Cloud computing overview & Running code on Google Cloud

Wesley Chun

Developer Advocate, Google
Adjunct CS Faculty, Foothill College
Google Cloud

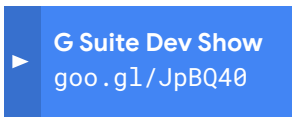
About the speaker

Developer Advocate, Google Cloud

- **Mission**: enable current **and future** developers everywhere to be successful using Google Cloud and other Google developer tools & APIs
- **Videos**: host of the *G Suite Dev Show* on YouTube
- **Blogs**: developers.googleblog.com & gsuite-developers.googleblog.com
- **Twitters**: @wescpy, @GoogleDevs, @GSuiteDevs

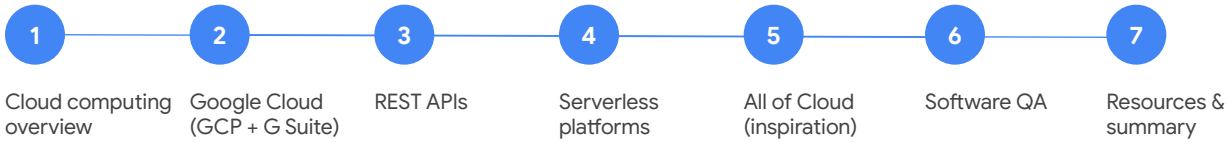
Previous experience / background

- Software **engineer** & architect for 20+ years
- One of the original Yahoo!Mail engineers
- **Author** of bestselling "Core Python" books (corepython.com)
- Technical trainer, **teacher**, instructor since 1983 (Computer Science, C, Linux, Python)
- Fellow of the Python Software Foundation
- AB (Math/CS) & CMP (Music/Piano), UC Berkeley and MSCS, UC Santa Barbara
- Adjunct Computer Science Faculty, Foothill College (Silicon Valley)



Why and Agenda

- Cloud has taken industry by storm (all?)
- Not enough cloud computing in higher-ed curriculum
- How Google Cloud can help with your courses/research
- How cloud can be used with software QA & testing!
- Help prep next-generation cloud-ready workforce



Cloud computing overview | Google Cloud (GCP + G Suite) | REST APIs | Serverless platforms | All of Cloud (inspiration) | Software QA | Resources & summary

Google Cloud

1 Cloud computing overview

All you need to know about the cloud

What is cloud computing?

Cloud service levels/"pillars"

SaaS
Software as a Service

- ***G Suite (Google Apps)***
  Yahoo!Mail, Hotmail, Salesforce, Netsuite
- ***Google Apps Script***
  Salesforce1/force.com

PaaS
Platform as a Service

- ***Google App Engine, Cloud Functions***
  Heroku, Cloud Foundry, Engine Yard, AWS Lambda
- ***Google BigQuery, Cloud SQL, Cloud Datastore, NL, Vision, Pub/Sub***
  AWS Kinesis, RDS; Windows Azure SQL, Docker

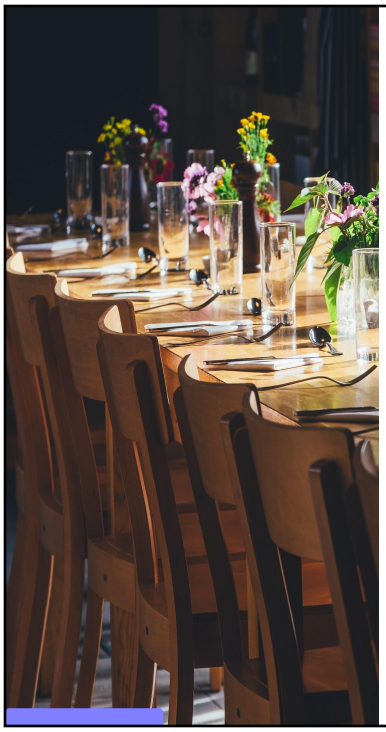
IaaS
Infrastructure as a Service

- ***Google Compute Engine, Google Cloud Storage***
  AWS EC2 & S3; Rackspace; Joyent

Summary of responsibility

on-prem all you, no cloud	IaaS Infrastructure as a Service	PaaS Platform as a Service	SaaS Software as a Service
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
OS	OS	OS	OS
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

Managed by YOU Managed by cloud vendor



Imagine you're hosting a party...

on-prem (DIY)	IaaS (Compute Engine)	PaaS (App Engine/GCF)	SaaS (G Suite)
Pick theme	Pick theme	Pick theme	Pick theme
Plan party	Plan party	Plan party	Hire planner
Find space	Rent hall	Rent hall	Rent hall
Cook	Cook	Hire Caterer	Hire caterer
On-call	On-call	Hire manager	Hire manager

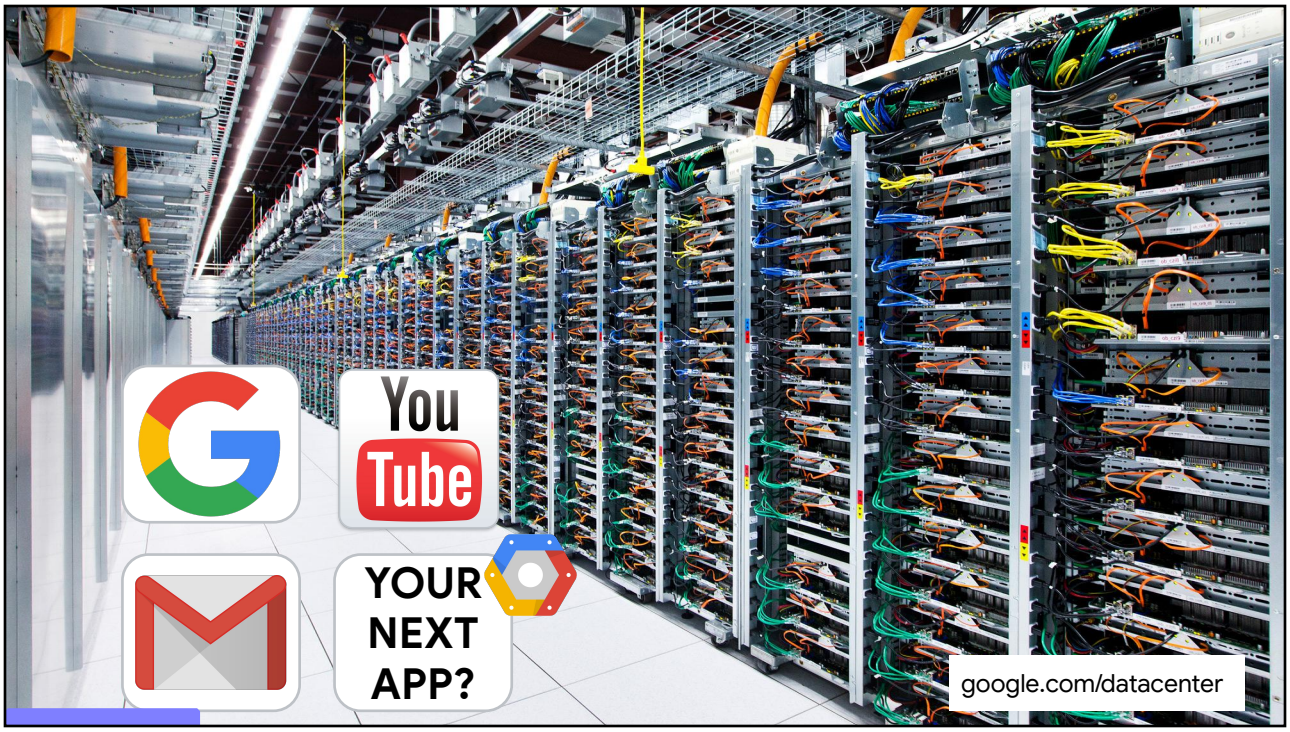
Theme - Spec/Reqs
Logistics - Design app
Space - Provision HW
Food - Build & serve app
Manage - Manage app

How can Google Cloud help in higher ed?

- What can we provide faculty, researchers, IT staff, students?
  - Virtual machines, GPUs, and variety of data storage
  - Ability to craft & design your own network/subnet
  - Pre-trained machine learning models
  - Container-hosting, ML build & deploy infrastructure
  - Serverless compute & data services
  - Additional or emergency compute & storage capacity
  - Productivity tools students already use (G Suite)
  - Education grants (use our cloud w/o personal credit cards)

2 Introduction to Google Cloud

GCP and G Suite tools & APIs



Google Cloud

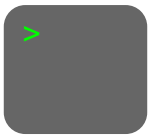
Google Cloud Platform

G Suite

Google Cloud Platform vs. G Suite

Compute
(running code)

Running Code: Compute Engine



Google Compute Engine delivers configurable virtual machines of all shapes and sizes, from "micro" to 416 vCPUs, 11.776 TB RAM, 256 TB HDD or SSD disk; GPUs & TPUs

(Debian, CentOS, CoreOS, SUSE, Red Hat Enterprise Linux, Ubuntu, FreeBSD; Windows Server 2008 R2, 2012 R2, 2016, 1803, 1809, 1903/2019, 1909)

cloud.google.com/compute

Running Code: App Engine

Got a great app idea? Now what? VMs? Operating systems? Big disk? Web servers? Load balancing? Database servers? Autoscaling? With **Google App Engine**, you don't think about those. Just upload your code; ***we*** do everything else.

cloud.google.com/appengine

Running Code: Cloud Functions

Don't have an entire app? Just want to deploy small microservices or "RPCs" online globally? That's what **Google Cloud Functions** are for! (+Firebase version for mobile apps)

cloud.google.com/functions
firebase.google.com/products/functions

Running Code: Cloud Run

Got a containerized app? Want its flexibility along with the convenience of serverless that's fully-managed plus auto-scales? **Google Cloud Run** is *exactly* what you're looking for!

cloud.google.com/run

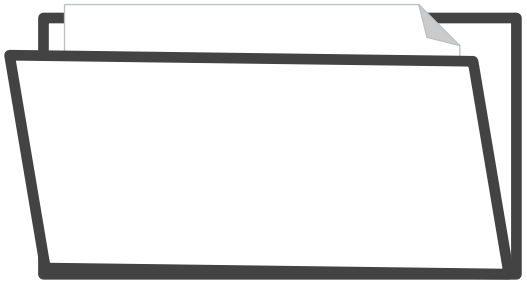
Managed containers: Kubernetes Engine

Got a containerized application?
Google Kubernetes Engine is an enterprise-grade, fully-managed container orchestration service.

cloud.google.com/kubernetes-engine

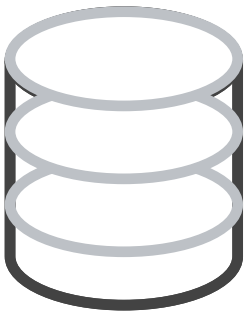
Storage
(where to put your data)

Storing Data: Cloud Storage & Cloud Filestore



cloud.google.com/storage
cloud.google.com/filestore

Storing Data: Cloud SQL



SQL servers in the cloud

High-performance, fully-managed

600MB to 416GB RAM; up to 64 vCPUs

Up to 10 TB storage; 40,000 IOPS

Types:
- MySQL
- Postgres
- SQLServer (2019)

cloud.google.com/sql

Storing Data: Cloud Datastore

Cloud Datastore is a fully-managed, highly-scalable **NoSQL** database for your web and mobile applications

cloud.google.com/datastore

Storing Data: Firebase

Firebase data is stored as **JSON** & synchronized in **real-time** to every connected client; other tools + FB == v2 mobile development platform

iOS C++

firebase.google.com

Storing Data: Cloud Firestore

The best of both worlds: the next generation of **Cloud Datastore** (w/product rebrand) plus features from the **Firebase** realtime database

cloud.google.com/firestore

Big data
(move, process, and analyze your data)

Storing *and* Analyzing Data: BigQuery

Google BigQuery is a fast, highly scalable, fully-managed data warehouse in the cloud for analytics with built-in machine learning (BQML); issue SQL queries across multi-terabytes of data

cloud.google.com/bigquery

Passing Data & Events: Pub/Sub

Google Pub/Sub: a fast, highly scalable, fully-managed multi fan-in/fan-out publisher-subscriber queuing system for messaging & event ingestion (and processing)

cloud.google.com/pubsub

Data processing: Dataflow

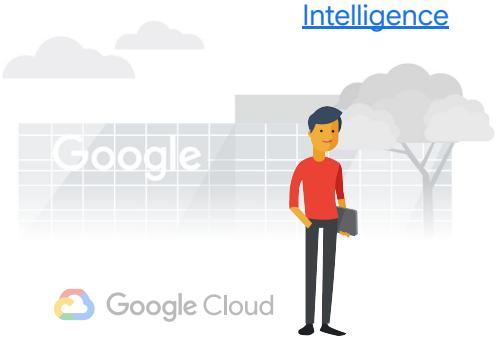
Google Dataflow: a flexible, parallel, automated, scalable, dynamic work- & resource-balancing, fully-managed stream-based as well as batched data processing pipeline service

cloud.google.com/dataflow

Machine Learning
(analyze your data)

GCP Machine Learning APIs

Vision | Video Intelligence | Speech (S2T & T2S) | Natural Language | Translation



- Gain insights from data using GCP's *pre-trained* machine learning models
- Leverage the same technology as Google Translate, Photos, and Assistant
- Requires ZERO prior knowledge of ML
- If you can call an API, you can use AI/ML!
- cloud.google.com/products/ai/building-blocks

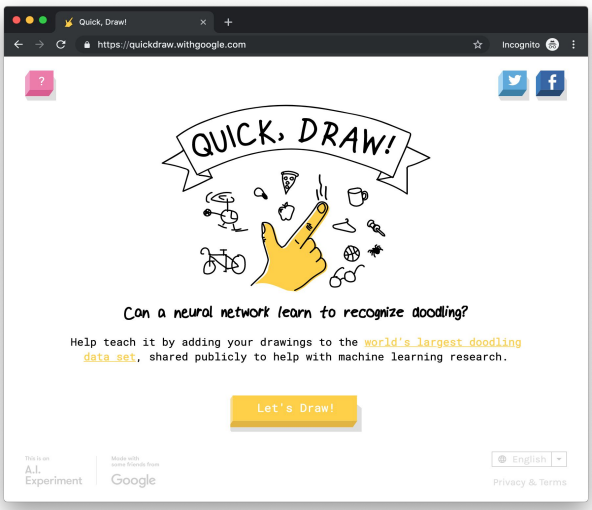
Machine Learning: Cloud Vision & Video Intelligence

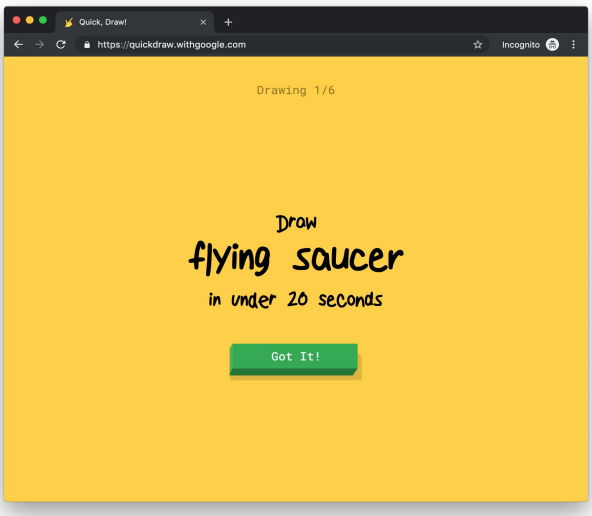
Google Cloud Vision & Video Intelligence APIs enable developers to extract metadata & understand the content of images & videos, making them searchable & discoverable.

cloud.google.com/vision
cloud.google.com/video-intelligence

Google Cloud Vision demo "experiment"

experiments.withgoogle.com/quick-draw





Google Cloud

quickdraw.withgoogle.com

Machine Learning: Cloud Natural Language

Google Cloud Natural Language API reveals the structure and meaning of text, performing sentiment analysis, content classification, entity extraction, and syntactical structure analysis; multi-lingual

cloud.google.com/language

Machine Learning: AutoML

AutoML: a suite of cloud APIs for developers with limited machine learning expertise; chooses the best models & allows for further training of those models for your data
(Translation, Vision, Natural Language, Video Intelligence, Tables)

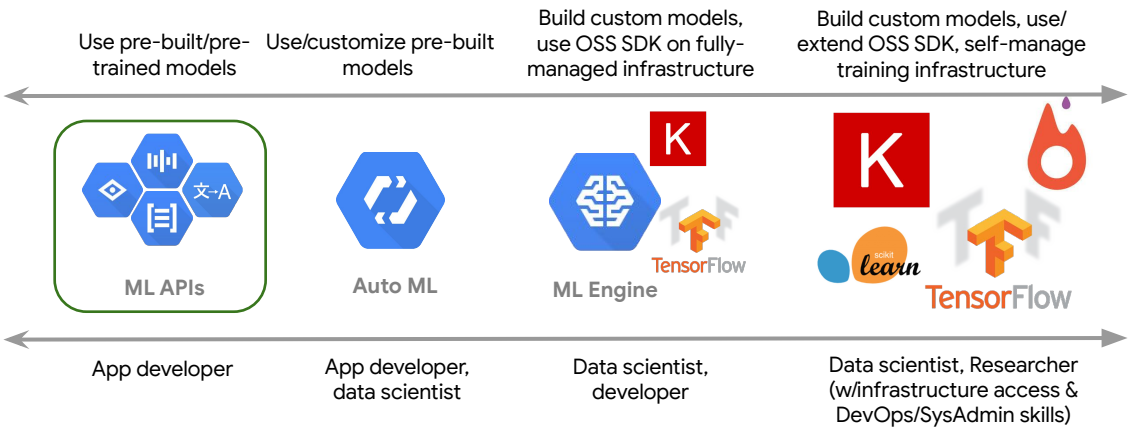
cloud.google.com/automl
cloud.google.com/automl-tables

Machine Learning: Cloud ML Engine

Google Cloud Machine Learning Engine is a managed service that lets you build, train, and deploy machine learning models (scikit-learn, XGBoost, Keras, TensorFlow), then make predictions with trained models

cloud.google.com/ml-engine

Full Spectrum of AI & ML Offerings



Google Cloud

G Suite
(collaborate & communicate)

G Suite: Google Sheets

Sheets API gives you programmatic access to spreadsheets; perform (w/code) almost any action you can do from the web interface as a user

developers.google.com/sheets

G Suite: Google Docs & Slides

Docs & Slides APIs give you access to read or write documents and presentations programmatically so you can auto-generate them with data integrated from various sources

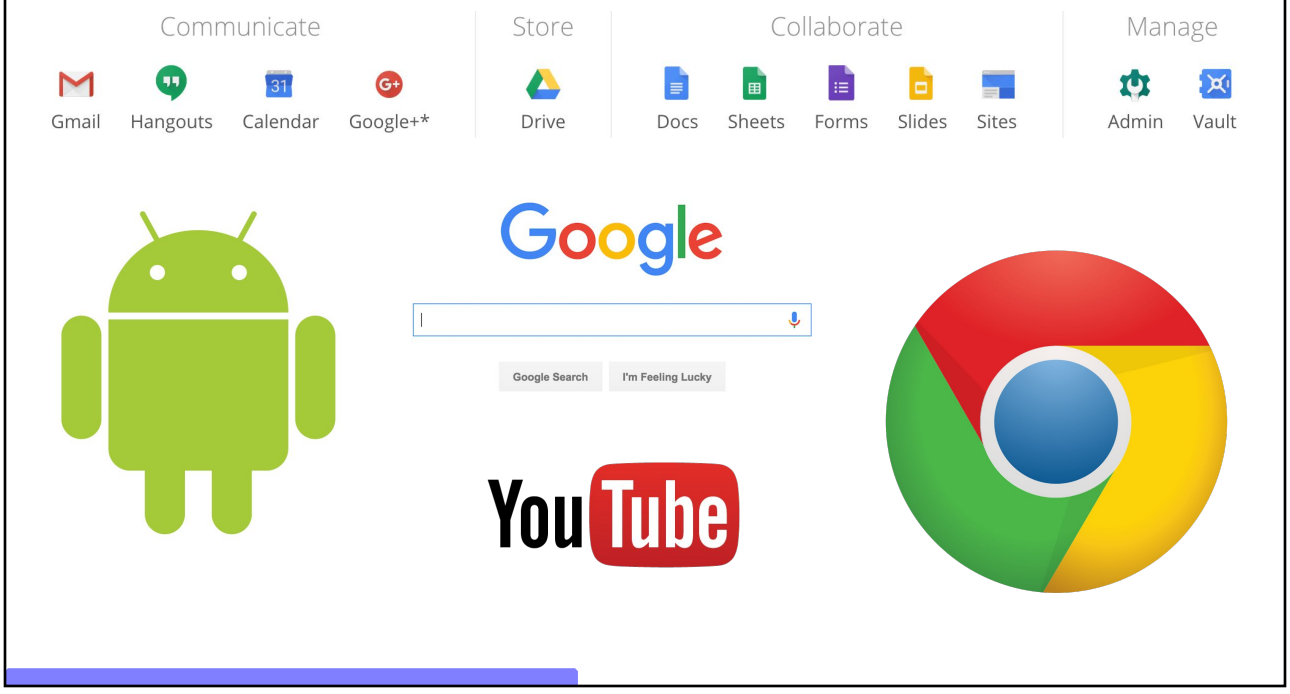
developers.google.com/docs
developers.google.com/slides

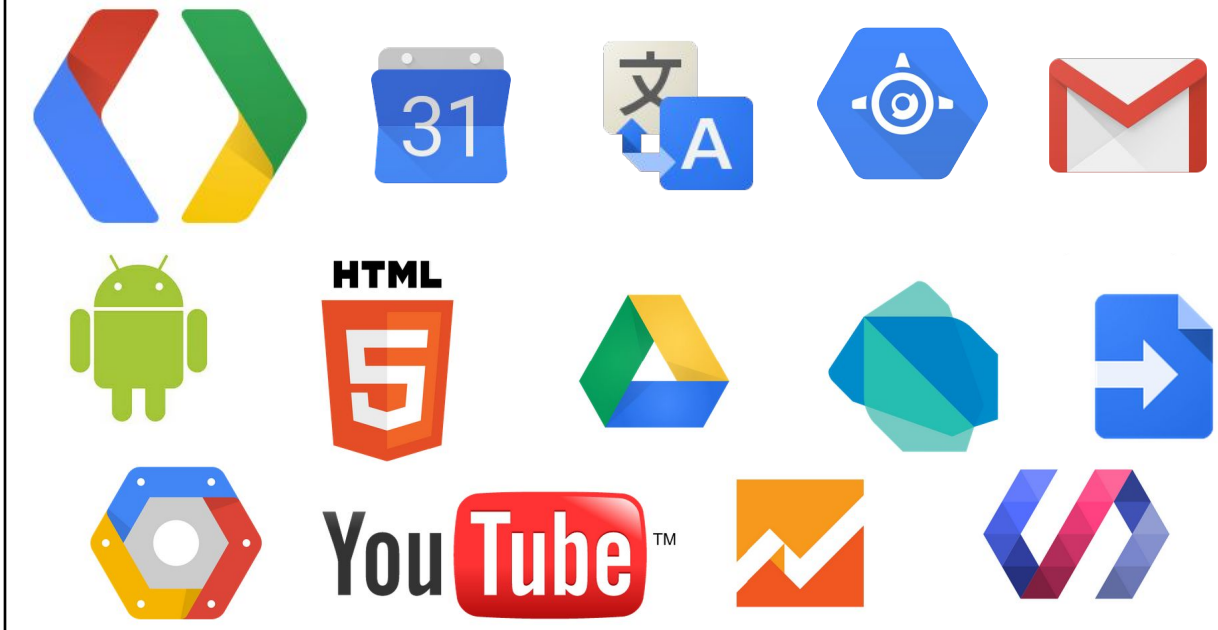
3

REST APIs

Short Python code snippets using GCP & G Suite APIs

API key (public data) vs. OAuth2 access (private data)



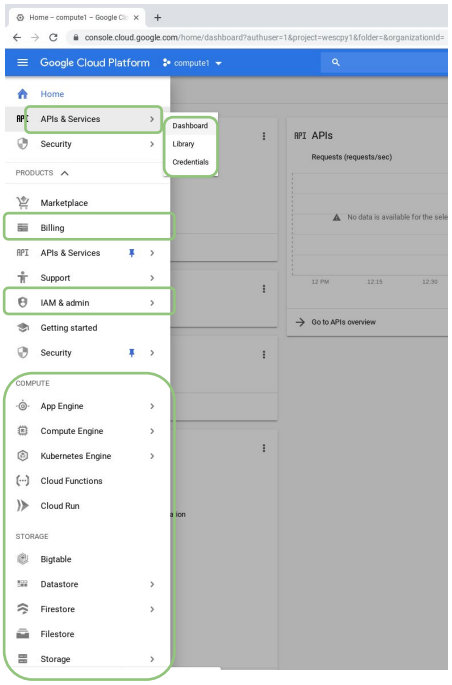


Cloud/GCP console

`console.cloud.google.com`

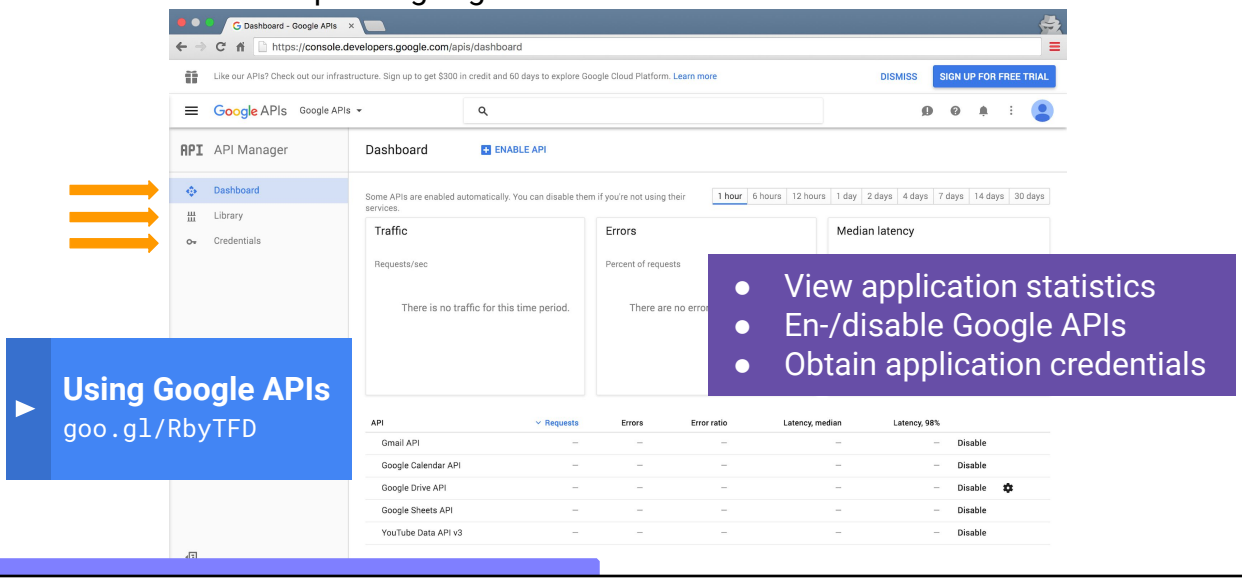
- Hub of all developer activity
- Applications == projects
  - New project for new apps
  - Projects have a billing acct
- Manage billing accounts
  - Financial instrument required
  - Personal or corporate credit cards, Free Trial, and education grants
- Access GCP product settings
- Manage users & security
- Manage APIs in devconsole

Google Cloud



API manager aka Developers Console (devconsole)

`console.developers.google.com`



- View application statistics
- En-/disable Google APIs
- Obtain application credentials

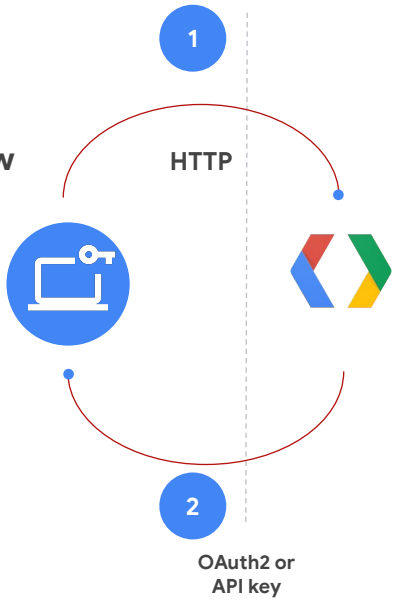
Using Google APIs
`goo.gl/RbyTFD`

HTTP-based REST APIs

Google APIs request-response workflow

- Application makes request
- Request received by service
- Process data, return response
- Results sent to application

(typical client-server model)



Google Cloud

Vision: image analysis & metadata extraction

```
from google.cloud import vision

IMG = 'https://google.com/services/images/section-work-card-img_2x.jpg'
client = vision.ImageAnnotatorClient()
image = vision.types.Image()
image.source.image_uri = IMG

response = client.label_detection(image=image)
print('** Labels detected (and confidence score):')
for label in response.label_annotations:
    print('%s (%.2f%%)' % (label.description, label.score*100.))
response = client.face_detection(image=image)
print('\n** Facial features detected (and likelihood):')
for label in response.face_annotations:
    for likelihood in dir(label):
        if likelihood.endswith('_likelihood'):
            llh = str(vision.enums.Likelihood(getattr(label,
                  likelihood))).split('.')[1].replace('_', ' ').lower()
            print('%s: %s' % (likelihood.split('_')[0].title(), llh))
```

Google Cloud

Vision: image analysis & metadata extraction

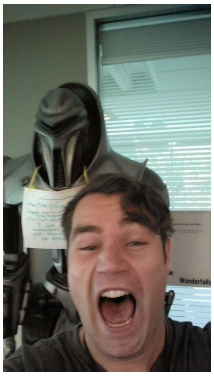
```
$ python3 viz_demo.py
** Labels detected (and confidence score):
Sitting (89.94%)
Interior design (86.09%)
Furniture (82.08%)
Table (81.52%)
Room (80.85%)
White-collar worker (79.04%)
Office (76.19%)
Conversation (68.18%)
Photography (62.42%)
Window (60.96%)

** Facial features detected (and likelihood):
Anger: very unlikely
Blurred: very unlikely
Headwear: very unlikely
Joy: very likely
Sorrow: very unlikely
Surprise: very unlikely
Under: very unlikely
```

Cloud Vision exercise

g.co/codelabs/vision-python

Google Cloud

(others at gcplab.me)



BigQuery: querying Shakespeare words

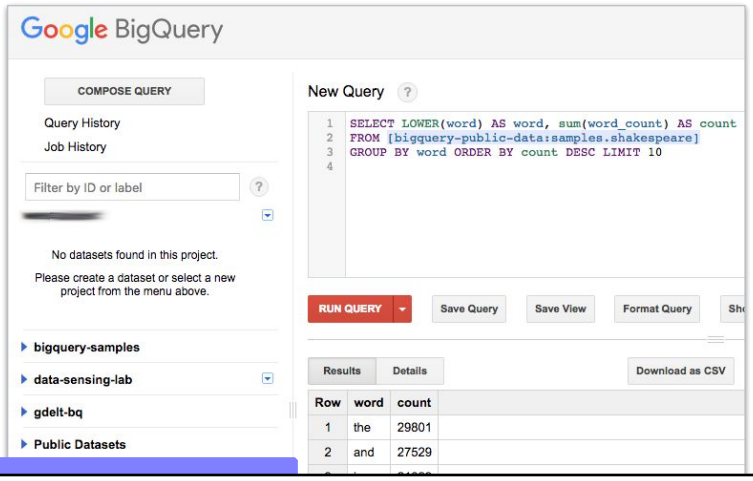
```
from google.cloud import bigquery

TITLE = "The most common words in all of Shakespeare's works"
QUERY = '''
    SELECT LOWER(word) AS word, sum(word_count) AS count
    FROM `bigquery-public-data.samples.shakespeare`
    GROUP BY word ORDER BY count DESC LIMIT 10
'''

rsp = bigquery.Client().query(QUERY).result()
print('\n*** Results for %r:\n' % TITLE)
print('\t'.join(col.name.upper() for col in rsp.schema))  # HEADERS
print('\n'.join('\t'.join(str(x) for x in row.values()) for row in rsp)) # DATA
```

Google Cloud

Top 10 most common Shakespeare words

```
$ python bq_shake.py

*** Results for "The most common words in all of Shakespeare's works":

WORD     COUNT
the      29801
and      27529
i        21029
to       20957
of       18514
a        15370
you      14010
my       12936
in       11722
that     11519
```

Google Cloud



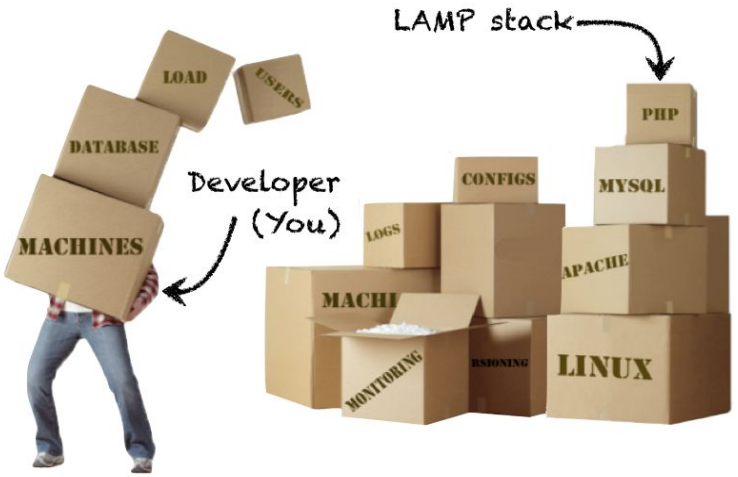
4 Run your code on Google Cloud serverless

GCP: Google App Engine , Google Cloud Functions

G Suite: Google Apps Script

Why does App Engine exist?

- Focus on app not DevOps
  - Web app
  - Mobile backend
  - Cloud service
- Enhance productivity
- Deploy globally
- Fully-managed
- Auto-scaling
- Pay-per-use
- Familiar languages



Google Cloud

Hello World (Python "MVP")

app.yaml

```
runtime: python37
```

main.py

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
    return 'Hello World!'
```

requirements.txt

```
Flask==1.0.2
```

Deploy:

```
$ gcloud app deploy
```

Access globally:

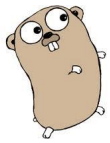
PROJECT_ID.appspot.com

Google Cloud

Quickstart tutorial and open source repo at
cloud.google.com/appengine/docs/standard/python3/quickstart

Why does Cloud Functions exist?

- Don't have entire app?
  - No framework "overhead" (LAMP, MEAN...)
  - Deploy microservices
- Event-driven
  - Triggered via HTTP or **background events**
    - Pub/Sub, Cloud Storage, Firebase, etc.
  - Auto-scaling & highly-available; pay per use
- Flexible development environment
  - Cmd-line or **developer console** (in-browser)
- Cloud Functions for Firebase
  - Mobile app use-cases


- Available runtimes
  - JS/Node.js 6, 8, *10*
  - Python 3.7
  - Go 1.11, *1.12*
  - *Java 8*

Google Cloud

Hello World (Python "MVP")

main.py

```
def hello_world(request):
    return 'Hello World!'
```

Deploy:

```
$ gcloud functions deploy hello --runtime python37 --trigger-http
```

Access globally (curl):

```
curl -X POST https://GCP_REGION-PROJECT_ID.cloudfunctions.net/hello \
   -H "Content-Type:application/json"
```

Access globally (browser):

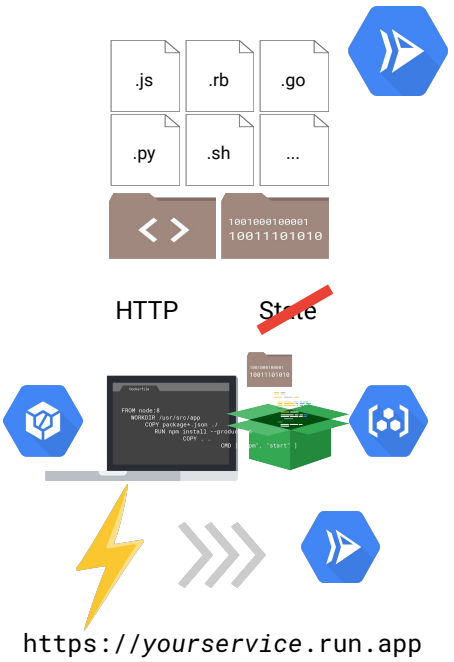
GCP_REGION-*PROJECT_ID*.cloudfunctions.net/hello

Google Cloud

Quickstart tutorial and open source repo at
cloud.google.com/functions/docs/quickstart-python

Code, build, deploy

- Any language, library, binary
  - HTTP port, stateless
- Bundle into container
  - Build w/Docker OR
  - Google Cloud Build
  - Image ⇒ Container Registry
- Deploy to Cloud Run (managed or GKE)



Google Cloud

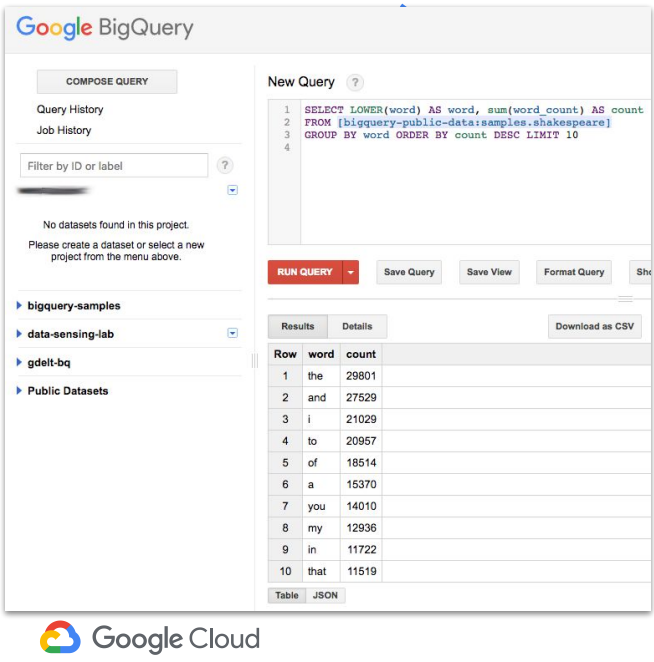
5 All of Cloud (inspiration)

Build powerful solutions with both GCP and G Suite

Big data analysis to slide presentation

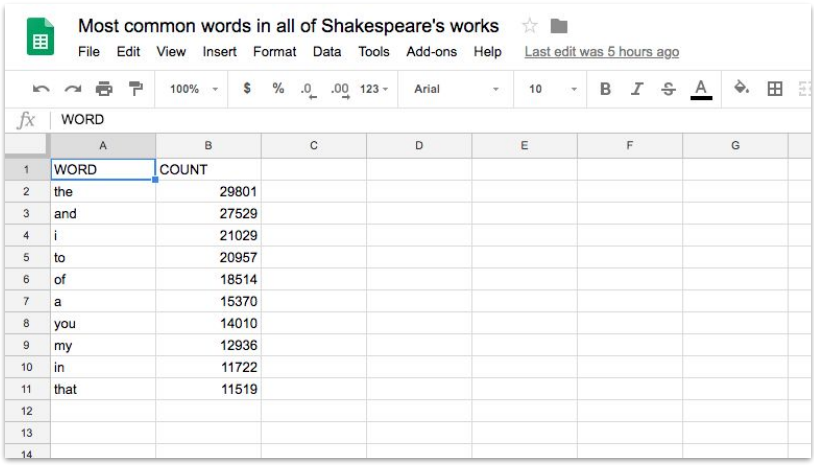
Access GCP tools from G Suite

Google Cloud



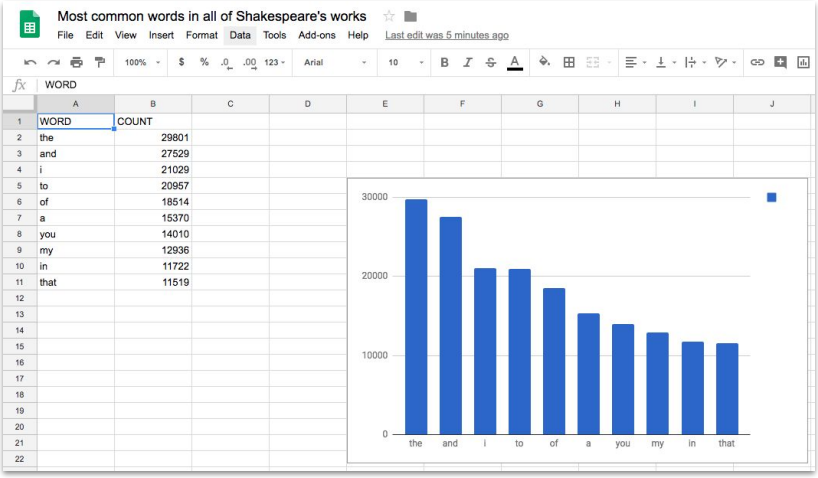
Google Cloud

Store big data results



Google Cloud

Visualize big data results



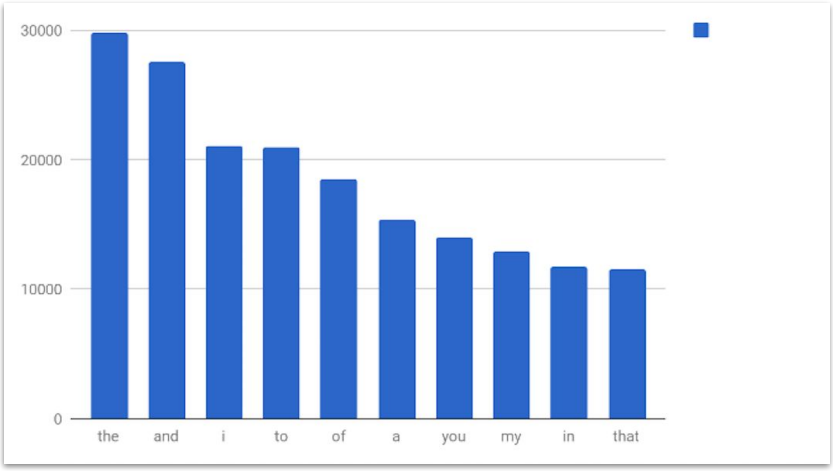
Google Cloud

Ingest data from Sheets

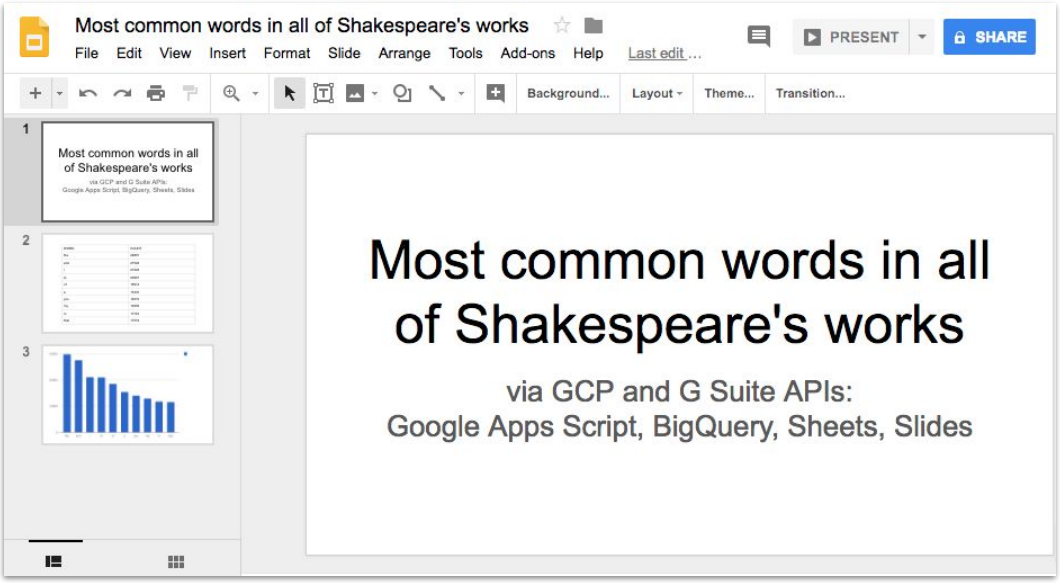
WORD	COUNT
the	29801
and	27529
i	21029
to	20957
of	18514
a	15370
you	14010
my	12936
in	11722
that	11519

Google Cloud

Link to chart in Sheets

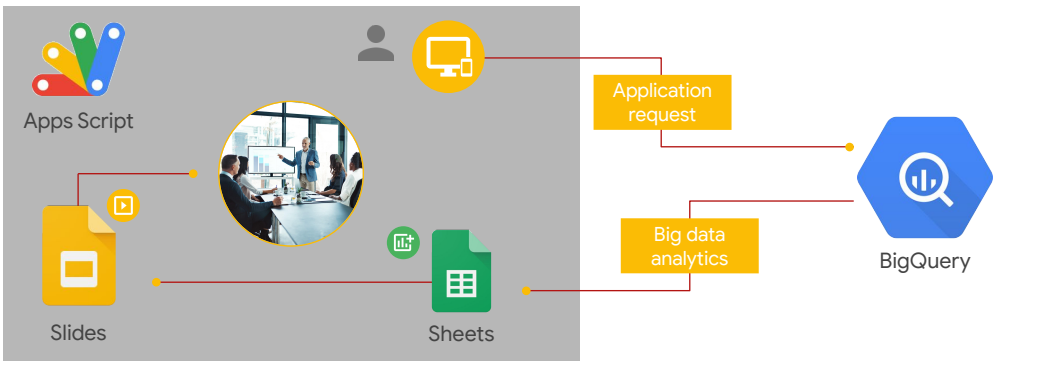


Google Cloud



Google Cloud

Supercharge G Suite with GCP



Google Cloud

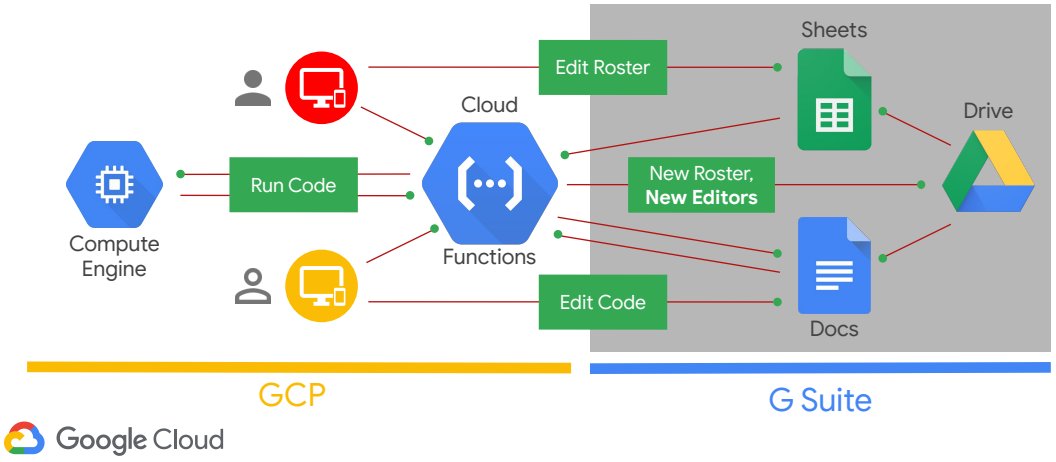
App summary

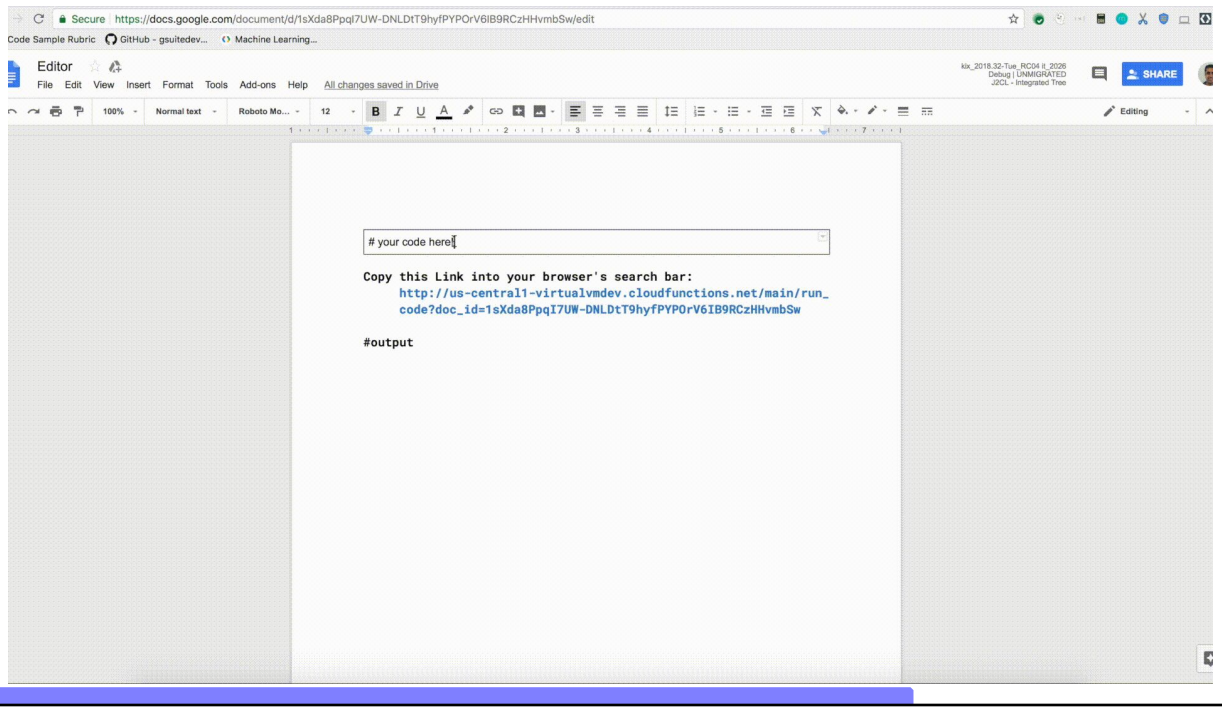
- Leverage GCP and build the "final mile" with G Suite
  - Driven by **Google Apps Script**
  - **Google BigQuery** for data analysis
  - **Google Sheets** for visualization
  - **Google Slides** for presentable results
  - "Glued" together w/G Suite serverless
- Build this app (codelab): `g.co/codelabs/bigquery-sheets-slides`
- Video and blog post: `bit.ly/2OcptaG`
- Application source code: `github.com/googlecodelabs/bigquery-sheets-slides`
- Presented at Google Cloud NEXT (Jul 2018 [DEV229] & Apr 2019 [DEV212])
  - `cloud.withgoogle.com/next18/sf/sessions/session/156878`
  - `cloud.withgoogle.com/next/sf/sessions?session=DEV212`

Using Google Docs as an IDE

Running Python code in Docs w/the help of GCP

Google Docs as a Code Editor/IDE





App summary

- Motivation
  - Lead TA during school year; manage student coursework
  - Needed roster app for non-engineering faculty for student coding
  - Use two new products (*Docs API* and *Python* Cloud Functions)

- Implementation
  - Use Drive & Sheets APIs to create student roster Sheet for professor
  - Faculty completes roster in Sheet, calls Cloud Function again
  - Autogenerate "Docs IDE" per student via Docs API
  - Student edits code in Doc, then calls Cloud Function to run
  - Read from Doc, spin-up GCE VM, run code, write results into Doc, bring down VM

- Application source code (*to-be-published*)
  - `github.com/gsuitedevs/python-samples/tree/master/docs/docs_editor`

6: Why are you here?

Or rather, why am *I* here?

Writing solid code (SWE perspective)

- Create your programs so that they are "idiot proof"
- When testing, put on the "any-way-to-break-it" hat
- Use unexpected or Byzantine input (sabotage!!)
- Test most common error conditions
- Use wide variety of common/general input
- Use the debugger for language
- Use unit-testing framework for language

Different types of testing

- Manual testing - human user running tests
- Automated testing - automatically-executed test scripts
  - Manual doesn't scale
  - Mostly this is key (think CI/CD anyway)
- Consider Python
  - Even if software written in other languages
  - Vast testing tool collection
    - http://goo.gl/Fpz0Z

Python testing tools

- Unit testing
- Mock testing
- Fuzz testing
- Web testing
- Acceptance/Business Logic testing
- GUI testing
- Source Code Checking
- Code Coverage
- Continuous Integration
- Automatic Test Runners
- Test Fixtures

Consider Jython

Jython is a port of the (original C) Python interpreter (re)written in Java, running on the JVM. It gives Java developers a scripting and shell-like tool with which to instatiate and/or test Java classes without a main driver or harness application.

Jython code (Python flavor)

```
$ jython
Jython 2.7.2 (v2.7.2:925a3cc3b49d, Mar 21 2020, 10:03:58)
[Java HotSpot(TM) 64-Bit Server VM (Oracle Corporation)]
on java1.8.0_221
Type "help", "copyright", "credits" or "license" for more
information.
>>>
>>> print 'Hello World!'  # print('Hello World!') in 3.x
Hello World!
```

Jython code (Python-to-Java)

```
>>> import sys
>>> sys.stdout.write('Hello World!\n')
Hello World!
>>>
>>> from java.lang import System
>>> System.out.write('Hello World!\n')
Hello World!
```

Jython code (Java flavor)

```
>>> print('Running on Java version: ' +
System.getProperty('java.version'))
Running on Java version: 1.8.0_221
>>>
>>> print('Unix time from Java: ' +
str(System.currentTimeMillis()))
Unix time from Java: 1586920010953
>>>
>>> import time
>>> time.ctime(System.currentTimeMillis()/1000)
'Tue Apr 14 20:07:34 2020'
```

Java code (embedded Jython)

```
import org.python.util.PythonInterpreter;

public class JythonHelloWorld {
  public static void main(String[] args) {
    try(PythonInterpreter pyInterp = new
PythonInterpreter()) {
      pyInterp.exec("print('Hello Python World!')");
    }
  }
}
```

Possible test cluster

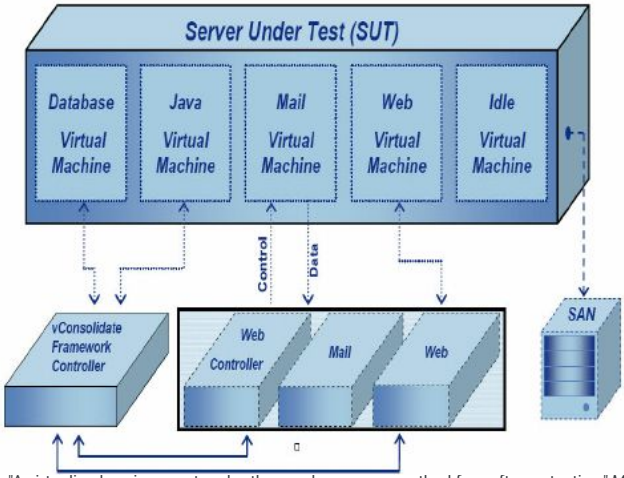


Image: Popović, Lazic, & Mastorakis. "A virtualized environment and orthogonal array as a method for software testing." *MMACTEE'09*, 2009.

Nested virtualization: what

- **def**: *nested virtualization*: virtualization that runs inside a(n already) virtualized environment. In other words, "VMs running on VMs" is possible by running a hypervisor inside of a virtual machine (VM), which itself runs on a hypervisor.
- **def**: *hypervisor*: specialized software programs that manage OSs needed within virtual environments; they're responsible for allocating essential resources like processing power, memory, and other resources that virtual environments require to function.
- Test cluster can be spun-up on one bare metal machine

Nested virtualization: why

- Advantages over traditional on-premise solutions (nVMs > VMs > bare metal)
- Can create a cluster of VMs on a single bare metal machine
- Enhanced flexibility - develop+test on your terms; you control test envs
- Save $$ - Bare metal (HW + maintenance + staff + power + NW + cooling)
- Maximize usage w/nested VMs; know when you need more
- Scalability - easily spin-up additional VMs w/o new hardware

GCP Nested virtualization

This is a huge win for QA and development teams who need native mobile systems, such as Android, for testing/validating mobile apps. Using Google's new hardware-accelerated nested virtualization, Functionize now enables QA teams to dramatically reduce costs, time-to-test, and the pain of maintaining a complex device inventory. — Tamas Cser, Founder and CEO, Functionize

- 2017 launch announcement
- Nested VMs documentation

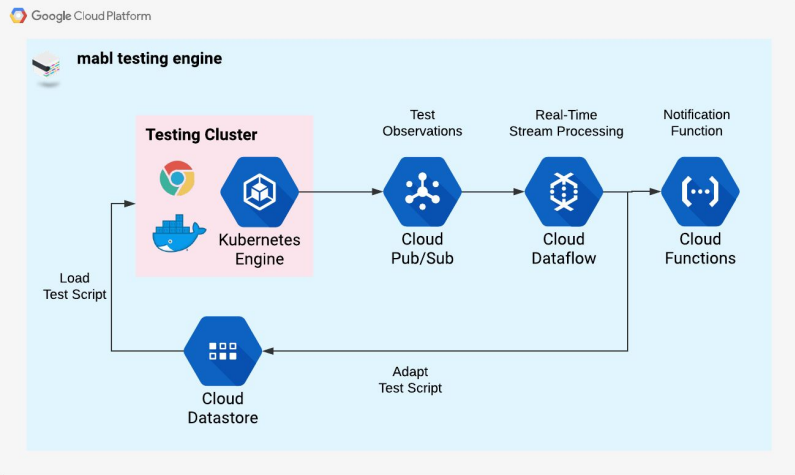
GCP and software testing

- `mabl` hints at SQA future
- GCP and the Future of Software Testing
- https://mabl.com/blog/gcp-and-the-future-of-software-testing
- Jul 2018, Google Cloud NEXT '18, San Francisco, CA, USA
- 4 main points
  - Adapt seamlessly to change
  - Run in the cloud
  - Produce insightful output
  - Integrate with continuous delivery pipeline

Adapt seamlessly to change

- Tests come from cloud-based data storage
- Dynamic evaluation of problem domain
  - Did a button move?
  - Did text copy change?
- Confirmed changes auto-update tests
- Then (new) tests executed against updated target

Adapt seamlessly to change



Run in the cloud

- Assume...
  - 50 test scripts
  - 3 scenarios: diff users (guest, reg'd), diff locs, diff CCs
  - 4 browsers * 2 (current & prev version)
  - 3 target envs: test, staging, prod
  - 3 releases/deployments/day
  - 7 mins/test-run
- Need: ??? compute hrs/day
- Solution: we need the cloud!!

Run in the cloud

Tests add up quickly

A simple case - web testing

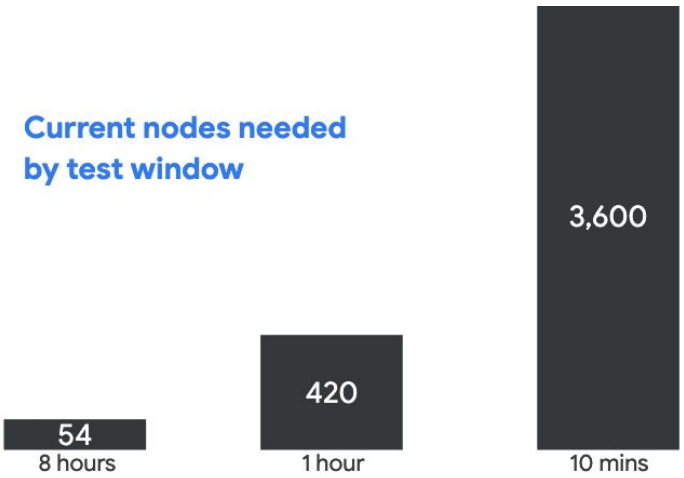
50 scripts x 3 scenarios (average) x 8 browser versions x 3 targets x 3 deploys = 10,800 runs/day

Compute requirements

10,800 runs/day x 7 mins/run = 1,260 compute hours/day

Run in the cloud

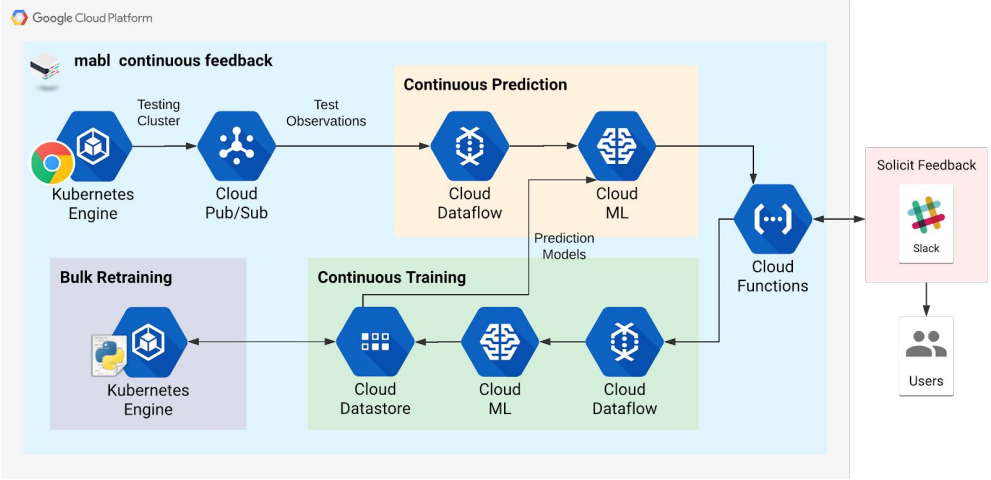
Current nodes needed by test window



Produce insightful output

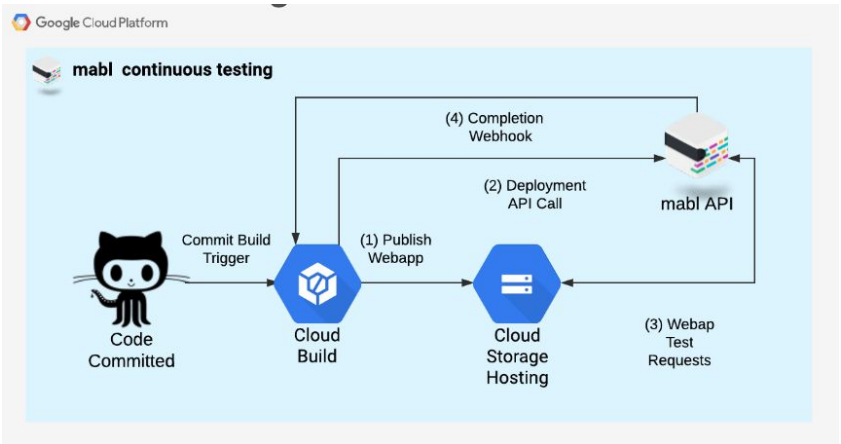
- Track everything (that matters)
  - Run logs, screenshots, measurements, etc.
- Process analytically to get...
  - Inventory changes
  - UI/UX changes
  - JS errors, broken links, etc.
  - Performance (time, memory)
    - Took too much memory
    - Test ran too fast/slow

Produce insightful output



Integrate with continuous delivery pipeline

- Automate test cycle
- Trigger upon commit



Final thoughts

- ML use-cases for SQA
  - Choosing most appropriate test cases
  - Use for "fuzzing" - intelligently choose how to generate random test data
  - Possible: code duplication
  - Possible: design patterns & code refactoring
- Possible workflow
  - Launch (multiple GCE nodes or GKE cluster)
  - Run tests
  - Export report to Sheets/Drive
  - Generate charts w/Sheets, send all to Slides
- Older books but with useful insights
  - Whittaker, Arbon, Carollo, "How Google tests software" (2012, Addison Wesley)
    - http://amzn.com/0321803027
  - Google, Site Reliability Eng (2016, after testing, what happens after deploying to prod?)
    - http://landing.google.com/sre/books

7 Resources & summary

What's available for students & educators?

Other Google APIs & platforms

- **G Suite** (you can ***code*** Gmail, Google Drive, Calendar, Docs, Sheets, Slides!)
  - developers.google.com/gsuite
- **Firebase** (mobile development platform + RT DB)
  - firebase.google.com
- **Google Data Studio** (data visualization, dashboards, etc.)
  - datastudio.google.com/overview
- **Actions on Google/Assistant/DialogFlow** (voice apps)
  - developers.google.com/actions
- **YouTube** (Data, Analytics, and Livestreaming APIs)
  - developers.google.com/youtube
- **Google Maps** (Maps, Routes, and Places APIs)
  - developers.google.com/maps
- **Flutter** (native apps [Android, iOS, web] w/1 code base[!])
  - flutter.dev

Resources (higher education)

- **GitHub resources:** goo.gle/google-cloud-edu (faculty) & goo.gle/hackathon-toolkit (students)
- **Codelabs:** self-paced, hands-on tutorials
  - **Google** codelabs: need a Gmail account, always free: gcplab.me or g.co/codelabs/cloud
  - **Qwiklabs** codelabs: don't need a Gmail acct; typically not free: google.qwiklabs.com
- **Google Cloud Computing Foundations course:** Course materials & curriculum (beta)
- **GCP documentation:** cloud.google.com/{docs,appengine,functions,run,vision,automl, language,speech,texttospeech,translate,video-intelligence,firestore,bigquery,translate}
- **Apply for GCP or QwikLabs credits:** Want to use GCP/QwikLabs for courses or research lab?
  - Go to cloud.google.com/edu to apply for teaching or research credits!
- **Know AWS?** Compare w/GCP at cloud.google.com/docs/compare/aws
- **Other references**
  - **G Suite docs** - developers.google.com/{gsuite,drive,docs,sheets,slides}
  - **Videos** - youtube.com/GoogleCloudPlatform (GCP) & goo.gl/JpBQ40 (G Suite)
  - **Free** trial (*ignore*) and Always Free (tier) - cloud.google.com/free

Google Cloud

Thank you! Questions?

Wesley Chun
@wescpy@

Progress bars: `goo.gl/69EJVw`
md2gslides: `github.com/gsuitedevs/md2googleslides`

Google Cloud