GDG DevFest Bucharest 2018

Architecting for the Google Cloud Platform

Márton Kodok / @martonkodok
Google Developer Expert at **REEA.net - Targu Mures**

November 2018 - Bucharest, Romania

About me

- Geek. Hiker. Do-er.
- Among the Top3 romanians on Stackoverflow 127k reputation
- Google Developer Expert on Cloud technologies
- Crafting Web/Mobile backends at **REEA.net**
- BigQuery/Redis and database engine expert
- Active in mentoring and IT community

Slideshare: **martonkodok**
Twitter: **@martonkodok**
StackOverflow: **pentium10**
GitHub: **pentium10**

REEA.net uses GCP

Build on the same infrastructure
that powers Google

Google Cloud Platform (GCP)

Compute

- Compute Engine
- App Engine
- Kubernetes Engine
- GPU
- Cloud Functions
- Container-Optimized OS

Big Data

- BigQuery
- Cloud Dataflow
- Cloud Dataproc
- Cloud Dataprep
- Cloud Datalab
- Cloud Pub/Sub
- Genomics
- Data Studio

Identity & Security

- Cloud IAM
- Cloud Resource Manager
- Cloud Security Scanner
- Key Management Service
- BeyondCorp
- Data Loss Prevention API
- Identity-Aware Proxy
- Security Key Enforcement

Internet of Things

- Cloud IoT Core

Machine Learning

- Cloud Machine Learning
- Cloud Vision API
- Cloud Speech API
- Cloud Video Intelligence API
- Cloud Natural Language API
- Cloud Translation API
- Cloud Jobs API
- Advanced Solutions Lab

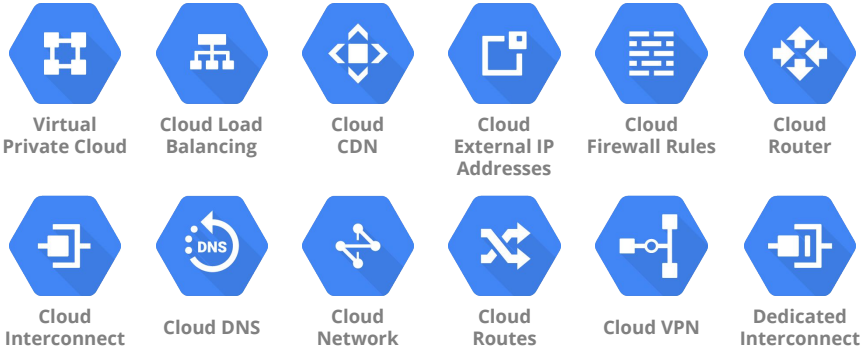
Storage & Databases

- Cloud Storage
- Cloud Bigtable
- Cloud Datastore
- Transfer Appliance
- Cloud SQL
- Cloud Spanner
- Persistent Disk

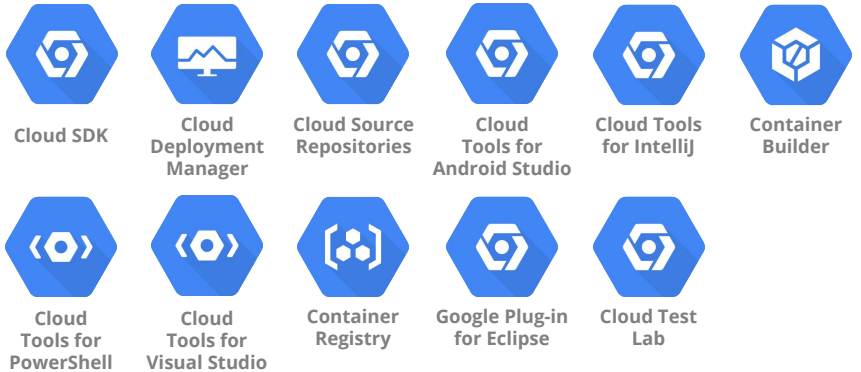
Google Cloud Platform (GCP)

Management Tools

Networking



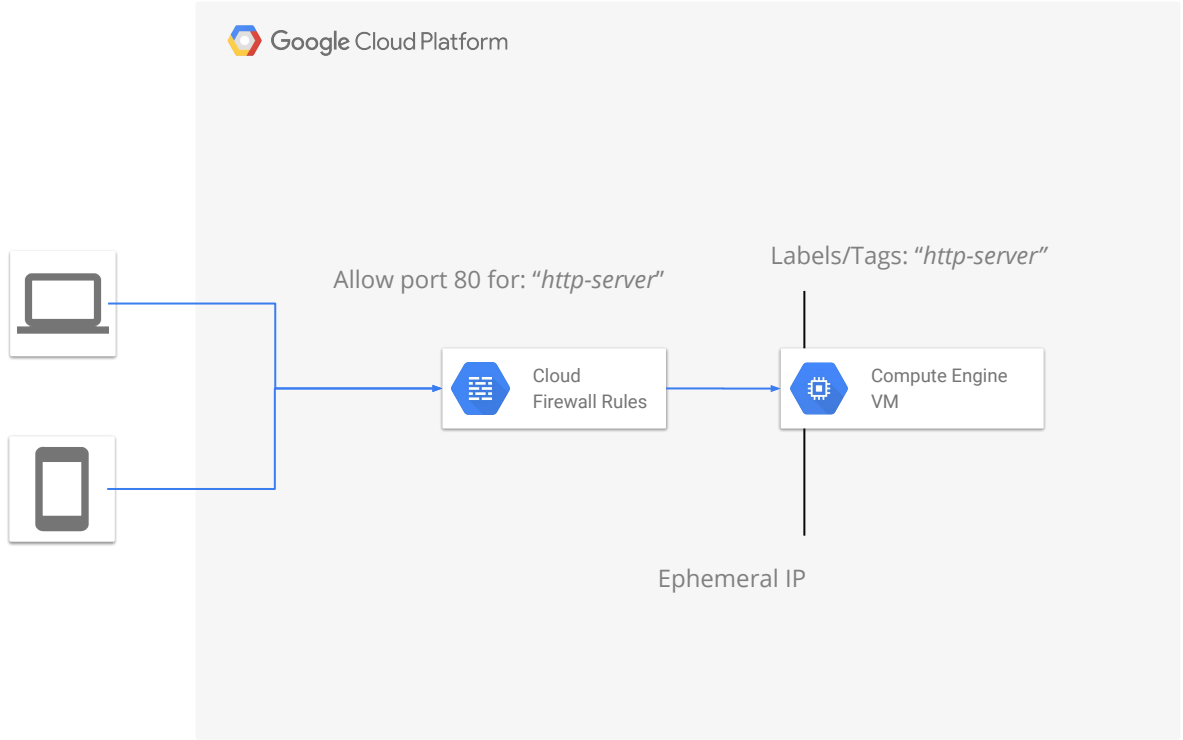
Developer Tools



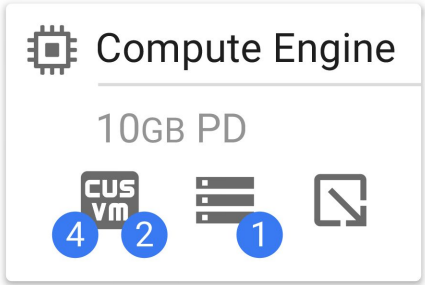
Agenda

1. Virtual Machines - Compute Engine
2. Autoscaling - Design for resiliency, scalability, and disaster recovery
3. Database layer - Identification of storage needs and mapping
4. Serverless - App Engine platform - Cloud Functions
5. Microservices architecture
6. Task Queues - Cloud Pub/Sub - DataFlow
7. Data warehouse in the Cloud
8. Practical use cases
9. Qwiklabs + Coursera

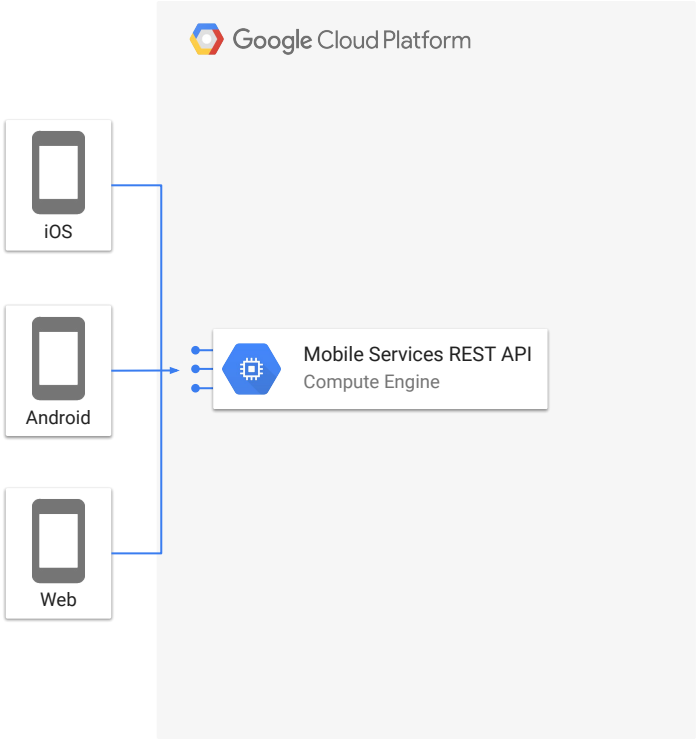
Compute Engine - Virtual Machine in the Cloud



VM details



Compute Engine - General Use

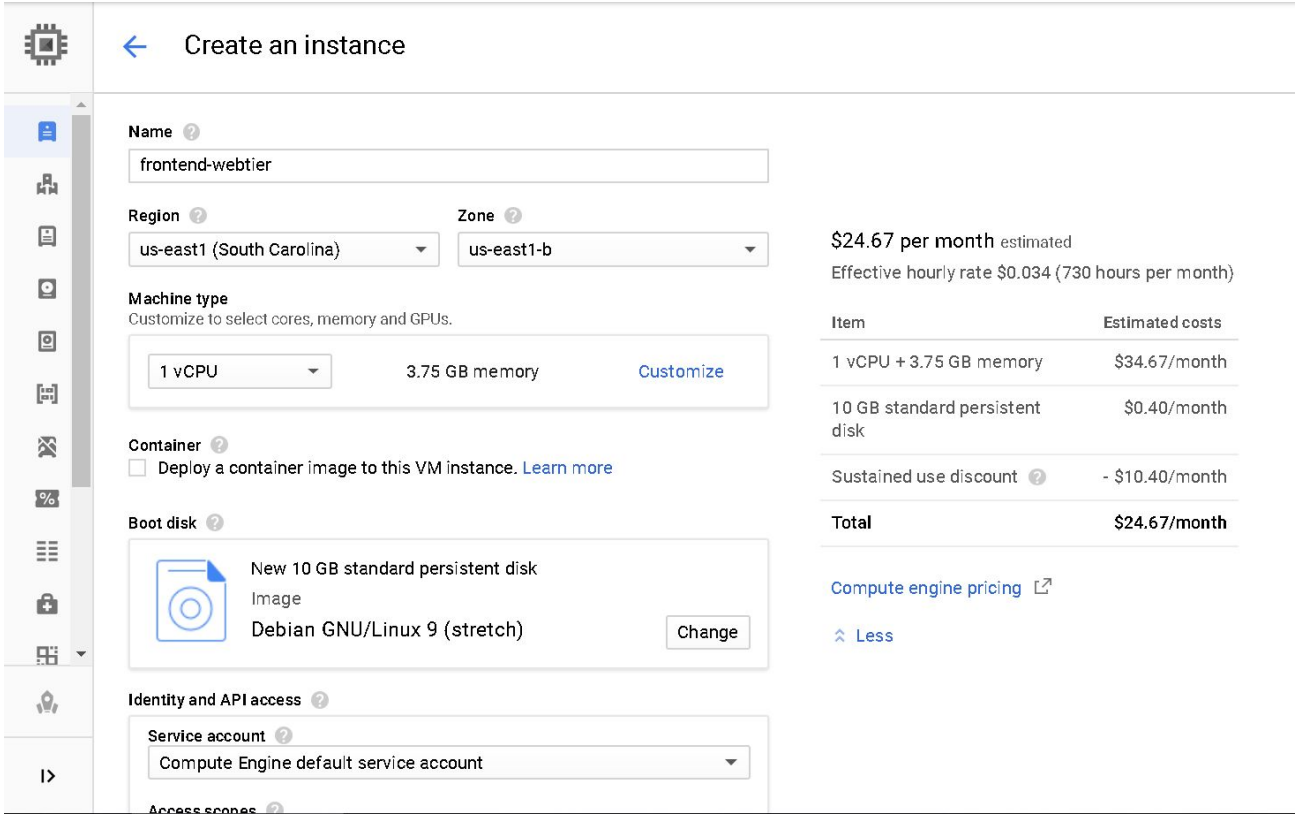


Flexibility for Every Workload

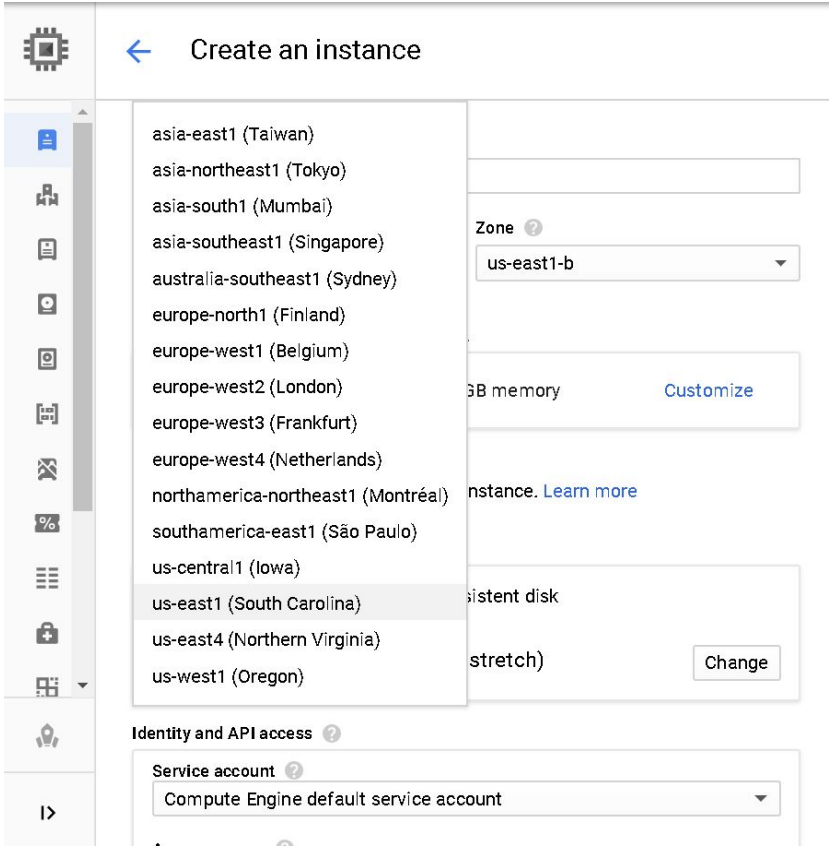
- High-Performance, Scalable VMs
- Custom Machine Types
- Persistent Disks or SSD (local or network based)
- Per-second billing
- Networking
- Firewall
- Load balancing
- Compliance & Security

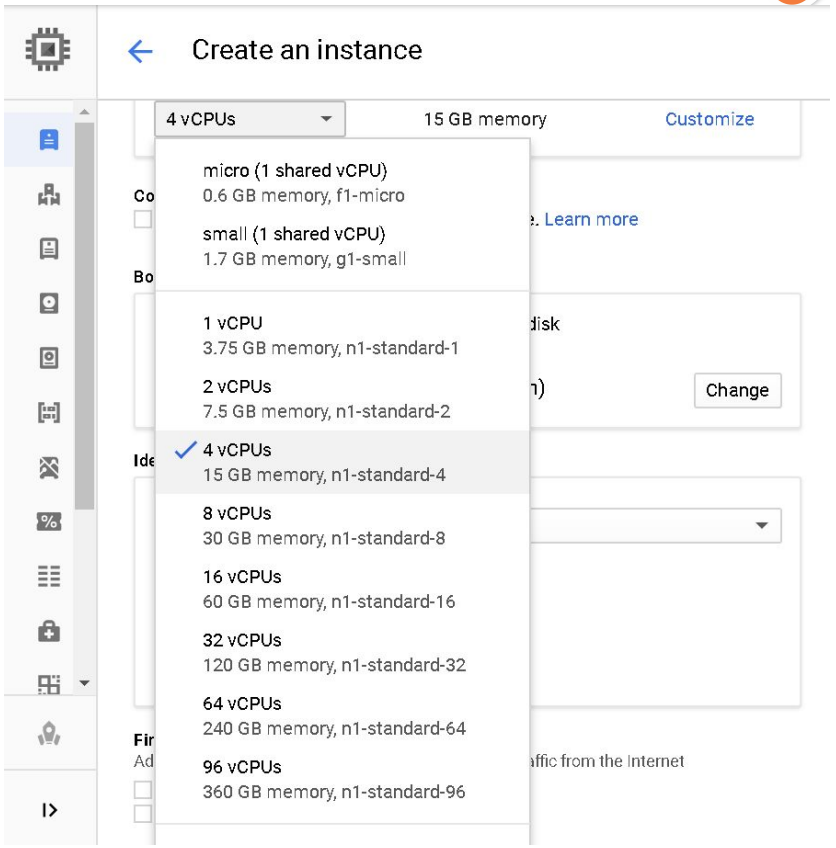
Creating a VM instance steps

Creating a VM instance



Creating a VM instance - Choosing Region, Instance type





Creating a VM instance - OS, boot disk, Network

Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk

OS images | Application images | Custom images | Snapshots | Existing disks

- CentOS 7
  x86_64 built on 20180611
- CoreOS alpha 1800.1.0
  amd64-usr published on 2018-06-11
- CoreOS beta 1772.4.0
  amd64-usr published on 2018-06-14
- CoreOS stable 1745.7.0
  amd64-usr published on 2018-06-14
- Ubuntu 14.04 LTS
  amd64 trusty image built on 2018-06-14
- Ubuntu 16.04 LTS
  amd64 xenial image built on 2018-06-12
- Ubuntu 17.10
  amd64 artful image built on 2018-06-12
- Ubuntu 18.04 LTS
  amd64 bionic image built on 2018-06-13
- Container-Optimized OS 68-10718.23.0 beta
  Kernel: ChromiumOS-4.14.41 Kubernetes: 1.10.2 Docker: 17.03.2
- Container-Optimized OS 69-10783.0.0 dev
  Kernel: ChromiumOS-4.14.48 Kubernetes: 1.10.4 Docker: 17.03.2

Can't find what you're looking for? Explore hundreds of VM solutions in Cloud Launcher

Boot disk type SSD persistent disk

Size (GB) 10

Create an instance

Identity and API access

Service account
Compute Engine default service account

Access scopes
- Allow default access
- Allow full access to all Cloud APIs
- Set access for each API

Firewall
Add tags and firewall rules to allow specific network traffic from the Internet
- [x] Allow HTTP traffic
- [x] Allow HTTPS traffic

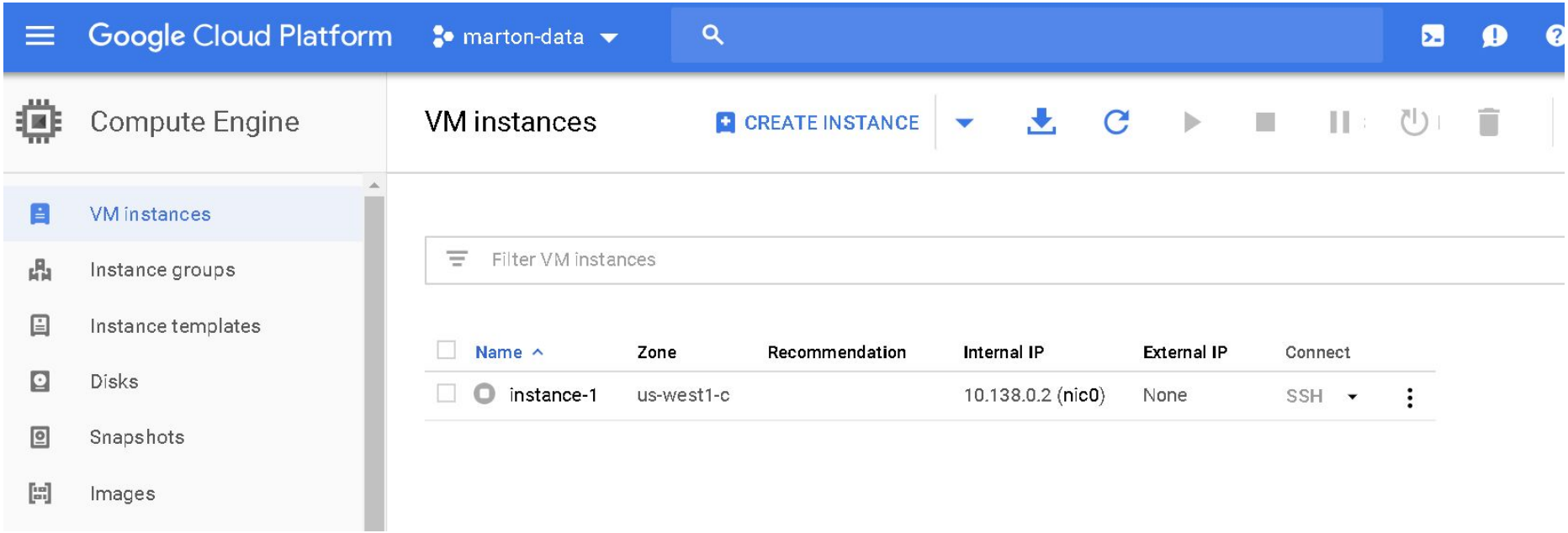
Management | Disks | Networking | SSH Keys

Network tags (Optional)

Network interfaces

default default (10.142.0.0/20)

VM instances



Virtual Private Network - Firewall rules

VPC network

- VPC networks
- External IP addresses
- Firewall rules
- Routes
- VPC network peering
- Shared VPC

Firewall rules

CREATE FIREWALL RULE | REFRESH | DELETE

incoming traffic from outside your network is blocked. Learn more

Note: App Engine firewalls are managed here.

Ingress | Egress

Name	Targets	Source filters	Protocols / ports	Action	Priority	Network
apache	Apply to all	IP ranges: 0.0.0.0/0	tcp:80,443	Allow	1000	default
beanstalkd	Apply to all	IP ranges: 173.193.140.142	tcp:11300	Allow	1000	default
cerebro-for-elasticsearch	Apply to all	IP ranges: 172.243.230.13, 1 more	tcp:9000,5601	Allow	1000	default
elasticsearch	Apply to all	IP ranges: 5.15.22.26	tcp:9200	Allow	1000	default
elasticsearch-cluster-2-node-0-firewall	elasticsearch-cluster-2-node-0	IP ranges: 0.0.0.0/0	tcp:22	Allow	1000	default

Cost savings! “Preemptible” - spot instances

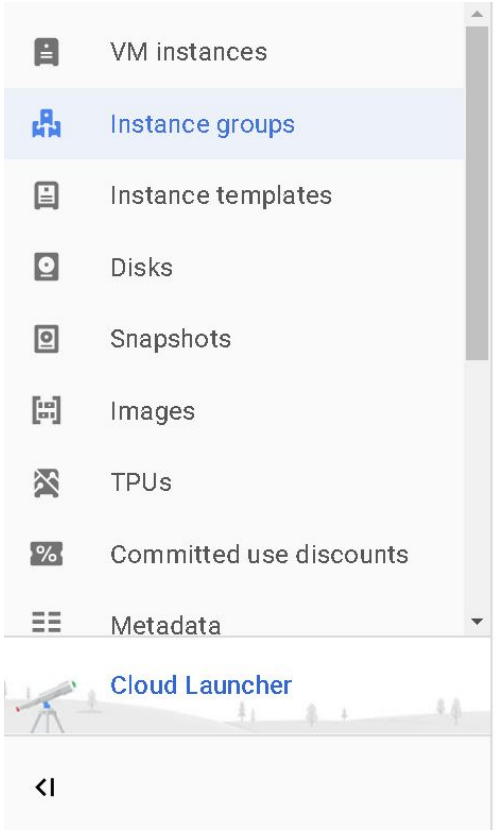
- Much lower price than normal instances
- For fault-tolerant applications
- Can withstand possible instance losses
- Runs for maximum 24 hours
- Might not be always available
- Instance templates

Good for: Nodes (eg: Elasticsearch)
Batch jobs
Worker async tasks

Pricing: ranges 3-4 times lower price

Autoscaling
with Instance Groups

Creating an Instance Group



Location

Multi-zone groups span multiple zones which assures higher availability
Learn more

- (•) Single-zone
- ( ) Multi-zone

Region us-central1 (Iowa)

Zone us-central1-c

Specify port name mapping (Optional)

Group type

- (•) Managed instance group
  Managed instance group contains identical instances, created from an instance template, and supports autoscaling, autohealing, rolling updating, load balancing and more. VM instances are stateless and disks are deleted on VM deletion or recreation. Learn more
- ( ) Unmanaged instance group
  Unmanaged instance group is best for load balancing dissimilar instances, which you can add and remove arbitrarily. Autoscaling, autohealing, and rolling updating are not supported. Learn more

Instance template

Setting Autoscaling settings

Autoscaling

On

Autoscale based on

For best results read Configuring autoscaling instance groups

CPU usage

Target CPU usage

Scaling dynamically creates or deletes VMs to meet the group target. Learn more

60 %

Minimum number of instances

1

Maximum number of instances

10

Cool-down period

60 seconds

Autohealing

VMs in the group are recreated as needed. You can use a health check to recreate a VM if the health check finds the VM unresponsive. If you do not select

Create another health check

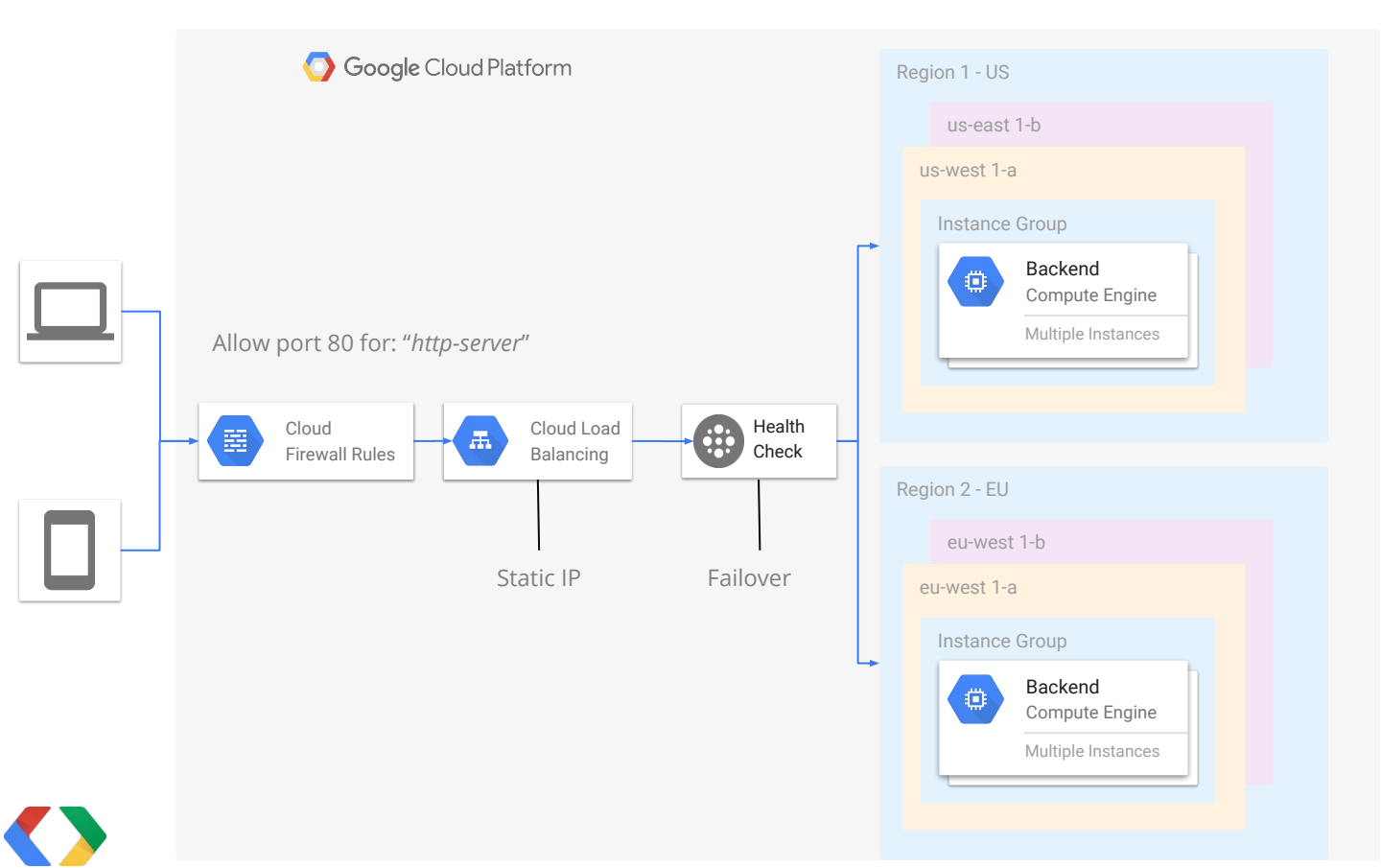
No health check

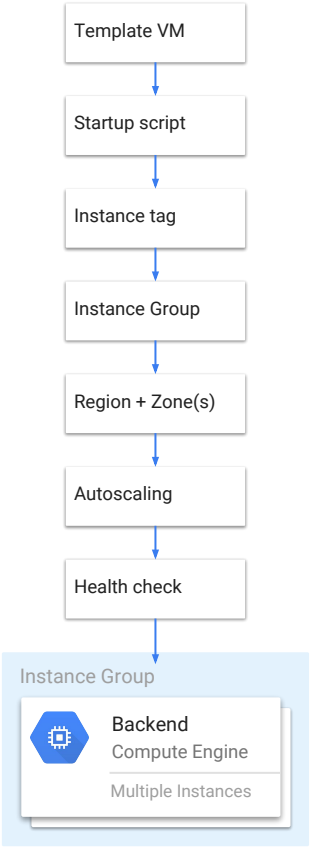
elastic-cluster-health-check (TCP)
port: 9200, timeout: 5s, check interval: 5s, unhealthy threshold: 2 attempts

Advanced creation options

Create Cancel

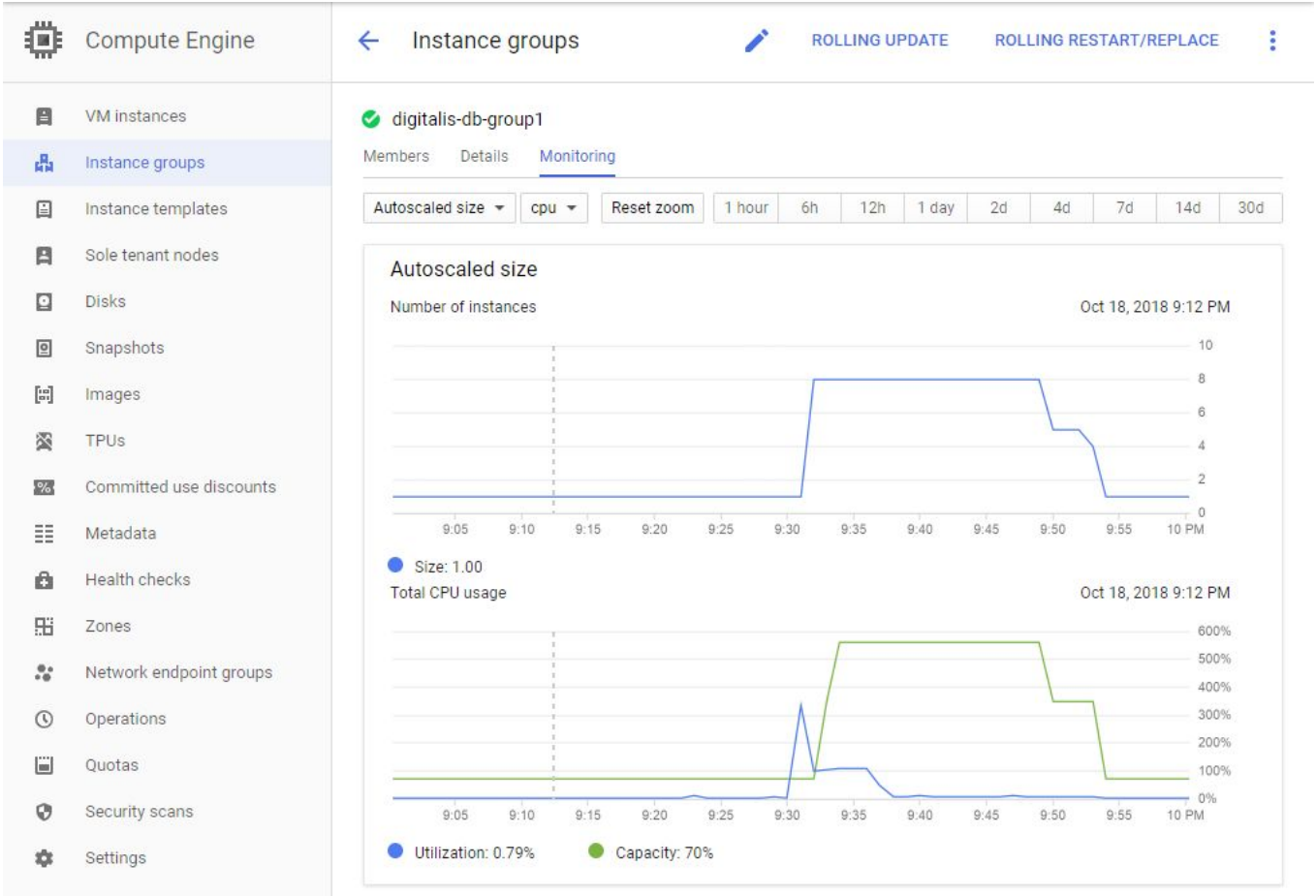
Instance Group - Autoscaling - Health Check





@martonkodok

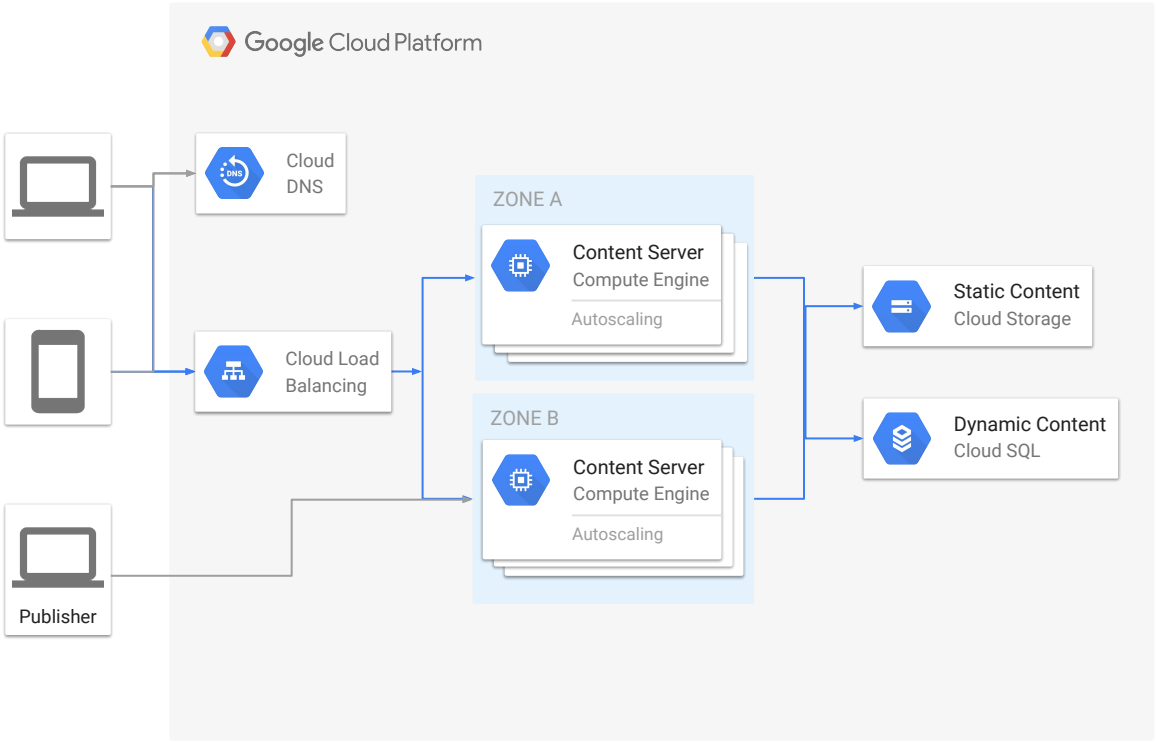
Downscale on no load after 10 minute



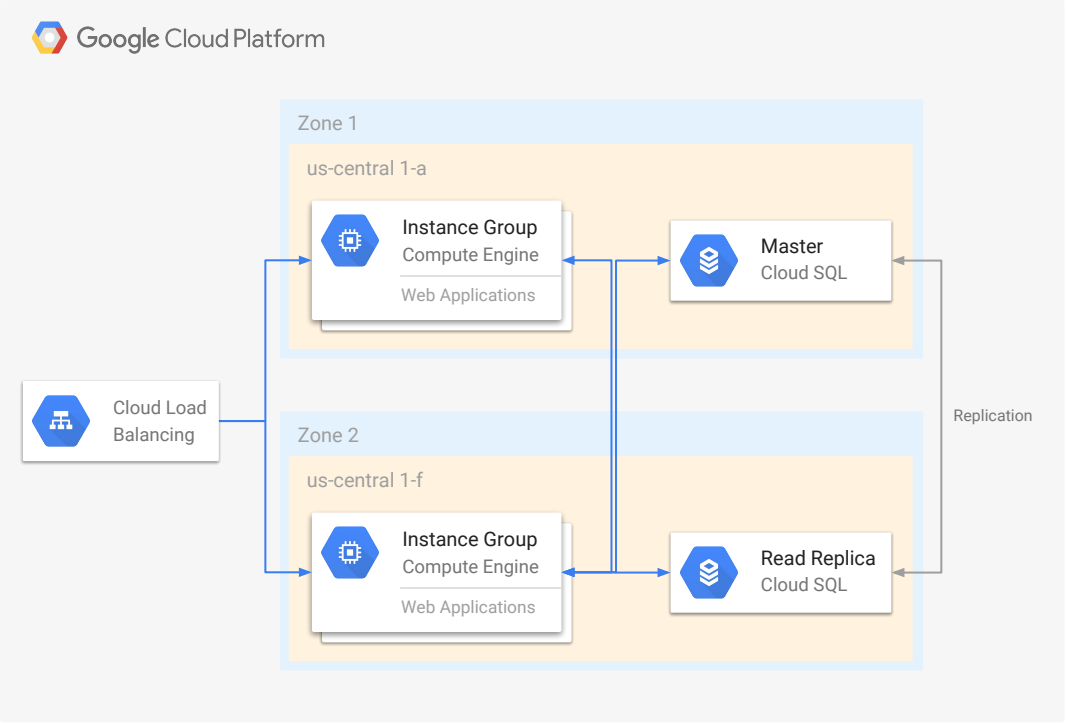
@martonkodok

Example Architecture for Autoscaling

CMS with Autoscaling

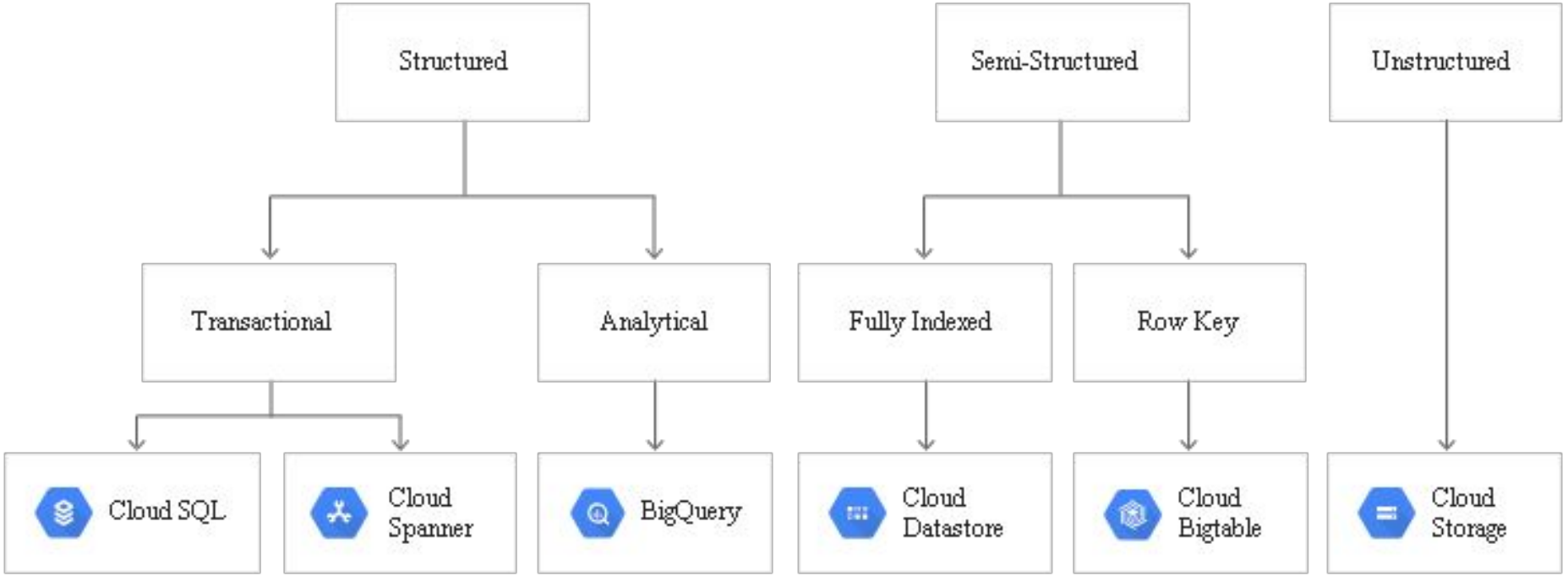


DB multiple zones and Master/Slave topology



Data layer design

Database Mapping



Serverless

Meet Serverless

Automatic scaling

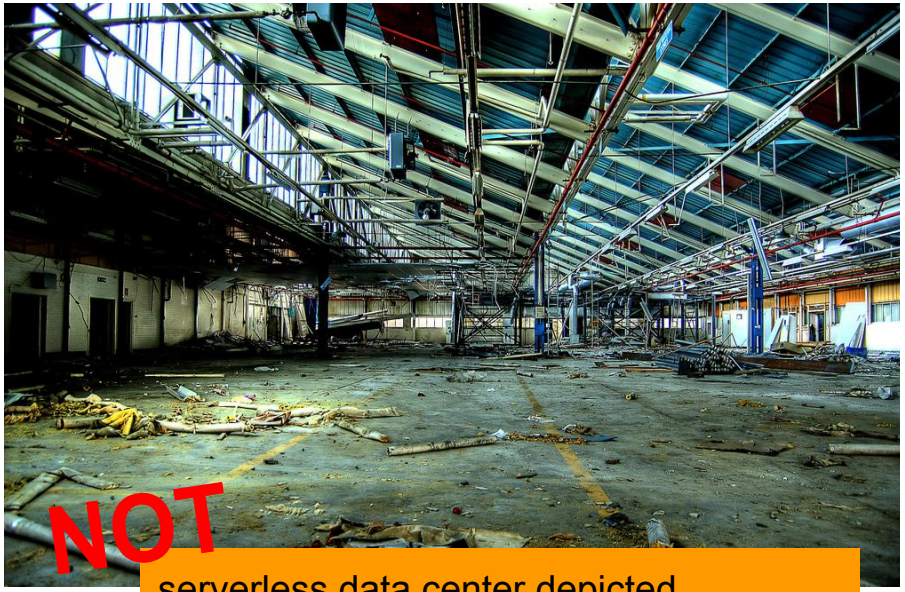
No server management

Only pay for what you use

Event-driven

Meet Serverless

- Automatic scaling
- No server management
- Only pay for what you use
- Event-driven



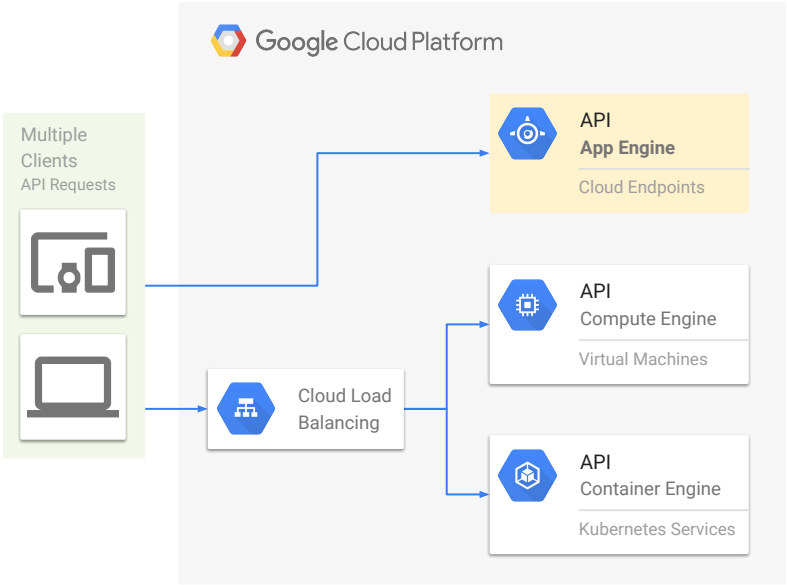
Serverless is about maximizing **elasticity**, **cost savings**, and **agility** of cloud computing.

Serverless types

Platforms

Triggered Code

App Engine - application platform



App Engine

- Fully managed serverless application platform
- Scale your applications seamlessly from zero to planet scale
- Automatically scales depending on your application traffic
- Traffic Splitting (app versions, A/B tests, incremental rollouts)
- Standard vs Flexible environments

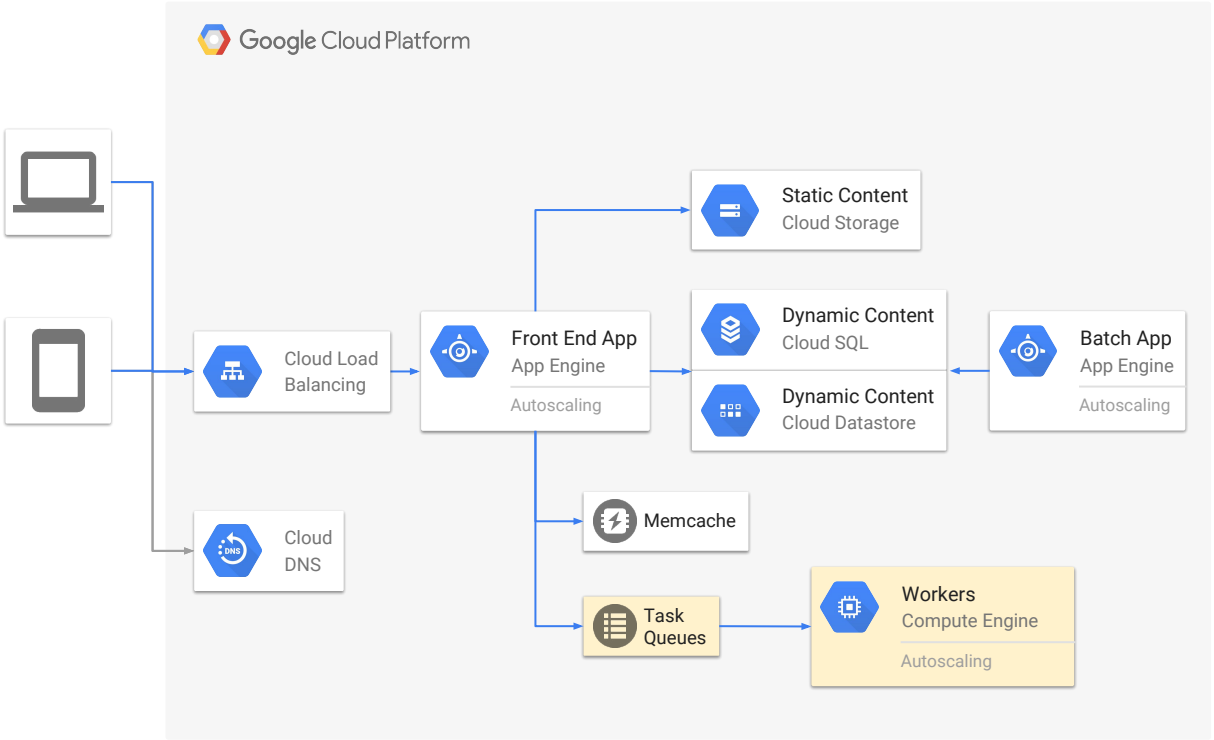
Python, Java, Node.js, Go, Ruby, PHP, .NET

App Engine

	Flexible	Standard
Instance startup time	Minutes	Seconds
Maximum request timeout	60 minutes	60 seconds
Background threads	Yes	Yes, with restrictions
SSH debugging	Yes	No
Writing to local disk	Yes, ephemeral (disk initialized on VM startup)	No
Modifying the runtime	Yes (through Dockerfile)	No
Supports installing third-party binaries	Yes	No
Pricing	Based on usage of vCPU, memory, and persistent disks	Based on instance hours

Message Queue systems: Task Queue
Cloud Pub/Sub

Task Queue - Message Queue systems



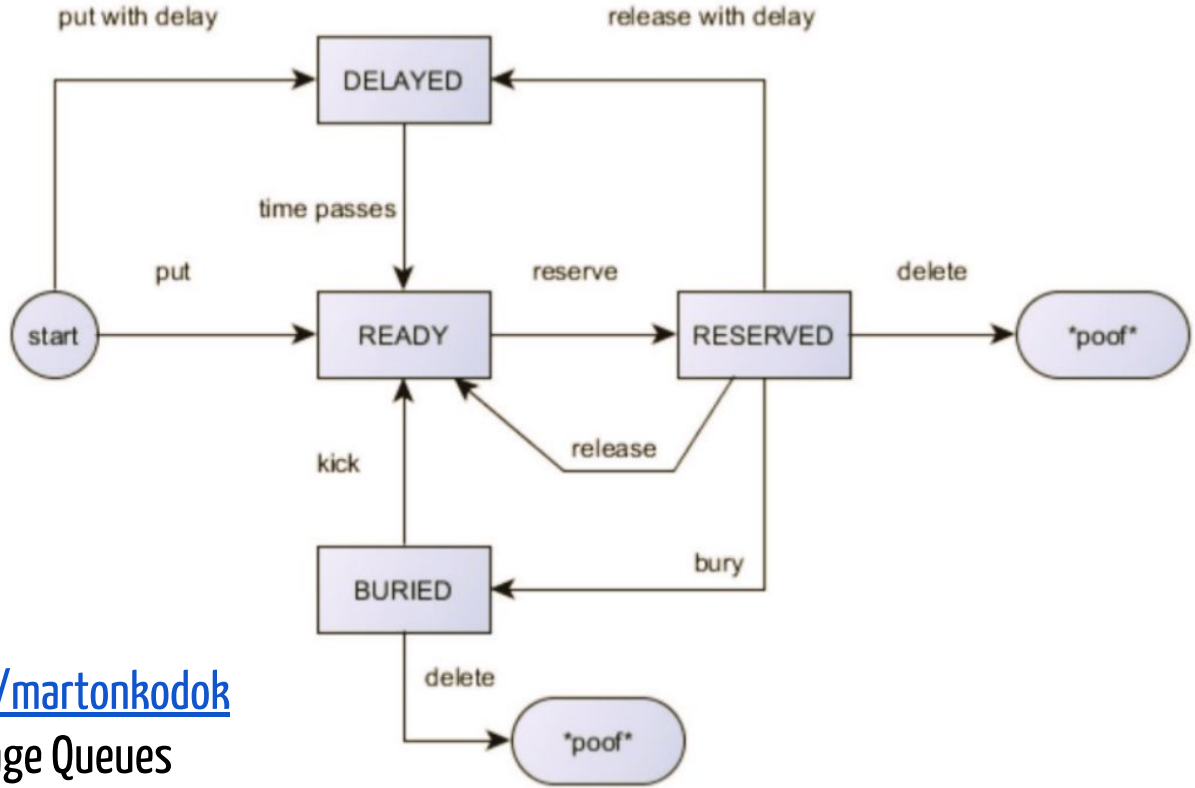
Task Queues

Push queues

- Put with delays

Pull queues

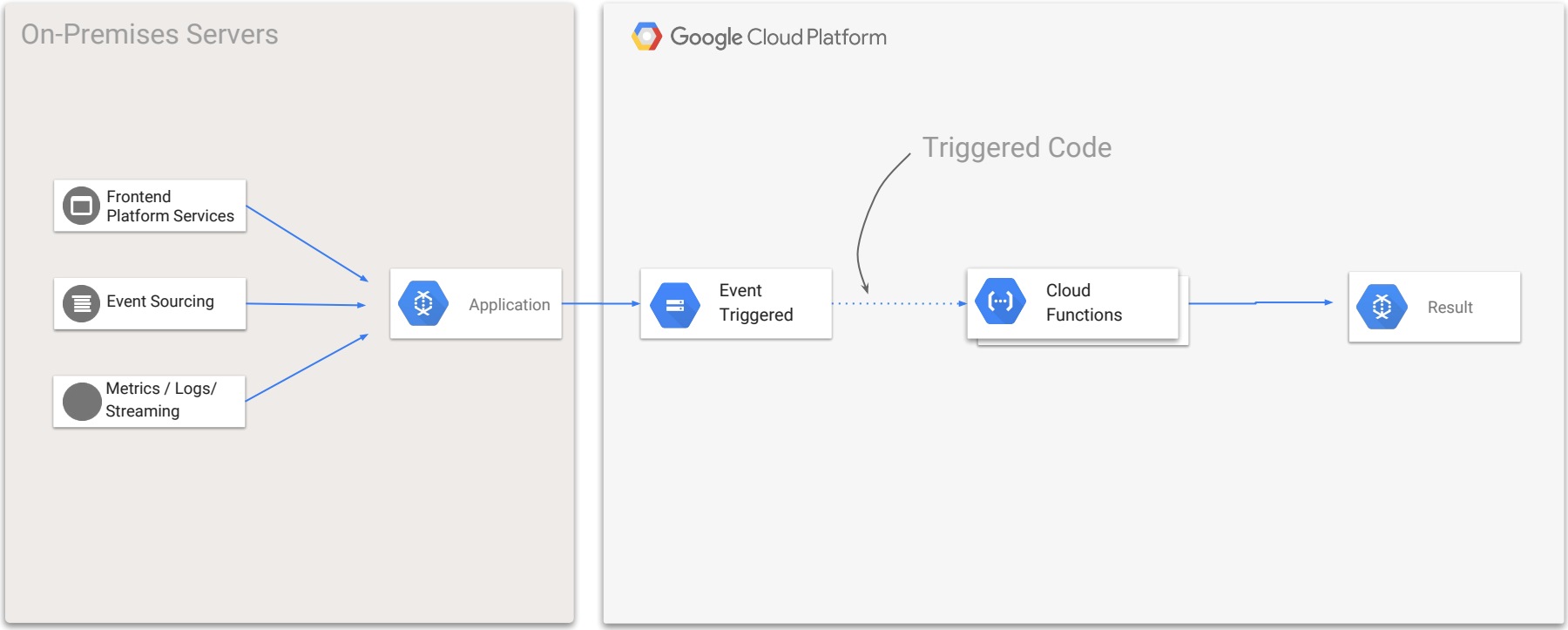
- Ability to “tag”
- Lease multiple eg: gameboard updated, game id as tag.



Slides: https://www.slideshare.net/martonkodok
Title: Architectural Patterns - Message Queues

Cloud Functions

Cloud Functions



Cloud Functions - Event-driven - FaaS

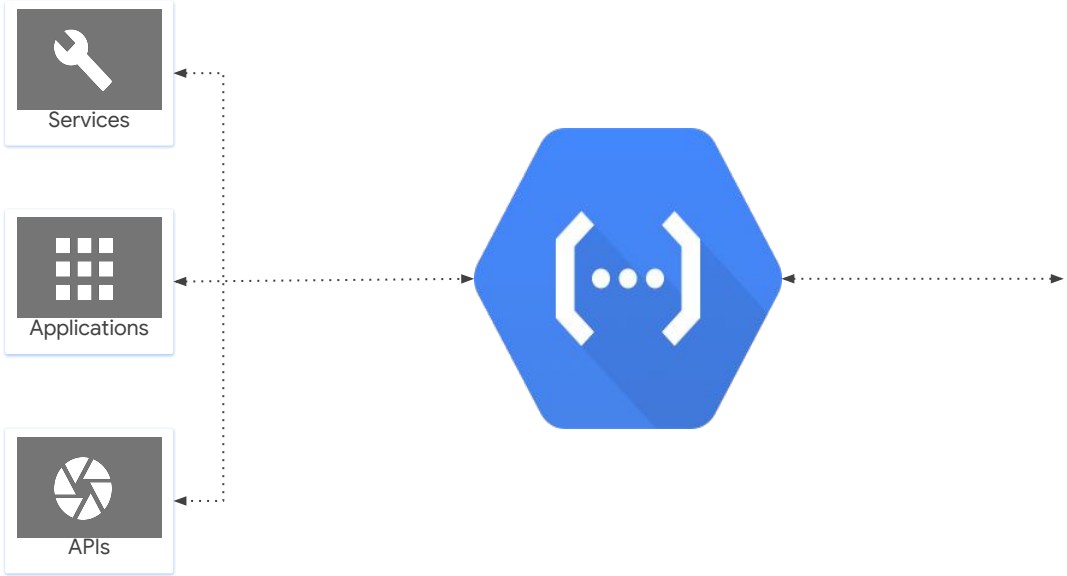
- Simplest way to run your code in the cloud - just deploy
- Automatically scales, highly available and fault tolerant
- No servers to provision, manage, patch or update
- Pay only while your code runs
- Connects and extends cloud services

(In alpha: Go and Java)

- Node 8.11
- Support for async/await

- Python 3.7.1
- Flask microframework

Functions: your gateway to GCP Services



Access 20+ Google services from GCF

BigQuery
Cloud Pub/Sub
Cloud Storage
Cloud Bigtable
Cloud Datastore
Cloud Machine Learning
Cloud Vision API
Cloud Speech API
Cloud Spanner
Cloud Natural Language API
Cloud Translation API
Stackdriver
Firebase
Assistant
Dialogflow

Why is adopting serverless hard today?

Dependencies

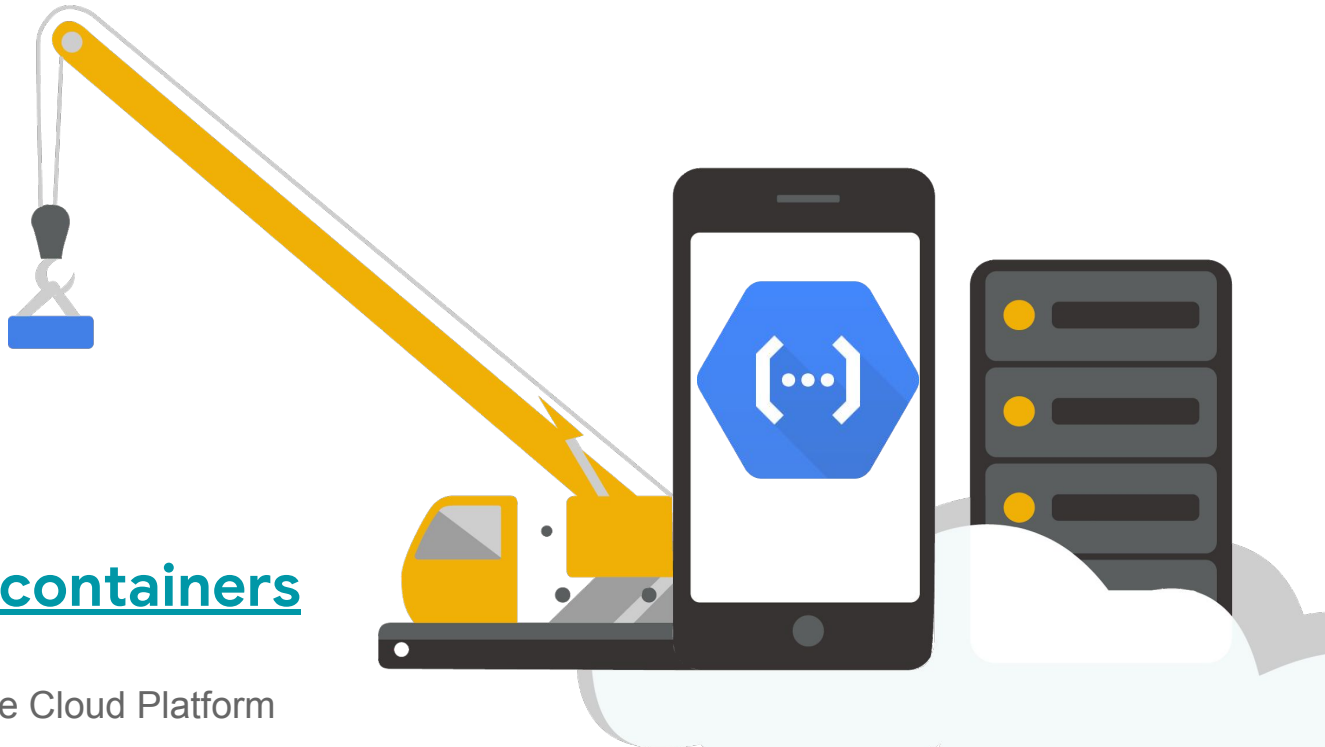
Constrained runtimes, frameworks and packages

Multi-Cloud

Unable to run your workloads on-prem, in the cloud or on a third party service provider

New Serverless containers on GCF (EAP)

- Serverless containers
- Fully managed
- BYO workloads
- Pay for use
- Alpha invitations coming later this year



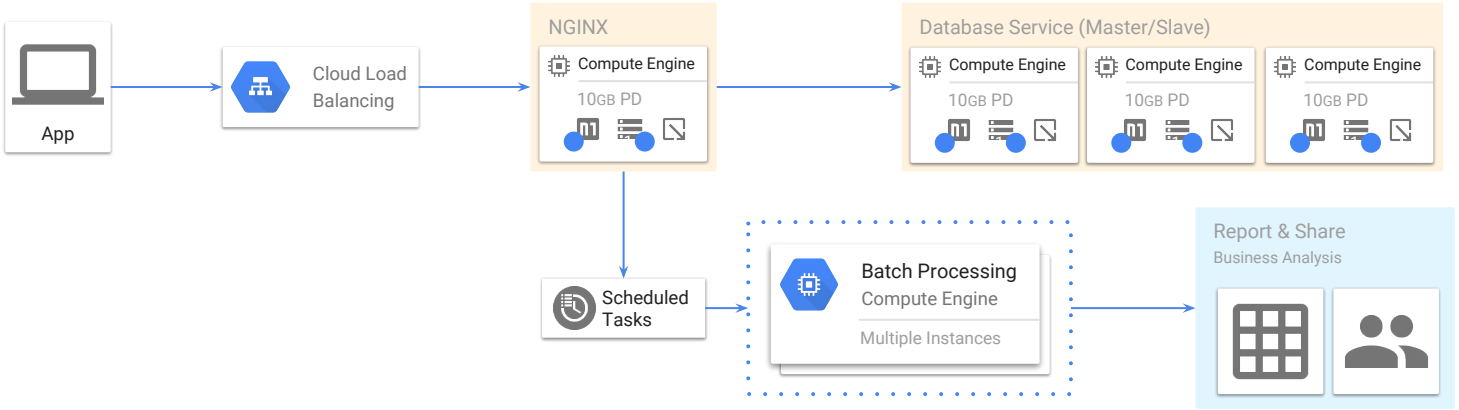
Sign up:
g.co/serverlesscontainers

Serverless means

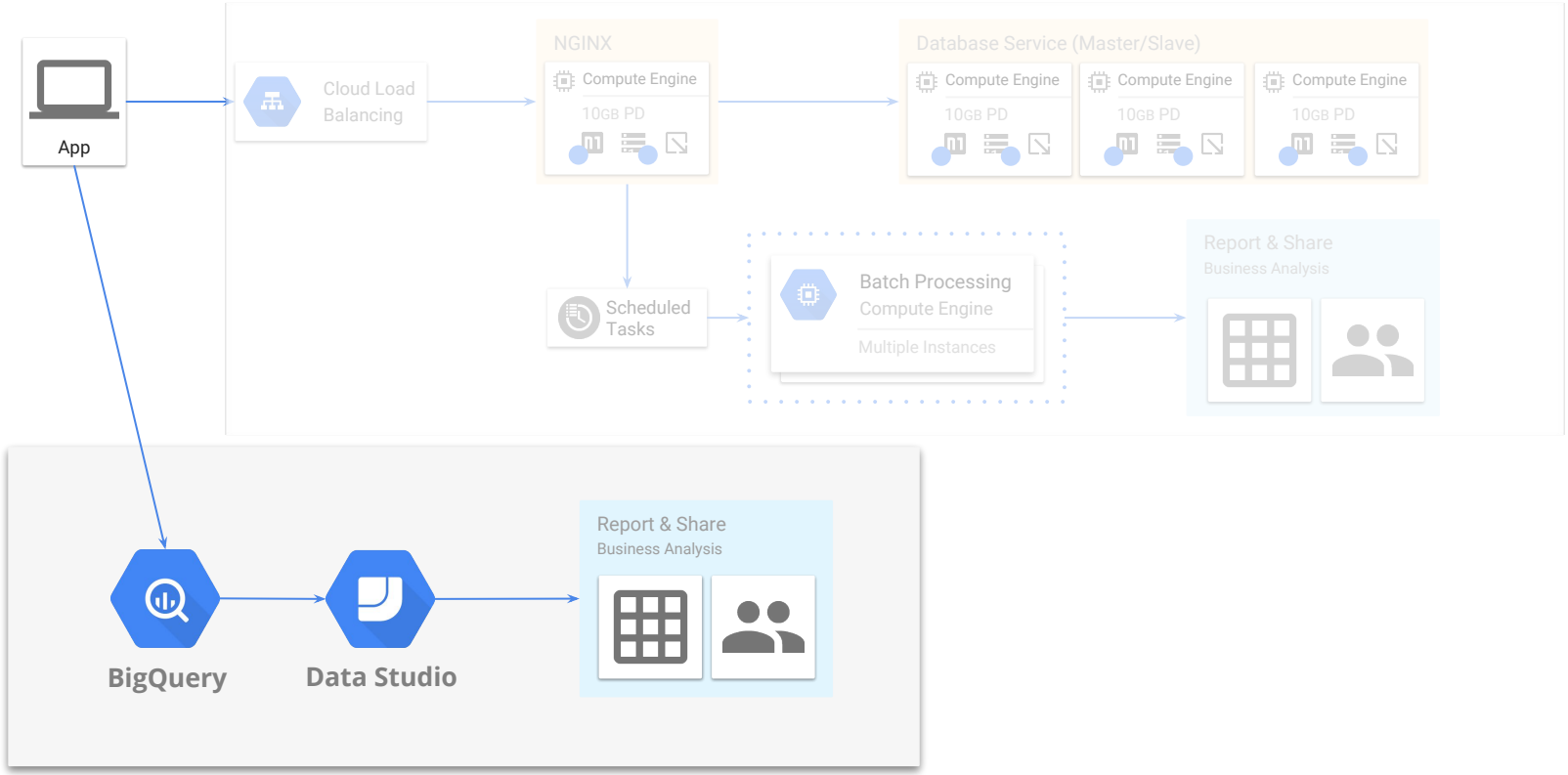
- No servers to provision or manage
- Scales with usage (ready every time for viral spikes or #BlackFriday)
- Availability and fault tolerance built in
- No orchestration in code
- Never pay for idle
- Decoupled: APIs as contracts (Abstract away the complexity)
- Monitored: Metrics and logging are a universal right
- Think concurrent, stateless, queue, stream based.

Reporting & Analytics moved to the Cloud

Legacy Reporting System



Serverless Reporting System




Google BigQuery

What is BigQuery?

Analytics-as-a-Service - Data Warehouse in the Cloud

Scales into Petabytes on Managed Google Infrastructure (US or EU zone)

SQL 2011 + Javascript UDF (User Defined Functions)

Familiar DB Structure (table, views, struct, nested, JSON)

Integrates with Google Sheets + Cloud Storage + Pub/Sub connectors

Decent pricing (queries `$5/TB`, storage: `$20/TB` cold: `$10/TB`) `*June 2018`

Open Interfaces (Web UI, BQ command line tool, REST, ODBC)

BigQuery: Convenience of SQL

Columnar storage (max 10 000 columns in table)

Large files for loading: `5TB` (`CSV` or `JSON`)

UDF in Javascript or SQL

Append-only tables prefered (DML syntax available)

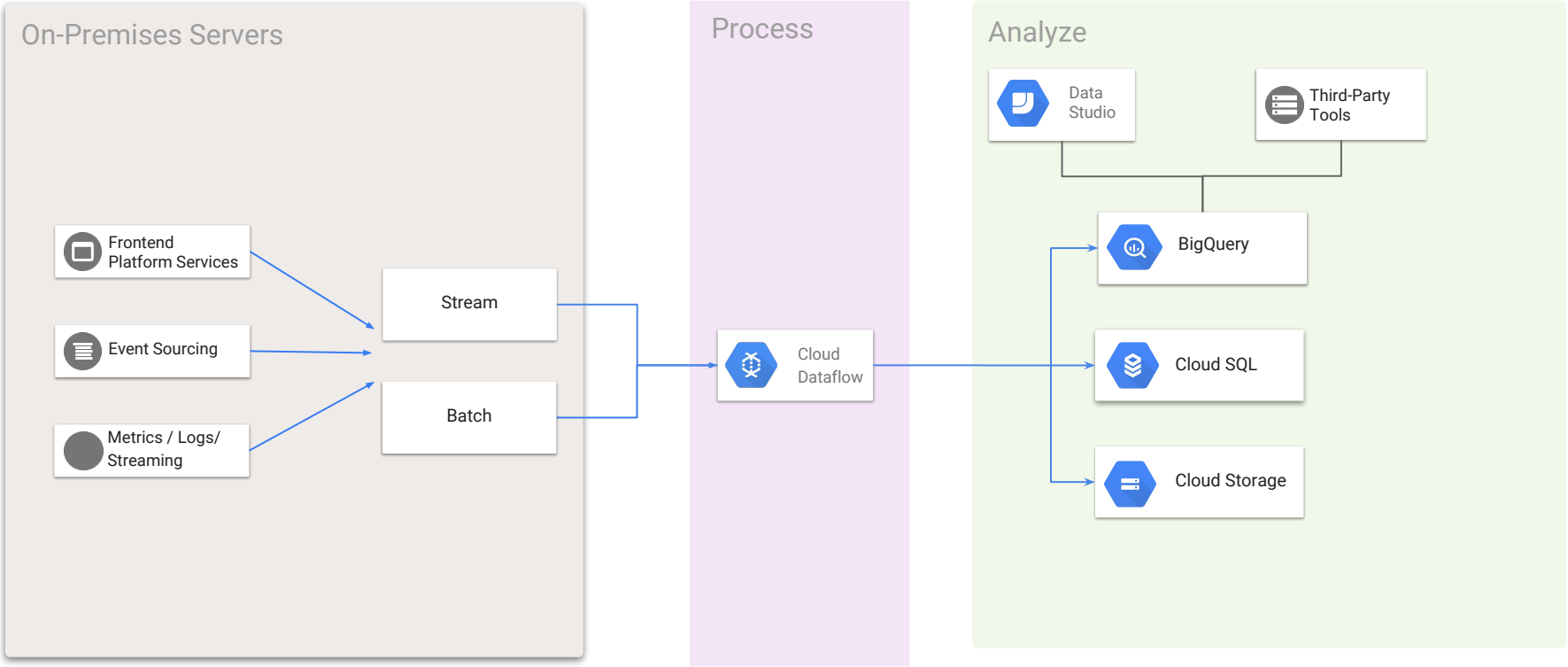
Day column partitioned tables (`select * from t where day='2018-01-01'`)

Rich SQL 2011: `JSON,IP,Math,RegExp,Geocode,Window functions`

Modern data types: `Record, Nested, Struct, Array`

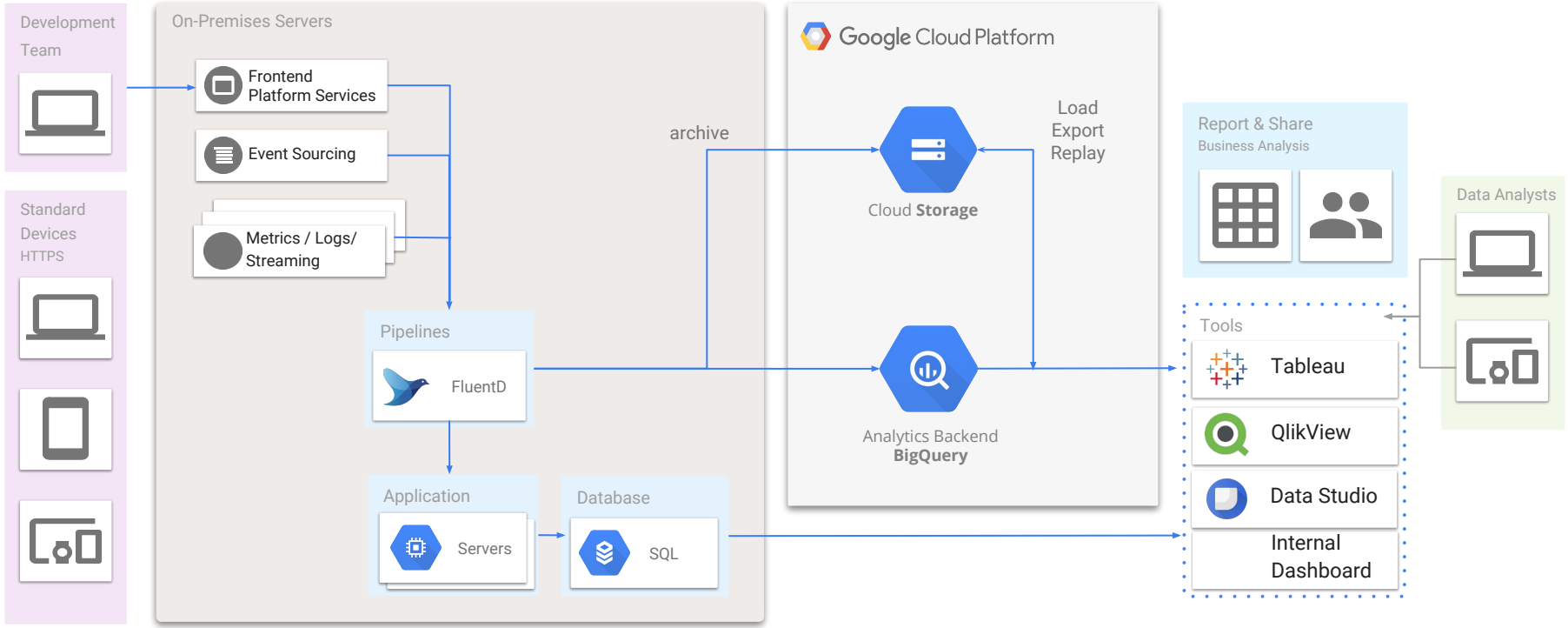
Data needs to be processed in multiple services.
How can we pipe to multiple places?

Architecting for The Cloud



We have our app outside of GCP.
How can we use the benefits of BigQuery?

Data Pipeline Integration at REEA.net



Our benefits using BigQuery

- SQL language to run BigData queries
- run raw ad-hoc queries (either by analysts/sales or Devs)
- no more throwing away-, expiring-, aggregating old data
- no provisioning/deploy
- no running out of resources
- no more focus on large scale execution plan

Summary: When to use what

Cloud Functions

Smallest unit of computing

Event driven architecture

Connect & extend services

App Engine

Unit of computing is apps

HTTP request/response

Large scalable backends

Serverless add-on

Run functions, apps & containers on GKE

Full portability of your artifacts

Run on your own cluster

Thank you.

Slides available on: **slideshare.net/martonkodok**

Reea.net - Integrated web solutions driven by creativity to deliver projects.