Google App Engine Fundamentals

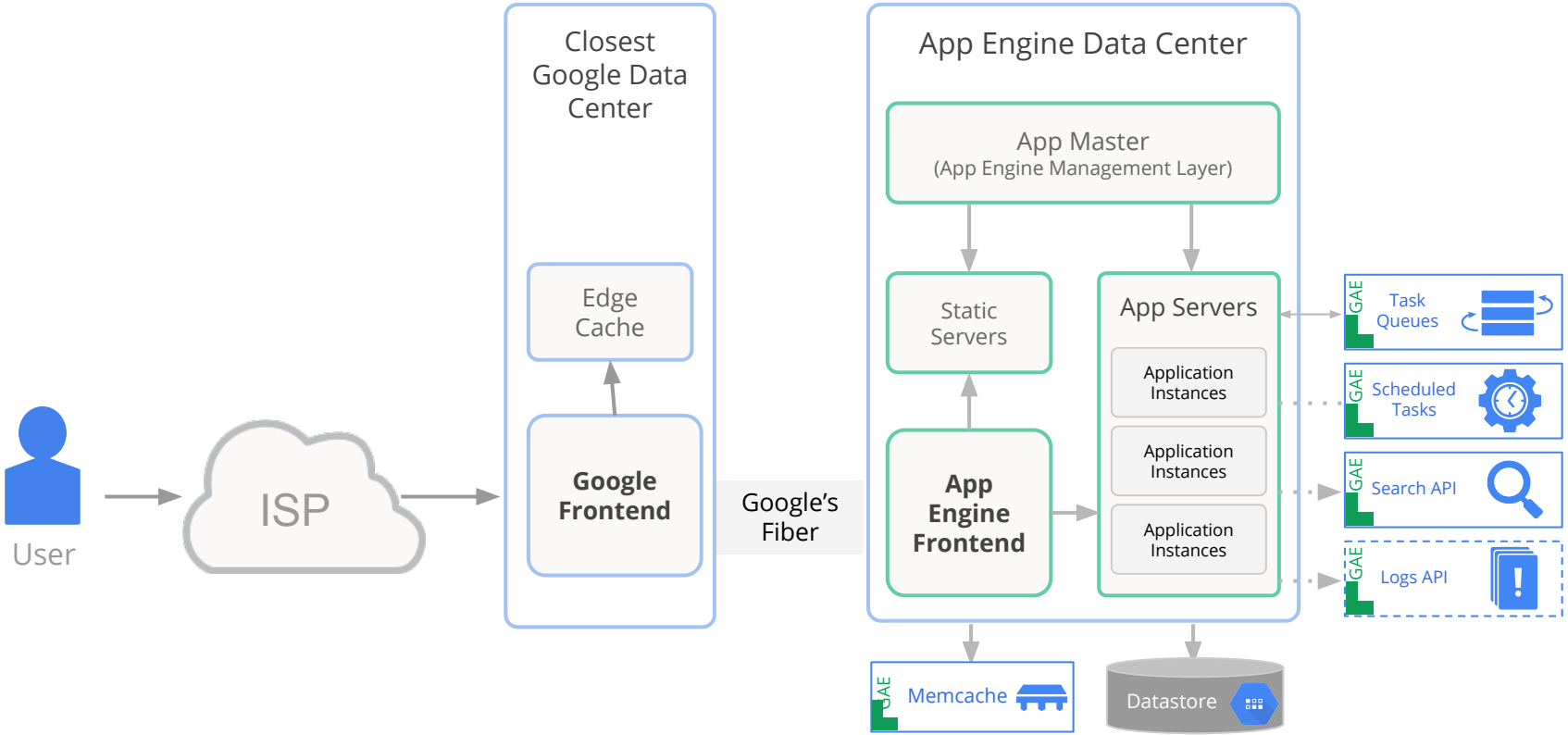
Google App Engine

Google Cloud Platform

Agenda

1. App Engine Architecture
2. Tuning the Auto Scaler
3. Getting Started: “Hello, Auto Scale”

App Engine Architecture

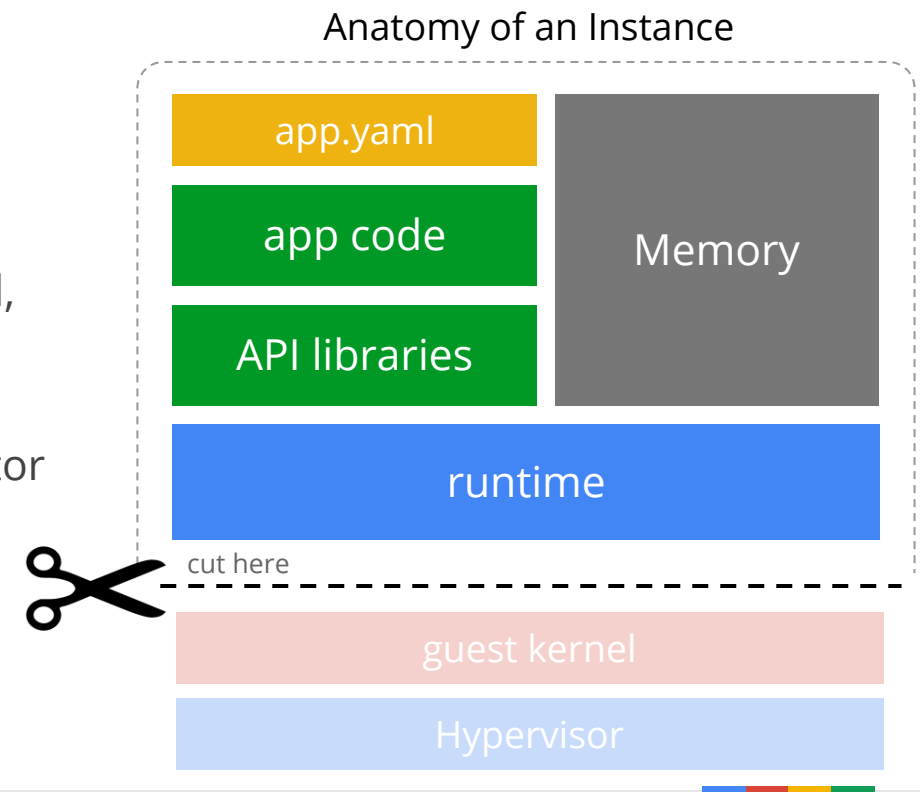


What Is An Application Instance?

Similar to a virtual machine, it provides a runtime environment for your application:

- **Dedicated memory** for your app
- **Fully managed** sandbox
- No burden of managing OS; the overhead, device drivers, security...

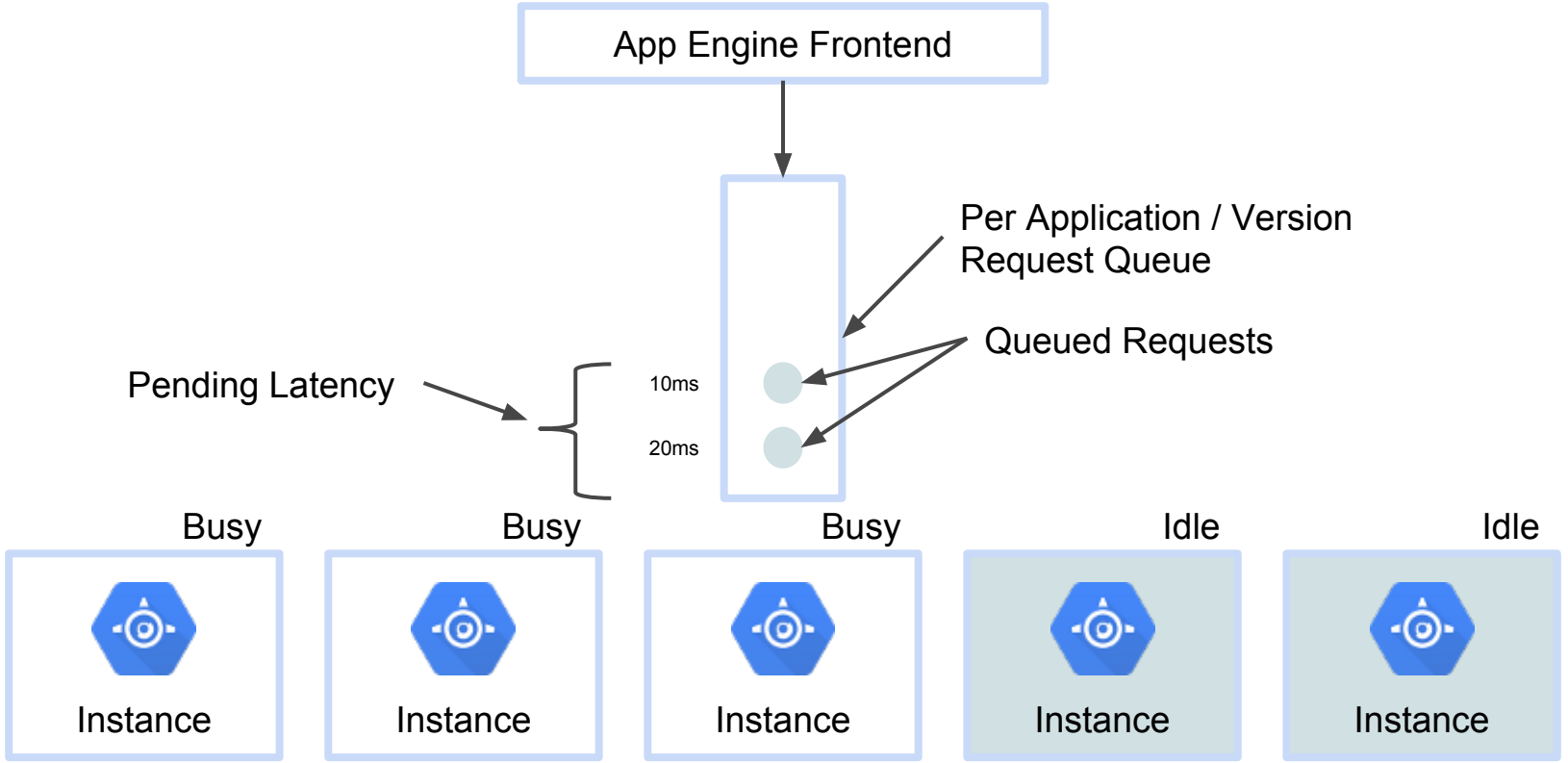
Unlike a virtual machine, AppEngine can monitor user request processing and scale to match demand.



Application Scaling

- **Manual**
  - Continuously running instances allow for complex setup and long requests
- **Basic**
  - Create instances on first request, remove when idle
- **Automatic**
  - Based on request rate, response latencies, and other metrics

Auto Scaling Architecture



Tuning Scaling Parameters

Google App Engine

Manual Scaling Settings

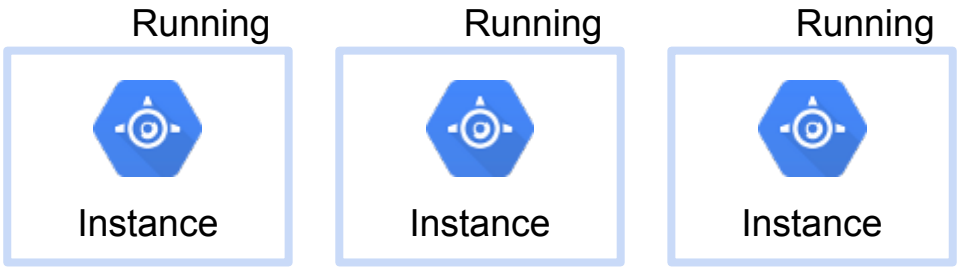
app.yaml

```
manual_scaling:
  instances: 2
```

- Creates a fixed number of instances before requests arrive
- Tasks may take up to 24 hours to complete
- Allows for complex / lengthy startup code to run

Manual Scaling

```
manual_scaling:
  instances: 3
```



Basic Scaling Settings

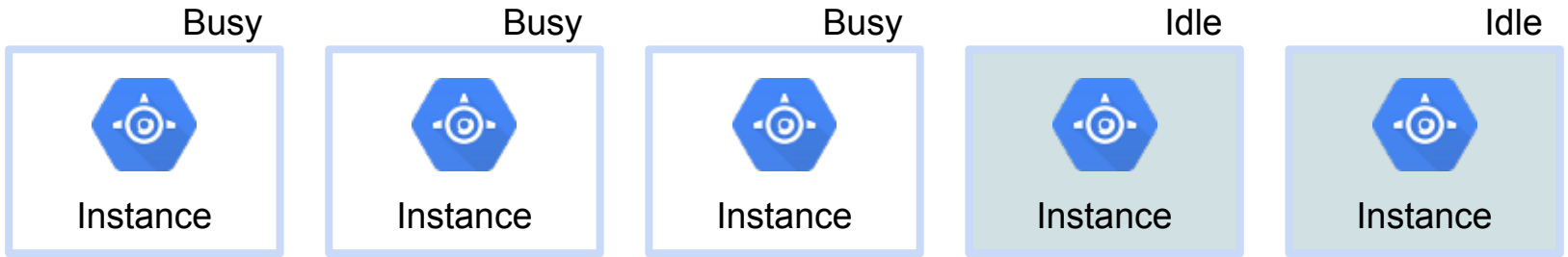
app.yaml

```
basic_scaling:
  max_instances: 5
  idle_timeout: 10s
```

- No instances created prior to first request
- Tasks may take up to 24 hours to complete
- Limits total number of running instances
- When downscaling, idle instances remain no longer than the set timeout (optional, default is 5 minutes)

Basic Scaling

```
basic_scaling:
  max_instances: 5
  idle_timeout: 500ms
```



Automatic Scaling Settings

app.yaml

```
automatic_scaling:
  min_idle_instances: 1
  max_idle_instances: 3
  min_pending_latency: 50ms
  max_pending_latency: automatic
  max_concurrent requests: 8
```

- Instances created and destroyed based on user demand
- 60-second deadline for HTTP requests
- 10-minute deadline for tasks
- Managed by AppEngine heuristics and algorithms
- Set maximum number of concurrent requests per instance (default: 8, maximum:80)

Idle Instances

```
automatic_scaling:
  min_idle_instances: 5          #default:automatic*
  max_idle_instances: 50         #default:automatic*
```

Min - Minimum number of resident, always ready instances

Max - When scaling back, system will keep not more than Max instance idle, ready for a burst of traffic

```
basic_scaling:
  idle_timeout: 500ms
```

Timeout - Amount of time after last processed request before instance shutdown

Pending Latency

```
automatic_scaling:
  min_pending_latency: 500ms      #default:automatic*
  max_pending_latency: 5s        #default:automatic*
```

Min - Minimum time requests are in queue before system considers adding another instance

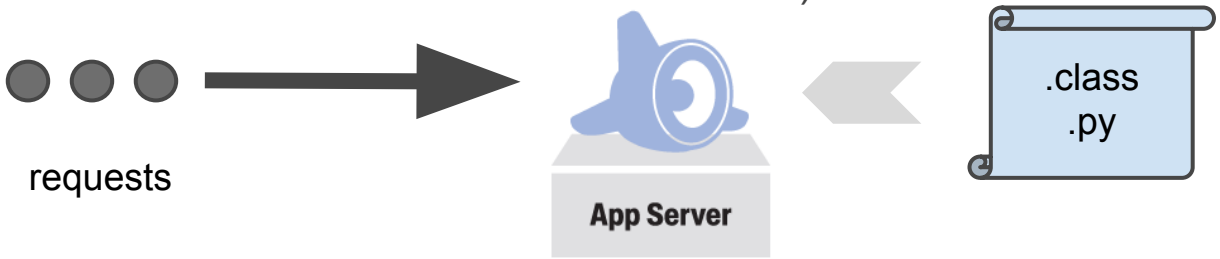
In Between - Heuristics determine whether to wait for a busy instance of create a new one

Max - If requests wait this long, force the creation of new instances. This is the longest permissible time to wait for processing.

Be Careful with Loading Requests

What Are Loading Requests?

- Requests that invoke application instantiation
- App Engine needs to **load libraries** and resources for your app
- Causes latency (hundreds of milliseconds ~ tens of seconds)



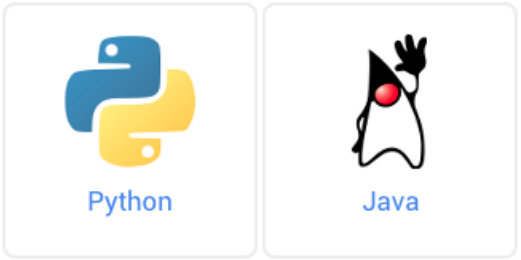
Best Practices

- Use lightweight libraries and frameworks; e.g., Prefer Objectify/Slim3 over Spring
- Lazy loading of libraries
- Package code files into .zip or .jar files

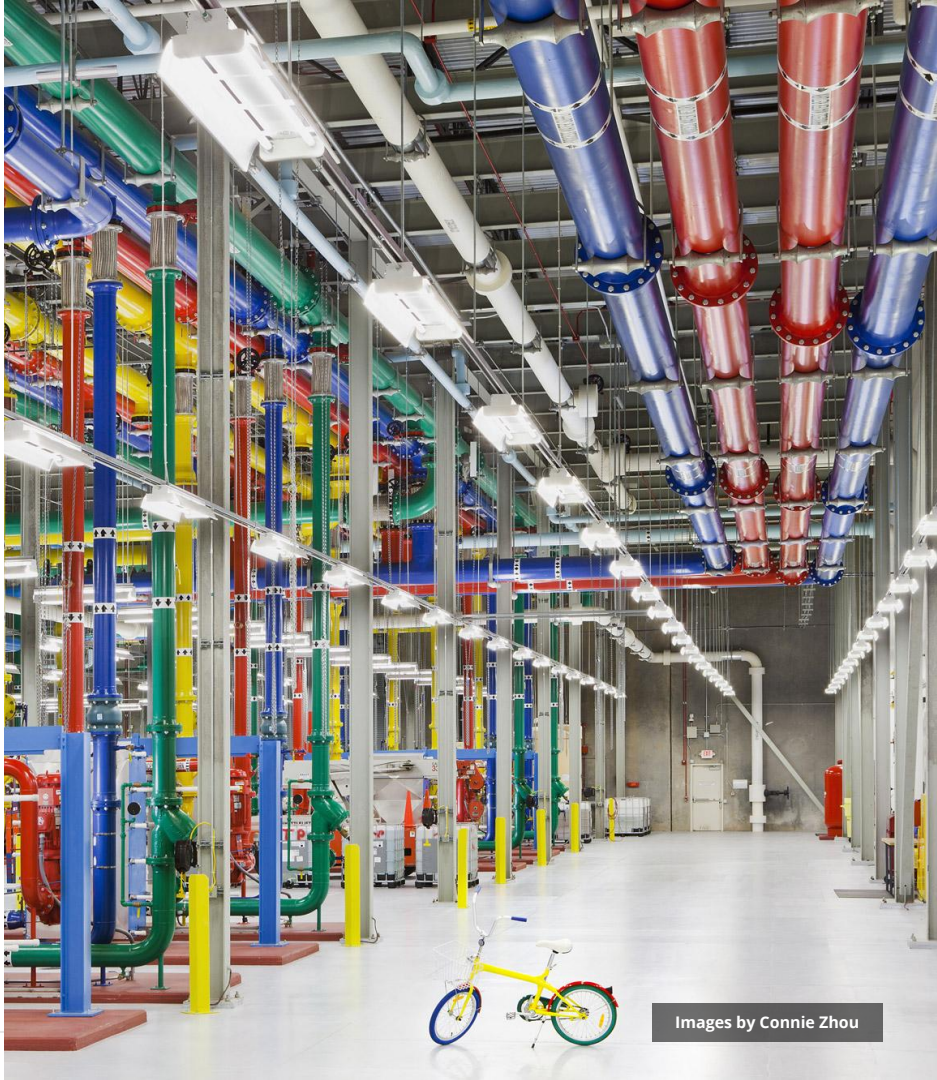
Hands-on

In this exercise, you will:

1. Choose a runtime:

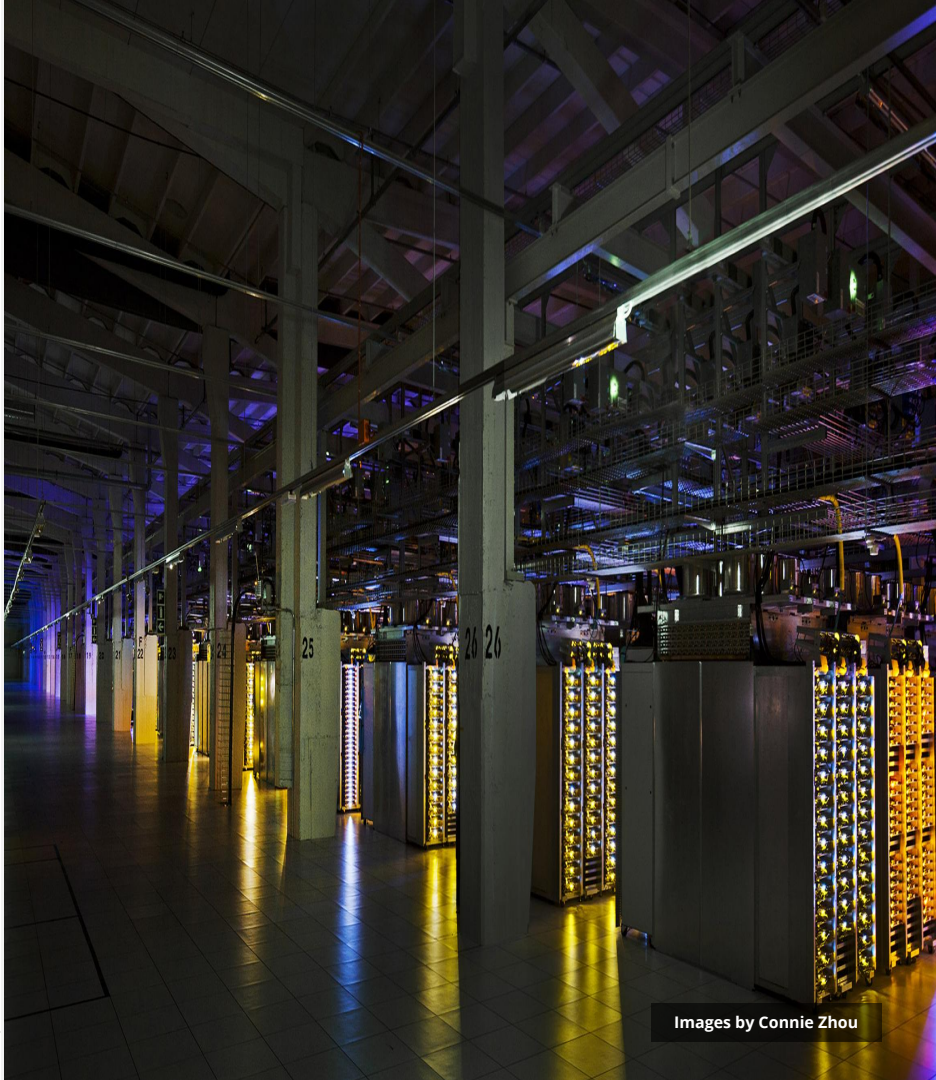


2. Register an App Engine project ID on Cloud Console
3. Create an App Engine app
4. Deploy to GAE



Demo

- Introducing Conference Central




cloud.google.com
Images by Connie Zhou