

Google Cloud Platform

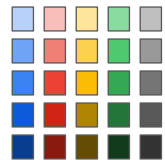


Application
Development

Application Development Services

v 1.0





- Google App Engine (GAE)
 - Platform as a Service (PaaS)



Agenda

- 1 → Google App Engine (GAE)
- 2 → Lab
- 3 → Review

GCP Compute & Processing Options

					
	Compute Engine	Container Engine	App Engine Standard	App Engine Flexible	Cloud Functions
Language support	Any	Any	Java 7 Python 2.7 Go PHP 5	Java 8 Python 2.7/3.5 Go, PHP 5/7 Node.js Ruby Custom Runtimes	Triggers
Usage model	IaaS	IaaS PaaS	PaaS	PaaS	Microservices Architecture
	Server	Cluster	Autoscaling managed servers		Serverless
Primary use case	General Workloads	Container Workloads	Scalable web applications Mobile backend applications		Lightweight Event Actions

Google App Engine (GAE)

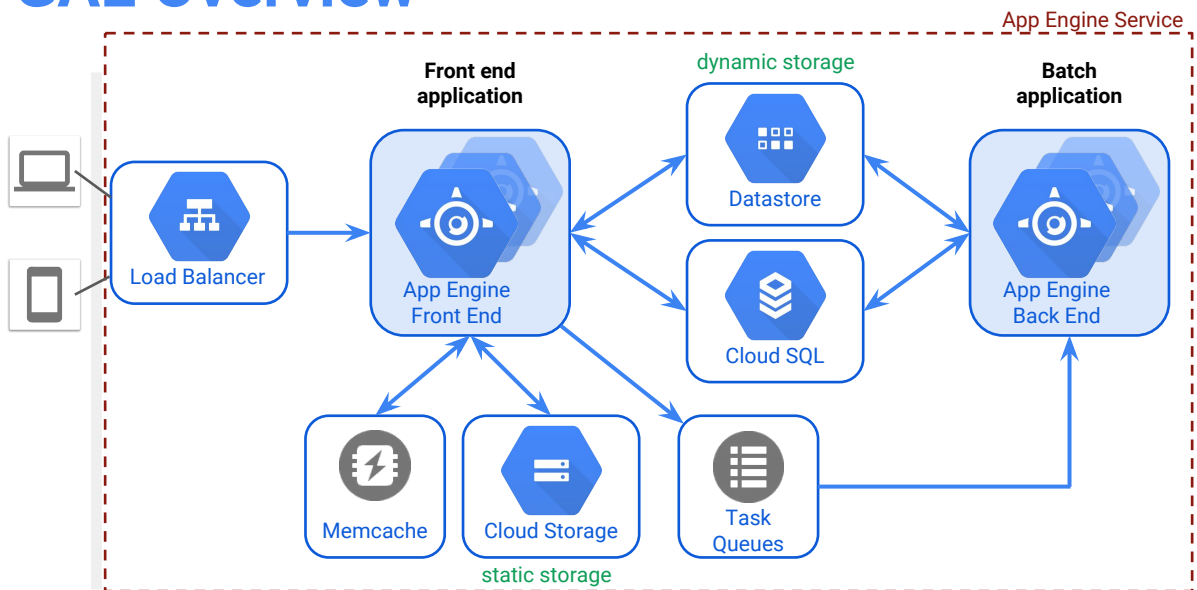
- Platform As A Service (PaaS)
 - Infinite autoscaling
- Standard environment
 - Fully-managed
 - Scale to zero
 - Java 7, Python 2.7, PHP 5, Go
- Flexible environment
 - Docker container support
 - VMs exposed
 - Any language in your container



Google
App Engine

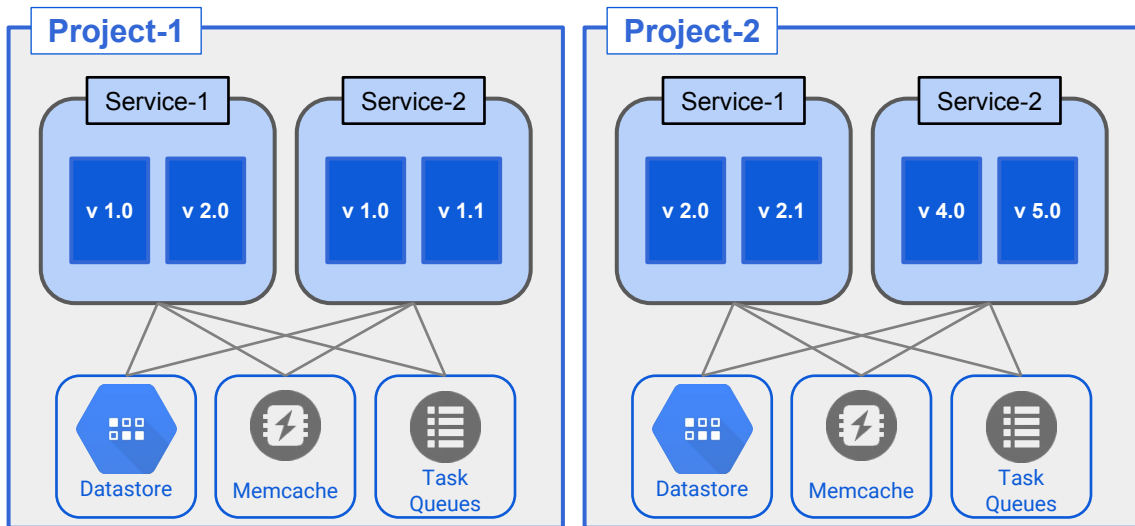
Development tool compatibility: Eclipse, IntelliJ, Maven, Git, Jenkins, and PyCharm

GAE Overview



Front end is often critical to user experience. Back end is for more intensive processing. This separation of function allows each part to scale as needed. The App Engine service is a module. This example shows a single Service. More complex architectures are possible.

Microservices on App Engine



Services are a modular abstraction in App Engine; a way to break up the application into separate parts. Instead of architecting your entire application inside a single App Engine Service as shown on the previous slide, you can implement your application in multiple services. You can have more than one service in a project. You can also use multiple projects to further isolate services. Splitting traffic to different versions enables incremental rollouts and A/B testing.

Each service can have multiple versions deployed simultaneously. Code is completely separate in versions, services, and projects.

Services in a single project share some resources, for example, Datastore, Memcache, and Task Queues.

Services in separate projects are completely isolated.

IAM -- you can assign different roles at the project, but not at the service-level.

When services are in the same project, they are isolated in some ways and share certain resources. Code in one service can't directly call code in another service. Code is deployed independently. Each service can be written in a different language. And the Autoscaling, load balancing, and machine instance type that we saw on the previous slide are independent for each service.

- Define strong contracts between the microservices

- Independent deployment cycles (and rollback if needed)
- Concurrent A/B release testing on subsystems
- Minimize test automation and quality-assurance overhead
- Logging and monitoring clarity
- More detailed cost accounting
- Increase application scalability and reliability

More information:

<https://cloud.google.com/appengine/docs/python/microservices-on-app-engine>

Comparing environments

Feature	Standard	Flexible
Instance startup time	Milliseconds	Minutes
Maximum request timeout	60 seconds	60 minutes
Background threads	Yes, with restrictions	Yes
Background processes	No	Yes
SSH debugging	No	Yes
Scaling	Manual, Basic, Automatic	Manual, Automatic
Writing to local disk	No	Yes, ephemeral (disk initialized on each VM startup)
Customizable serving stack	No	Yes (built by customizing a Dockerfile)
Automatic in-place security patches	Yes	Yes
Network access	Only via App Engine services (includes outbound sockets)	Yes
Supports installing third-party binaries	No	Yes

Agenda

- 1 → Google App Engine (GAE)
- 2 → Lab
- 3 → Review

Lab #1

In this lab you won't write any code, but you will perform many of the activities and use several features that you would use if you were developing code for GAE.

You will clone a sample Guestbook application, test it out in a local development server running on Cloud Shell, and then deploy it to GAE and verify that it is running on the Internet.

Agenda

- 1 → Google App Engine (GAE)
- 2 → Lab
- 3 → Review

More...

- Google App Engine (GAE)
 - <https://cloud.google.com/appengine/docs/>

More to learn on this subject. Here are some suggestions and links.

