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

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developers</td>
<td>30,000+</td>
</tr>
<tr>
<td>Builds per day</td>
<td>800,000</td>
</tr>
<tr>
<td>Source files</td>
<td>9 million</td>
</tr>
<tr>
<td>Lines of code</td>
<td>2 billion</td>
</tr>
<tr>
<td>Commits per workday</td>
<td>45,000</td>
</tr>
<tr>
<td>Build outputs per day</td>
<td>2+ PB</td>
</tr>
<tr>
<td>Engineering offices</td>
<td>40+</td>
</tr>
<tr>
<td>Test cases run per day</td>
<td>150 million</td>
</tr>
</tbody>
</table>
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?](https://example.com)
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

apiVersion: apps/v1
kind: Deployment

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

spec:
  replicas: 3
  strategy:
    type: RollingUpdate

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
“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]

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](https://www.openpolicyagent.org/docs/latest/philosophy)
Policy journey

- Alice
  - App Operator
  - Check config changes pre-deploy

- Authoring

- Admission
  - Block non-compliant changes at the K8s API server

- Audit
  - Alert on compliance violations in the live environment

- Charlie
  - Platform Admin
  - Policy Controller / OPA Gatekeeper constraint templates and constraints
Select best practices and policies from public repos.

Policy baseline

Custom policies

Author company-specific policies, incl. unit tests for logic.

Constraints

Scope policies to resources, e.g. prod vs non-prod.

Policy repo

Git repo of policies.

Kubernetes

Validate resources against company policies at deployment time.

Build

Validate resources against company policies before commit.

Config repo

Git repo of app config and resources.

Validation

Developer

Describe app and resources using unified model (KRM).

Editor (IDE)

Admin

Ongoing validation of resources against company policies.

Audit

Notify of violations and surface details in dashboard.

Security Command Center
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

- Status: Start/Stop
- Labels: Billing and Cross-Charging
- Machine Type: Optimize
- Network Tags: Rollout to Production
<table>
<thead>
<tr>
<th><strong>Management of Virtual Machines - How</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GCP Console</strong></td>
</tr>
<tr>
<td><strong>Cloud SDK</strong></td>
</tr>
<tr>
<td><strong>gcloud</strong></td>
</tr>
<tr>
<td><strong>custom scripts</strong></td>
</tr>
</tbody>
</table>
Managing VMs in GCP Console

Managing VMs at scale does not come out of the box.
Integrating Stakeholders through Models

Not every user may be a GCP whizz...

... but you may need to involve business stakeholders!
Spreadsheet as a Domain Specific Language

A well-suited interface

A spreadsheet would be something
- many users are familiar with
- with powerful native functionalities
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.