Deploying Applications on the App Engine Standard Environment

Vitthal Srinivasan
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Standard environment for restrictive lightweight sandbox

Very few supported runtimes

Very fast scaling

Integrate with Cloud Datastore for storage

Memcache for in-memory distributed caching

App Engine Standard Environment

App Engine Environments

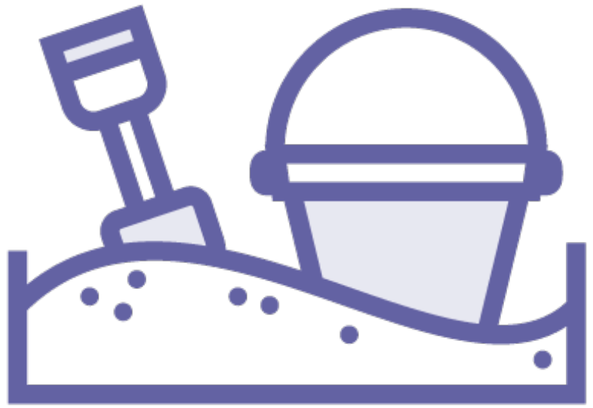
Standard Environment

Flexible Environment

Standard Environment Runtimes

First Generation	Second Generation
Whitelisted extensions and libraries only	Any extension or library
External access via URL Fetch API	Full access to external network
No file system access	Read and write access to /tmp
App Engine-specific runtimes	Unmodified language runtimes
Proprietary sandbox mechanism	gVisor-based container sandbox
Python 2.7, PHP 5.5, Go 1.9	Python 3.7, PHP 7.2, Node.js 8 and 10, Go 1.11, Java 8

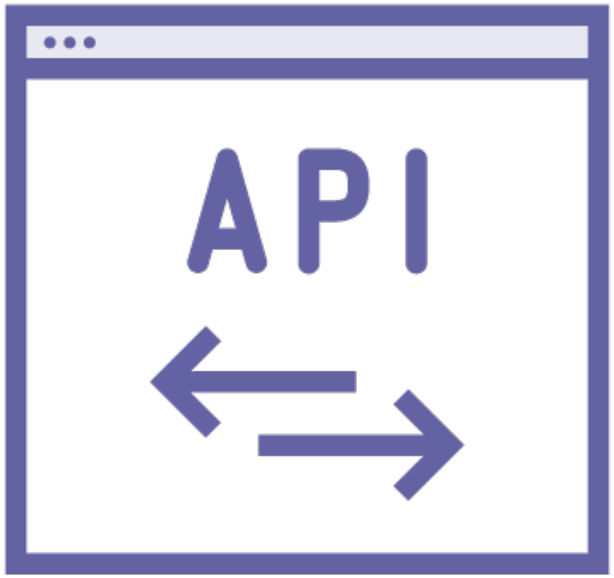
Application Execution



App runs in lightweight instance inside sandbox

Very restrictive

App Engine APIs



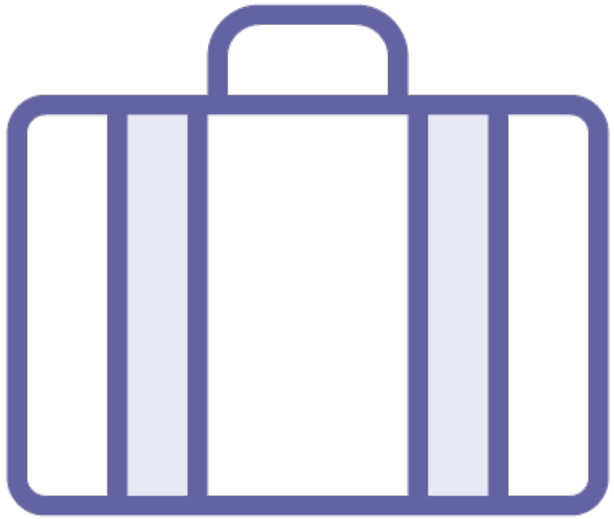
Standard environment uses built-in App Engine APIs (not client libraries)

Datastore

Memcache

Task queues

Local Development



Run in App Engine SDK

Code need not be portable

Scaling

Scaling from zero

Custom-designed, fast scaling

Zero to thousands very quickly

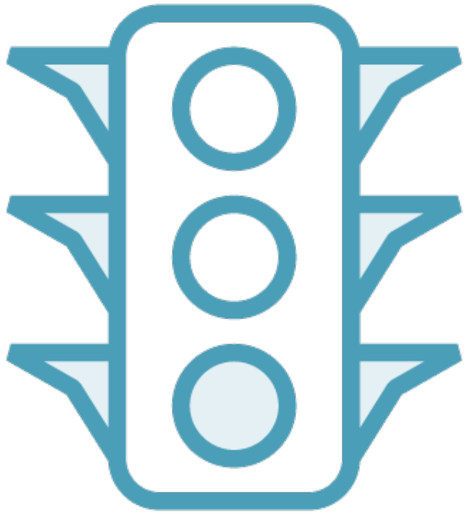
Health Checks

No health checks used or needed

Instances monitored by platform

Traffic Splitting and Migration

Migrating Traffic



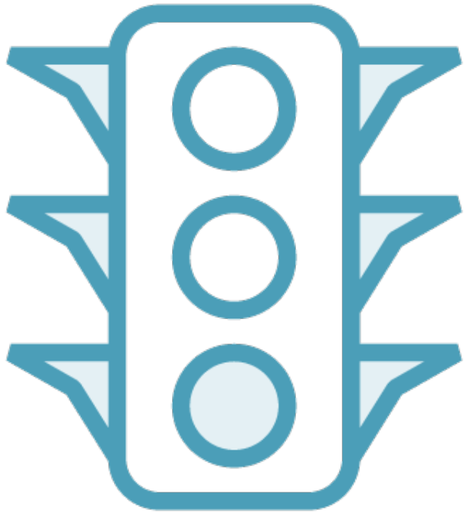
Deploying a new version causes immediate migration of traffic

May cause spike in latency of requests

Migrate traffic gradually

- Add warmup requests to your application
- New instance receive warm up requests before actual traffic

Splitting Traffic



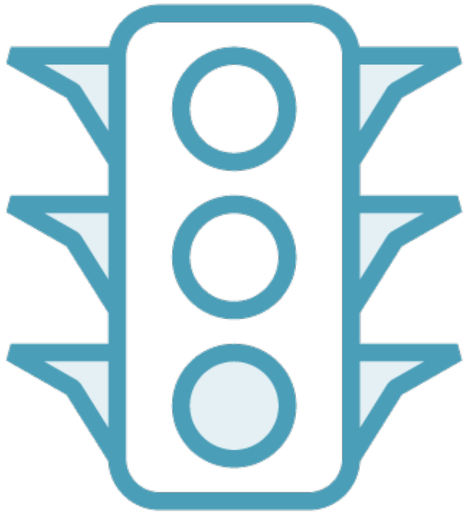
Specify a percentage distribution of traffic

Splitting traffic allows you to conduct A/B testing

Traffic splitting is applied to URLs that do not explicitly target a version

Traffic Splitting

Splitting Traffic Across Multiple Versions



IP address splitting

- Hashes the IP address to a value between 0–999
- Uses that number to route the request

Cookie splitting

- If cookie exists use its value to route
- Otherwise route traffic randomly

Demo

Deploying applications to a specific service

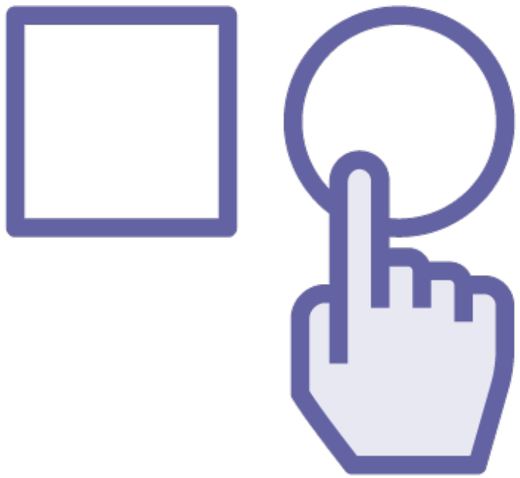
Traffic splitting and migration

Datastore

Google Cloud Datastore

Schemaless NoSQL document database service that provides autoscaling, ACID transactions and SQL-like queries

Choose Datastore For



Highly structured data

- XML, HTML

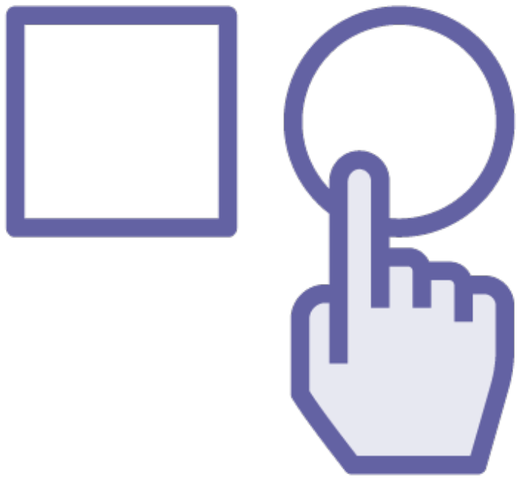
Highly scaled web serving

- Need very fast reads
- No need for interactivity

Transactional (ACID) support

- With non-relational data

Avoid Datastore For



Relational data data needing full SQL support

Unstructured data e.g. blobs

Interactive querying

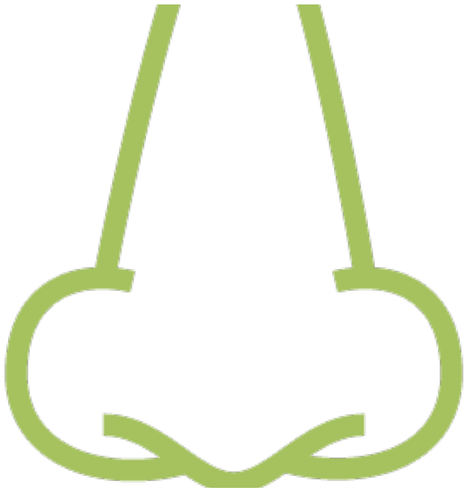
Datastore Data Model

Datastore	RDBMS Equivalent
Kind	Table
Entity	Row
Property	Column
Key	Primary key

Datastore Data Model

Kind is the category of the object i.e. Products, Customers

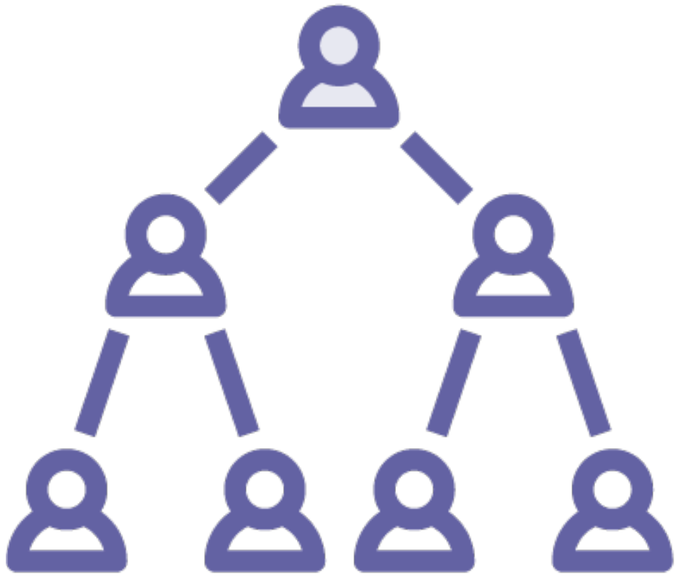
Entity refers to objects of a particular kind



Properties are the attributes of entities

Keys are used to uniquely identify entities

Ancestors



Entities are arranged hierarchically within a namespace

Likes files in a directory system

An entity can have a designated parent

Or can be a root entity

Entity keys are a combination of:

`Namespace:Kind:Identifier:Ancestor_Path`

Demo

Implementing a guestbook review application using Cloud Datastore

Memcache

Memcache

Distributed in-memory cache in front of, or in place of, persistent storage

Memcache

Shared Memcache

Free default for App Engine applications

Provides cache capacity on a best-effort basis

Shared across all App Engine applications

Dedicated Memcache

Provides a fixed cache capacity assigned exclusively to your application

It's billed by the GB-hour of cache size

More predictably and with fewer reads from more costly durable storage

Demo

Integrating caching in the guestbook application using Memcache

Summary

Standard environment for restrictive lightweight sandbox

Very few supported runtimes

Very fast scaling

Integrate with Cloud Datastore for storage

Memcache for in-memory distributed caching