

Week 3, Cloud Application Development

Azamat Serek, PhD, Assist.Prof.

App Engine

App Engine is a fully managed, serverless platform for developing and hosting web applications at scale. You can choose from several popular languages, libraries, and frameworks to develop your apps, and then let App Engine take care of provisioning servers and scaling your app instances based on demand. [Learn more.](#)

- The Google Cloud Platform (GCP) offers Google App Engine, a fully managed platform-as-a-service (PaaS) solution. It enables application development and deployment without the need to worry about infrastructure administration.
- Features of Google App Engine include numerous language support, seamless interaction with managed services in Google Cloud Platform, automatic scaling and high availability without infrastructure administration.
- The creation of web applications with user authentication, database integration, and dynamic content generation; the backend of mobile applications; API services; microservices; and fast prototyping for testing and validating novel concepts are use cases for Google App Engine.

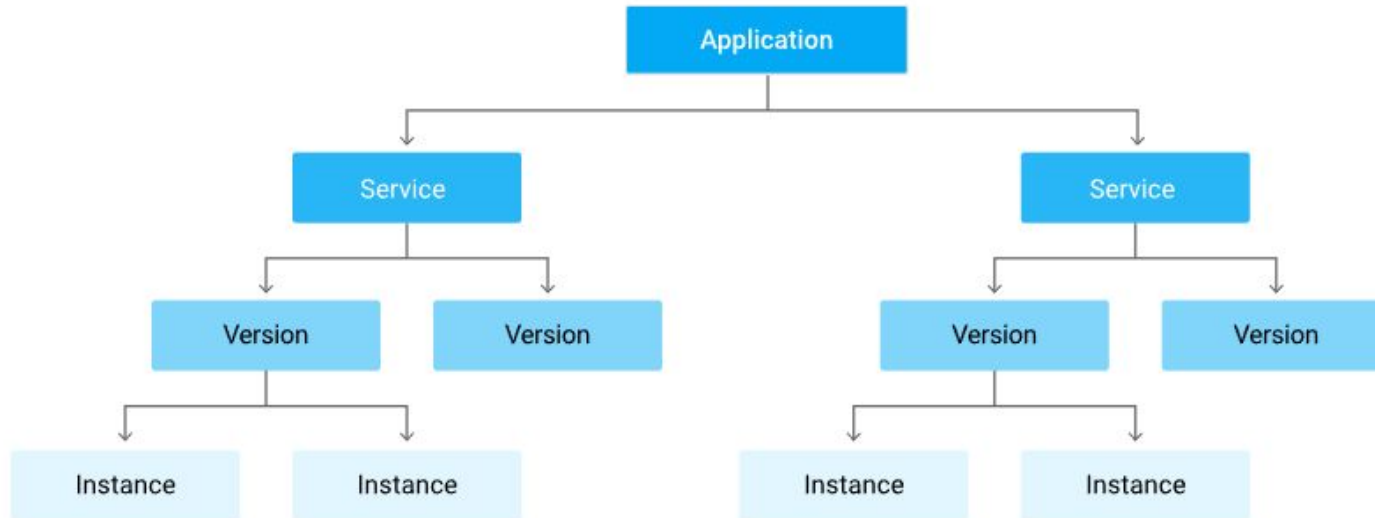
Components of an application

Your App Engine app is created under your Google Cloud project when you create an application resource. The App Engine application is a top-level container that includes the service, version, and instance resources that make up your app. When you create your App Engine app, all your resources are created in the region that you choose, including your app code along with a collection of settings, credentials, and your app's metadata. Learn more about "*application resources*" ([standard](#) | [flexible](#)) and in which regions you can create them.


Each App Engine application includes at least one service, the `default` service, which can hold many versions, depending on your app's billing status. For more information, see [Limits](#) below.

Diagram

The following diagram illustrates the hierarchy of an App Engine app running with multiple services. In this diagram, the app has two services that contain multiple versions, and two of those versions are actively running on multiple instances:



Services

Use *services* in App Engine to factor your large apps into logical components that can securely share App Engine features and communicate with one another. Generally, your App Engine services behave like [microservices](#) . Therefore, you can run your whole app in a single service or you can design and deploy multiple services to run as a set of microservices.

For example, an app that handles your customer requests might include separate services that each handle different tasks, such as:

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

Each service in App Engine consists of the source code from your app and the corresponding App Engine configuration files. The set of files that you deploy to a service represent a single *version* of that service and each time that you deploy to that service, you are creating additional versions within that same service.

Versions

Having multiple versions of your app within each service allows you to quickly switch between different versions of that app for rollbacks, testing, or other temporary events. You can route all traffic to a specific version of your app by "*migrating traffic*" ([standard](#) | [flexible](#)) or route to multiple versions of your app by "*splitting traffic*" ([standard](#) | [flexible](#)).

Instances

The versions within your services run on one or more *instances*. By default, App Engine scales your app to match the load. Your apps will scale up the number of instances that are running to provide consistent performance, or scale down to minimize idle instances and reduces costs. For more information about instances, see "*How instances are managed*" ([standard](#) | [flexible](#)).

In the App Engine flexible environment, instances are backed by Compute Engine resources. Some of the resources used by instances in the App Engine flexible environment, such as disk, CPU, and memory, count towards the [Compute Engine API quotas](#) of your project. For more details on how App Engine uses Compute Engine resources, see the [App Engine flexible environment overview](#).

Application requests ↗

Each of your app's services and each of the versions within those services must have a unique name. You can then use those unique names to target and route traffic to specific resources using URLs, for example:

```
https://VERSION-dot-SERVICE-dot-PROJECT_ID.REGION_ID.r.appspot.com
```

Incoming user requests are routed to the services or versions that are configured to handle traffic. You can also target and route requests to specific services and versions. For more information, see "*Communicating between Services*" ([standard](#) | [flexible](#)).

Logging application requests

When your application handles a request, it can also write its own logging messages to `stdout` and `stderr` [↗](#). For details about your app's logs, see "*Writing Application Logs*" ([standard](#) | [flexible](#)).

Limits

Both the flexible environment and the standard environment share the same limits for services and versions. For example, if you have standard versions and flexible versions in the same app, those versions count towards the same limit. For details, see "*Quotas and limits*" ([standard](#) | [flexible](#)).

Активал
чтобы акт
"Параметр

App Engine standard environment

[Send feedback](#)

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

The standard environment makes it easy to build and deploy an application that runs reliably even under heavy load and with large amounts of data.

Applications run in a secure, sandboxed environment, allowing the standard environment to distribute requests across multiple servers and scale 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.

Standard environment languages and runtimes

The standard environment supports the following languages:

[Go](#)[Java](#)[Node.js](#)[PHP](#)[Python](#)[Ruby](#)

Активаци

Чтобы актив

"Параметр

Instance classes

The *instance class* determines the amount of memory and CPU available to each instance, the amount of [free quota](#), and the [cost per hour](#) after your app exceeds the free quota.

The memory limits vary by [runtime generation](#). For all runtime generations, the memory limit includes the memory your app uses along with the memory that the runtime itself needs to run your app. The Java runtimes use more memory to run your app than other runtimes.

To override the default instance class, use the `instance_class` setting in your app's `app.yaml` file.

Second gen runtimes

First gen runtimes

The second generation runtimes that use this specification are: [Python](#), [Java](#), [Node.js](#), [PHP](#), [Ruby](#), and [Go](#).

★ **Note:** The [Go 1.11](#) runtime has the same instance class specifications as the second generation runtimes.

Instance Class	Memory Limit	CPU Limit	Supported Scaling Types
F1 (default)	384 MB	600 MHz	automatic
F2	768 MB	1.2 GHz	automatic

Активация

Чтобы активир

"Параметры"

Quotas and limits

The standard environment gives you 1 GiB of data storage and traffic for free, which can be increased by enabling paid applications. However, some features impose limits unrelated to quotas to protect the stability of the system. For more details on quotas, including how you can edit them to suit your needs, see [Quotas](#).

Sample applications

<https://cloud.google.com/appengine/docs/legacy/standard/python/samples>

Example

<https://cloud.google.com/appengine/docs/standard/python3/building-app>

Another example

<https://github.com/prayag-sangode/gcp>

Exercise

Deploy your app to google app engine

References

1. <https://medium.com/@prayag-sangode/deploy-a-simple-html-application-on-google-app-engine-gae-4374280eabca>
2. <https://cloud.google.com/>