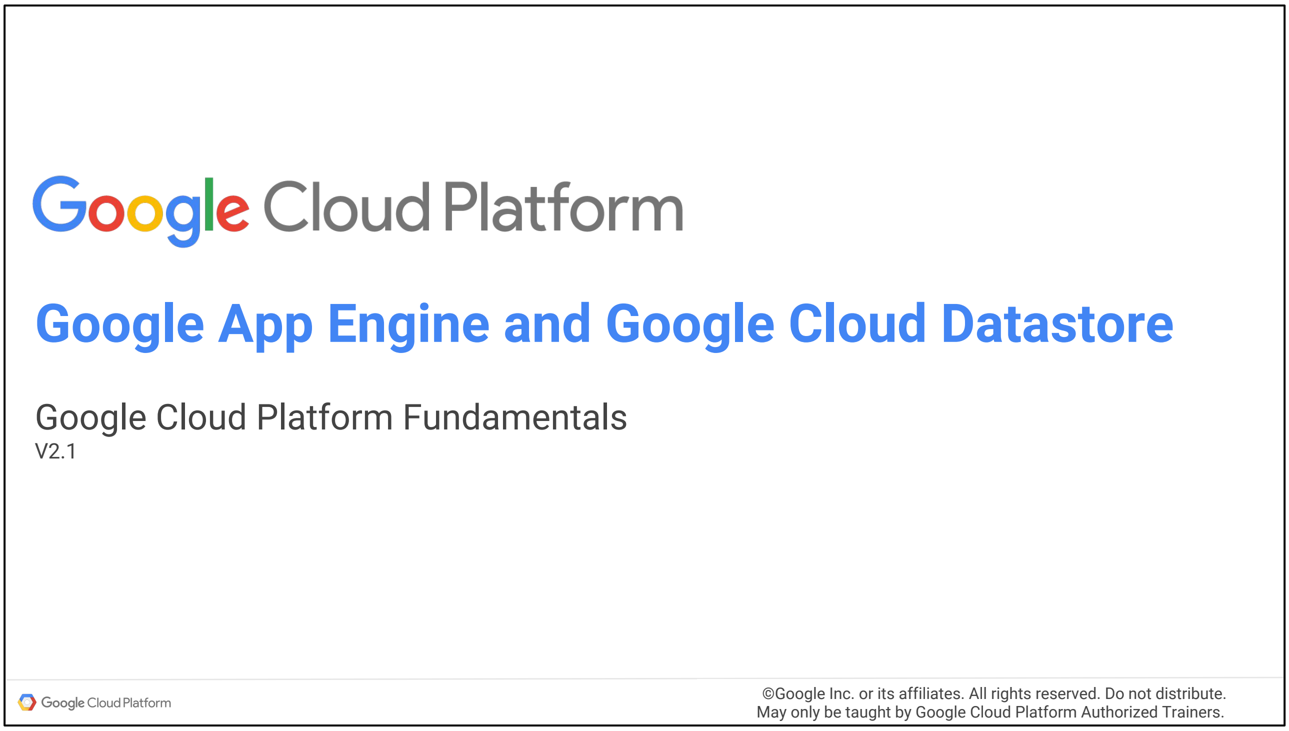


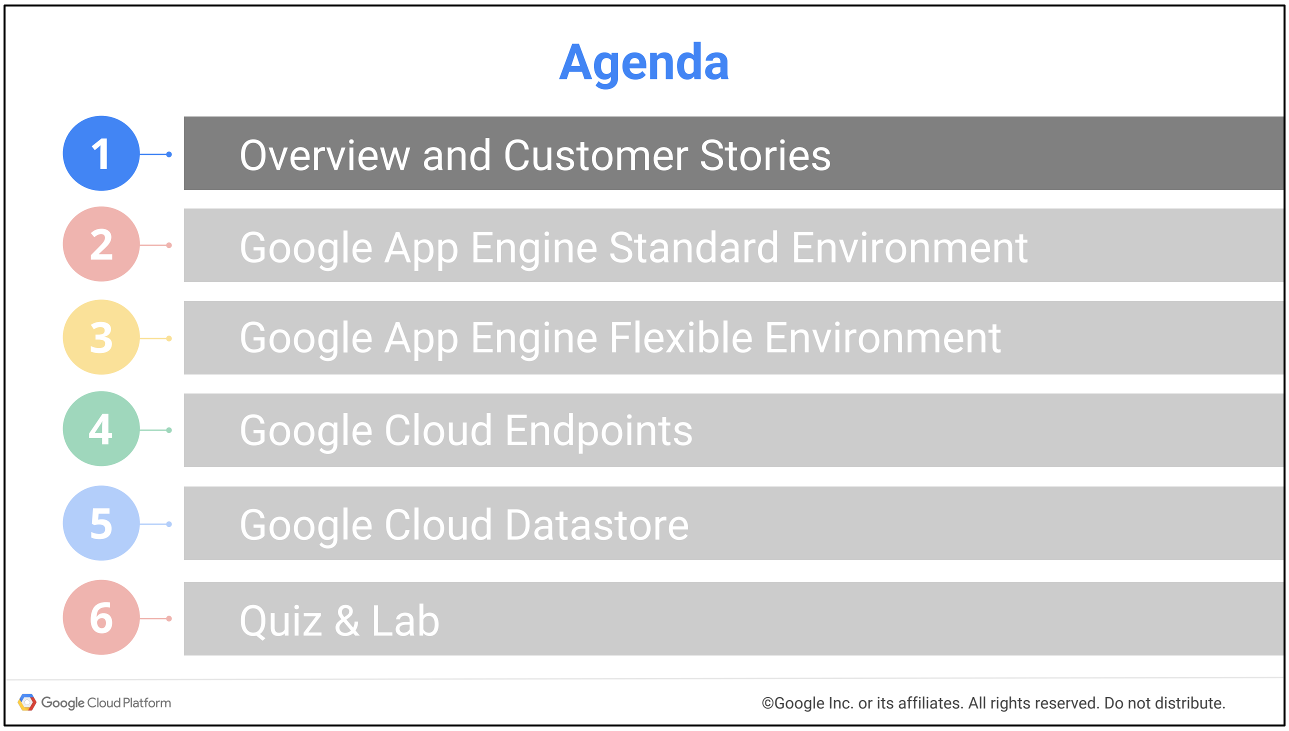
Google Cloud Platform

Google App Engine and Google Cloud Datastore

Google Cloud Platform Fundamentals
V2.1

Google Cloud Platform

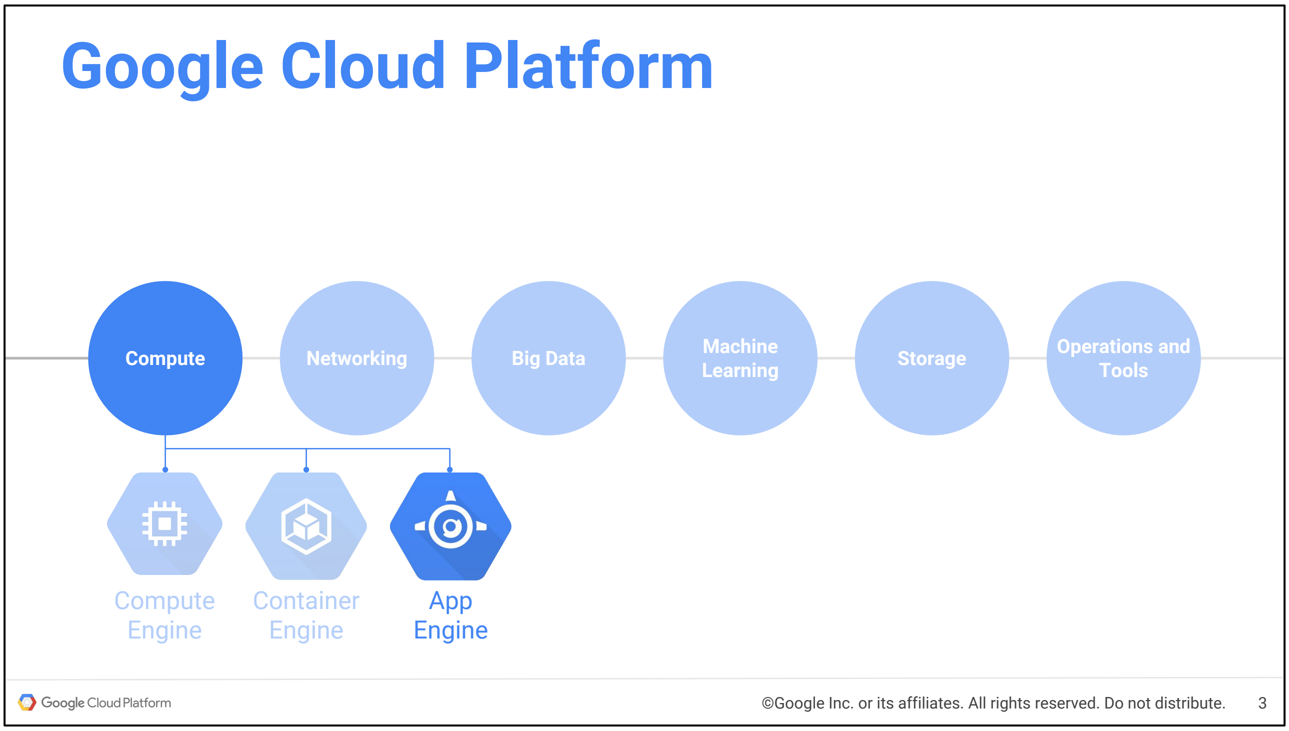
Timing: Approximately 30 minutes



Agenda

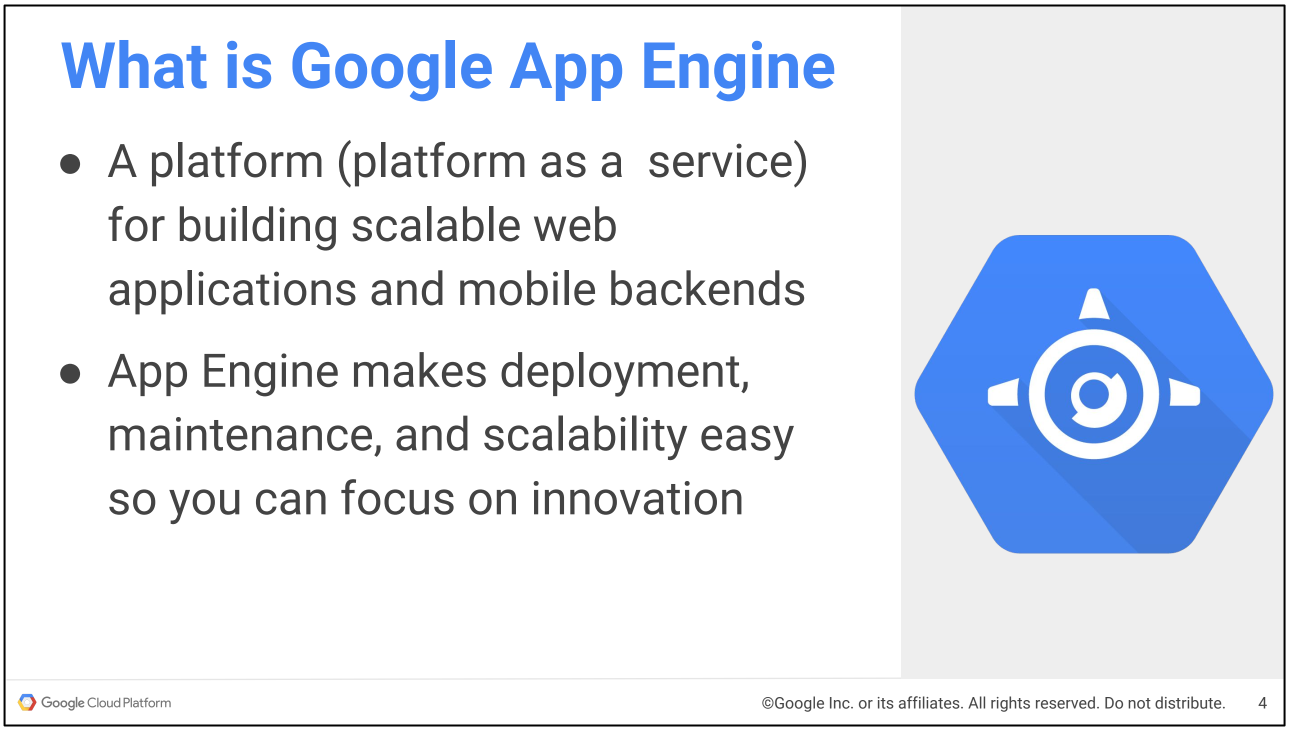
1. Overview and Customer Stories
2. Google App Engine Standard Environment
3. Google App Engine Flexible Environment
4. Google Cloud Endpoints
5. Google Cloud Datastore
6. Quiz & Lab

Google Cloud Platform



Google Cloud Platform

Notes:
App Engine is one of several Google Cloud Platform compute options for running your applications.



What is Google App Engine

- A platform (platform as a service) for building scalable web applications and mobile backends
- App Engine makes deployment, maintenance, and scalability easy so you can focus on innovation

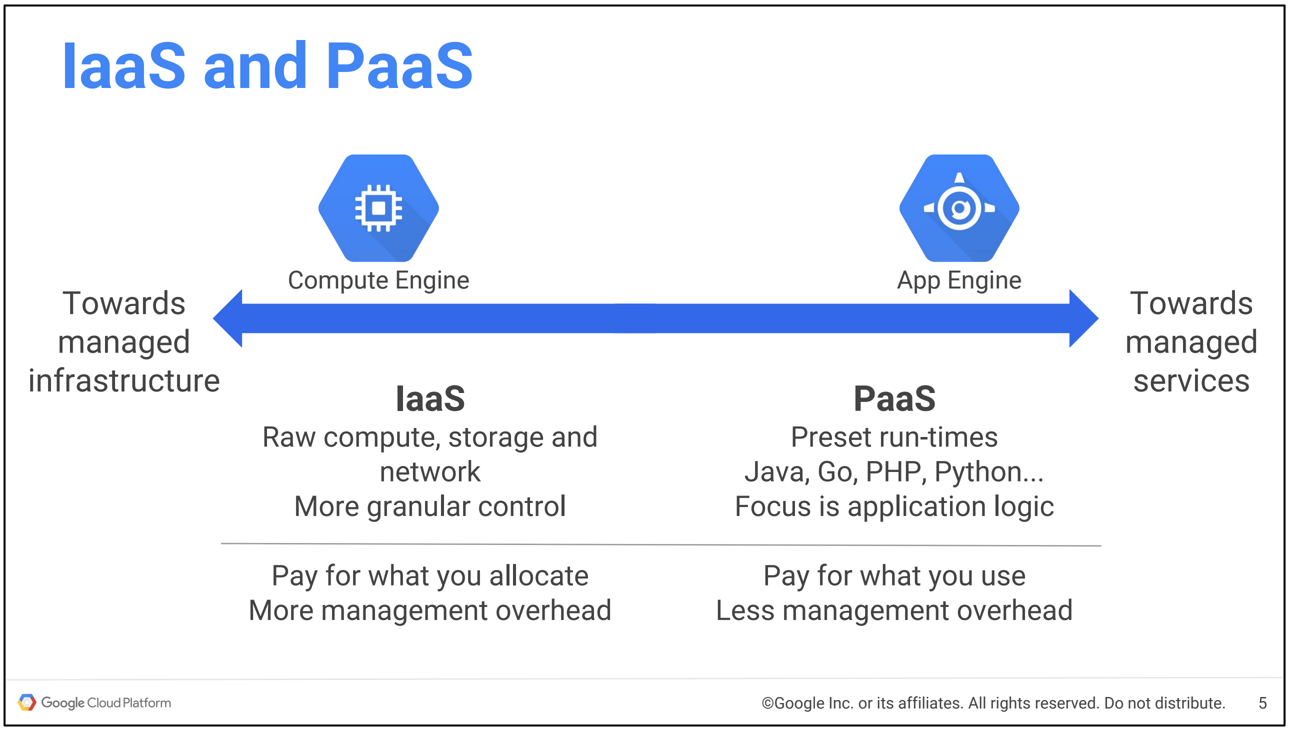
Google Cloud Platform

Notes:
Google App Engine is a platform for building scalable web applications and mobile backends. App Engine provides you with built-in services and APIs such as NoSQL datastores, memcache, load balancing, health checks, application logging, and a user authentication API, common to most applications.

App Engine will scale your application automatically in response to the amount of traffic it receives so you only pay for the resources you use. Just upload your code and Google will manage your app's availability. There are no servers for you to provision or maintain.

Security Scanner automatically scans and detects common web application vulnerabilities. It enables early threat identification and delivers very low false positive rates. You can easily setup, run, schedule, and manage security scans from the Google Cloud Platform Console.

App Engine works with popular development tools such as Eclipse, IntelliJ, Maven, Git, Jenkins, and PyCharm. You can build your apps with the tools you love without changing your workflow.



IaaS and PaaS

Compute Engine
App Engine
Towards managed infrastructure
Towards managed services

IaaS
Raw compute, storage and network
More granular control

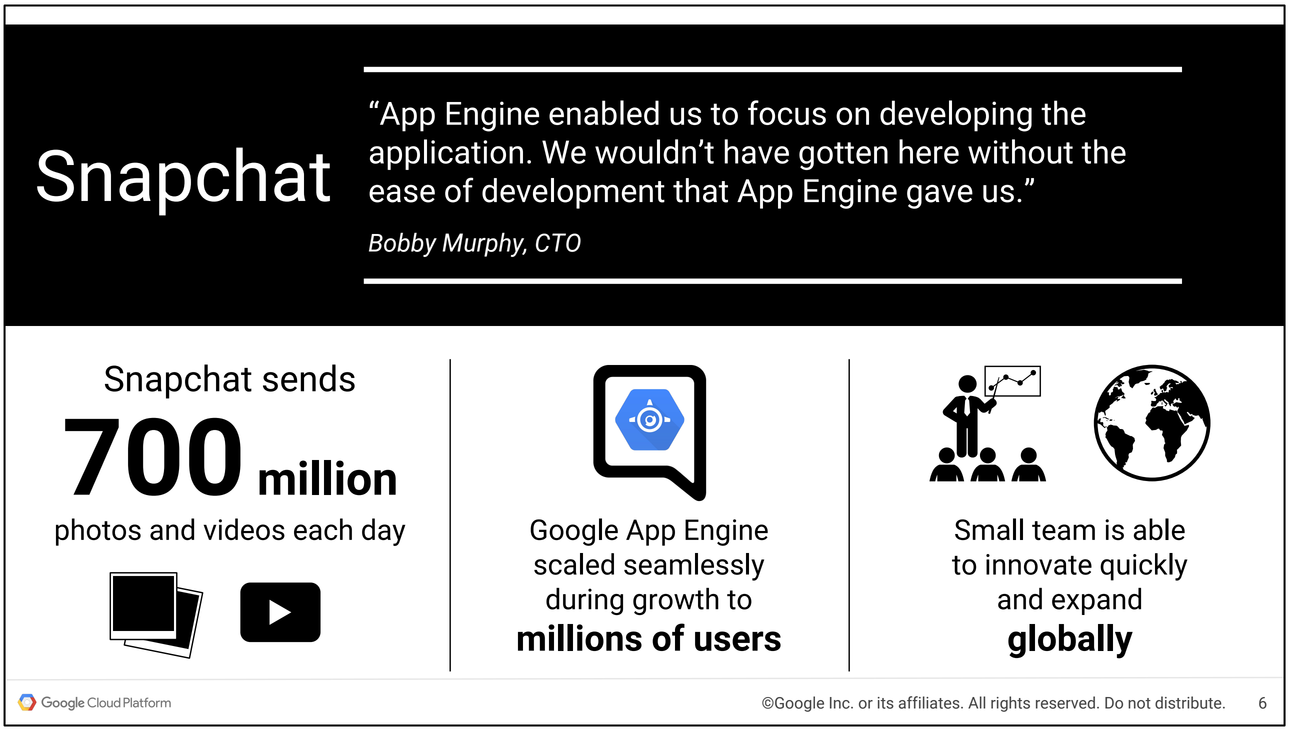
Pay for what you allocate
More management overhead

PaaS
Preset run-times
Java, Go, PHP, Python...
Focus is application logic

Pay for what you use
Less management overhead

Google Cloud Platform

Notes:
App Engine is a platform as a service option for running applications on Google Cloud Platform. Platform as a service allows you to concentrate on innovating your applications by managing the application infrastructure for you. For example, App Engine manages the hardware and networking infrastructure required to run your code.



Snapchat

"App Engine enabled us to focus on developing the application. We wouldn't have gotten here without the ease of development that App Engine gave us."

Bobby Murphy, CTO

Snapchat sends **700 million** photos and videos each day

Google App Engine scaled seamlessly during growth to **millions of users**

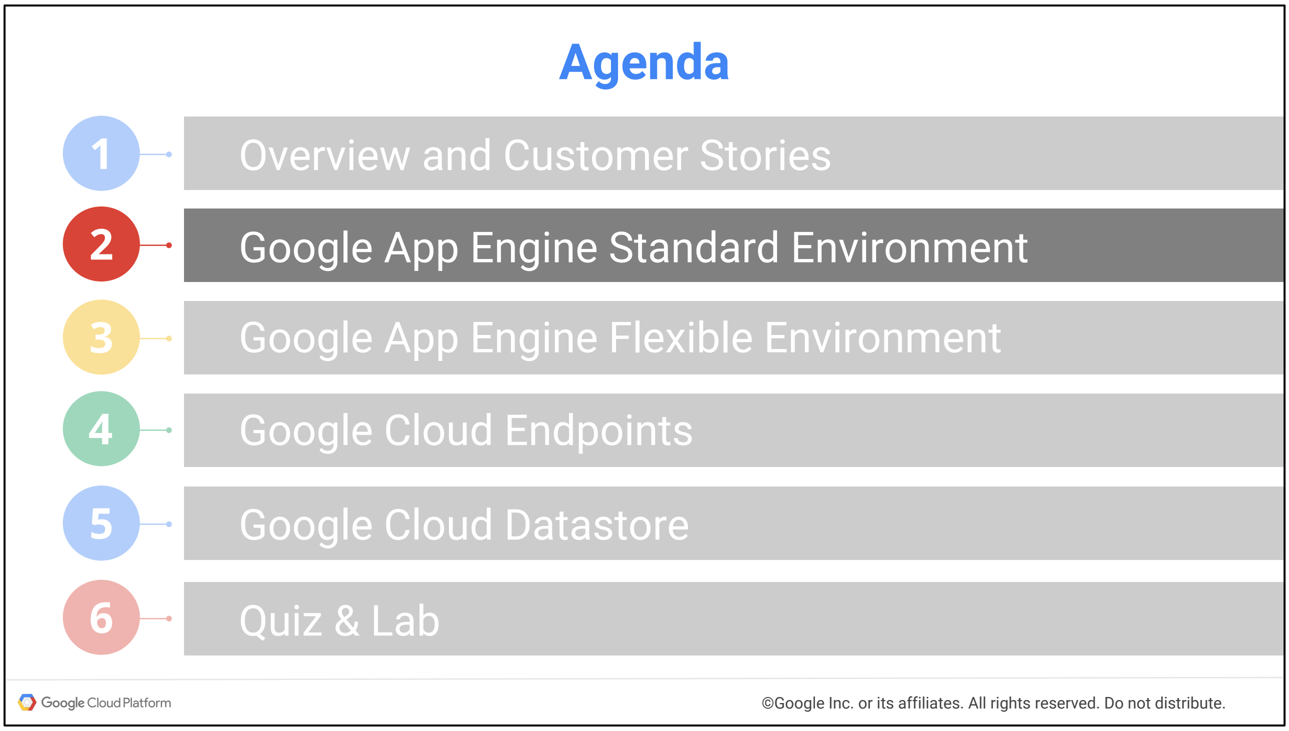
Small team is able to innovate quickly and expand **globally**

Google Cloud Platform

Notes:

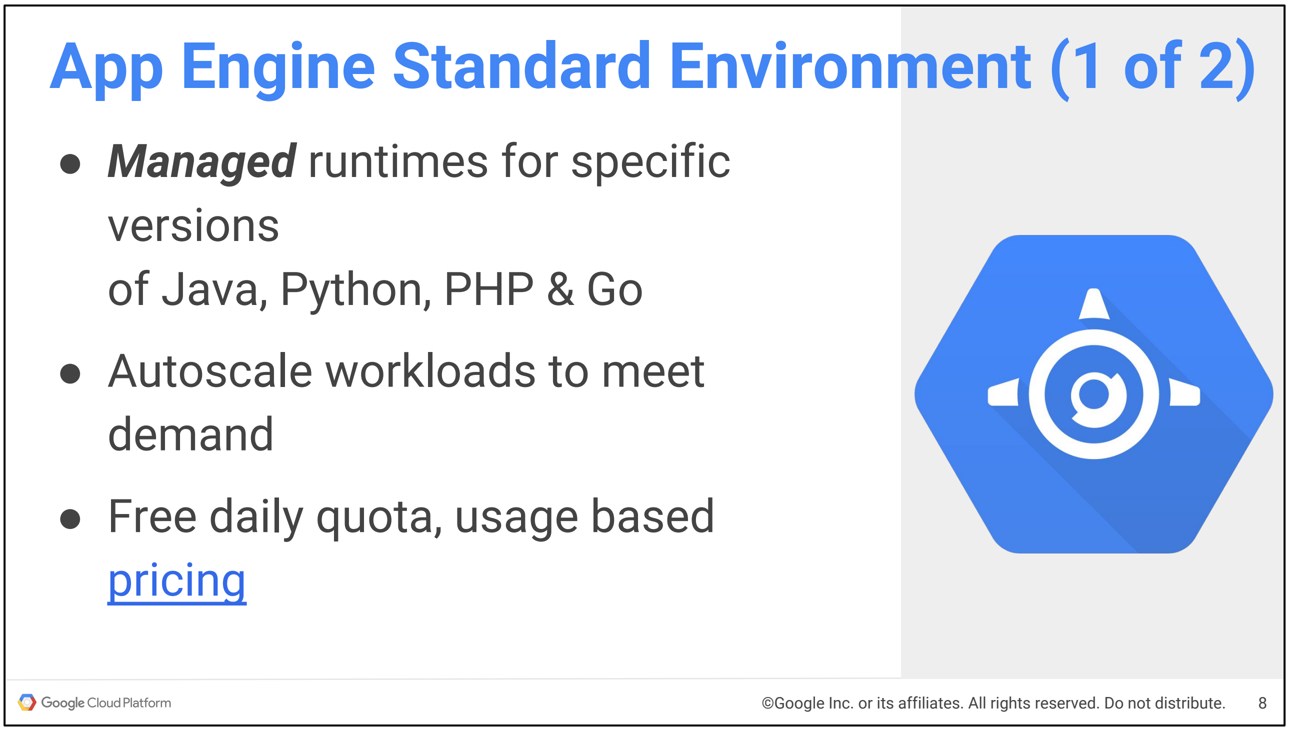
- Today, Snapchat's photo messaging app is among the top 10 most downloaded mobile apps of any type for both Android and iOS.
- They launched in 2011, and just two years later, had tens of millions of users across the world sending hundreds of millions of photos and videos each day.
- Before the era of Cloud computing, supporting this level of growth with a developer and operations team of just a few people would have been impossible.
- With Google App Engine, Snapchat doesn't have to worry about managing the underlying infrastructure. Cloud Platform allows their app to scale with minimal effort from their team.
- But it's not just about scale. App Engine also allows Snapchat to quickly release new features or versions of the application - a difficult task when your app is constantly in use by millions of people.

Read more about Snapchat here: https://cloudplatform.googleblog.com/2016/03/Snapchat-shares-security-best-practices-for-running-on-GCP-practices.html.



Agenda

1. Overview and Customer Stories
2. Google App Engine Standard Environment
3. Google App Engine Flexible Environment
4. Google Cloud Endpoints
5. Google Cloud Datastore
6. Quiz & Lab

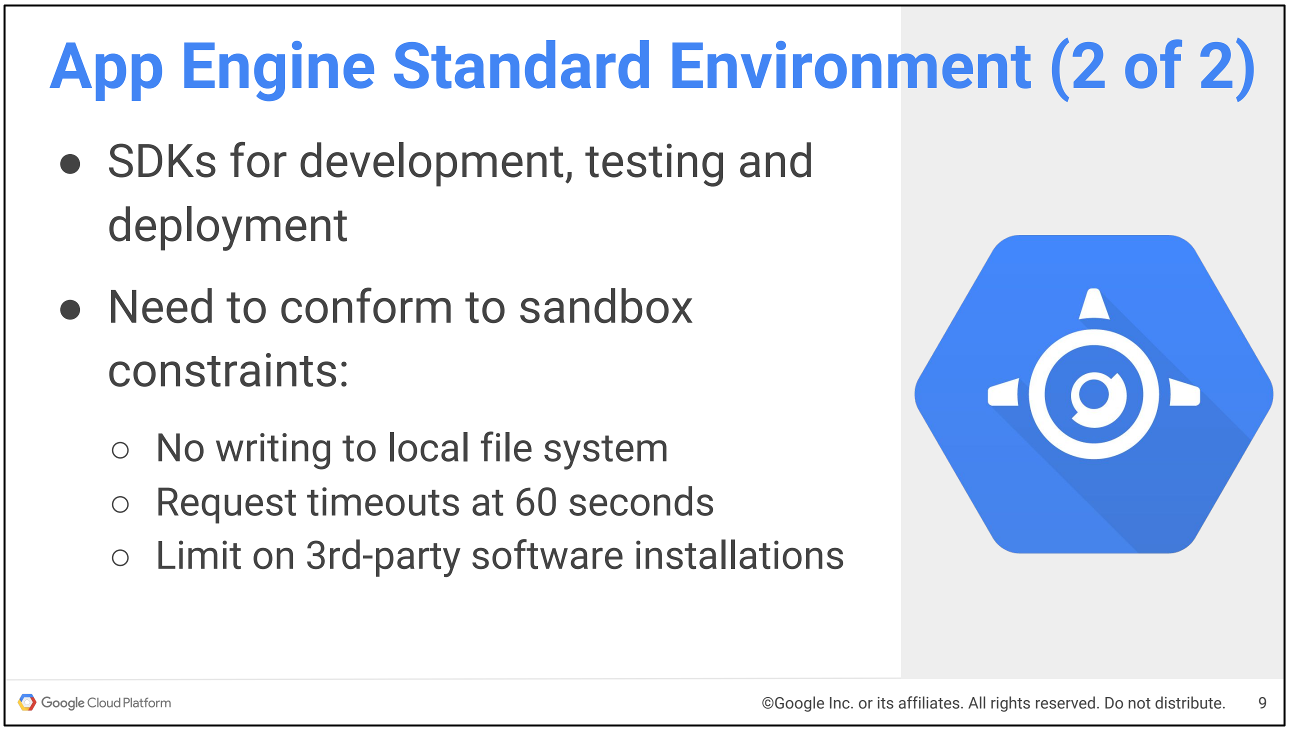


App Engine Standard Environment (1 of 2)

- ***Managed*** runtimes for specific versions
  of Java, Python, PHP & Go
- Autoscale workloads to meet demand
- Free daily quota, usage based pricing

Notes:
The App Engine Standard environment is based on container instances running on Google's infrastructure. Containers are preconfigured with one of several available runtimes (Java 7, Python 2.7, Go and PHP). Each runtime also includes libraries that support App Engine Standard APIs. For many applications, the Standard environment runtimes and libraries may be all you need.

The App Engine Standard environment makes it easy to build and deploy an application that runs reliably even under heavy load and with large amounts of data. It includes the following features:
- Persistent storage with queries, sorting, and transactions
- Automatic scaling and load balancing
- Asynchronous task queues for performing work outside the scope of a request
- Scheduled tasks for triggering events at specified times or regular intervals
- Integration with other Google cloud services and APIs



App Engine Standard Environment (2 of 2)

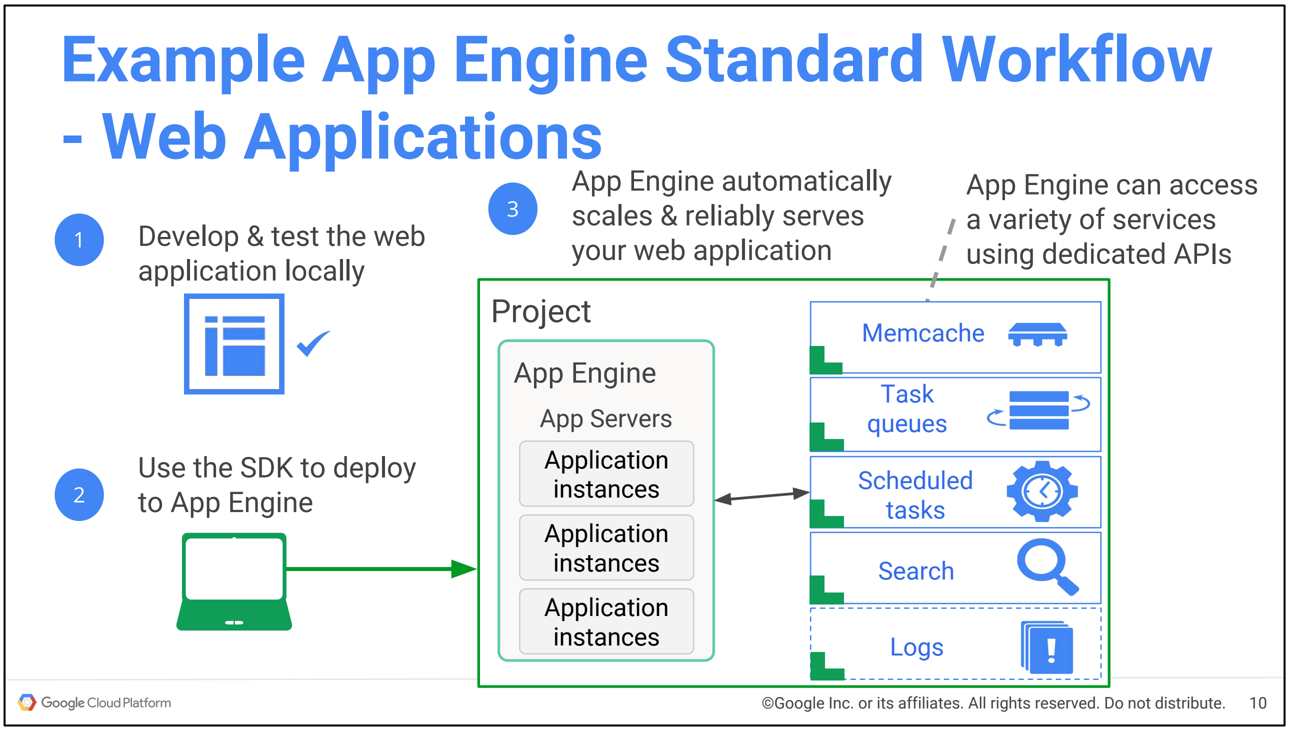
- SDKs for development, testing and deployment
- Need to conform to sandbox constraints:
  - No writing to local file system
  - Request timeouts at 60 seconds
  - Limit on 3rd-party software installations

Notes:
Software Development Kits (SDKs) for App Engine are available in all supported languages. Each SDK includes:

- All of the APIs and libraries available to App Engine
- A simulated, secure sandbox environment, that emulates all of the App Engine services on your local computer
- Deployment tools that allow you to upload your application to the cloud and manage different versions of your application

The SDK manages your application locally, while the Google Cloud Platform Console manages your application in production. The Cloud Platform Console uses a web-based interface to create new applications, configure domain names, change which version of your application is live, examine access and error logs, and much more.

Applications run in a secure, sandboxed environment, allowing App Engine Standard environment to distribute requests across multiple servers, and scaling servers to meet traffic demands. Your application runs within its own secure, reliable environment that is independent of the hardware, operating system, or physical location of the server.



Example App Engine Standard Workflow - Web Applications

1
Develop & test the web application locally
2
Use the SDK to deploy to App Engine
3
App Engine automatically scales & reliably serves your web application
App Engine can access a variety of services using dedicated APIs
Project
App Engine
App Servers
Application instances
Application instances
Application instances
Memcache
Task queues
Scheduled tasks
Search
Logs

Notes:
There are many services available to your applications running in the App Engine Standard environment. For the most up-to-date information on App Engine services, see: https://cloud.google.com/appengine/docs/about-the-standard-environment#index_of_features.

Below are details on some of the more commonly used services available to your App Engine Standard applications.

Users API
App Engine Standard environment applications can authenticate users using Google Accounts or accounts on your own Google Apps domains. An application can detect whether the current user has signed in, and can redirect the user to the appropriate sign-in page to sign in or, if your app uses Google Accounts authentication, create a new account. While a user is signed in to the application, the app can access the user's email address. The app can also detect whether the current user is an administrator, making it easy to implement admin-only areas of the app.

Modules API
Modules are used to factor large applications into logical components that

can share stateful services and communicate in a secure fashion. An app that handles customer requests might include separate modules to handle other tasks:

- API requests from mobile devices
- Internal, admin-like requests
- Backend processing such as billing pipelines and data analysis

Modules can have different versions, performance levels, and authorization. While running, a particular module will have one or more instances, which may be managed statically or dynamically. Incoming requests are routed to an existing or new instance of the appropriate module.
Scaling types control the creation of instances. There are three scaling types. Each type offers a variety of instance classes, with different amounts of CPU and Memory:

- A module with manual scaling runs continuously, allowing you to perform complex initialization and rely on the state of its memory over time.
- A module with basic scaling will create an instance when the application receives a request. The instance will be turned down when the app becomes idle. Basic scaling is ideal for work that is intermittent or driven by user activity.
- Automatic scaling is the scaling policy that App Engine has used since its inception. It is based on request rate, response latencies, and other application metrics. Previously users could use the Cloud Platform Console to configure the automatic scaling parameters (instance class, idle instances and pending latency) for an application's frontend versions only. These settings now apply to every version of every module that has automatic scaling.

Task Queue API
With the Task Queue API, applications can perform work outside of a user request, initiated by a user request. If an app needs to execute some background work, it can use the Task Queue API to organize that work into small, discrete units, called tasks. The app adds tasks to task queues to be executed later.

App Engine Standard environment provides two different queue configurations:

- Push queues process tasks based on the processing rate configured in the queue definition. App Engine Standard environment automatically scales processing capacity to match your queue configuration and processing volume, and also deletes tasks after processing. Push queues are the default.

- Pull queues allow a task consumer (either your application or code external to your application) to lease tasks at a specific time for processing within a specific timeframe. Pull queues give you more control over when tasks are processed, and also allow you to integrate your application with non-App-Engine code using the Task Queue REST API. When using pull queues, your application needs to handle scaling of instances based on processing volume, and also needs to delete tasks after processing.

Sockets API Beta
App Engine Standard environment supports regular outbound sockets in all runtimes, without requiring you to import any special App Engine libraries or add any special App Engine code. However, there are certain limitations and behaviors you need to be aware of when using sockets. The details vary depending on the runtime. Read the runtime documentation for more information.

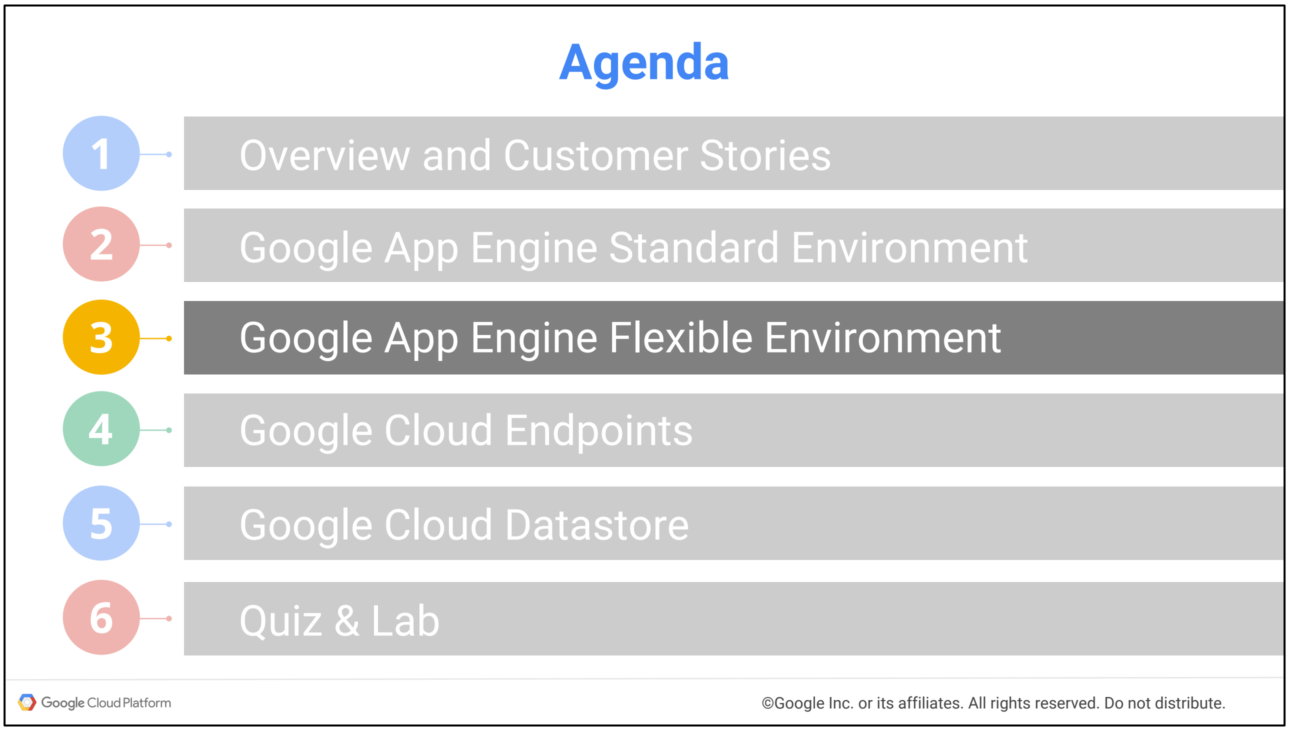
Search API
The Search API provides a model for indexing documents that contain structured data (text and HTML strings, atoms, dates, geopoints, numbers). Documents and indexes are saved in a separate persistent store optimized for search operations. You can search an index, and organize and present search results. The API supports partial text matching on string fields. The Search API can index any number of documents, however, a single search can return no more than 10,000 matching documents. The App Engine Datastore may be more appropriate for applications that need to retrieve very large result sets.

Memcache API
- Used to cache read-heavy operations
  - Database values
  - Tokens
  - API calls
- Key / value pair, in-memory datastore
- Improves performance and reduces costs

App Engine Standard environment supports two classes of the memcache service:
- Shared memcache is the free default for App Engine applications. It provides cache capacity on a best-effort basis and is subject to the overall demand of all applications served by App Engine.
- Dedicated memcache provides a fixed cache capacity assigned exclusively to your application. It's billed by the GB-hour of cache size. Having control over cache size means your app can perform more

- predictably and with fewer accesses to more costly durable storage.

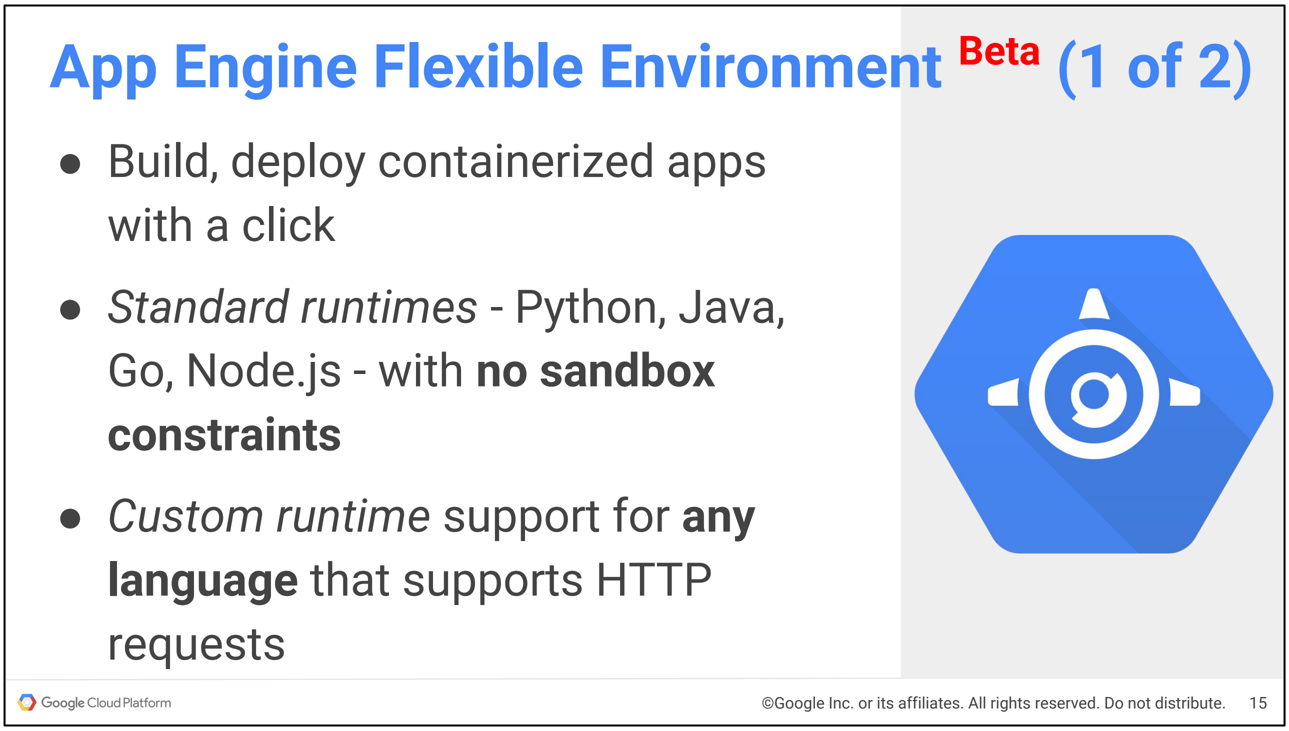
Mail API
App Engine Standard environment applications can send email messages on behalf of the app's administrators, and on behalf of users with Google Accounts. Apps can receive email at various addresses. Apps send messages using the Mail service and receive messages in the form of HTTP requests initiated by App Engine and posted to the app.
This slide shows many of the most commonly used App Engine services. For a complete list of App Engine Standard Environment services, see:
https://cloud.google.com/appengine/docs/about-the-standard-environment



Agenda

1. Overview and Customer Stories
2. Google App Engine Standard Environment
3. **Google App Engine Flexible Environment**
4. Google Cloud Endpoints
5. Google Cloud Datastore
6. Quiz & Lab

Google Cloud Platform



App Engine Flexible Environment Beta (1 of 2)

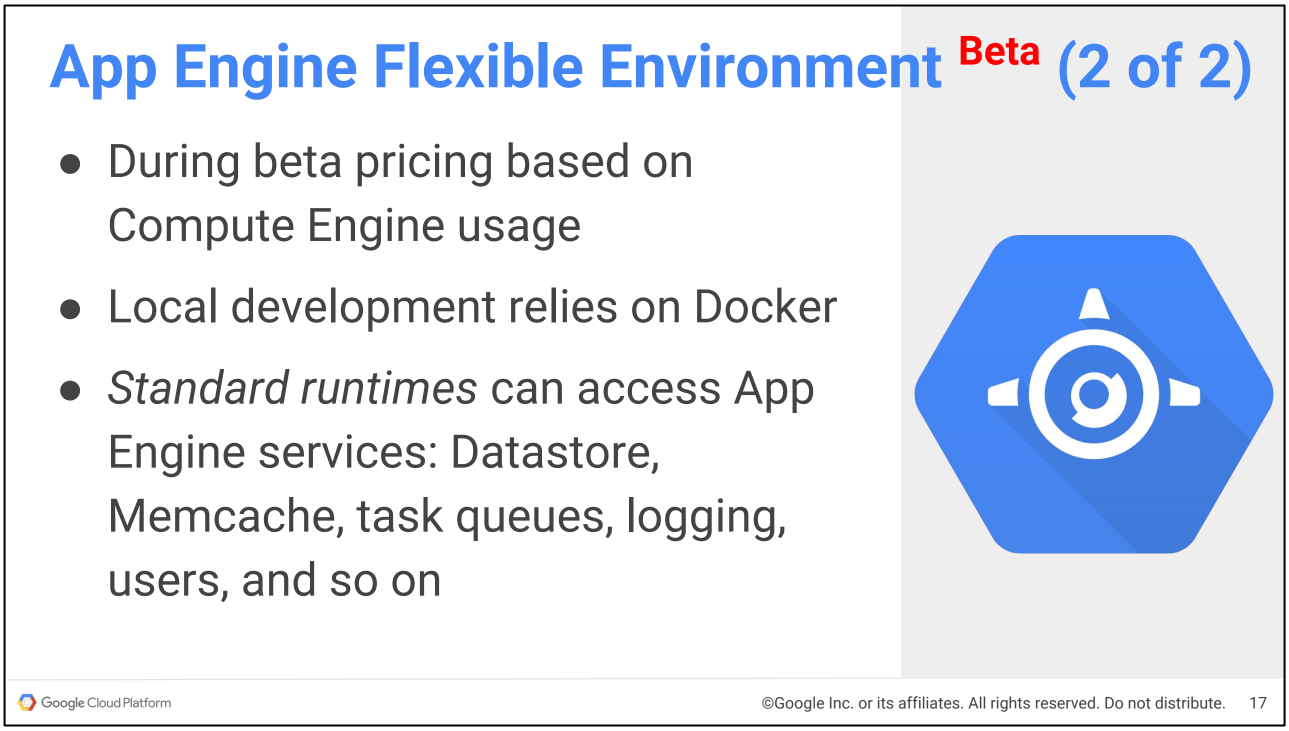
- Build, deploy containerized apps with a click
- *Standard runtimes* - Python, Java, Go, Node.js - with **no sandbox constraints**
- *Custom runtime* support for **any language** that supports HTTP requests

Notes:
Based on Google Compute Engine, the App Engine flexible environment automatically scales your app up and down while balancing the load. Microservices, authorization, SQL and noSQL databases, traffic splitting, logging, search, versioning, security scanning, memcache, and content delivery networks are all supported natively. In addition, the App Engine flexible environment allows you to customize your runtime and even the operating system of your virtual machine using Dockerfiles.

- *Runtimes* - The flexible environment includes native support for Java 8 / Servlet 3.1 / Jetty 9, Python 2.7 and Python 3.4, Node.js, and Go. Developers can customize these runtimes or provide their own runtime, such as Ruby or PHP, by supplying a custom Docker image or Dockerfile from the open source community.
- *Infrastructure Customization* - Because VM instances in the flexible environment are Google Compute Engine virtual machines, you can use SSH to connect to every single VM and Docker container for debugging purposes and further customization.
- *Performance* - Take advantage of a wide array of CPU and memory configurations. You can specify how much CPU and memory each instance of your application needs and the flexible environment will provision the necessary infrastructure for you.

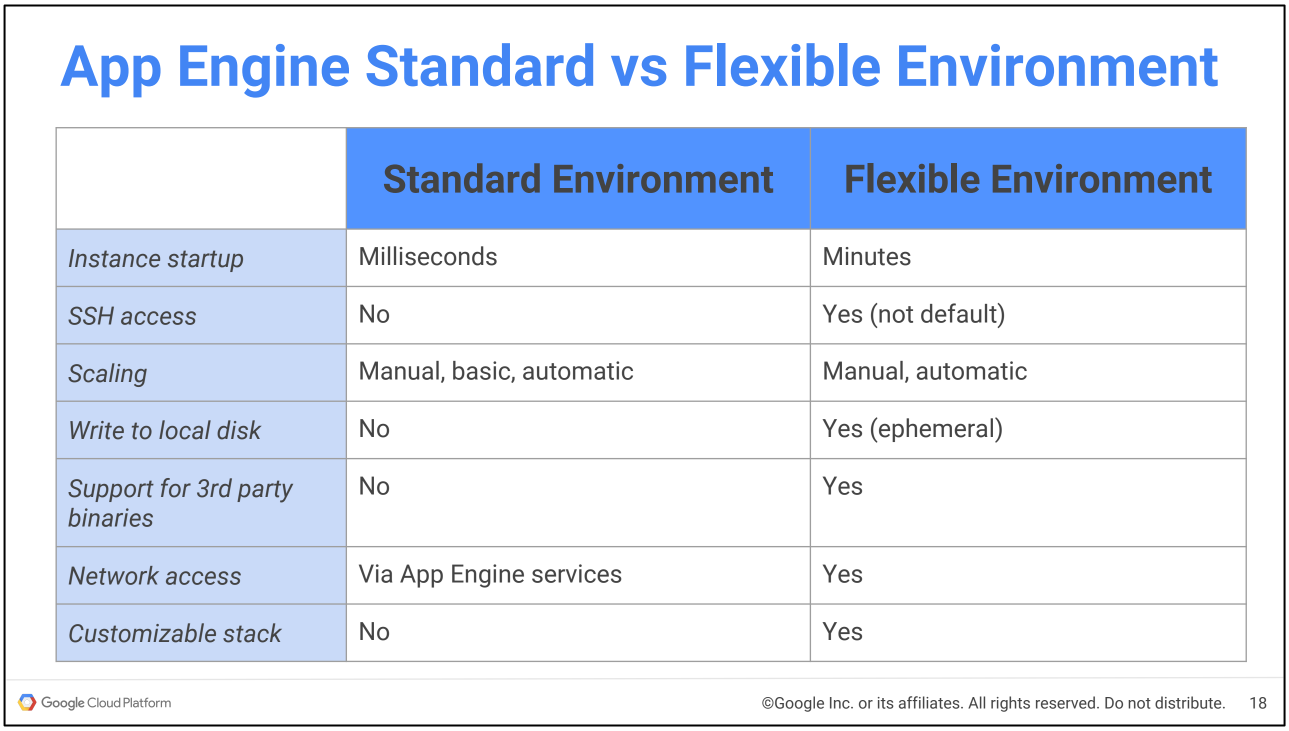
App Engine manages your virtual machines, ensuring that:

- Instances are health-checked, healed as necessary, and co-located with other module instances within the project.
- Critical, backwards compatible updates are automatically applied to the underlying operating system.
- VM instances are automatically located by geographical region according to the settings in your project. Google's management services ensures that all of a project's VM instances are co-located for optimal performance.
- VM instances are restarted on a weekly basis. During restarts Google's management services will apply any necessary operating system and security updates.



App Engine Flexible Environment Beta (2 of 2)

- During beta pricing based on Compute Engine usage
- Local development relies on Docker
- *Standard runtimes* can access App Engine services: Datastore, Memcache, task queues, logging, users, and so on



App Engine Standard vs Flexible Environment

	Standard Environment	Flexible Environment
Instance startup	Milliseconds	Minutes
SSH access	No	Yes (not default)
Scaling	Manual, basic, automatic	Manual, automatic
Write to local disk	No	Yes (ephemeral)
Support for 3rd party binaries	No	Yes
Network access	Via App Engine services	Yes
Customizable stack	No	Yes

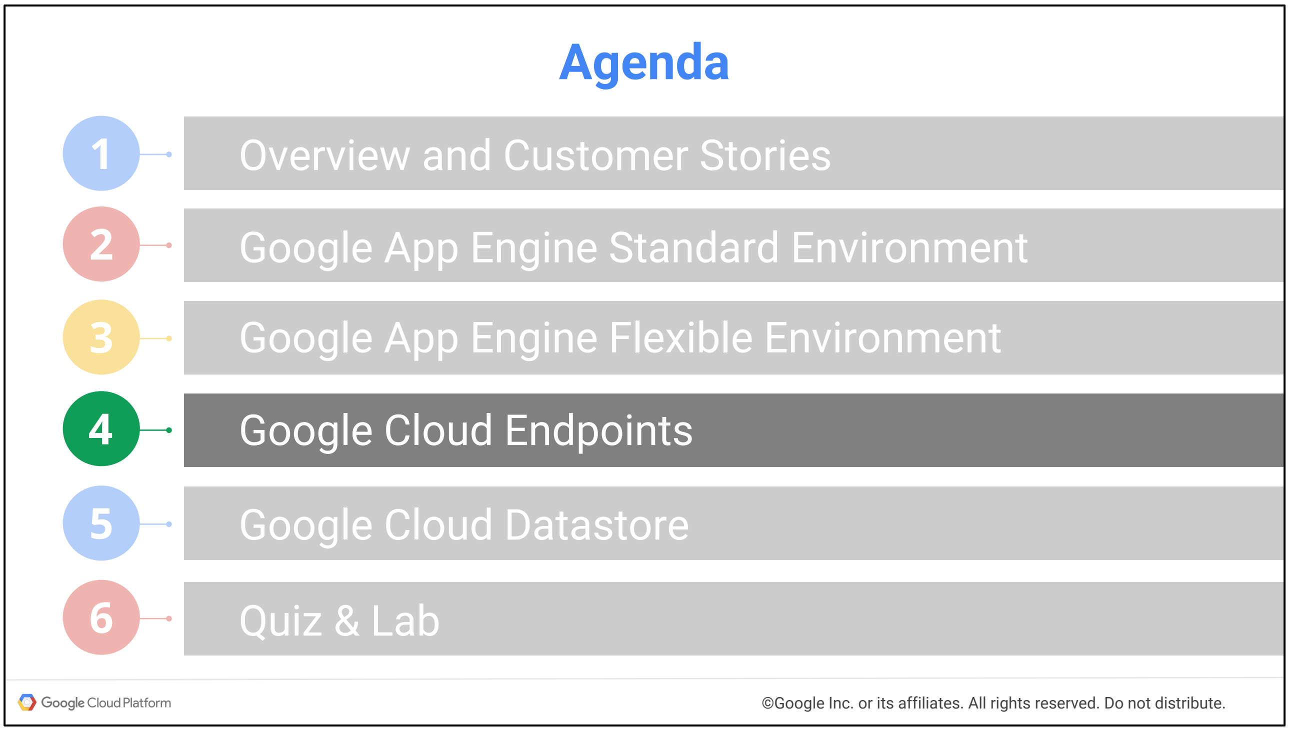
Google Cloud Platform

Notes:
You can run an App Engine application in two environments, the standard environment and the flexible environment. You can use both environments in your application at the same time if you structure your application using the microservices architecture.

Flexible environment versus Compute Engine

While flexible environment runs modules in Compute Engine VM instances, it differs from Compute Engine in the following ways:

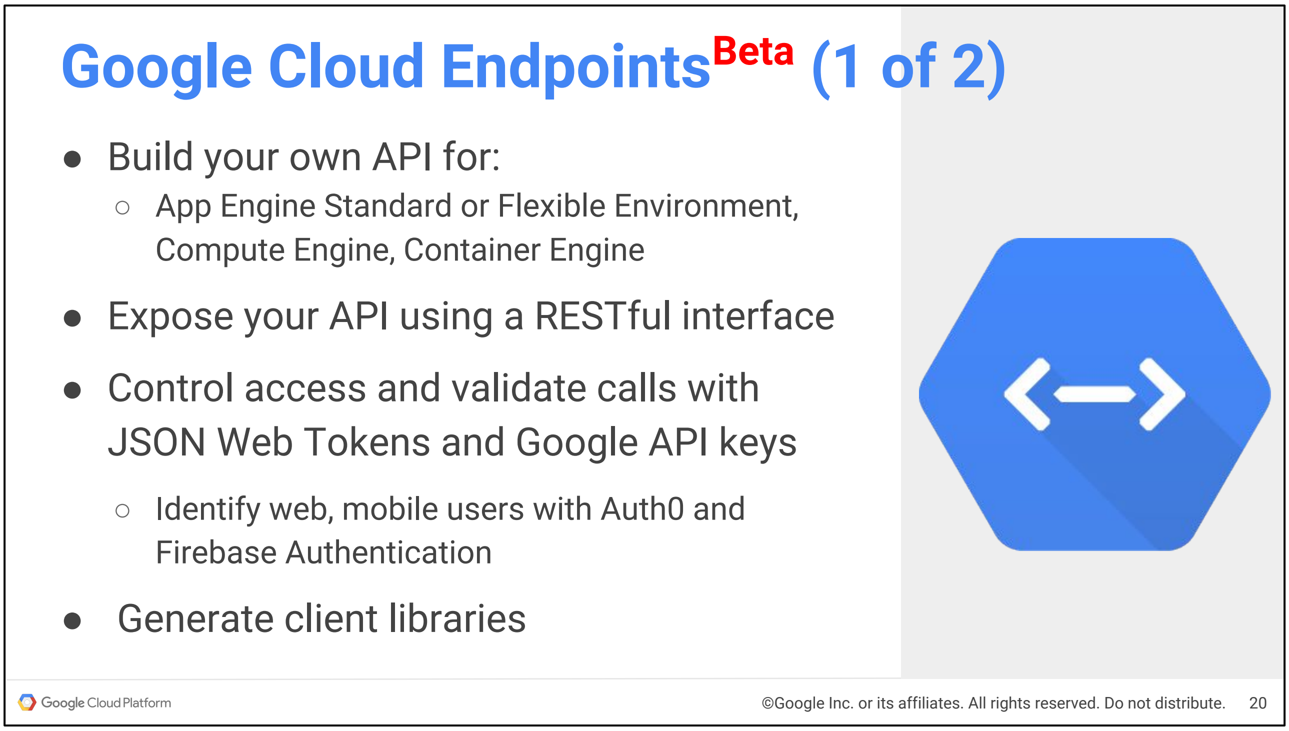
- Flexible environment VM instances are restarted on a weekly basis. During restarts Google's management services will apply any necessary operating system and security updates.
- You always have root access to Compute Engine VM instances. SSH access to VM instances in the flexible environment is disabled by default. If you choose, you can enable root access to your app's VM instances.
- Flexible environment VM instances are automatically located by geographical region according to the settings in your project. Google's management services will ensure that all the VM instances for a project are co-located for optimal performance.



Agenda

1. Overview and Customer Stories
2. Google App Engine Standard Environment
3. Google App Engine Flexible Environment
4. Google Cloud Endpoints
5. Google Cloud Datastore
6. Quiz & Lab

Google Cloud Platform



Google Cloud Endpoints Beta (1 of 2)

- Build your own API for:
  - App Engine Standard or Flexible Environment, Compute Engine, Container Engine
- Expose your API using a RESTful interface
- Control access and validate calls with JSON Web Tokens and Google API keys
  - Identify web, mobile users with Auth0 and Firebase Authentication
- Generate client libraries

Notes:
Google Cloud Endpoints is a distributed API management system. It provides an API console, hosting, logging, monitoring and other features to help you create, share, maintain, and secure your APIs. You can use Cloud Endpoints with any APIs that supports the OpenAPI Specification, formerly known as the Swagger spec.

Cloud Endpoints uses the distributed Extensible Service Proxy to provide low latency and high performance for serving even the most demanding APIs. Extensible Service Proxy is a service proxy based on NGINX. It runs in its own Docker container for better isolation and scalability. The proxy is containerized and distributed in the Google Container Registry and Docker registry, and can be used with Google App Engine, Google Container Engine, Google Compute Engine or Kubernetes.

Cloud Endpoints features
User Authentication
- JSON Web Token validation and a streamlined developer experience for Firebase Auth, Google Auth and Auth0.

Automated Deployment
- With Google App Engine, the proxy is deployed automatically with your application. On Google Container Engine or Compute Engine, use our

- containerized ESP for simple deployment.

Logging and Monitoring

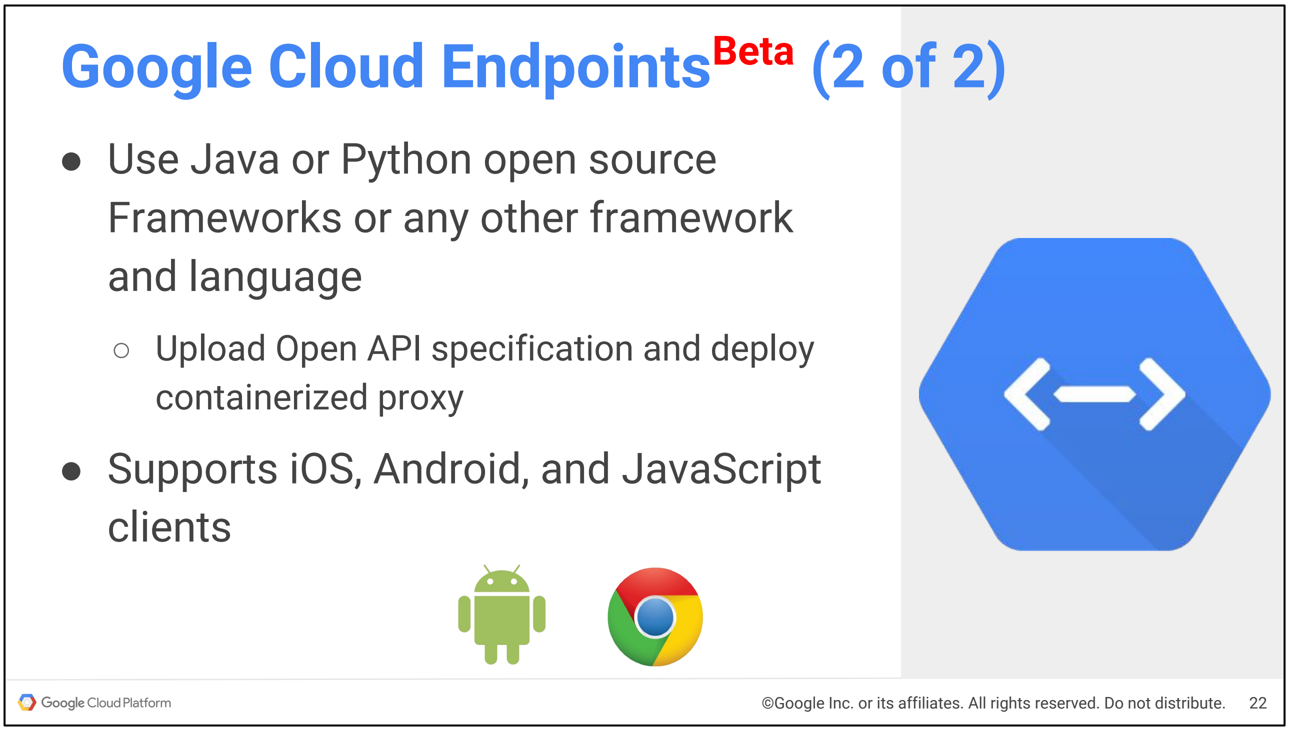
- Monitor traffic, error rates and latency, and review logs in Google Cloud Logging. Use Google Cloud Trace to dive into performance and BigQuery for analysis.

API Keys

- Generate API keys in Google Cloud Platform Console and validate on every API call. Share your API with other developers to allow them to generate their own keys.

Easy integration

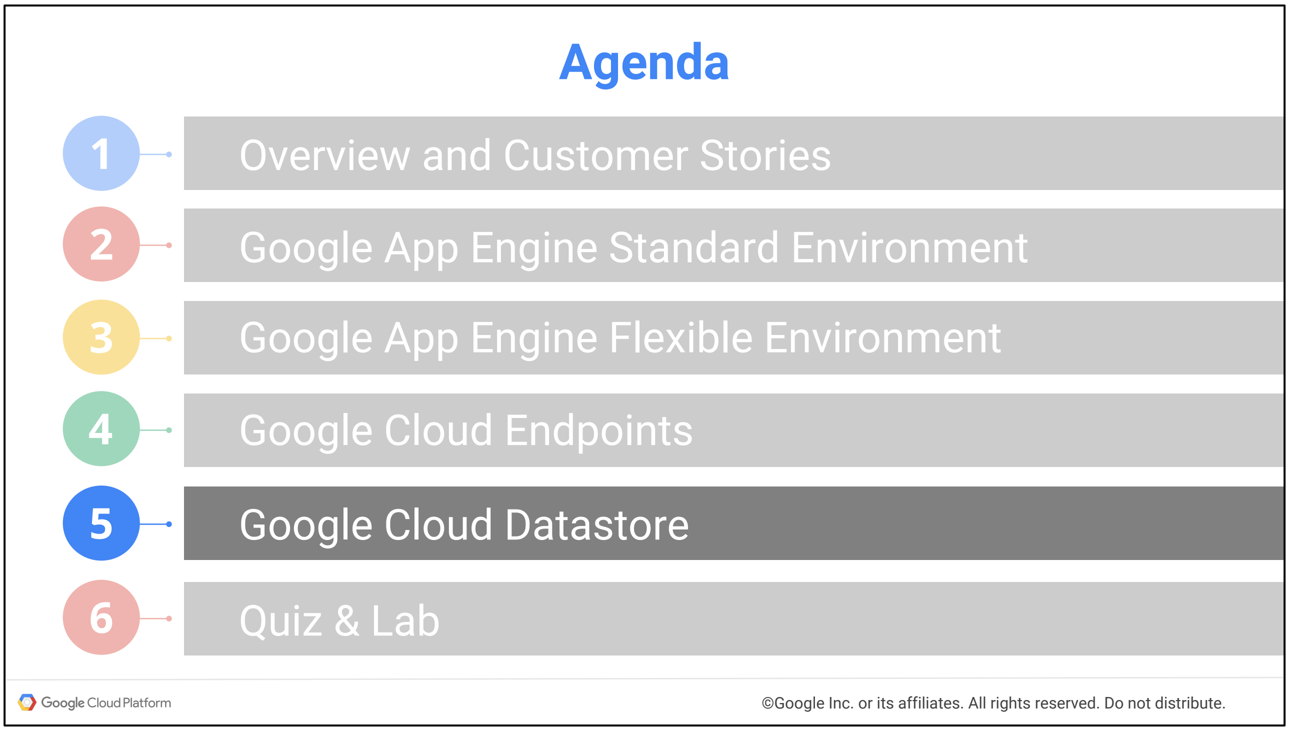
- Get started fast by using one of our Cloud Endpoints Frameworks or by simply adding an Open API specification to your deployment.



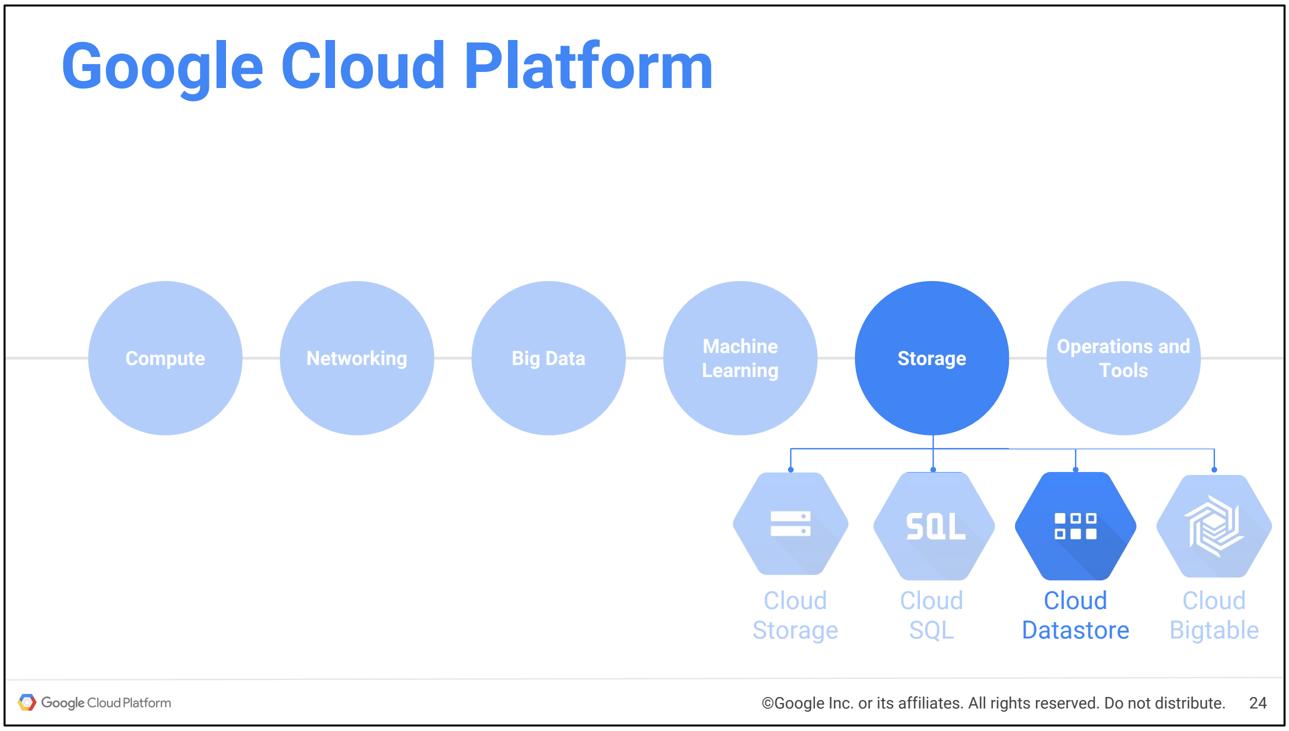
Google Cloud Endpoints Beta (2 of 2)

- Use Java or Python open source Frameworks or any other framework and language
  - Upload Open API specification and deploy containerized proxy
- Supports iOS, Android, and JavaScript clients

Notes:
For more information on Google Cloud Endpoints, see:
https://cloud.google.com/endpoints/docs/about-cloud-endpoints.

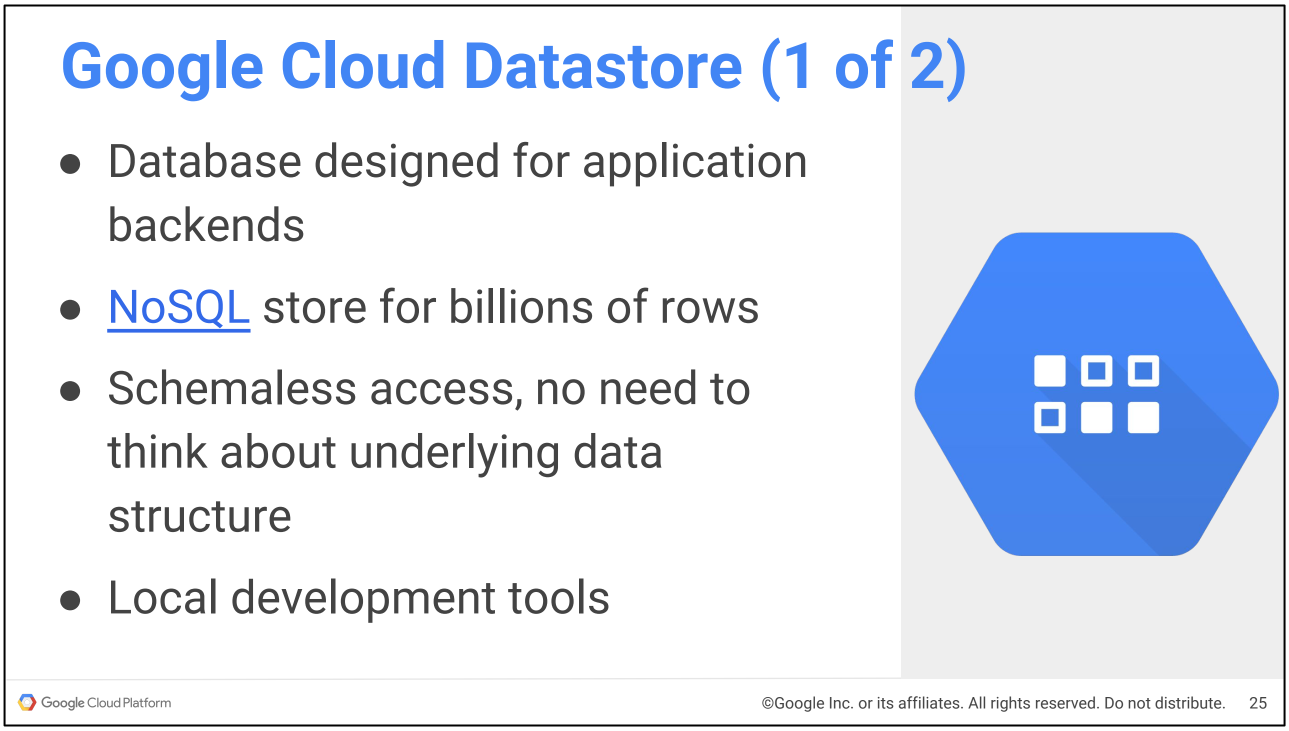


Agenda



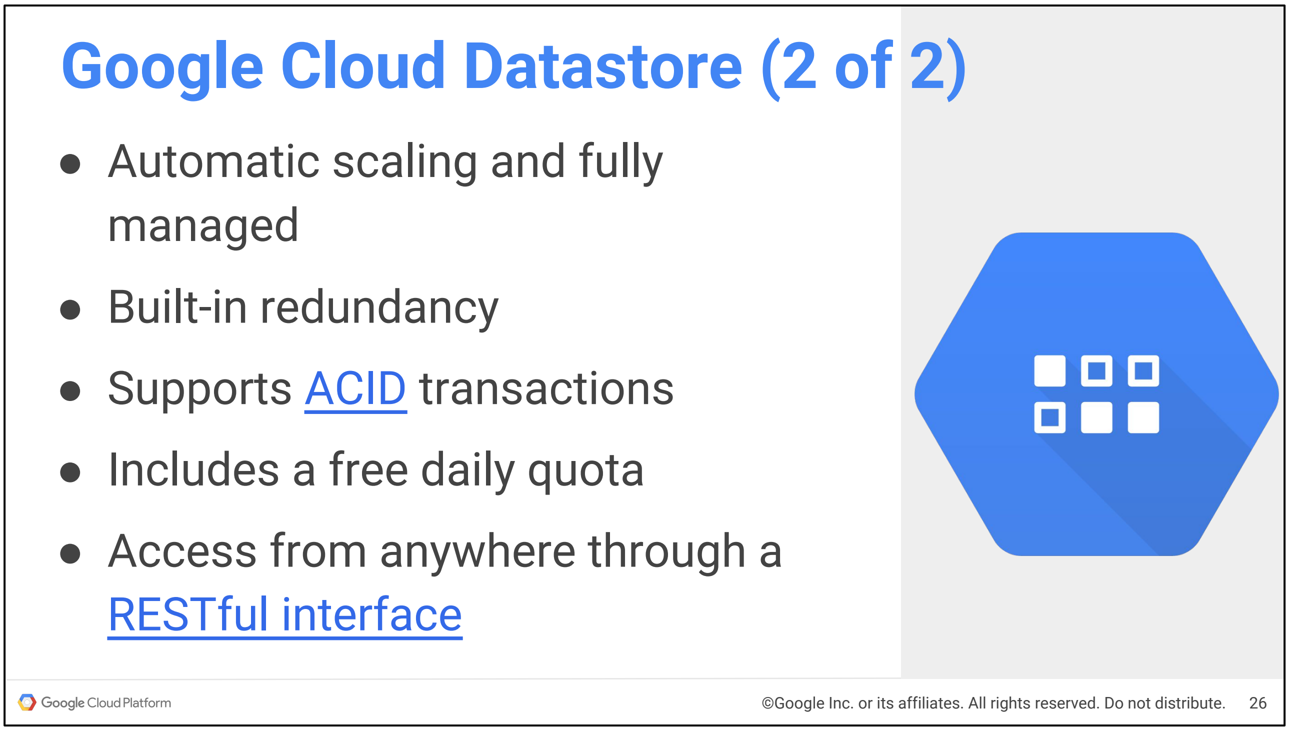
Google Cloud Platform

Notes:
Cloud Datastore is a highly-scalable NoSQL database for your applications. Cloud Datastore automatically handles sharding and replication, providing you with a highly available and durable database that scales automatically to handle your applications' load. Cloud Datastore provides a myriad of capabilities such as ACID transactions, SQL-like queries, indexes and much more.



Google Cloud Datastore (1 of 2)

- Database designed for application backends
- NoSQL store for billions of rows
- Schemaless access, no need to think about underlying data structure
- Local development tools



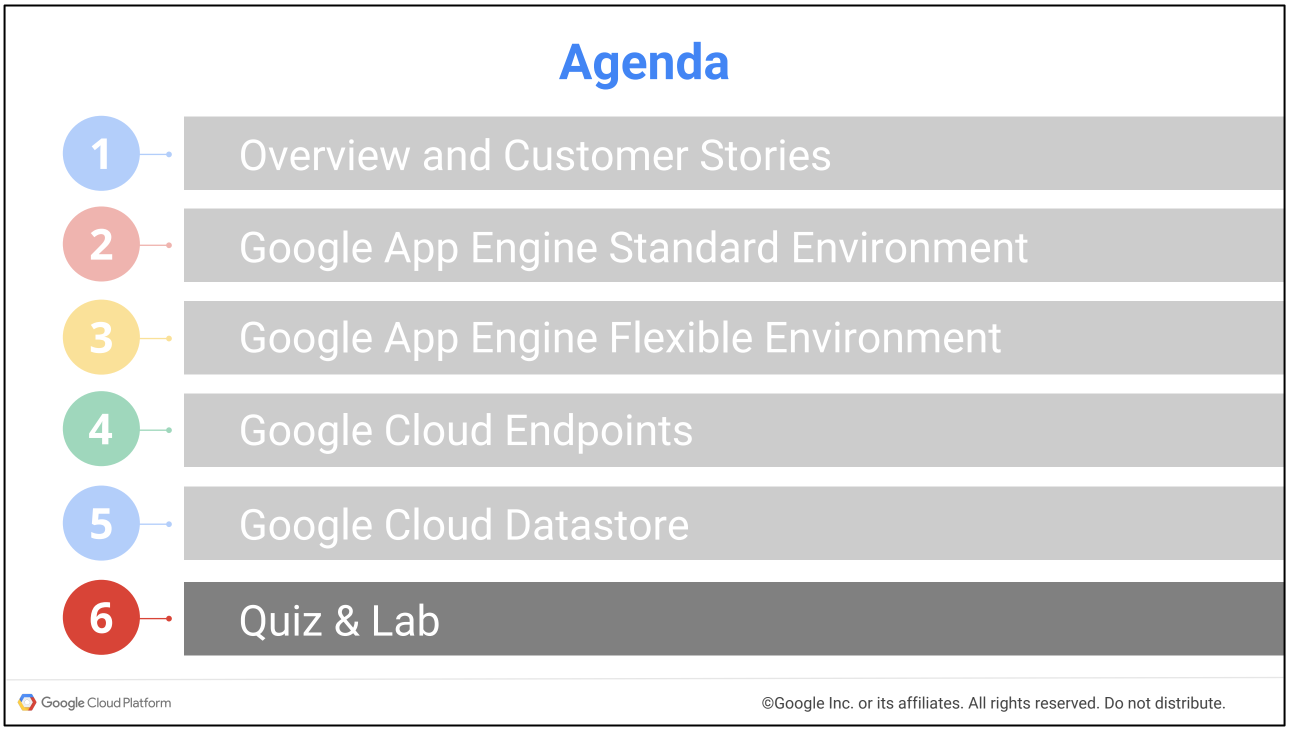
Google Cloud Datastore (2 of 2)

- Automatic scaling and fully managed
- Built-in redundancy
- Supports ACID transactions
- Includes a free daily quota
- Access from anywhere through a RESTful interface

Notes:
Cloud Datastore features:

- *Atomic transactions*. Datastore can execute a set of operations where either all succeed, or none occur.
- *High availability of reads and writes*. Datastore runs in Google data centers, which use redundancy to minimize impact from points of failure.
- *Massive scalability with high performance*. Datastore uses a distributed architecture to automatically manage scaling. Datastore uses a mix of indexes and query constraints so your queries scale with the size of your result set, not the size of your data set.
- *Flexible storage and querying of data*. Datastore maps naturally to object-oriented and scripting languages, and is exposed to applications through multiple clients. It also provides a SQL-like query language.
- *Balance of strong and eventual consistency*. Datastore ensures that entity lookups and ancestor queries always receive strongly consistent data. All other queries are eventually consistent. The consistency models allow your application to deliver a great user experience while handling large amounts of data and users.
- *Encryption at rest*. Datastore automatically encrypts all data before it is written to disk and automatically decrypts the data when read by an authorized user. For more information, see Server-Side Encryption.

- *Fully managed with no planned downtime*. Google handles the administration of the Datastore service so you can focus on your application. Your application can still use Datastore when the service receives a planned upgrade.



Agenda

Google Cloud Platform

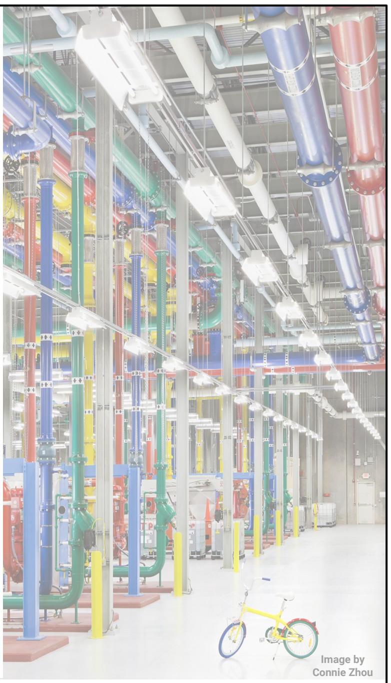
Quiz

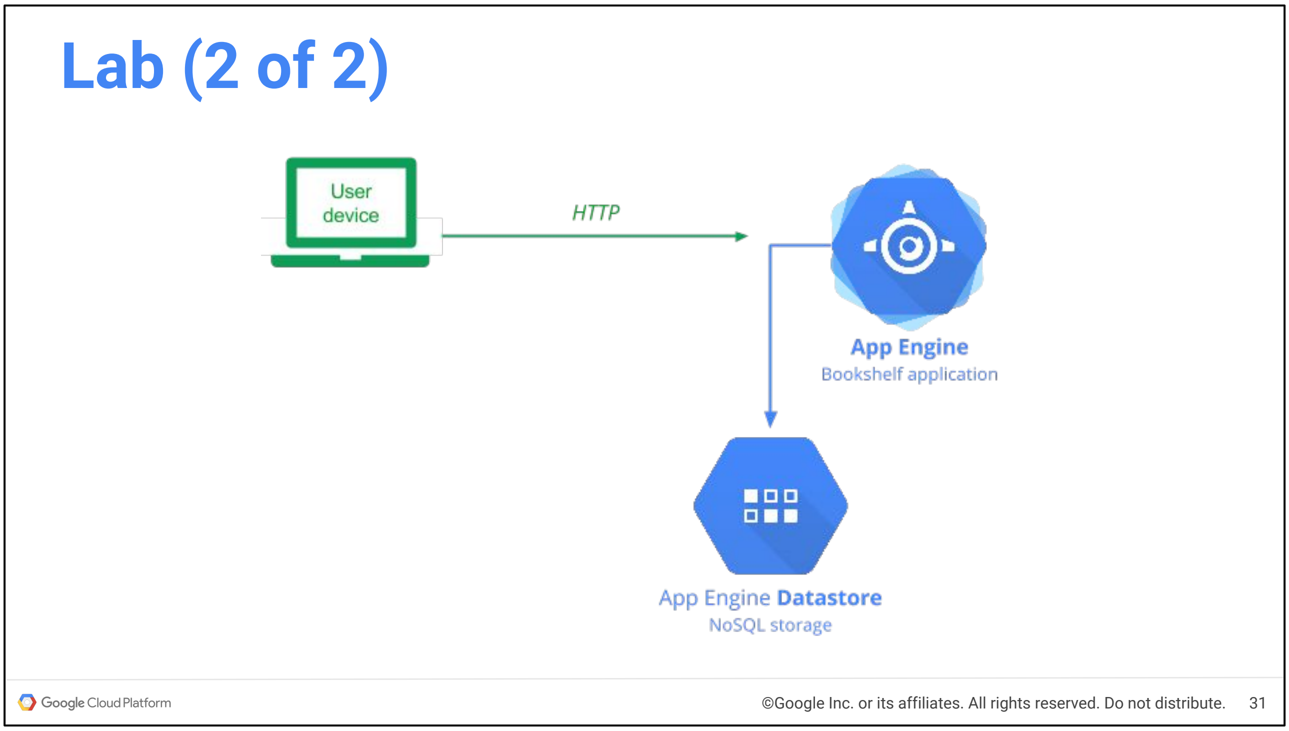
1. Name 3 differences between the App Engine Standard and App Engine Flexible Environments.
2. *True or False*: Google Cloud Datastore supports ACID transactions.

Lab (1 of 2)

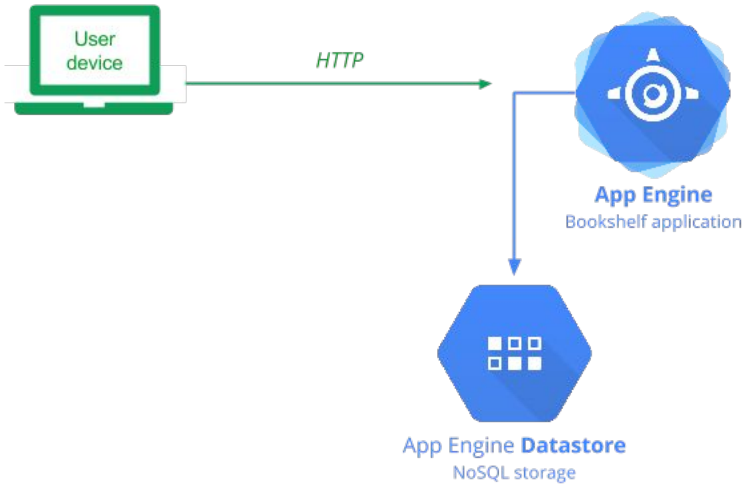
Deploy the Bookshelf application to App Engine using Cloud Datastore for data persistence.

1. Clone and review the application code
2. Deploy the Bookshelf application to App Engine using Cloud Shell
3. Test the application in your browser





Lab (2 of 2)



Notes:
Bookshelf is a simple Python web application that lets users create and manage a list of books. The application uses the Flask web microframework to coordinate reading and writing to storage. This version of Bookshelf uses Cloud Datastore to persistently store book data.

Resources

- Overview: App Engine
  https://cloud.google.com/appengine/
- DevBytes - Your app, at scale with Google App Engine
  https://www.youtube.com/watch?v=ytT2-kL9v2o
- Datastore Concepts Overview
  https://cloud.google.com/datastore/docs/concepts/overview
- Getting started with Google Cloud Datastore API
  https://cloud.google.com/datastore/docs/datastore-api-tutorial

Quiz Answers

1. Name 3 advantages of using the App Engine Flexible Environment over App Engine Standard.

   Answer: The Flexible Environment allows SSH access, allows disk writes, and supports third-party binaries (also allows stack customization and background processes).

2. *True*: Google Cloud Datastore supports ACID transactions.

cloud.google.com