Google Cloud Computing Foundation Course
Seth Vargo
Google Cloud

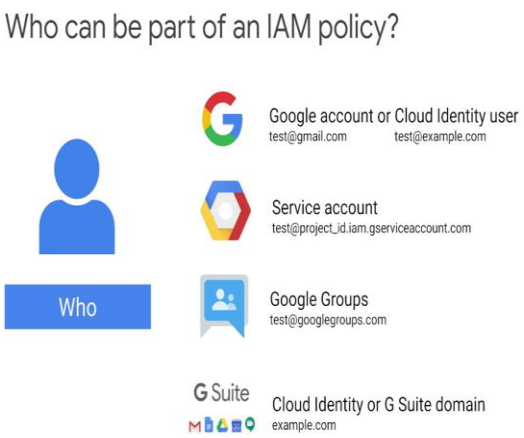
Lecture-47
Understand authentication and authorization

In this next lesson you will learn how to leverage authentication and authorization with Google Cloud IAM to improve the security of your infrastructure.

(Refer Slide Time: 00:11)

Cloud Identity and Access Management or cloud IAM enables cloud administrators to authorize who can do what on which resource in Google Cloud.

(Refer Slide Time: 00:21)



IAM policies can apply to many types of user like resources the who part of an I am policy can be a Google account or a cloud identity user, a service account, a Google Group or an entire G suite or cloud identity domain.

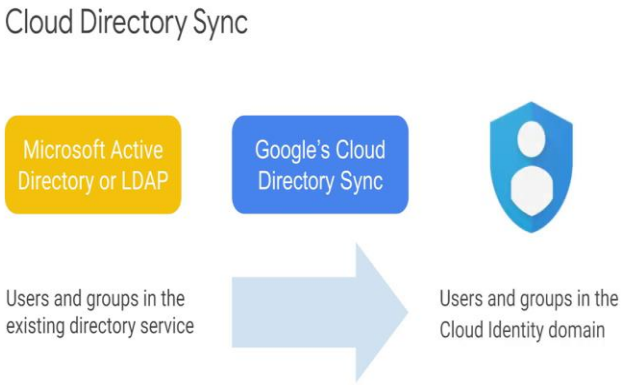
(Refer Slide Time: 00:40)

Many users get started by logging into the GCP console with a personal Gmail account to collaborate with their teammates they use Google Groups to gather together people who are in the same role. This approach is easy to get started but its disadvantages that teams identities aren't centrally managed. For example if someone leaves the organization or team there is no central way to remove their access to the cloud resources immediately.

(Refer Slide Time: 01:07)

GCP users who are also G suite users can define Google cloud policies in terms of G suite users and groups this way when someone leaves the organization an administrator can immediately disable their account using the Google cloud admin console for G suite. GCP users who are not G suite users can gain these same capabilities through cloud identity. Cloud identity allows users and groups to be managed using the Google cloud admin console but the G suite collaboration products like Gmail, Docs, Drive and calendar are not included. For this reason cloud identity is available for free.

(Refer Slide Time: 01:47)

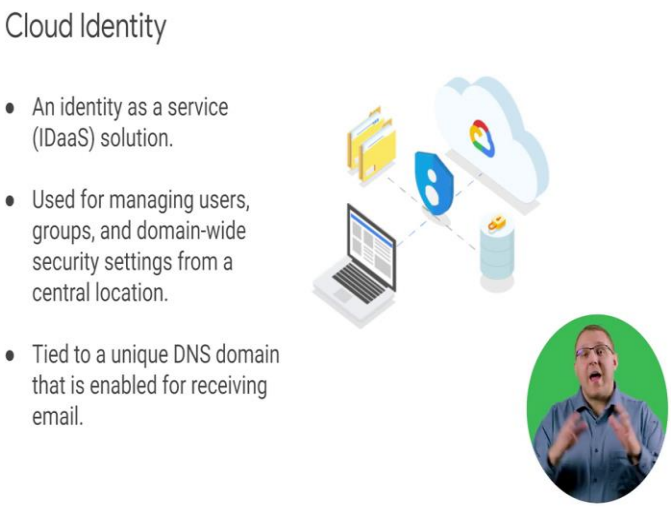


But what if you already have a centralized user management and identity system like Microsoft Active Directory or LDAP well Google clouds directory sync can help. This tool synchronizes

users and groups from an existing Active Directory or LDAP system. Mapping the users and groups in a cloud identity domain this synchronization is only one way though. Cloud directory sync cannot modify information in Microsoft Active Directory or LDAP systems.

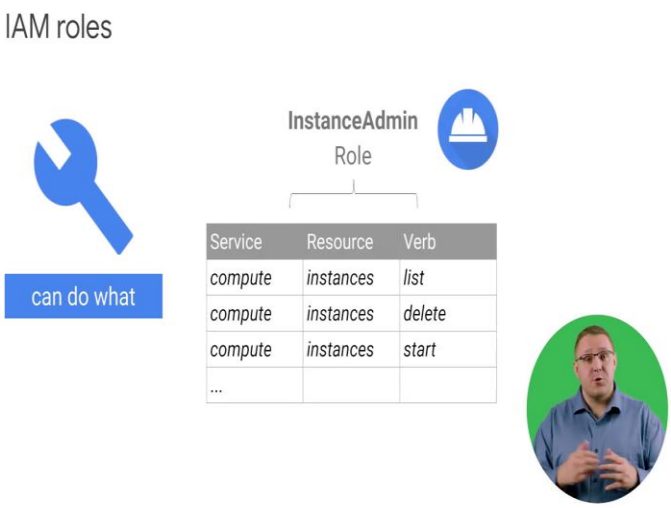
Cloud directory sync is usually scheduled to run without supervision on a fixed interval like every 24 hours.

(Refer Slide Time: 02:24)



We have mentioned cloud identity a few times. Now let us dive into a little more detail cloud identity is a unified identity access and device management platform. Cloud identity is an identity as a service solution it is a service for managing users groups and security settings. Cloud identity can be used as a central source for domain wide settings. Cloud identity is associated with a unique public domain. It can work with any domain name that is enabled for receiving email messages.

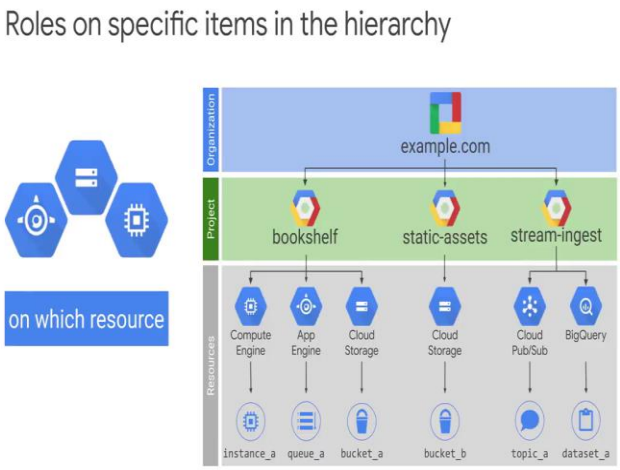
(Refer Slide Time: 02:58)



Now that we have talked about the who? Let us discuss the can do what part of IAM. The can do what part is defined by an IAM role which is a collection of I am permissions. Permissions are very low-level and fine-grained. For example to manage a virtual machine you need permissions to create delete stop start and change an instance. To make this process easier permissions are often grouped together into an IAM role to make them easier to manage.

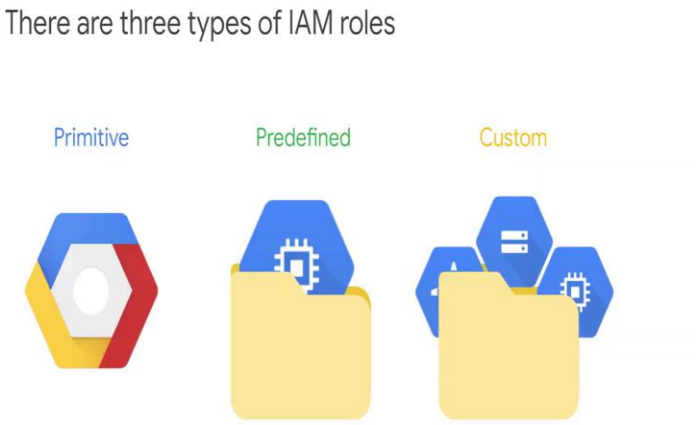
There are built-in roles available to all GCP users and you can also build your own and customize your own roles for your organization.

(Refer Slide Time: 03:43)



Finally let us discuss the on which resource part of IAM. When you give a user group or service account permissions on a specific element of the resource hierarchy the resulting policy applies to the element you choose as well as the elements below that resource in the hierarchy.

(Refer Slide Time: 04:02)



There are three kinds of roles in cloud IAM primitive predefined and custom. Let us talk about each of them in turn.

(Refer Slide Time: 04:11)

IAM primitive roles apply across all GCP resources in a project. These primitive roles include owner editor viewer and billing admin. If you are a viewer you can examine resources but you cannot change their state. If you are an editor you can do everything a viewer can do plus modify

state and if you are an owner you can do everything in editor can do plus manage roles and permissions. The owner role of a project also gives you control of our billing and cost management functionality.

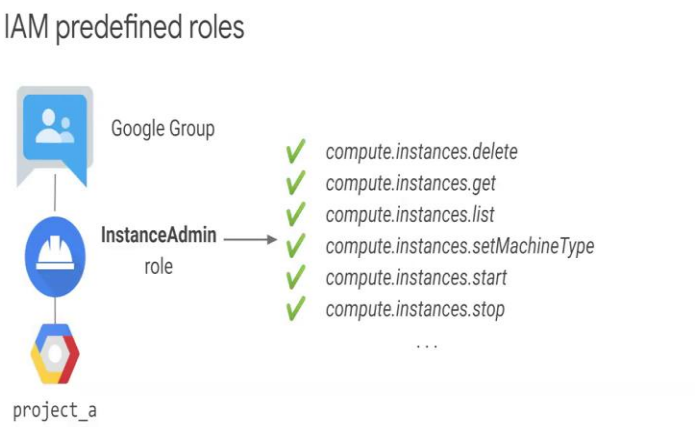
Often organizations want someone to be able to control the billing for a project without the right to change the resources in that project. You can grant someone the billing administrator role which grants access to billing information but does not grant access to resources inside the project.

(Refer Slide Time: 05:07)

IAM predefined roles apply to a particular GCP service in a project. GCP services offer their own set of predefined roles and they define where those roles can be applied. For example Google compute engine offers a set of predefined roles and you can apply them to compute engine resources in a given project, a given folder or the entire organization. Another example is cloud BigTable which is a managed database service.

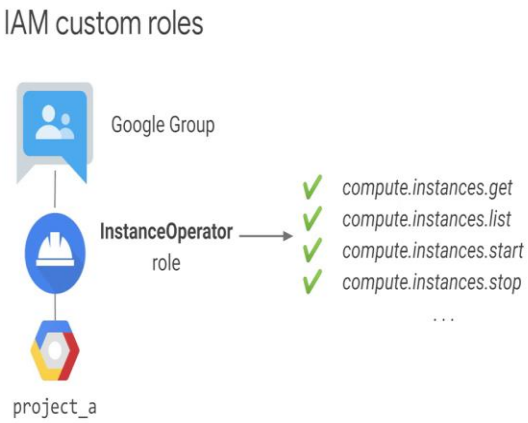
Cloud BigTable offers roles that can apply across an entire organization to a particular project or even individual cloud BigTable database instances.

(Refer Slide Time: 05:47)



IAM predefined roles offer more fine-grained permissions on particular services. The Google compute engine instance admin role allows whoever has it to perform a certain set of actions on virtual machines. In this example all the users of a certain Google Group have the role and they have it on all virtual machines in project A. The last kind of IAM role is a custom role, for some organizations the primitive and predefined IAM roles may not offer enough granularity.

(Refer Slide Time: 06:22)



I am custom roles allow you to create your own roles that are composed of very granular permissions. In this example we have defined a new custom roll named instance operator that allows users to start and stop instances but does not give them permission to delete or

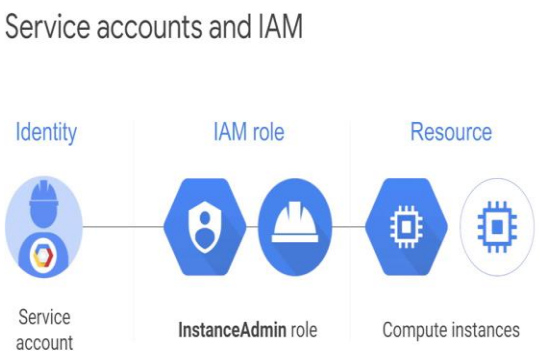
reconfigure them. At this time rolls can only be applied at project and organization levels it is not currently possible to apply custom roles at the folder level.

(Refer Slide Time: 06:49)

Another important concept related to identity and access management is service accounts. Service accounts control service to service communication in order for services to interact with each other they need some kind of identity. Service accounts are used to authenticate service to service communication. With service accounts you can give a role level access from one service to another. Suppose you have an application running in a virtual machine that needs to access data in cloud storage.

You only want that virtual machine to have access to that data you can create a service account that is authorized to access that data in cloud storage and then attach that service account to the virtual machine. Service accounts are named with an email address oftentimes ending in G service account com.

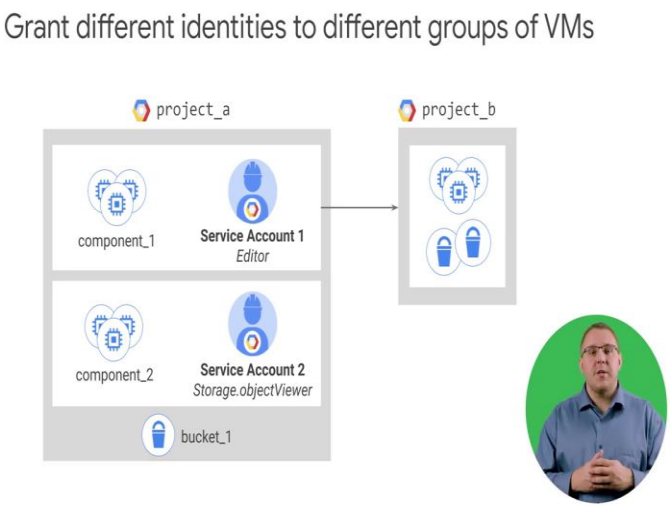
(Refer Slide Time: 07:44)



In this example a service account has been granting the instance admin role. This would allow an application running in a virtual machine with that service account to create modify and delete other virtual machines incidentally service accounts need to be managed too. For example maybe Alice needs to manage what can act as a given service account while Bob just needs to be able to view what a particular service account can do. Fortunately in addition to being an identity a service account is also a resource.

So, it can have IAM policies of its own attached to it. For instance Alice can have the editor role on a service account and Bob can have the viewer role this is just like granting roles for any other GCP resource like a virtual machine.

(Refer Slide Time: 08:33)



You can grant virtual machines different identities this makes it easier to manage different project permissions across your applications. You can also manage the permissions of the service accounts without having to recreate the virtual machines. Here is a more complex scenario, say you have an application that is implemented across a group of virtual machines. One component of your application requires the editor role on another project, project B but another component does not need any permission on project B.

You would want to create two different service accounts one for each subgroup of virtual machines. In this example VM is running in component one are granted editor access to project B by using service account one. Virtual machines running component two are granted object viewer access to bucket one using service account two. Service account permissions can be changed without recreating the VM's.