Infrastructure Automation with Google Cloud Platform APIs

Elastic Cloud Infrastructure: Scaling and Automation

COMPUTE ENGINE, CLOUD APIS

QWIKLABS GOOGLE CLOUD PLATFORM API INFRASTRUCTURE AUTOMATION

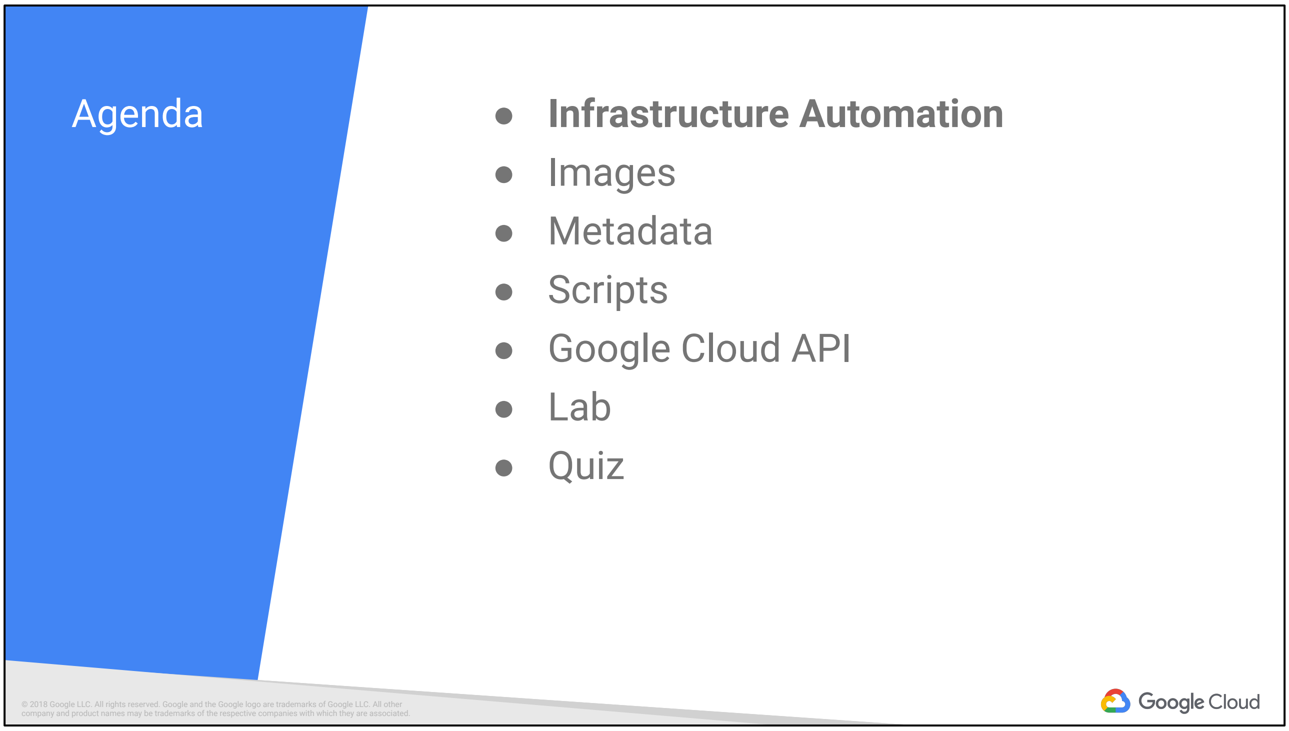
Google Cloud

Last modified 2018-01-29

[Script]
Hello and welcome, I'm Mylene Biddle and I'm a Technical Curriculum Developer for Google Cloud Platform. This is Module 4, Infrastructure Automation with Google Cloud Platform APIs.

Up to this point, most of the infrastructure you've created has been manually provisioned using the GCP console or cloud shell. In one of the earlier labs, you used cloud launcher to provision an entire infrastructure solution. In the autoscaling lab, you triggered the auto scaling servers that were provisioned for you dynamically.

Now we're going to look at ways you can automate infrastructure provisioning and configuration. These techniques are critical to any production system because they make the generation of the infrastructure documentable, accountable, and repeatable.



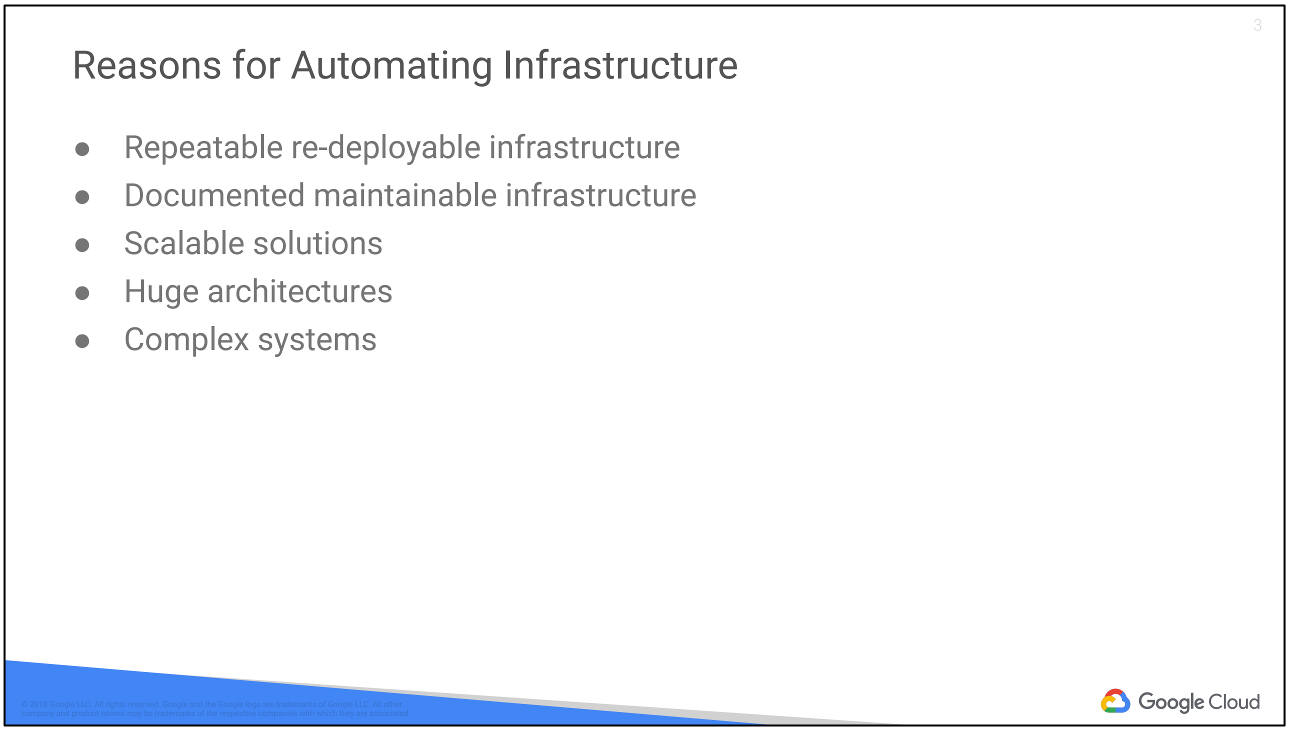
Agenda

- **Infrastructure Automation**
- Images
- Metadata
- Scripts
- Google Cloud API
- Lab
- Quiz

Google Cloud

[Script]
We will discuss infrastructure automation, images, metadata, scripts, and the Google Cloud Platform API.

You will complete a hands-on lab that explores GCP infrastructure automation. Let's get started!



Reasons for Automating Infrastructure

- Repeatable re-deployable infrastructure
- Documented maintainable infrastructure
- Scalable solutions
- Huge architectures
- Complex systems

Google Cloud

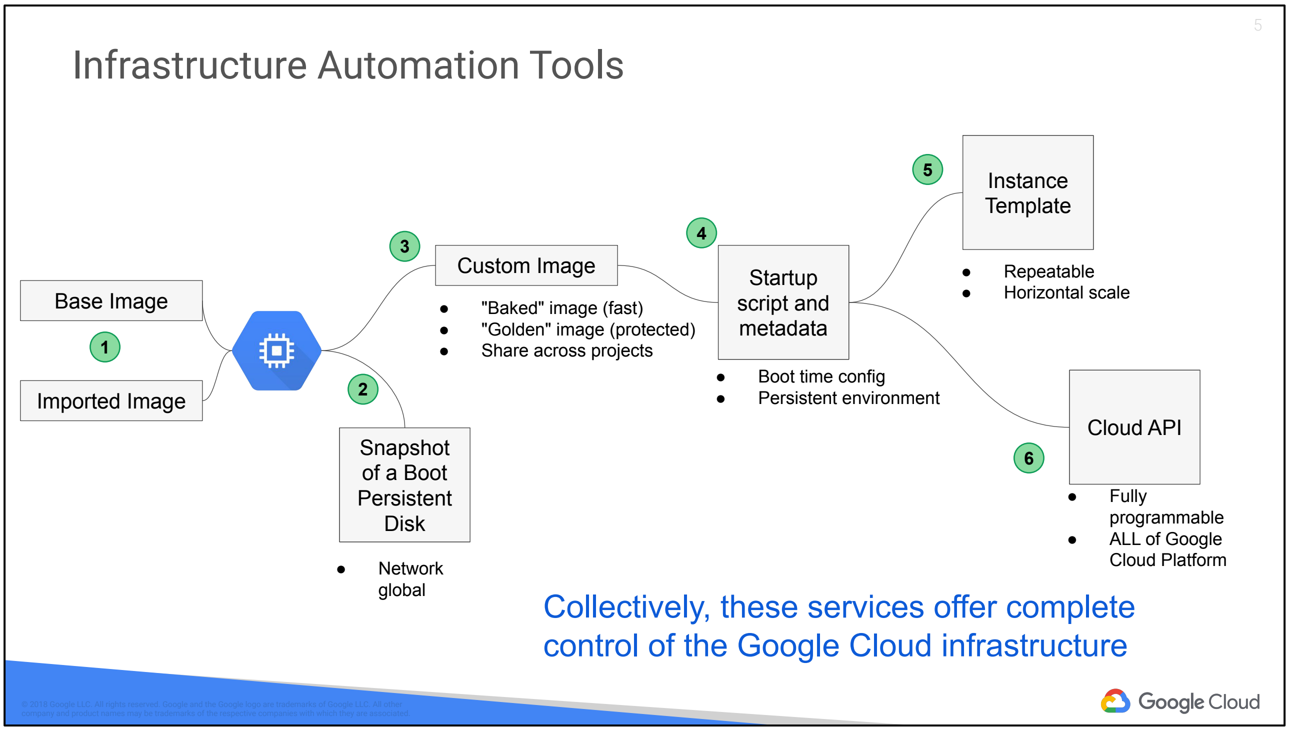
[Script]
Why would you ever want to automate infrastructure if you could just set it up once, go back and
just maintain it whenever you need to? You automate because you need to make it repeatable. Automation allows you to redeploy your infrastructure or rebuild it from scratch because you have a repeatable documented process.

Automation allows you to scale, ensuring consistency, as hopefully the success of your
business grows. You can create a consistent automated process to ensure ease of growth.

If you're going to be setting up large complex systems, you really need a way to ensure that
there's a repeatable process that prevents and corrects human error.

[Student Notes]
You know how to build infrastructure "by hand" with Console and Cloud Shell. This module and the next explore methods for automating the building of infrastructure, which is sometimes called "orchestration". The technologies covered are "force multipliers" for the methods you've already learned.

Listed above are some reasons for Automation. Other reasons include standards and policy compliance, support for internal or external audits, risk abatement, disaster recovery, survivability of the system due to loss of expertise, and many other reasons.



Infrastructure Automation Tools

Collectively, these services offer complete control of the Google Cloud infrastructure

Google Cloud

[Script]
When looking at the entire spectrum of automation tools, oftentimes you start with some kind of base image.hiccup Google base images are maintained, updated, and patched. Imported images can be useful for synchronizing with an existing system. You can use the image to create a running VM and you can connect to the VM and change its configuration, install software, and so forth.

After the VM is tailored to your liking, you can create a Snapshot of its boot disk. Snapshots are global resources. You can use them to reconstitute a boot disk in any region in any network in your project.

You can also create a custom image. A custom image can be shared between projects and there are tools for managing those shared images with your image users. When we say "custom image" we generally mean that the OS and system settings are customized. A "baked" image is one with pre-installed and preconfigured software. Baked images are generally faster to become operational than other methods of installing software during or soon after boot. A "Golden image" is one with all the settings "just right", ready for sharing. Another reason to bake an application into an image is to lock the application so it is harder to make changes to its configuration.

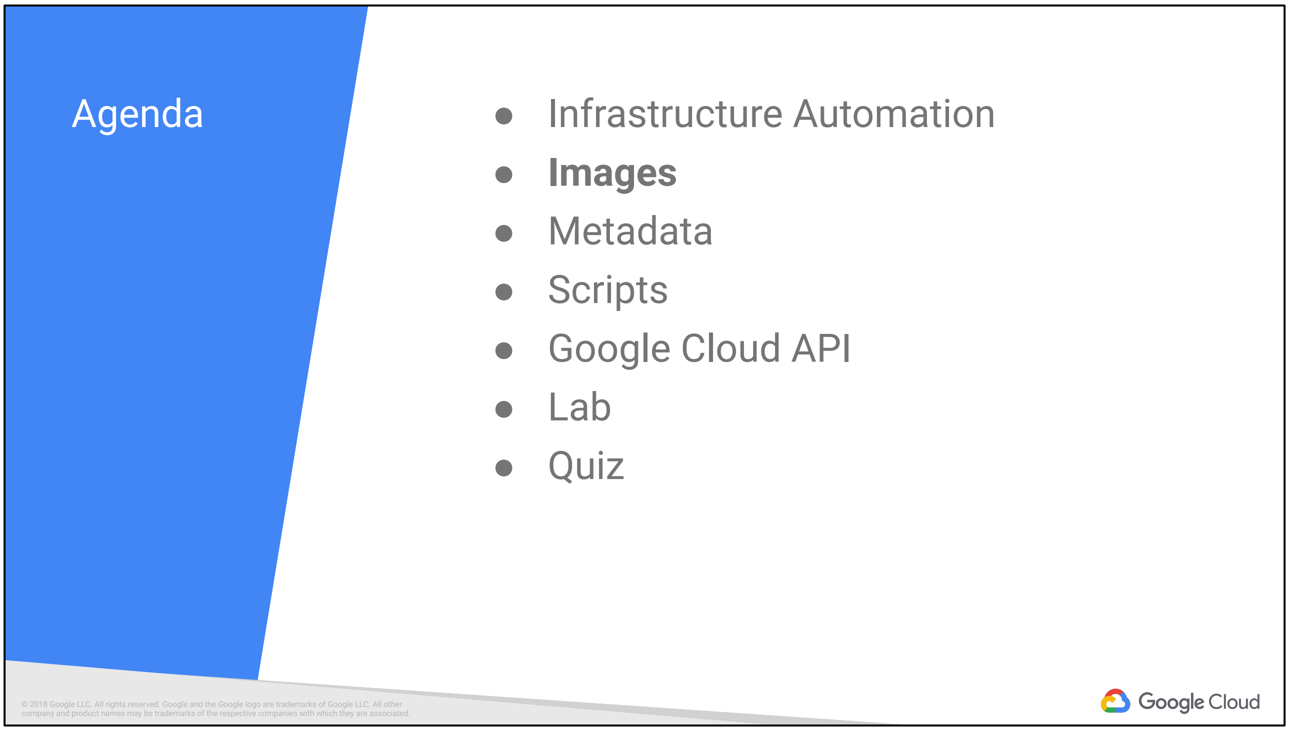
The use of startup scripts and metadata is one method used to implement boot time customization and software installation. The benefit is being able to change configurations (including which software to install) on the fly. It is ideal for passing parameters that can only be known when the VM is being created and not before. The metadata provides a persistent environment that survives the termination of any individual VM. So you can use metadata to maintain system-level infrastructure data and state information.

Any kind of image, startup scripts, and metadata can be used in an Instance Template. Instance Templates give you a system-documented repeatable way to make identical VMs. When used with a Managed Instance Group and an Autoscale they provide horizontal scaling.

All of these tools and methods get you consistent, reliable, and automatic VM creation. But what they don't do is provide automation over the rest of the GCP infrastructure; they don't create load balancers, vpn connections, or networks. One solution is to install the Cloud SDK on a VM, authorize the VM to use parts of the Google Cloud APIs, and write programs that automate infrastructure creation. Remember that everything you've learned to do with the Console and the gcloud and gsutil commands was implemented by calling the Cloud API. You could write programs that call the Cloud API to programmatically create and manage infrastructure.

[Student Notes]
(1) You can start with a standard base image or you can import an image from another domain (on prem or another cloud). Google base images are maintained, updated, and patched. Imported images are useful for synchronizing with an existing system.
You can use the image to create a running VM. And you can connect to the VM and change its configuration, install software, and so forth.
(2) After the VM is tailored to your liking, you can create a Snapshot of its boot disk. Snapshots are global resources. You can use them to reconstitute a boot disk in any region in any network in your project. However, you can't share them between projects.
(3) A custom image can be shared between projects. And there are tools for managing those shared images with your image users. When we say "custom image" we generally mean that the OS and system settings are customized. A "baked" image is one with pre-installed and preconfigured software. Baked images are generally faster to become operational than other methods of installing software during or soon after boot. A "Golden image" is one with all the settings "just right", ready for sharing. Another reason to bake an application into an image is to lock the application so it is harder to make changes to its configuration.

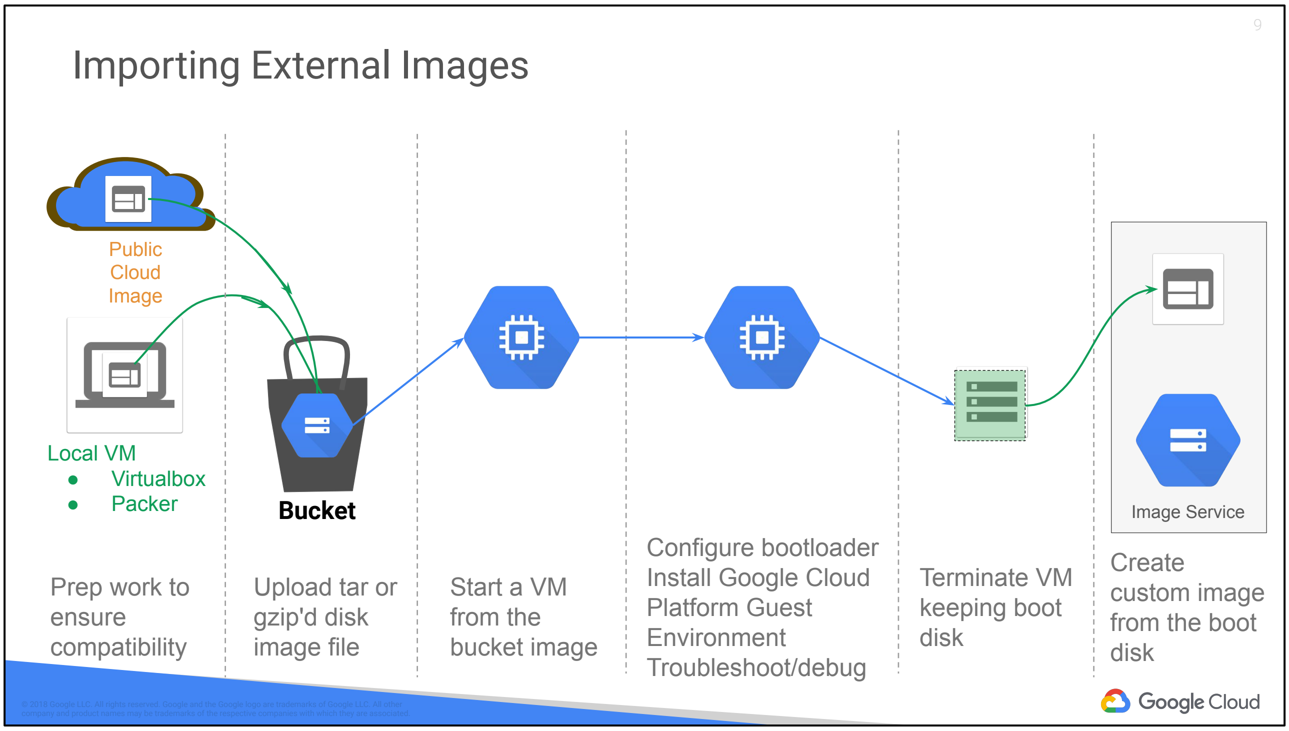
(4) A startup script and metadata is one method to implement boot time customization and software installation. The benefit is being able to change configurations (including which software to install) on the fly. It is ideal for passing parameters that can only be known when the VM is being created and not before. The metadata provides a persistent environment that survives the termination of any individual VM. So you can use metadata to maintain system-level infrastructure data and state information.
(5) Any kind of image, startup scripts, and metadata can be used in an Instance Template. Instance Templates give you a system-documented repeatable way to make identical VMs. When used with a Managed Instance Group and an Autoscale they provide horizontal scaling.
(6) All of these tools and methods get you consistent, reliable, and automatic VM creation. But what they don't do is provide automation over the rest of the GCP infrastructure; they don't create load balancers, vpn connections, or networks. One solution is to install the Cloud SDK on a VM, authorize the VM to use parts of the Google Cloud APIs, and write programs that automate infrastructure creation. Remember that everything you've learned to do with the Console and the gcloud and gsutil commands was implemented by calling the Cloud API. You could write programs that call the Cloud API to programmatically create and manage infrastructure.



Agenda

- Infrastructure Automation
- **Images**
- Metadata
- Scripts
- Google Cloud API
- Lab
- Quiz

[Script]
Let’s delve into exporting, creating, and managing custom images.



Importing External Images

Prep work to ensure compatibility	Upload tar or gzip'd disk image file	Start a VM from the bucket image	Configure bootloader Install Google Cloud Platform Guest Environment Troubleshoot/debug	Terminate VM keeping boot disk	Create custom image from the boot disk

[Script]
You can import boot disk images from your physical datacenters, from virtual machines on your local workstation, or from virtual machines that run on another cloud platform. The image import process can import only one disk at a time, and this example focuses on how to import boot disk images. To import a boot disk image to Compute Engine, use the following process:

- Plan your import path. You must identify where you are going to prepare your boot disk image before you upload it, and how you are going to connect to that image after it boots in the Compute Engine environment.
- Prepare your boot disk so it can boot within the Compute Engine environment and so you can access it after it boots.
- Create and compress the boot disk image file.
- Upload the image file to Google Cloud Storage and import the image to Compute Engine as a new custom image.
- Use the imported image to create a virtual machine instance and make sure it boots properly.
- If the image does not successfully boot, you can troubleshoot the issue by attaching the boot disk image to another instance and reconfiguring it.
- Optimize the image and install the Linux Guest Environment so that your imported operating system image can communicate with the metadata server

- and use additional Compute Engine features.

[Student Notes]
Importing an external image:
"There is significant work involved. You might not want to do it every day."

Required prep work:
https://cloud.google.com/compute/docs/images/import-existing-image

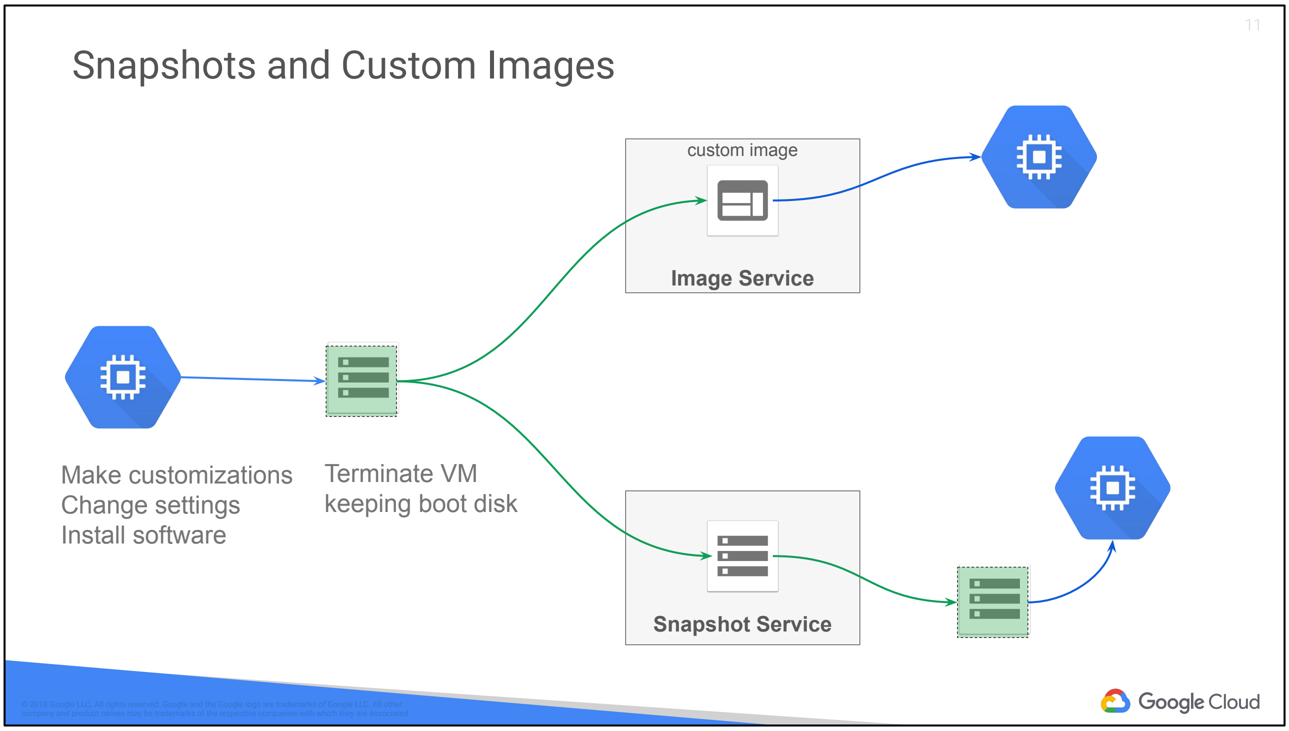
GCP Guest Environment (Linux example)

- Accounts daemon to setup and manage user accounts, and to enable SSH key based authentication.
- Clock skew daemon to keep the system clock in sync after VM start and stop events.
- Disk expand scripts to expand the VM root partition for boot disks with CentOS/RHEL 6 and 7 operating systems.
- Instance setup scripts to execute VM configuration scripts during boot.
- IP forwarding daemon that integrates network load balancing with forwarding rule changes into the guest.
- Metadata scripts to run user provided scripts at VM startup and shutdown.
- Network setup service to enable multiple network interfaces on boot.

https://cloud.google.com/compute/docs/images/import-existing-image

Migrating VMs to GCP: https://cloud.google.com/migrate/

Developing Packer images for GCP:
https://www.packer.io/docs/builders/googlecompute.html



Snapshots and Custom Images

[Script]
To create a VM from a Snapshot backup of a Persistent Boot Disk, you need to first create a new disk from the snapshot. Then you can use that disk to boot a new VM instance. Persistent Disks are Zonal resources. However, Snapshots are Global resources. So you can use a snapshot to create a copy of the boot disk in any region, and also, in any network in your project. Note that when you create a new disk in a different region or network that egress charges apply to the data.

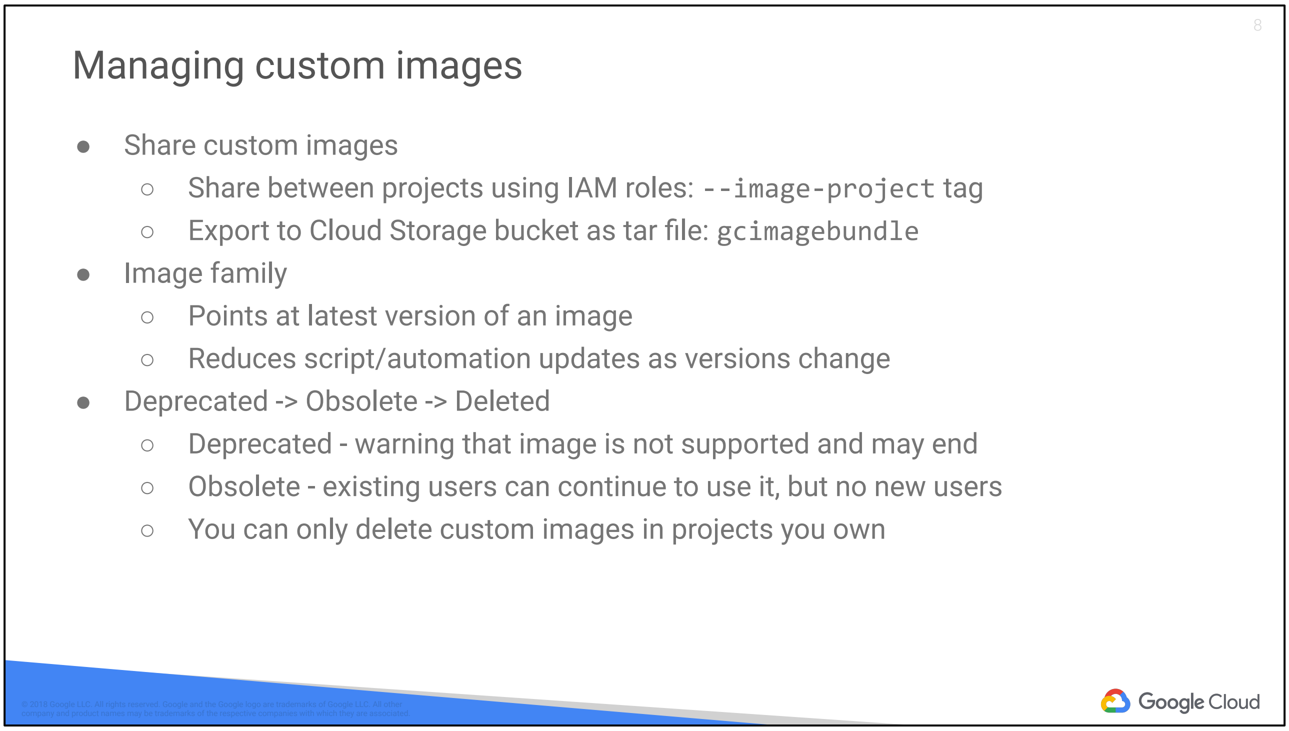
There are many benefits to creating an image from a VM's Persistent Boot Disk.
A VM can be generated directly from the custom image without first manually restoring to a disk.
The custom image can be used in Instance Templates
The custom image can be shared across projects

[Student Notes]
To create a VM from a Snapshot backup of a Persistent Boot Disk, you need to first create a new disk from the snapshot. Then you can use that disk to boot a new VM instance. Persistent Disks are Zonal resources. However, Snapshots are Global resources. So you can use a snapshot to create a copy of the boot disk in any region, and also, in any network in your project. Note that when you create a new disk in a different region or network that egress charges apply to the data.

There are many benefits to creating an image from a VM's Persistent Boot Disk.

- A VM can be generated directly from the custom image without first manually restoring to a disk.
- The custom image can be used in Instance Templates
- The custom image can be shared across projects

https://cloud.google.com/compute/docs/instances/windows/creating-windows-os-image
On some versions of Windows you need to run a preparation program that updates drivers and some system services: GCESysprep



Managing custom images

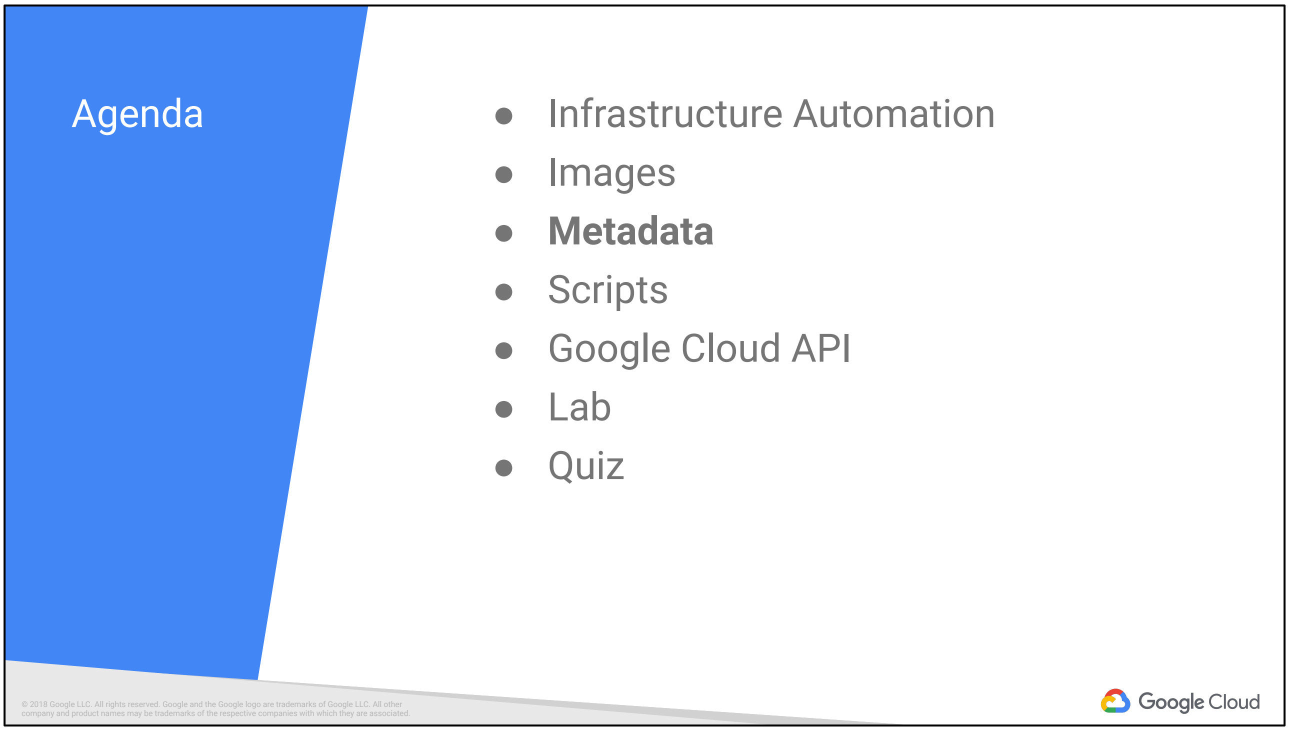
- Share custom images
  - Share between projects using IAM roles: `--image-project` tag
  - Export to Cloud Storage bucket as tar file: `gcimagebundle`
- Image family
  - Points at latest version of an image
  - Reduces script/automation updates as versions change
- Deprecated -> Obsolete -> Deleted
  - Deprecated - warning that image is not supported and may end
  - Obsolete - existing users can continue to use it, but no new users
  - You can only delete custom images in projects you own

[Script]
When you're managing custom images you can share those utilizing IAM. You can set image families, so that for example you can always point to the latest version of the image. This way people don't accidentally go and pick up the older versions.

You can also define a deprecation schedule. So when is something going to be deprecated? This gives a warning to the users that they might want to update one that's obsolete. This also lets users know they shouldn't be using that image because it's eventually going to be deleted.

[Student Notes]
https://cloud.google.com/compute/docs/images/create-delete-deprecate-private-images

Alternatively to using native Google Cloud Platform image, you could deploy Docker Containers to VMs. This feature is in Beta and more information can be found here:
https://cloud.google.com/compute/docs/containers/deploying-containers



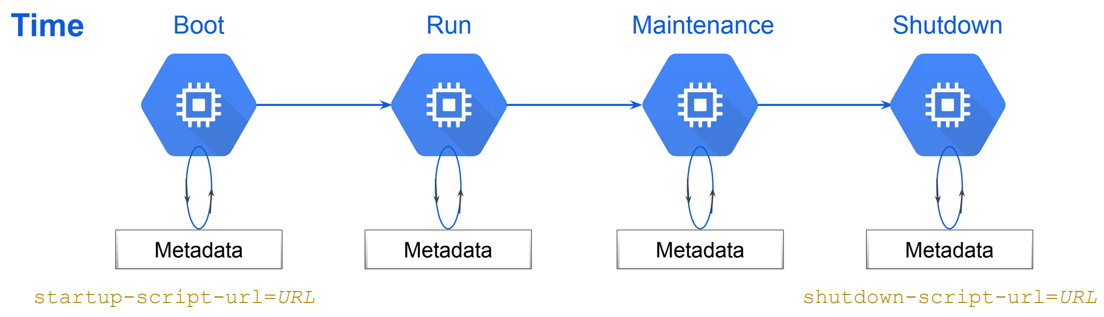
Agenda

- Infrastructure Automation
- Images
- **Metadata**
- Scripts
- Google Cloud API
- Lab
- Quiz

Google Cloud

[Script]
Let’s explore metadata for infrastructure automation.

Metadata and Scripts



Google Cloud

[Script]
Metadata is especially important when starting a VM. Throughout each of the individual phases as a machine is booting up, until it's running, and before it's actually put in a maintenance mode and shut down, we will query different metadata a tags that are available to you.

Many data tags are not labels. They are also not tags in the sense that a tag might be used for networking, firewalls, or routing purposes. In this case, the metadata are custom key value pairs that we've actually reserved to obtain specific information about the virtual machine. You can create your own custom metadata so you can store shut down and startup scripts in the metadata tags.

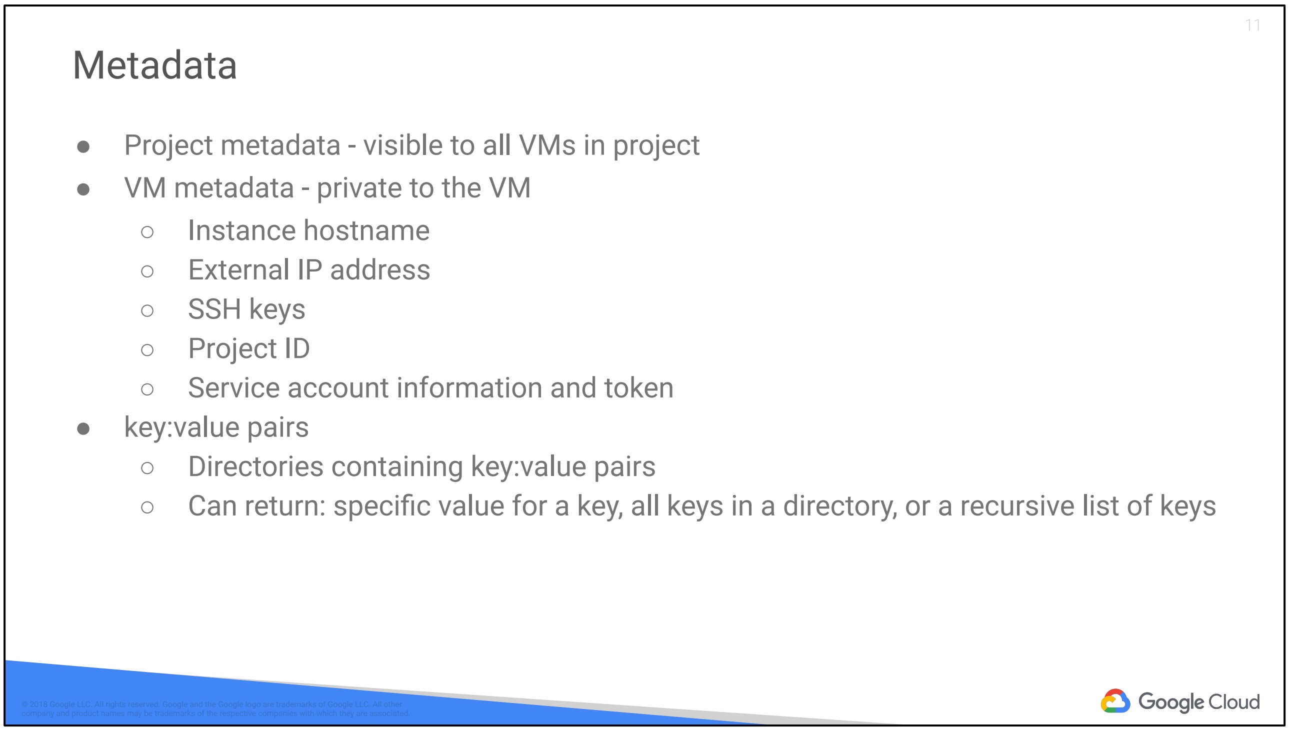
[Student Notes]
VMs have a special relationship with metadata, because the metadata is the standard place for persistent state information that survives the creation and termination of a VM.

During Boot, the Metadata tells the VM where to find the startup script.
When that Startup script is running, it can access additional metadata and get current information about its environment.

Also, scripts and applications can access metadata while the VM is running. And an application can watch a metadata object and receive notification when that object changes. So it can be used to coordinate actions between VMs in a system. (Note that Cloud Pub/Sub might be the better tool for this).

When a maintenance event is about to occur, such as a live migration, there is a (not guaranteed) 30 to 60 second warning. An application can register to watch a special value to get notification when the VM that the application is running on is about to undergo a live migration event. Then the application can take appropriate action.

Finally, when a VM is shutting down, it uses Metadata to find the shutdown script. In Managed Instance Groups the shutdown script might drain the connection from the Load Balancer to the VM to minimize user impact to the loss of the VM. Other shutdown activities: write logs, save state.



Metadata

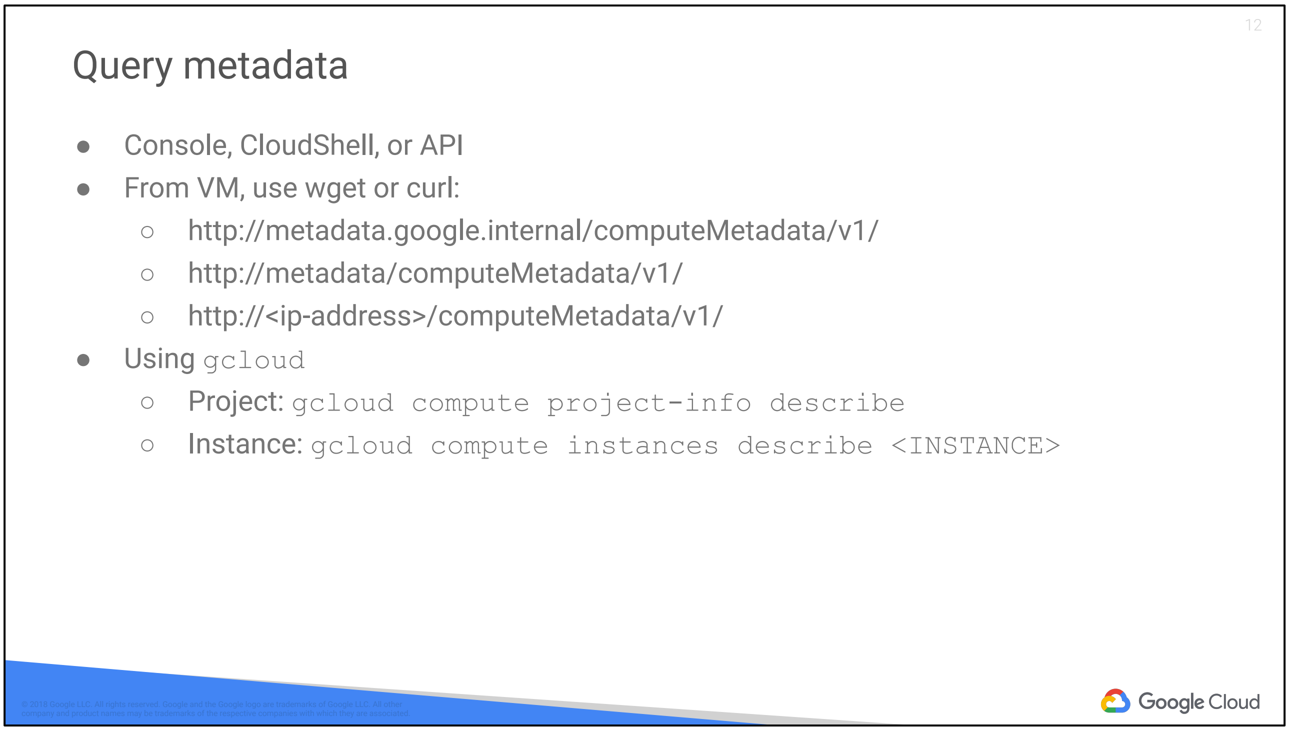
- Project metadata - visible to all VMs in project
- VM metadata - private to the VM
  - Instance hostname
  - External IP address
  - SSH keys
  - Project ID
  - Service account information and token
- key:value pairs
  - Directories containing key:value pairs
  - Can return: specific value for a key, all keys in a directory, or a recursive list of keys

Google Cloud

[Script]
There are different types of metadata tags available. Project-wide metadata is queryable by all of the VMs inside of our project.

Virtual machine specific metadata is private to the virtual machine. The VM, from within the VM, can run a query and find out its instance hostname, the SSH keys, project IDs and additional details. All metadata are simple key value pairs so you can return individual values for the key and probe an entire directory tree or a list of recursive keys as well.

[Student Notes]
https://cloud.google.com/compute/docs/storing-retrieving-metadata#default



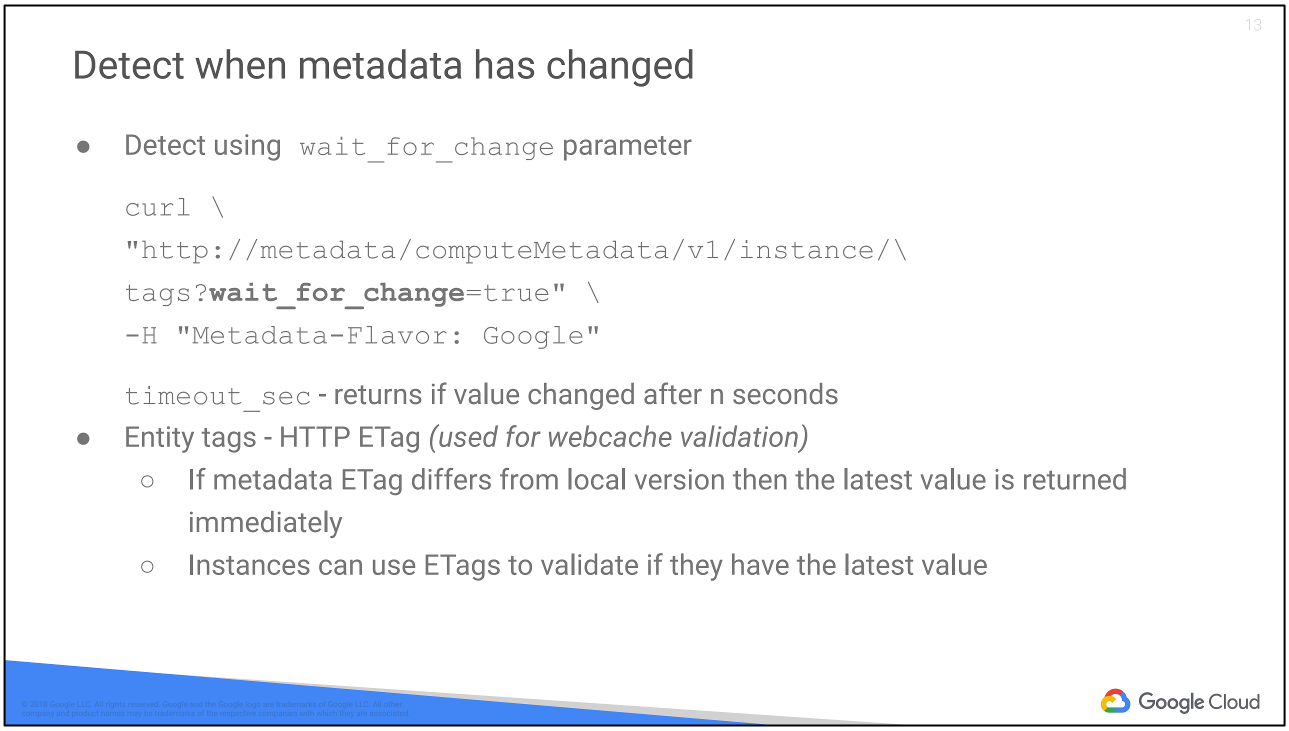
Query metadata

- Console, CloudShell, or API
- From VM, use wget or curl:
  - http://metadata.google.internal/computeMetadata/v1/
  - http://metadata/computeMetadata/v1/
  - http://<ip-address>/computeMetadata/v1/
- Using `gcloud`
  - Project: `gcloud compute project-info describe`
  - Instance: `gcloud compute instances describe <INSTANCE>`

Google Cloud

[Script]
You can query this metadata a number different ways obviously using the web console, cloud shell, or using the Google Cloud Platfrom API.

[Student Notes]
https://cloud.google.com/compute/docs/storing-retrieving-metadata#querying



Detect when metadata has changed

- Detect using `wait_for_change` parameter

  ```
  curl \
  "http://metadata/computeMetadata/v1/instance/\
  tags?wait_for_change=true" \
  -H "Metadata-Flavor: Google"
  ```

  `timeout_sec` - returns if value changed after n seconds
- Entity tags - HTTP ETag *(used for webcache validation)*
  - If metadata ETag differs from local version then the latest value is returned immediately
  - Instances can use ETags to validate if they have the latest value

[Script]
You can use the curl command to repetitively identify if there are any metadata changes.

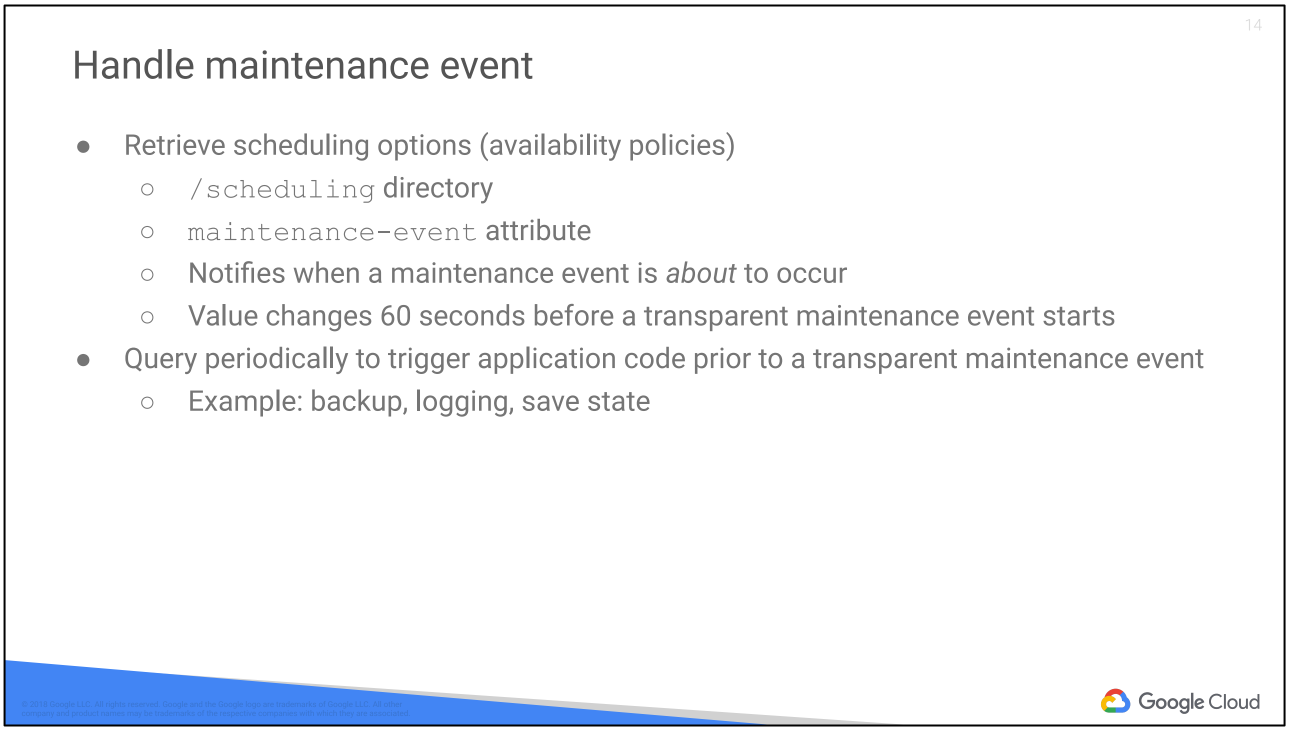
For example, with a preemptible virtual machine that preempt API notification is actually going to come as a metadata update, so you need to watch and wait for that particular update.

If your server's going to be migrated live, or if it has to go down for emergency host maintenance, those actions will all be done by updating the metadata tag.

In the first curl example, you can use curl to watch for the metadata tag change.

[Student Notes]
Use hanging GET request to wait for metadata changes and programmatically react by including the `wait_for_chage` parameter.

ETags do not work on directory listings or service account tokens.



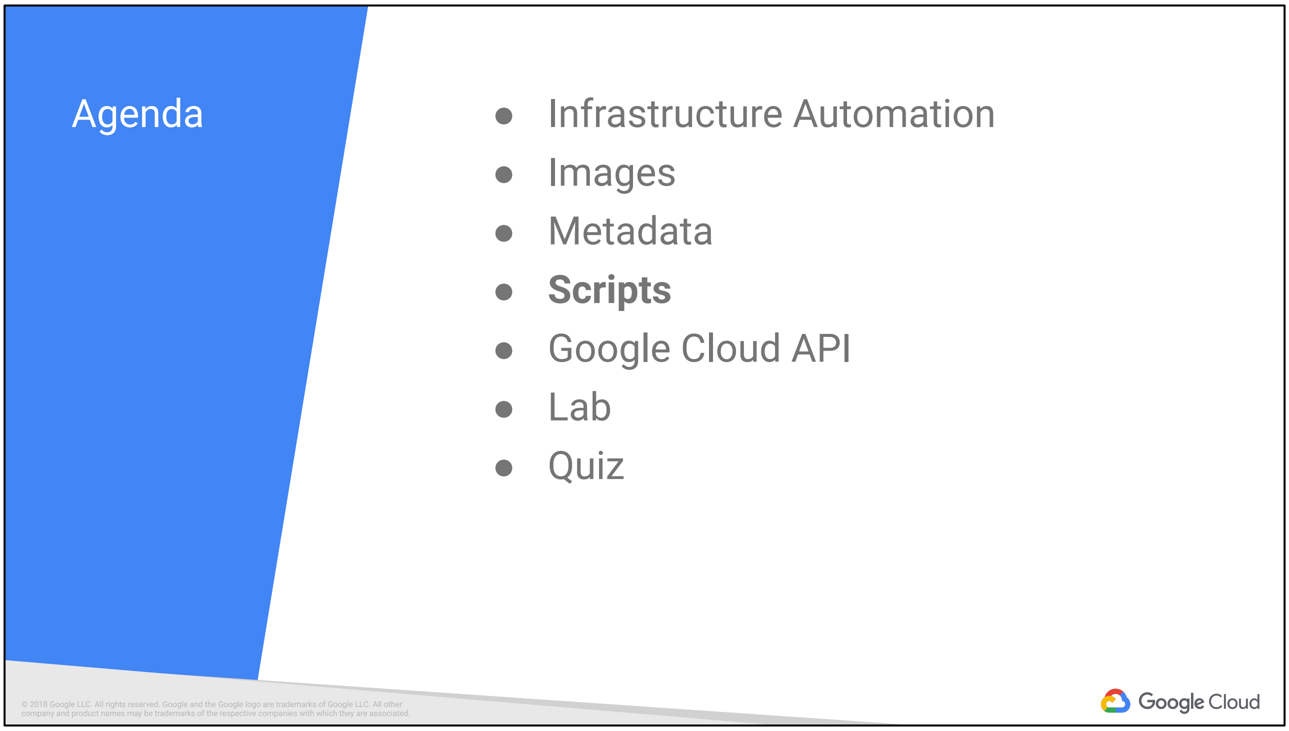
Handle maintenance event

- Retrieve scheduling options (availability policies)
  - `/scheduling` directory
  - `maintenance-event` attribute
  - Notifies when a maintenance event is *about* to occur
  - Value changes 60 seconds before a transparent maintenance event starts
- Query periodically to trigger application code prior to a transparent maintenance event
  - Example: backup, logging, save state

[Script]
Maintenance events involve updating metadata attributes. You're going to get and update about
60 to 90 seconds before the event actually occurs.

It's your applications' job to monitor for the notifications to react appropriately.

[Student Notes]
https://cloud.google.com/compute/docs/storing-retrieving-metadata#maintenanceevents

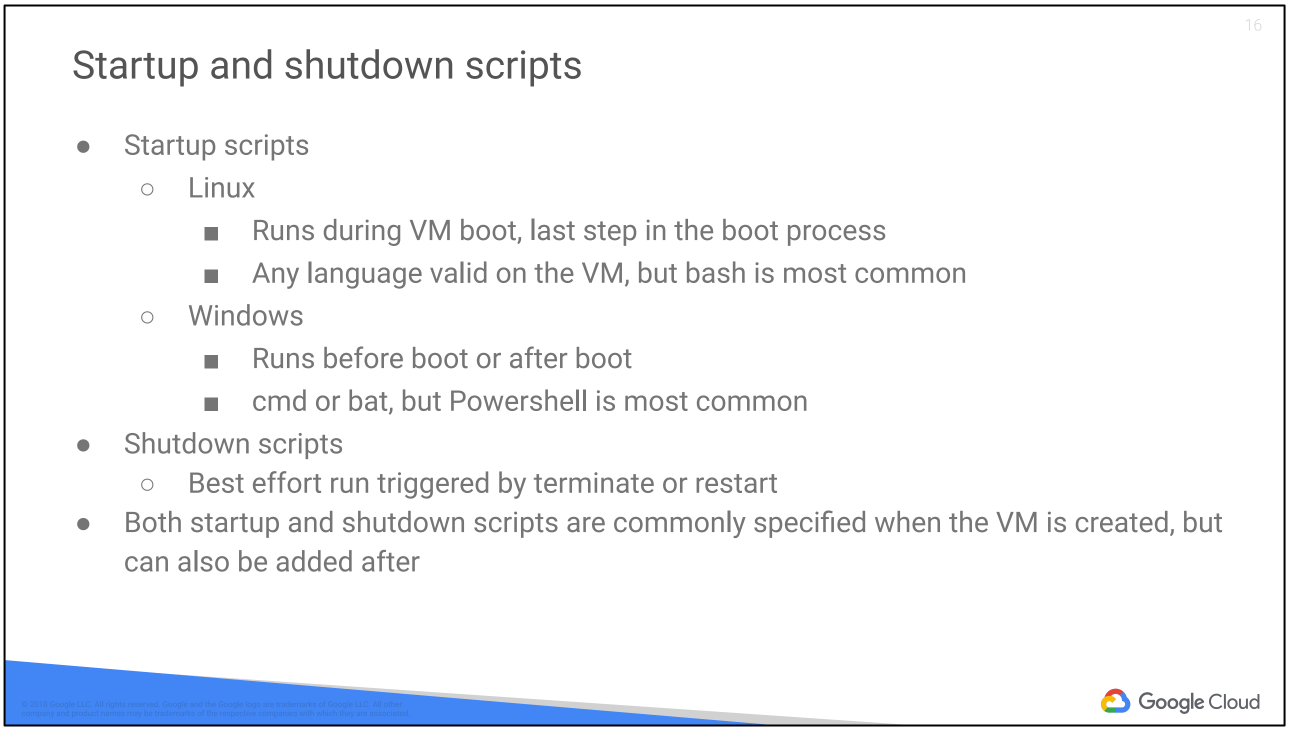


Agenda

- Infrastructure Automation
- Images
- Metadata
- **Scripts**
- Google Cloud API
- Lab
- Quiz

Google Cloud

[Script]
Let's learn about how you can leverage scripts in infrastructure automation.



Startup and shutdown scripts

- Startup scripts
  - Linux
    - Runs during VM boot, last step in the boot process
    - Any language valid on the VM, but bash is most common
  - Windows
    - Runs before boot or after boot
    - cmd or bat, but Powershell is most common
- Shutdown scripts
  - Best effort run triggered by terminate or restart
- Both startup and shutdown scripts are commonly specified when the VM is created, but can also be added after

Google Cloud

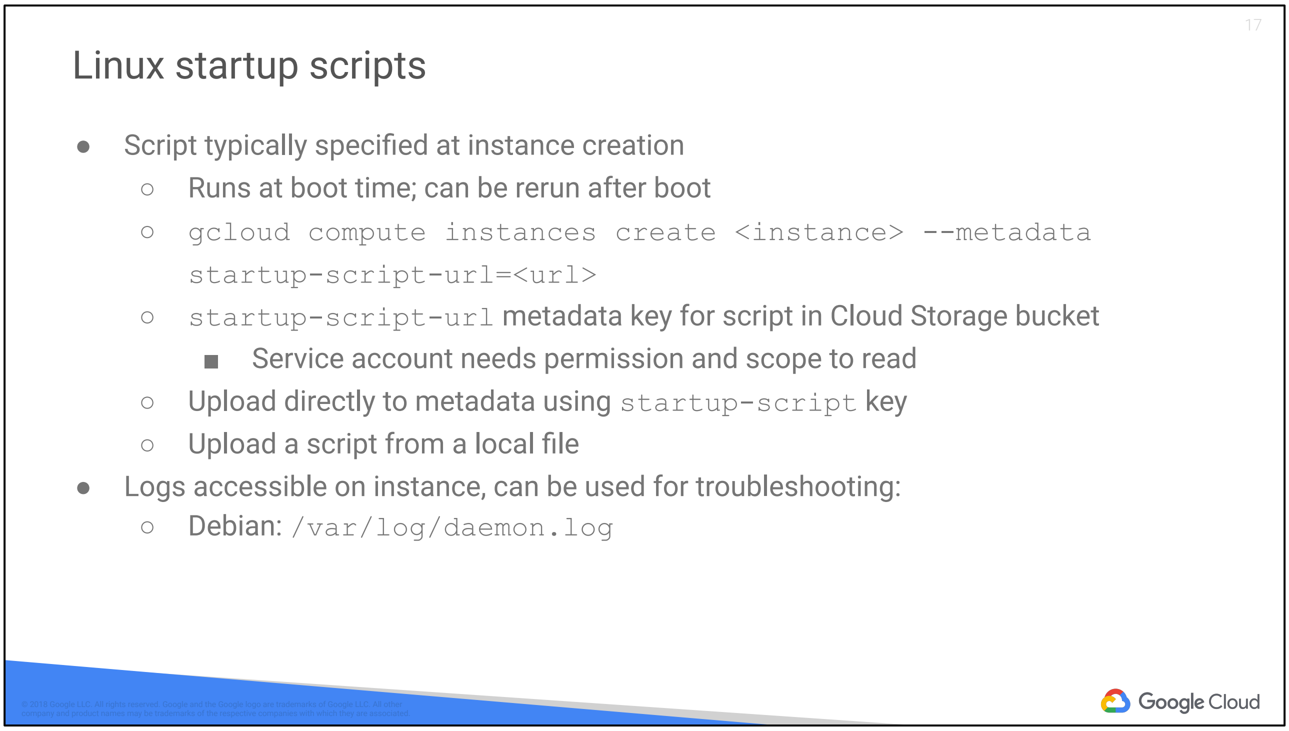
[Script]
VMs start and stop with startup and shutdown scripts. If you're a Linux or Windows admin, you're probably familiar with this process already.

We're going to run these scripts both pre and post start.

It’s important to note that startup scripts always get run. Shutdown scripts are a best effort. If we're shutting down the device, we cannot guarantee that the shutdown script will finish.

You can update these startup and shutdown scripts at any time, even after the virtual machine has been created.

[Student Notes]
https://cloud.google.com/compute/docs/startupscript



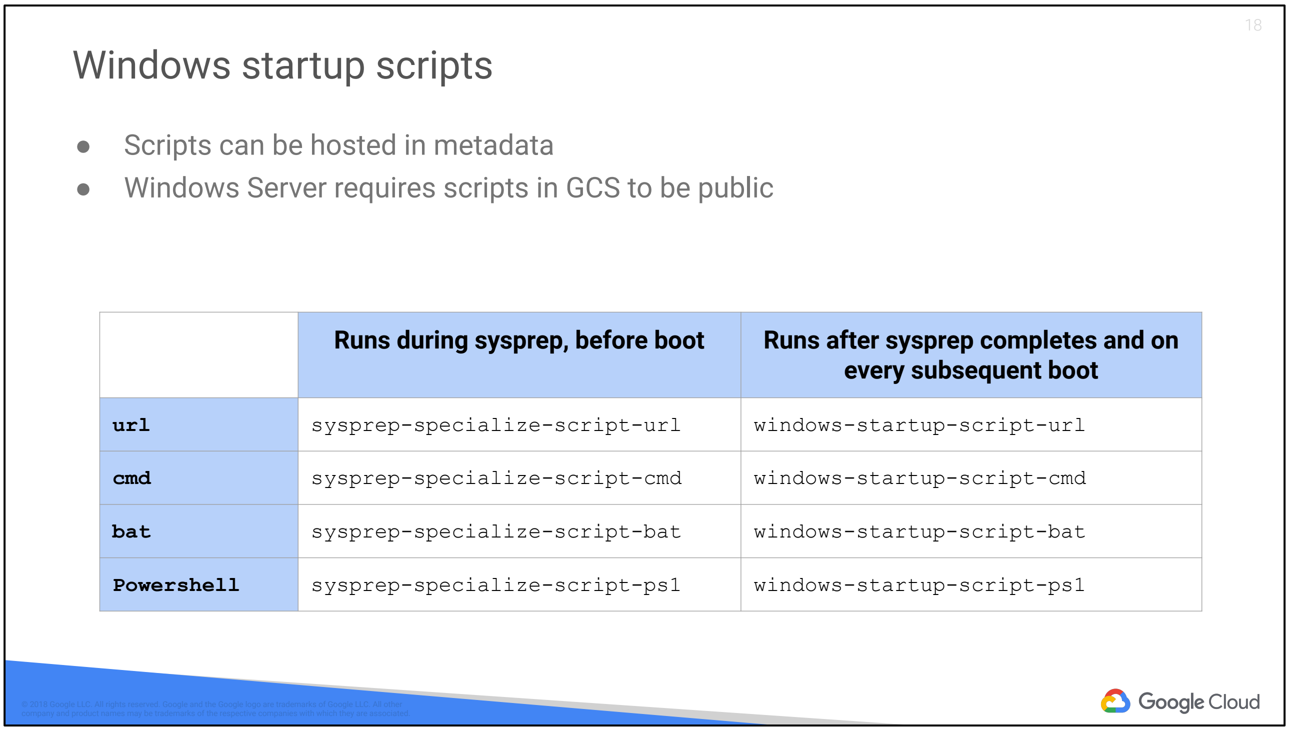
Linux startup scripts

- Script typically specified at instance creation
  - Runs at boot time; can be rerun after boot
  - `gcloud compute instances create <instance> --metadata startup-script-url=<url>`
  - `startup-script-url` metadata key for script in Cloud Storage bucket
    - Service account needs permission and scope to read
  - Upload directly to metadata using `startup-script` key
  - Upload a script from a local file
- Logs accessible on instance, can be used for troubleshooting:
  - Debian: `/var/log/daemon.log`

Google Cloud

[Script]
Here's an example of how you might actually create a startup script. This is a good reference, so feel free to pause the video here.

[Student Notes]
https://cloud.google.com/compute/docs/startupscript#rerunthescript



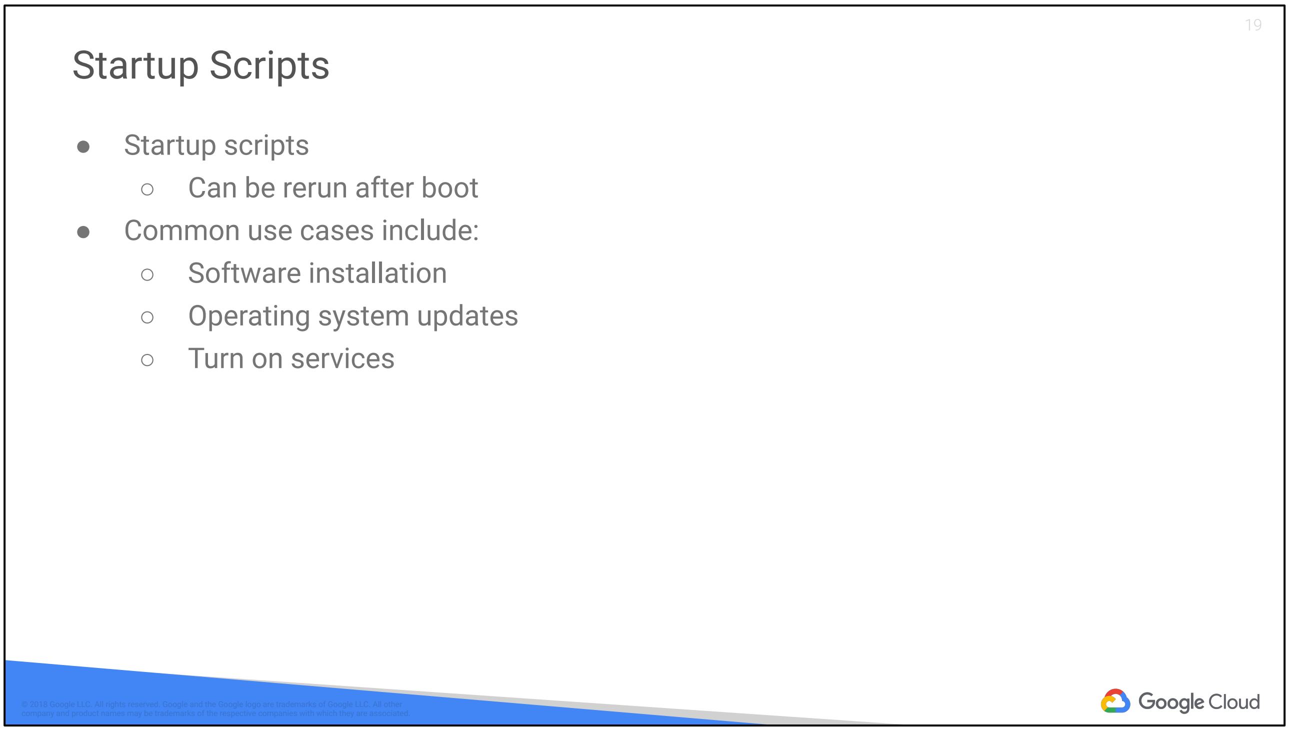
Windows startup scripts

- Scripts can be hosted in metadata
- Windows Server requires scripts in GCS to be public

	Runs during sysprep, before boot	Runs after sysprep completes and on every subsequent boot
`url`	`sysprep-specialize-script-url`	`windows-startup-script-url`
`cmd`	`sysprep-specialize-script-cmd`	`windows-startup-script-cmd`
`bat`	`sysprep-specialize-script-bat`	`windows-startup-script-bat`
`Powershell`	`sysprep-specialize-script-ps1`	`windows-startup-script-ps1`

Google Cloud

[Script]
Windows utilizes sysprep. A major difference is that these startup scripts are going to be hosted in GCS, and they need to be publicly available although the VMs tags will be private.

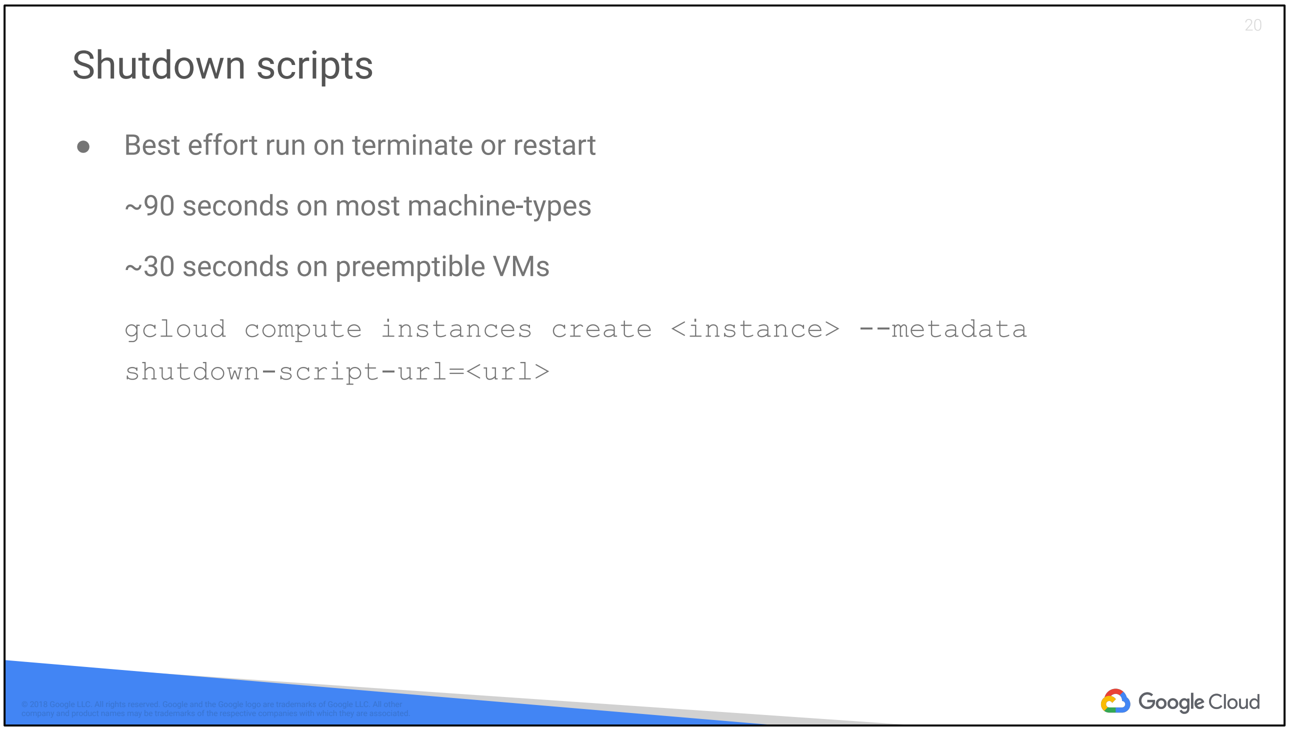


Startup Scripts

- Startup scripts
  - Can be rerun after boot
- Common use cases include:
  - Software installation
  - Operating system updates
  - Turn on services

[Script]
Startup scripts can be rerun after boot and common use cases include: Software installation, operating system updates, and turn on services.

[Student Notes]
For more information on startup scripts, see:
https://cloud.google.com/compute/docs/startupscript



Shutdown scripts

- Best effort run on terminate or restart

  ~90 seconds on most machine-types

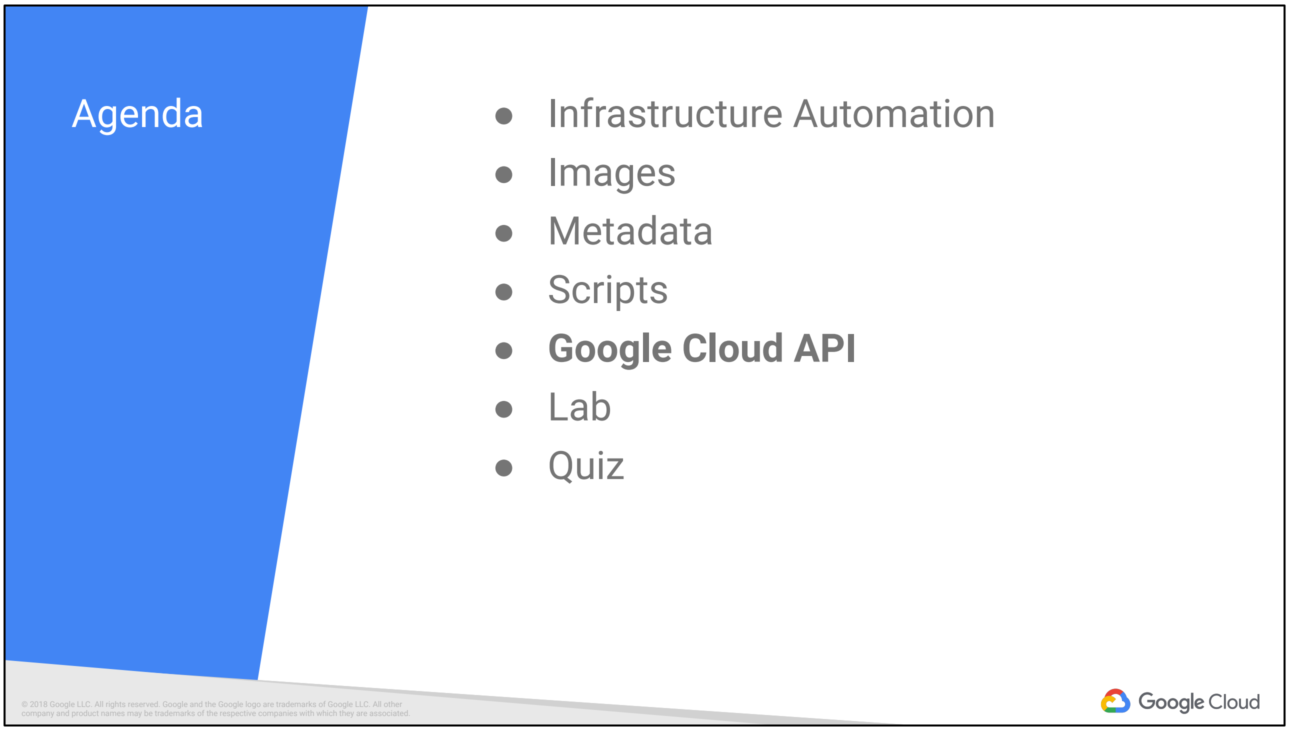
  ~30 seconds on preemptible VMs

```
gcloud compute instances create <instance> --metadata
shutdown-script-url=<url>
```

Google Cloud

[Script]
To reiterate, shutdown scripts are based on best effort. Note that you should give a typical virtual machine 90 seconds for a shutdown even and a preemptible VM 30 seconds.

[Student Notes]
https://cloud.google.com/compute/docs/shutdownscript

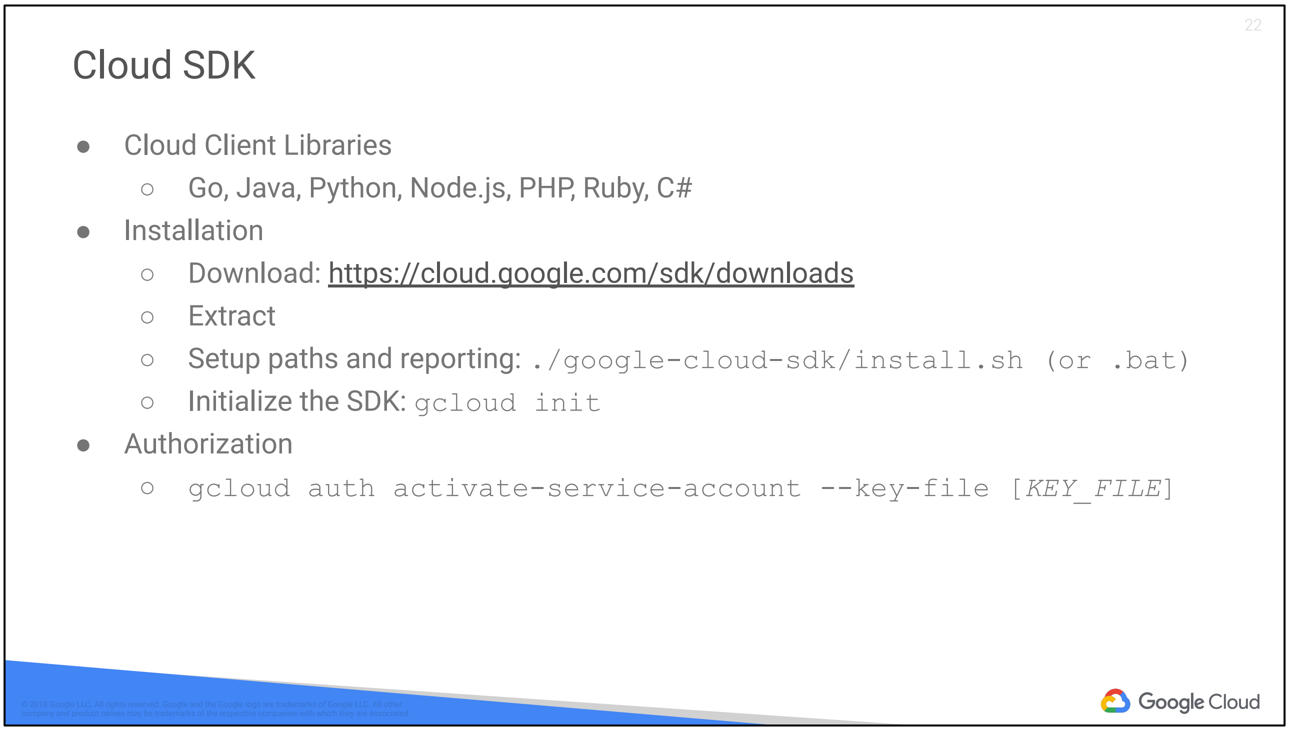


Agenda

- Infrastructure Automation
- Images
- Metadata
- Scripts
- **Google Cloud API**
- Lab
- Quiz

Google Cloud

[Script]
We've actually been working with the Google Cloud Platform API already throughout this course. Let's learn about it.



Cloud SDK

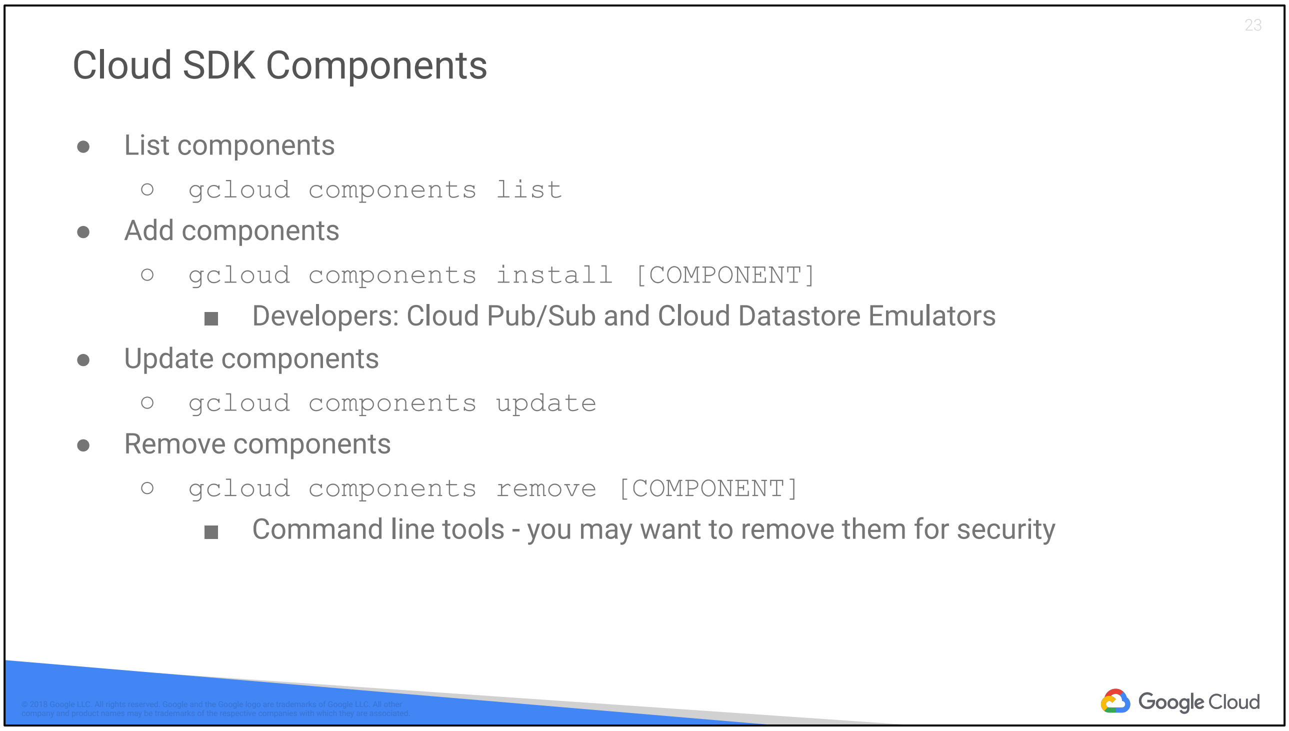
- Cloud Client Libraries
  - Go, Java, Python, Node.js, PHP, Ruby, C#
- Installation
  - Download: https://cloud.google.com/sdk/downloads
  - Extract
  - Setup paths and reporting: `./google-cloud-sdk/install.sh (or .bat)`
  - Initialize the SDK: `gcloud init`
- Authorization
  - `gcloud auth activate-service-account --key-file [KEY_FILE]`

Google Cloud

[Script]
You can download the cloud SDK and create your own automation tools. *[point to listed languages]* Client libraries are available for these languages. The SDK is easy to install, just click on this link *[point to link]* and run the install command. If you're logged in as a user you just type gcloud init, that's going to give you a custom URL which you will paste into your browser, then you log in as that individual user so that will actually save a secure token and then you authenticate on behalf of that user. But you may have to authorize as a service account. Here is the command to activate a service account and supply the supported key file that was generated when you generated the service account.

[Student Notes]
Google API Libraries: https://developers.google.com/products/
Technically, the Cloud SDK is one library among the *many* libraries associated with Google products, including Maps, gMail, Drive, and so forth.

https://cloud.google.com/sdk/docs/initializing



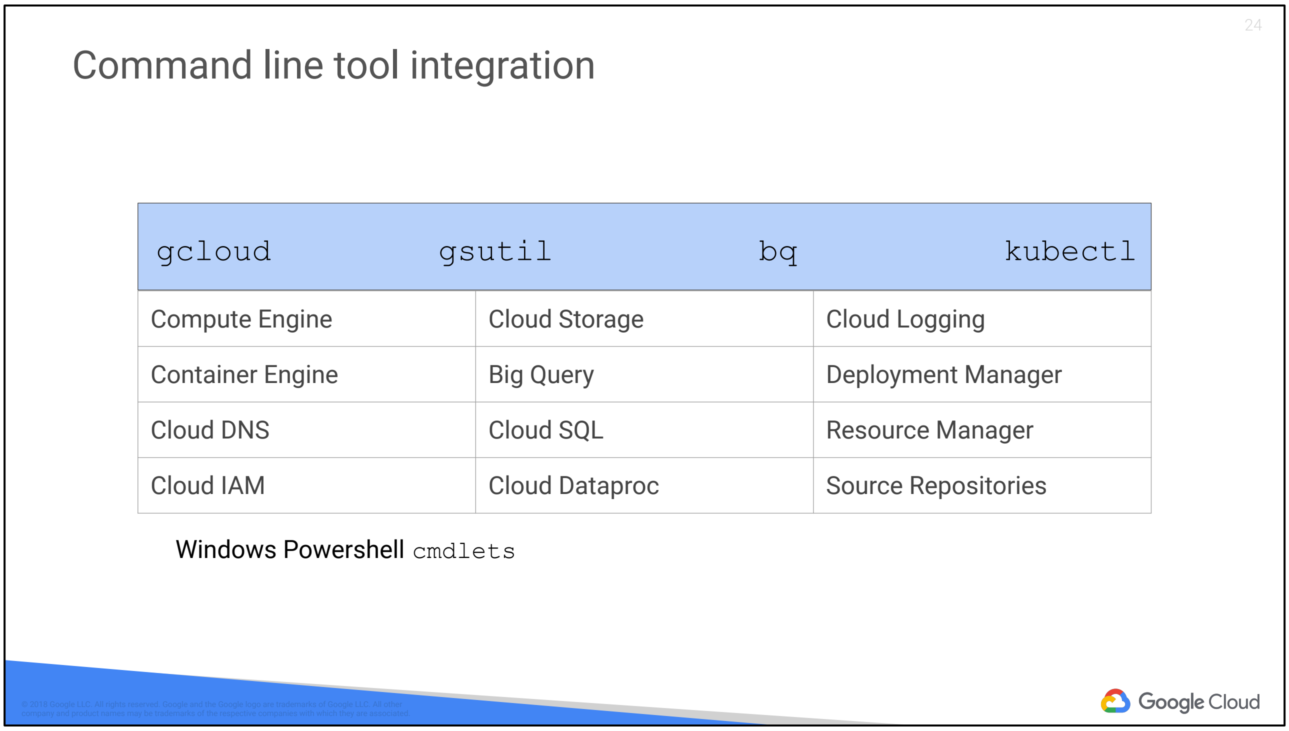
Cloud SDK Components

- List components
  - `gcloud components list`
- Add components
  - `gcloud components install [COMPONENT]`
    - Developers: Cloud Pub/Sub and Cloud Datastore Emulators
- Update components
  - `gcloud components update`
- Remove components
  - `gcloud components remove [COMPONENT]`
    - Command line tools - you may want to remove them for security

Google Cloud

[Script]
Here we see a few examples of how to use gcloud to list, add, update, and remove components from the command line.

[Student Notes]
The command line tools are components. So are all the Beta commands. For developers, there is a Cloud Pub/Sub Emulator and a Cloud Datastore Emulator.



Command line tool integration

`gcloud` `gsutil` `bq` `kubectl`

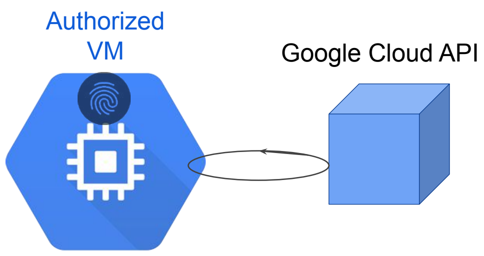
Compute Engine	Cloud Storage	Cloud Logging
Container Engine	Big Query	Deployment Manager
Cloud DNS	Cloud SQL	Resource Manager
Cloud IAM	Cloud Dataproc	Source Repositories

Windows Powershell `cmdlets`

[Script]
Most of the two labs we've completed use gcloud and gsutil. There are two other command line utilities that take advantage of the cloud SDK, and one is called bq, which is is a python-based tool that accesses BigQuery from the command line. The other one is kubectl, which allows you to manage Kubernetes from the command line.

[Student Notes]
This is the top of the gcloud reference documentation. Most functions are in the gcloud tool. Only, storage (gsutil), Big Query (bq), and Kubernetes (kubectl) are separate.
https://cloud.google.com/sdk/gcloud/reference/

VM as an Automation Engine



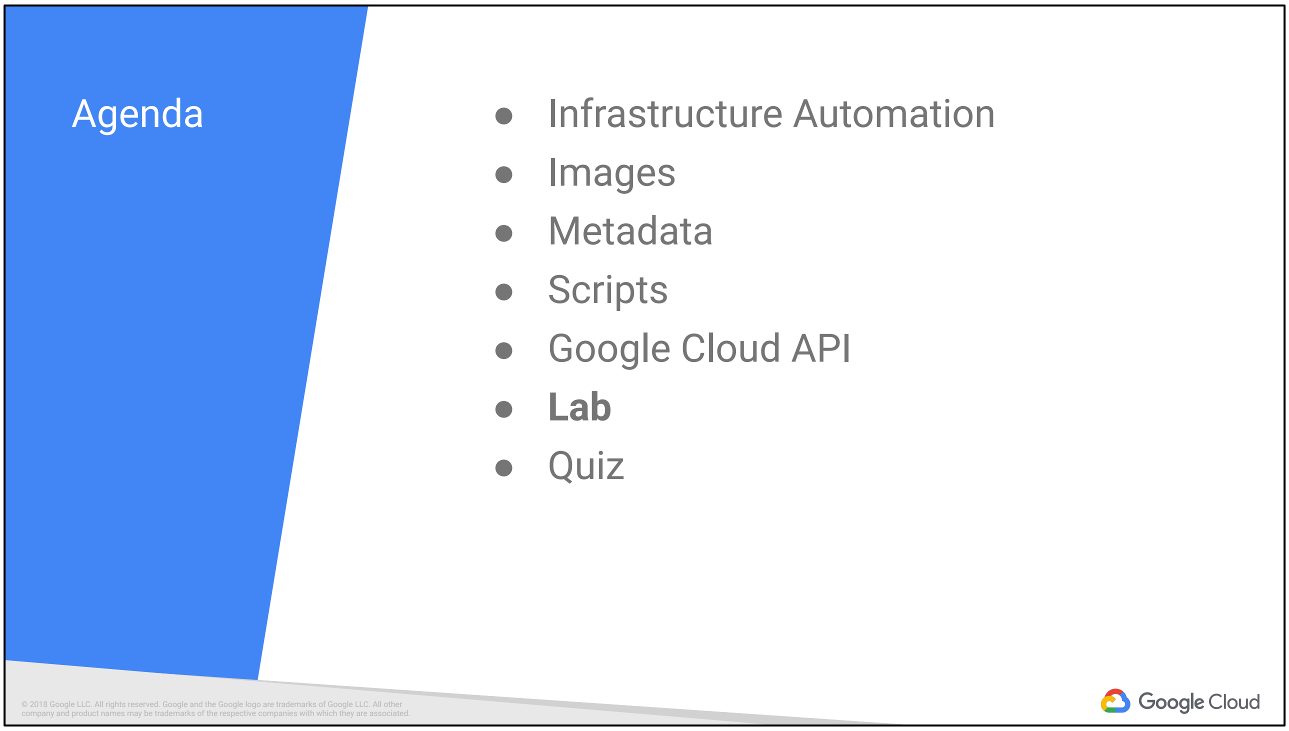
- A VM is authorized with a service account to use the Google Cloud API
- Software on the VM uses the API to create infrastructure on your behalf

[Script]
Using a VM to create infrastructure can be a powerful tool. You can accomplish this by authorizing a VM with a service account to use the Google Cloud API, and then the software on the VM uses the API to create infrastructure on your behalf. Without having to code the solution, you will be learning how to setup and operate such an infrastructure automation tool in the next lab.

[Student Notes]
This isn't a coding class. Developer Track classes cover application development including interaction with the Cloud API.

However, using a VM to create infrastructure is a very powerful tool.

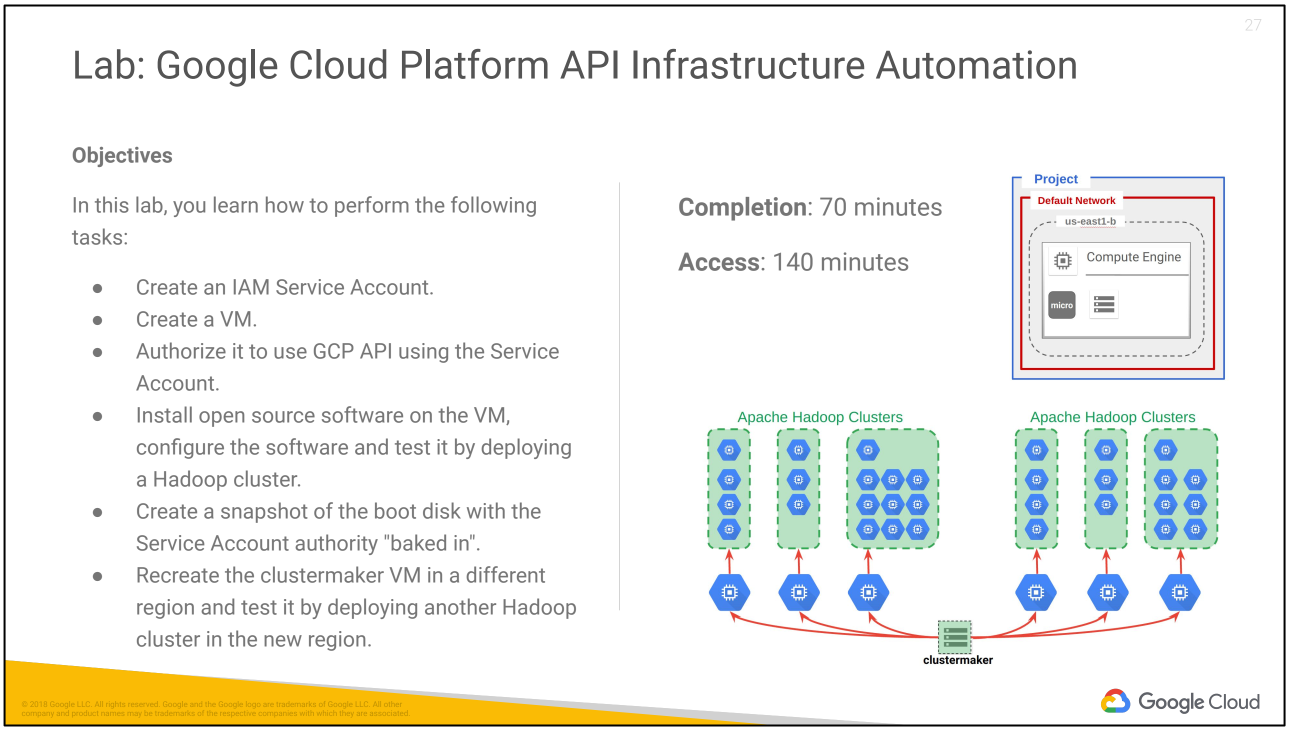
So… without having you code… you will be learning how to setup and operate such an infrastructure automation tool in the lab.



Agenda

- Infrastructure Automation
- Images
- Metadata
- Scripts
- Google Cloud API
- **Lab**
- Quiz

Google Cloud



Lab: Google Cloud Platform API Infrastructure Automation

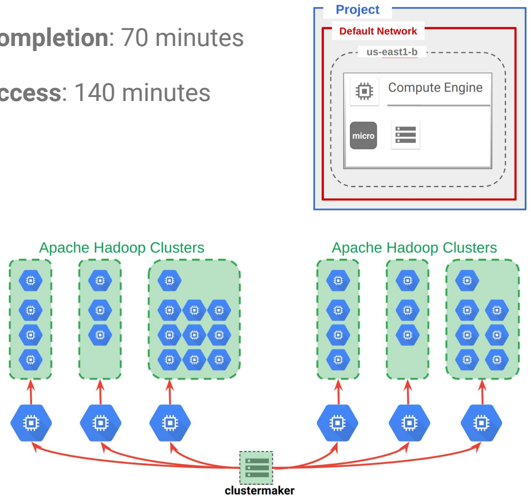
Objectives

In this lab, you learn how to perform the following tasks:

- Create an IAM Service Account.
- Create a VM.
- Authorize it to use GCP API using the Service Account.
- Install open source software on the VM, configure the software and test it by deploying a Hadoop cluster.
- Create a snapshot of the boot disk with the Service Account authority "baked in".
- Recreate the clustermaker VM in a different region and test it by deploying another Hadoop cluster in the new region.

Completion: 70 minutes

Access: 140 minutes



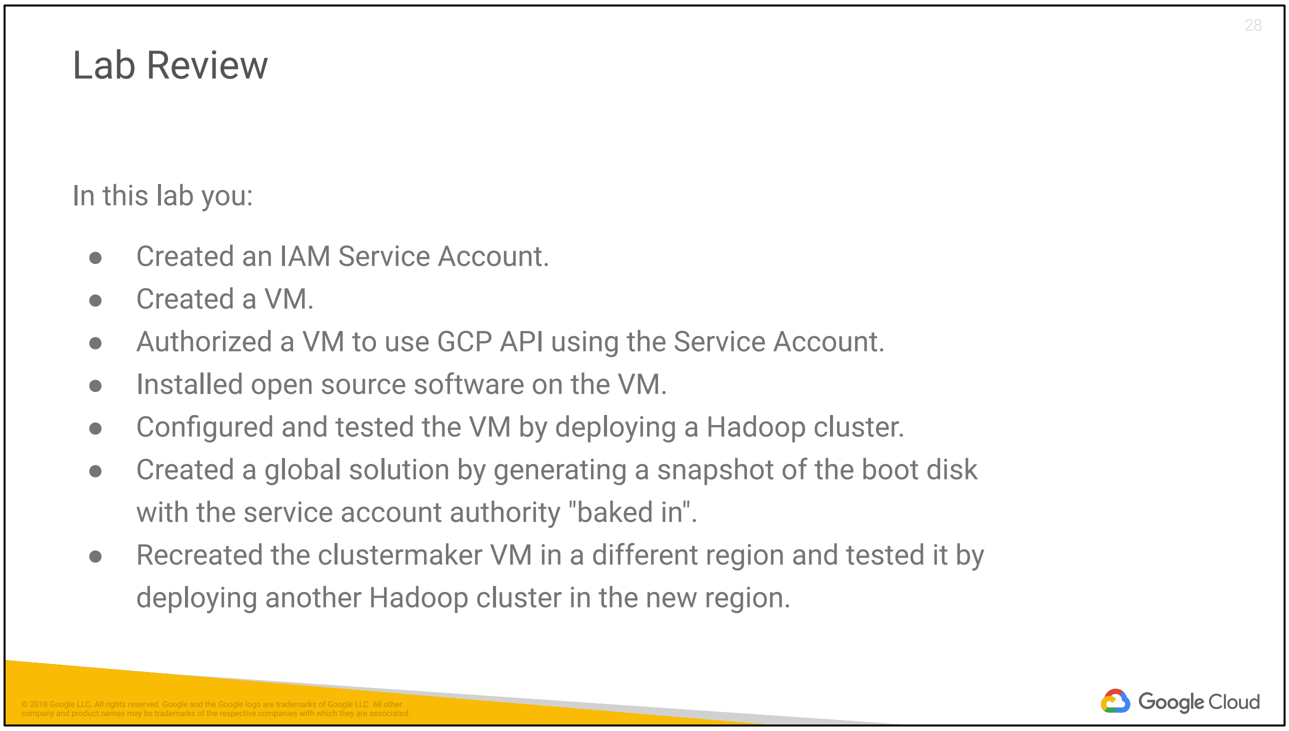
Google Cloud

[Script]
Let's take what we've learned about infrastructure automation so far and apply it in a lab.

In this lab, you will:

- Create an IAM Service Account.
- Create a VM.
- Authorize it to use GCP API using the Service Account.
- Install open source software on the VM, configure the software and test it by deploying a Hadoop cluster.
- Create a snapshot of the boot disk with the Service Account authority "baked in".
- Recreate the clustermaker VM in a different region and test it by deploying another Hadoop cluster in the new region.

The lab will take about 70 minutes to complete, but you have 140 minutes before it times out.



Lab Review

In this lab you:

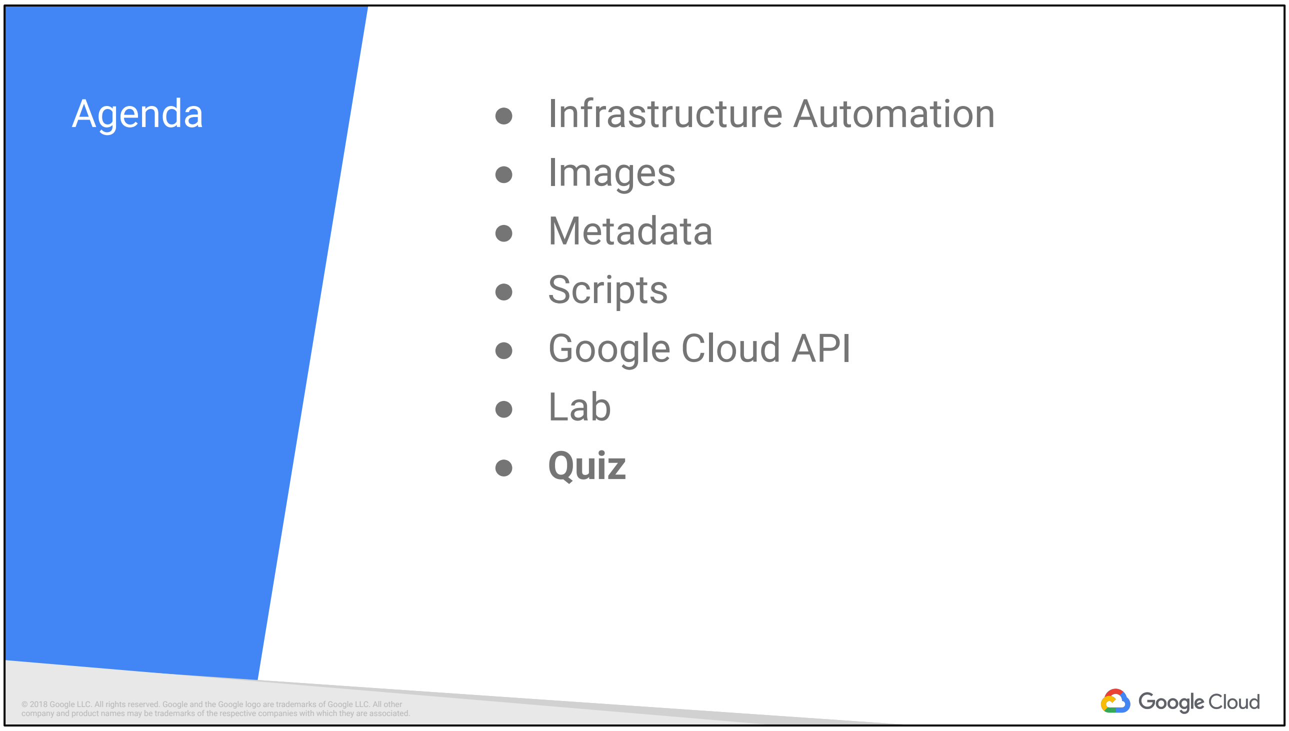
- Created an IAM Service Account.
- Created a VM.
- Authorized a VM to use GCP API using the Service Account.
- Installed open source software on the VM.
- Configured and tested the VM by deploying a Hadoop cluster.
- Created a global solution by generating a snapshot of the boot disk with the service account authority "baked in".
- Recreated the clustermaker VM in a different region and tested it by deploying another Hadoop cluster in the new region.

Google Cloud

[Script]
In this lab you:

- Created an IAM Service Account.
- Created a VM.
- Authorized a VM to use GCP API using the Service Account.
- Installed open source software on the VM.
- Configured and tested the VM by deploying a Hadoop cluster.
- Created a global solution by generating a snapshot of the boot disk with the service account authority "baked in".
- Recreated the clustermaker VM in a different region and tested it by deploying another Hadoop cluster in the new region.

I hope you enjoyed applying the concepts learned in this module in a real-world application.



Agenda

- Infrastructure Automation
- Images
- Metadata
- Scripts
- Google Cloud API
- Lab
- **Quiz**

Google Cloud

More...

Startup scripts

- https://cloud.google.com/compute/docs/startupscript

Shutdown scripts

- https://cloud.google.com/compute/docs/shutdownscript

Storing and retrieving metadata

- https://cloud.google.com/compute/docs/storing-retrieving-metadata

Google Cloud

Google Cloud