Containerizing the Cloud

with Kubernetes and Docker

Google Cloud Platform Developer Roadshow - 2014

Google Cloud Platform

What is a container?

- Lightweight Linux environment
- Hermetically sealed, deployable application
- Introspectable, runnable artifact
- Recently popularized by Docker

Why do developers care?

Static application environment
= reliable deployments

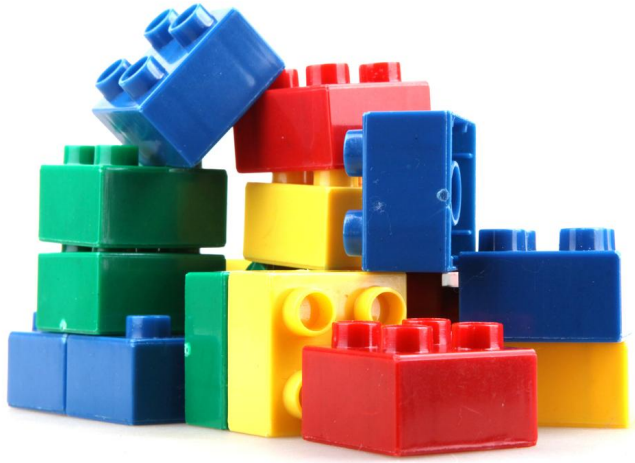
No stress deployment and update

Repeatable, runnable artifact
= portability

Develop here, run there
Pick your cloud solely on its merits

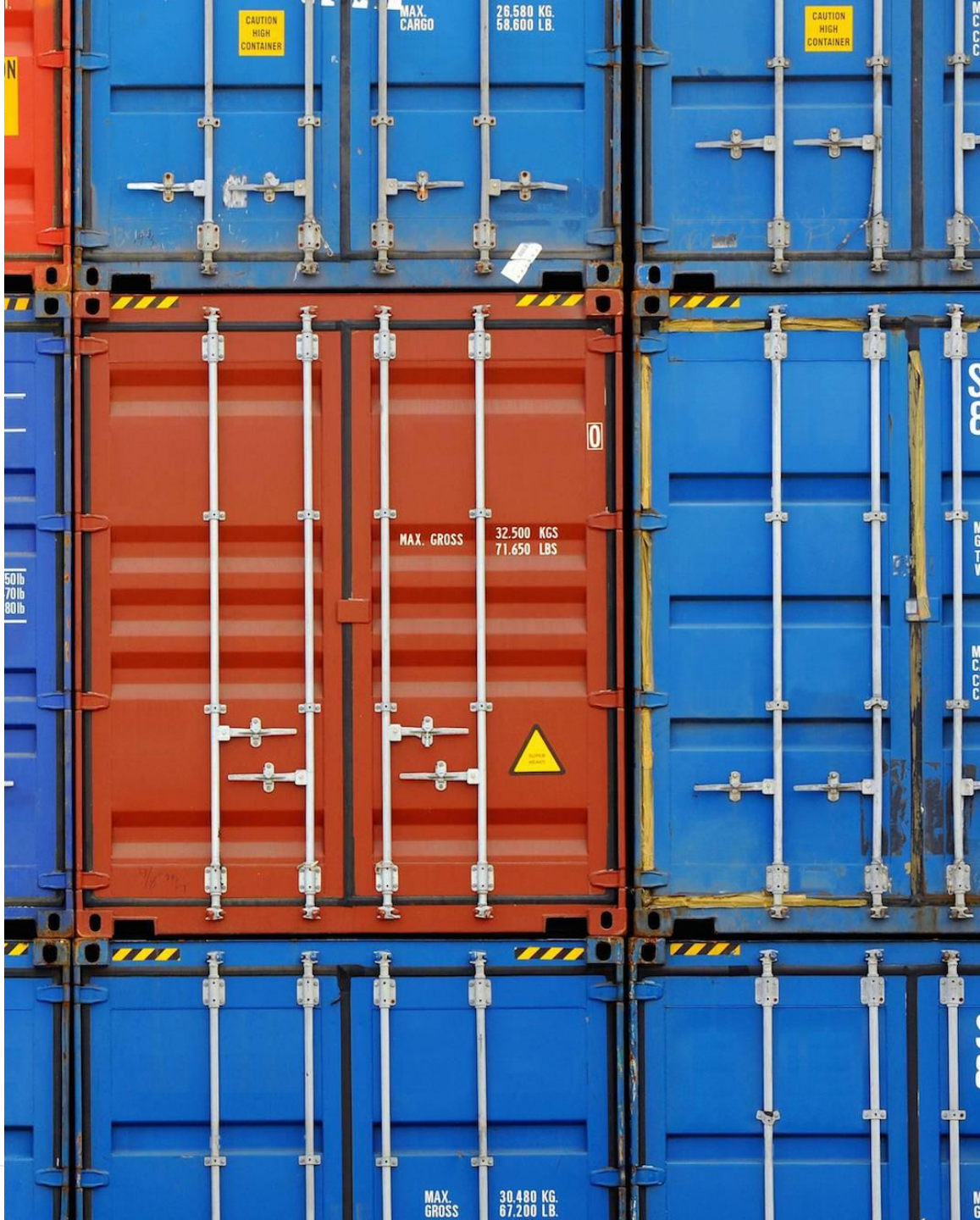
Loosely coupled
= easier to build and manage

Compose applications from micro-services
Mix in and extend third party services



Everything at Google runs in a container.

- Resource isolation
- Predictability
- Quality of service
- Efficient overcommit
- Resource accounting

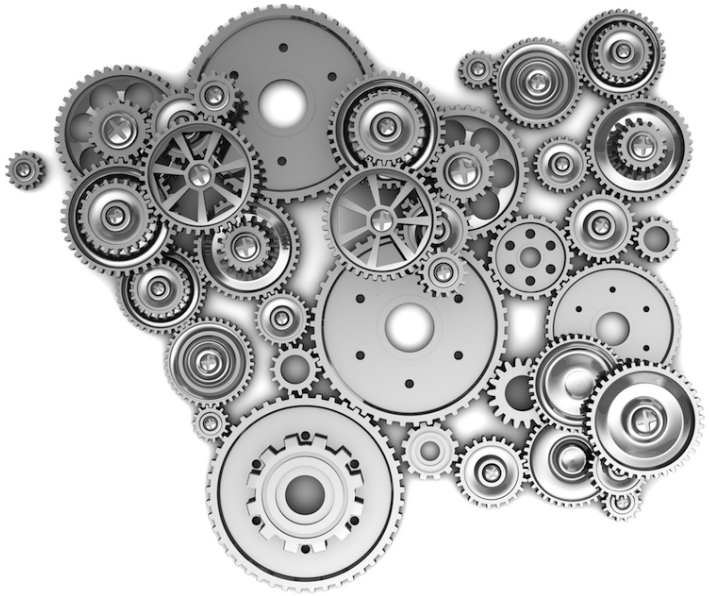


We start over 2 billion containers per week.

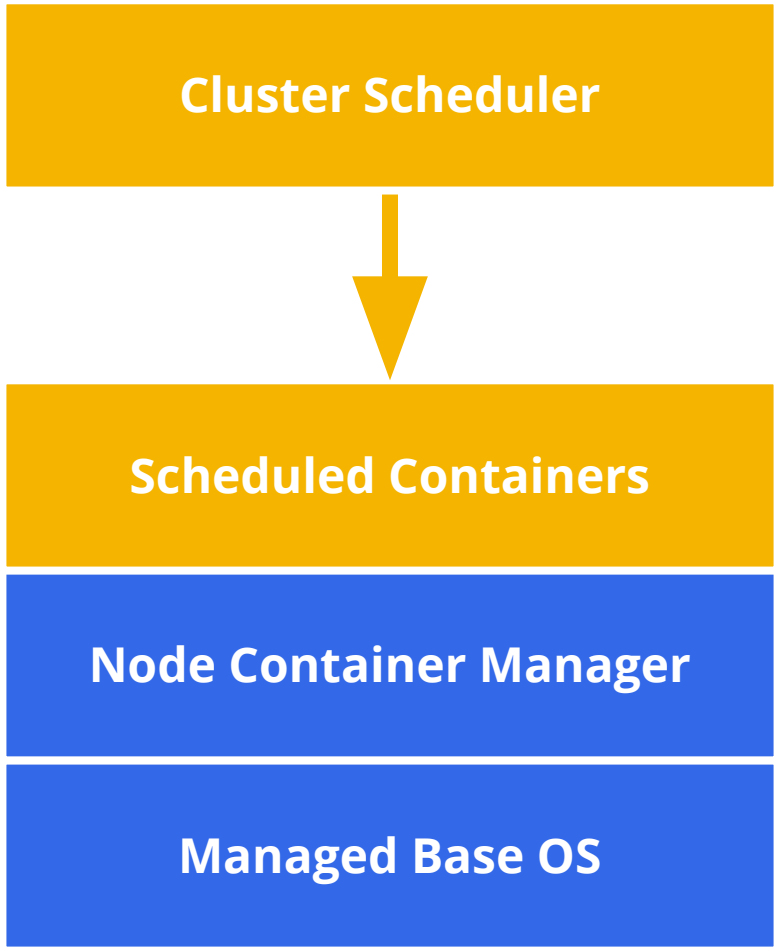
Why do developers care?

Highly automatable = path to active management

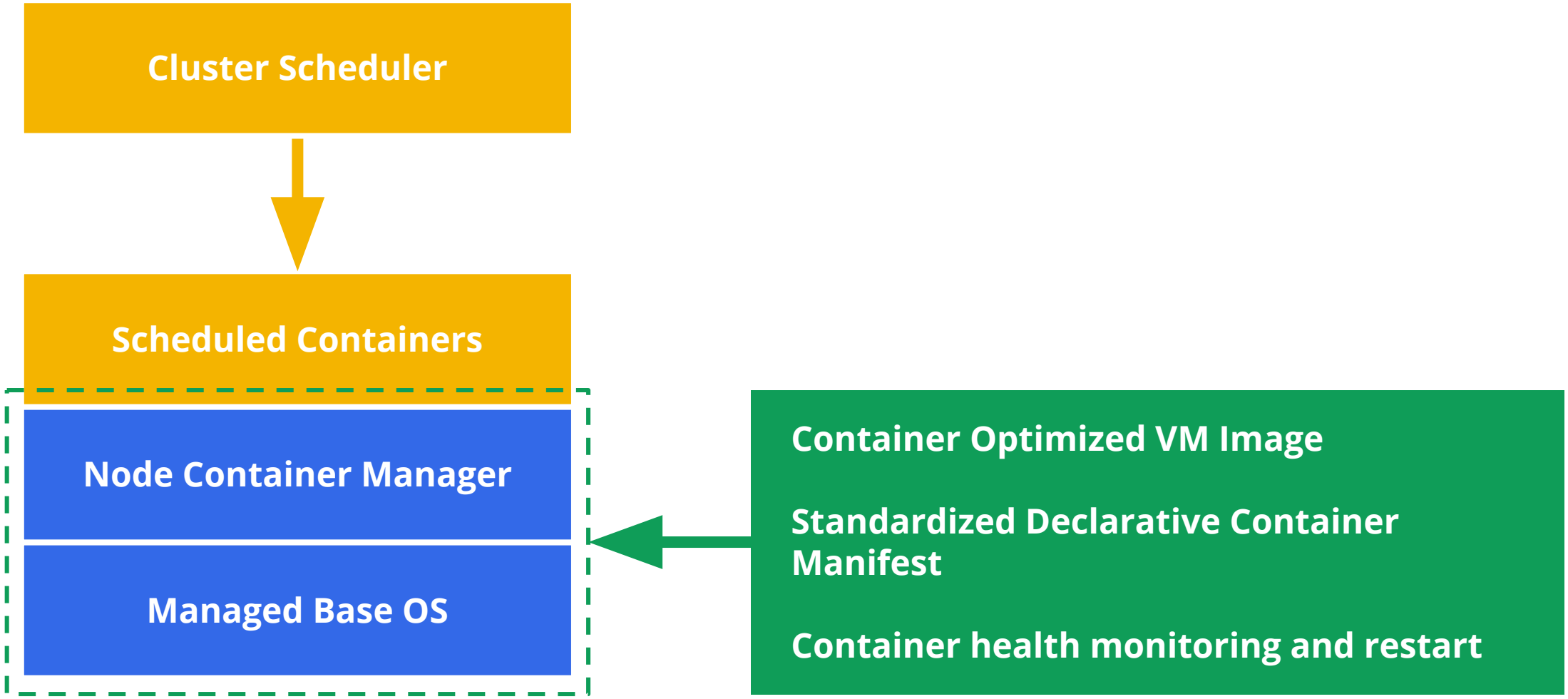
- *Efficiency*: optimized packing, better scaling
- *Performance*: active environment tuning
- *Continuous integration*: easy and reliable
- *Robustness*: active monitoring, self healing



Google cluster management stack



Node container management on the Google Cloud



Example with nginx

command line

```
$ gcloud compute instances create my-nginx-container
    --metadata-from-file google-container-manifest=containers.yaml
    --zone us-central1-a
    --machine-type f1-micro
    --image projects/google-containers/global/images/container-vm-v20140522
```

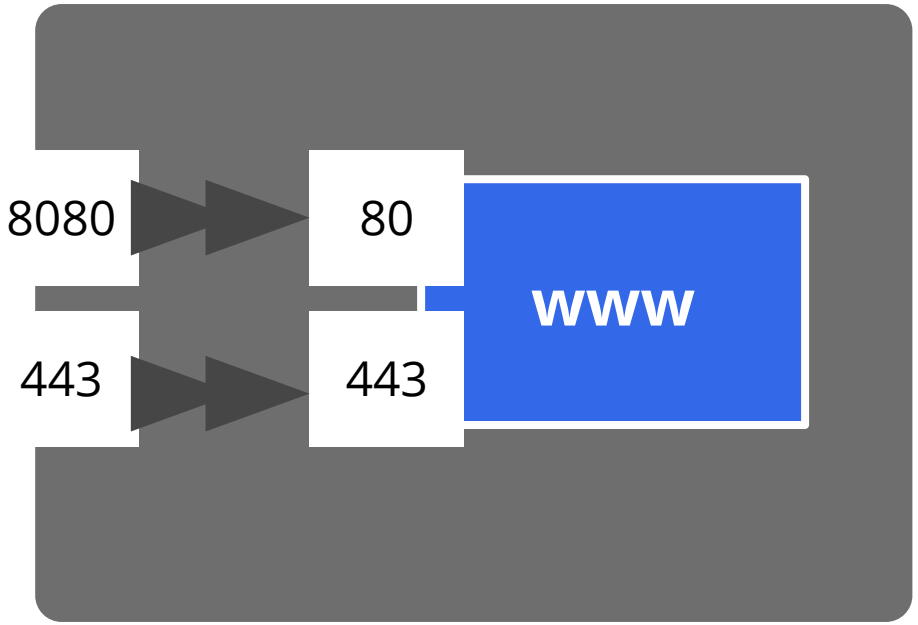
containers.yaml

```
version: v1beta1
containers:
  - name: www
    image: nginx
    ports:
      - name: http
        hostPort: 8080
        containerPort: 80
```

Example with nginx

containers.yaml

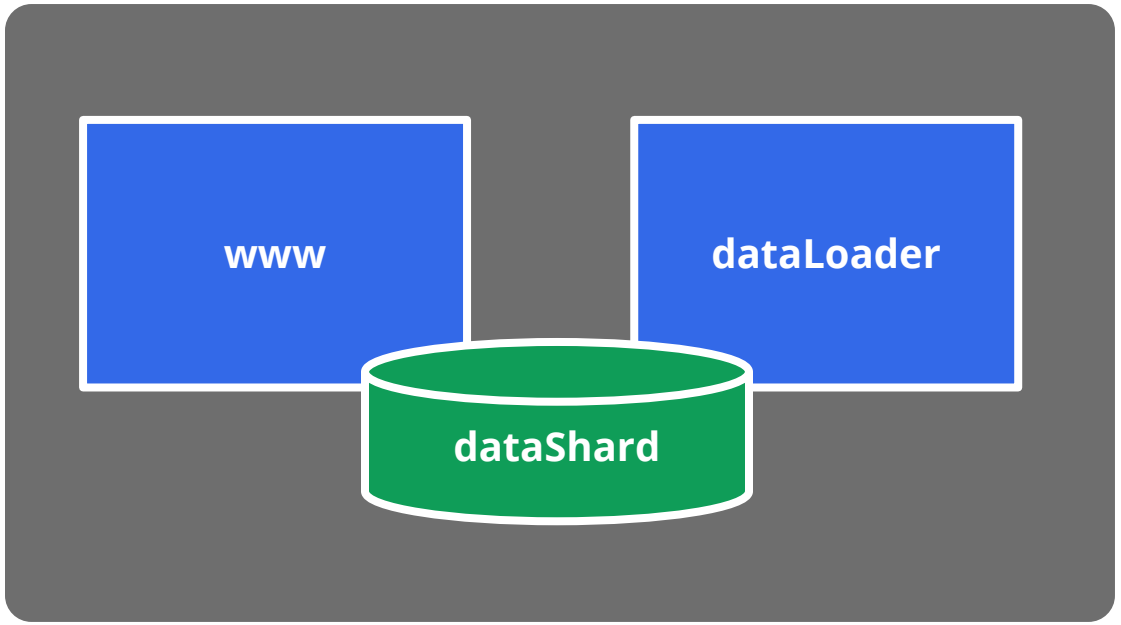
```
version: v1beta1
containers:
  - name: www
    image: nginx
    ports:
      - name: http
        hostPort: 8080
        containerPort: 80
      - name: https
        hostPort: 443
        containerPort: 443
```

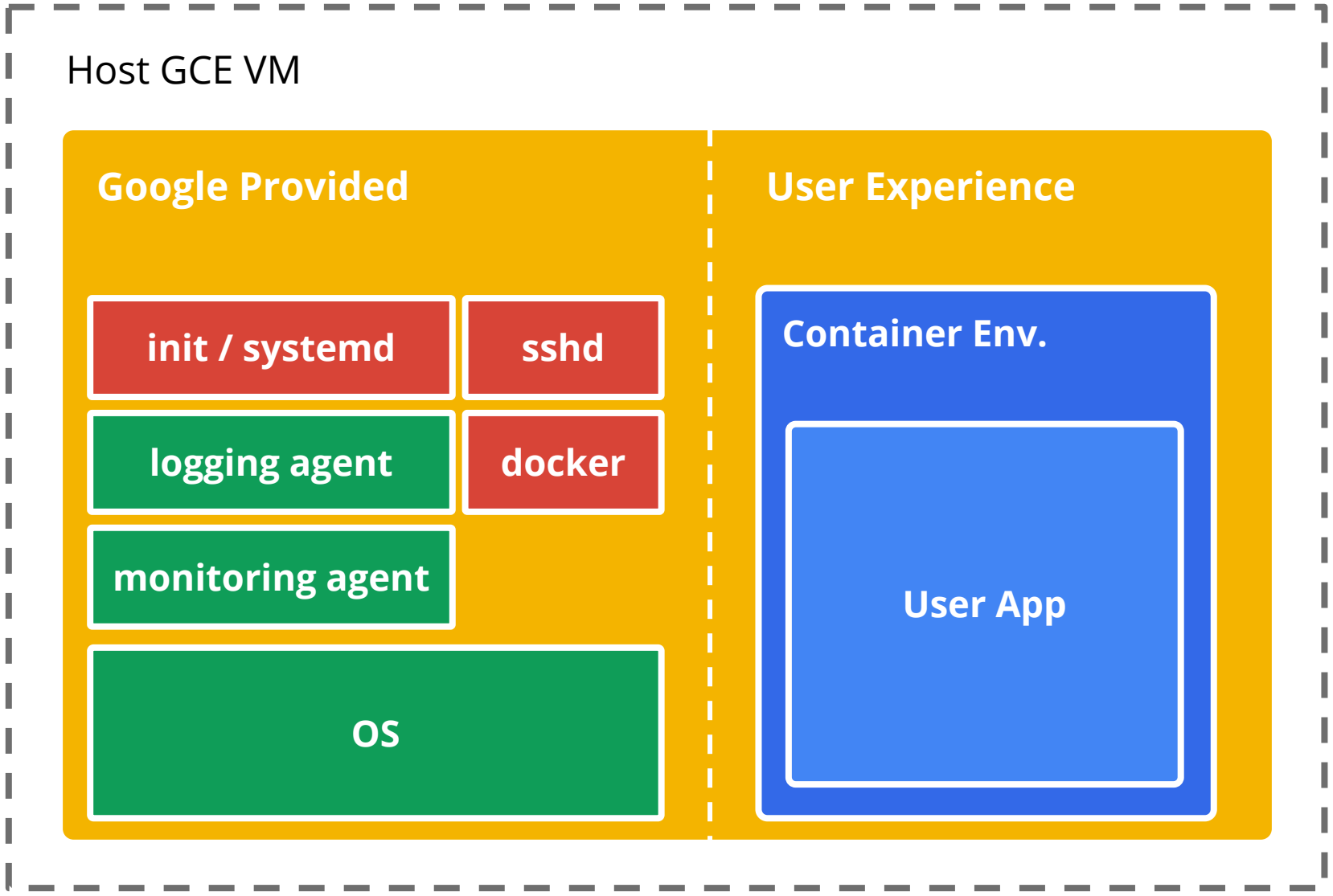


Example with data sharding

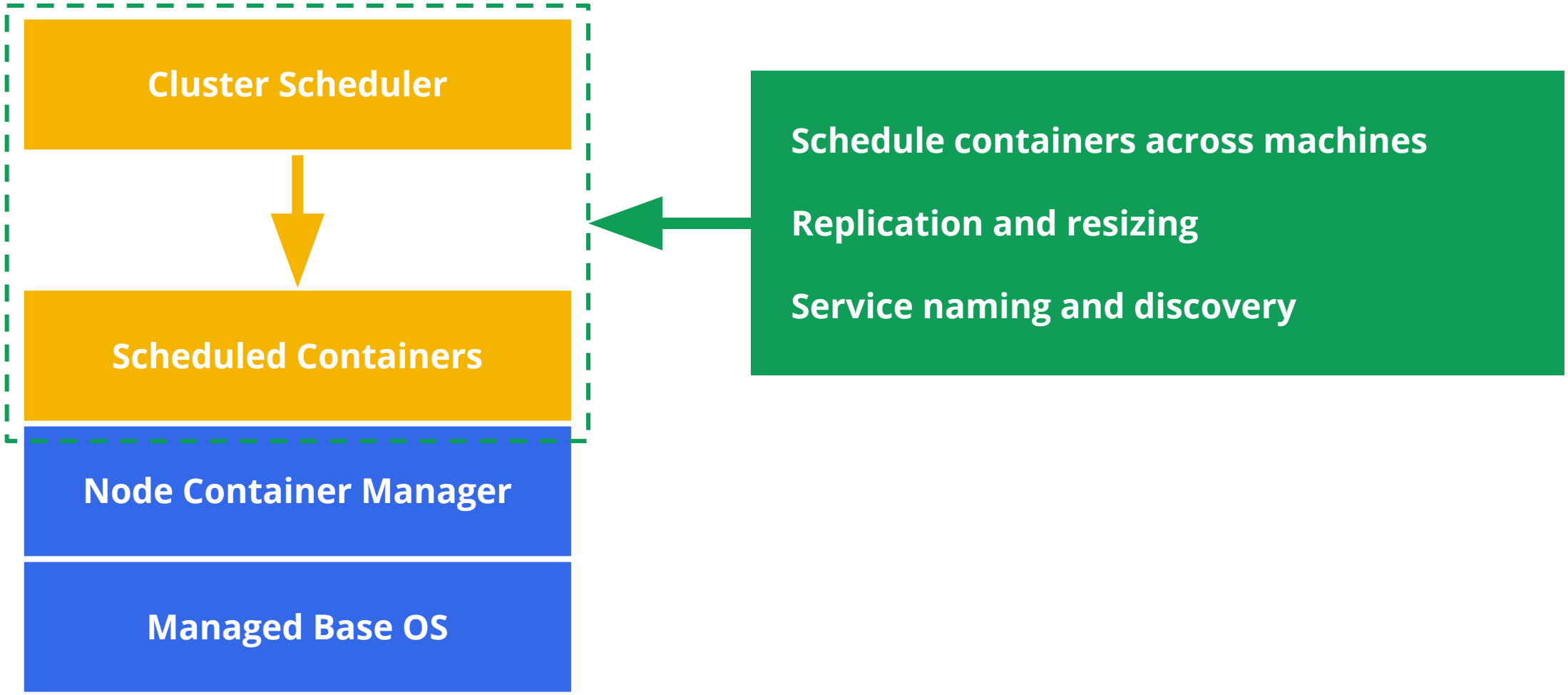
containers.yaml

```
version: v1beta1
containers:
  - name: www
    …
    volumeMounts:
      - name: dataShard
        path: /mnt/shard
        readOnly: true

  - name: dataLoader
    …
    volumeMounts:
      - name: dataShard
        path: /mnt/output

volumes:
  - name: dataShard
```




Host GCE VM
Google Provided
User Experience
init / systemd
sshd
logging agent
docker
monitoring agent
OS
Container Env.
User App

Cluster container scheduling on the Google Cloud



But it takes a community...

Containers are **portable**. The active management framework must be portable too:

- Run on your development machine.
- Run on your cloud.
- Run on Google Cloud Platform.
- Run a different provider or your own hardware.

And it must be enterprise grade.

The community is working to create a framework that runs well everywhere: **Kubernetes**

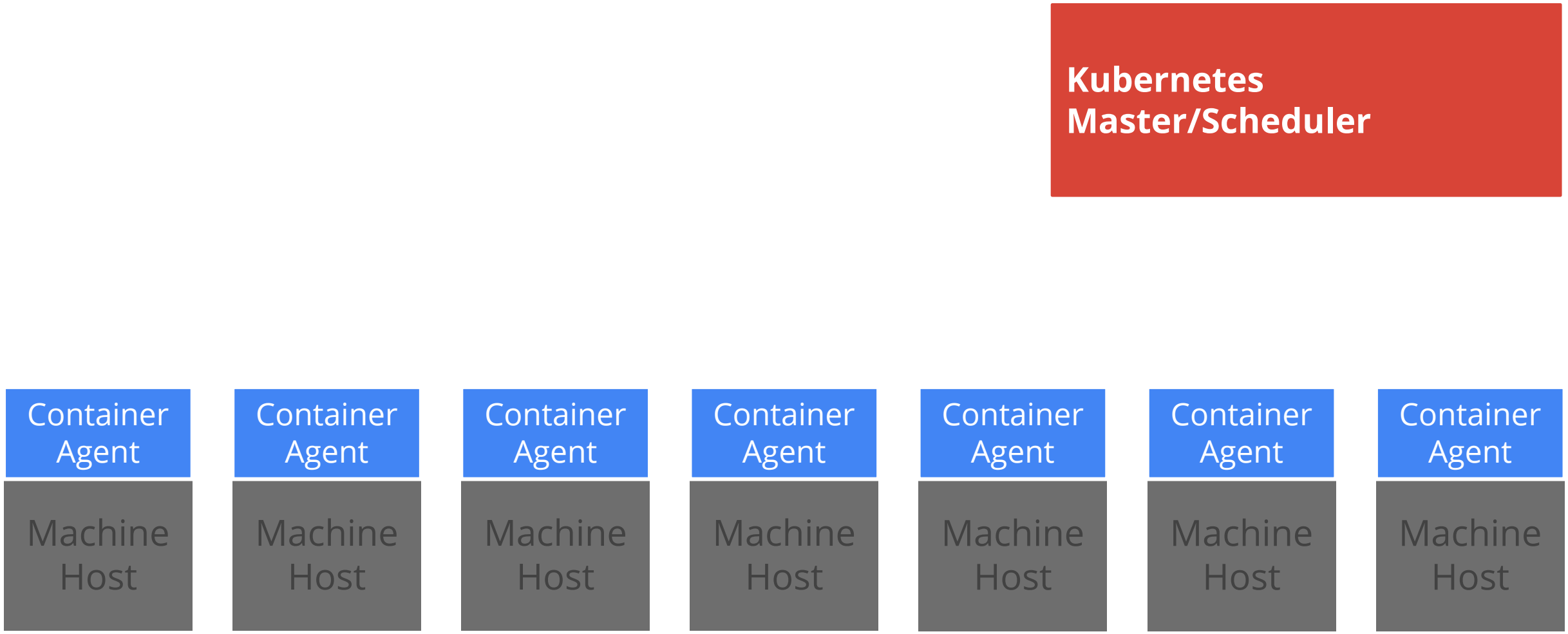
Microsoft, IBM, Red Hat, Docker, Mesosphere, SaltStack, and CoreOS, have joined the family.

Kubernetes

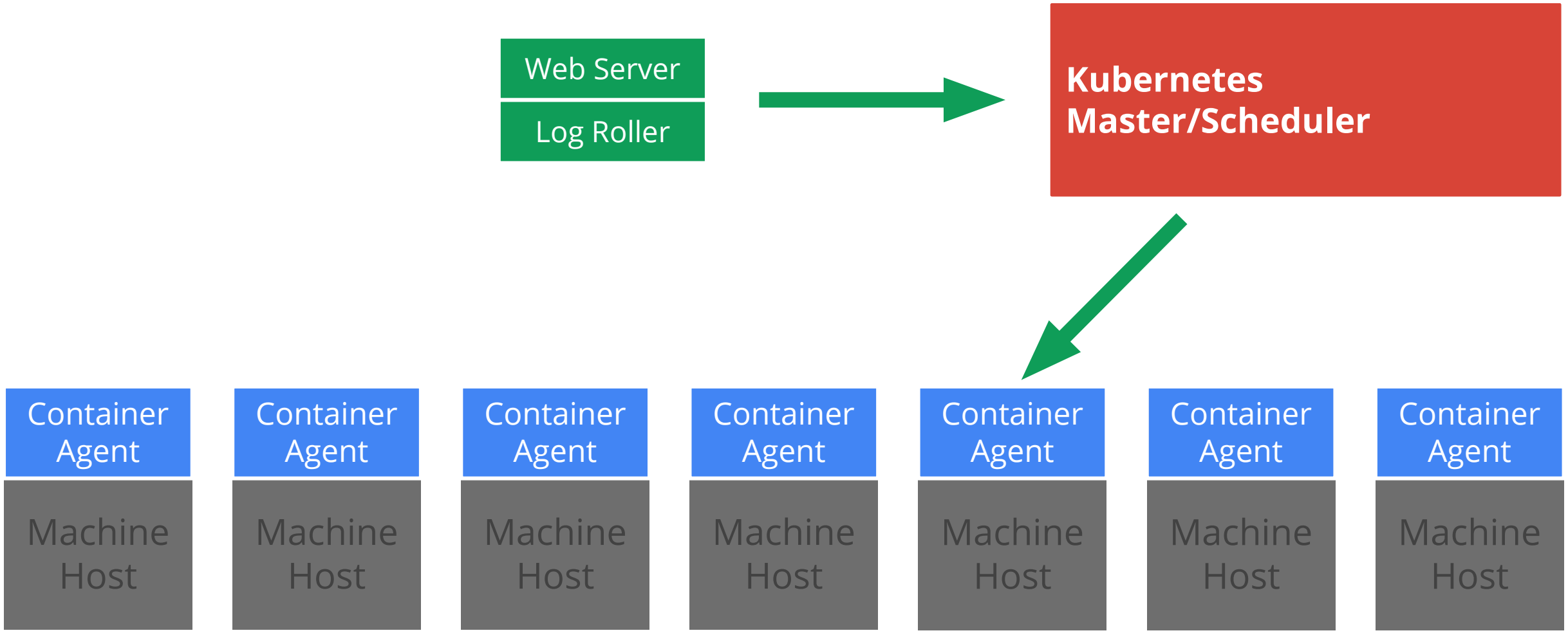
κυβερνήτης: *Greek for “pilot” or “helmsman of a ship”*

the open source cluster manager from Google

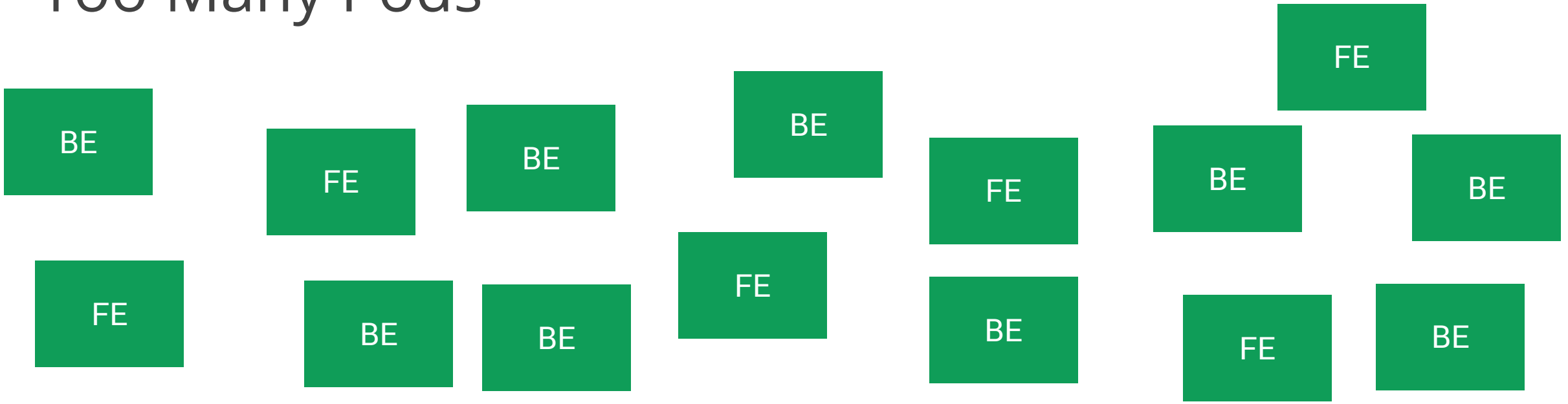
Kubernetes


Kubernetes
Master/Scheduler
Container
Agent
Machine
Host
Container
Agent
Machine
Host
Container
Agent
Machine
Host
Container
Agent
Machine
Host
Container
Agent
Machine
Host
Container
Agent
Machine
Host
Container
Agent
Machine
Host

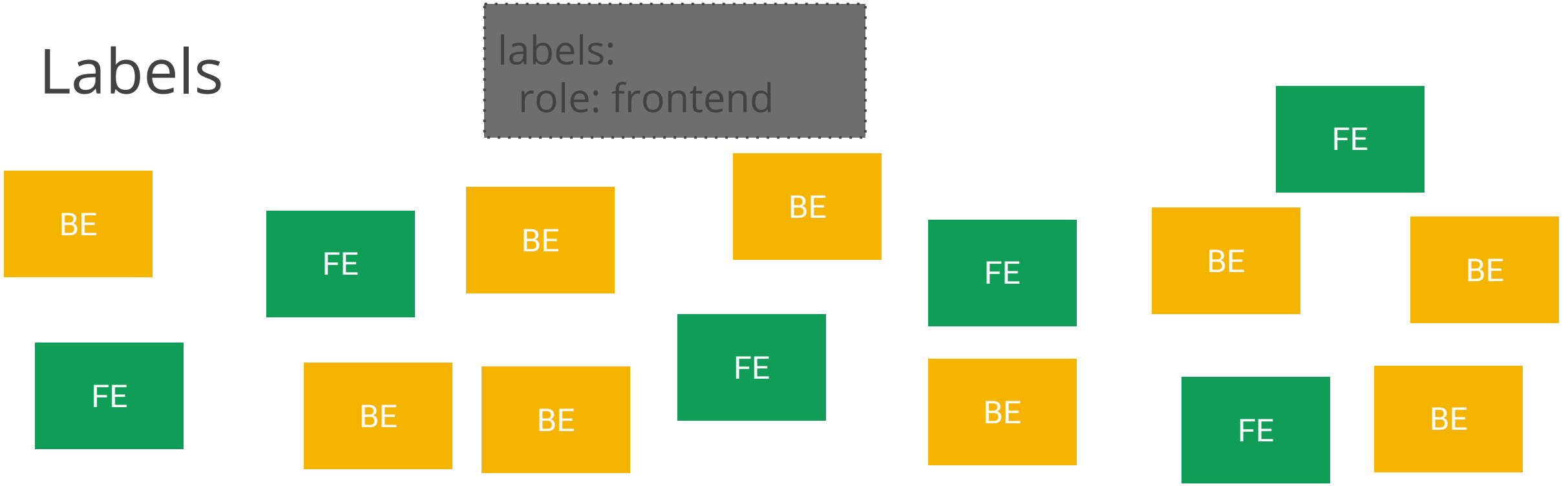
Pods


Web Server
Log Roller
Kubernetes
Master/Scheduler
Container Agent
Machine Host
Container Agent
Machine Host
Container Agent
Machine Host
Container Agent
Machine Host
Container Agent
Machine Host
Container Agent
Machine Host
Container Agent
Machine Host

Too Many Pods



Labels

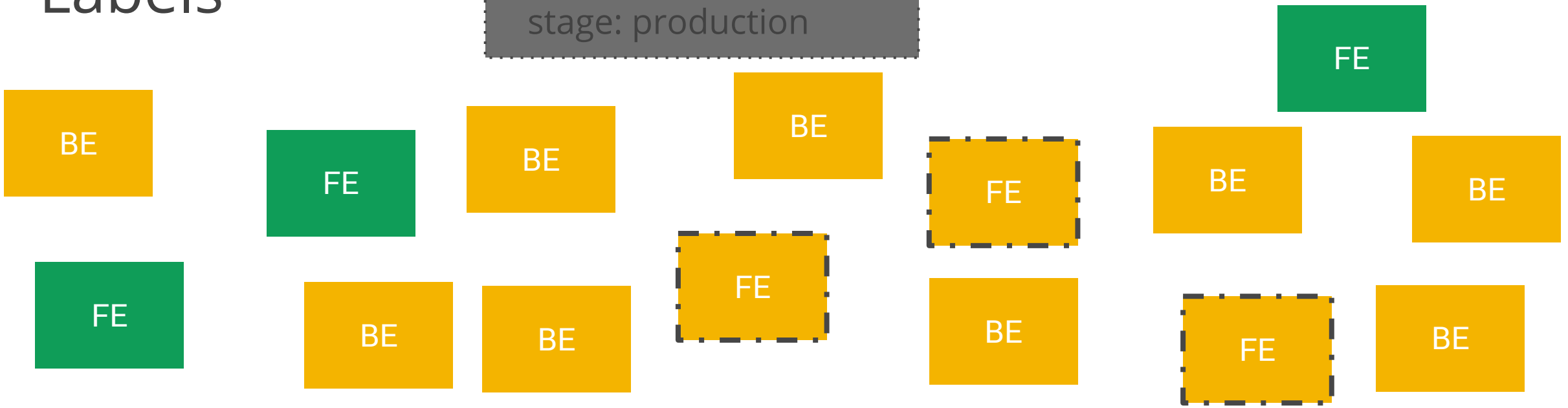


Kubernetes - Master/Scheduler

Container Agent	Container Agent	Container Agent	Container Agent	Container Agent	Container Agent	Container Agent
Machine Host	Machine Host	Machine Host	Machine Host	Machine Host	Machine Host	Machine Host

Labels

labels:
role: frontend
stage: production



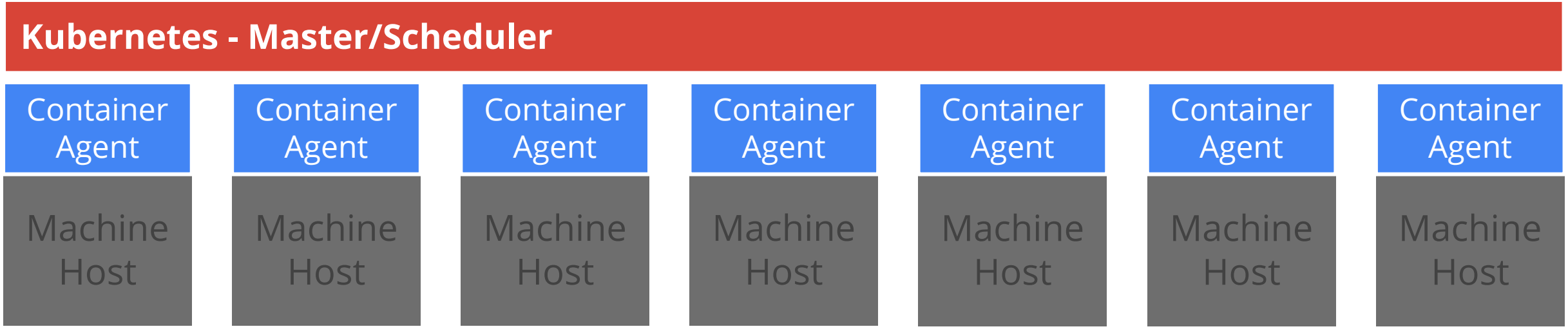
Kubernetes - Master/Scheduler

Container Agent	Container Agent	Container Agent	Container Agent	Container Agent	Container Agent	Container Agent
Machine Host	Machine Host	Machine Host	Machine Host	Machine Host	Machine Host	Machine Host

Replica Controller

FE FE FE FE

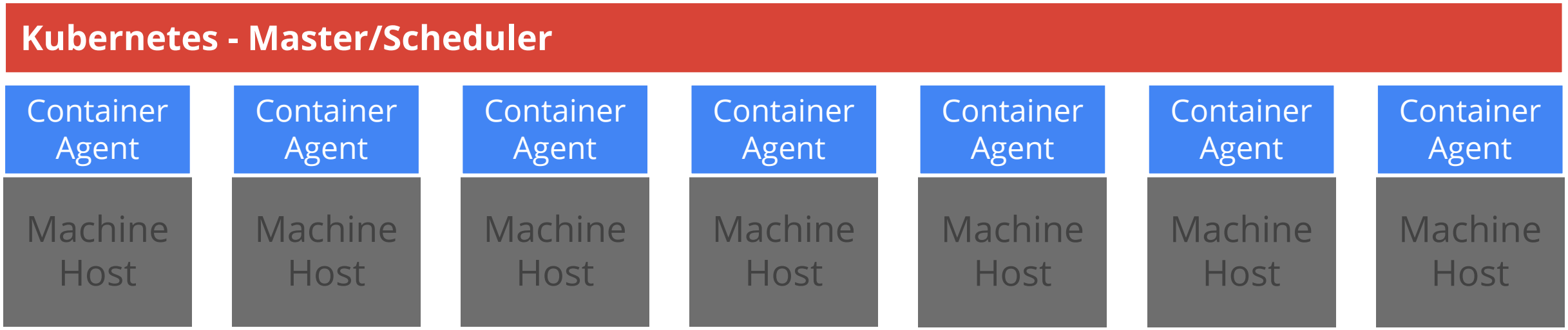
```
replicas: 4
template:
  ...
labels:
  role: frontend
  stage: production
```



Replica Controller

FE

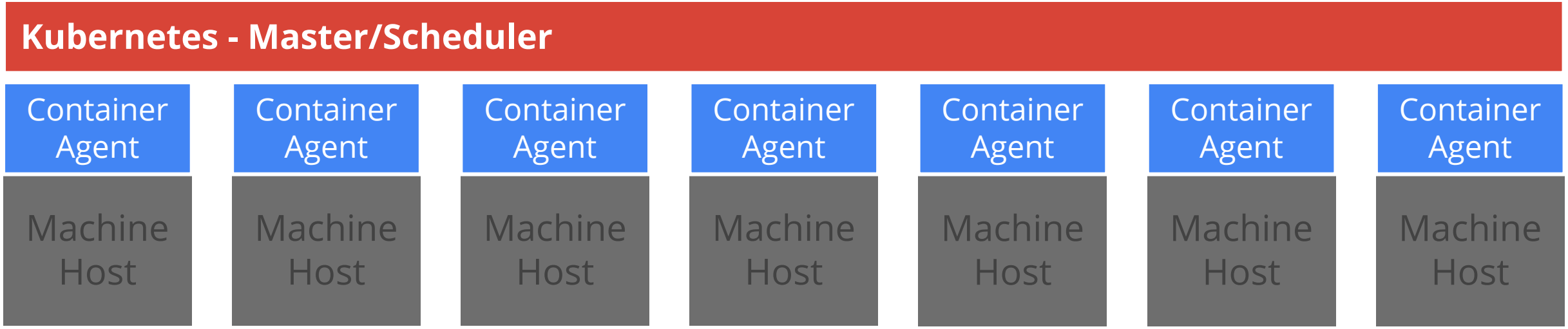
```
replicas: 1
template:
  ...
labels:
  role: frontend
  stage: production
```



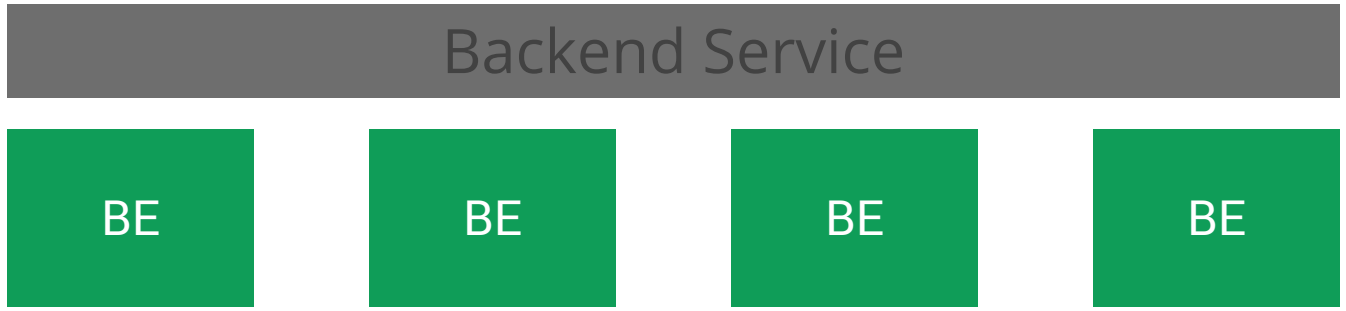
Replica Controller

FE FE FE

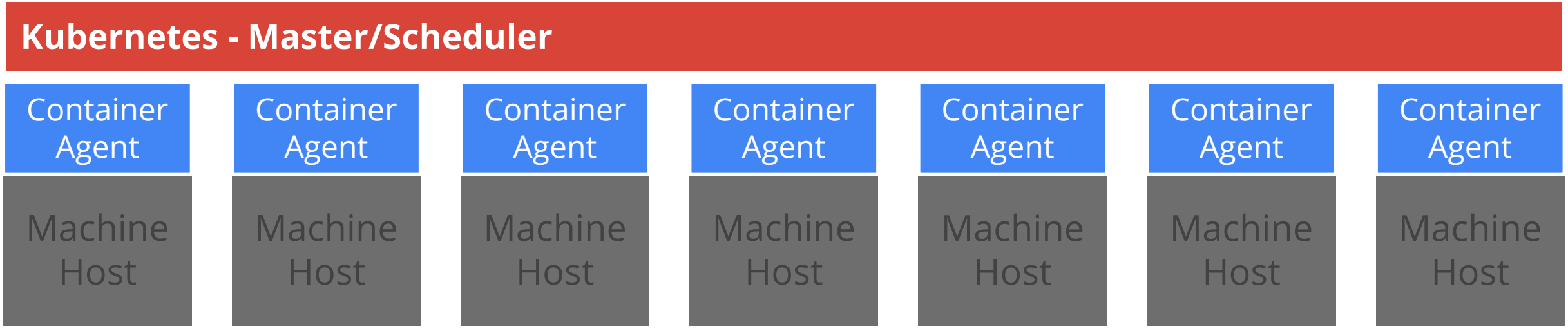
```
replicas: 3
template:
  ...
labels:
  role: frontend
  stage: production
```



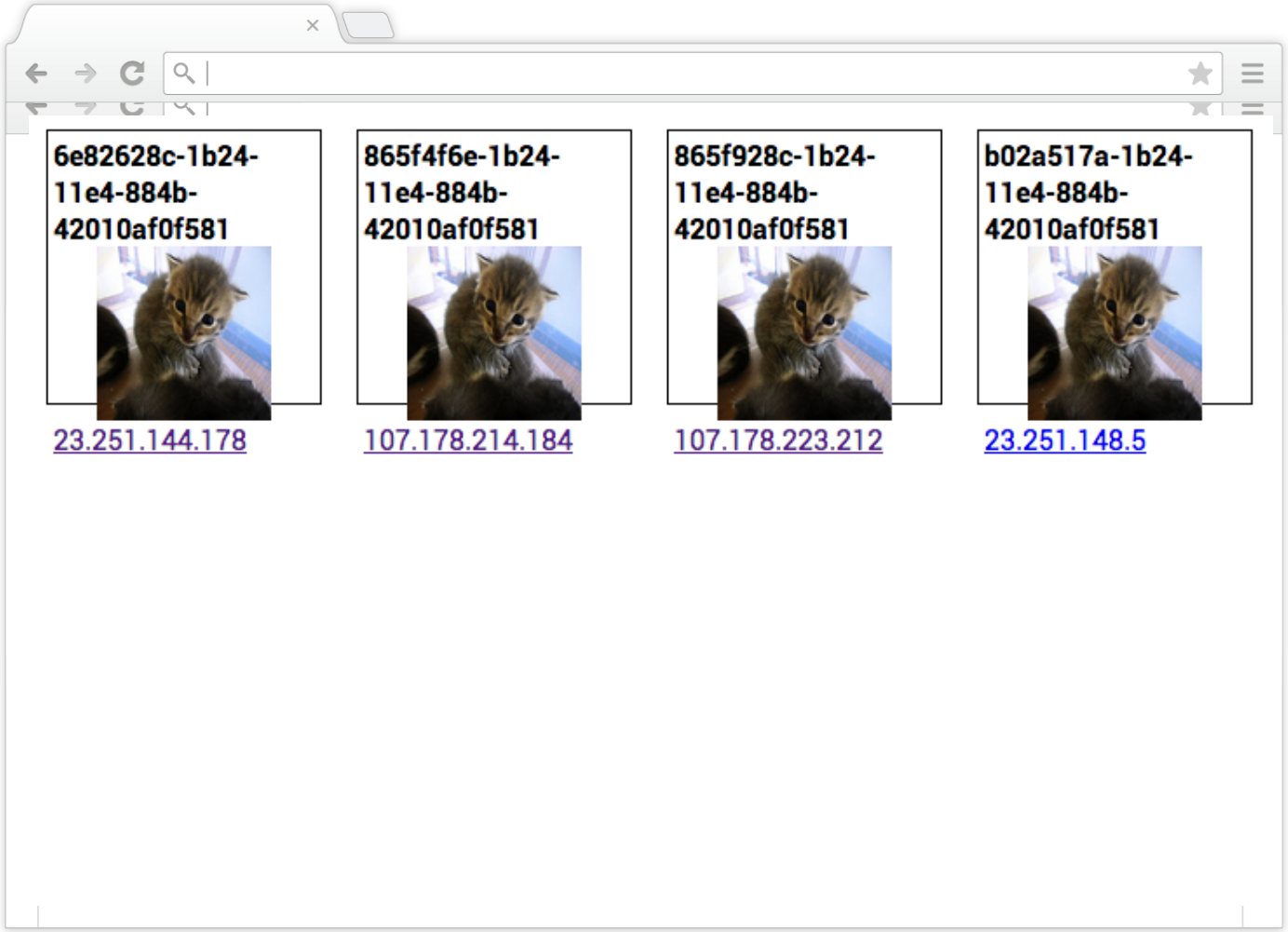
Service



id: backend-service
port: 9000
labels:
  role: backend
  stage: production



Live Demo!


6e82628c-1b24-11e4-884b-42010af0f581
865f4f6e-1b24-11e4-884b-42010af0f581
865f928c-1b24-11e4-884b-42010af0f581
b02a517a-1b24-11e4-884b-42010af0f581
23.251.144.178
107.178.214.184
107.178.223.212
23.251.148.5

We’re just getting started

- Clone Kubernetes at: https://github.com/GoogleCloudPlatform/kubernetes
- Check out container VMs at: https://developers.google.com/compute/docs/containers
- Join the discussion on freenode: http://webchat.freenode.net/?channels=google-containers

Summary

1. We're taking lessons we've learned and open sourcing them
2. Kubernetes is our evolving effort to make cluster management easy
3. We're eager to hear from you!

End