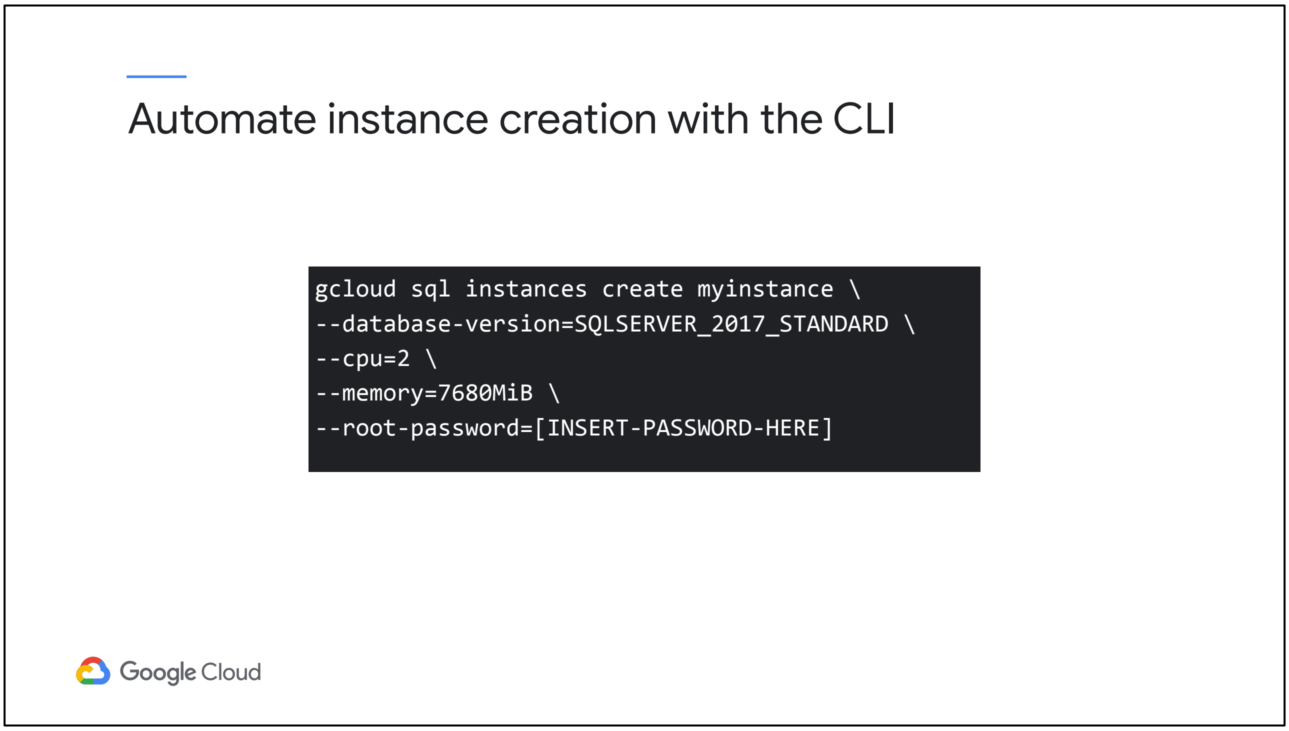
Password
Your default service admin username is 'sqlserver'.
••••••••••••••••

Location
For better performance, keep your data close to the services that need it.

Region
Choice is permanent
us-central1 (Iowa)

Zone
Can be changed at any time
Any

It is very quick and easy to create a Cloud SQL instance of SQL Server. There's a lot less to configure than when building your own VM. Basically, you just need an instance ID and a password and to choose the region for it. There are additional options, but these are the ones you always need to set.

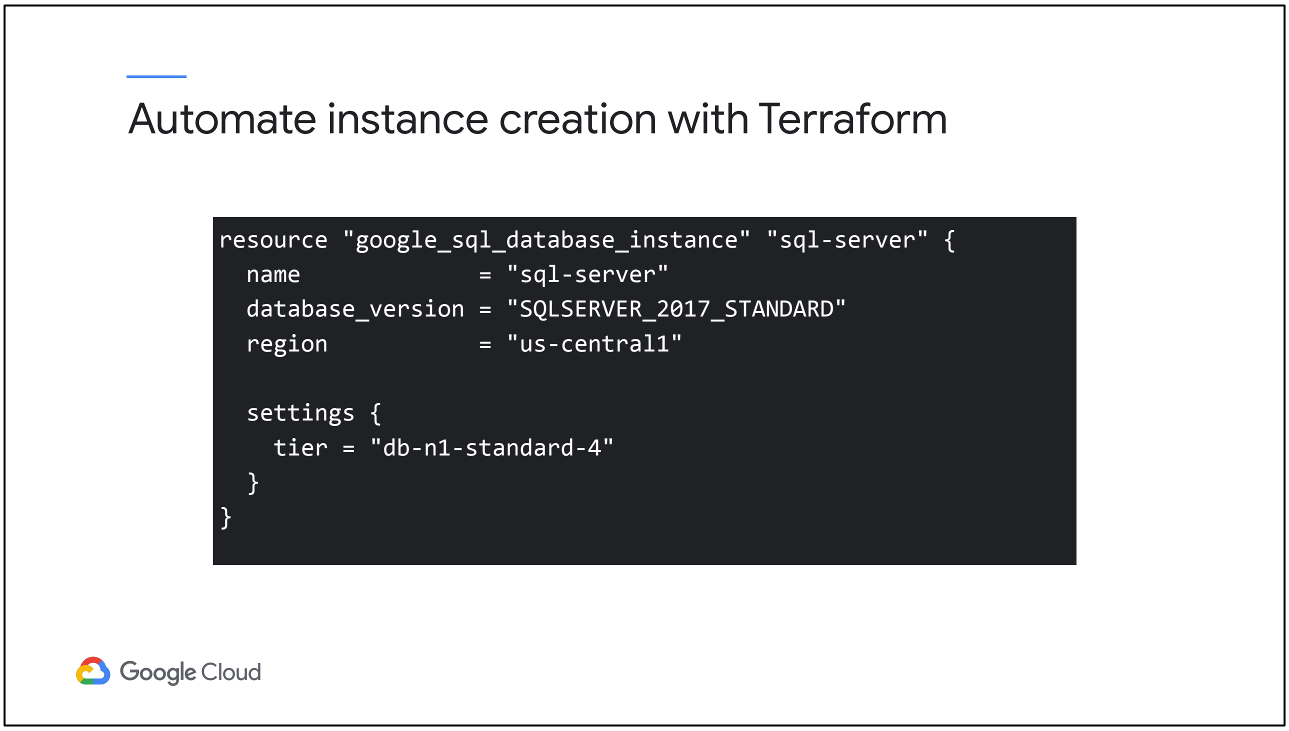


Automate instance creation with the CLI

```
gcloud sql instances create myinstance \
--database-version=SQLSERVER_2017_STANDARD \
--cpu=2 \
--memory=7680MiB \
--root-password=[INSERT-PASSWORD-HERE]
```

Google Cloud

It is also very easy to create a Cloud SQL machine using the CLI. Compare the gcloud command for creating this instance to creating a VM, and you can see how much simpler it is.



Automate instance creation with Terraform

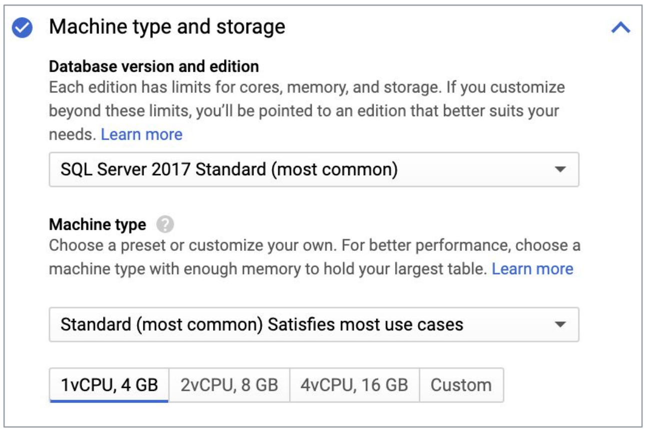
```
resource "google_sql_database_instance" "sql-server" {
  name             = "sql-server"
  database_version = "SQLSERVER_2017_STANDARD"
  region           = "us-central1"

  settings {
    tier = "db-n1-standard-4"
  }
}
```

Google Cloud

Or if you are using Terraform, Cloud SQL databases can be created with a simple template.

As you saw earlier in the course, Terraform makes it easier to configure your data center resources by defining what you want to configure with a simple configuration language.

Choose SQL Server Edition and machine type when configuring the database



Machine type and storage

Database version and edition
Each edition has limits for cores, memory, and storage. If you customize beyond these limits, you'll be pointed to an edition that better suits your needs. Learn more

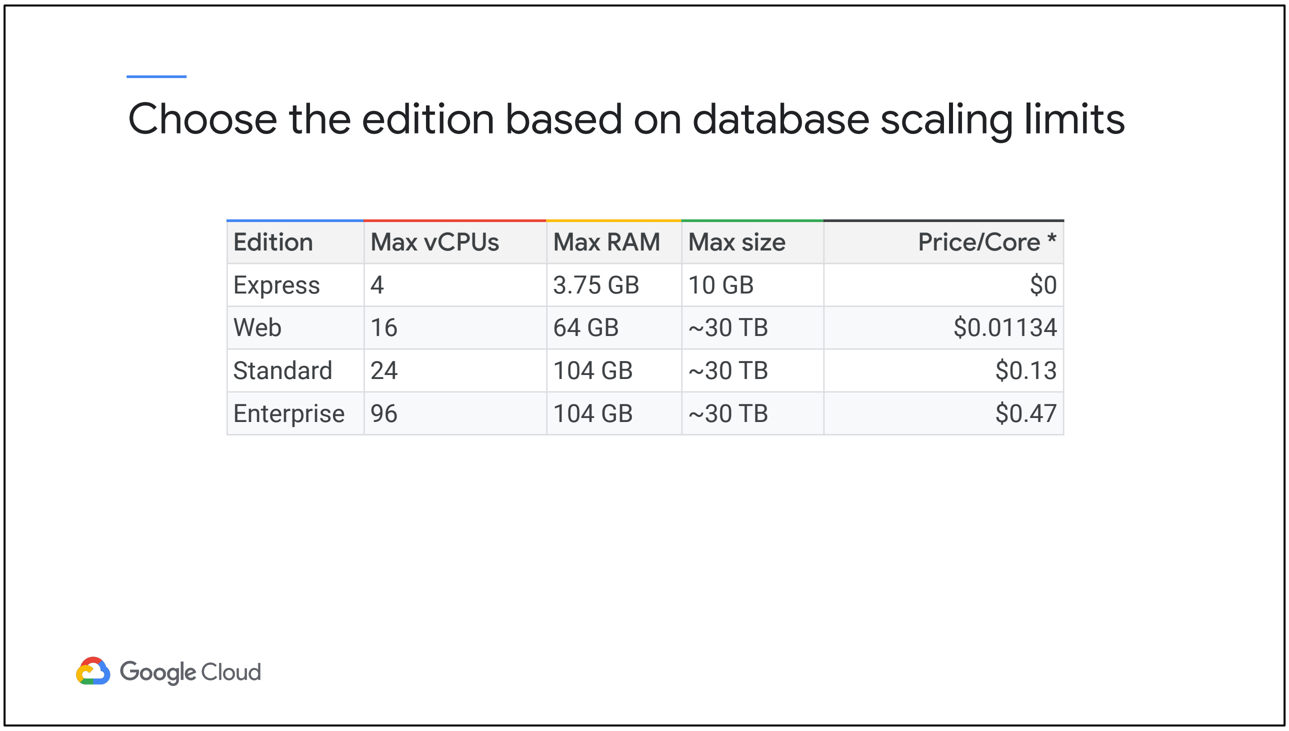
SQL Server 2017 Standard (most common)

Machine type
Choose a preset or customize your own. For better performance, choose a machine type with enough memory to hold your largest table. Learn more

Standard (most common) Satisfies most use cases

Google Cloud

You can also choose among several SQL Server editions and machine configurations.



Choose the edition based on database scaling limits

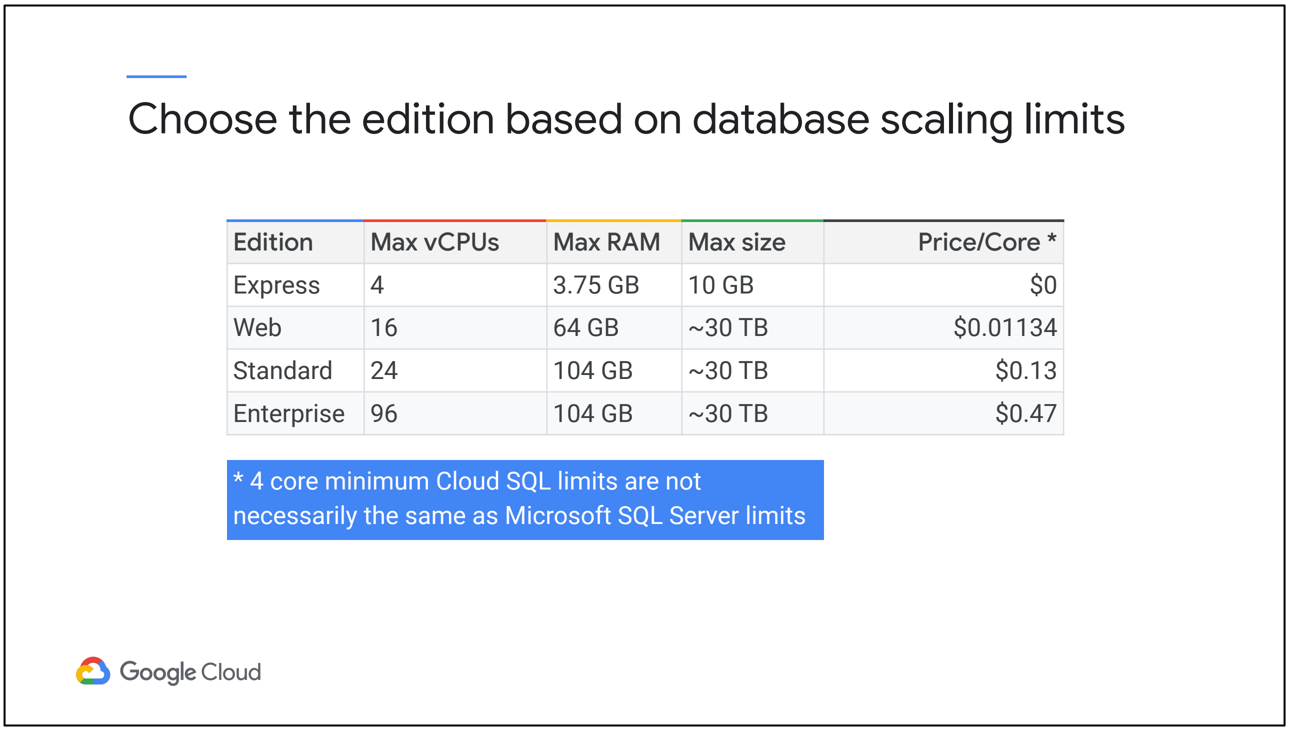
Edition	Max vCPUs	Max RAM	Max size	Price/Core *
Express	4	3.75 GB	10 GB	$0
Web	16	64 GB	~30 TB	$0.01134
Standard	24	104 GB	~30 TB	$0.13
Enterprise	96	104 GB	~30 TB	$0.47

Google Cloud

The choice of version is largely influenced by your database size needs. Pricing is significantly different, so it’s important to plan ahead; otherwise, you might have to migrate the databases to another server later, or you might spend much more than you need.

There are four editions of SQL Server: Express, Web, Standard, and Enterprise. Express edition has no license charge but is limited to a 10-GB database.

The other three editions require licensing; Enterprise being the most expensive. See the table on this slide for vCPU, RAM, and Database size features and license costs.



Choose the edition based on database scaling limits

Edition	Max vCPUs	Max RAM	Max size	Price/Core *
Express	4	3.75 GB	10 GB	$0
Web	16	64 GB	~30 TB	$0.01134
Standard	24	104 GB	~30 TB	$0.13
Enterprise	96	104 GB	~30 TB	$0.47

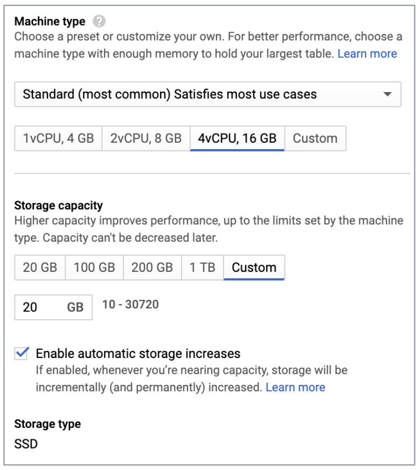
* 4 core minimum Cloud SQL limits are not necessarily the same as Microsoft SQL Server limits

Google Cloud

An important point to note is licensing is per-vCPU Core, but you are always charged for at least 4 vCPUs.

Configure the machine type according to required capacity

- Specify the number of vCPUs and RAM for your instance.
- Specify disk space from 10 GB to 30 TB:
  - Enable automatic storage increases to increase disk space as needed.



Machine type
Choose a preset or customize your own. For better performance, choose a machine type with enough memory to hold your largest table. Learn more

Standard (most common) Satisfies most use cases

1vCPU, 4 GB | 2vCPU, 8 GB | [✓ 4vCPU, 16 GB] | Custom

Storage capacity
Higher capacity improves performance, up to the limits set by the machine type. Capacity can't be decreased later.

20 GB | 100 GB | 200 GB | 1 TB | [✓ Custom]

20 GB 10 - 30720

✓ Enable automatic storage increases
If enabled, whenever you're nearing capacity, storage will be incrementally (and permanently) increased. Learn more

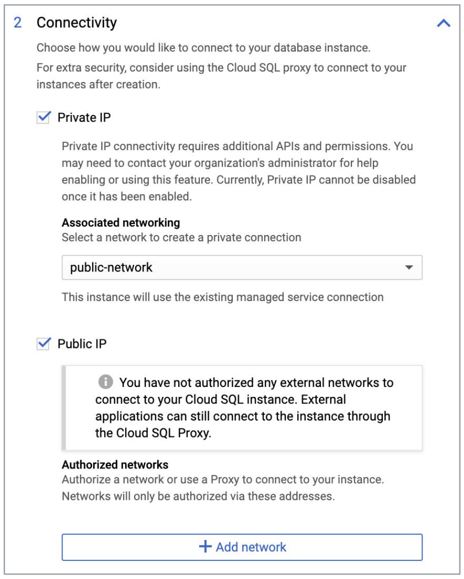
Storage type
SSD

Google Cloud

Configure the machine type according to required capacity.

Enabling Automatic storage increases is a good option to consider because it allows you to start off with smaller disks and grow them as needed. Remember, you pay for what you allocate, not what you use.

You can give your server a private IP, a public IP, or both


2 Connectivity

Choose how you would like to connect to your database instance.

For extra security, consider using the Cloud SQL proxy to connect to your instances after creation.

☑ Private IP

Private IP connectivity requires additional APIs and permissions. You may need to contact your organization's administrator for help enabling or using this feature. Currently, Private IP cannot be disabled once it has been enabled.

Associated networking
Select a network to create a private connection

public-network

This instance will use the existing managed service connection

☑ Public IP

You have not authorized any external networks to connect to your Cloud SQL instance. External applications can still connect to the instance through the Cloud SQL Proxy.

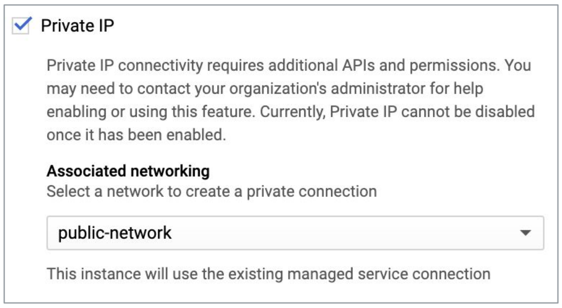
Authorized networks
Authorize a network or use a Proxy to connect to your instance. Networks will only be authorized via these addresses.

Google Cloud

Cloud SQL instances can have either a private or public IP address, or both.

Choose only a private IP for a more secure architecture

- A VPC peering is created between the Cloud SQL database network and your network.
- Only machines in your peered network can access the database.



☑ Private IP

Private IP connectivity requires additional APIs and permissions. You may need to contact your organization's administrator for help enabling or using this feature. Currently, Private IP cannot be disabled once it has been enabled.

Associated networking
Select a network to create a private connection

public-network

This instance will use the existing managed service connection

Google Cloud

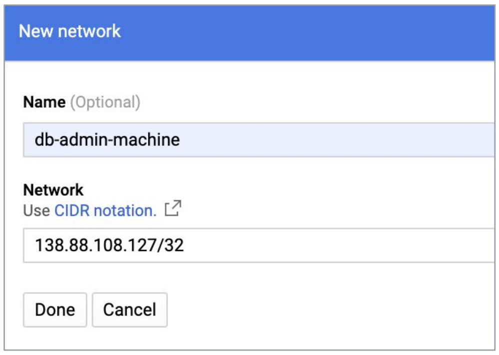
A private IP would make your instance more secure, but you'd have to configure authorized clients to connect to that instance through the Private IP using a peered network.

Enabling Private IP creates a VPC peering between the Cloud SQL database network and your network.

Then, Only machines in your peered network can access the database.

A firewall protects machines with public IP addresses

- Authorize one or more external sources to allow access to the database.
- Use CIDR notation to allow IP addresses or ranges.
- By default, no machine outside your project has access to the database.



Name (Optional)

db-admin-machine

Network

Use CIDR notation.

138.88.108.127/32

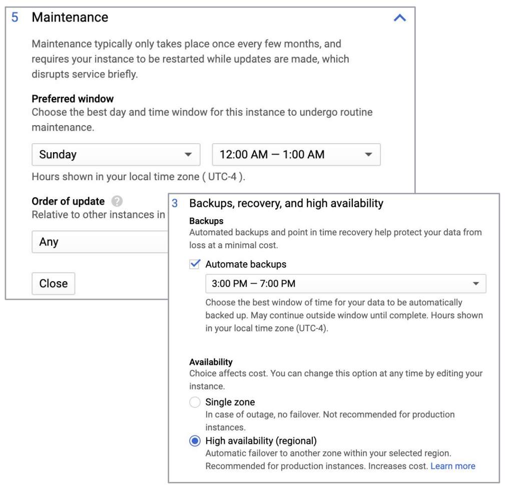
Google Cloud

A firewall is used for protection in the case of public IPs and, by default, only allows machines within the project to have access to the server.

However, you could authorize one or more external connections. As with configuring any firewall rule, use CIDR notation to allow IP addresses or ranges.

Cloud SQL automates administration

- Choose a backup window when there is low utilization.
- Enabling high availability creates a second failover server in another zone.
- Specify a maintenance window.
  - Maintenance requires a reboot.



5 Maintenance

Maintenance typically only takes place once every few months, and requires your instance to be restarted while updates are made, which disrupts service briefly.

Preferred window
Choose the best day and time window for this instance to undergo routine maintenance.

Sunday | 12:00 AM — 1:00 AM

Hours shown in your local time zone (UTC-4).

Order of update
Relative to other instances in

Any

Close

3 Backups, recovery, and high availability

Backups
Automated backups and point in time recovery help protect your data from loss at a minimal cost.

✓ Automate backups

3:00 PM — 7:00 PM

Choose the best window of time for your data to be automatically backed up. May continue outside window until complete. Hours shown in your local time zone (UTC-4).

Availability
Choice affects cost. You can change this option at any time by editing your instance.

○ Single zone
In case of outage, no failover. Not recommended for production instances.

● High availability (regional)
Automatic failover to another zone within your selected region. Recommended for production instances. Increases cost. Learn more

Google Cloud

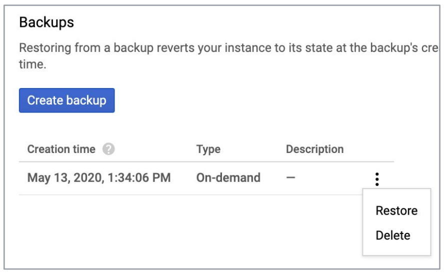
Cloud SQL provides its own maintenance options and automation.

Backups should be made during a period of low utilization.

You can also easily create a failover cluster by just enabling the High Availability option.

The service will also automate typical maintenance tasks like patch management. As with backups Choose a maintenance window when there is low utilization. If maintenance requires a reboot, the server will be unavailable for a short time.

Backups and restores can be done from the Console or by using the CLI


Backups

Restoring from a backup reverts your instance to its state at the backup's cre time.

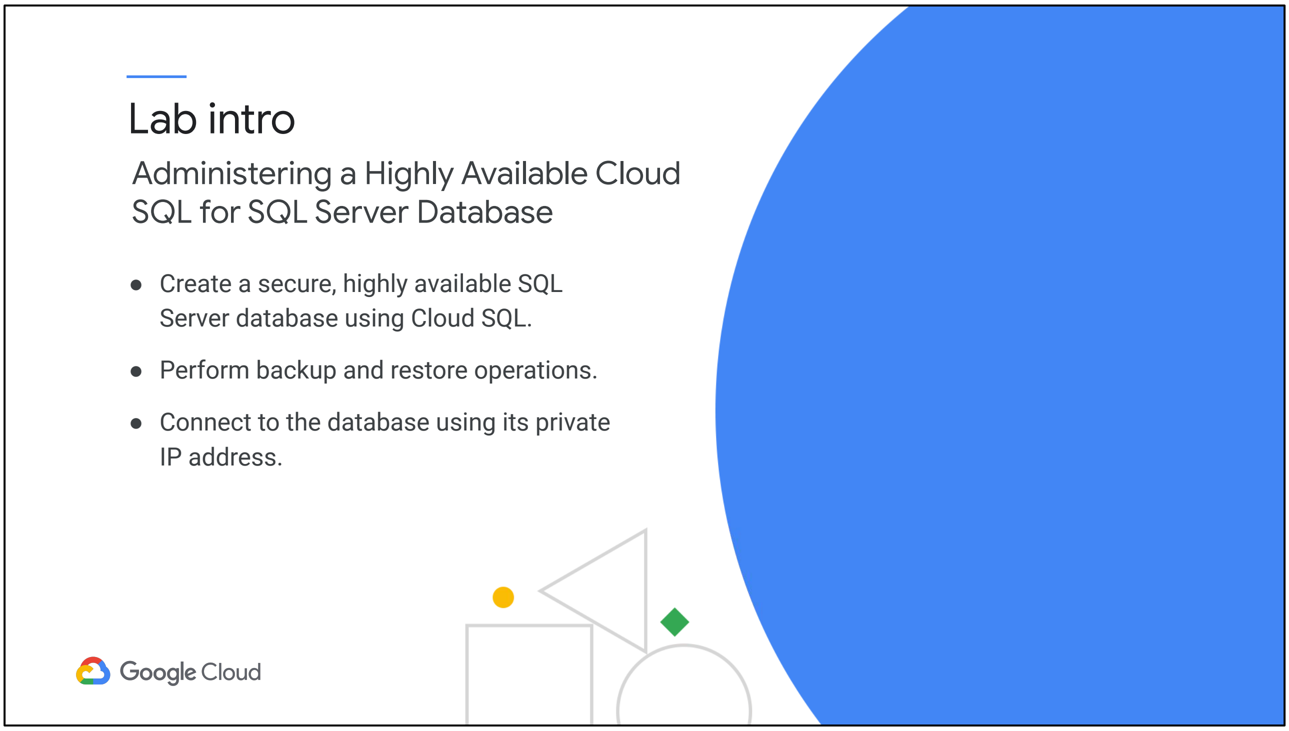
Creation time	Type	Description
May 13, 2020, 1:34:06 PM	On-demand	—

```
gcloud sql backups create --async
--instance sql-server
--root-password=**********

gcloud sql backups list --instance
sql-server

gcloud sql backups restore
[BACKUP_ID]
--restore-instance=sql-server
```

Google Cloud

You can also manually perform a backup or restore with either the console or the CLI. Remember, these are not the same type of backups as native SQL Server database and log backups. A Cloud SQL backup is a snapshot of the persistent disk with the data. It can only be restored to another Cloud SQL database.

In addition, you can always run standard SQL Server backups as you would with any SQL Server database.



Lab intro

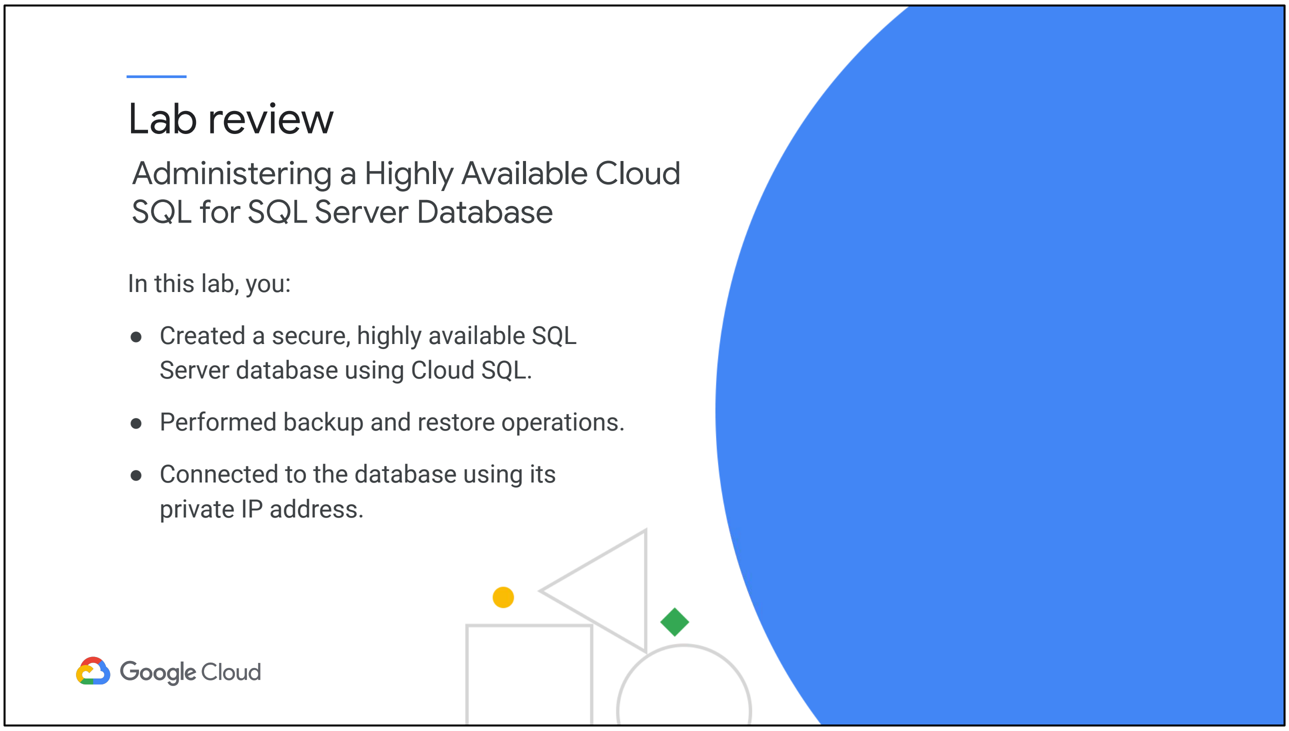
Administering a Highly Available Cloud SQL for SQL Server Database

- Create a secure, highly available SQL Server database using Cloud SQL.
- Perform backup and restore operations.
- Connect to the database using its private IP address.

Google Cloud

While Cloud SQL has some limitations, it makes setting up and administering a database much easier. You can create failover servers by simply choosing an option. Backup, restore, and admin tasks are automated.

In this lab, you create a SQL Server Cloud SQL instance that is secure and highly available. You will perform a backup and a restore operation and connect to the instance from a client using its private IP address.



Lab review

Administering a Highly Available Cloud SQL for SQL Server Database

In this lab, you:

- Created a secure, highly available SQL Server database using Cloud SQL.
- Performed backup and restore operations.
- Connected to the database using its private IP address.

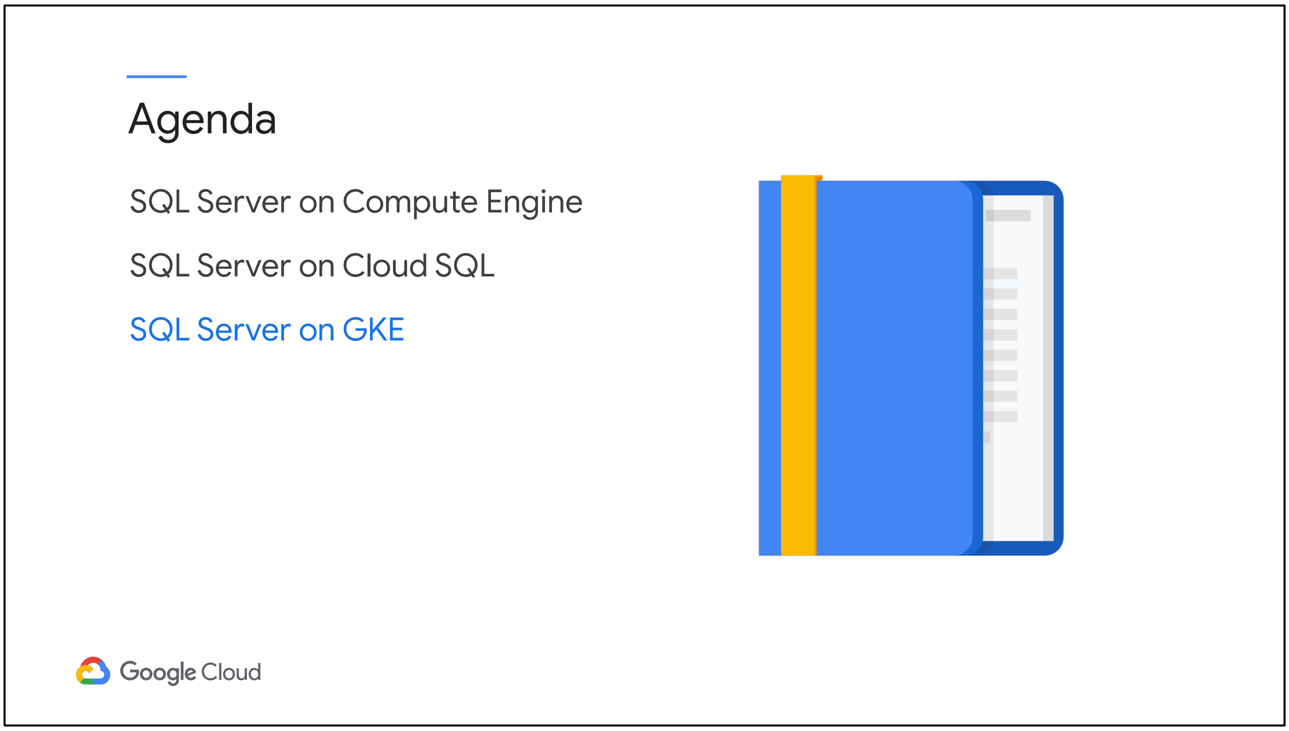
Google Cloud

In this lab, you created a secure, highly available SQL Server database using Cloud SQL.

You also performed backup and restore operations.

And You connected to the database using its private IP address.

Cloud SQL allows you to deploy SQL Server in a managed environment. Administration like backups and patches are automated for you. You can create a failover machine just by setting a parameter and Google manages a secure network for you.

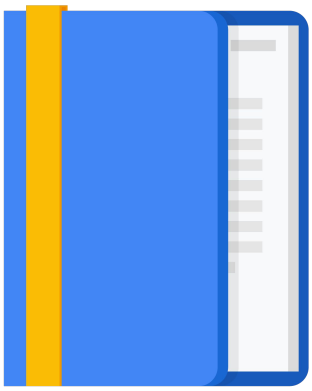


Agenda

SQL Server on Compute Engine

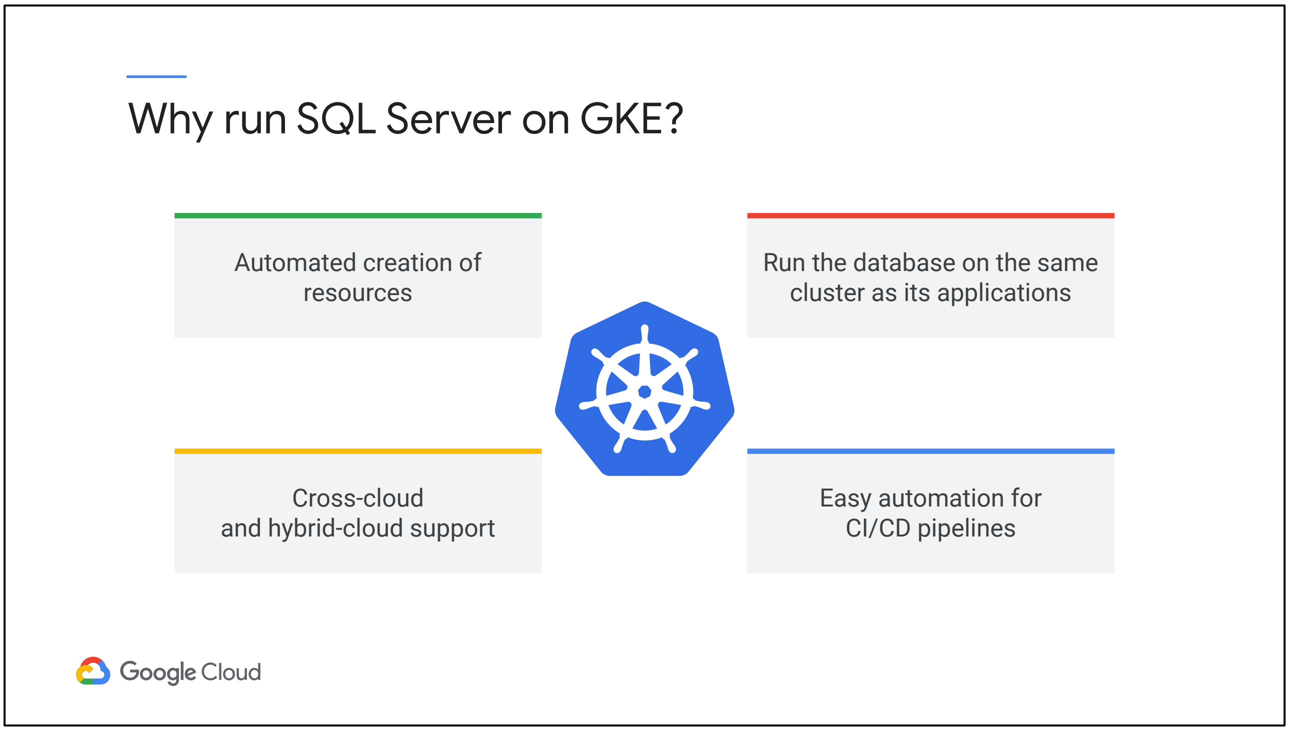
SQL Server on Cloud SQL

SQL Server on GKE



Google Cloud

A third option for running SQL Server is to deploy it on GKE.



Why run SQL Server on GKE?

Automated creation of resources

Run the database on the same cluster as its applications

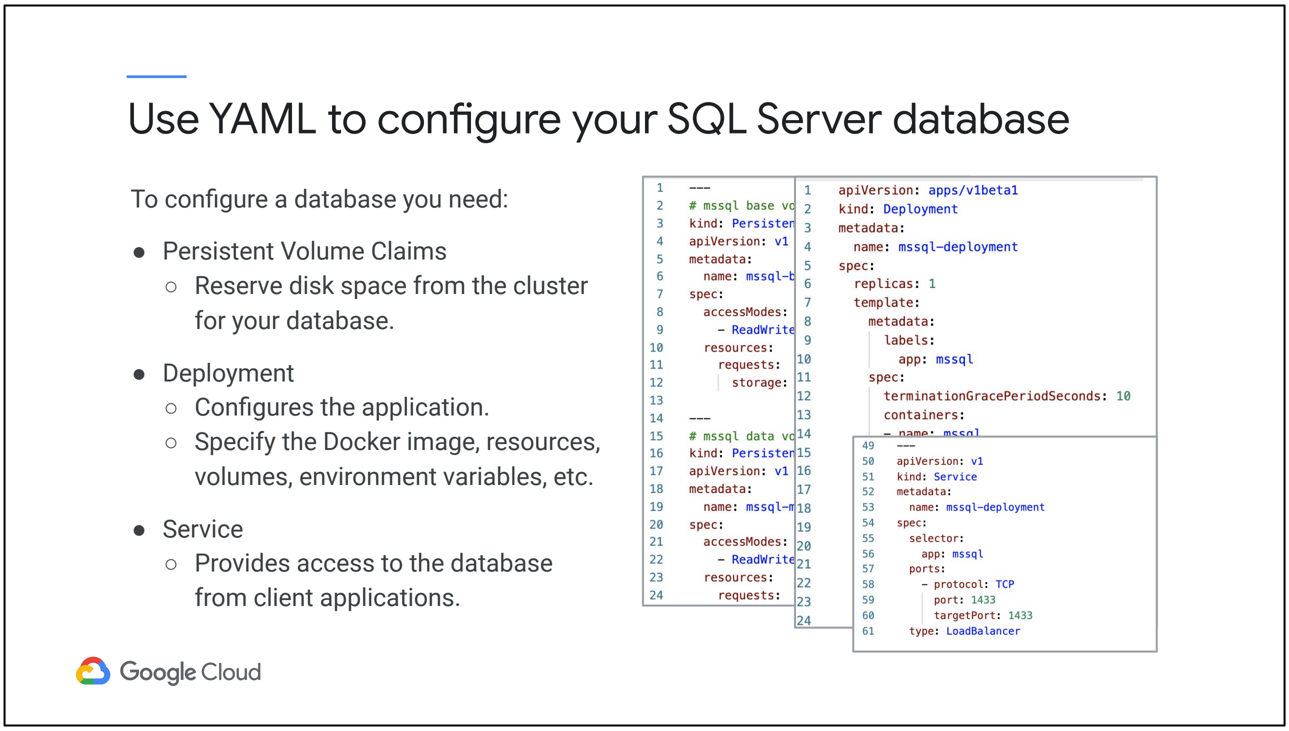
Cross-cloud and hybrid-cloud support

Easy automation for CI/CD pipelines

Google Cloud

Running SQL Server on GKE offers some significant benefits over the other options.

- Most notable is the ability to run the server across various cloud platforms or a hybrid on-premises/off-premises option.
- Using Kubernetes simplifies the automation of deploying cloud resources and applications.
- Additionally, you have a higher degree of automation available for creating CI/CD pipelines.
- It also provides the ability to move the server to the same cluster as the application for better performance.



Use YAML to configure your SQL Server database

To configure a database you need:

- Persistent Volume Claims
  - Reserve disk space from the cluster for your database.
- Deployment
  - Configures the application.
  - Specify the Docker image, resources, volumes, environment variables, etc.
- Service
  - Provides access to the database from client applications.

```yaml
---
# mssql base vo
kind: Persisten
apiVersion: v1
metadata:
  name: mssql-b
spec:
  accessModes:
    - ReadWrite
  resources:
    requests:
      storage:

---
# mssql data vo
kind: Persisten
apiVersion: v1
metadata:
  name: mssql-m
spec:
  accessModes:
    - ReadWrite
  resources:
    requests:
```

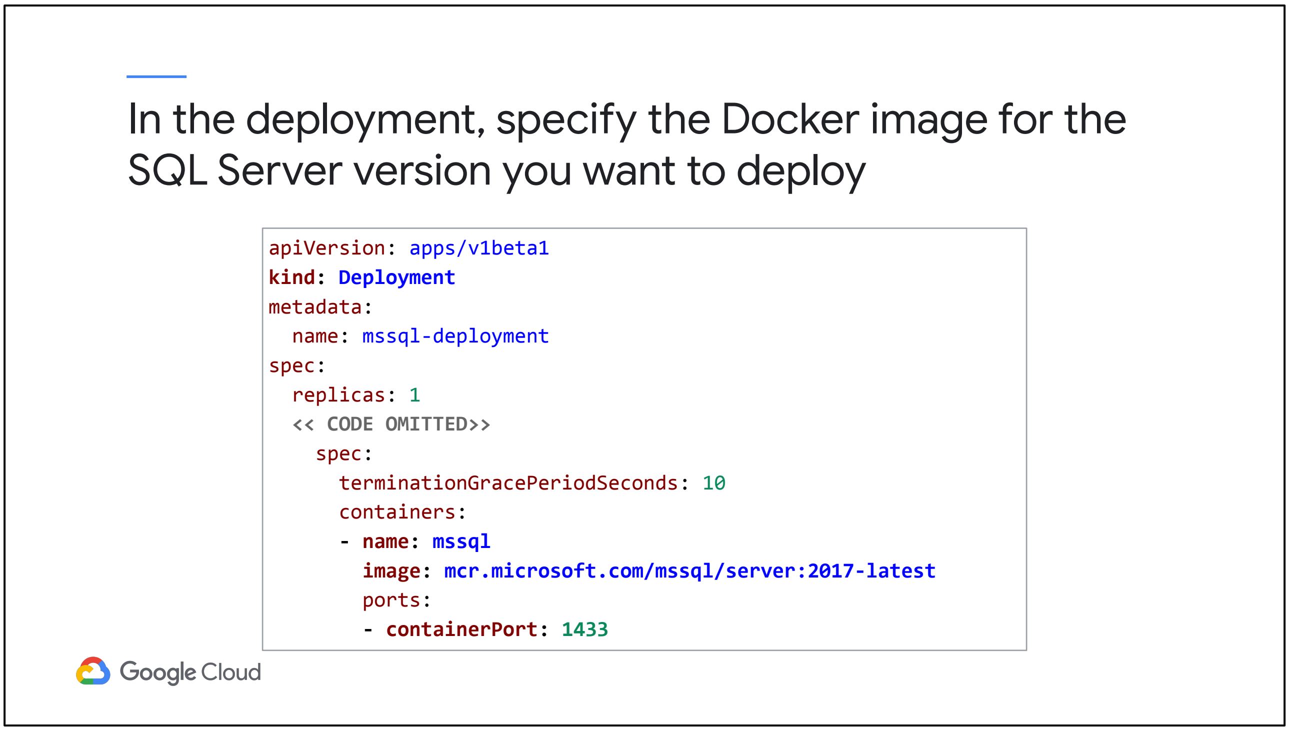
```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: mssql-deployment
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: mssql
    spec:
      terminationGracePeriodSeconds: 10
      containers:
      - name: mssql
```

```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: mssql-deployment
spec:
  selector:
    app: mssql
  ports:
    - protocol: TCP
      port: 1433
      targetPort: 1433
  type: LoadBalancer
```

Google Cloud

You configure SQL Server databases to run in a Kubernetes cluster by using YAML files.

- Persistent Volume claims are used to allocate disk space for your database running in the cluster.
- A Deployment configuration is used to define the Docker image, environment variables, and instance resources.
- And A Service configuration is used to provide access to your SQL Server from its clients.



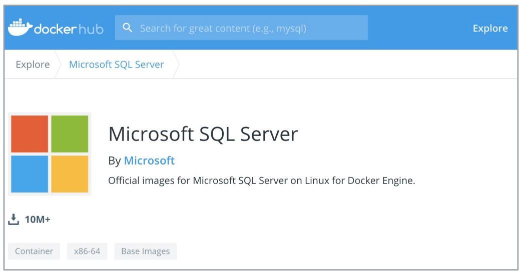
In the deployment, specify the Docker image for the SQL Server version you want to deploy

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: mssql-deployment
spec:
  replicas: 1
  << CODE OMITTED>>
    spec:
      terminationGracePeriodSeconds: 10
      containers:
      - name: mssql
        image: mcr.microsoft.com/mssql/server:2017-latest
        ports:
        - containerPort: 1433
```

Here is an example of using a Microsoft standard Docker image for your server. This image is maintained by Microsoft and stored in Docker Hub. Alternatively, you could build your own Docker image with SQL Server installed if you needed a higher degree of customization.

Official Microsoft SQL Server Docker images can be found at Docker Hub

- Has details on versions, editions, and configurations
- Required environment variables:
  - ACCEPT_EULA=Y
  - SA_PASSWORD=<your-pw>
  - MSSQL_PID=<your_product_id | edition_name> (default: Developer)



Microsoft SQL Server

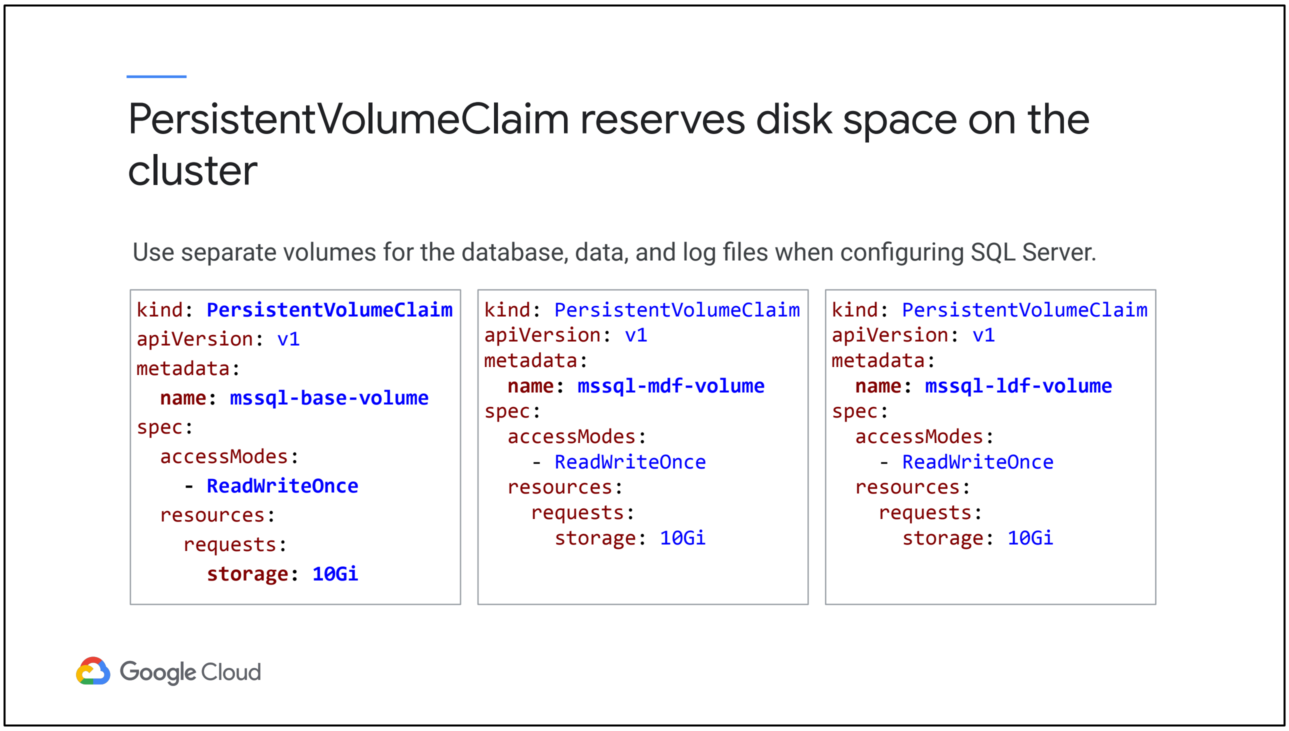
By Microsoft

Official images for Microsoft SQL Server on Linux for Docker Engine.

Google Cloud

Many Microsoft pre-built Docker images are available. Choose the version that suits your needs and make sure to set the required environment variables.

You use license agreements to specify an initial SA password, the edition, and so on. More information can be found at the Docker Hub web page for each image.



PersistentVolumeClaim reserves disk space on the cluster

Use separate volumes for the database, data, and log files when configuring SQL Server.

```yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: mssql-base-volume
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
```

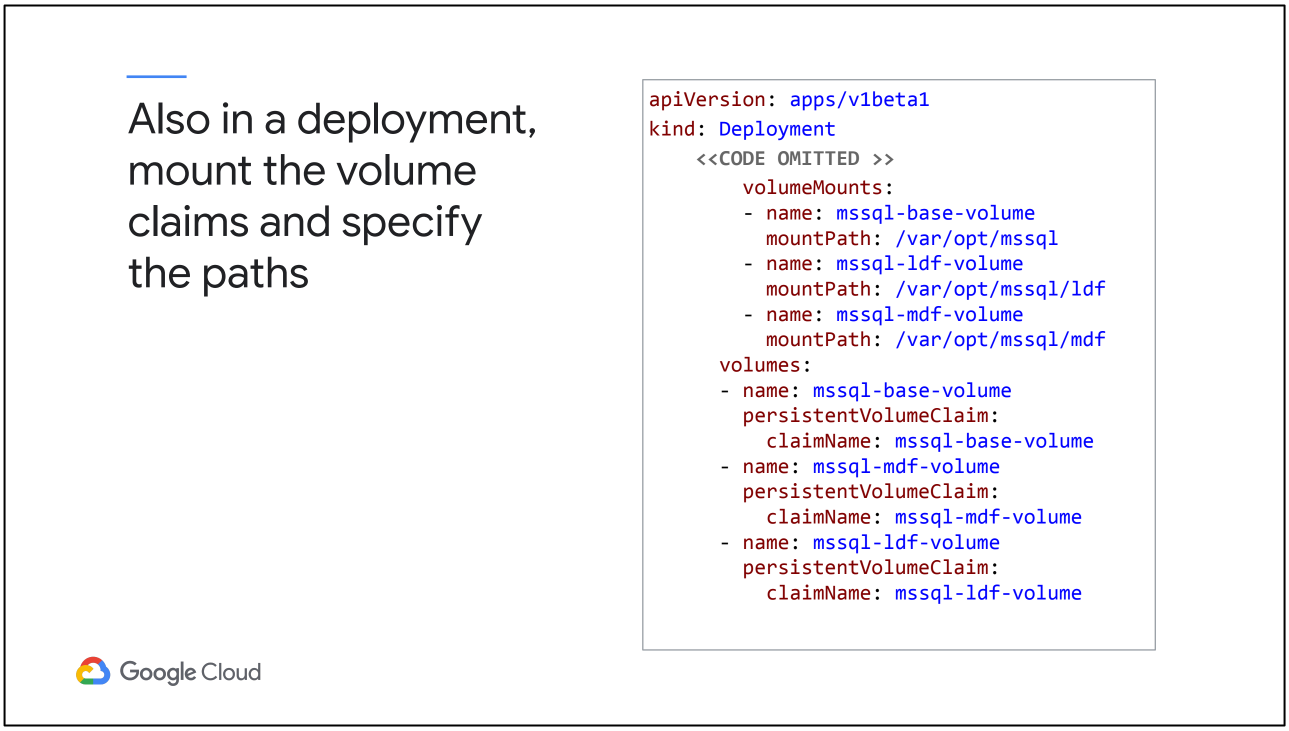
```yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: mssql-mdf-volume
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
```

```yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: mssql-ldf-volume
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
```

Google Cloud

Just as in a normal SQL Server configuration, you want the software, data, log, and tempdb on different physical volumes. So, here you would configure different Persistent Volume Claims for each.

Note that each PersistentVolumeClaim has a unique name, and the amount of disk space for each is specified.

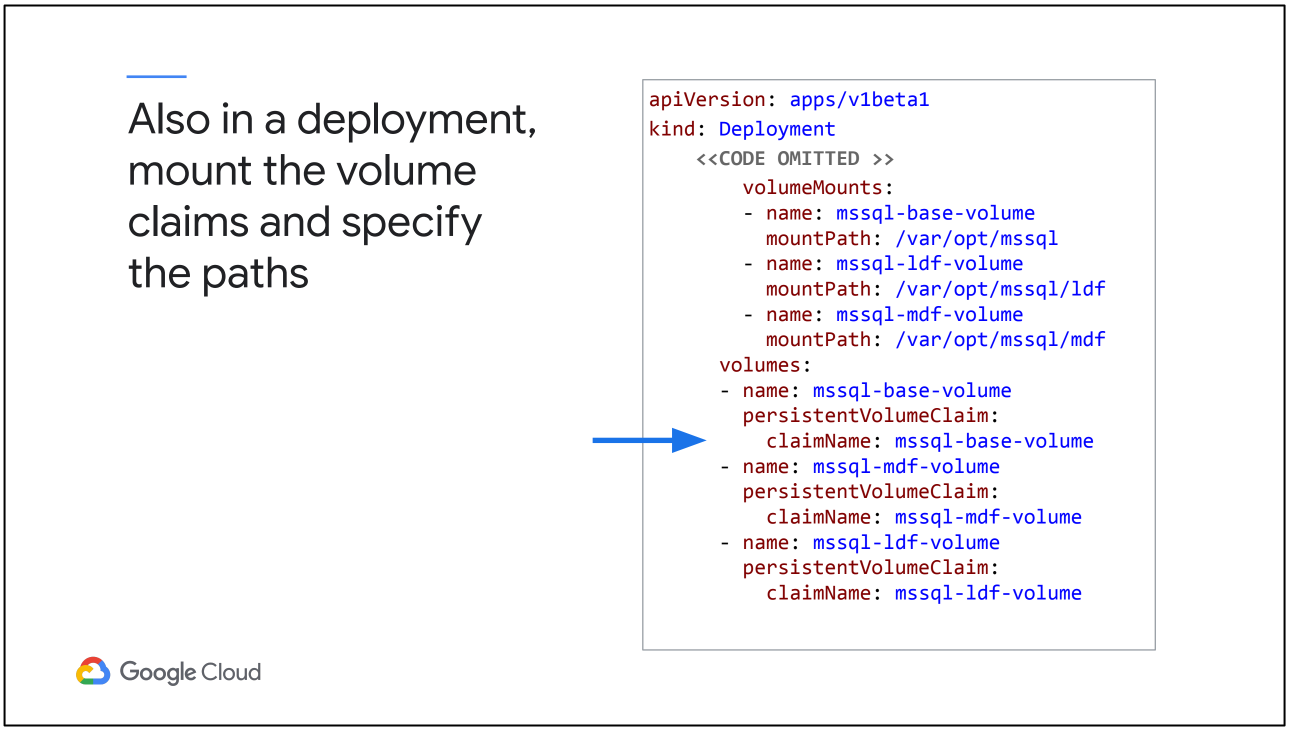


Also in a deployment, mount the volume claims and specify the paths

```
apiVersion: apps/v1beta1
kind: Deployment
    <<CODE OMITTED >>
        volumeMounts:
        - name: mssql-base-volume
          mountPath: /var/opt/mssql
        - name: mssql-ldf-volume
          mountPath: /var/opt/mssql/ldf
        - name: mssql-mdf-volume
          mountPath: /var/opt/mssql/mdf
      volumes:
      - name: mssql-base-volume
        persistentVolumeClaim:
          claimName: mssql-base-volume
      - name: mssql-mdf-volume
        persistentVolumeClaim:
          claimName: mssql-mdf-volume
      - name: mssql-ldf-volume
        persistentVolumeClaim:
          claimName: mssql-ldf-volume
```

Google Cloud

Finally, put all the pieces together in a Deployment.

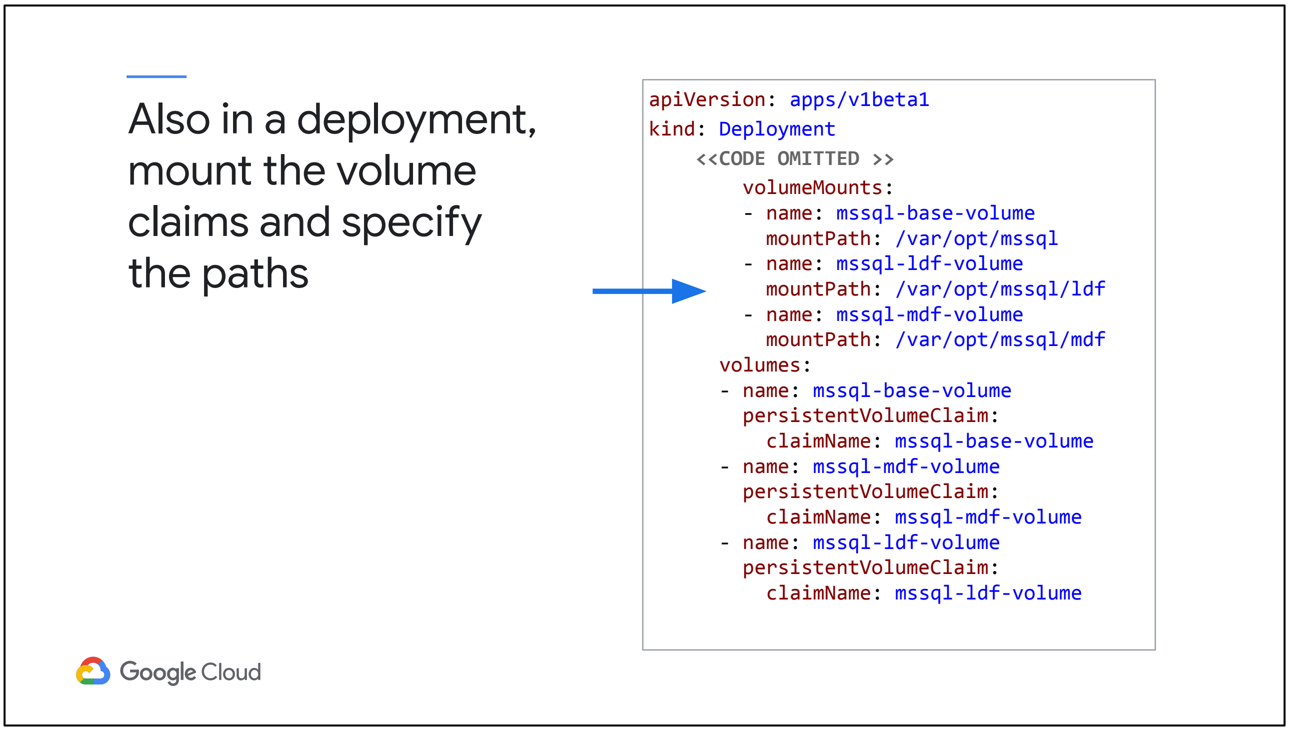


Also in a deployment, mount the volume claims and specify the paths

```
apiVersion: apps/v1beta1
kind: Deployment
  <<CODE OMITTED >>
    volumeMounts:
    - name: mssql-base-volume
      mountPath: /var/opt/mssql
    - name: mssql-ldf-volume
      mountPath: /var/opt/mssql/ldf
    - name: mssql-mdf-volume
      mountPath: /var/opt/mssql/mdf
   volumes:
   - name: mssql-base-volume
     persistentVolumeClaim:
       claimName: mssql-base-volume
   - name: mssql-mdf-volume
     persistentVolumeClaim:
       claimName: mssql-mdf-volume
   - name: mssql-ldf-volume
     persistentVolumeClaim:
       claimName: mssql-ldf-volume
```

Google Cloud

The Persistent Volume Claims from the previous slide are added as Volumes in the pod you are configuring.

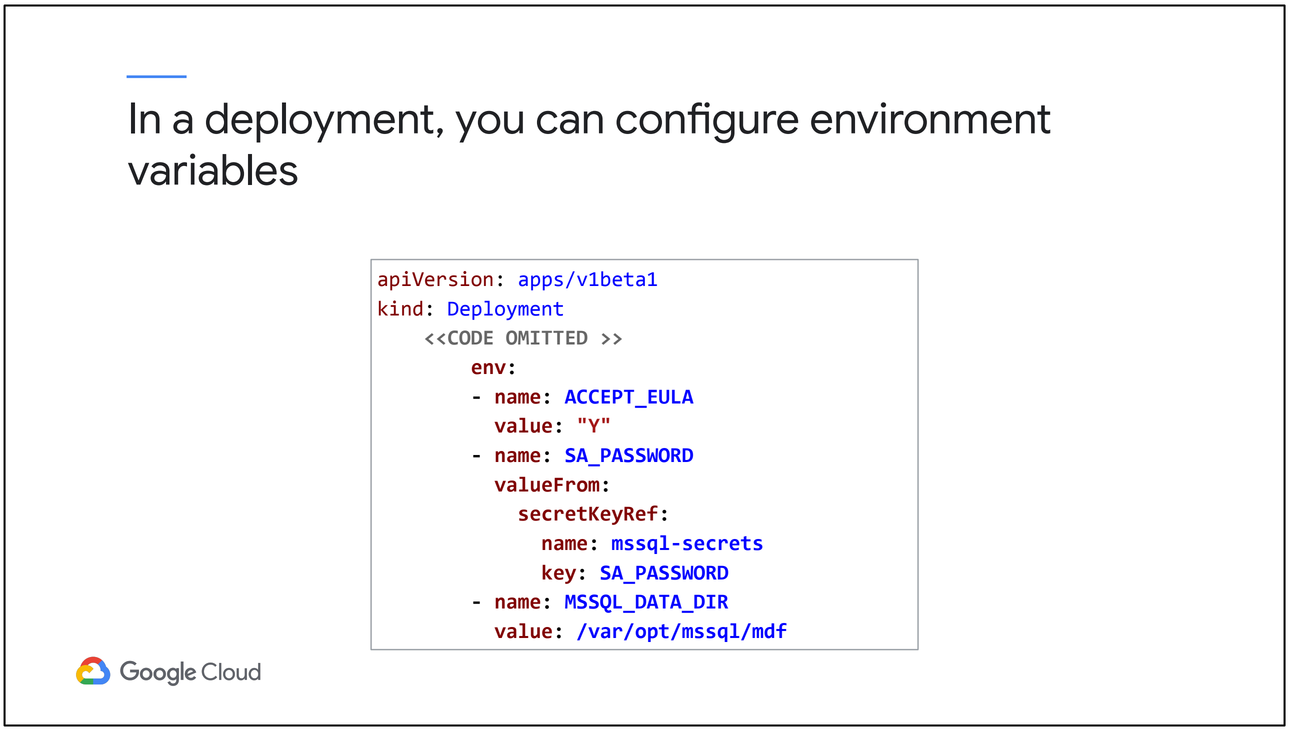


Also in a deployment, mount the volume claims and specify the paths

```
apiVersion: apps/v1beta1
kind: Deployment
    <<CODE OMITTED >>
        volumeMounts:
        - name: mssql-base-volume
          mountPath: /var/opt/mssql
        - name: mssql-ldf-volume
          mountPath: /var/opt/mssql/ldf
        - name: mssql-mdf-volume
          mountPath: /var/opt/mssql/mdf
      volumes:
      - name: mssql-base-volume
        persistentVolumeClaim:
          claimName: mssql-base-volume
      - name: mssql-mdf-volume
        persistentVolumeClaim:
          claimName: mssql-mdf-volume
      - name: mssql-ldf-volume
        persistentVolumeClaim:
          claimName: mssql-ldf-volume
```

Google Cloud

Then, the volumes are mounted using the mountPath property when you configure the SQL Server container that will also run in the pod. Note that path names are Linux, not Windows, paths.

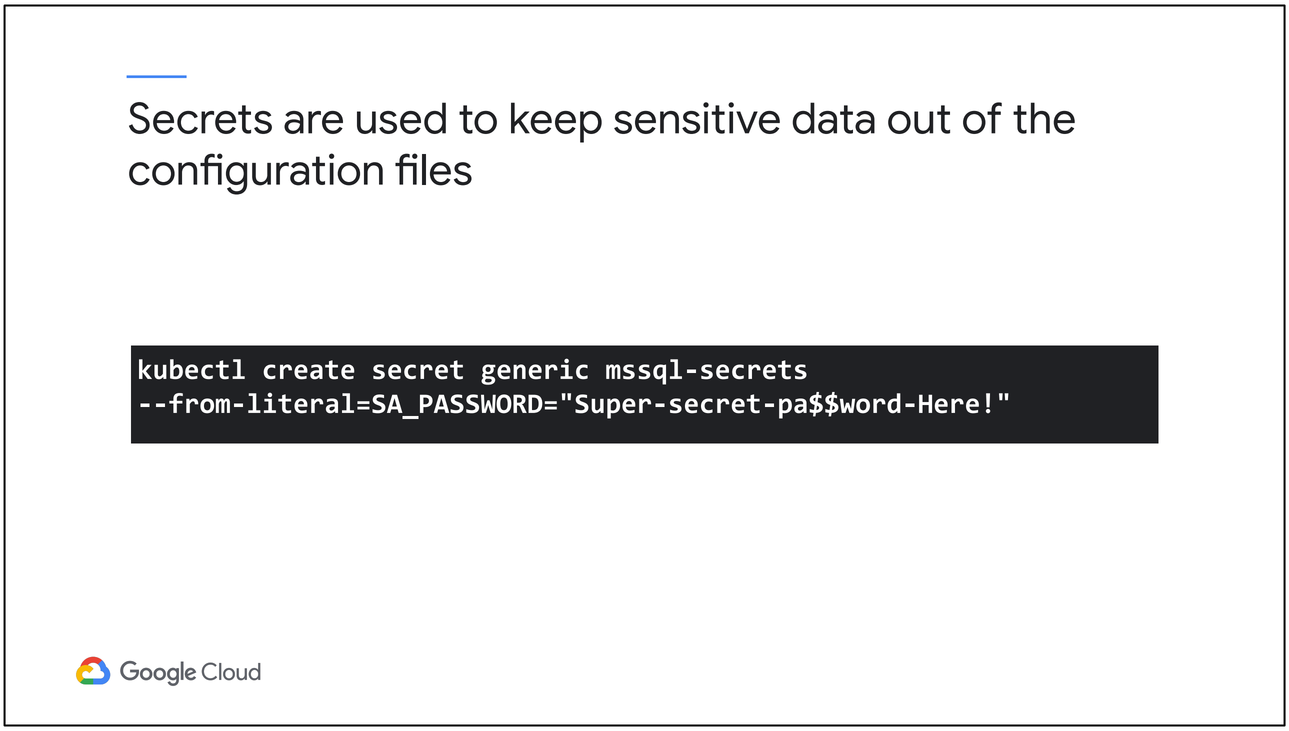


In a deployment, you can configure environment variables

```
apiVersion: apps/v1beta1
kind: Deployment
    <<CODE OMITTED >>
        env:
        - name: ACCEPT_EULA
          value: "Y"
        - name: SA_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mssql-secrets
              key: SA_PASSWORD
        - name: MSSQL_DATA_DIR
          value: /var/opt/mssql/mdf
```

Google Cloud

You can also configure the required environment variables here. Check out the SA_PASSWORD variable. This is being set from a Kubernetes secret.



Secrets are used to keep sensitive data out of the configuration files

```
kubectl create secret generic mssql-secrets
--from-literal=SA_PASSWORD="Super-secret-pa$$word-Here!"
```

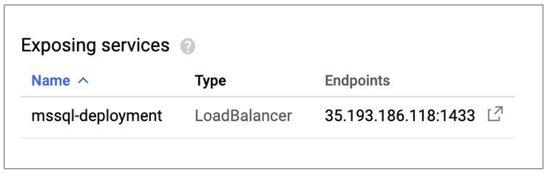
Google Cloud

Storing sensitive data in plain text in a configuration file is never a good idea, so you can use secrets to store the actual password and use the secret in the config file.

Here the password is being set using the CLI. Run this code before the deployment.

Service provides access to the database running in the cluster

```yaml
apiVersion: v1
kind: Service
metadata:
  name: mssql-deployment
spec:
  selector:
    app: mssql
  ports:
    - protocol: TCP
      port: 1433
      targetPort: 1433
  type: LoadBalancer
```



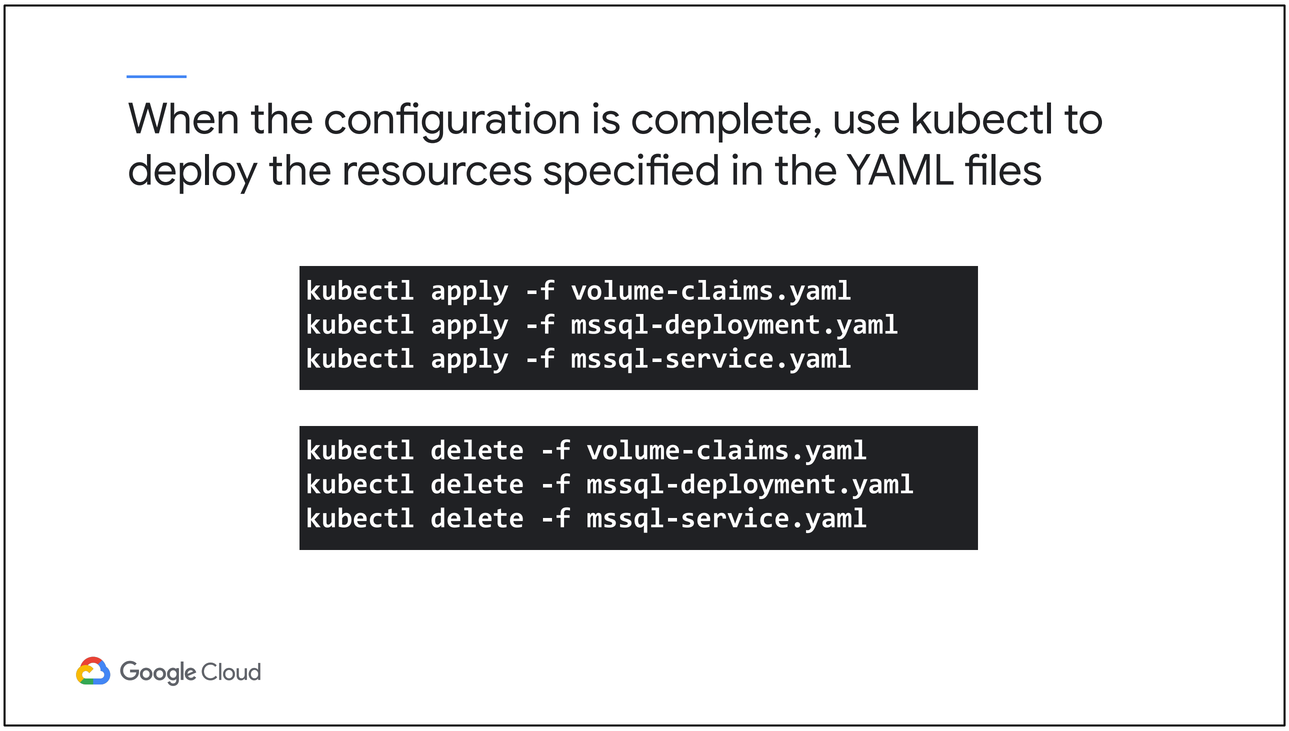
Exposing services

Name	Type	Endpoints
mssql-deployment	LoadBalancer	35.193.186.118:1433

Google Cloud

The service configuration provides access to the database running inside the cluster. Note the port and target port variables: 1433 is the default SQL Server port.

The service type LoadBalancer creates a public IP address, allowing the database to be accessed from outside the cluster. If you set the type to ClusterIP (or didn't set the type property at all), the service would only have a private IP address. Thus, the database would only be available from inside the cluster.



When the configuration is complete, use kubectl to deploy the resources specified in the YAML files

```
kubectl apply -f volume-claims.yaml
kubectl apply -f mssql-deployment.yaml
kubectl apply -f mssql-service.yaml
```

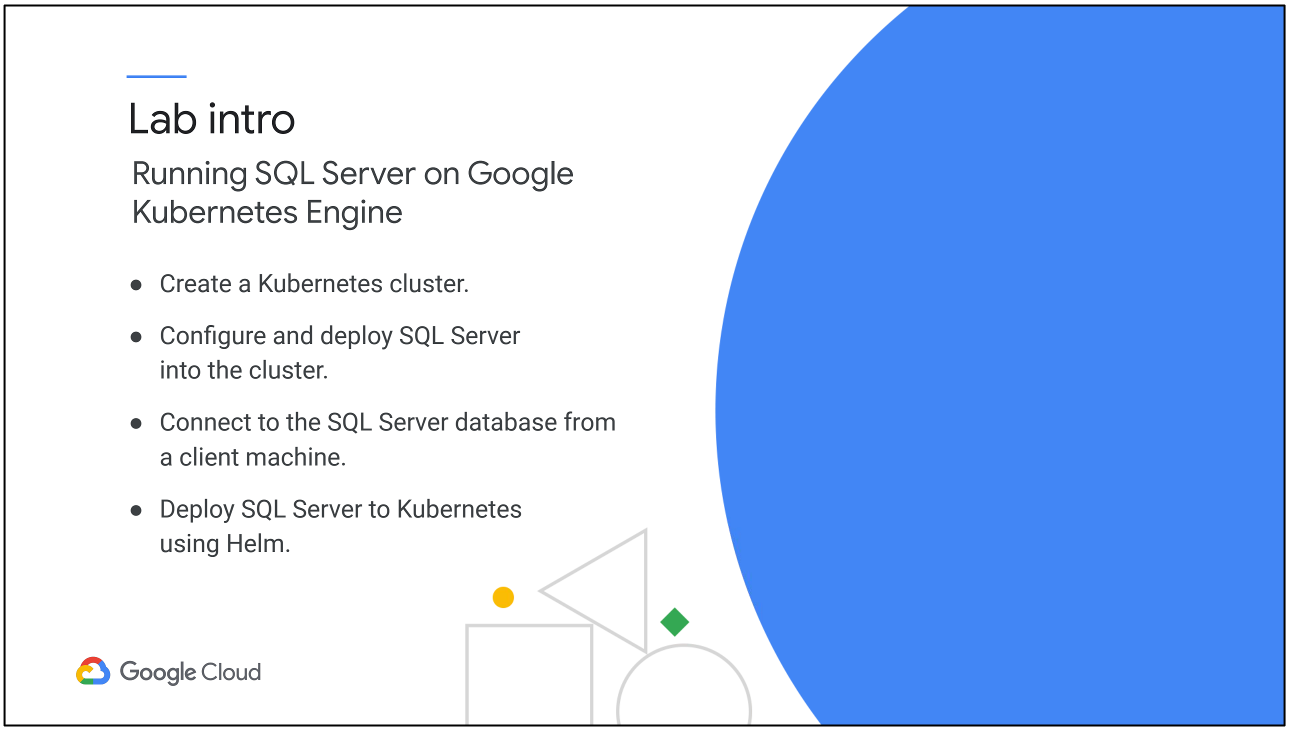
```
kubectl delete -f volume-claims.yaml
kubectl delete -f mssql-deployment.yaml
kubectl delete -f mssql-service.yaml
```

Google Cloud

After you configure the various files, you can use the kubectl command line interface to put them all together to deploy the server.

Deploy each of the YAML files with the command `kubectl apply -f` and the name of the file.

You can delete everything by running `kubectl delete`.



Lab intro

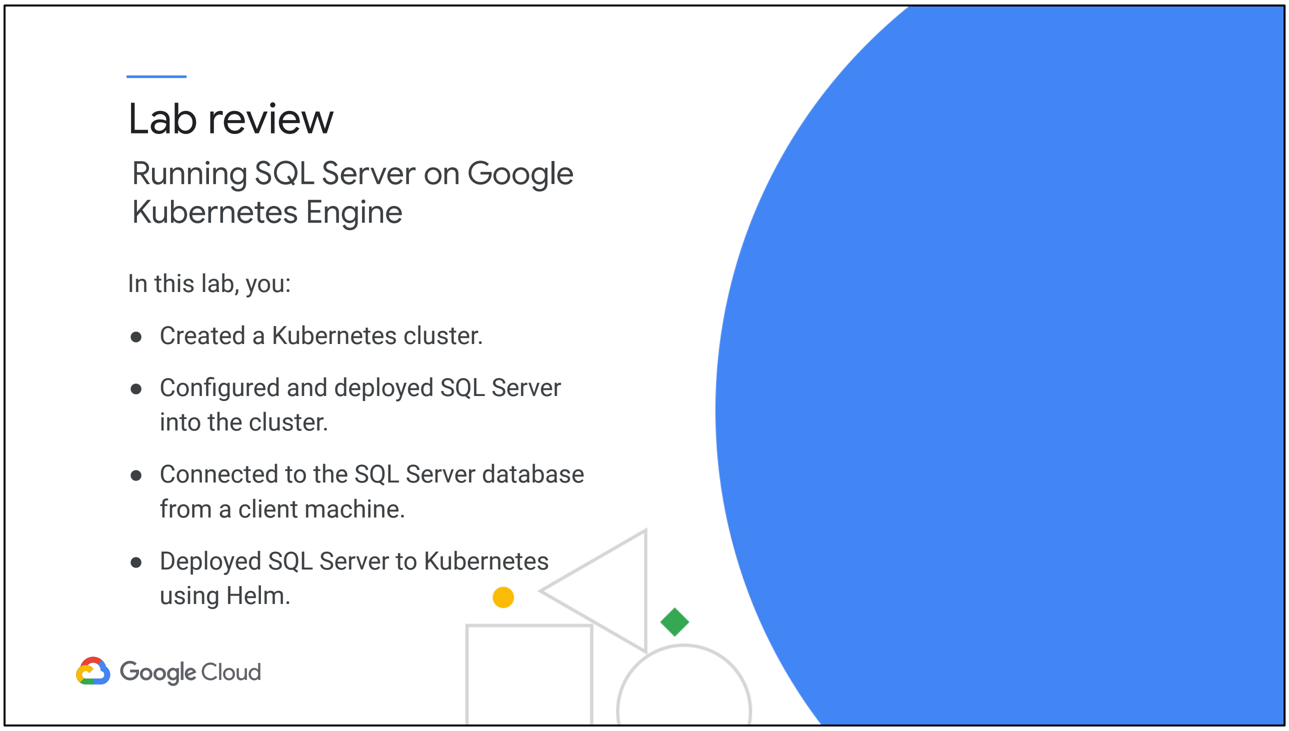
Running SQL Server on Google Kubernetes Engine

- Create a Kubernetes cluster.
- Configure and deploy SQL Server into the cluster.
- Connect to the SQL Server database from a client machine.
- Deploy SQL Server to Kubernetes using Helm.

Google Cloud

Kubernetes is becoming an increasingly popular way to deploy automated, cross-cloud and hybrid cloud applications. Applications of course need databases. So, if you're deploying your web apps and services to Kubernetes, why not deploy your databases there as well.

This lab gives you the opportunity to create a Kubernetes cluster and configure and deploy a SQL Server database in it. Then, you connect to the database from a client and look at an alternative way to deploy SQL Server to the cluster using Helm.



Lab review

Running SQL Server on Google Kubernetes Engine

In this lab, you:

- Created a Kubernetes cluster.
- Configured and deployed SQL Server into the cluster.
- Connected to the SQL Server database from a client machine.
- Deployed SQL Server to Kubernetes using Helm.

Google Cloud

In this lab, you created a Kubernetes cluster, and configured and deployed SQL Server into the cluster.

You then connected to the SQL Server database from a client machine.

Lastly, you deployed SQL Server to Kubernetes using Helm.

Running your SQL Server databases in a Kubernetes cluster is ideal when you want a cross-cloud or hybrid-cloud solution that is automated.

Module review

Google Cloud

In this module, you:

- Used Compute Engine to lift and shift SQL Server databases to Google Cloud.
- Employed Cloud SQL for managed SQL Server databases.
- Architected SQL Server for security, high availability, and disaster recovery.
- And configured SQL Server to run with Kubernetes on GKE.

Google Cloud

Google Cloud