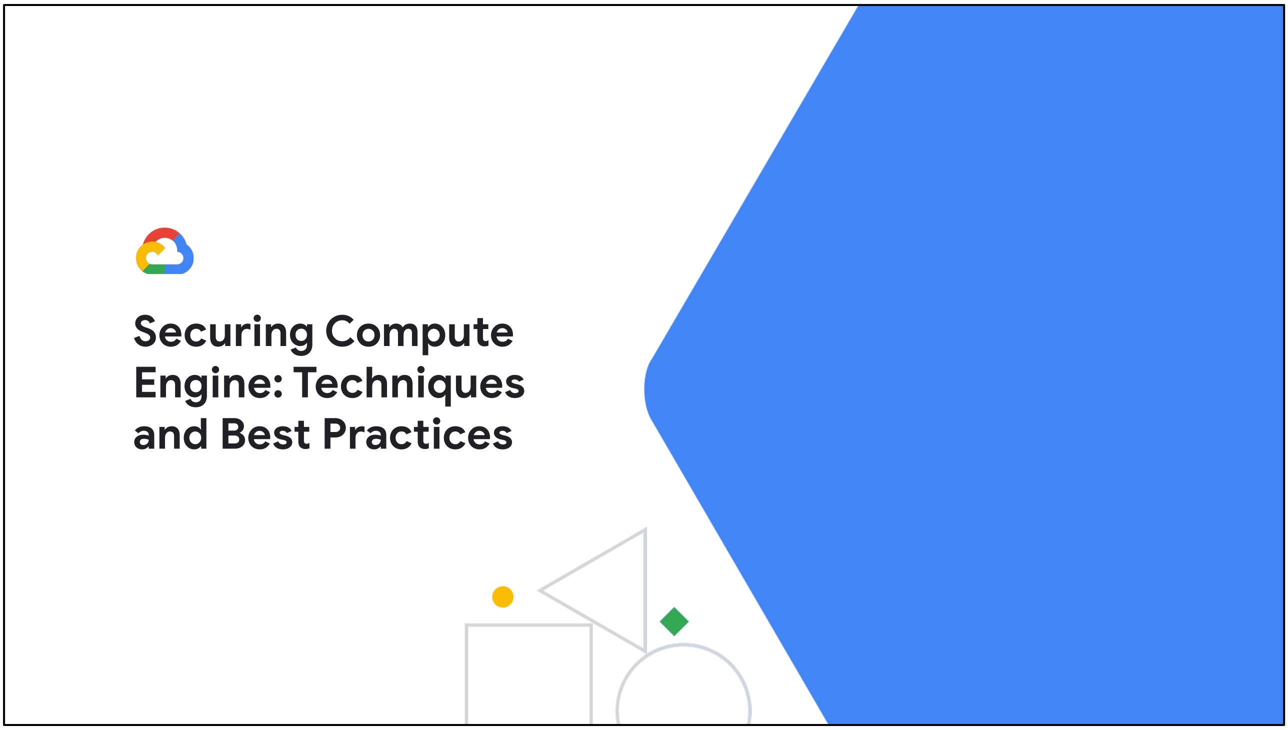
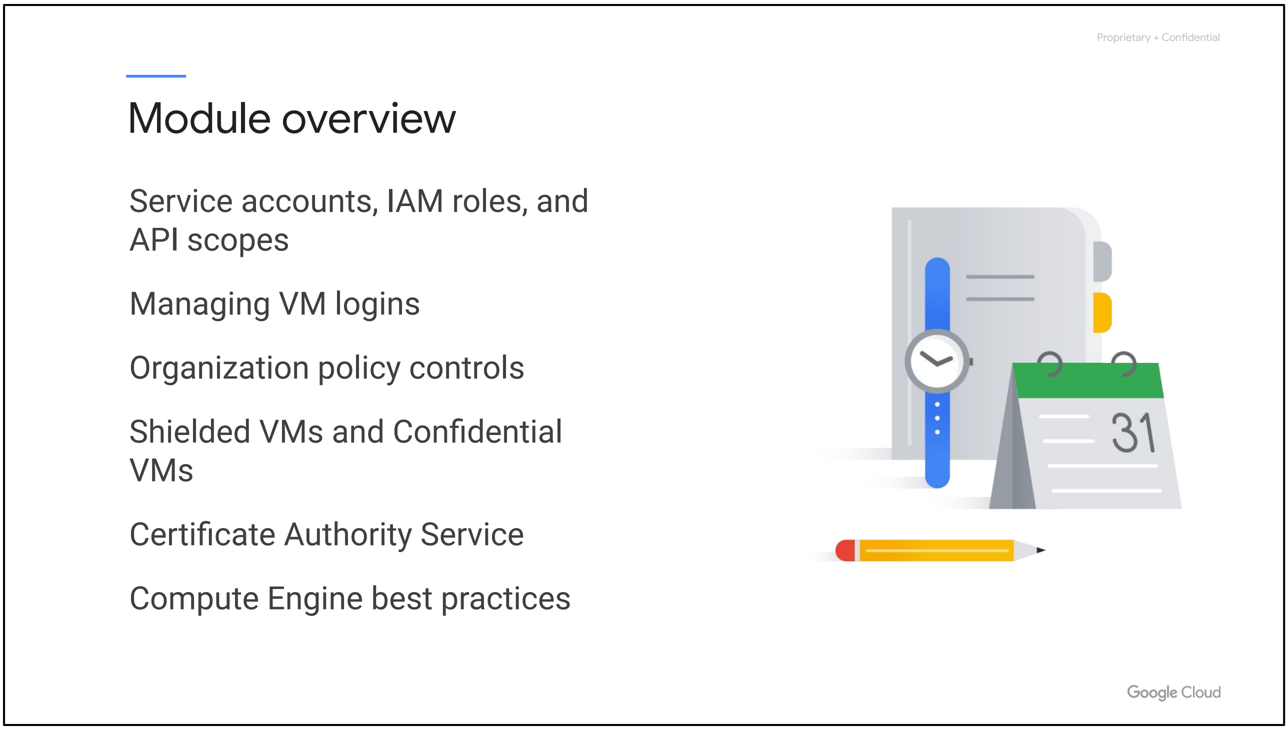


# Securing Compute Engine: Techniques and Best Practices

Welcome to the first module of Security Best Practices in Google Cloud, **Securing Compute Engine: Techniques and Best Practices**.



Proprietary + Confidential

## Module overview

Service accounts, IAM roles, and API scopes

Managing VM logins

Organization policy controls

Shielded VMs and Confidential VMs

Certificate Authority Service

Compute Engine best practices

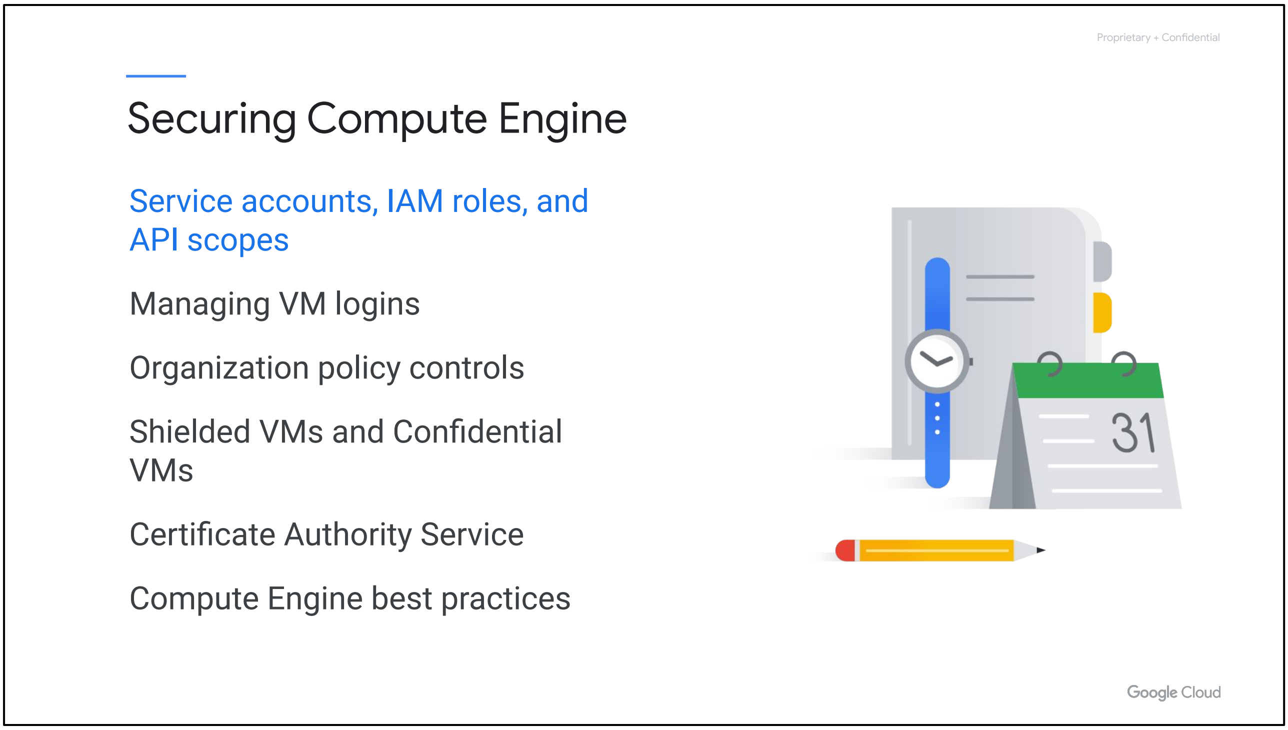
Google Cloud

Compute Engine security encompasses many different topics.

In this module we will start with a discussion of service accounts, IAM roles and API scopes as they apply to Compute Engine.

We will also discuss managing VM logins, organization policy controls, as well as shielded VMs and confidential VMs.

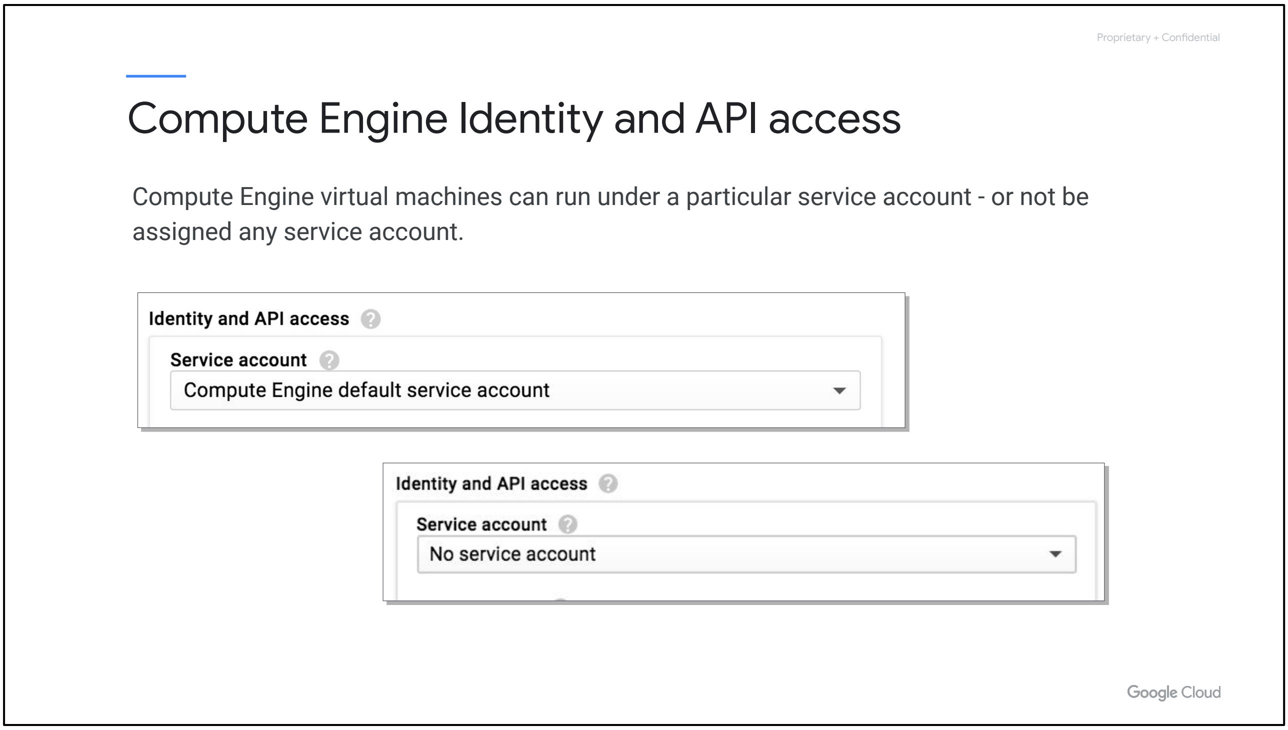
We will wrap up by discussing Certificate Authority Service and best practices for Securing Compute Engine.



# Securing Compute Engine

- Service accounts, IAM roles, and API scopes
- Managing VM logins
- Organization policy controls
- Shielded VMs and Confidential VMs
- Certificate Authority Service
- Compute Engine best practices

OK, let’s get started with Service accounts, IAM roles and API scopes.



Proprietary + Confidential

# Compute Engine Identity and API access

Compute Engine virtual machines can run under a particular service account - or not be assigned any service account.

Identity and API access

Service account: Compute Engine default service account

Identity and API access

Service account: No service account

Google Cloud

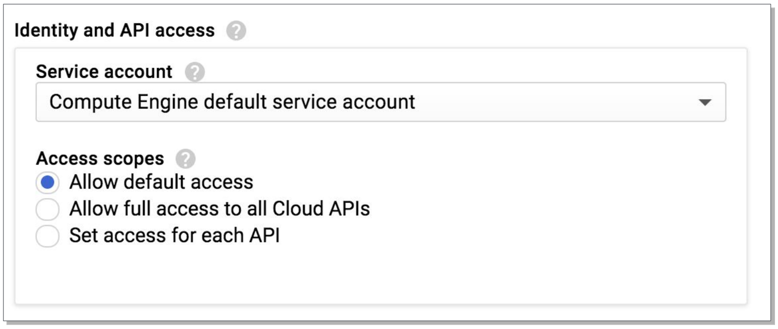
Service accounts are an identity that a resource such as a VM instance can use to run API requests on your behalf. When launching a virtual machine in Compute Engine, a service account can be associated directly to that VM. When a service account is specified, the VM authenticates using the identity of that service account when making calls to the Google APIs.

It also possible to specify no service account association. In that case, API requests running on the VM will not assume the service account identity by default and would therefore need to be manually configured.

Since service accounts control how resources are managed and used, it’s very important that you provide them roles and permissions using the principle of least privilege.

# Default service account

- Created automatically when the Compute Engine is enabled.
- Assigned the Project Editor role.
- Used by default when creating a VM.



Identity and API access

Service account

Compute Engine default service account

Access scopes

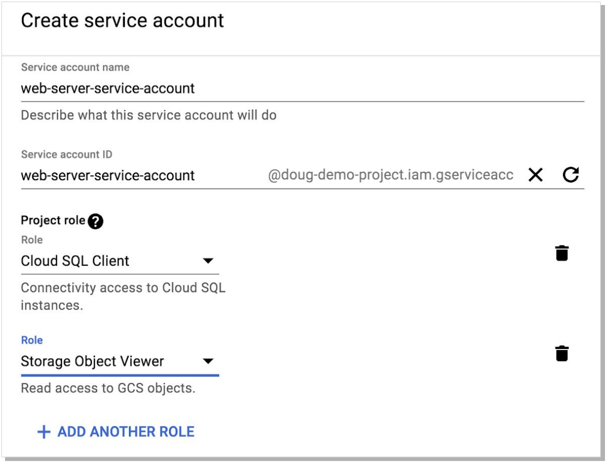
● Allow default access

○ Allow full access to all Cloud APIs

○ Set access for each API

Every project has a default service account that is automatically created when Compute Engine is first enabled for the project. In this instance the service account is assigned the role of Project Editor and is used by default when launching VMs.

# Create service accounts using IAM



Create service account

Service account name
web-server-service-account

Describe what this service account will do

Service account ID
web-server-service-account @doug-demo-project.iam.gserviceacc

Project role

Role
Cloud SQL Client
Connectivity access to Cloud SQL instances.

Role
Storage Object Viewer
Read access to GCS objects.

You can also create and manage your own service accounts using Identity and Access Management. These user-managed service accounts are granted “necessary” permissions just like any member in IAM - by assigning roles.

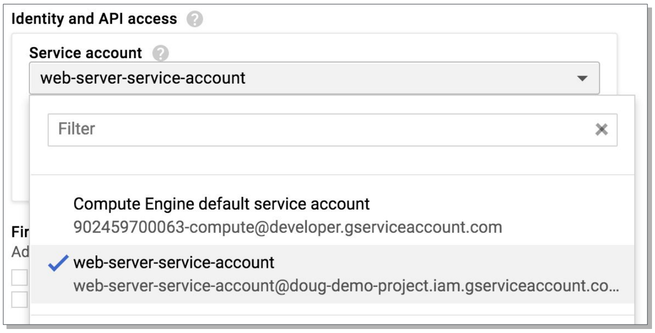
This gives you full control over exactly which permissions the service account will have.

If you do not grant any roles, the service account will not have any access to services.

# Assign custom service accounts to machines

Access to APIs controlled by the roles, not by scopes:

- Assign 1 or more roles to those service accounts.
- Scopes are only used by default service accounts.



Identity and API access

Service account

web-server-service-account

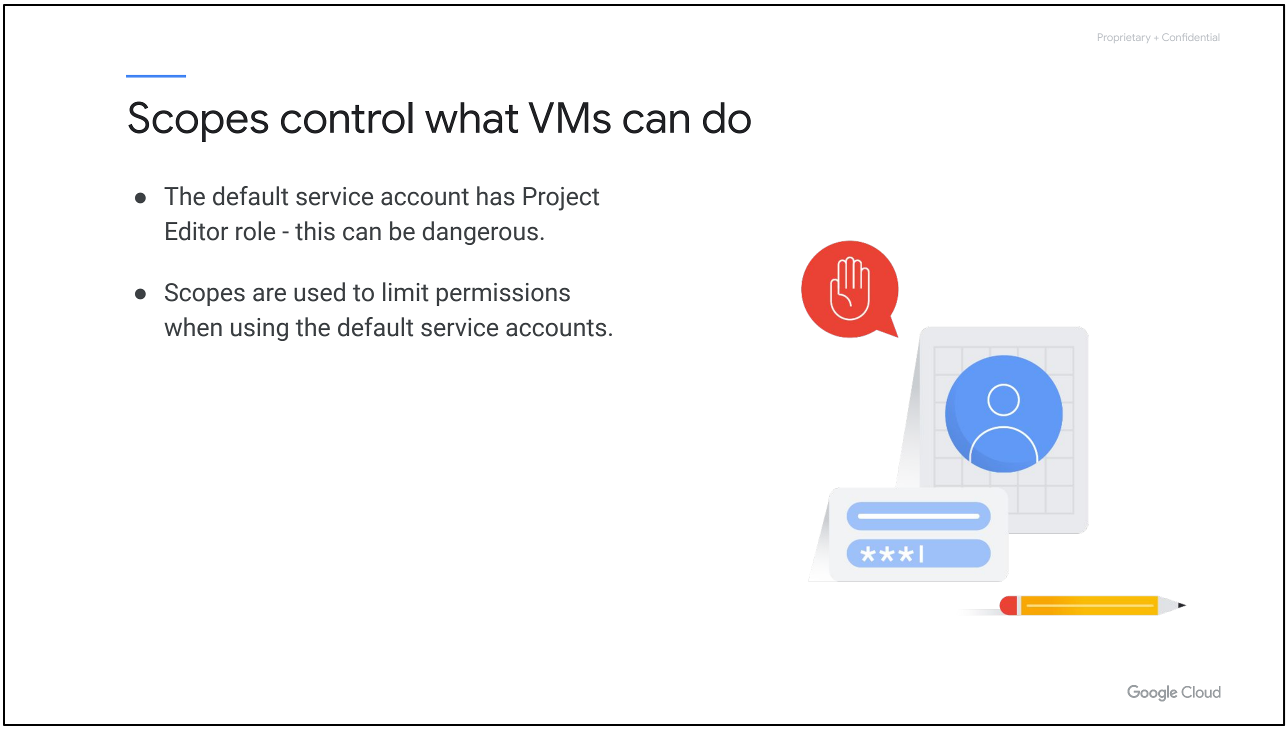
Filter

Compute Engine default service account
902459700063-compute@developer.gserviceaccount.com

✓ web-server-service-account
web-server-service-account@doug-demo-project.iam.gserviceaccount.co...

When you create a new service account, it can be assigned to instances in exactly the same way as the default service account. The only difference is user-managed service accounts do not use the “access scope” concept, which we’ll cover in the next slide.

Instead, permissions are controlled through the IAM roles assigned to the account. Applications running on instances associated with the service account can make authenticated requests to other Google APIs using the service account identity.



## Scopes control what VMs can do

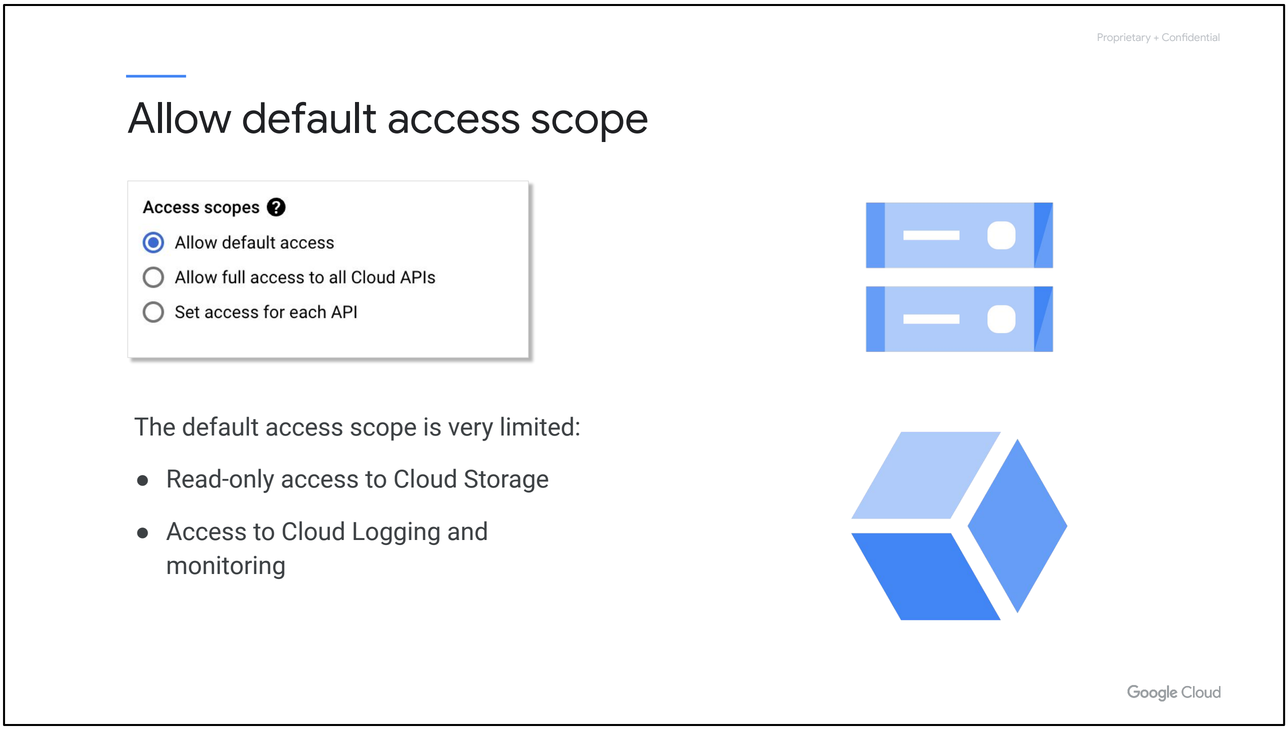
- The default service account has Project Editor role - this can be dangerous.
- Scopes are used to limit permissions when using the default service accounts.

Google Cloud

The IAM Project Editor role contains permissions to create and delete resources for most Google Cloud services and can be dangerous to use as-is. Access scopes provide the ability to limit what permissions are allowed when using the default service account containing this role.

Before the existence of IAM roles, access scopes were the only mechanism for granting permissions to service accounts. Although they are not the primary way of granting permissions now, you must still configure access scopes when initiating an instance to run under the *default* service account.

It is important to remember that access scopes only apply on a *per-instance* basis. You set access scopes when creating an instance and the access scopes persists only for the life of the instance.



# Allow default access scope

Access scopes

● Allow default access

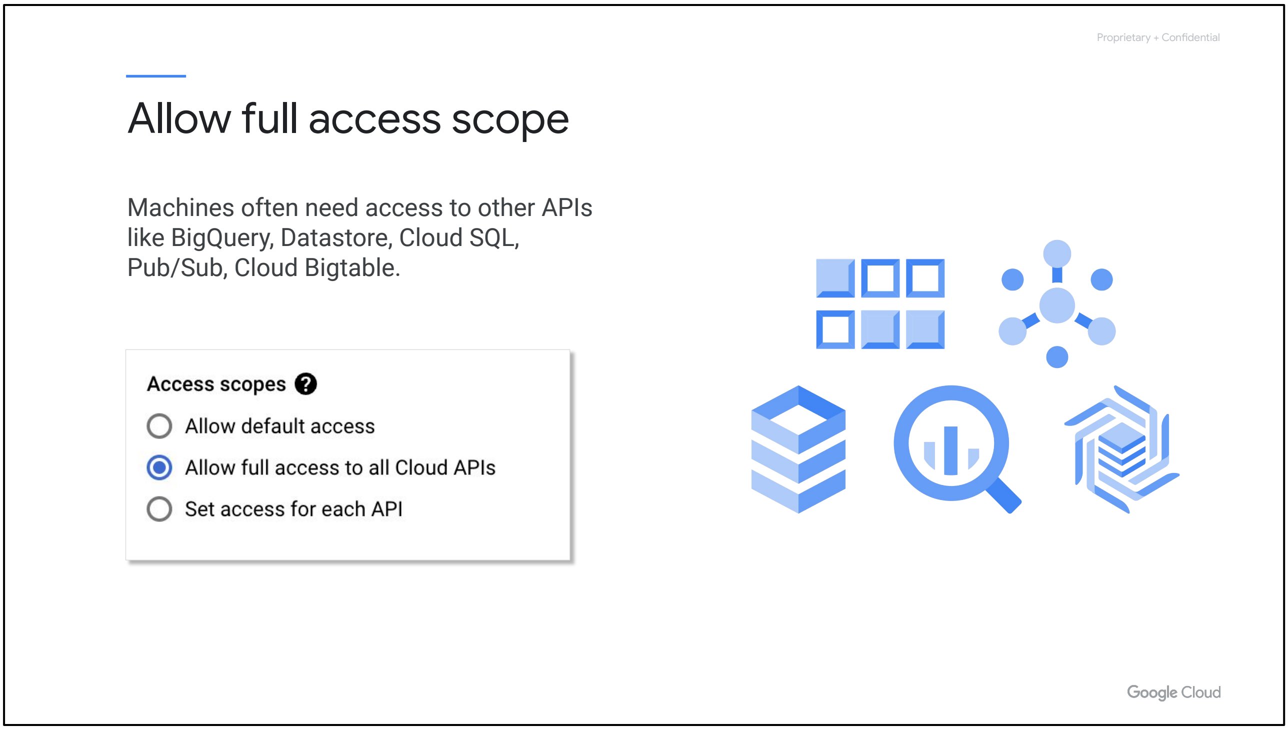
○ Allow full access to all Cloud APIs

○ Set access for each API

The default access scope is very limited:

- Read-only access to Cloud Storage
- Access to Cloud Logging and monitoring

There are several options when setting access scopes. The first is called “Allow default access”. The default access scope is actually very narrow and allows read-only access to Cloud Storage, as well as access to Cloud Logging and Cloud Monitoring. Other API access using the default service account will obviously be restricted.



Proprietary + Confidential

# Allow full access scope

Machines often need access to other APIs like BigQuery, Datastore, Cloud SQL, Pub/Sub, Cloud Bigtable.

**Access scopes** ?

○ Allow default access

● Allow full access to all Cloud APIs

○ Set access for each API

Google Cloud

Consider the situation where your VMs need access to other APIs, such as BigQuery, Datastore, Cloud SQL, Pub/Sub, or Cloud Bigtable. The default access scope does not include these APIs and would cause a security error when accessing these, or other APIs not included in scope.

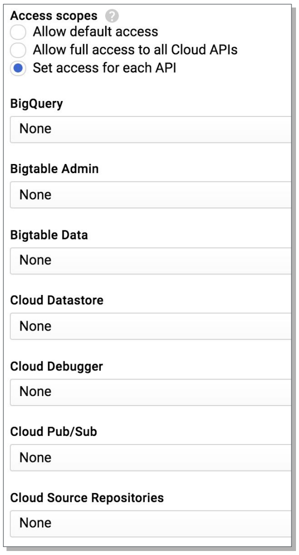
The next access scope option is to “allow full access”. This grants full access to all Cloud APIs. Choosing this option would violate the Principle of Least Privilege, and therefore is definitely **NOT** a best practice!

Proprietary + Confidential

# Set access for each API with scopes

Can grant access to only to the APIs required by the programs running on the machine:

- Choose only the scopes required by your application.
- Better practice than granting full access.



**Access scopes**

○ Allow default access
○ Allow full access to all Cloud APIs
● Set access for each API

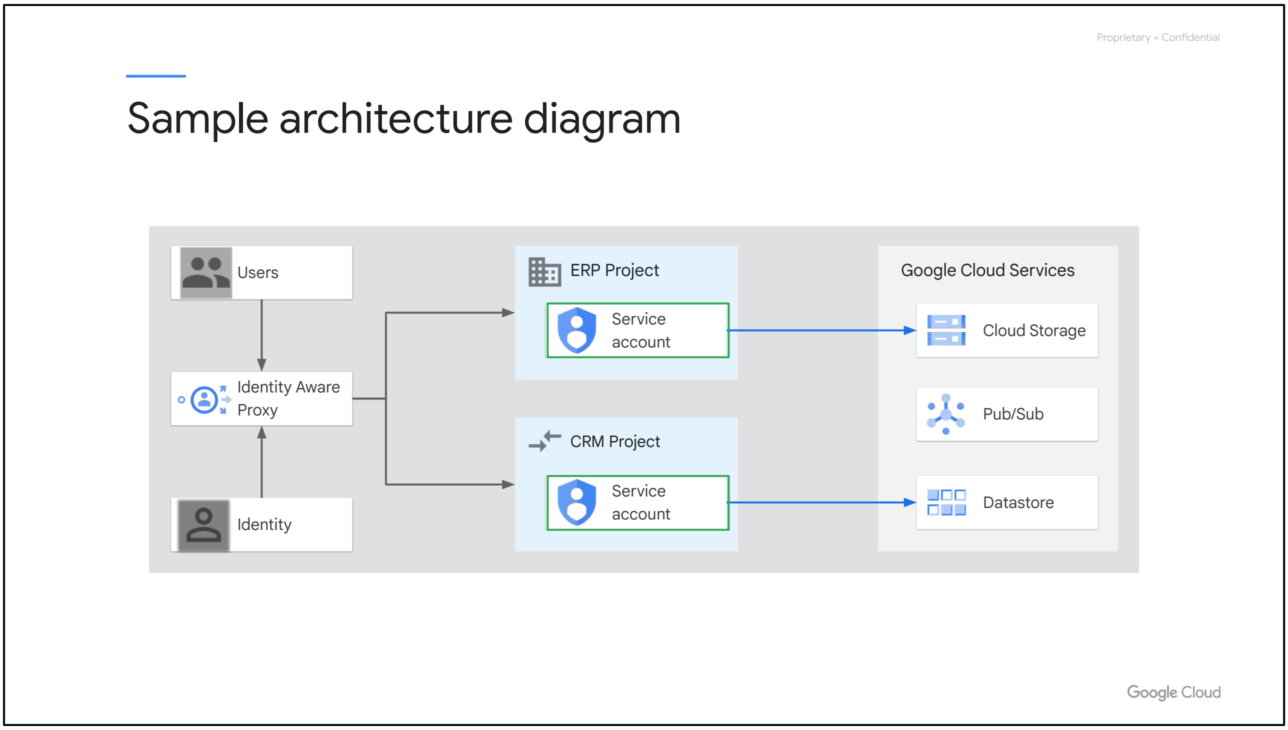
| | |
|---|---|
| BigQuery | None |
| Bigtable Admin | None |
| Bigtable Data | None |
| Cloud Datastore | None |
| Cloud Debugger | None |
| Cloud Pub/Sub | None |
| Cloud Source Repositories | None |

Google Cloud

There is another option, and that is to set the each API access required individually. This will allow you to grant access to only the APIs required by the programs running on the VM.

You can choose only the scopes required by your application.

If you are using the default service account, then this is a much better practice than granting full API access.

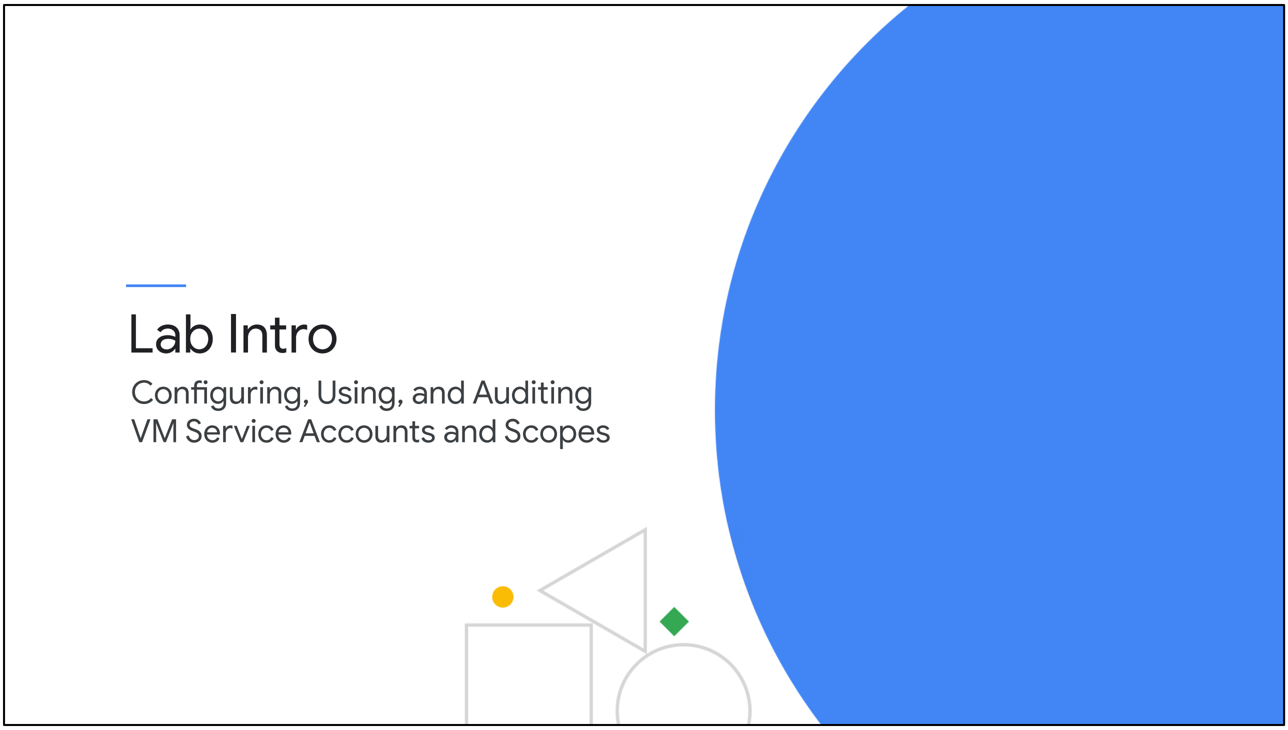


Proprietary + Confidential

# Sample architecture diagram

Google Cloud

This diagram illustrates an example of where a service account might fit in an enterprise’s architecture running on Google Cloud.

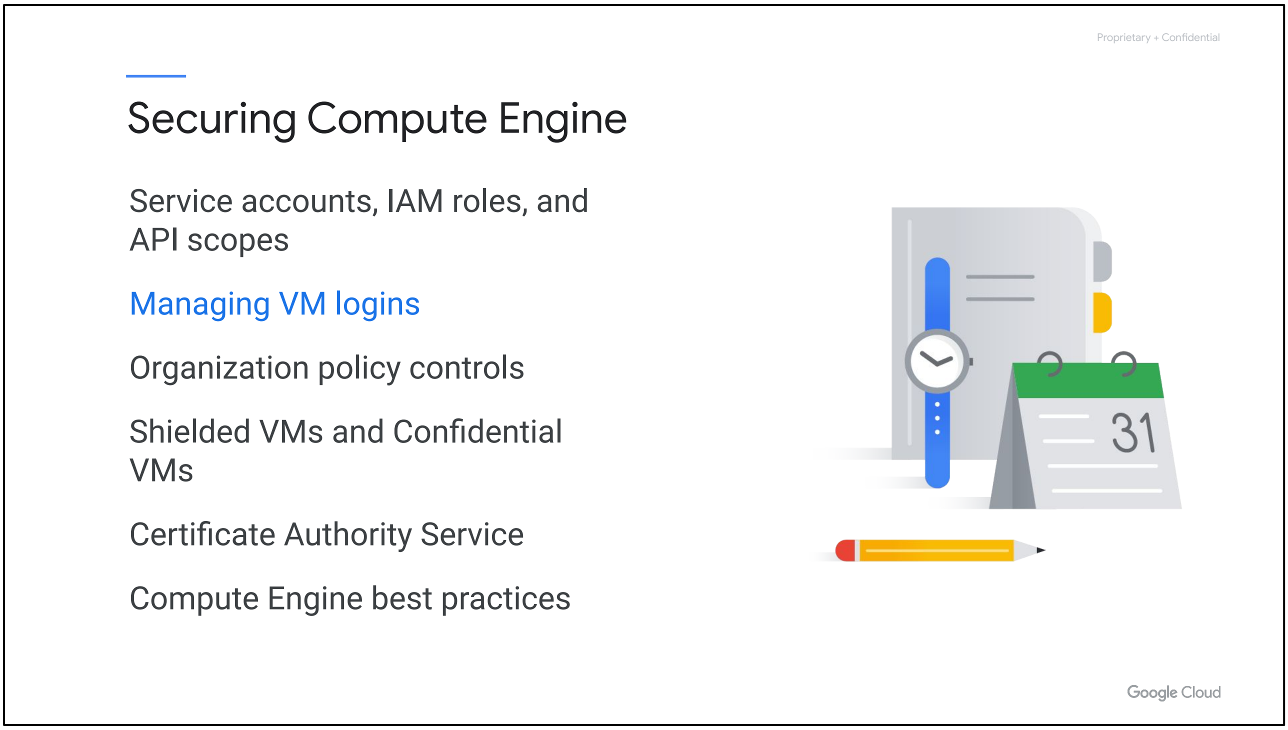


# Lab Intro

Configuring, Using, and Auditing
VM Service Accounts and Scopes

OK, now you will get a chance to configure and use service accounts and scopes. In this lab, you will learn how to:

- Create and manage service accounts.
- Create a virtual machine and associate it with a service account.
- Use client libraries to access BigQuery from a service account.
- And run a query on a BigQuery public dataset from a Compute Engine instance.



## Securing Compute Engine

Service accounts, IAM roles, and API scopes

Managing VM logins

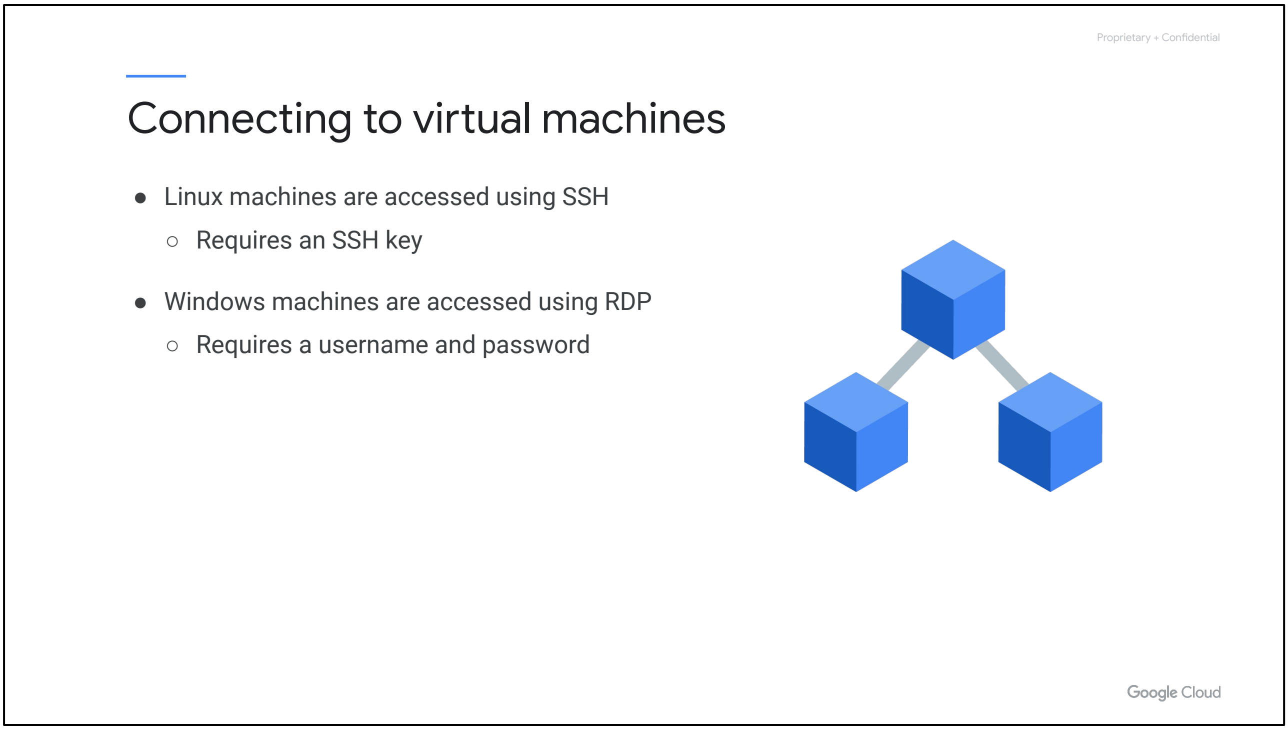
Organization policy controls

Shielded VMs and Confidential VMs

Certificate Authority Service

Compute Engine best practices

In this section, we will discuss various options for logging into and securing VMs.



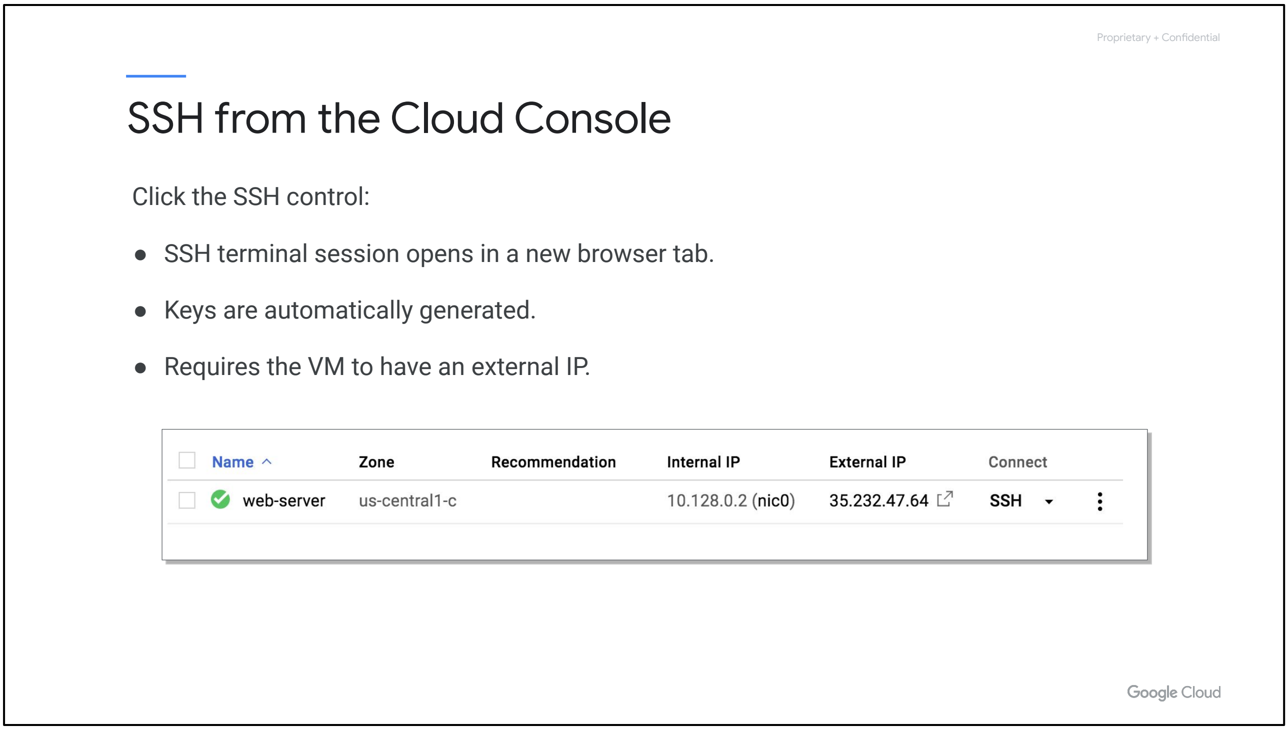
## Connecting to virtual machines

- Linux machines are accessed using SSH
  - Requires an SSH key
- Windows machines are accessed using RDP
  - Requires a username and password

Connecting to virtual machines in the cloud is generally very easy.

By default, Linux instances on Google Cloud are accessed with Secure Shell (i.e SSH) and require a username and an SSH key for authentication. Password authentication is disabled by default.

Window instances are accessed with Remote Desktop Protocol (i.e., RDP) and require a username and password to authenticate.



## SSH from the Cloud Console

Click the SSH control:

- SSH terminal session opens in a new browser tab.
- Keys are automatically generated.
- Requires the VM to have an external IP.

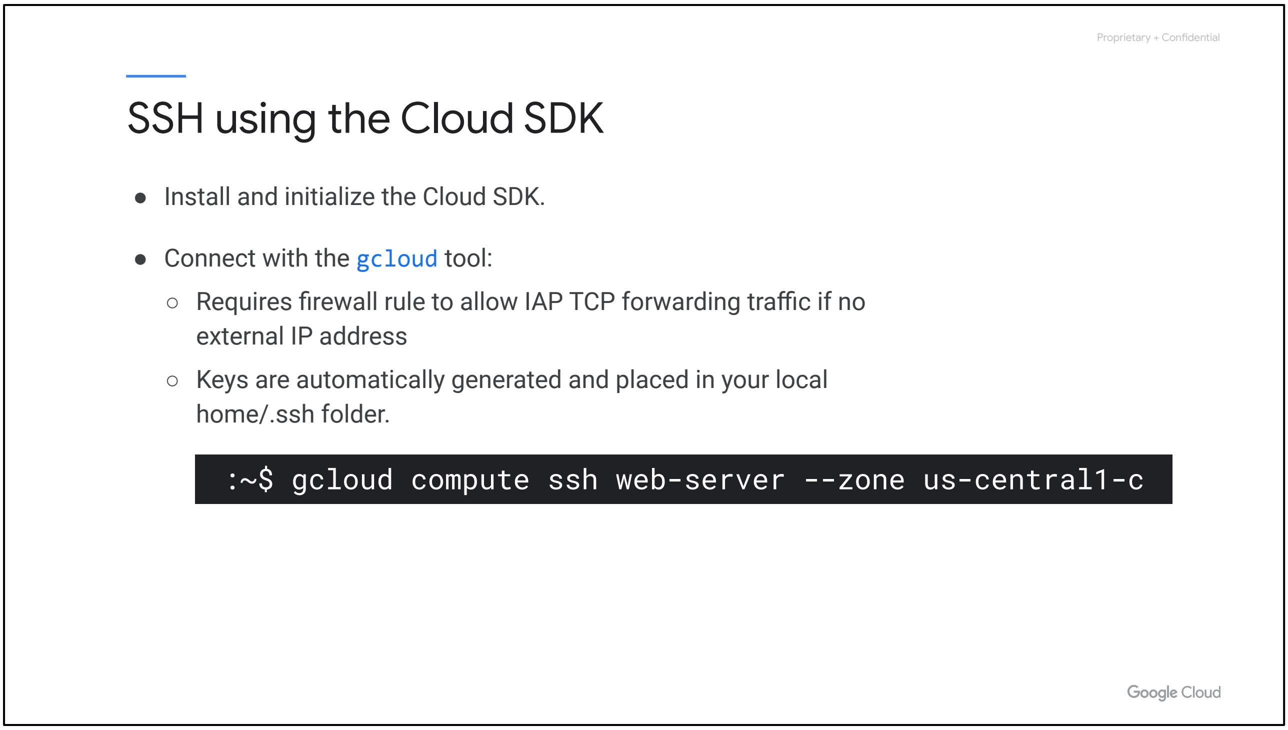
| □ | Name ^ | Zone | Recommendation | Internal IP | External IP | Connect | |
|---|---|---|---|---|---|---|---|
| □ | ✓ web-server | us-central1-c | | 10.128.0.2 (nic0) | 35.232.47.64 | SSH ▾ | ⋮ |

Google Cloud

When connecting to Linux instances, the Cloud Console provides a built-in SSH access mechanism. To connect to an instance, simply click the SSH button in the console and a SSH terminal session will open in a new browser window.

As part of the connection process, the browser window performs an HTTPS connection to a Google Web server, which in turn creates an SSH connection to the instance. SSH keys are automatically generated and propagated to the instance during this process.

For this to work, the VM must have a public IP address and a firewall rule to allow TCP port 22 traffic from Google's servers.



## SSH using the Cloud SDK

- Install and initialize the Cloud SDK.
- Connect with the `gcloud` tool:
  - Requires firewall rule to allow IAP TCP forwarding traffic if no external IP address
  - Keys are automatically generated and placed in your local home/.ssh folder.

```
:~$ gcloud compute ssh web-server --zone us-central1-c
```

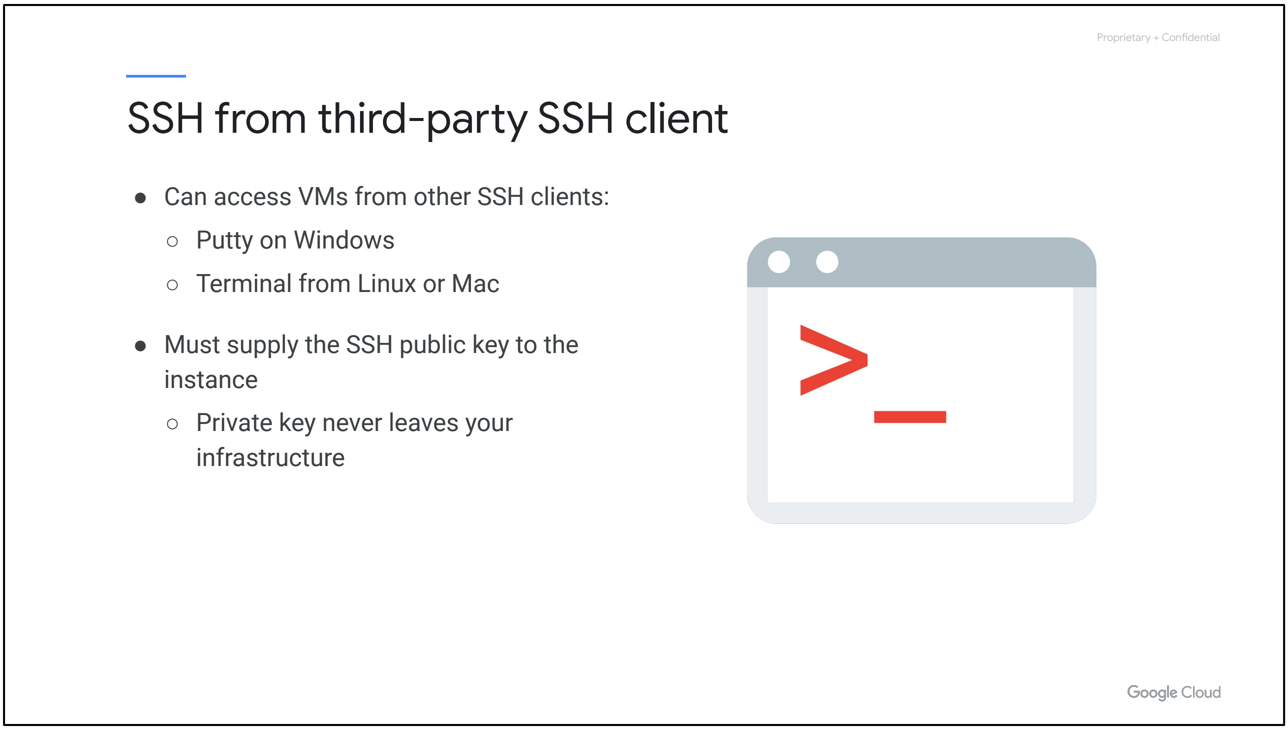
Google Cloud

It is also possible to connect to a Linux instance via SSH using the gcloud SDK. Once you have the Cloud SDK installed and configured, simply connect with the command `gcloud compute ssh` , passing in the <instance-name> and the <zone-name>.

In this command, the <instance-name> is the name given to the instance when it was launched. Notice you do not need to connect to, or even know, the VMs IP address.

If your VM instance has no external IP address, you need a firewall rule in place that allows IAP TCP forwarding traffic.

The <zone-name> is the name of the zone where the instance is running. Running this command will automatically generate SSH keys and place a copy of them in your local home/.ssh folder.



Proprietary + Confidential

## SSH from third-party SSH client

- Can access VMs from other SSH clients:
  - Putty on Windows
  - Terminal from Linux or Mac
- Must supply the SSH public key to the instance
  - Private key never leaves your infrastructure

Google Cloud

But what if you just want to SSH into the instance and do not have access to either the console or gcloud credentials? Don’t worry, there are further options available.

It is possible to connect from alternative SSH clients such as PuTTY on Windows or similar applications on Linux or Mac operating systems. To connect remember you just need to SSH to the public IP address of the instance and provide a valid username and SSH private key to authenticate.

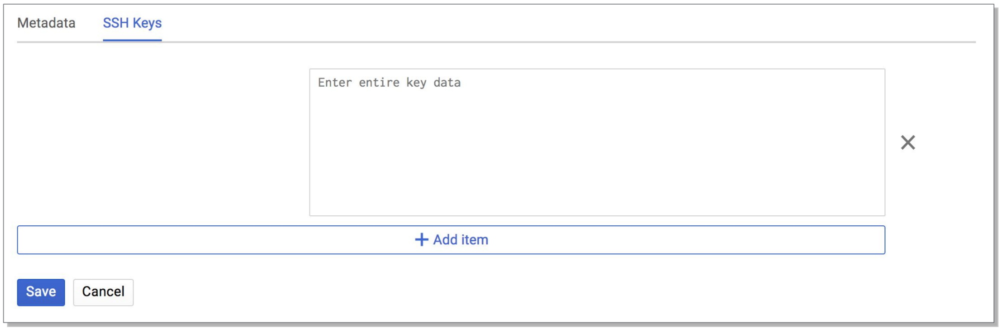
The firewall rule for the instance still requires TCP port 22 to be allowed for the IP address range of your SSH client. Note the SSH keys to use in this case are managed outside of Google Cloud. You can manually create your own SSH key pairs, using tools like PuTTYgen or ssh-keygen.

The public key must be provided to the instance that you wish to authenticate against, but your private key never leaves your infrastructure.

# Adding SSH keys to projects

Can add SSH keys as project metadata:

- Provide only the public key.
- Automatically added to all VMs by default.

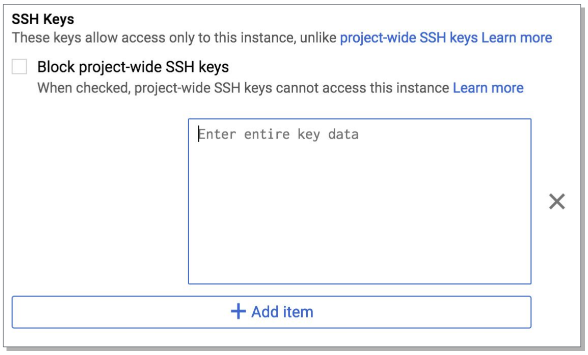


The public key is provided to the instance using the project metadata. The project metadata can be accessed in the Cloud Console from the Compute Engine dashboard. Simply select the “add item” option and upload your public key.

Note: by default, all keys added to the project metadata are available to **ALL** VMs in the project.

# Adding SSH keys to instances

- Can configure instances to NOT use project-wide keys:
  - Can specify public key for individual instances.
- Add SSH keys to instance metadata when creating a VM:
  - Provide access to only this machine.



SSH Keys

These keys allow access only to this instance, unlike project-wide SSH keys Learn more

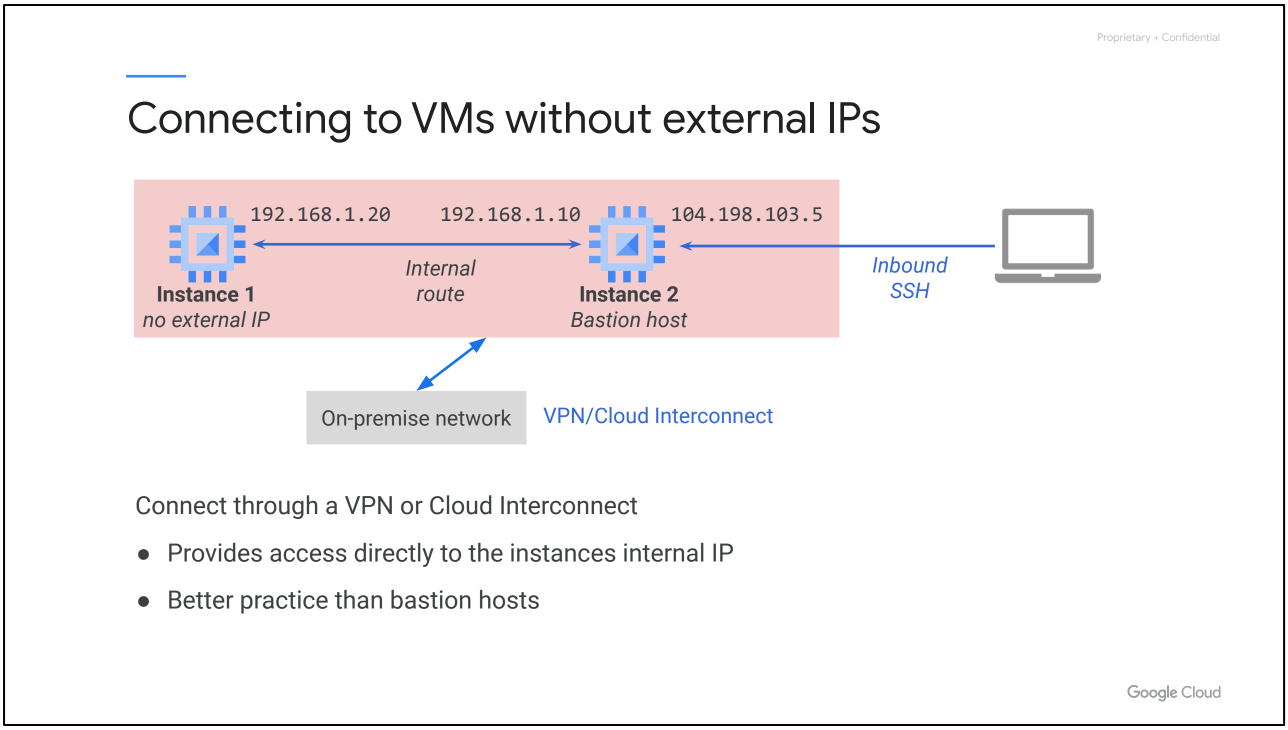
□ Block project-wide SSH keys

When checked, project-wide SSH keys cannot access this instance Learn more

However if you do not want keys to be available to all VMs in the project, you can configure individual VMs to not use project-wide keys.

When launching a VM, the “Block project-wide SSH keys” option can be selected to enable this restriction.

SSH keys can also be added to instance-specific metadata and will only be available to that instance.



## Connecting to VMs without external IPs

192.168.1.20

192.168.1.10

104.198.103.5

Internal route

Instance 1
*no external IP*

Instance 2
*Bastion host*

*Inbound SSH*

On-premise network

VPN/Cloud Interconnect

Connect through a VPN or Cloud Interconnect

- Provides access directly to the instances internal IP
- Better practice than bastion hosts

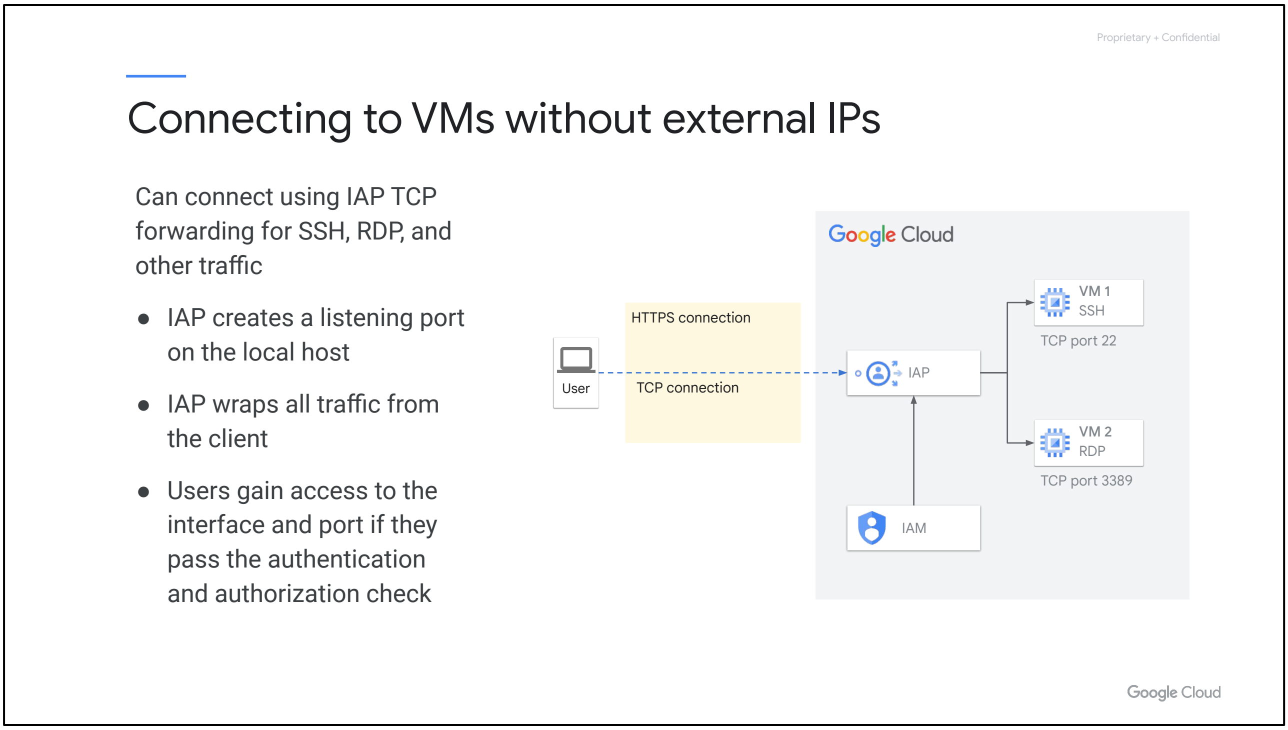
Let’s review a couple of different ways to connect to your instance if it does not have a public IP address.

One solution to this is to use a bastion host.

To implement this solution, create a second VM with a Public IP address in the same network as the instance you want to connect to. Then, connect to the bastion host and from there SSH to the private VM.

Be sure to harden the bastion host and ensure the firewall rules limit the source IPs able to connect to the Bastion, and then only allow SSH traffic to private instances from the bastion.

An even better practice would be to use a VPN or some other more secure form of connection, such as Cloud Interconnect, for ordinary activities. Only use SSH with the bastion host as the maintenance avenue of last resort.



## Connecting to VMs without external IPs

Can connect using IAP TCP forwarding for SSH, RDP, and other traffic

- IAP creates a listening port on the local host
- IAP wraps all traffic from the client
- Users gain access to the interface and port if they pass the authentication and authorization check

Google Cloud

User

HTTPS connection

TCP connection

IAP

IAM

VM 1
SSH

TCP port 22

VM 2
RDP

TCP port 3389

Another solution to this type of situation is to use IAP TCP forwarding.

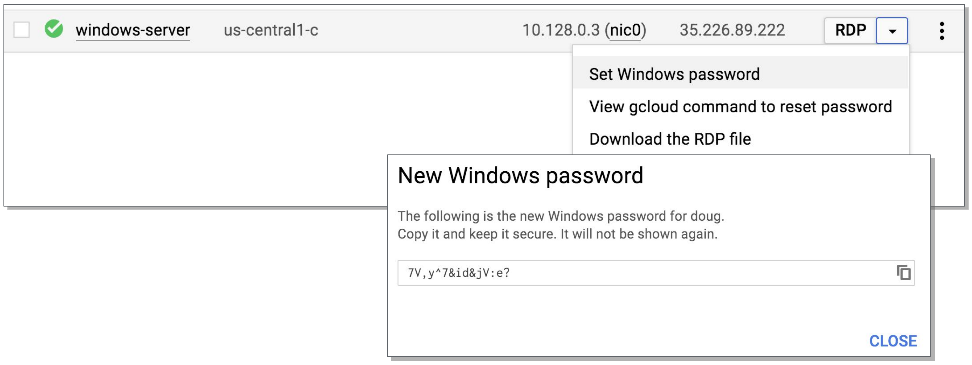
IAP TCP forwarding allows you to establish an encrypted tunnel over which you can forward SSH—as well as RDP and other traffic—to VM instances.

For general TCP traffic, IAP creates a listening port on the local host that forwards all traffic to a specified instance. IAP then wraps all traffic from the client in HTTPS.

Users gain access to the interface and port if they pass the authentication and authorization check of the target resource's Identity and Access Management (IAM) policy.

# Connecting to Windows with RDP

- Set the username and password using the Google Cloud console or gcloud.
- Can download an RDP file.



New Windows password

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

```
7V,y^7&id&jV:e?
```

For Windows VMs, connect using RDP and login in with a username and password.

The username and password can be set from the Admin Console or using gcloud.

From the console, click the down arrow next to the RDP button and select Set WIndows password.

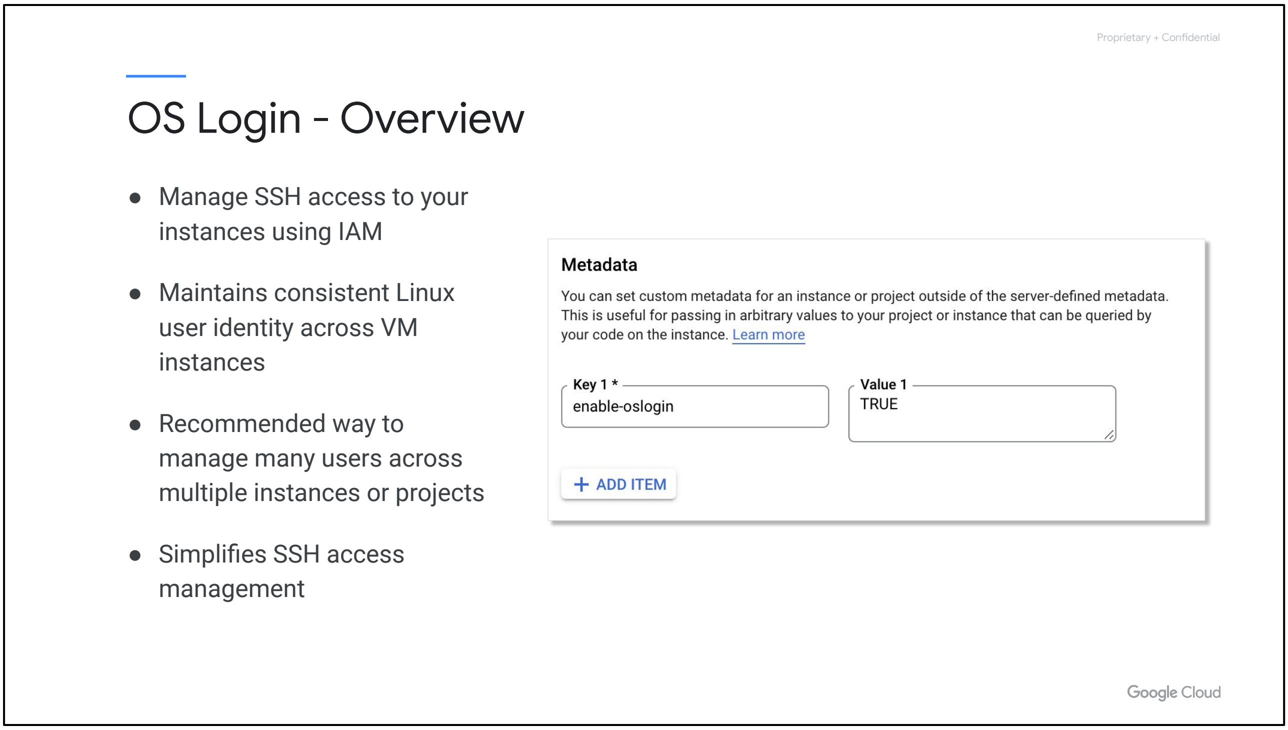
From `gcloud`, the command is:

```
gcloud compute reset-windows-password instance-name
```

and then specify the username (i.e. **`--user=`**) whose password will be reset.

To connect, simply use an RDP client and connect to the external address of the instance. You can optionally download an RDP file if you wish.

Compute Engine will automatically generate a random password for your Windows instance. Once you connect, you should change this to a custom password.



Proprietary + Confidential

## OS Login - Overview

- Manage SSH access to your instances using IAM
- Maintains consistent Linux user identity across VM instances
- Recommended way to manage many users across multiple instances or projects
- Simplifies SSH access management

**Metadata**

You can set custom metadata for an instance or project outside of the server-defined metadata. This is useful for passing in arbitrary values to your project or instance that can be queried by your code on the instance. Learn more

Key 1 *: enable-oslogin

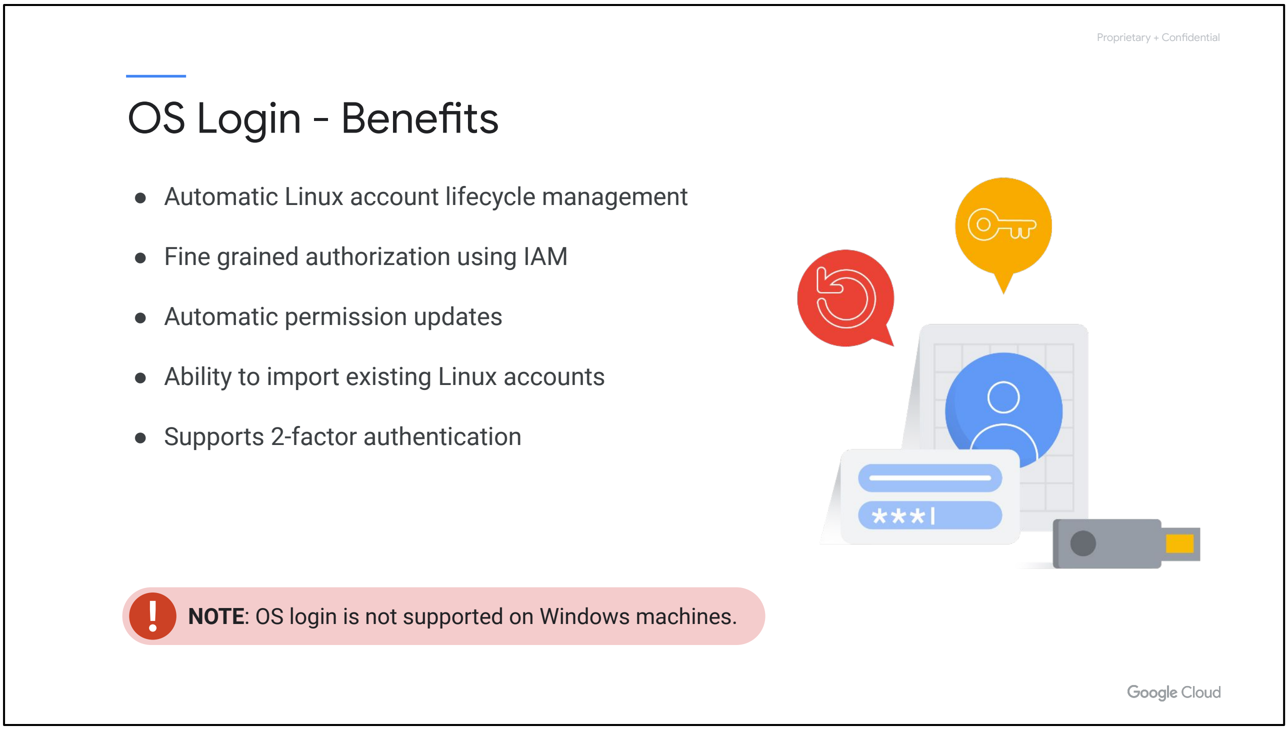
Value 1: TRUE

Google Cloud

Now that we've covered the different ways on how we can connect to our VMs, let's talk about a method you can use to manage access to instances—OS Login.

OS Login is used to manage SSH access to your instances using IAM without having to create and manage individual SSH keys. OS Login maintains a consistent Linux user identity across VM instances and is the recommended way to manage many users across multiple instances or projects.

OS Login simplifies SSH access management by linking your Linux user account to your Google identity. Administrators can easily manage access to instances at either an instance or project level by setting IAM permissions.



## OS Login - Benefits

- Automatic Linux account lifecycle management
- Fine grained authorization using IAM
- Automatic permission updates
- Ability to import existing Linux accounts
- Supports 2-factor authentication

❗ **NOTE**: OS login is not supported on Windows machines.

Google Cloud

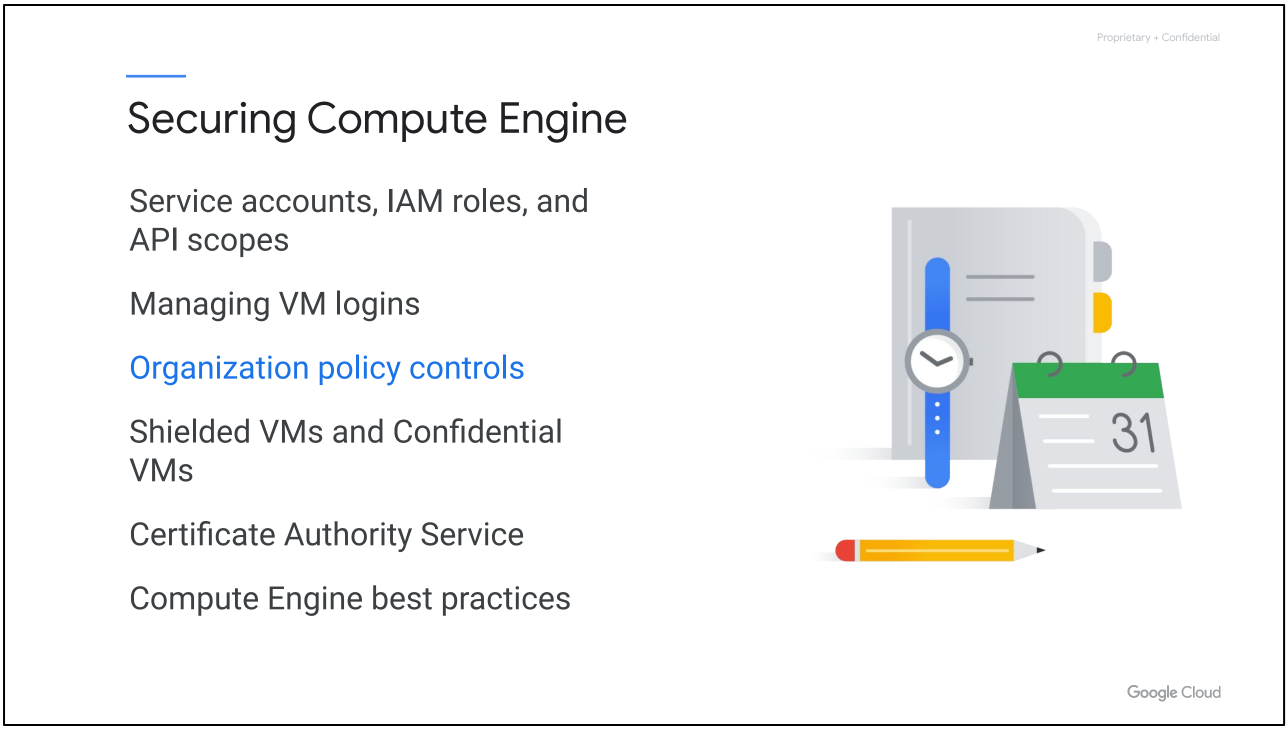
OS Login provides the following benefits:

- **Automatic Linux account lifecycle management:** You can directly tie a Linux user account to a user's Google identity so that the same Linux account information is used across all instances in the same project or organization.
- **Fine grained authorization using IAM:** Project and instance-level administrators can use IAM to grant SSH access to a user's Google identity without granting a broader set of privileges. For example, you can grant a user permissions to log into the system, but not the ability to run commands such as sudo. Google checks these permissions to determine whether a user can log into a VM instance.
- **Automatic permission updates:** With OS Login, permissions are updated automatically when an administrator changes IAM permissions. For example, if you remove IAM permissions from a Google identity, then access to VM instances is revoked. Google checks permissions for every login attempt to prevent unwanted access.
- **Ability to import existing Linux accounts:** Administrators can choose to optionally synchronize Linux account information from Active Directory (AD) and Lightweight Directory Access Protocol (LDAP) that are set up on-premises. For example, you can ensure that users have the same user ID (UID) in both your Cloud and on-premises environments.
- **Supports 2-factor authentication:** If you use OS Login to manage access to your virtual machine (VM) instances, you can add an extra layer of security by using 2-step verification.

**NOTE**: OS login is not supported on Windows machines.

For more information on OS Login, check out the documentation link in the speaker notes.

- **Link:** cloud.google.com/compute/docs/oslogin



## Securing Compute Engine

Service accounts, IAM roles, and API scopes

Managing VM logins

Organization policy controls

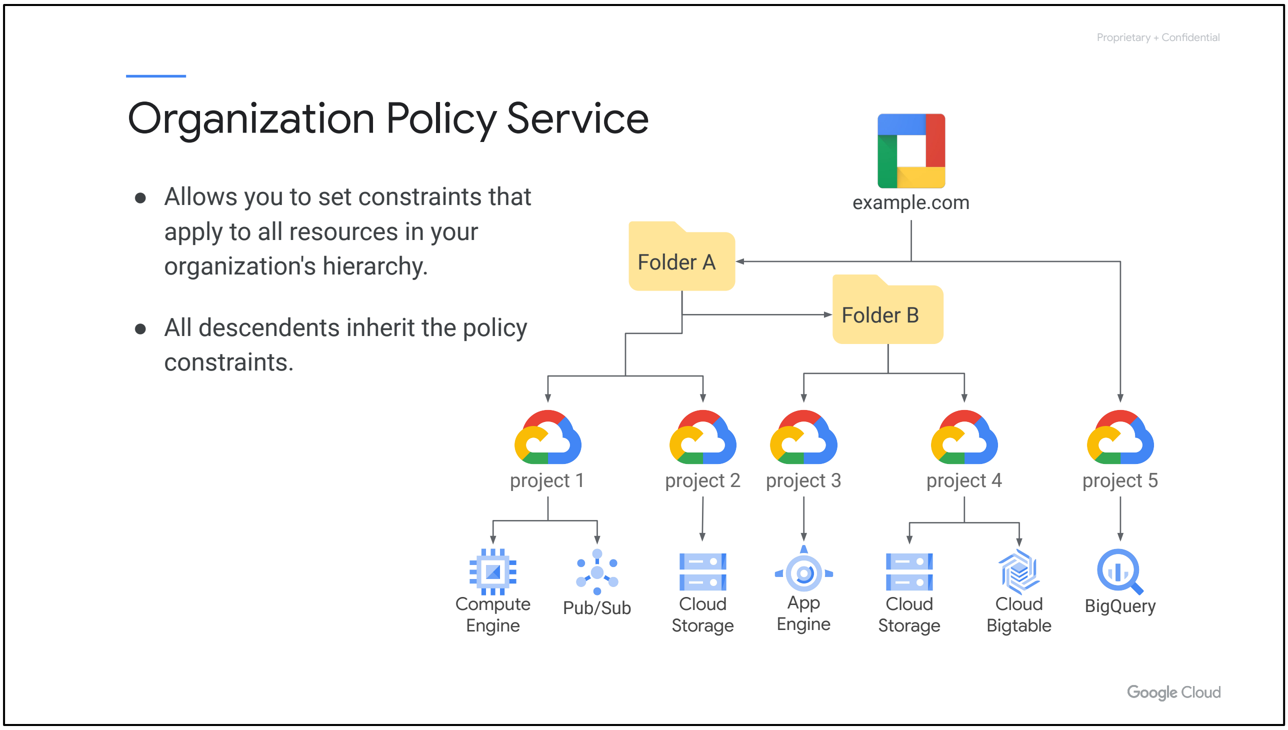
Shielded VMs and Confidential VMs

Certificate Authority Service

Compute Engine best practices

Google Cloud

We are now going to talk about Organization policy controls and how they apply to securing your Compute Engine resources and workloads.

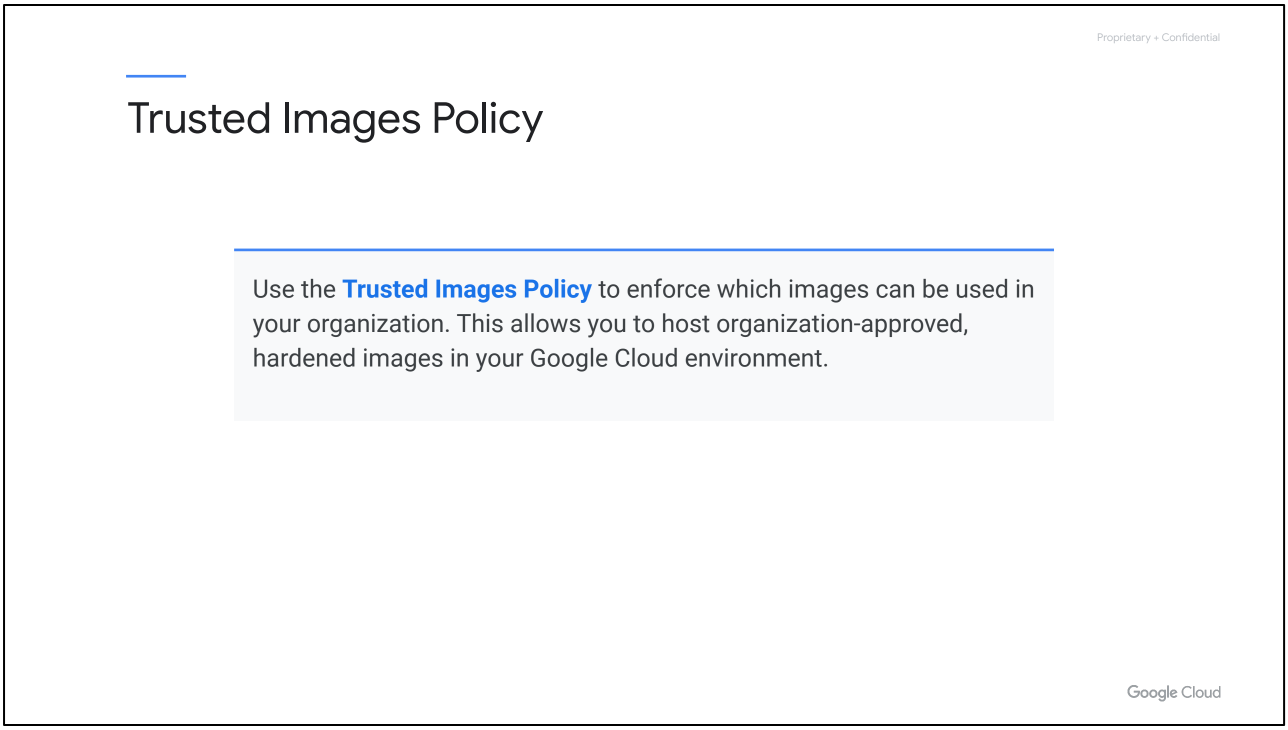


## Organization Policy Service

- Allows you to set constraints that apply to all resources in your organization's hierarchy.
- All descendents inherit the policy constraints.

The Organization Policy Service gives you both centralized and programmatic control over your organization's cloud resources.

To define an organization policy, you choose a constraint, which is a particular type of restriction against either a Google Cloud service or a group of Google Cloud services.



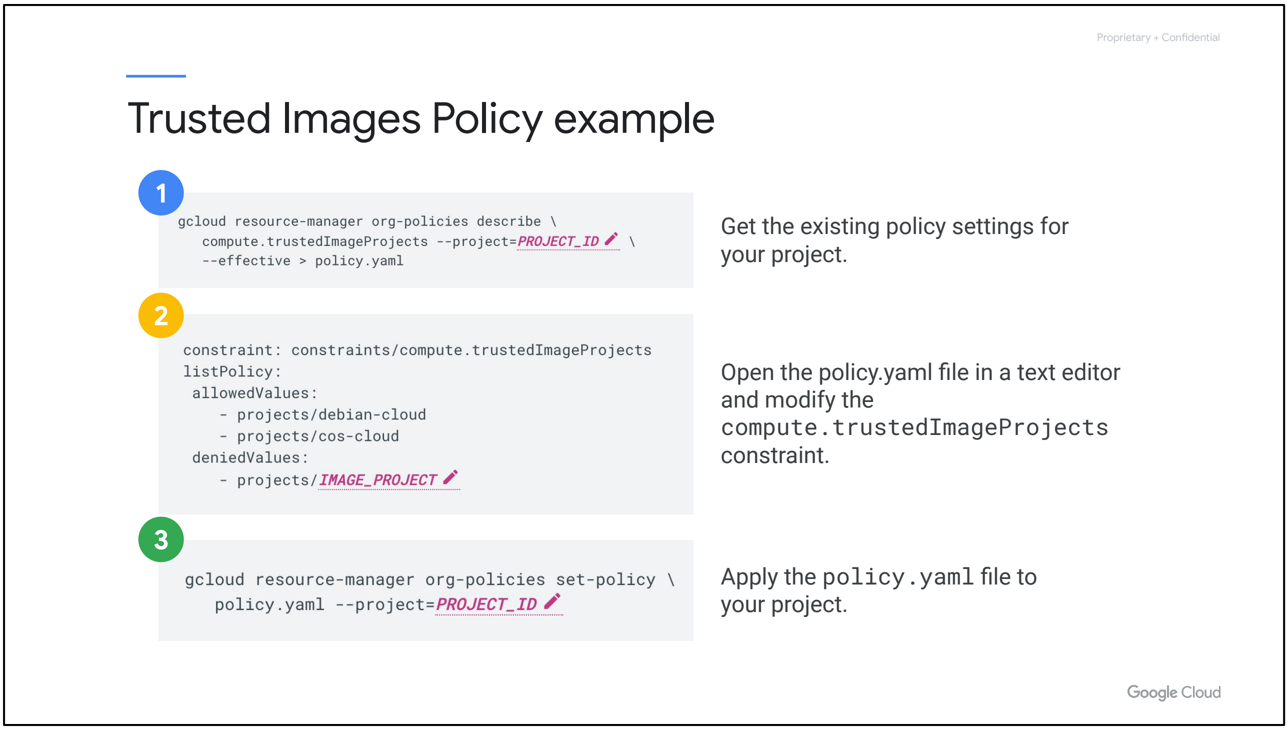
## Trusted Images Policy

Use the **Trusted Images Policy** to enforce which images can be used in your organization. This allows you to host organization-approved, hardened images in your Google Cloud environment.

The compute.trustedImageProjects constraint has an interesting use case.

By default, project users can create persistent disks or copy images using either public images and any other images that project members can access through IAM roles. However, you may want to restrict your projects to only use images that contain approved software to create boot disks that meets your policy or security requirements.

The Trusted Images Policy can be used to enforce which images can be used in your organization. This allows you to host organization-approved, hardened images in your Google Cloud environment.



Proprietary + Confidential

# Trusted Images Policy example

1

```
gcloud resource-manager org-policies describe \
   compute.trustedImageProjects --project=PROJECT_ID \
   --effective > policy.yaml
```

Get the existing policy settings for your project.

2

```
constraint: constraints/compute.trustedImageProjects
listPolicy:
 allowedValues:
    - projects/debian-cloud
    - projects/cos-cloud
 deniedValues:
    - projects/IMAGE_PROJECT
```

Open the policy.yaml file in a text editor and modify the `compute.trustedImageProjects` constraint.

3

```
gcloud resource-manager org-policies set-policy \
   policy.yaml --project=PROJECT_ID
```

Apply the `policy.yaml` file to your project.

Google Cloud

Let's see how you would apply the `compute.trustedImageProjects` constraint to a project.

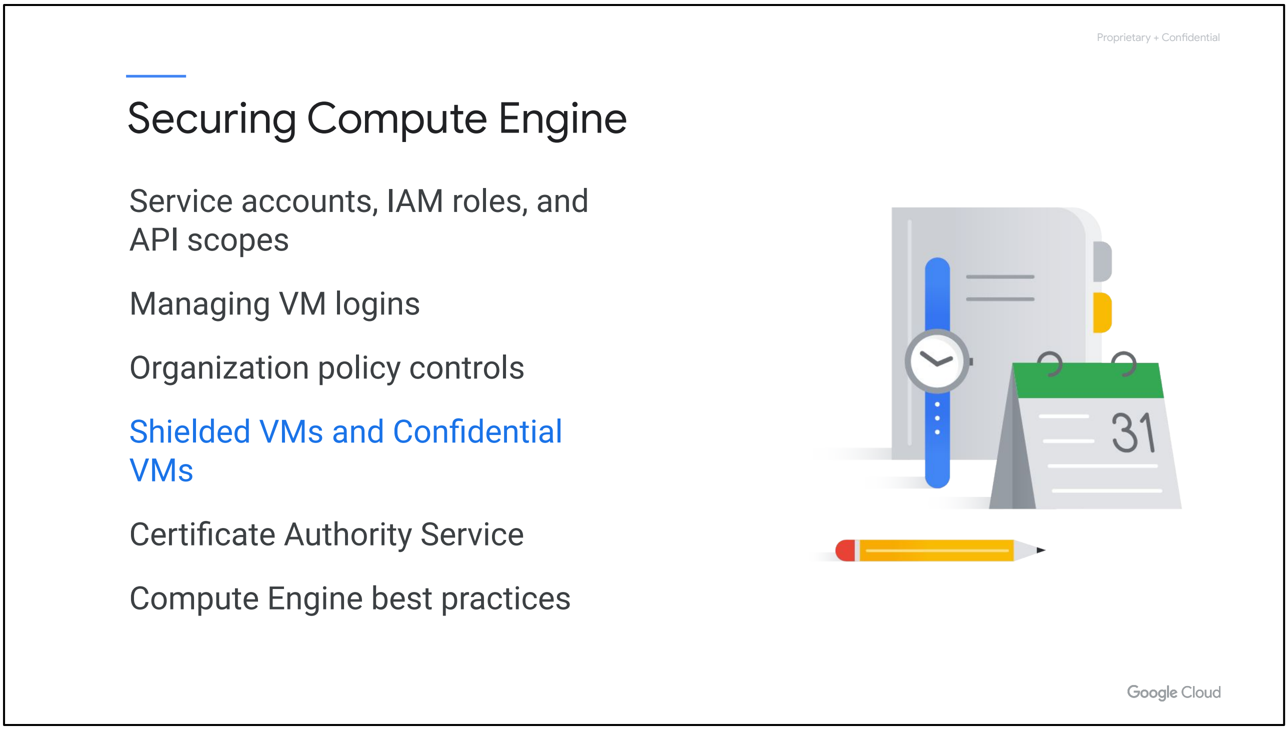
First, get the existing policy settings for your project by using the resource-manager org-policies describe command.

Second, open the policy.yaml file in a text editor and modify the `compute.trustedImageProjects` constraint.

Add the restrictions that you need and remove the restrictions that you no longer require. When you have finished editing the file, save your changes.

Third, apply the policy.yaml file to your project. If your organization or folder has existing constraints, those constraints might conflict with project-level constraints that you set.

To apply the constraint, use the resource-manager org-policies set-policy command.



## Securing Compute Engine

Service accounts, IAM roles, and API scopes

Managing VM logins

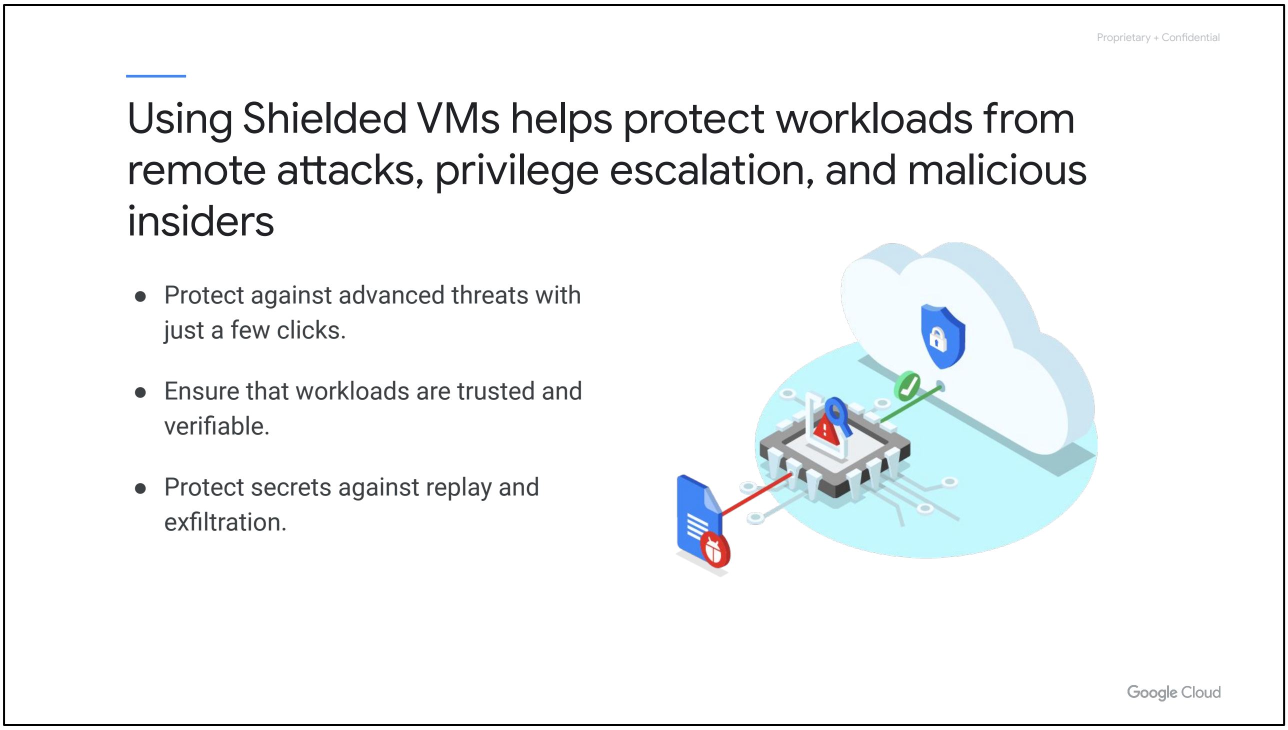
Organization policy controls

Shielded VMs and Confidential VMs

Certificate Authority Service

Compute Engine best practices

Now let's discuss Shielded VMs and Confidential VMs.



Proprietary + Confidential

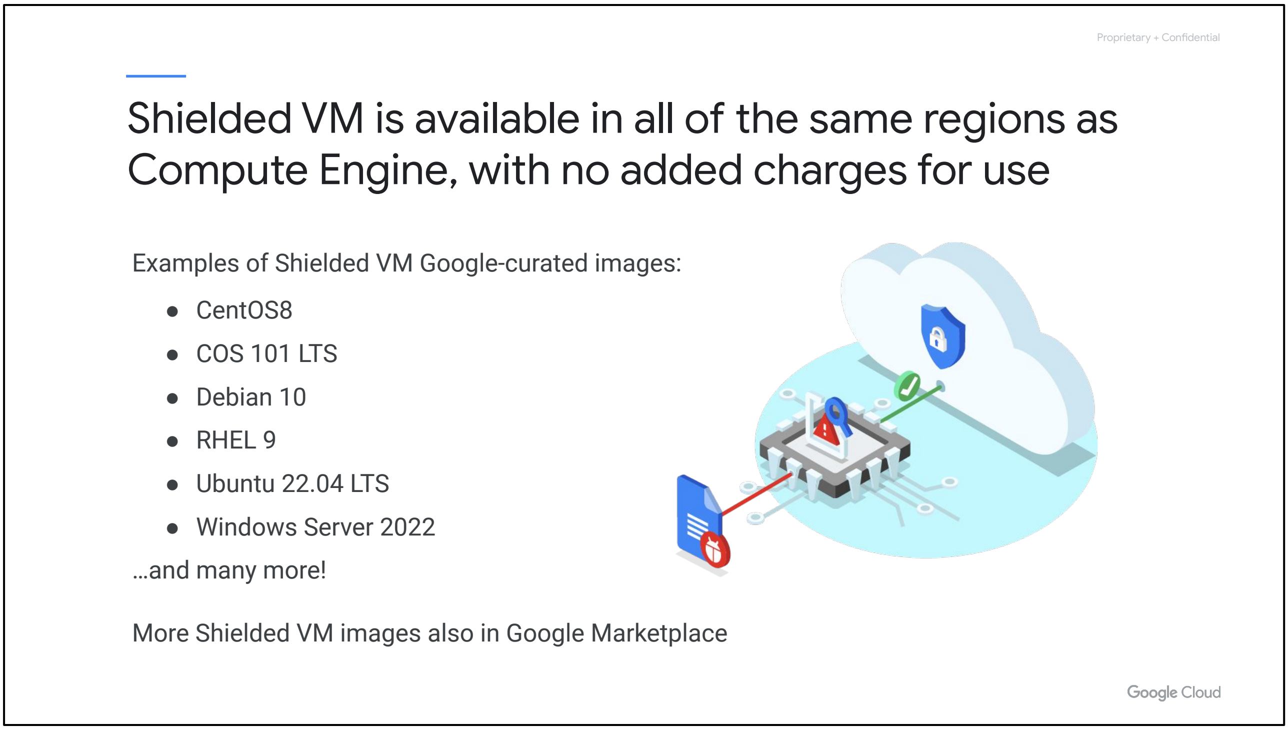
# Using Shielded VMs helps protect workloads from remote attacks, privilege escalation, and malicious insiders

- Protect against advanced threats with just a few clicks.
- Ensure that workloads are trusted and verifiable.
- Protect secrets against replay and exfiltration.

Google Cloud

Protecting your hardware and firmware and host and guest operating systems is an important part of securing your workloads and data from malicious use and attacks. Unfortunately, some types of malware attacks can remain undetected on your virtual machines for long periods of time.

Shielded VM offers **verifiable integrity** of your Compute Engine VM instances, so you can be confident that your instances haven't been compromised by boot-level or kernel-level malware or rootkits, or that your secrets are exposed and used by others.



## Shielded VM is available in all of the same regions as Compute Engine, with no added charges for use

Examples of Shielded VM Google-curated images:

- CentOS8
- COS 101 LTS
- Debian 10
- RHEL 9
- Ubuntu 22.04 LTS
- Windows Server 2022

…and many more!

More Shielded VM images also in Google Marketplace

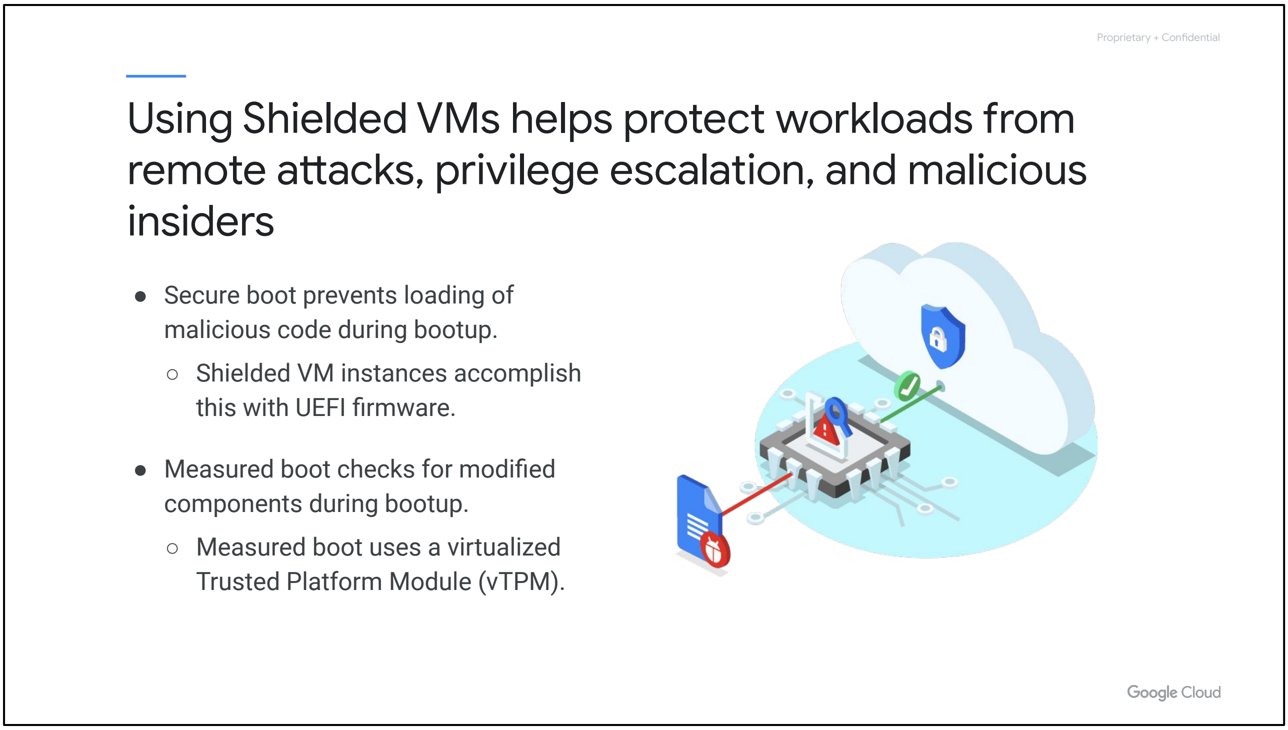
Google Cloud

When creating a Shielded VM, there are a wide range of image options. This slide shows a sample of currently available Google-curated images. You can find even more shielded VM images the Google Cloud Marketplace.

In addition, if your organization relies on custom images, you can now transform an existing VM into a Shielded VM that runs on Google Cloud.

To see a full list of images that support Shielded VMs, see the link in the speaker notes.

**Link:** https://cloud.google.com/compute/docs/images/os-details#security-features



Proprietary + Confidential

## Using Shielded VMs helps protect workloads from remote attacks, privilege escalation, and malicious insiders

- Secure boot prevents loading of malicious code during bootup.
  - Shielded VM instances accomplish this with UEFI firmware.
- Measured boot checks for modified components during bootup.
  - Measured boot uses a virtualized Trusted Platform Module (vTPM).

Google Cloud

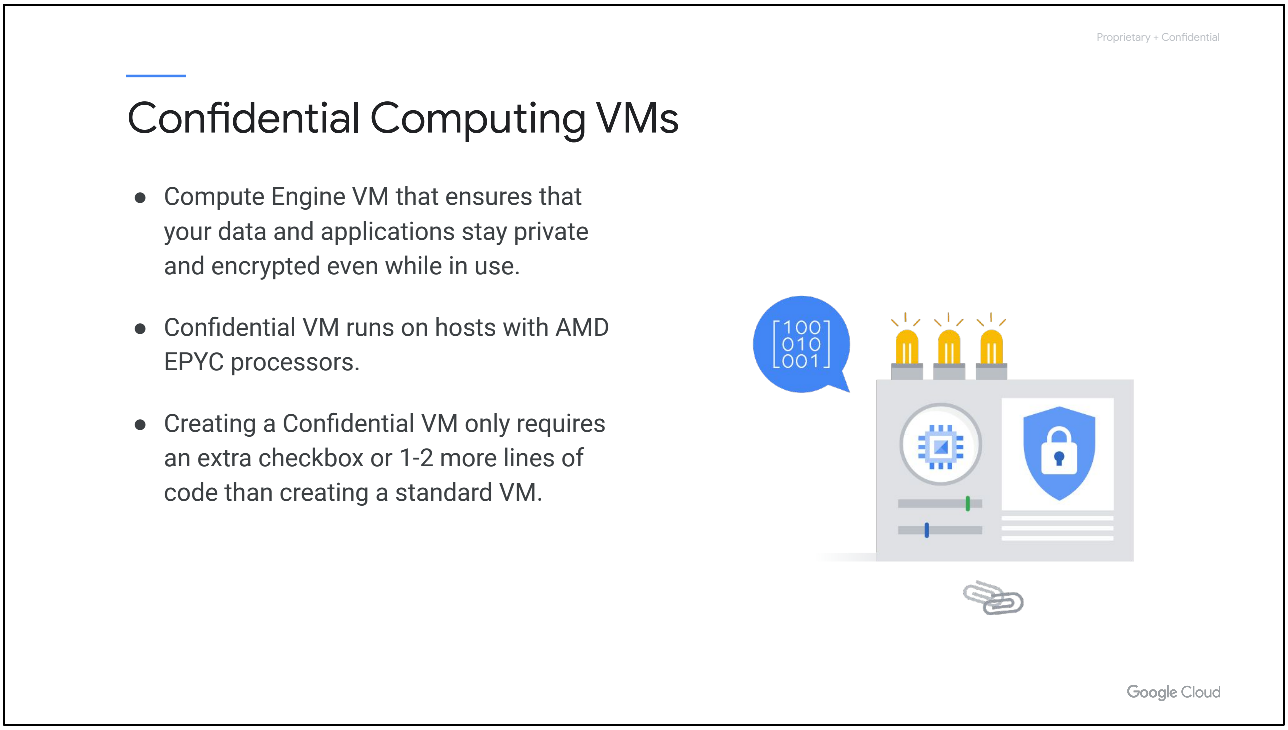
Each time your VM starts up, secure boot makes certain that the software it is loading is authentic and unmodified by verifying that the firmware has been digitally signed with Google’s Certificate Authority Service (CAS).

Shielded VM instances use Unified Extensible Firmware Interface (UEFI) firmware, which securely manages the certificates that contain the keys used by the software manufacturers to sign the system firmware, the system boot loader, and any binaries loaded. UEFI firmware verifies the digital signature of each boot component in turn against its secure store of approved keys, and if that component isn't properly signed (or isn't signed at all), it isn't allowed to run. This verification ensures that the instance's firmware is unmodified and establishes the “root of trust” for Secure Boot.

Measured boot creates a hash of each component as it loads, concatenates that hash with other components that have already been loaded, and then rehashes it. This allows measured boot to record the number of components loaded on boot-up and their sequence.

The first time your Shielded VM is booted, this initial hash is securely stored and used as the baseline for verification of that VM during subsequent boots. This is called “integrity monitoring,” and it helps ensure that your VM’s boot components and boot sequence have not been altered.

Shielded VMs use a virtual Trusted Platform Module, which is the “virtualized” version of a specialized computer chip you can use to protect objects, like keys and certificates, that are used to provide authenticated access to your system. This vTPM allows Measured Boot to perform the measurements needed to create a known good boot baseline, called the integrity policy baseline, upon the first bootup of your Shielded VM.



Proprietary + Confidential

## Confidential Computing VMs

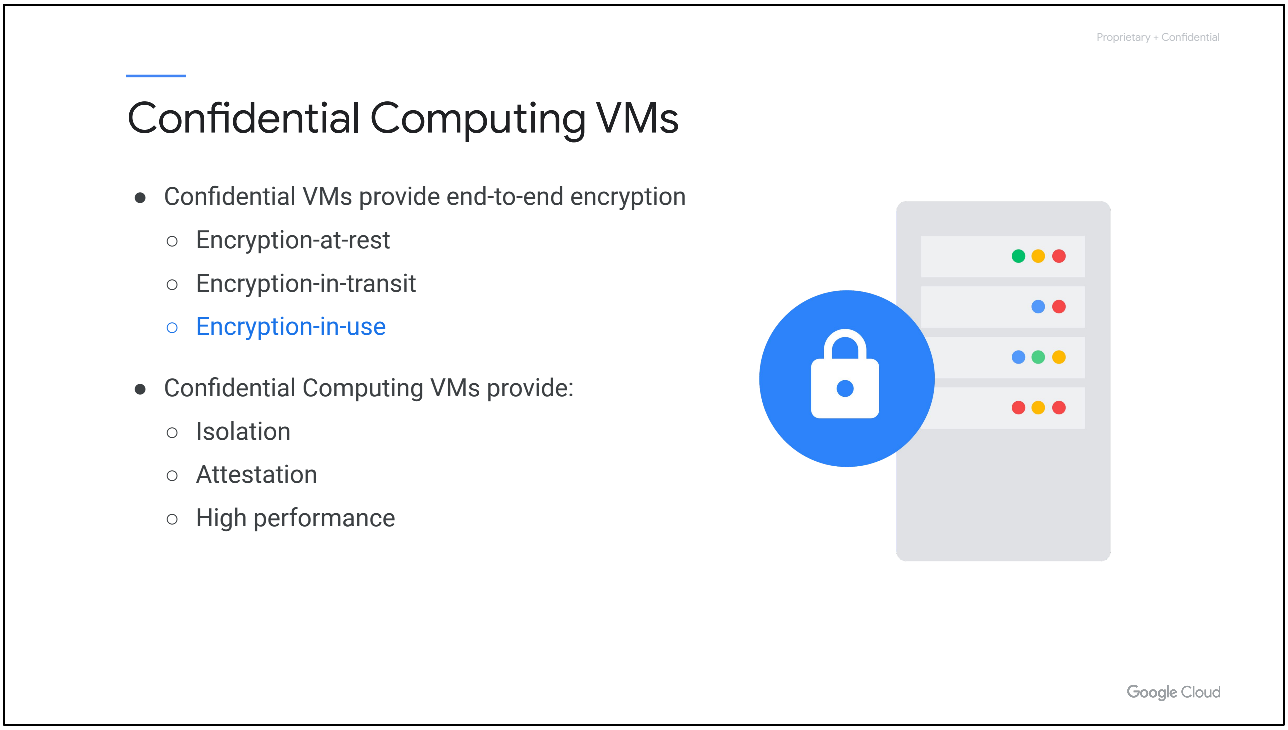
- Compute Engine VM that ensures that your data and applications stay private and encrypted even while in use.
- Confidential VM runs on hosts with AMD EPYC processors.
- Creating a Confidential VM only requires an extra checkbox or 1-2 more lines of code than creating a standard VM.

Google Cloud

A Confidential VM is a type of Compute Engine VM that ensures that your data and applications stay private and encrypted even while in use. You can use a Confidential VM as part of your security strategy so you do not expose sensitive data or workloads during processing.

Confidential VM runs on hosts with AMD EPYC processors which feature AMD Secure Encrypted Virtualization (SEV). Incorporating SEV into Confidential VM provides the following benefits and features.

You can enable Confidential Computing whenever you create a new VM. Creating a Confidential VM only requires an extra checkbox or 1-2 more lines of code than creating a standard VM. You can continue using the other tools and workflows you're already familiar with. Adding Confidential Computing requires no changes to your existing applications.



Proprietary + Confidential

# Confidential Computing VMs

- Confidential VMs provide end-to-end encryption
  - Encryption-at-rest
  - Encryption-in-transit
  - Encryption-in-use
- Confidential Computing VMs provide:
  - Isolation
  - Attestation
  - High performance

Google Cloud

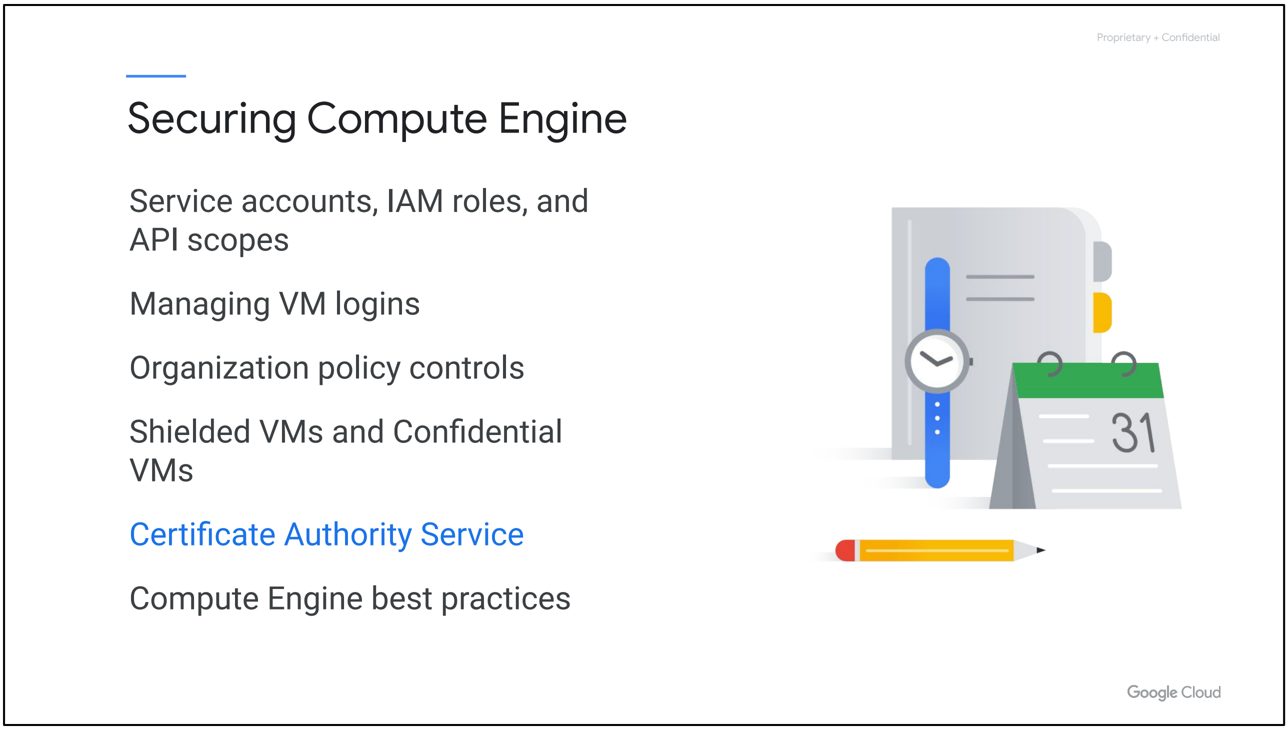
Confidential VMs provide end-to-end encryption. End-to-end encryption is comprised of three states.

- Encryption-at-rest protects your data while it is being stored.
- Encryption-in-transit protects your data when it is moving between two points.
- Encryption-in-use protects your data while it is being processed.

Confidential Computing VMs give you the the last piece of end-to-end encryption: encryption-in-use.

Confidential Computing VMs provide:

- **Isolation:** Encryption keys are generated by the AMD Secure Processor (SP) during VM creation and reside solely within the AMD System-On-Chip (SOC). These keys are not even accessible by Google, offering improved isolation.
- **Attestation:** Confidential VM uses Virtual Trusted Platform Module (vTPM) attestation. Every time an AMD SEV-based Confidential VM boots, a launch attestation report event is generated.
- **High performance:** AMD SEV offers high performance for demanding computational tasks. Enabling Confidential VM has little or no impact on most workloads, with only a 0-6% degradation in performance.



# Securing Compute Engine

Service accounts, IAM roles, and API scopes

Managing VM logins

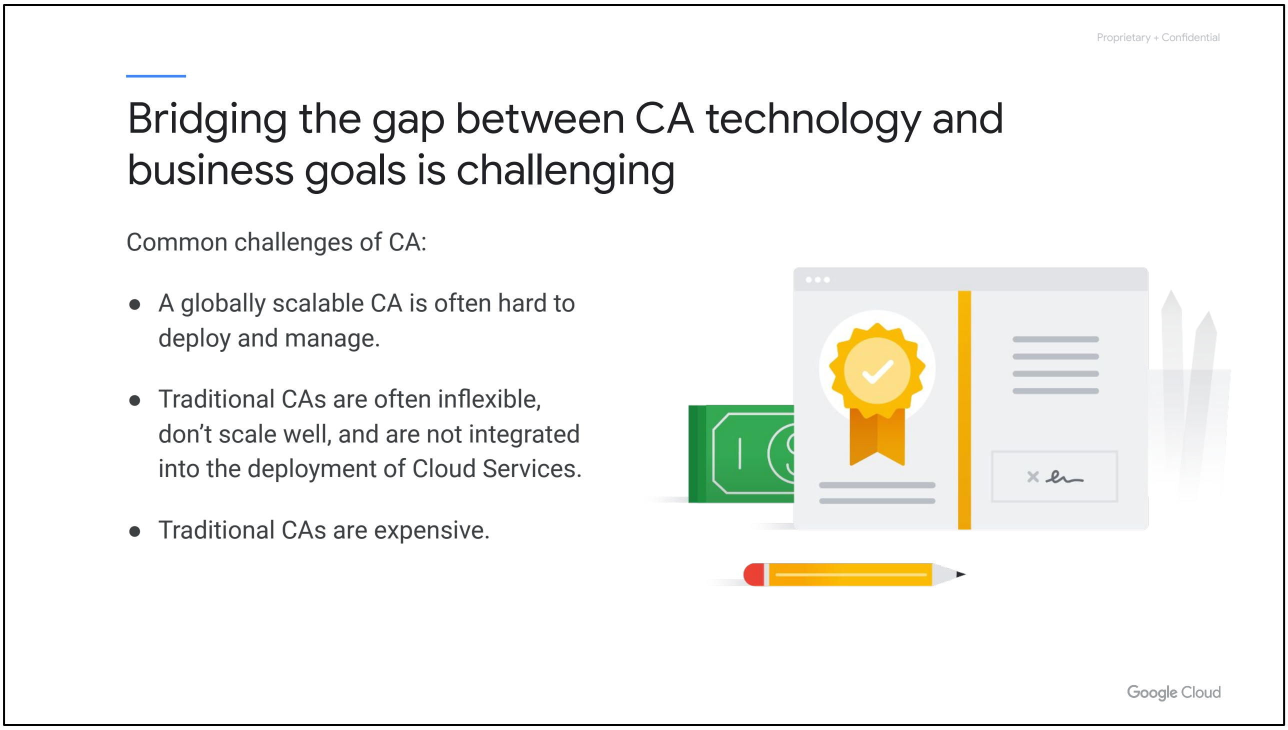
Organization policy controls

Shielded VMs and Confidential VMs

Certificate Authority Service

Compute Engine best practices

Now let’s discuss Certificate Authority Service.



## Bridging the gap between CA technology and business goals is challenging

Common challenges of CA:

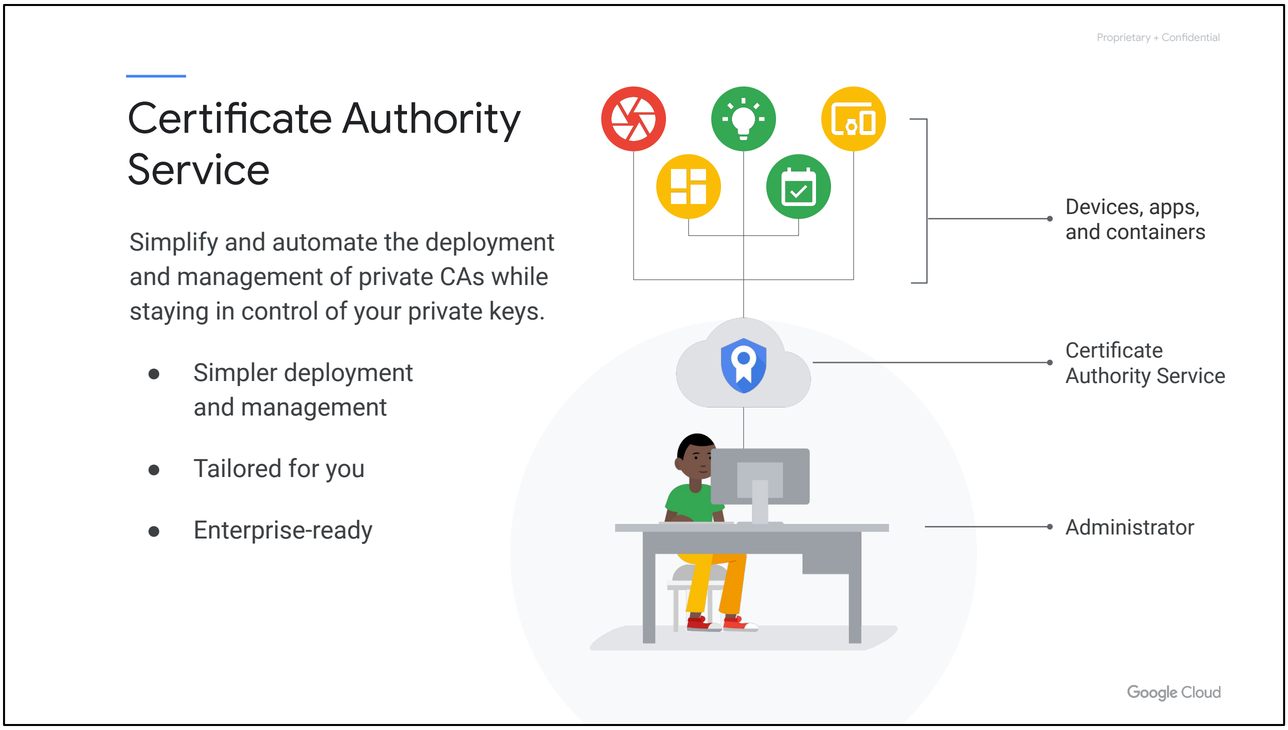
- A globally scalable CA is often hard to deploy and manage.
- Traditional CAs are often inflexible, don’t scale well, and are not integrated into the deployment of Cloud Services.
- Traditional CAs are expensive.

Enterprise organizations are frequently surprised at the extent of extra work that they must do having bought CA technology to bridge the gap between technology and their organizational and business goals.

A globally scalable CA is often hard to deploy and manage because it requires deep expertise that many organization do not have.

Traditional CAs are also not well suited for new cases, such as microservices and DevOps because they are often inflexible, don’t scale well, and are not integrated into the deployment and access control infrastructures commonly used within the Cloud Services.

Lastly, traditional CAs are expensive with high Capex and infrastructure, hardware and software licensing, in addition to expensive ongoing costs.



Proprietary + Confidential

## Certificate Authority Service

Simplify and automate the deployment and management of private CAs while staying in control of your private keys.

- Simpler deployment and management
- Tailored for you
- Enterprise-ready

Devices, apps, and containers

Certificate Authority Service

Administrator

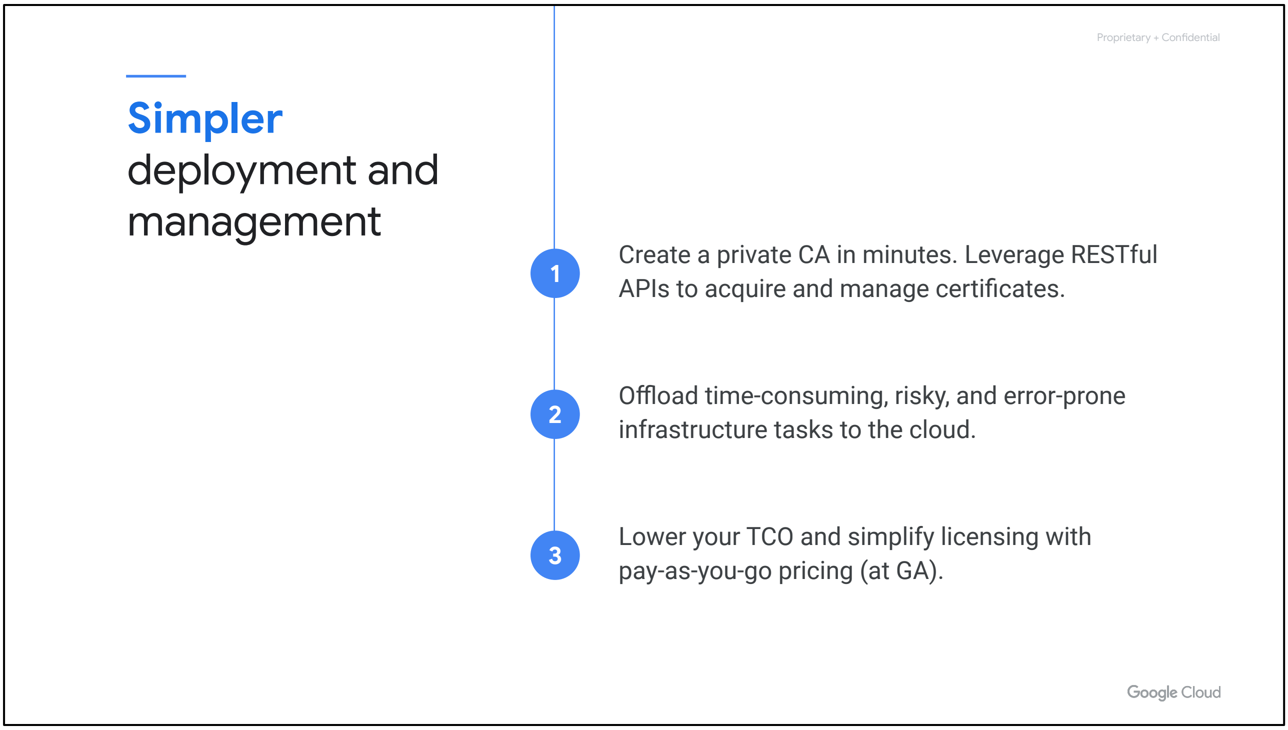
Google Cloud

With Google Cloud’s Certificate Authority Service, you can solve these challenges.

Certificate Authority Service is a highly-available, scalable Google Cloud service that enables IT and security teams to simplify and automate the deployment, management, and security of private certificate authorities (CA) while staying in control of their private keys.

Certificate Authority Service provides you with:

- **Simpler deployment and management.** Simplify the deployment, management, and security of your enterprise infrastructure with a cloud service that automates time-consuming, risky, and error-prone infrastructure tasks, giving you more time to focus on higher-value projects.
- **Tailored for you**. Customize Certificate Authority Service to your needs by configuring custom CAs and certificates, enforcing granular access controls, automating common tasks with robust APIs, and integrating with your existing systems.
- **Enterprise-ready.** Have peace of mind knowing that the service is highly available, scalable, backed by an SLA, auditable, and ready to help you achieve compliance with advanced hardware and software security controls.



Proprietary + Confidential

## Simpler deployment and management

1 Create a private CA in minutes. Leverage RESTful APIs to acquire and manage certificates.

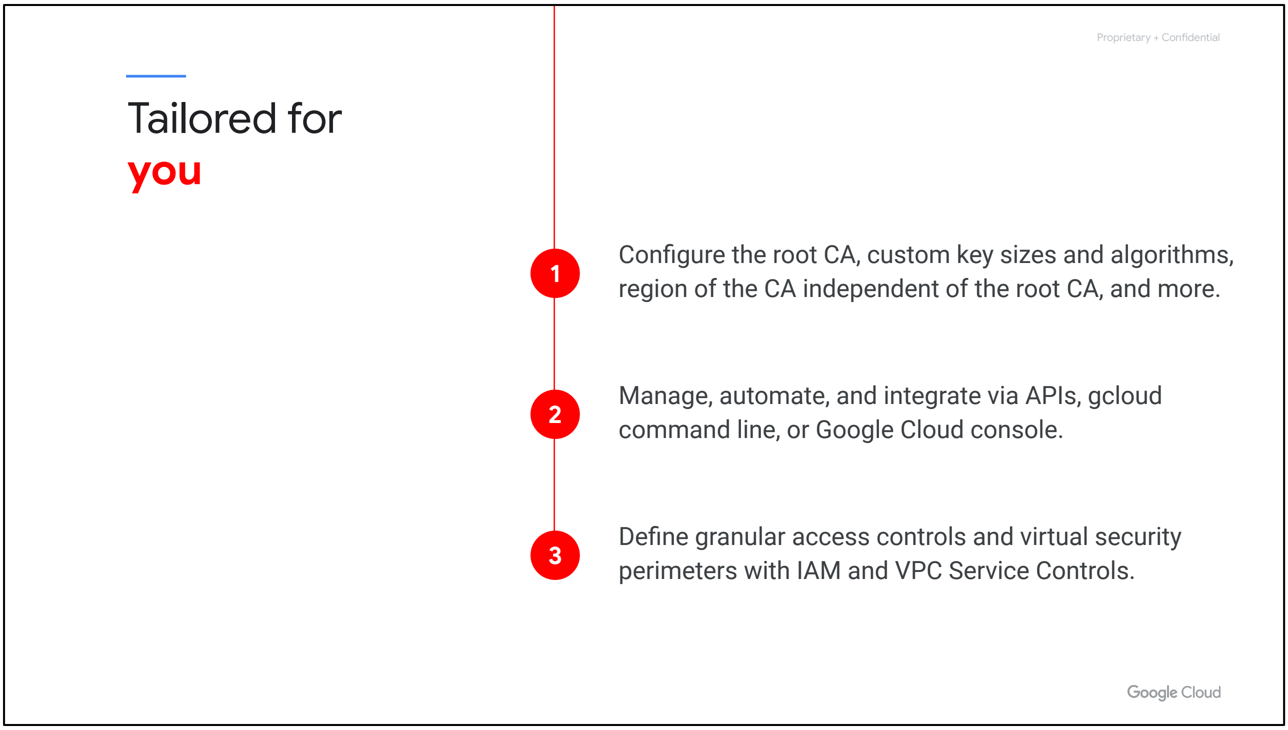
2 Offload time-consuming, risky, and error-prone infrastructure tasks to the cloud.

3 Lower your TCO and simplify licensing with pay-as-you-go pricing (at GA).

Google Cloud

Certificate Authority Service allows you to:

- **Deploy in minutes.** Create a private CA in minutes versus the days and weeks that it takes to deploy a traditional CA. Leverage descriptive RESTful APIs to acquire and manage certificates without being a PKI expert.
- **Focus on higher-value tasks.** Offload time-consuming tasks like hardware provisioning, infrastructure security, software deployment, high-availability configuration, disaster recovery, backups, and more to the cloud.
- **Pay-as-you-go (at GA)**. Lower your total cost of ownership and simplify licensing with pay-as-you-go pricing and zero capital expenditures. Pay only for what you use (when the service becomes generally available [GA]).



Proprietary + Confidential

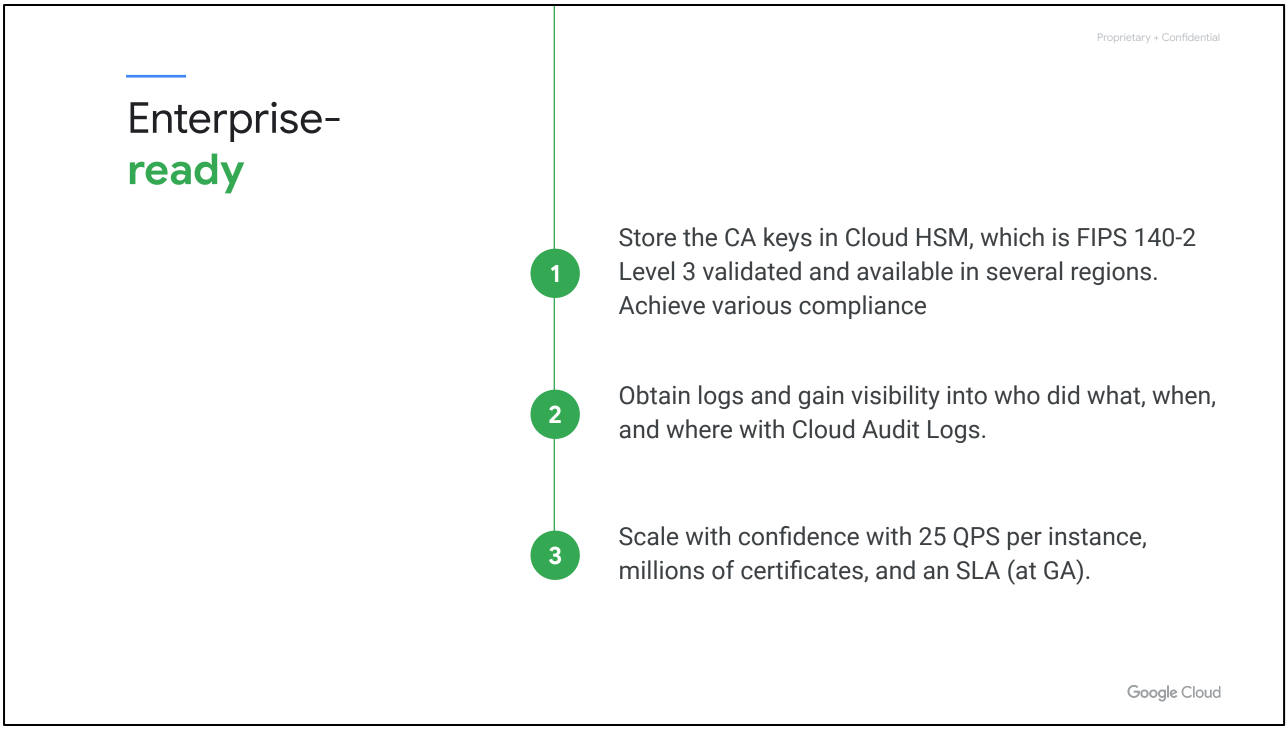
## Tailored for you

1. Configure the root CA, custom key sizes and algorithms, region of the CA independent of the root CA, and more.
2. Manage, automate, and integrate via APIs, gcloud command line, or Google Cloud console.
3. Define granular access controls and virtual security perimeters with IAM and VPC Service Controls.

Google Cloud

Certificate Authority Service allows you to:

- **Customize to your needs.** Scale from simple use cases to advanced by configuring the root CA (e.g. existing on-premises or cloud), custom key sizes and algorithms, region of the CA independent of the root CA, and more.
- **Manage it your way.** Manage, automate, and integrate private CAs and certificates in a way that is most convenient for you: via APIs, gcloud command line, or the Google Cloud console.
- **Enforce granular access.** Define granular and context-aware access controls and virtual security perimeters with IAM and VPC Service Controls (at GA).



Proprietary + Confidential

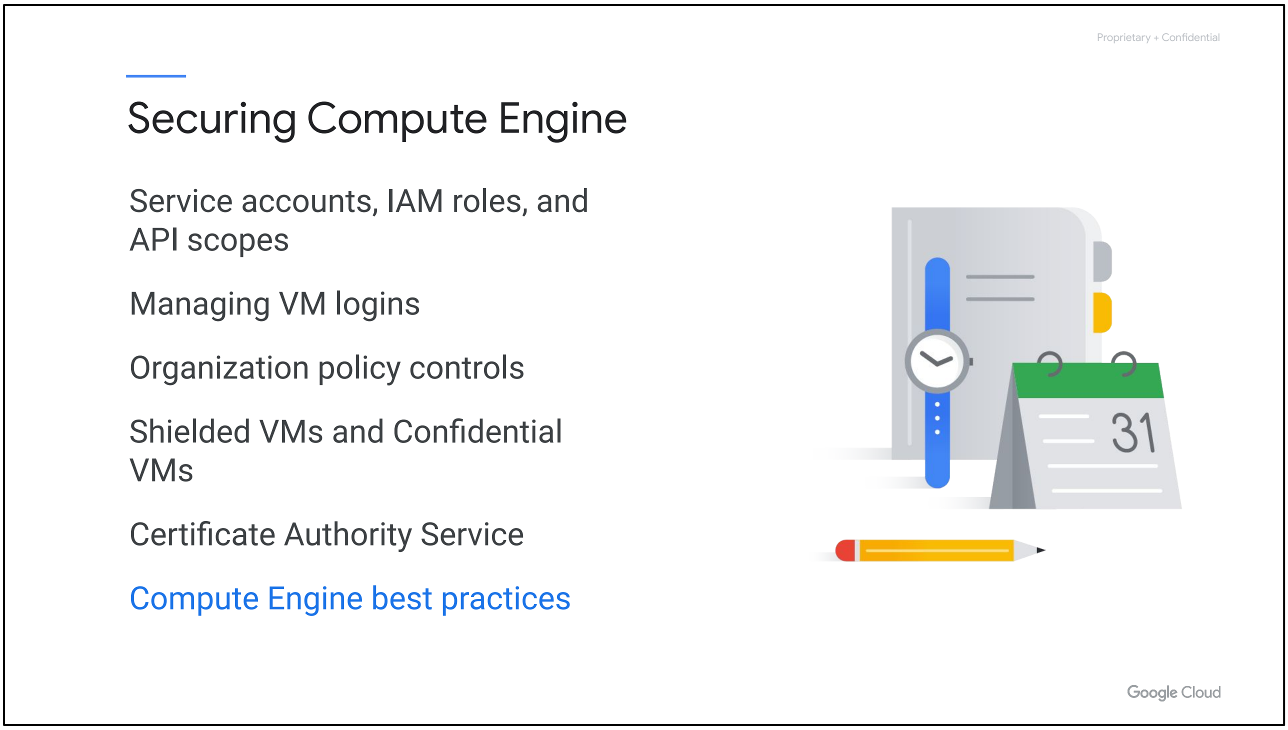
## Enterprise-**ready**

1. Store the CA keys in Cloud HSM, which is FIPS 140-2 Level 3 validated and available in several regions. Achieve various compliance
2. Obtain logs and gain visibility into who did what, when, and where with Cloud Audit Logs.
3. Scale with confidence with 25 QPS per instance, millions of certificates, and an SLA (at GA).

Google Cloud

Certificate Authority Service gives you the ability to:

- **Protect the keys with a HSM.** Store the CA keys in Cloud HSM, which is FIPS 140-2 Level 3 validated and available in several regions across the Americas, Europe, and Asia Pacific.
- **Audit user activity.** Obtain tamper-proof logs and gain visibility into who did what, when, and where with Cloud Audit Logs.
- **Scale with confidence.** Scale with confidence knowing that the service supports 25 queries per second (QPS) per instance, can issue millions of certificates, and comes with an enterprise-grade SLA (at GA).



## Securing Compute Engine

Service accounts, IAM roles, and API scopes

Managing VM logins

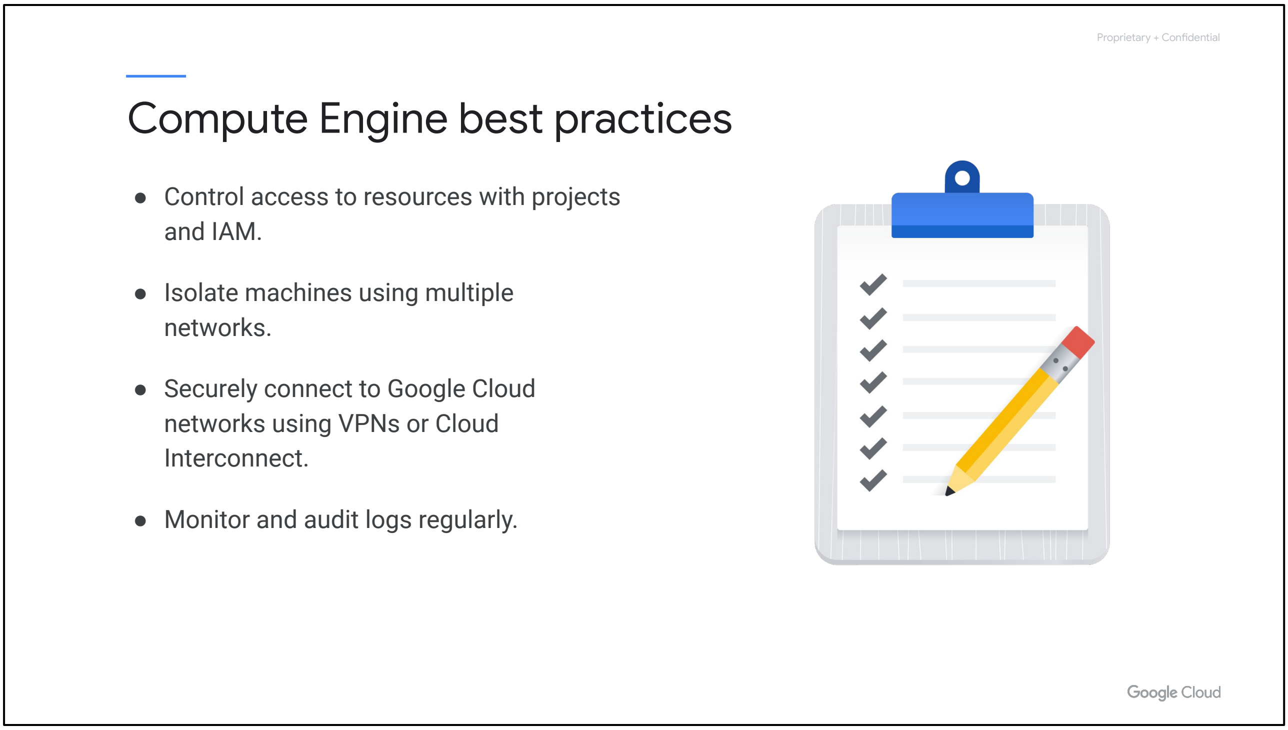
Organization policy controls

Shielded VMs and Confidential VMs

Certificate Authority Service

Compute Engine best practices

Now let’s discuss Compute Engine best practices.



## Compute Engine best practices

- Control access to resources with projects and IAM.
- Isolate machines using multiple networks.
- Securely connect to Google Cloud networks using VPNs or Cloud Interconnect.
- Monitor and audit logs regularly.

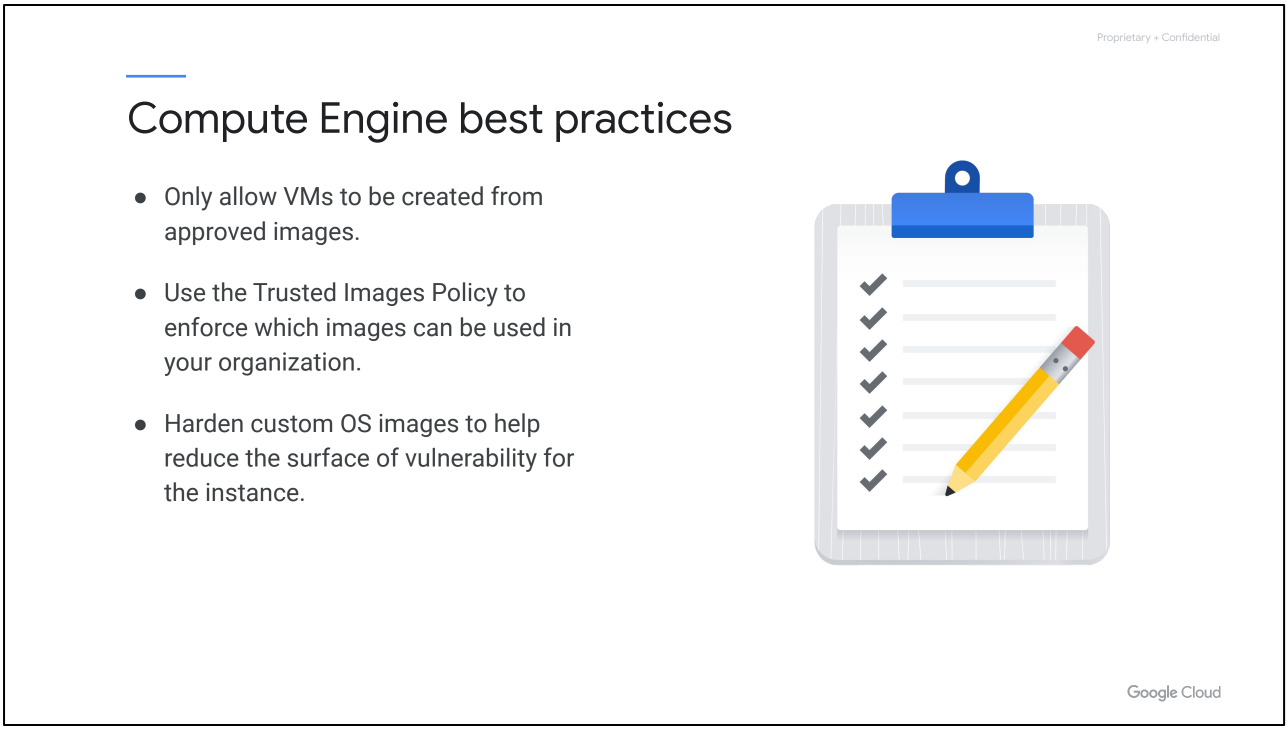
Google Cloud

First of all, always ensure the proper permissions are given to control access to resources. Projects form the basis for creating, enabling, and using all Google Cloud services, including managing resource permissions. Build for success, and utilize projects and IAM roles to control access.

Host Compute Engine resources on the same VPC network where they require network based communication. If the resources aren’t related and don’t require network communication among themselves, consider hosting them on different VPC networks.

Secure connections to public cloud providers are a concern for all organizations. You can securely extend your data center network into projects with Cloud Interconnect or Cloud VPN.

Use Cloud Audit Logging to generate logs for API operations performed in Google Compute Engine. Audit logs help you determine “who did what”, “where”, and “when”. Specifically, audit logs track how Compute Engine resources are modified and accessed within projects for auditing purposes.



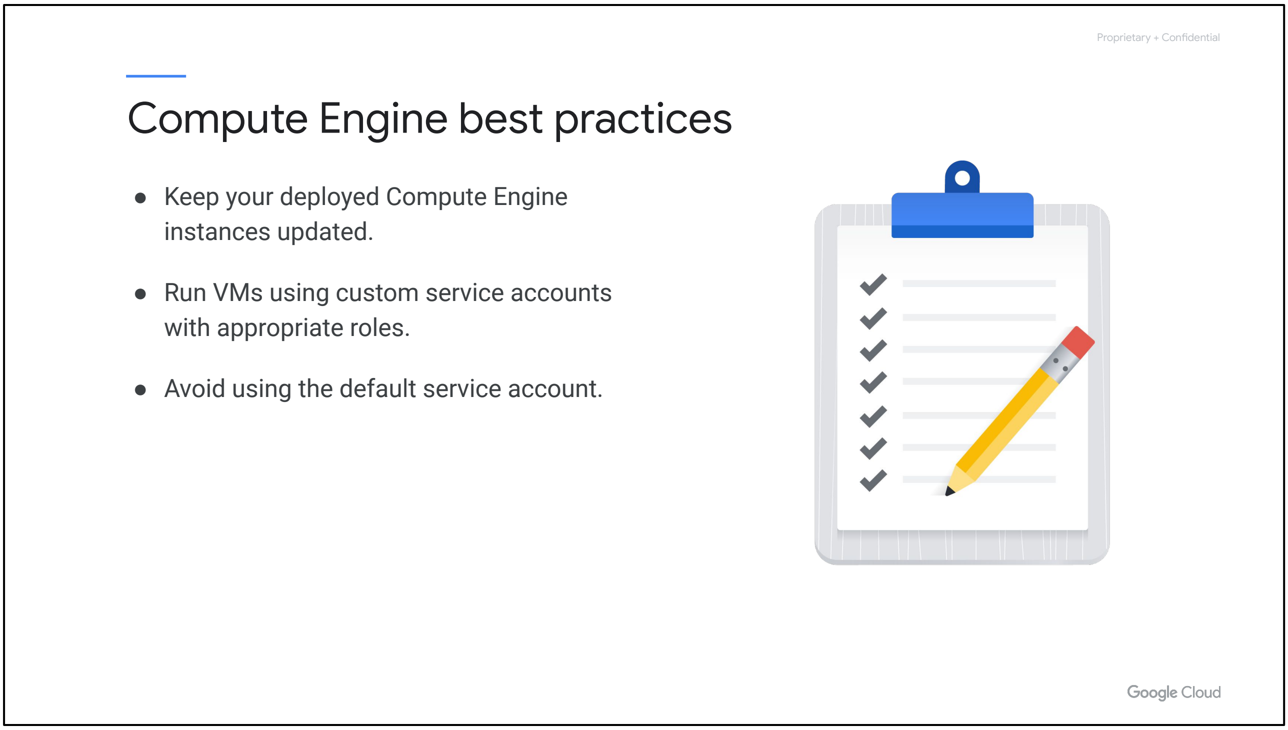
## Compute Engine best practices

- Only allow VMs to be created from approved images.
- Use the Trusted Images Policy to enforce which images can be used in your organization.
- Harden custom OS images to help reduce the surface of vulnerability for the instance.

By default, users in a project can create persistent disks or copy images using any of the public images and any images that your project members can access through IAM roles. You may want to restrict your project members so that they can create boot disks only from images that contain approved software that meets your policy or security requirements. You can define an organization policy that only allows Compute Engine VMs to be created from approved images.

This can be done by using the trusted Images Policy to enforce which images can be used in your organization. This allows you to host organization-approved, hardened images in your Google Cloud environment.

Hardening a custom OS image will help reduce the attack surface for the instance. Making hardened images available in your organization can help reduce your organization’s overall risk profile. However, if you create a custom image, formulate a plan for how to maintain the image with security patches and other updates.



## Compute Engine best practices

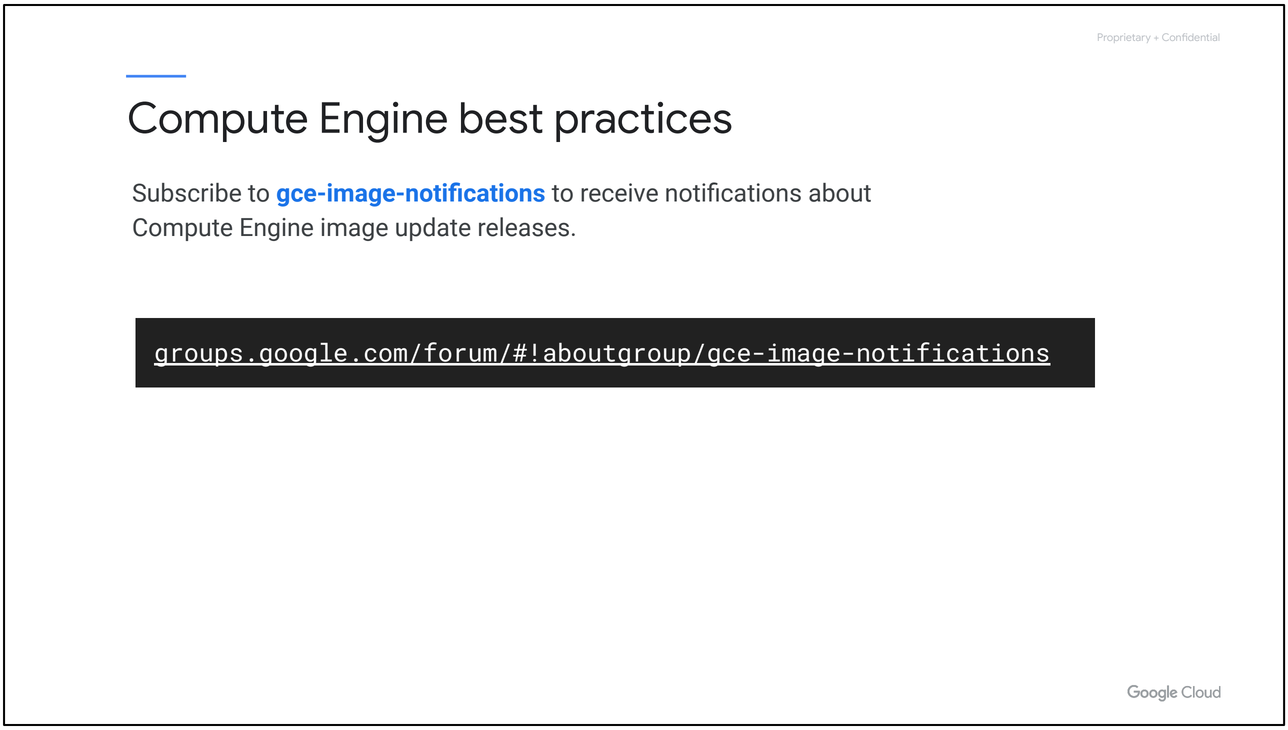
- Keep your deployed Compute Engine instances updated.
- Run VMs using custom service accounts with appropriate roles.
- Avoid using the default service account.

Compute Engine doesn't automatically update the OS or the software on your deployed instances. You will need to patch or update your deployed Compute Engine instances when necessary. However, it is not recommended that you patch or update individual running instances.

This could end up being a lot of work and risks a chance that something could be missed in the process. Instead, it is best to patch the image that was used to launch the instance and then replace each affected instance with a new copy.

In general, Google recommends that each instance that needs to call a Google API should run as a service account with the minimum permissions necessary for that instance to do its job. In practice, this means you should configure service accounts for your instances like this:

- Create a new service account rather than using the Compute Engine default service account.
- Grant IAM roles to that service account for only the resources that it needs.
- Configure the instance to run as that service account.



Proprietary + Confidential

## Compute Engine best practices

Subscribe to **gce-image-notifications** to receive notifications about Compute Engine image update releases.

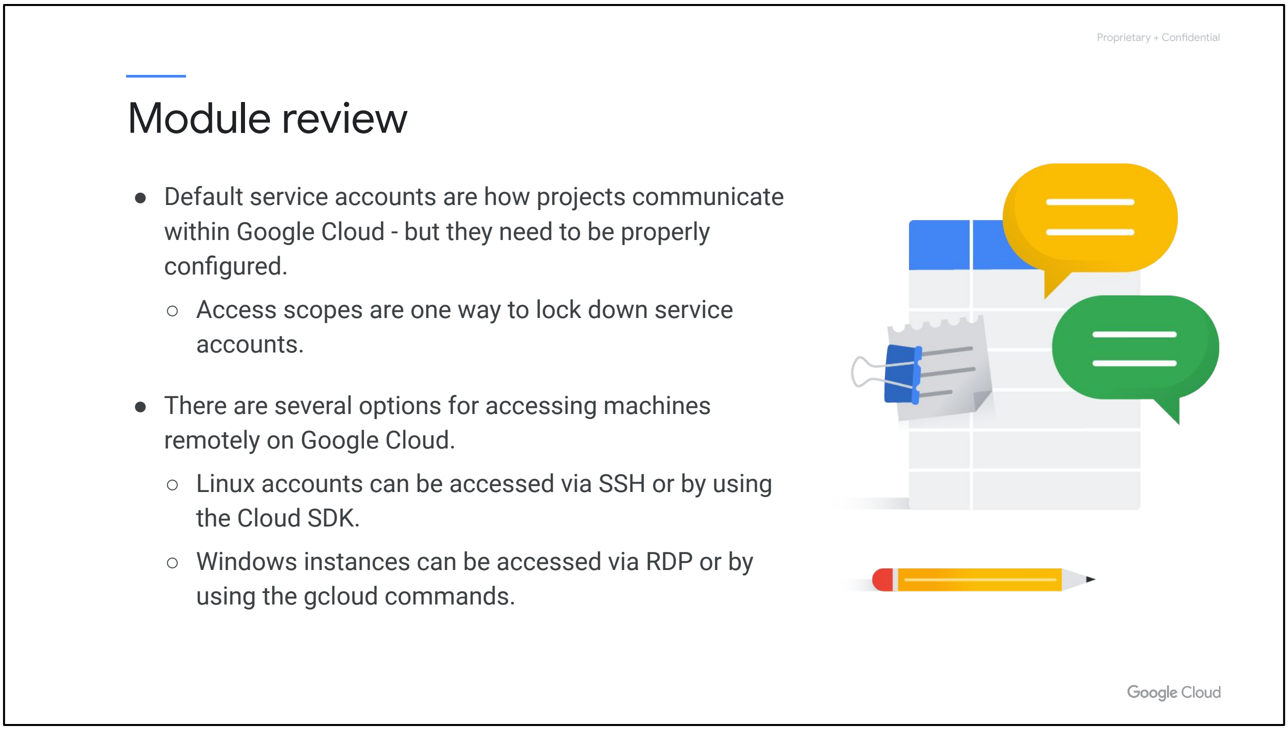
```
groups.google.com/forum/#!aboutgroup/gce-image-notifications
```

Google Cloud

Subscribe to “gce-image-notifications” announcements to receive release notes and other updates regarding public Compute Engine images.

- **Link:** groups.google.com/forum/#!aboutgroup/gce-image-notifications

This will be of interest to anyone looking to keep up with the latest information about Compute Engine Images, feel free to subscribe.



## Module review

- Default service accounts are how projects communicate within Google Cloud - but they need to be properly configured.
  - Access scopes are one way to lock down service accounts.
- There are several options for accessing machines remotely on Google Cloud.
  - Linux accounts can be accessed via SSH or by using the Cloud SDK.
  - Windows instances can be accessed via RDP or by using the gcloud commands.

Before we move to the next module, let’s review some key concepts from this one.

**Default service accounts** are how projects communicate within Google Cloud - but they need to be properly configured

- Every Google Cloud project has a default service account that is automatically created when compute engine is first enabled for the project.
- This default service account is assigned the “**project editor role”** and is used by default when launching VMs.
- You can also create and manage your own service accounts using Google Identity and Access Management.

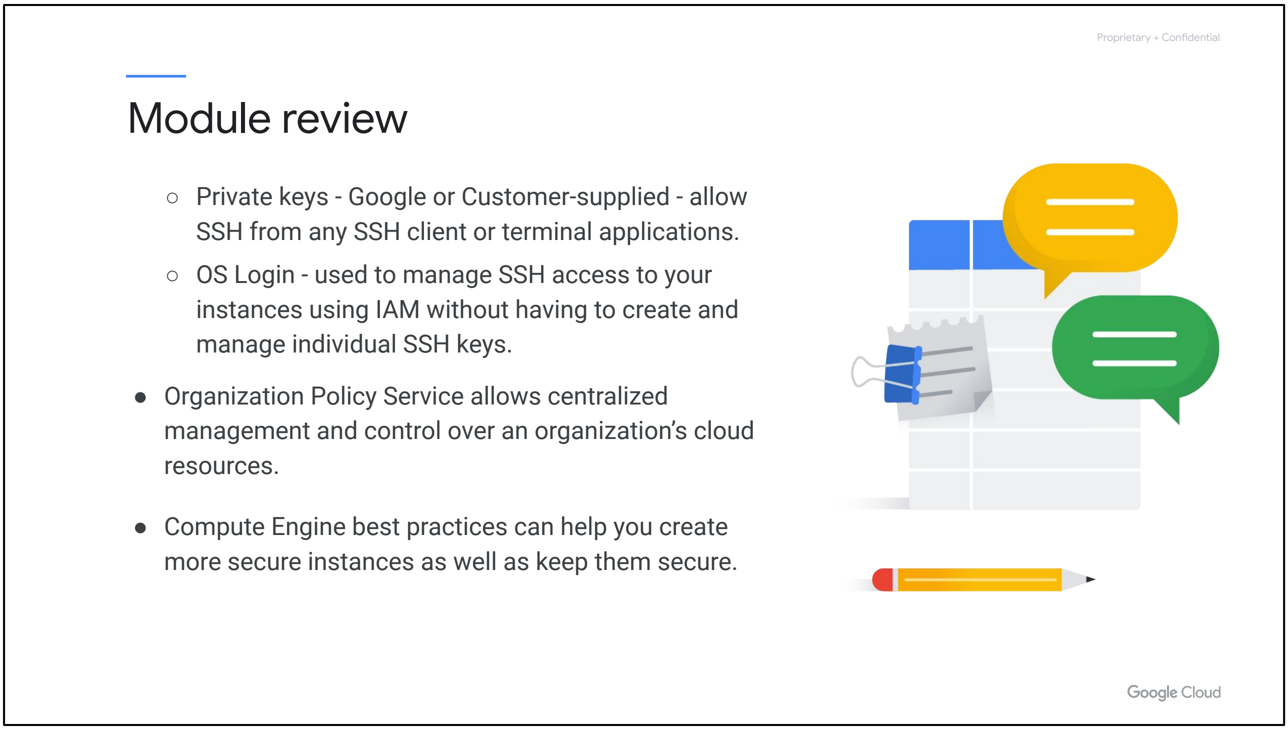
**Access scopes** provide the ability to limit what permissions are allowed when using the default service account with IAM Project Editor role permissions

- You set access scopes when creating an instance and the access scopes persists only for the life of that instance.
- Another option is to set the access for each API individually, which allows you to grant access to only to the APIs required by the programs running on the VM.

**There are several options for accessing machines remotely on Google Cloud**

- By default, Linux instances on Google Cloud are accessed with SSH and require a username and an SSH key for authentication. Password authentication is disabled by default.  It is also possible to connect to a Linux instance via SSH using the gcloud SDK.

- Window instances are accessed using RDP and require a username and password to authenticate. The username and password can be set from the Cloud Console or using gcloud.



## Module review

- ○ Private keys - Google or Customer-supplied - allow SSH from any SSH client or terminal applications.
- ○ OS Login - used to manage SSH access to your instances using IAM without having to create and manage individual SSH keys.
- Organization Policy Service allows centralized management and control over an organization’s cloud resources.
- Compute Engine best practices can help you create more secure instances as well as keep them secure.

Google Cloud

- It’s also possible to SSH from any SSH client such as PuTTY on Windows or ssh applications on Linux or Mac computer using the public IP address of the instance, a valid username and SSH private key. You can manually create your own SSH key pairs, using tools like PuTTYgen or ssh-keygen.
- You can also use OS Login to manage SSH access to your instances using IAM without having to create and manage individual SSH keys.

**The Organization Policy Service** gives you centralized and programmatic control over your organization's cloud resources. An organization policy administrator can configure restrictions across your entire resource hierarchy. A constraint is a particular type of restriction (List or Boolean) against a Google Cloud service or a list of Google Cloud services.

**Compute Engine best practices** can help you create more secure instances as well as keep them secure. Compute Engine best practices include: control access, isolate machines, connect securely and regularly monitor and audit logs. In addition, be sure to keep instances updated, and avoid using the default service account, especially if it is unmodified.

Ok, you are now ready to move on to the next module!