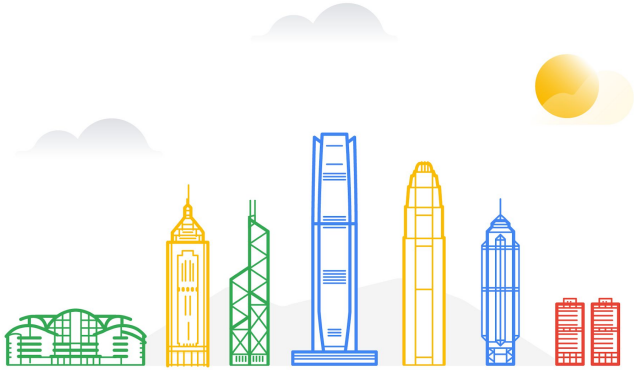
Google Cloud

Models at Runtime @ Google

15th International Workshop on Models@Run.Time
in conjunction with the
ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems
Fukuoka (福岡市), Japan *(virtual)*
11th October 2021

Speaker introduction

Ta'id Holmes
Industry Lead
Healthcare and Life Sciences
Google Cloud PSO EMEA



Abstract

This lightning talk studies Kubernetes as a real world production system with its Kubernetes Resource Model as well as the Open Policy Agent Gatekeeper and the policy language Rego.
Practical adoption possibilities for establishing causal connections in an industrial context are highlighted by showcasing a tool for the management of virtual machines at scale.

Part 1
Existing Systems and Technologies

A single day at Google

30,000+
developers

800,000
builds per day

9 million
source files

2 billion
lines of code

45,000
commits per workday

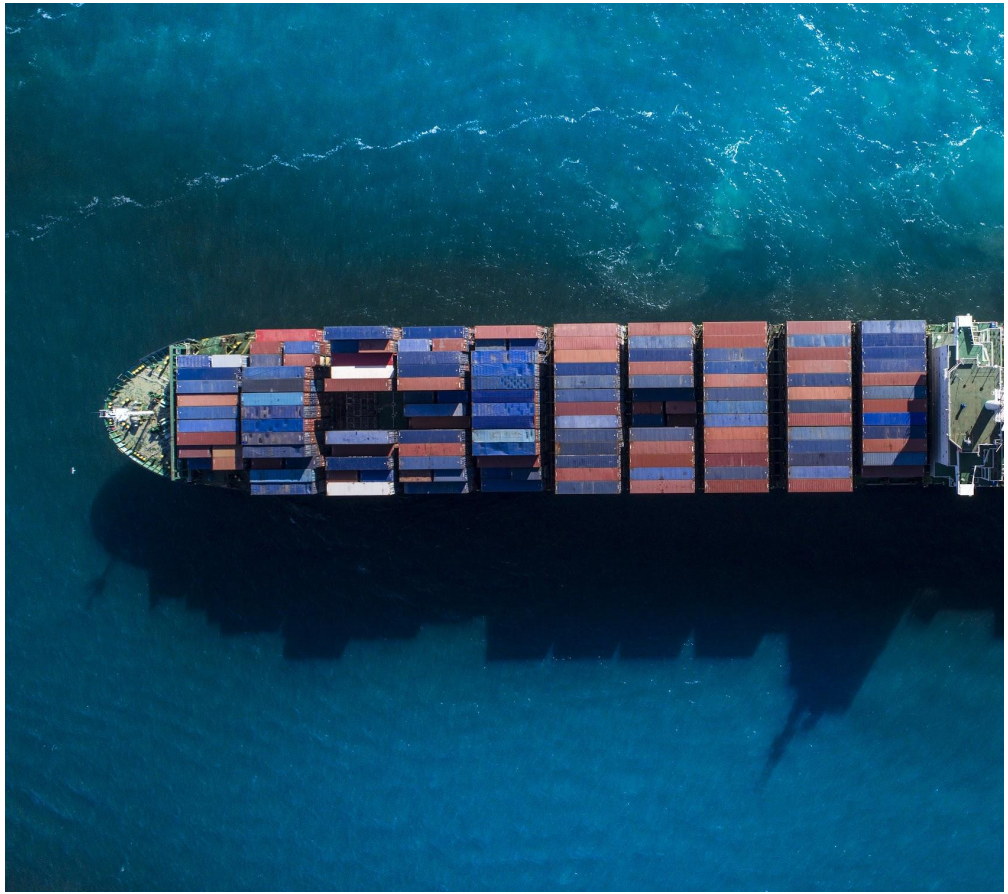
2+ PB
of build outputs per day

40+
engineering offices

150 million
test cases run per day

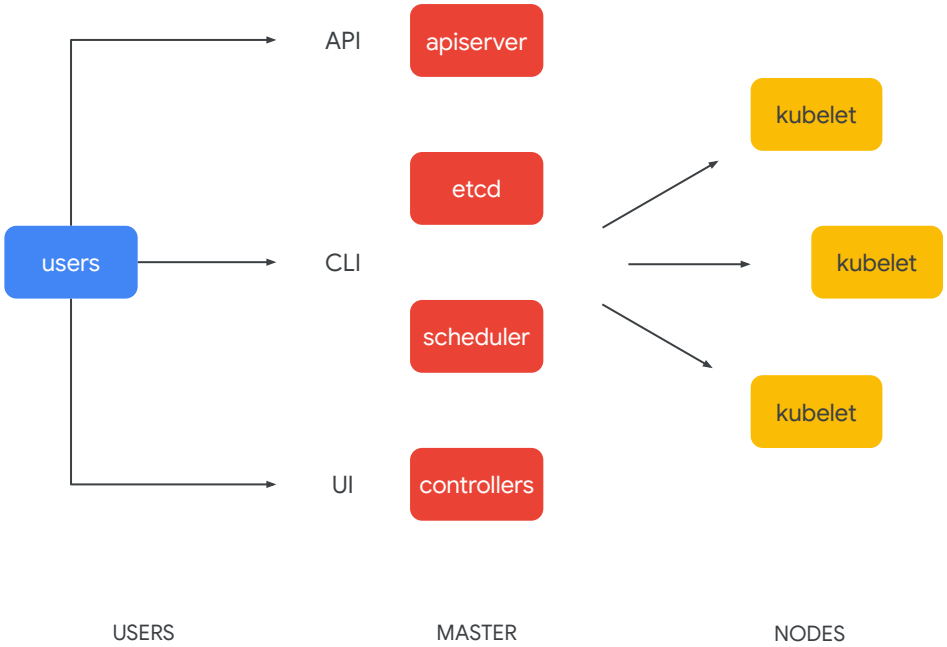
Google runs in containers

In any given week, we launch over two billion containers across Google.



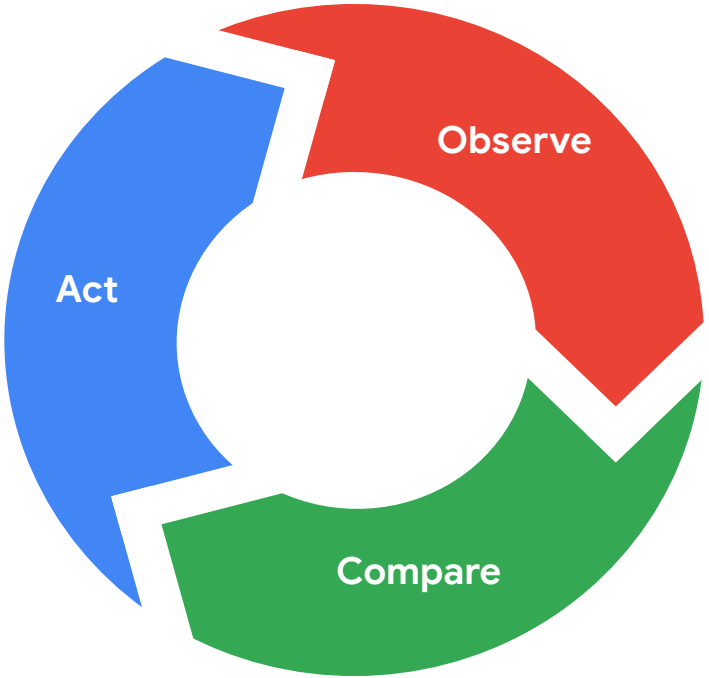
Kubernetes at a glance

- Manages container clusters
- Supports multi-cloud & bare metal environments
- Supports multiple container runtimes



Kubernetes control loop

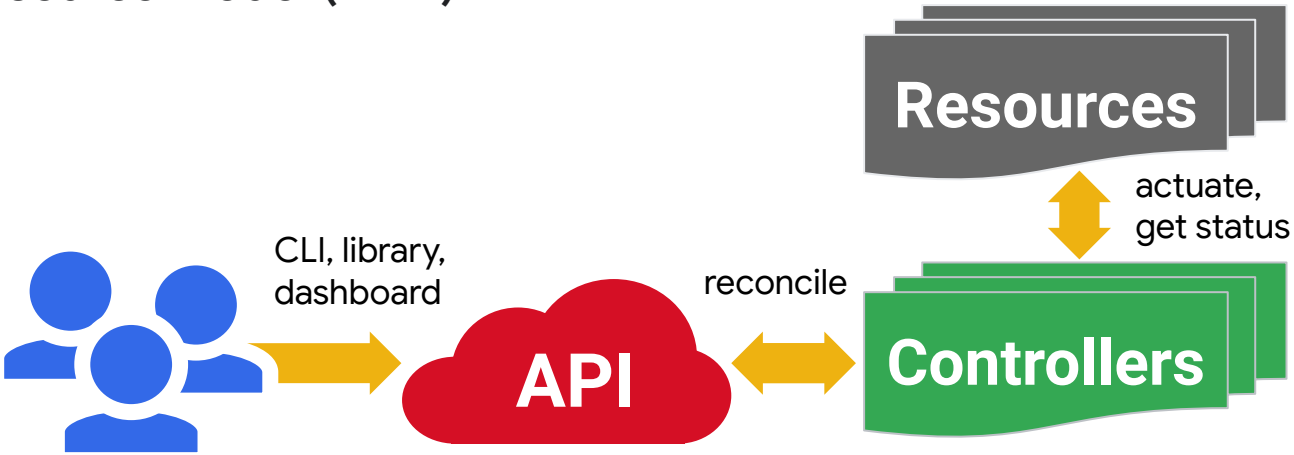
- **Observe** existing state.
- **Compare** existing state with declared state.
- **Act** on the system bring to the desire state.



Source: Brian Grant @ KubeCon 2017: What is Kubernetes?

Kubernetes resource model (KRM)

Concept



Realization:

The way Kubernetes models and manages resources is a useful general model.

https://github.com/kubernetes/community/blob/master/contributors/design-proposals/architecture/resource-management.md

Resource model

schema

```
apiVersion: apps/v1
kind: Deployment
```

metadata

```
metadata:
  name: hello-node
  namespace: test
  labels:
    app: hello-node
```

spec
=
desired state

```
spec:
  replicas: 3
  strategy:
    type: RollingUpdate
```

status
=
observed state

```
status:
  availableReplicas: 3
  conditions:
  - lastTransitionTime: "2021-04-30T14:13:34Z"
    message: ReplicaSet "hello-node-7567d9fdc9" has successfully progressed.
    reason: NewReplicaSetAvailable
    status: "True"
    type: Progressing
```

Model repository

“etcd is a strongly consistent, distributed key-value store that provides a reliable way to store data that needs to be accessed by a distributed system or cluster of machines”

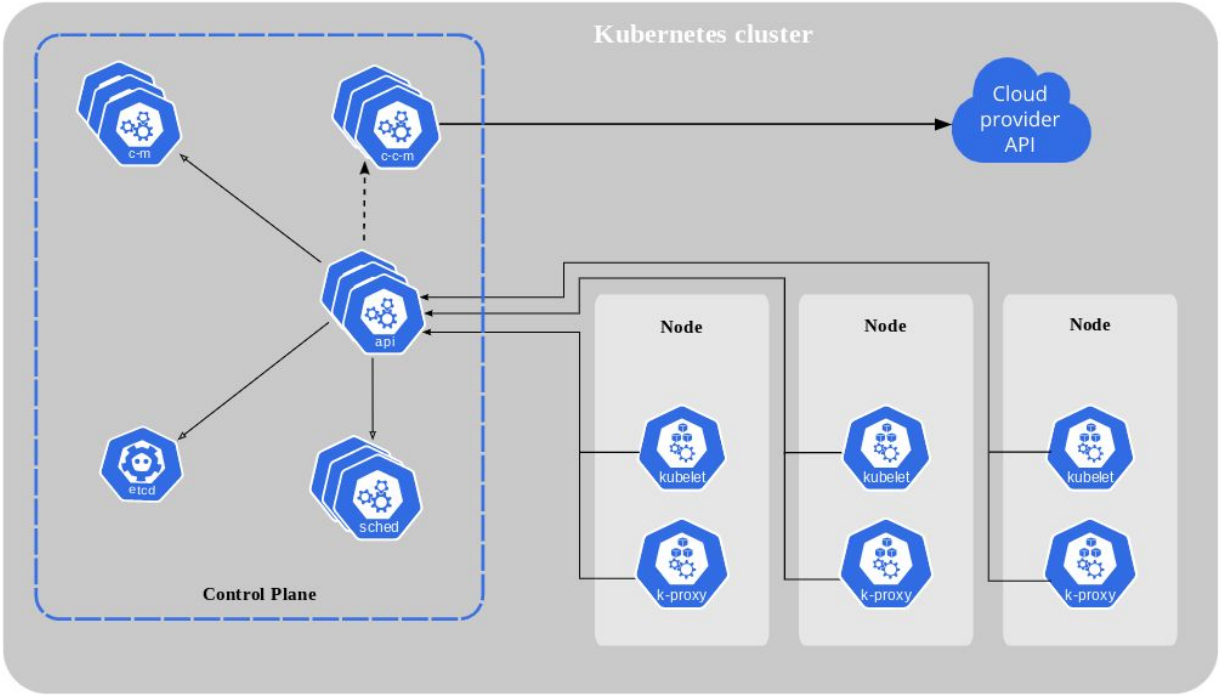
Source: https://etcd.io

Model consistency

“Raft is a consensus algorithm for managing a replicated log. It produces a result equivalent to (multi-)Paxos, and it is as efficient as Paxos, but its structure is different from Paxos; this makes Raft more understandable than Paxos and also provides a better foundation for building practical systems. In order to enhance understandability, Raft separates the key elements of consensus, such as leader election, log replication, and safety, and it enforces a stronger degree of coherency to reduce the number of states that must be considered. Results from a user study demonstrate that Raft is easier for students to learn than Paxos. Raft also includes a new mechanism for changing the cluster membership, which uses overlapping majorities to guarantee safety.” [1]

[1] Diego Ongaro and John Ousterhout. 2014. In search of an understandable consensus algorithm. In *Proceedings of the 2014 USENIX conference on USENIX Annual Technical Conference* (*USENIX ATC'14*). USENIX Association, USA, 305–320.

Kubernetes architecture



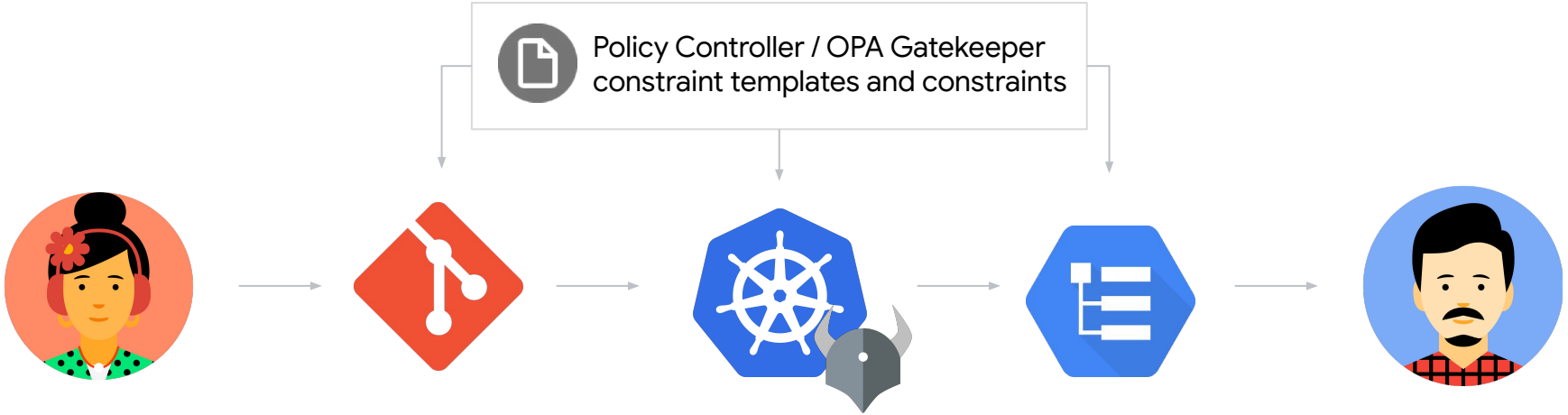
Source: https://kubernetes.io/docs/concepts/overview/components/

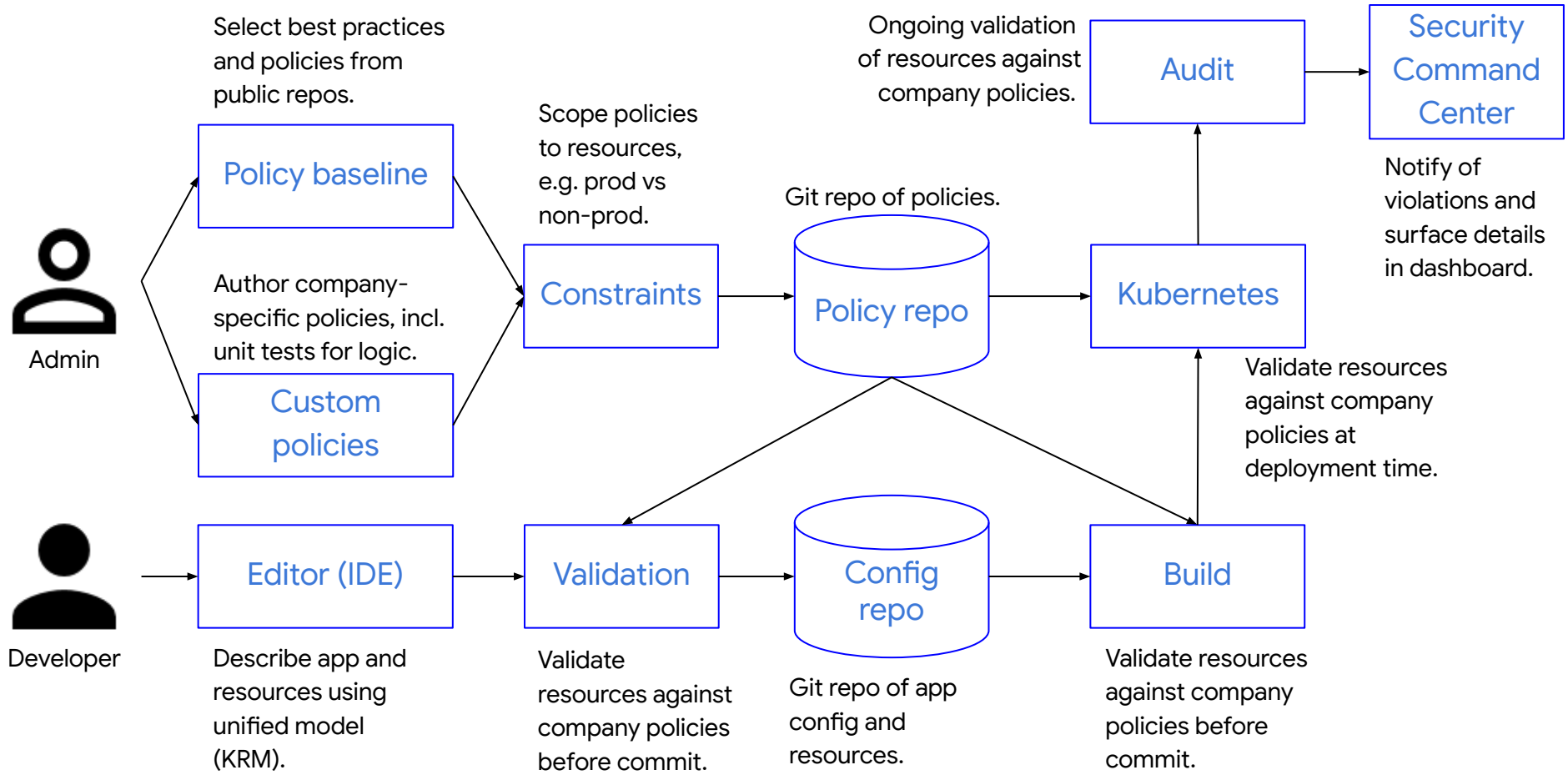
Model verification with Open Policy Agent (OPA) and Rego

“OPA policies (written in Rego) make decisions based on hierarchical structured data. Sometimes we refer to this data as a document, set of attributes, piece of context, or even just “JSON”. Importantly, OPA policies can make decisions based on arbitrary structured data. OPA itself is not tied to any particular domain model. Similarly, OPA policies can represent decisions as arbitrary structured data (e.g., booleans, strings, maps, maps of lists of maps, etc.)”

Source: https://www.openpolicyagent.org/docs/latest/philosophy

Policy journey





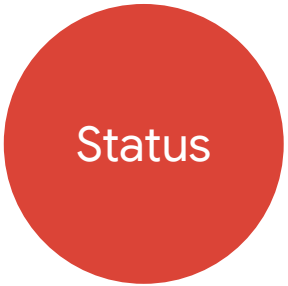
Part 2
Establishing causal connections

Practical approach in an industrial context

The following example showcases a practical approach for introducing some tooling based on models at runtime principles, that is establishing a causal connection between a model and system(s). Its scope is the management of virtual machines in the cloud at scale.

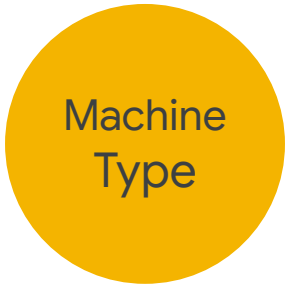
1. Find a problem
2. Uncover models at runtime
3. Establish a causal connection

Management of Virtual Machines - What



Start/Stop

Billing and Cross-Charging



Optimize

Rollout to Production

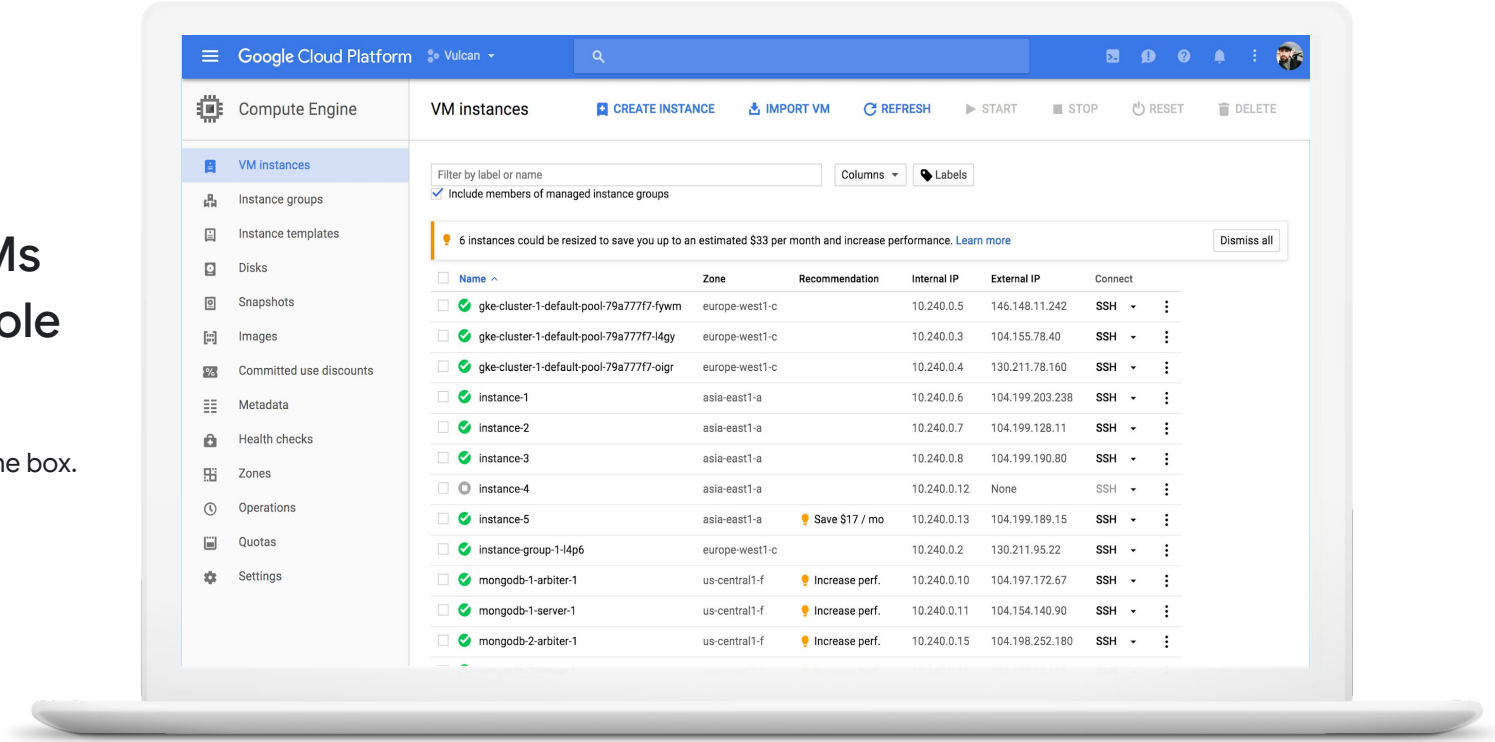
Management of Virtual Machines - How

GCP Console	Graphical User Interface; best fit for business users
Cloud SDK	set of libraries and tools interacting with and abstracting GCP APIs
gcloud	CLI for interactive commands and shell scripts
custom scripts	using the Cloud SDK client libraries or interacting with GCP APIs

Managing VMs in GCP Console

Managing VMs at scale

does not come out of the box.



Integrating Stakeholders through Models

Spreadsheet as a Domain Specific Language

A spreadsheet would be something

- many users are familiar with
- with powerful native functionalities

Google to the help !

Search & Sorting

the algorithms come out of the box

Filtering

e.g., by instance name, machine type, or labels

Views

Represent a subset of instances and/or data as appropriate for additional collaborateurs

Copy & Paste

For efficiency and for saving time.

Demo

Thank you.

Google Cloud