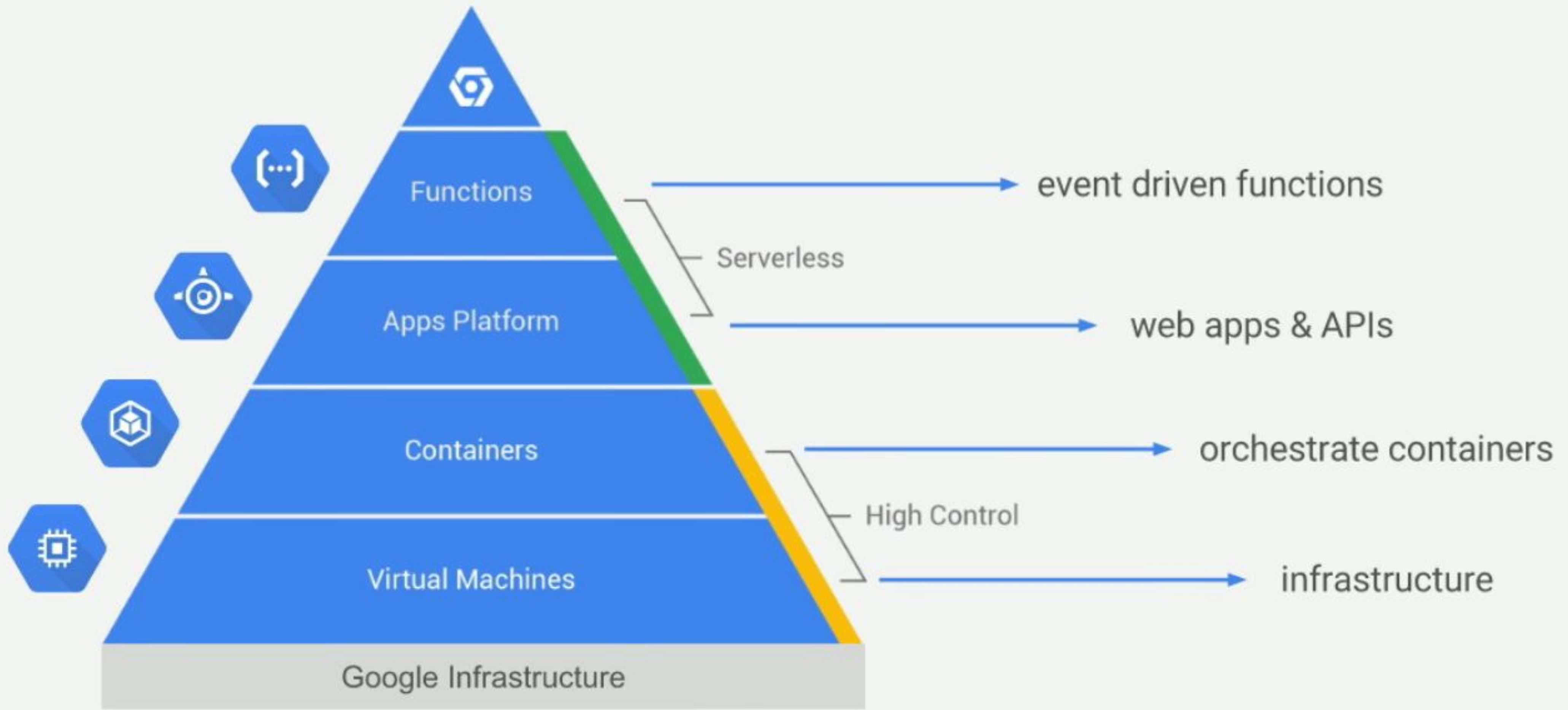GOOGLE CLOUD PLATFORM

APP ENGINE

WHAT IS APP ENGINE

- Fully managed infrastructure to deploy and run your apps

- Build infinitely scalable and high performance apps

- Serverless infrastructure

- Web applications, Mobile backends, HTTP APIs, LOB applications

- Integrated with other Google Cloud offerings

- Examples: Snapchat, mobile gaming companies,

WHERE IT FITS

ENVIRONMENTS

- You can run your applications in App Engine using the flexible environment or standard environment.

STANDARD ENVIRONMENT	FLEXIBLE ENVIRONMENT
Application instances run in a sandbox, using the runtime environment of a supported language	Application instances run within Docker containers on Compute Engine virtual machines (VM).
Applications that need to deal with rapid scaling.	Applications that receive consistent traffic, experience regular traffic fluctuations, or meet the parameters for scaling up and down gradually.
Source code is written in **specific versions of the supported programming languages**	Source code is written in **any version of the supported programming languages**
Intended to **run for free or at very low cost**	Accesses the resources or services of your Google Cloud Platform project inside the Compute engine

GOOGLE CLOUD PLATFORM   APP ENGINE

ENVIRONMENTS

- ## Standard environment
  - Based on container instances running on Google's infrastructure
  - Containers are preconfigured with one of several available runtimes
  - Supports: Go, Java, PHP, Node.js, Python
  - Your application runs within its own secure, reliable environment that is independent of the hardware, operating system, or physical location of the server.
  - Each application running in the standard environment has an *instance class*, which determines its compute resources and pricing.

ENVIRONMENTS

- ## Flexible environment

  - App Engine flexible environment automatically scales your app up and down while balancing the load.

  - Microservices, authorization, SQL and NoSQL databases, traffic splitting, logging, versioning, security scanning, and content delivery networks are all supported natively.

  - App Engine flexible environment allows you to customize the runtime and even the operating system of your virtual machine using Dockerfiles.

  - Supported languages: Go, Java 8, PHP 5/7, Python 2.7/3.6, .Net, Node.js, Custom runtimes

APP ENGINE

ENVIRONMENTS

- Flexible environment

  - **Runtimes** - Developers can customize these runtimes or provide their own runtime by supplying a custom Docker image or Dockerfile from the open source community.

  - **Infrastructure Customization** - You can take advantage of custom libraries, use SSH for debugging, and deploy your own Docker containers.

  - **Performance** – Specify how much CPU and memory each instance of your application needs and the flexible environment will provision the necessary infrastructur.

Python

Java

Custom

.NET

Node.js

Ruby

PHP

ENVIRONMENTS

- ## Flexible environment
  - **Custom runtimes** in the App Engine flexible environment to use an alternative implementation of Java, Python, Node.js, or Go, or write code in any other language.
  - Defining new runtime environments allow you to include additional components like language interpreters or application servers.
  - When you use a custom runtime, you must write your application code to handle certain flexible environment life-cycle and health checking requests.

FEATURE	FLEXIBLE RUNTIME	CUSTOM RUNTIME
Dockerfile	Default file supplied automatically by the SDK	Hand-written by the developer
Dockerfile modifications permitted	Yes	Yes
Languages	Python, Java, Node.js, Go, Ruby, PHP, .NET	Any software that can service HTTP requests

ENVIRONMENTS

- ## Flexible environment

  - Instances are health-checked, healed as necessary, and co-located with other services 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 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.

  - 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.

FEATURES

- Logging: Stackdriver provides a logs API and features such as Request logs, app logs, shutdown logs, runtime logs, and audit logs

- Monitoring: Stackdriver Monitoring provides visibility into the performance, uptime, and overall health of cloud-powered applications.

- Autoscaling: Makes scaling decisions based on multiple metrics, but only works with managed instance groups.

- Load balancing: A single anycast IP front-ends all your backend instances in regions around the world

- SSL & Domain: Offers globally-distributed SSL endpoints and built-in load balancing to serve your app securely, reliably, and quickly to a worldwide audience.

- Health checking: Provides health checking mechanisms that determine whether VM instances respond properly to traffic via **health checks** and **legacy health checks**

Logging

Monitoring

Autoscaling

Load balancing

SSL & Domains
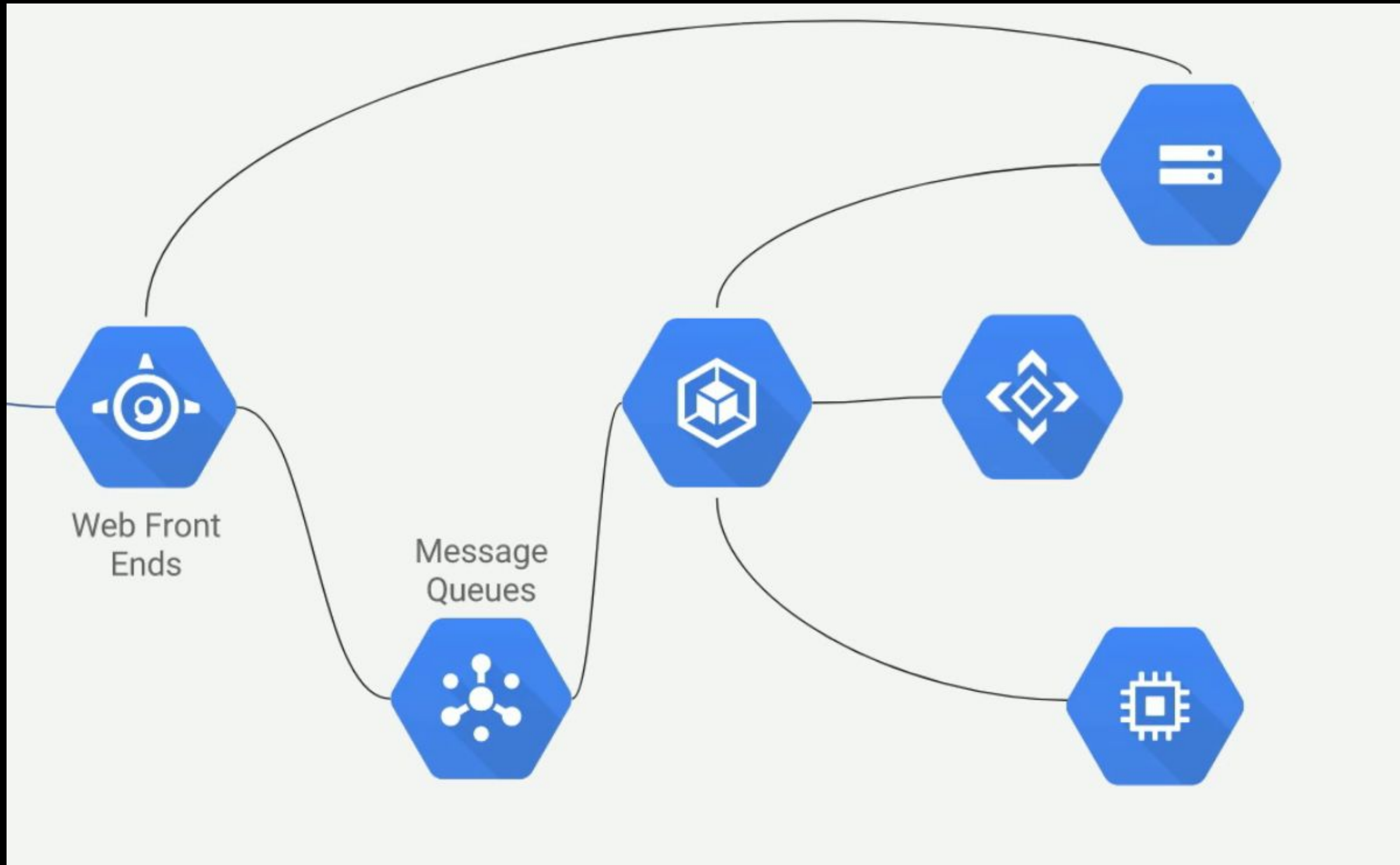
Health checking

TYPICAL ARCHITECTURE
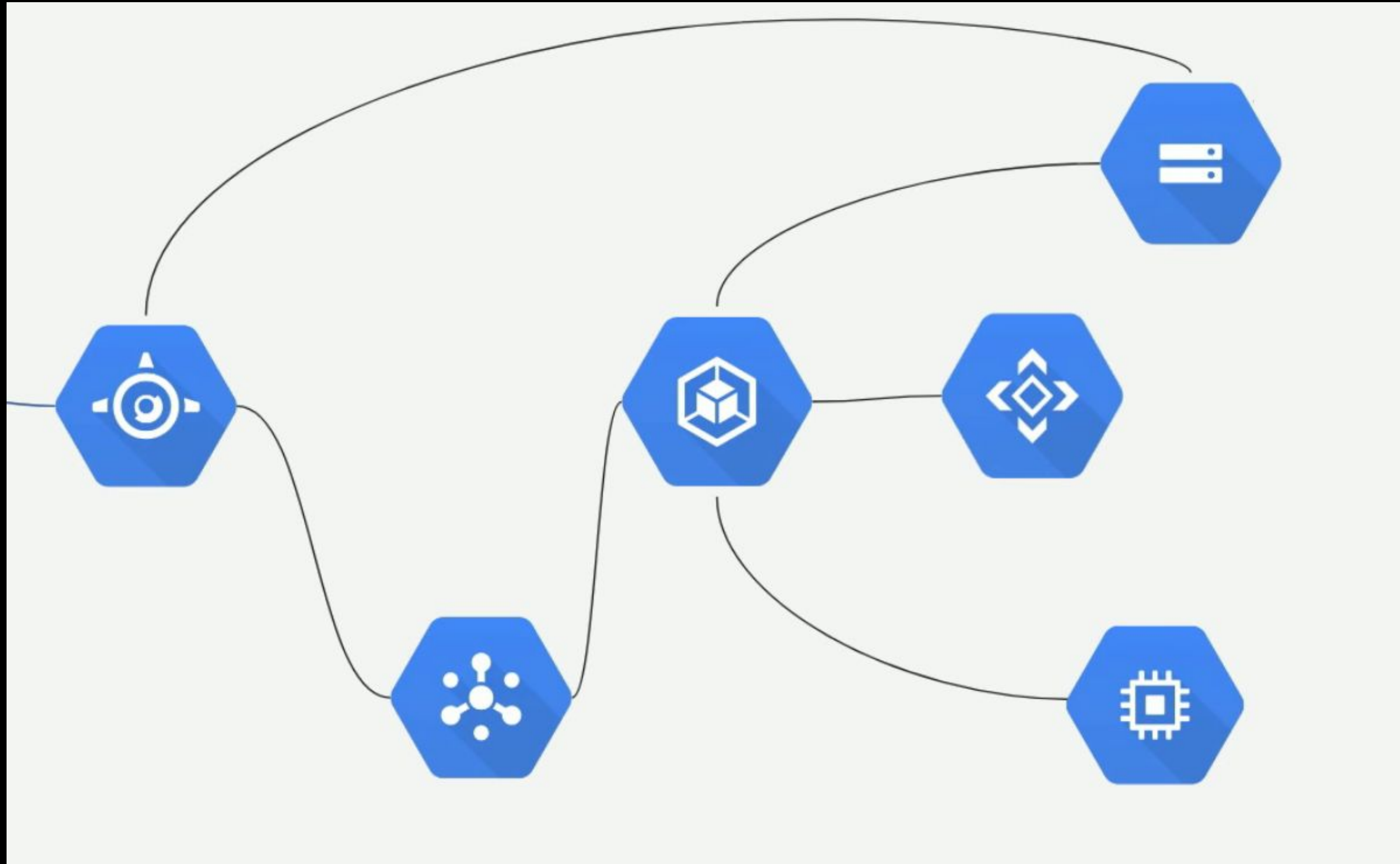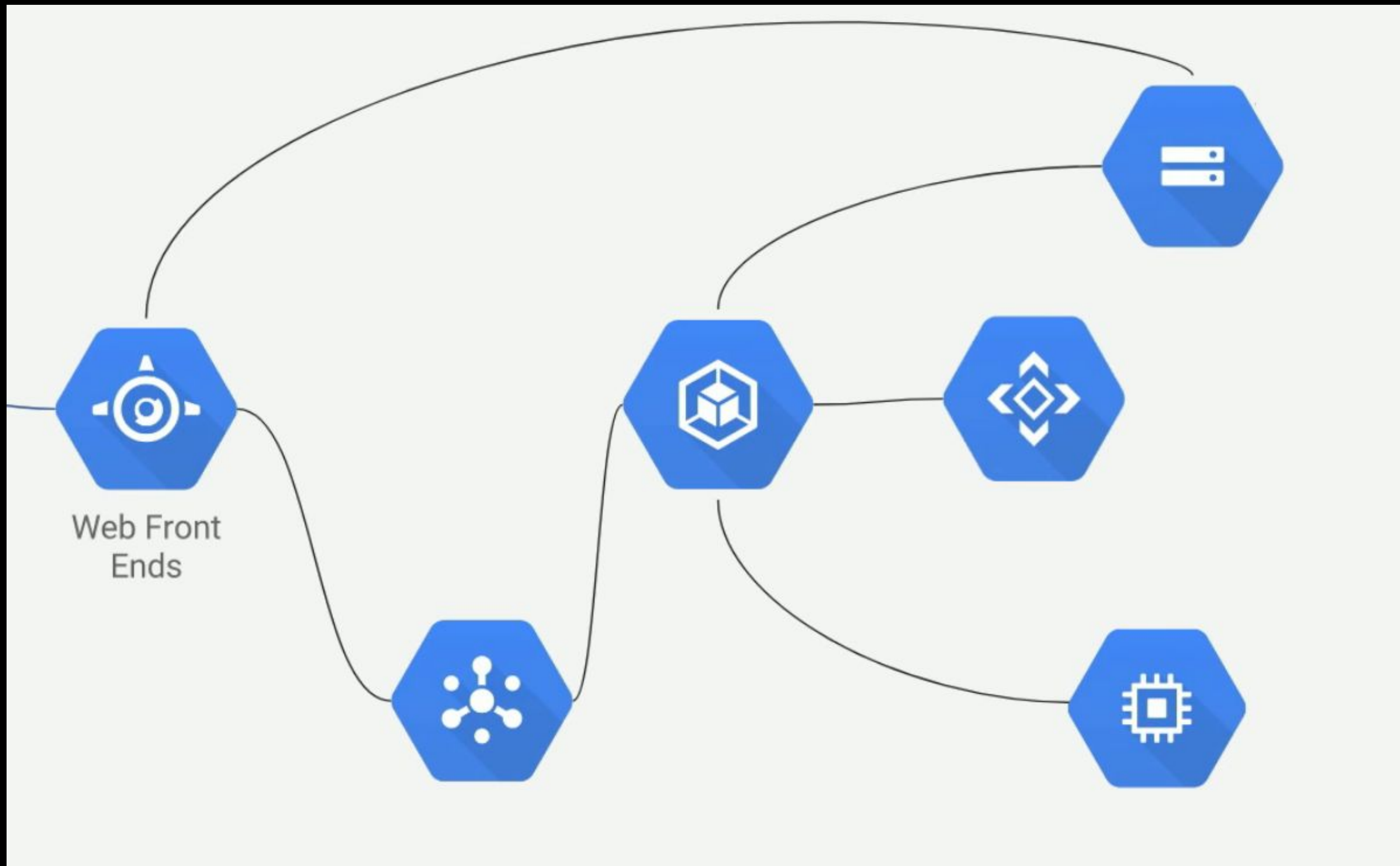
APP ENGINE

TYPICAL ARCHITECTURE

Web Front
Ends

TYPICAL ARCHITECTURE
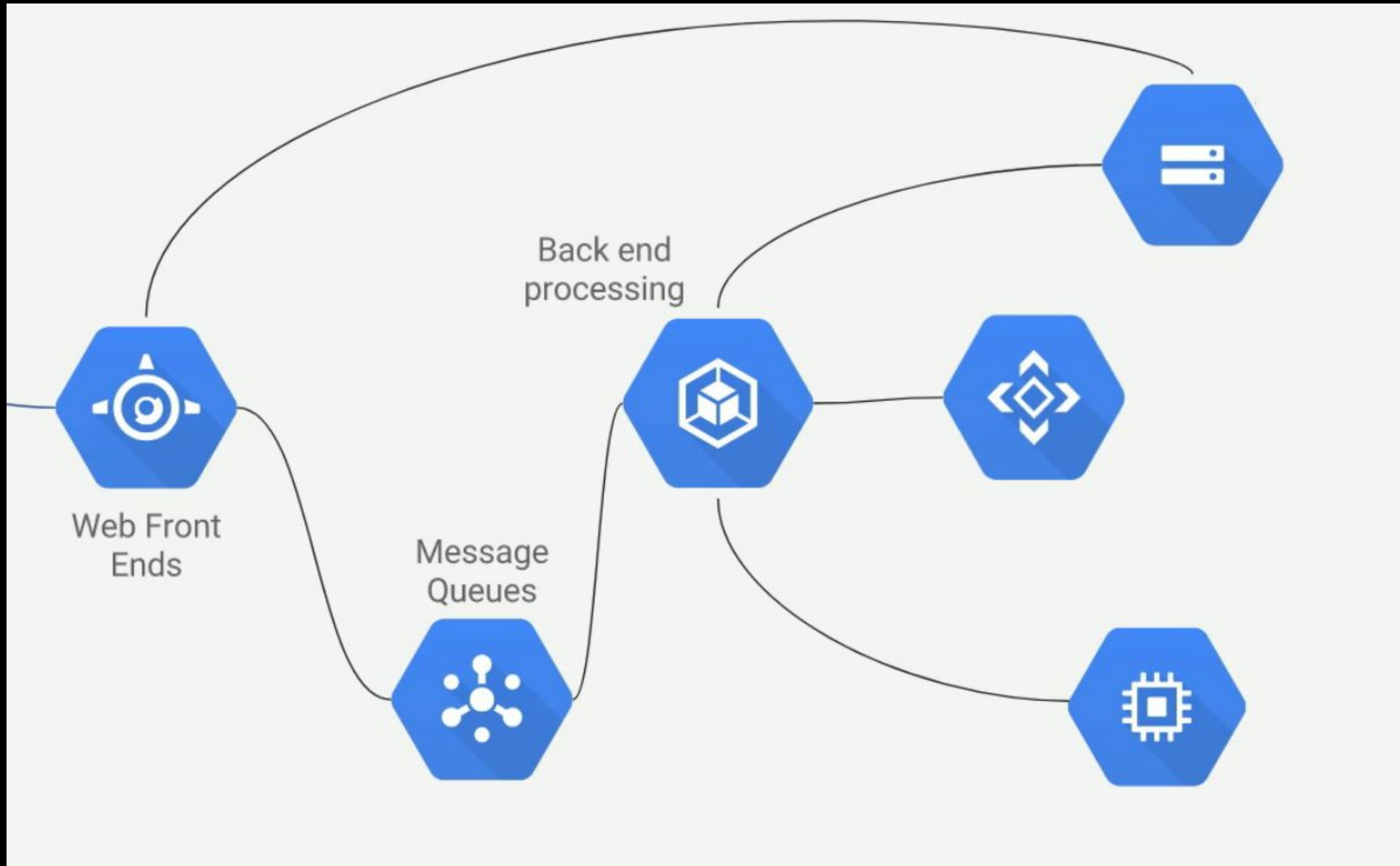

Web Front Ends

Message Queues
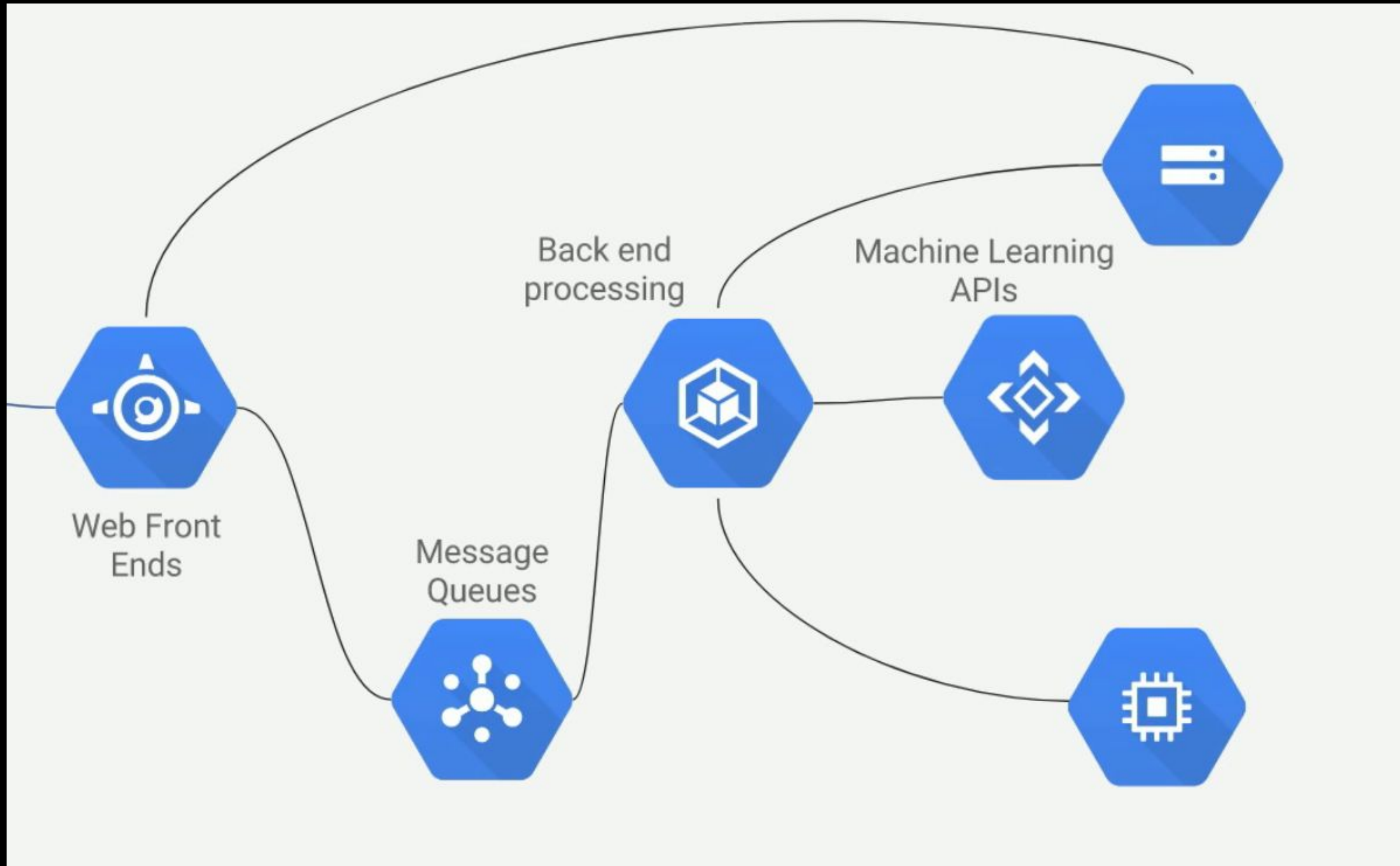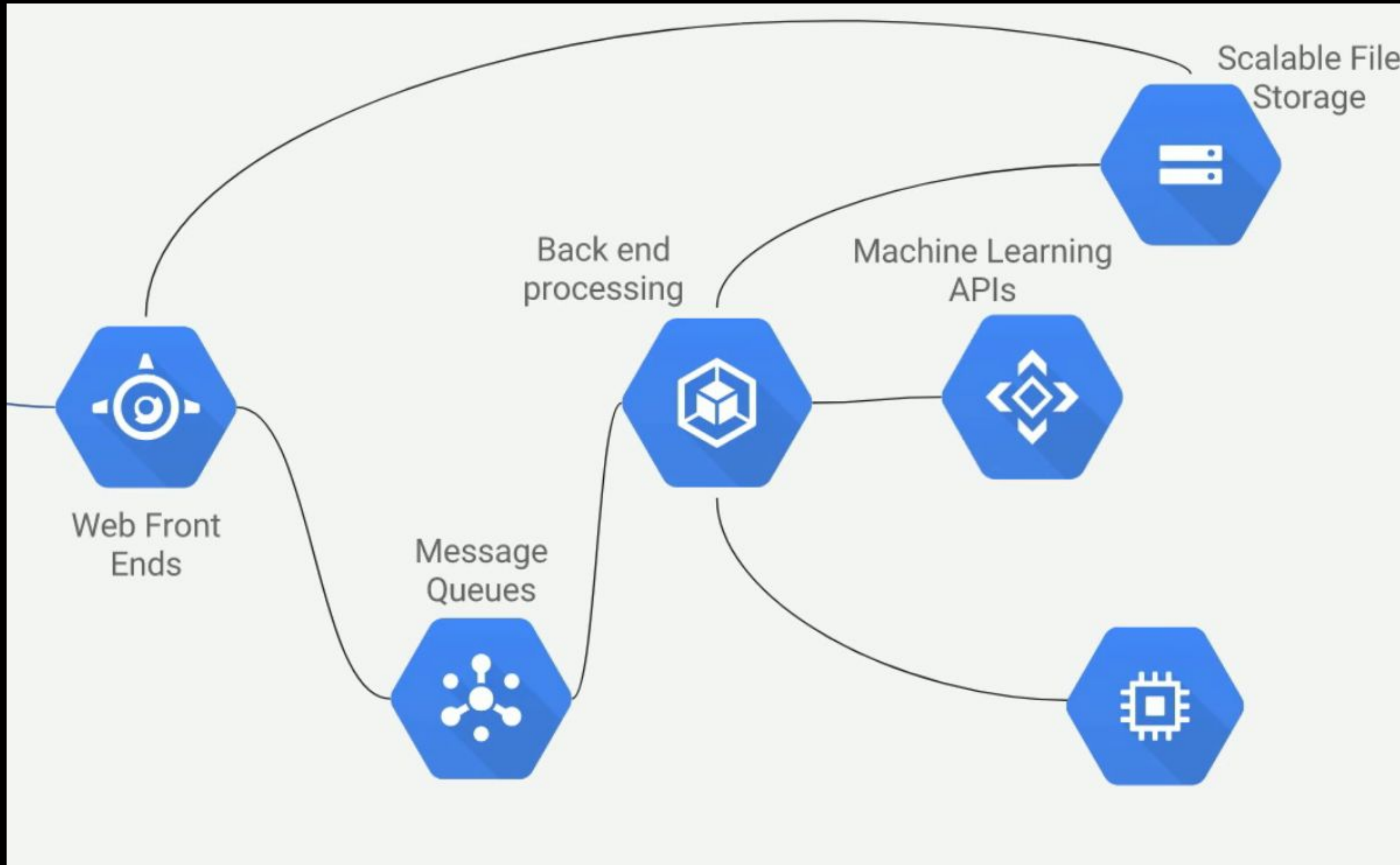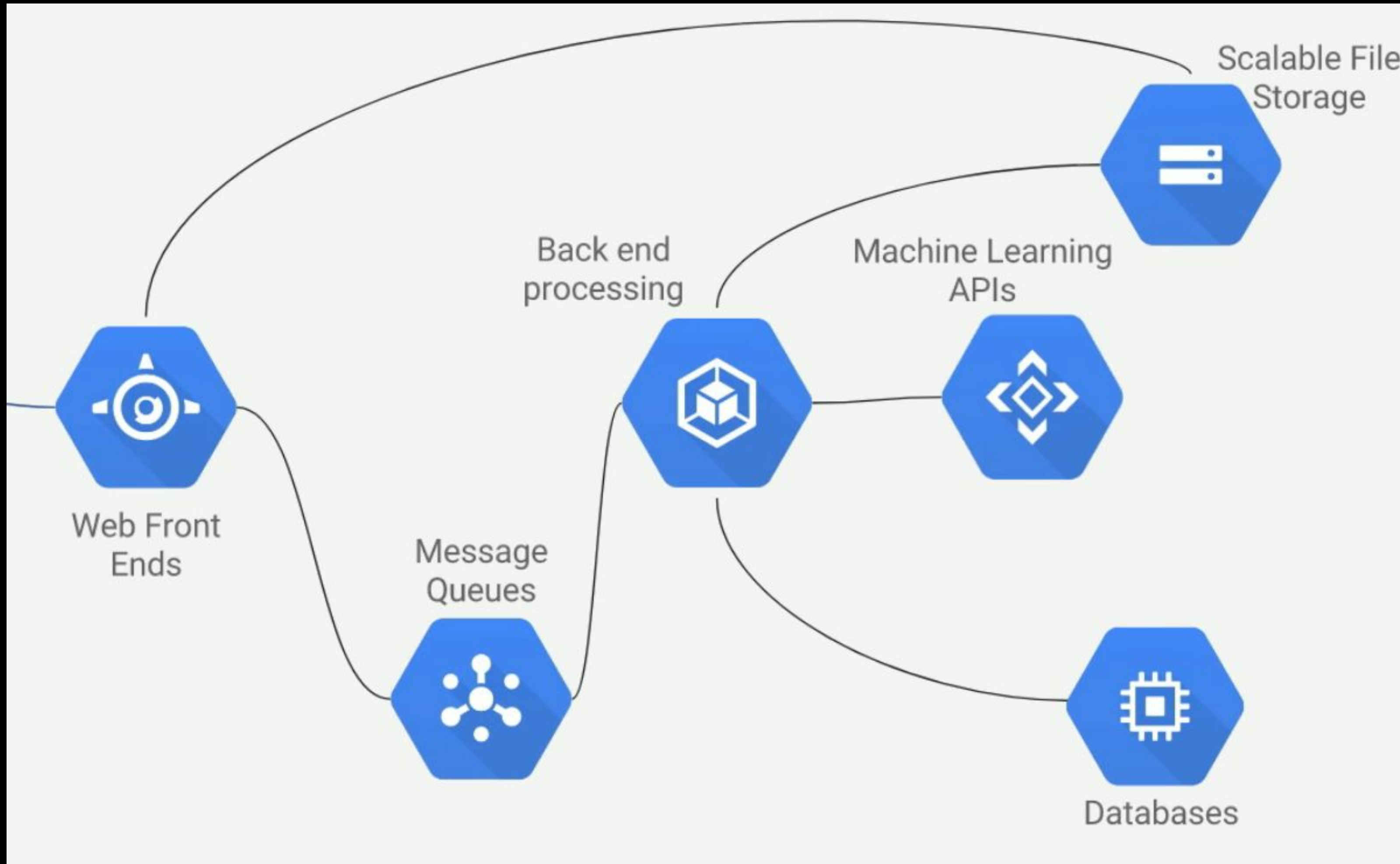
TYPICAL ARCHITECTURE

TYPICAL ARCHITECTURE

TYPICAL ARCHITECTURE

Scalable File Storage

Back end processing

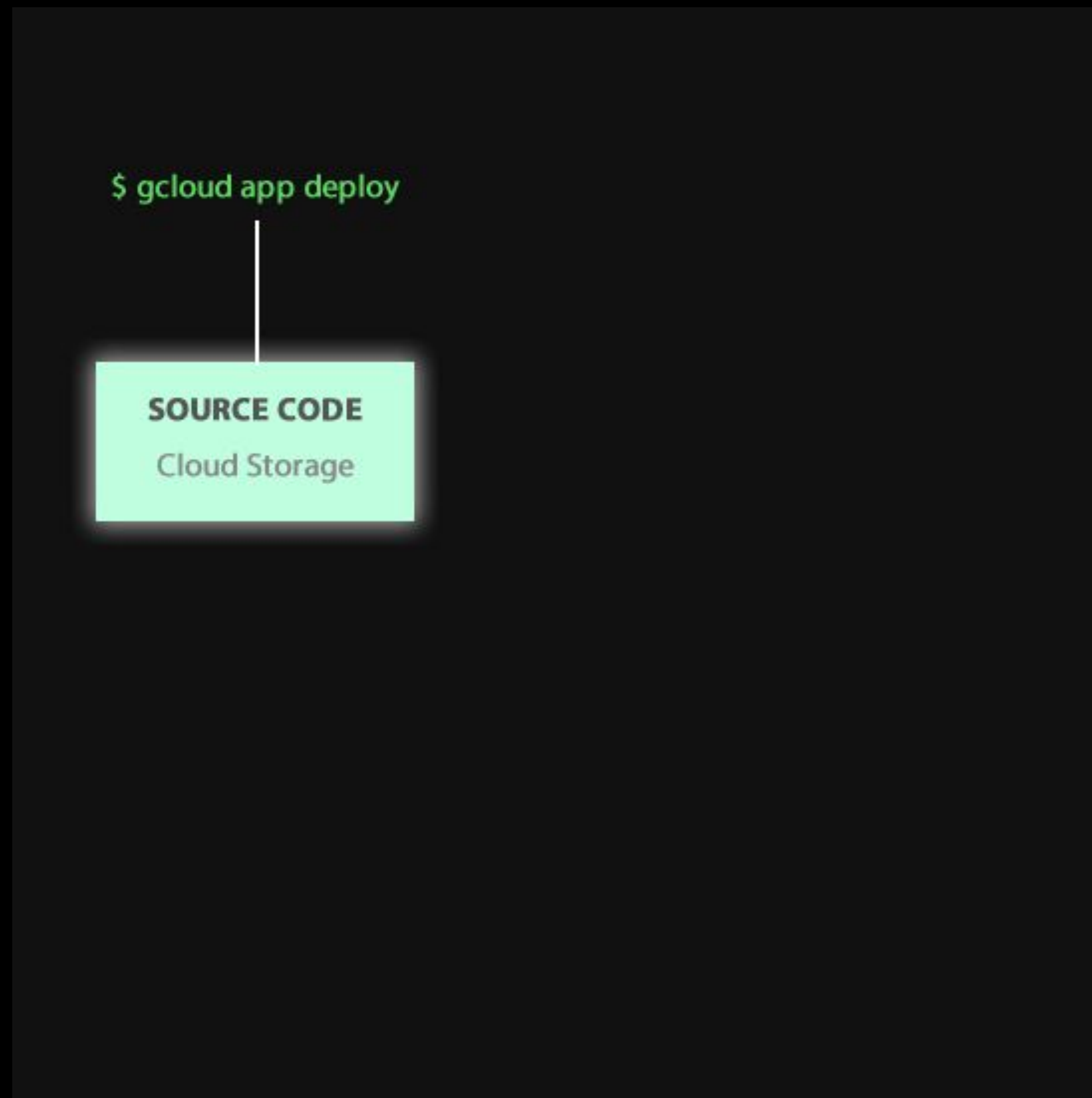Machine Learning APIs

Web Front Ends

Message Queues

TYPICAL ARCHITECTURE

APP DEPLOYMENT

```
$ gcloud app deploy
```

A single line of code starts the deployment from gcloud

APP DEPLOYMENT
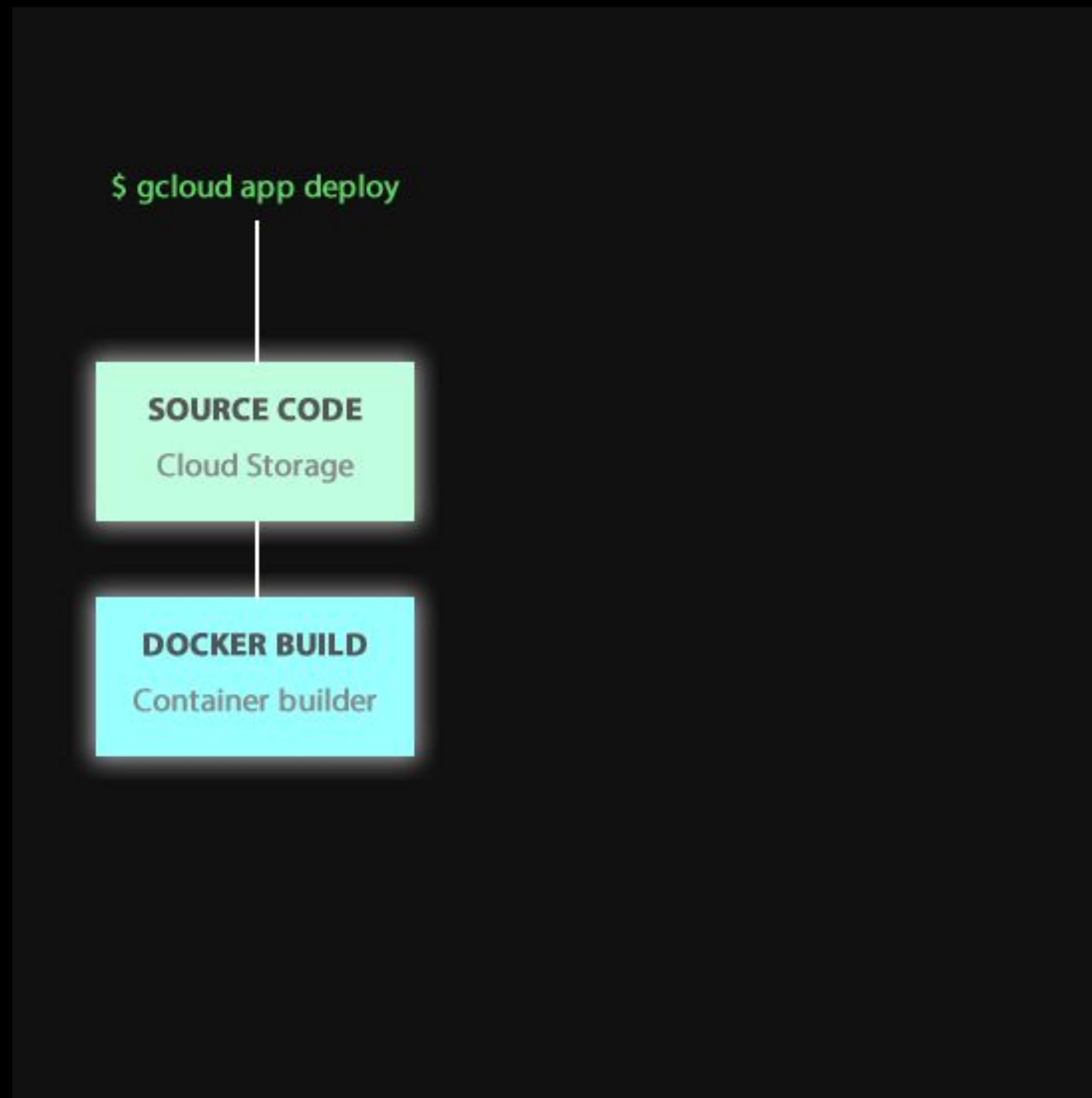
```
$ gcloud app deploy
```

SOURCE CODE
Cloud Storage

Code is uploaded to
Google cloud storage

APP DEPLOYMENT

```
$ gcloud app deploy
```

SOURCE CODE
Cloud Storage

DOCKER BUILD
Container builder

The code is built with the runtimes in a Docker container build

APP DEPLOYMENT

```
$ gcloud app deploy
```

SOURCE CODE
Cloud Storage

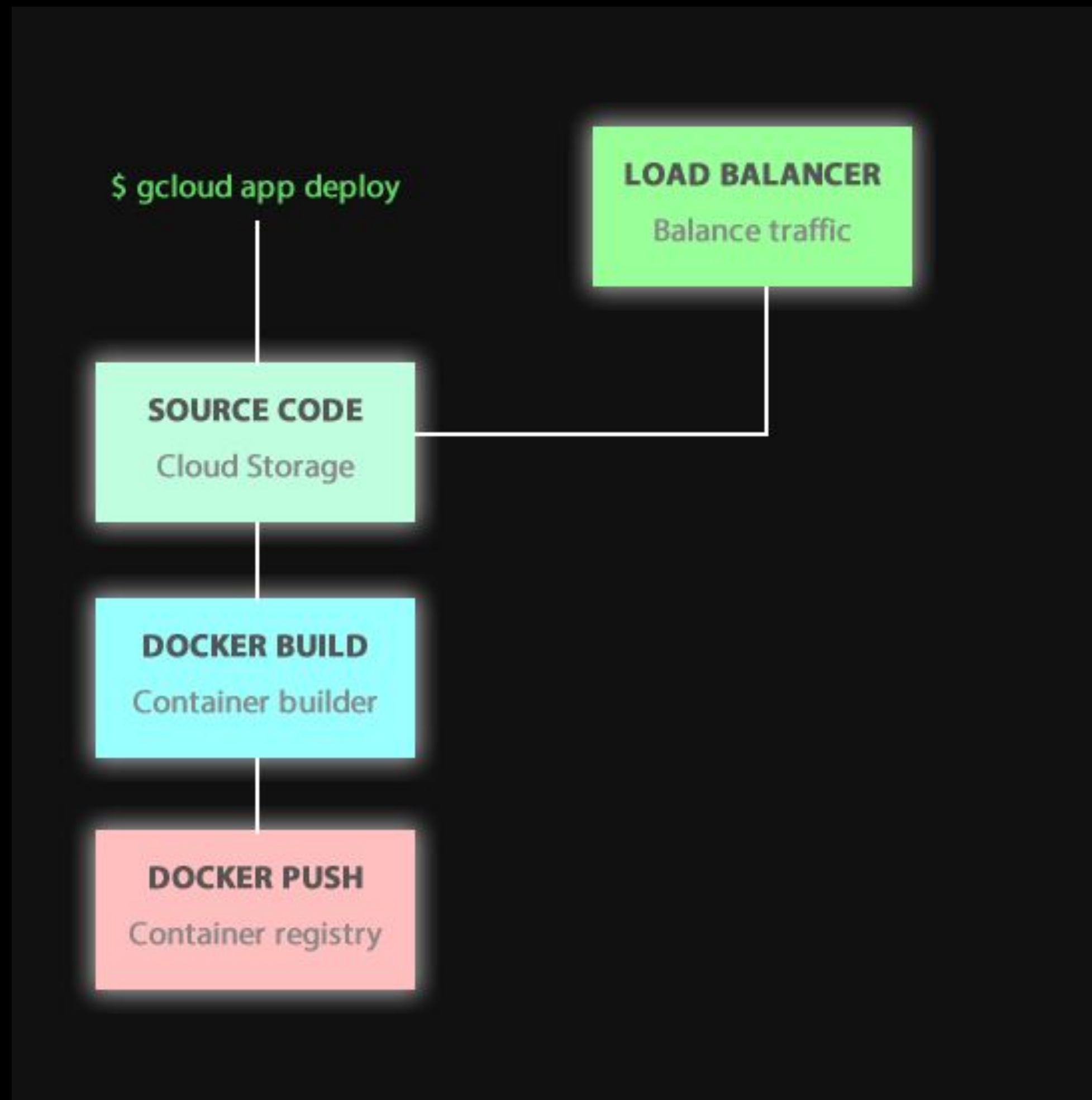**DOCKER BUILD**
Container builder

DOCKER PUSH
Container registry
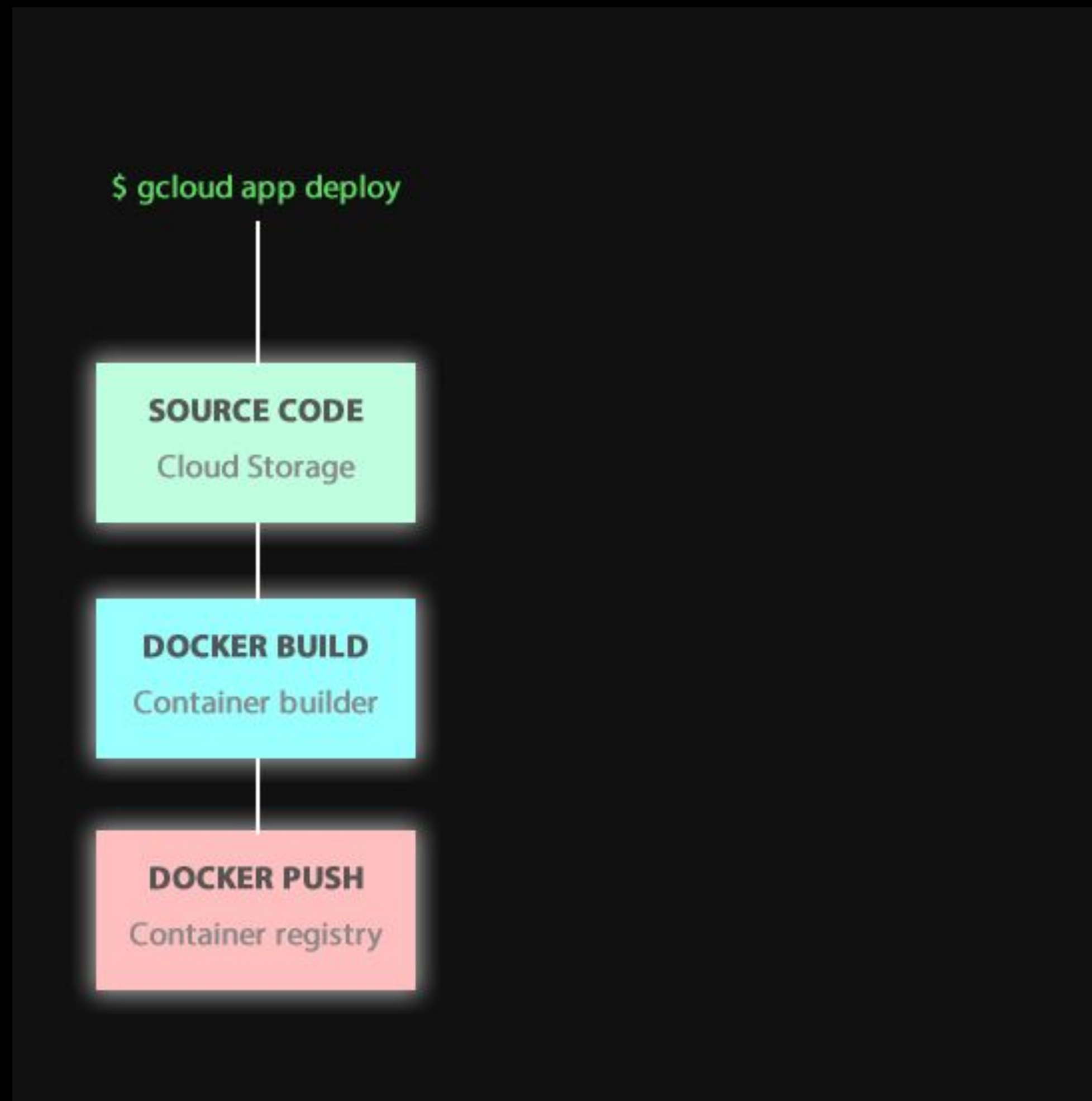
The container is pushed into the Google container registry – this is available across GCP
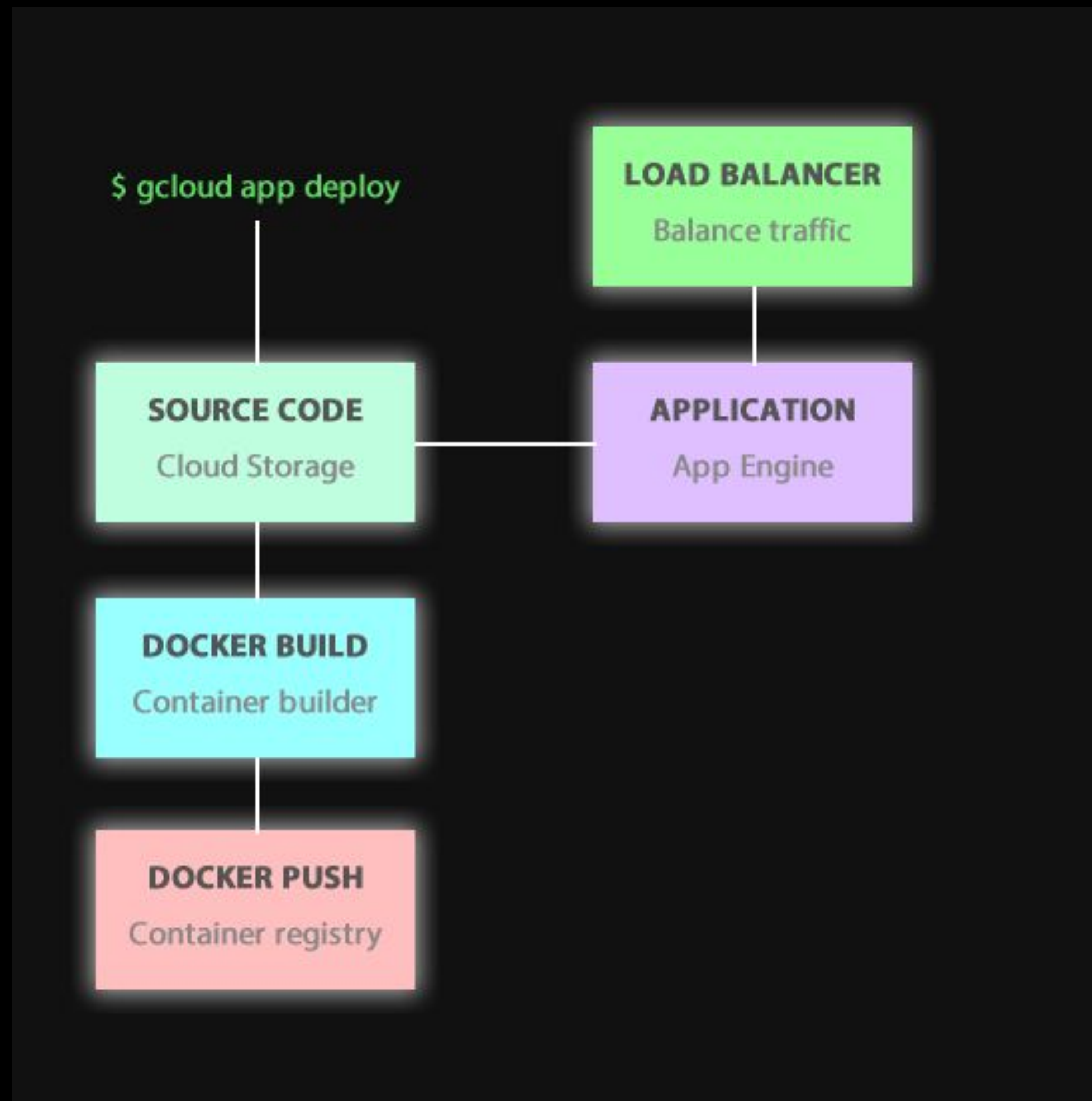
APP DEPLOYMENT

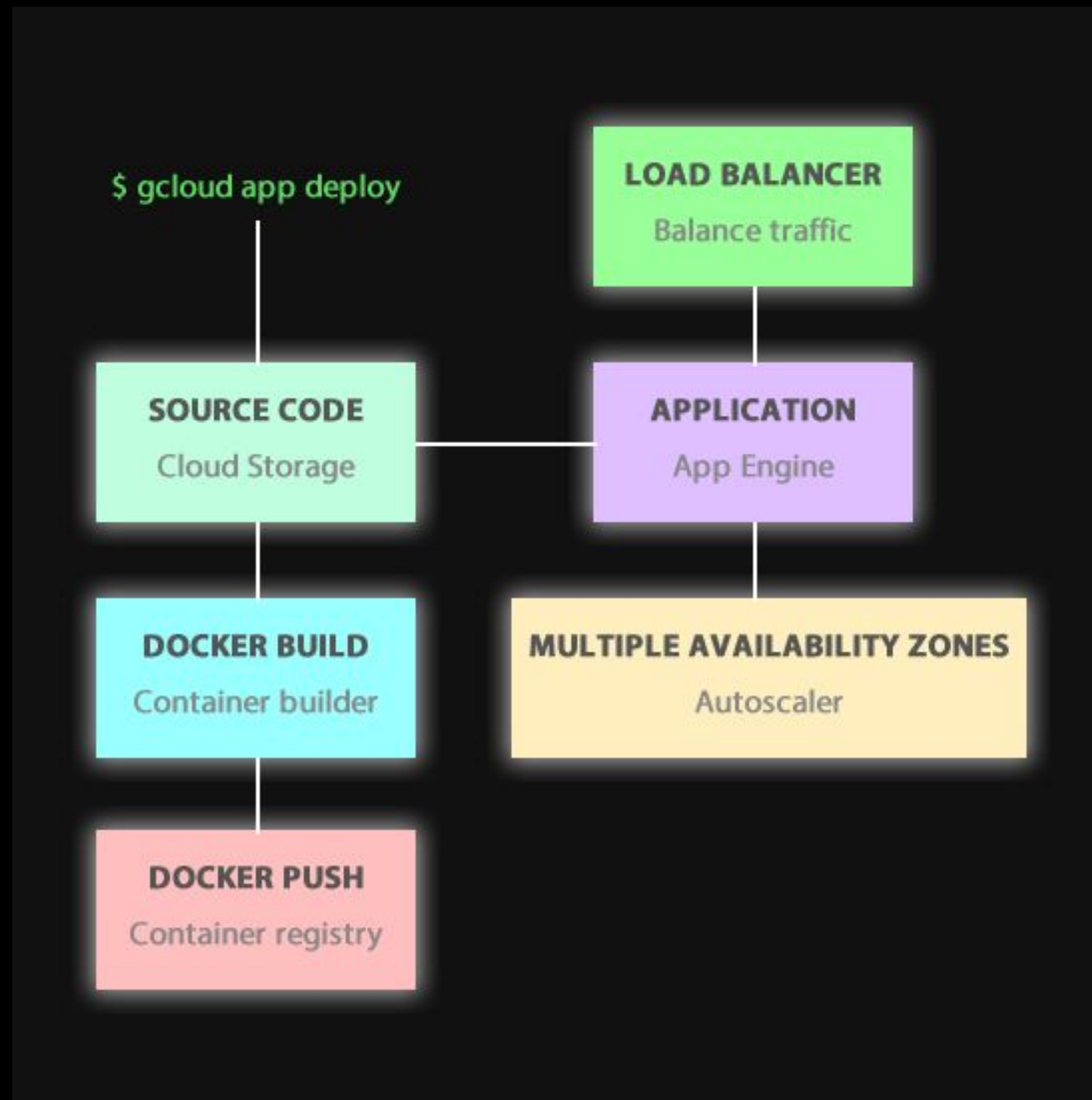A loadbalancer is attached – this is the same across the Google Cloud Platform

APP DEPLOYMENT

The application is run on App Engine based on the requirements

APP DEPLOYMENT

The application is actually run in three availability zones for high availability and replication

New instances are provisioned based on the requirements

DEMO