Applications in the Cloud
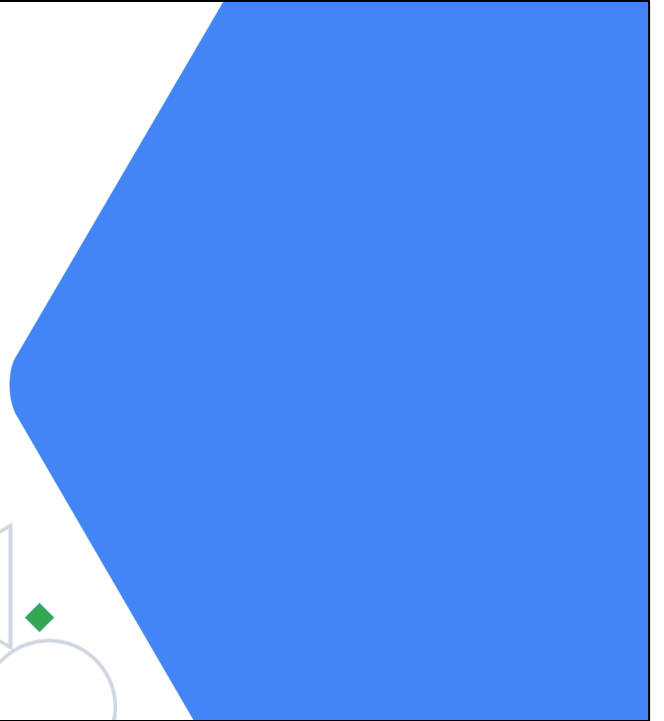
Agenda

Google Cloud

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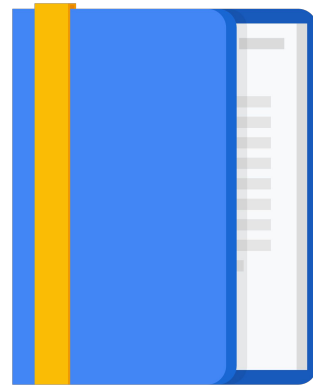
App Engine provides you with built-in services and APIs such as NoSQL datastores, memcache, load balancing, health checks, application logging, and a user authentication API, common to most applications.

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 set up, run, schedule, and manage security scans from the Cloud 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.

Agenda

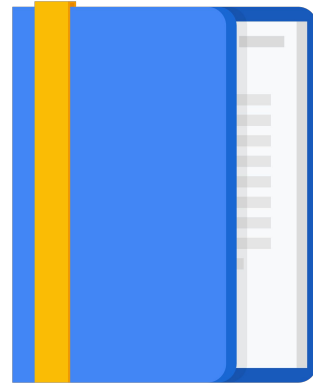App Engine

App Engine Standard Environment

App Engine Flexible Environment

Cloud Endpoints and Apigee Edge

Cloud Run

Quiz and Lab

Resources

Google Cloud

The App Engine standard environment is based on container instances running on Google's infrastructure. Containers are preconfigured with one of several available runtimes. 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: Requirements

- Specific versions of Java, Python, PHP, Go, Node.js, and Ruby (beta) are supported.

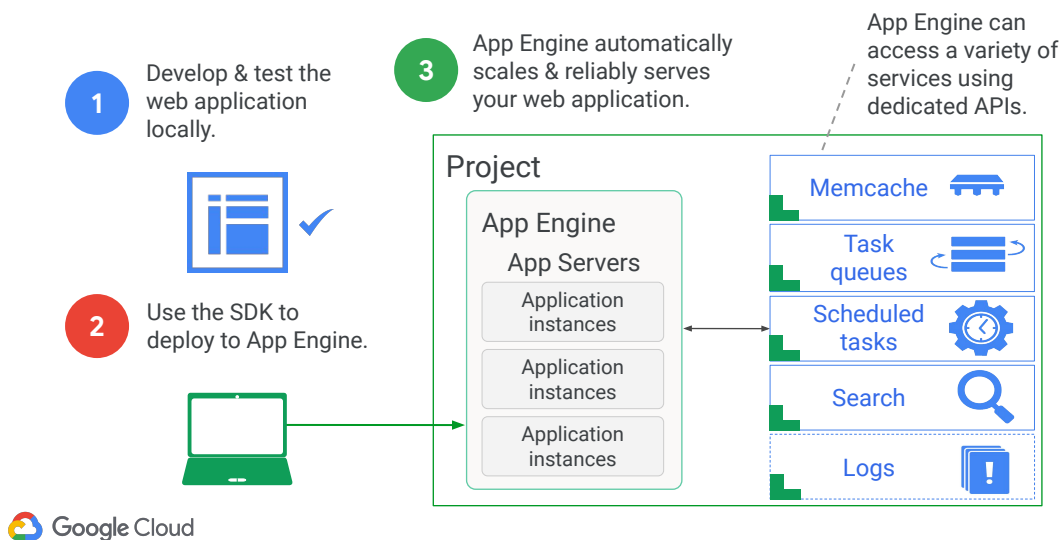- Your application must conform to certain sandbox constraints dependant on runtime.

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, and the Cloud Console manages your application in production. The Cloud 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 the 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 apps

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

Google Cloud

In this diagram we see App Engine standard environment in practice. You'll develop your application and run a test version of it locally using the App Engine SDK. Then, when you're ready, you'll use the SDK to deploy it.

Each App Engine application runs in a Google Cloud project. App Engine automatically provisions server instances and scales and load-balances them. Meanwhile, your application can make calls to a variety of services using dedicated APIs. For example, a NoSQL datastore to make data persistent; caching of that data using memcache; searching; logging; user login; and the ability to launch actions not triggered by direct user requests, like task queues and a task scheduler.

Agenda

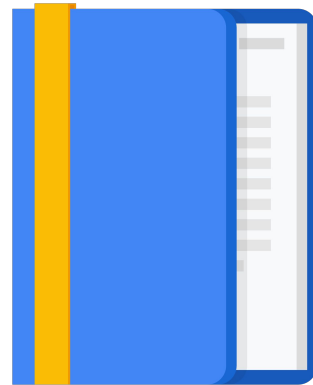App Engine

App Engine Standard Environment

App Engine Flexible Environment

Cloud Endpoints and Apigee Edge

Cloud Run

Quiz and Lab

Resources

Google Cloud

App Engine flexible environment

- Build and deploy containerized apps with a click.

- No sandbox constraints.

- Can access App Engine resources.

- *Standard runtimes:* Python, Java, Go, Node.js, PHP, .NET, and Ruby.

- *Custom runtime support:* Any language that supports HTTP requests.

- Package your runtime as a Dockerfile.

Google Cloud

If the restrictions of App Engine standard environment's sandbox model don't work for you, but you still want to take advantage of the benefits of App Engine (like automatic scaling up and down), consider App Engine flexible environment. Instead of the sandbox, App Engine flexible environment lets you specify the container your application runs in.

Your application runs inside Docker containers on Google Compute Engine virtual machines (VMs). App Engine manages these Compute Engine machines for you. They're health-checked, healed as necessary, and you get to choose what geographical region they run in. And critical, backward-compatible updates to their operating systems are automatically applied. All this so that you can just focus on your code.

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 Go, Java 8, PHP 5/7, Python 2.7/3.6, .NET, Node.js, and Ruby. 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 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, backward-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 ensure 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 apps that use standard runtimes can access App Engine services: Datastore, Memcache, task queues, logging, users, and so on.
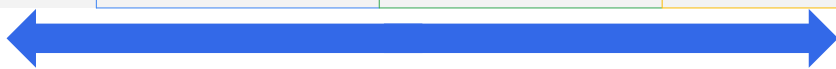
Comparing the App Engine environments

	Standard environment	Flexible environment
Instance startup	Seconds	Minutes
SSH access	No	Yes (although not by default)
Write to local disk	No (some runtimes have read and write access to the /tmp directory)	Yes, ephemeral (disk initialized on each VM startup)
Support for 3rd-party binaries	For certain languages	Yes
Network access	Via App Engine services	Yes
Pricing model	After free daily use, pay per instance class, with automatic shutdown	Pay for resource allocation per hour; no automatic shutdown

Google Cloud

Here's a side-by-side comparison of the standard and flexible environments. Notice that the standard environment starts up instances of your application faster, but that you get less access to the infrastructure in which your application runs. For example, the flexible environment lets you ssh into the virtual machines on which your application runs; it lets you use local disk for scratch space; it lets you install third-party software; and it lets your application make calls to the network without going through App Engine. On the other hand, the standard environment's billing can drop to zero for a completely idle application.

Deploying applications: GKE versus App Engine

	Google Kubernetes Engine	App Engine flexible environment	App Engine standard environment
Language support	Any	Any	Specific versions of Python, Java, Node.js, Go, PHP, and Ruby (beta)
Service model	Hybrid	PaaS	PaaS
Primary use case	Container-based workloads	Web and mobile applications, container-based workloads	Web and mobile applications

Toward managed infrastructure ← → Toward dynamic infrastructure

Google Cloud

Because we mentioned App Engine's use of Docker containers, you may be wondering how App Engine compares to Google Kubernetes Engine. Here's a side-by-side comparison of App Engine with GKE. App Engine standard environment is for people who want the service to take maximum control of their application's deployment and scaling. GKE gives the application owner the full flexibility of Kubernetes. App Engine flexible environment is in between.

Also, App Engine treats containers as a means to an end. But for GKE, containers are a fundamental organizing principle.

Agenda

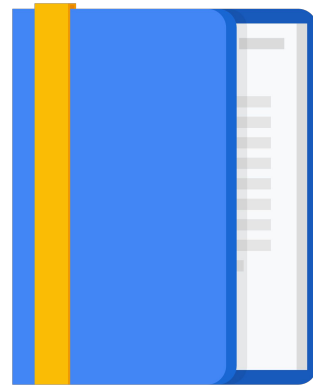App Engine

App Engine Standard Environment

App Engine Flexible Environment
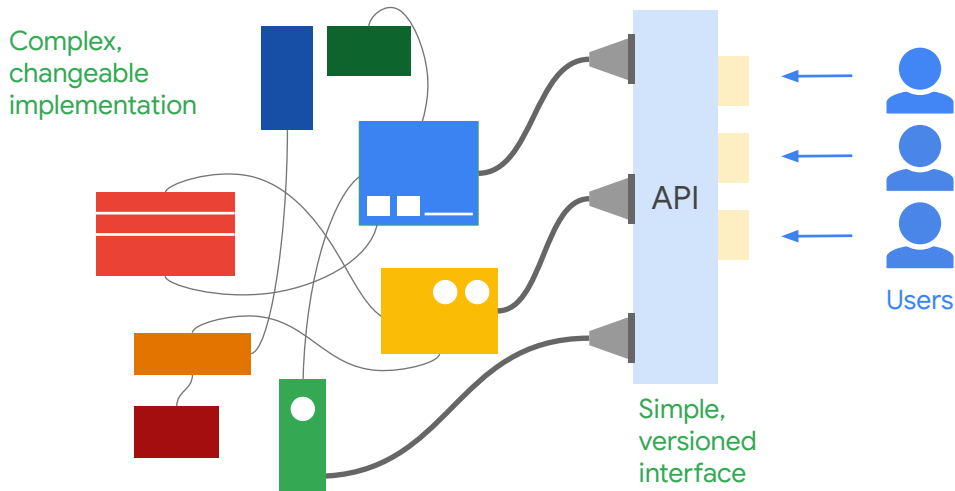
Cloud Endpoints and Apigee Edge

Cloud Run

Quiz and Lab

Resources

Google Cloud

APIs hide the details and enforce contracts

Complex, changeable implementation

API

Users

Simple, versioned interface

Google Cloud

Let's be precise about what an Application Programme Interface, or API, is. A software service's implementation can be complex and changeable. If other software services had to be explicitly coded all that detail in order to use that service, the result would be brittle and error-prone. So instead, application developers structure the software they write so that it presents a clean, well-defined interface that abstracts away needless detail, and then they document that interface. That's an Application Programming Interface. The underlying implementation can change, as long as the interface doesn't, and other pieces of software that use the API don't have to know or care.

Sometimes you do have to change an API, such as to add or deprecate a feature. To make this kind of API change cleanly, developers version their APIs. Version 2 of an API might contain calls that version 1 does not; programs that consume the API can specify the API version they want to use in their calls.

Supporting an API is a very important task, and Google Cloud provides two API management tools. They approach related problems in a different way, and each has a particular forte.

Cloud Endpoints helps you create and maintain APIs

- Distributed API management through an API console.

- 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.

Google Cloud

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 support 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 Container Registry and Docker registry, and can be used with App Engine, Google Kubernetes Engine, 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 App Engine, the proxy is deployed automatically with your application. On Google Kubernetes Engine or Compute Engine, use Google's containerized ESP for simple deployment.

Logging and monitoring: Monitor traffic, error rates and latency, and review logs in Cloud Logging. Use Cloud Trace to dive into performance and BigQuery for analysis.
API keys: Generate API keys in the Cloud 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 quickly by using one of Google's Cloud Endpoints

Frameworks or by simply adding an Open API specification to your deployment.

Cloud Endpoints: Supported platforms

Runtime environment
App Engine flexible environment
Google Kubernetes Engine
Compute Engine

Clients
Android
iOS
Javascript

Google Cloud

Cloud Endpoints supports applications running in Google Cloud's compute platforms, in your choice of languages, and your choice of client technologies.

Apigee Edge helps you secure and monetize APIs

Partner apps

Consumer apps

Employee apps

Cloud apps

Systems of record

Internet of Things

apigee

Design, Secure, Publish, Analyze, Monitor, and Monetize APIs
Manage microservices, Leverage developer portal

Connected experiences

Backend systems

Google Cloud

Apigee Edge is also a platform for developing and managing API proxies. It has a different orientation, though: it has a focus on business problems like rate limiting, quotas, and analytics. Many users of Apigee Edge are providing a software service to other companies, and those features come in handy. Because the backend services for Apigee Edge need not be in Google Cloud, engineers also often use it when they are working to take a legacy application apart. Instead of replacing a monolithic application in one risky move, they can instead use Apigee Edge to peel off its services one by one, standing up microservices to implement each in turn, until the legacy application can finally be retired.

Agenda

App Engine
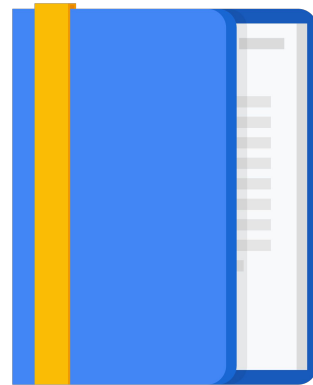
App Engine Standard Environment

App Engine Flexible Environment

Cloud Endpoints and Apigee Edge

Cloud Run

Quiz and Lab

Resources

Google Cloud

Cloud Run

- Enables stateless containers.
- Abstracts away infrastructure management.
- Automatically scales up and down.
- Open API and runtime environment.

Cloud Run is a managed compute platform that enables you to run stateless containers via web requests or Pub/Sub events. Cloud Run is serverless: it abstracts away all infrastructure management so you can focus on developing applications. It is built on Knative, an open-source, Kubernetes-based platform. It builds, deploys, and manages modern serverless workloads. Cloud Run gives you the choice of running your containers either fully-managed or in your own GKE cluster.

Cloud Run automatically scales up and down from zero depending on traffic almost instantaneously, so you never have to worry about scale configuration. Cloud Run also charges you only for the resources you use (calculated down to the nearest 100 milliseconds), so you will never have to pay for your over-provisioned resources.

With Cloud Run you can deploy your stateless containers with a consistent developer experience to a fully managed environment or to your own GKE cluster. This common experience is enabled by Knative, an open API and runtime environment built on Kubernetes that gives you freedom to move your workloads across different environments and platforms: fully managed on Google Cloud, on GKE, or anywhere Knative runs.
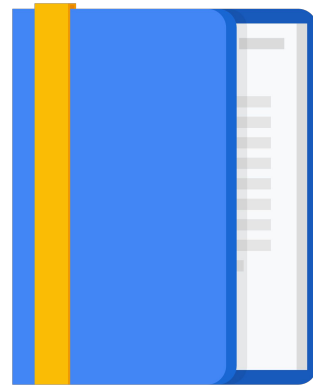
Agenda

App Engine

App Engine Standard Environment

App Engine Flexible Environment

Cloud Endpoints and Apigee Edge

Cloud Run

Quiz and Lab

Resources

Google Cloud

Quiz 1

Name 3 advantages of using the App Engine flexible environment over App Engine standard environment.

Quiz 1

Name 3 advantages of using the App Engine flexible environment over App Engine standard environment.

The flexible environment allows SSH access, allows disk writes, and supports third-party binaries (also allows stack customization and background processes).

Quiz 2

What is the difference between Cloud Endpoints and Apigee Edge?
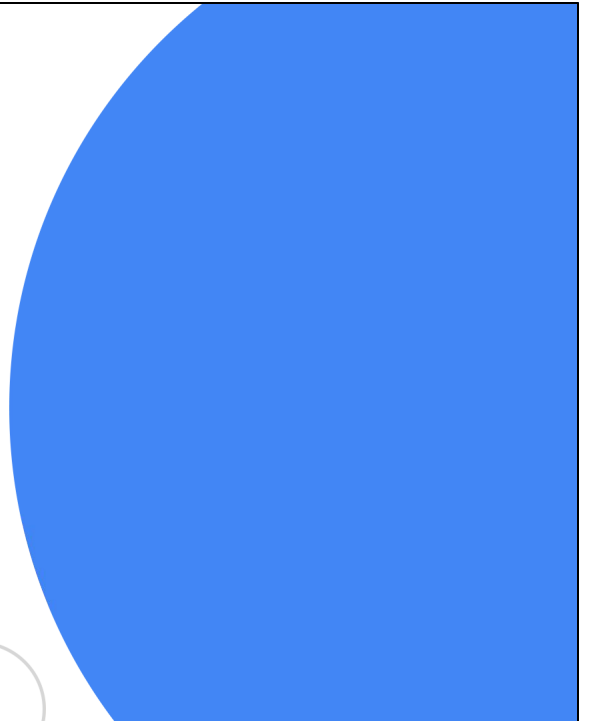
Google Cloud

What is the difference between Cloud Endpoints and Apigee Edge?

Cloud Endpoints helps you create and maintain APIs; Apigee Edge helps you secure and monetize APIs.

Lab Intro

Getting Started with App Engine

Duration: 20 minutes

Google Cloud

The objectives for this lab are for you to be able to:

- Preview an App Engine application using Cloud Shell.
- Launch an App Engine application.
- Disable an App Engine application.
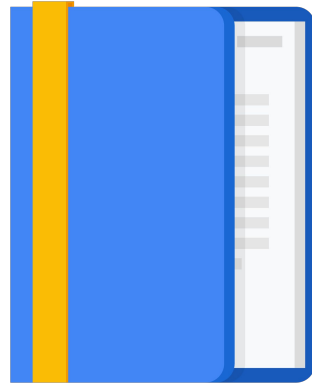
Agenda

App Engine

App Engine Standard Environment

App Engine Flexible Environment

Cloud Endpoints and Apigee Edge

Cloud Run

Quiz and Lab

Resources

Google Cloud

Resources

App Engine   https://cloud.google.com/appengine/docs/

App Engine flexible environment   https://cloud.google.com/appengine/docs/flexible/

App Engine standard environment   https://cloud.google.com/appengine/docs/standard/

Cloud Endpoints   https://cloud.google.com/endpoints/docs/

Apigee Edge   http://docs.apigee.com/api-services/content/what-apigee-edge

Cloud Run   https://cloud.google.com/run/docs

Google Cloud