CONTAINERS AND KUBERNETES

CONTAINERS

DOCKER

KUBERNETES

SERVICES, LABELS, SELECTORS

VOLUMES

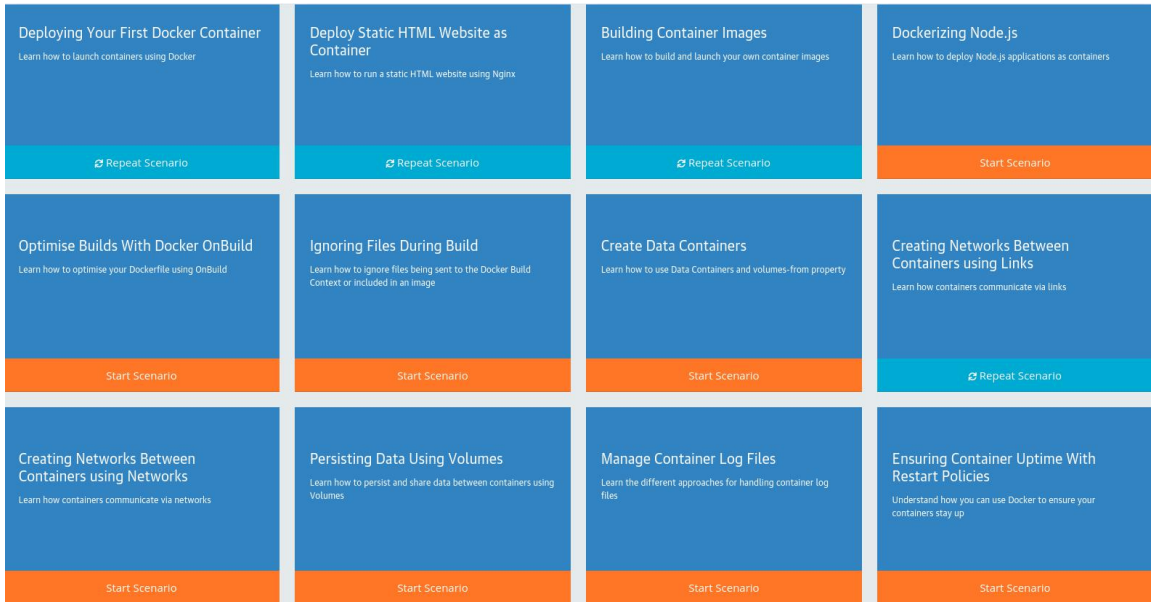
KUBERNETES ON GCP - GKE

KUBERNETES ON AWS

KUBERNETES ON AZURE

DOCKER KNOWLEDGE

Supplementary materials on Katacoda



CONTAINERS

CONTAINERS
DOCKER
KUBERNETES
SERVICES, LABELS, SELECTORS
VOLUMES
KUBERNETES ON GCP - GKE
KUBERNETES ON AWS
KUBERNETES ON AZURE

PRE-REQUISITES

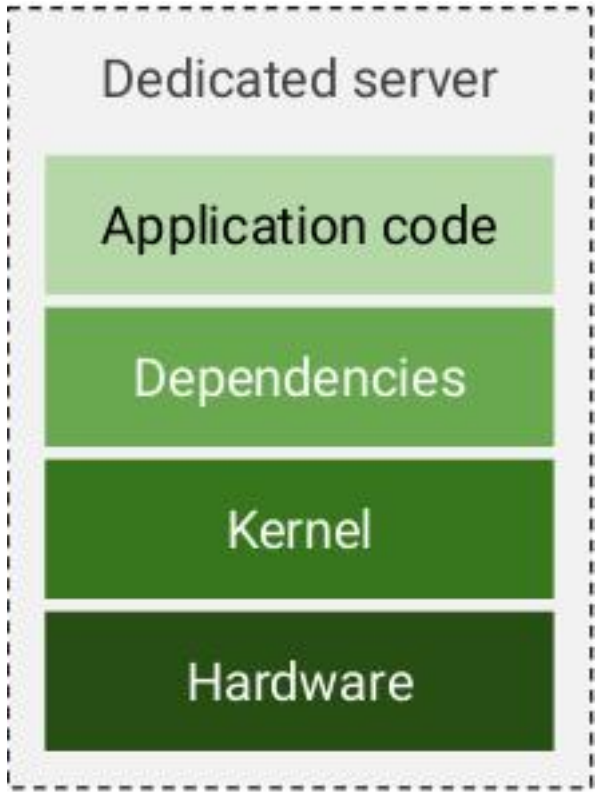
In this section, we assume that you know enough about Docker and Kubernetes

And, we will cover running these in the cloud

However,

- To be on the same page,
- let's dedicate a few slides to a quick overview

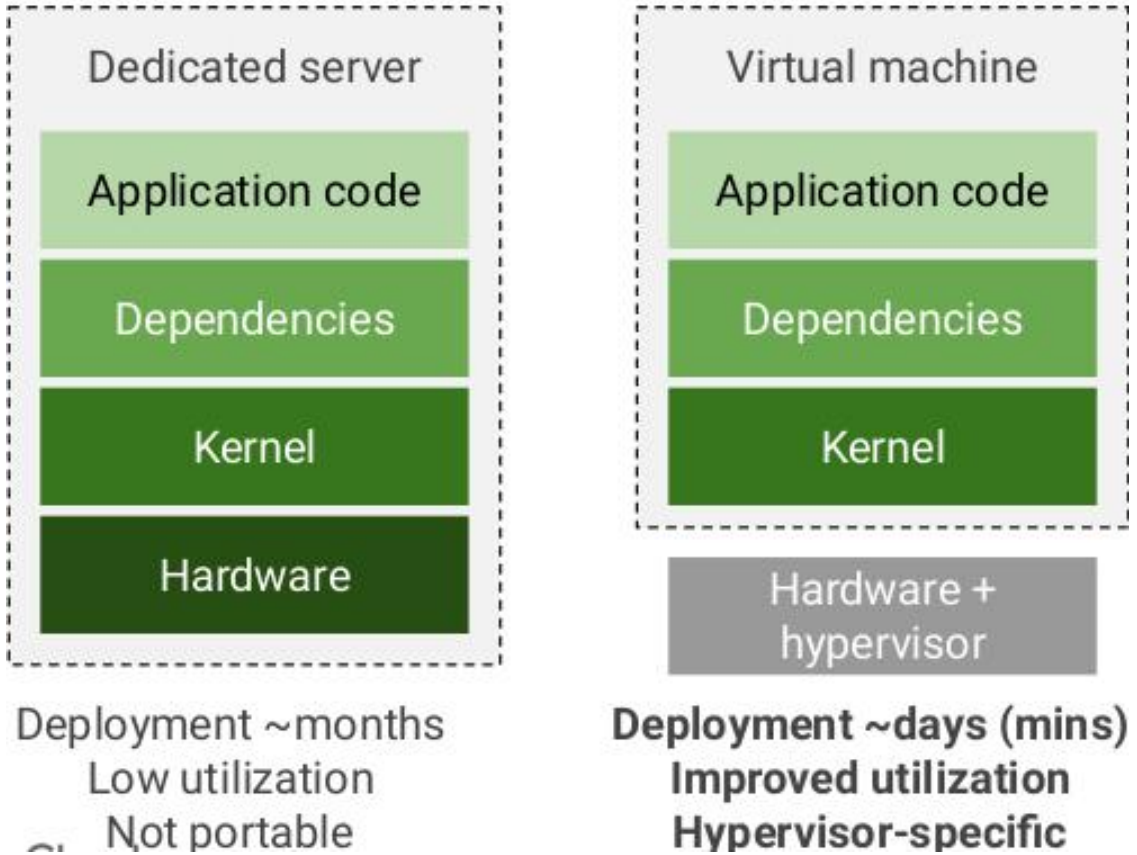
BUILDING FROM SCRATCH



Deployment ~months
Low utilization
Not portable

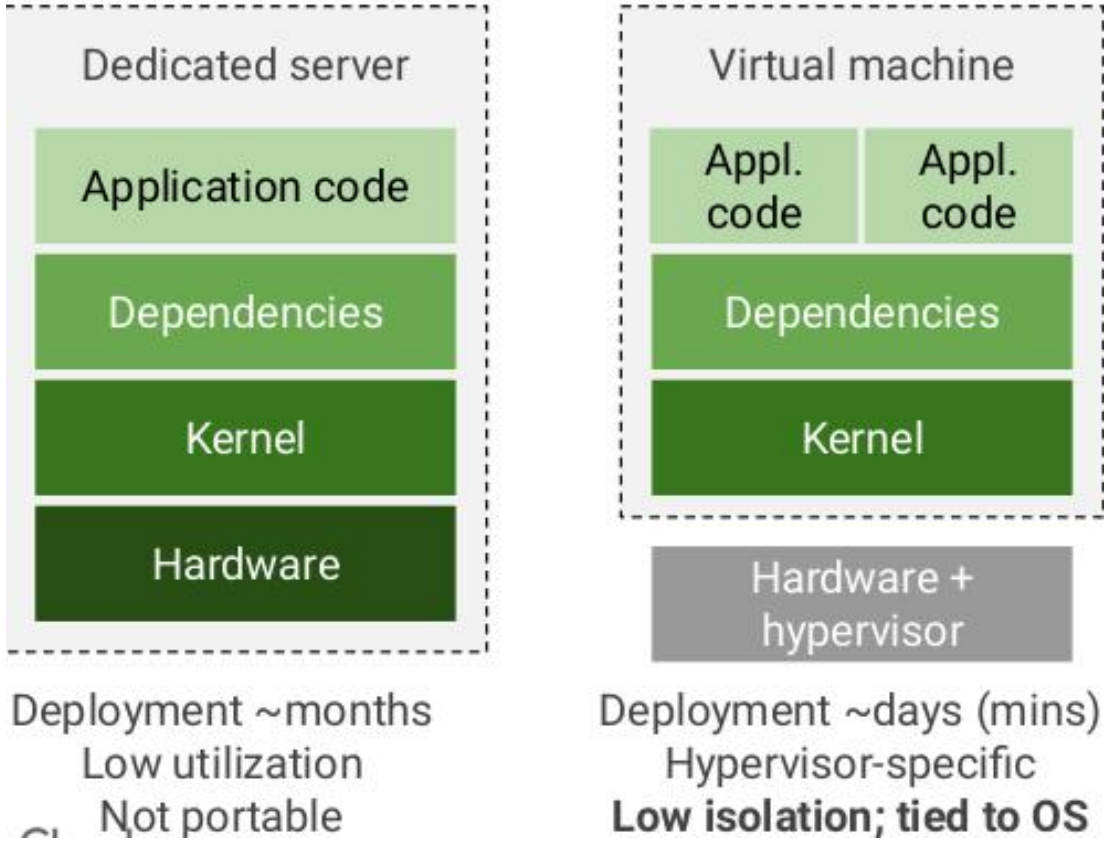
VM STEP

Then VMware popularized running multiple servers and operating systems on the same hardware



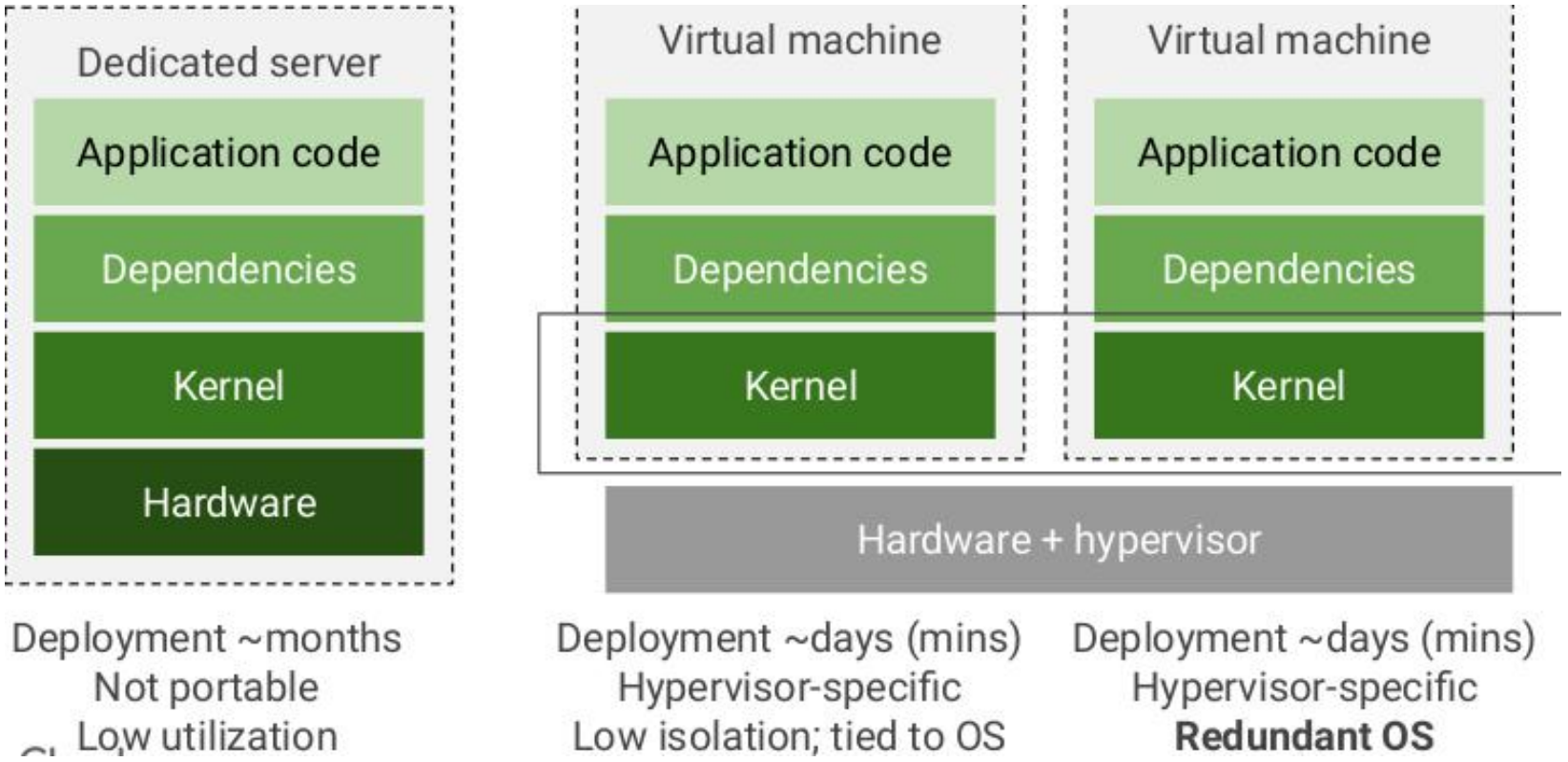
STILL HARD

it was difficult to run and maintain multiple applications on a single VM, even with policies

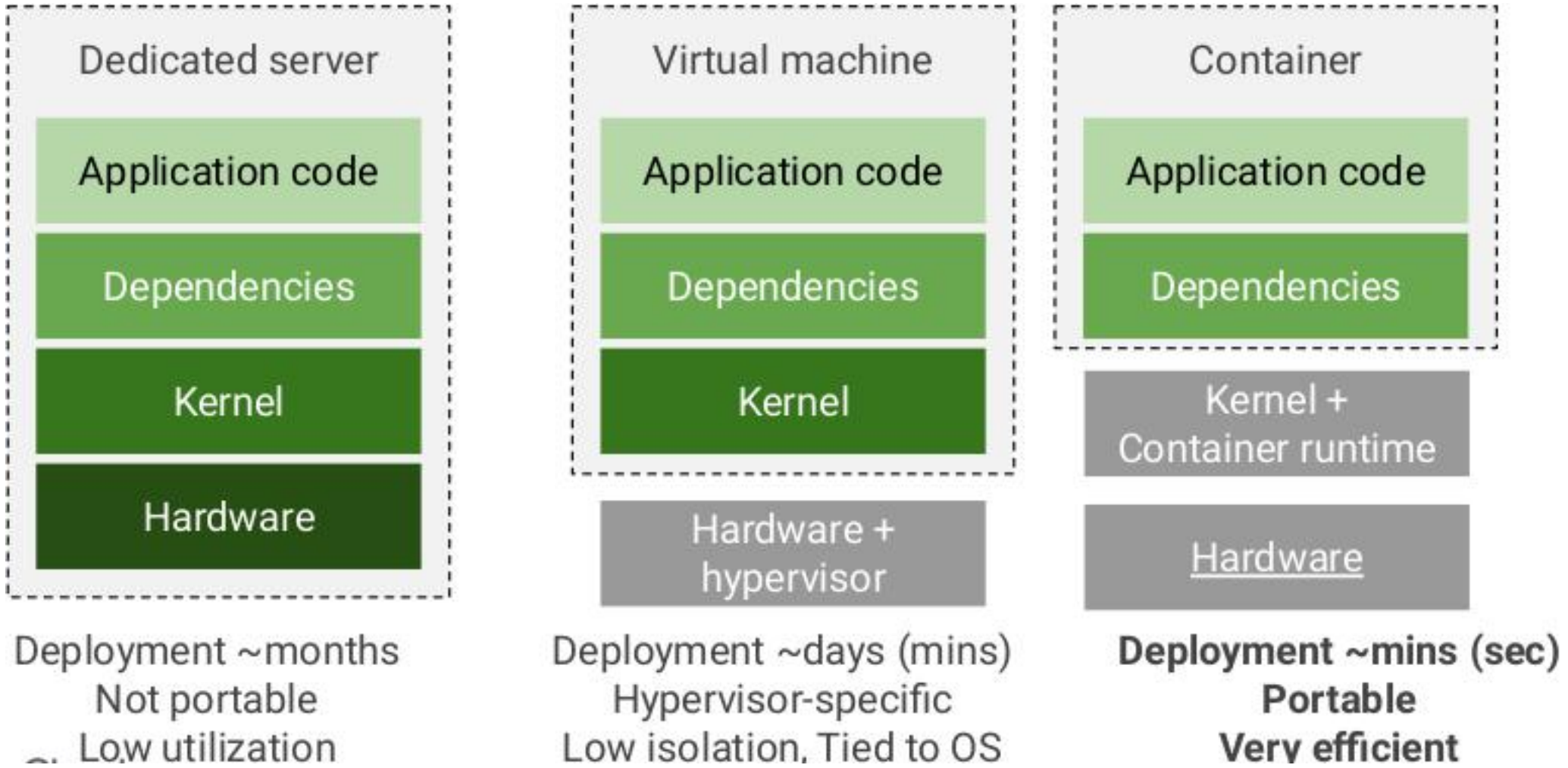


STILL WASTEFUL

The VM-centric way to solve this is to run each app on its own server with its own dependencies, but that's wasteful



VIRTUALIZE THE OS!



Deployment ~months
Not portable
Low utilization

Deployment ~days (mins)
Hypervisor-specific
Low isolation, Tied to OS

Deployment ~mins (sec)
Portable
Very efficient

WHAT'S GOOD ABOUT CONTAINERS

Code works the same everywhere:

- Across dev, test, & production
- Across bare-metal, VMs, cloud

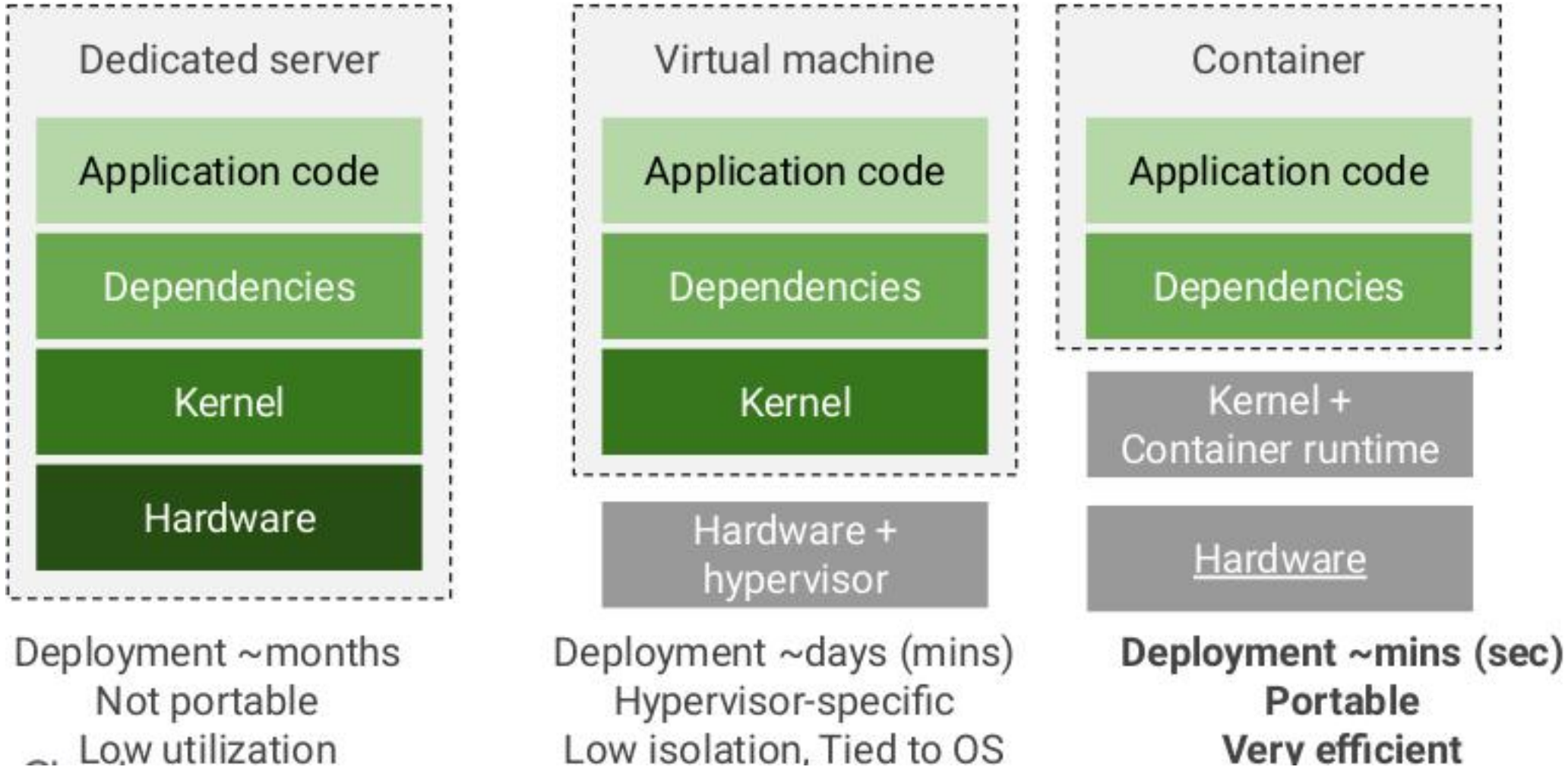
Packaged apps speed development:

- Agile creation and deployment
- Continuous integration/delivery
- Single file copy

They provide a path to microservices:

- Introspectable
- isolated
- elastic

VIRTUALIZE THE OS!



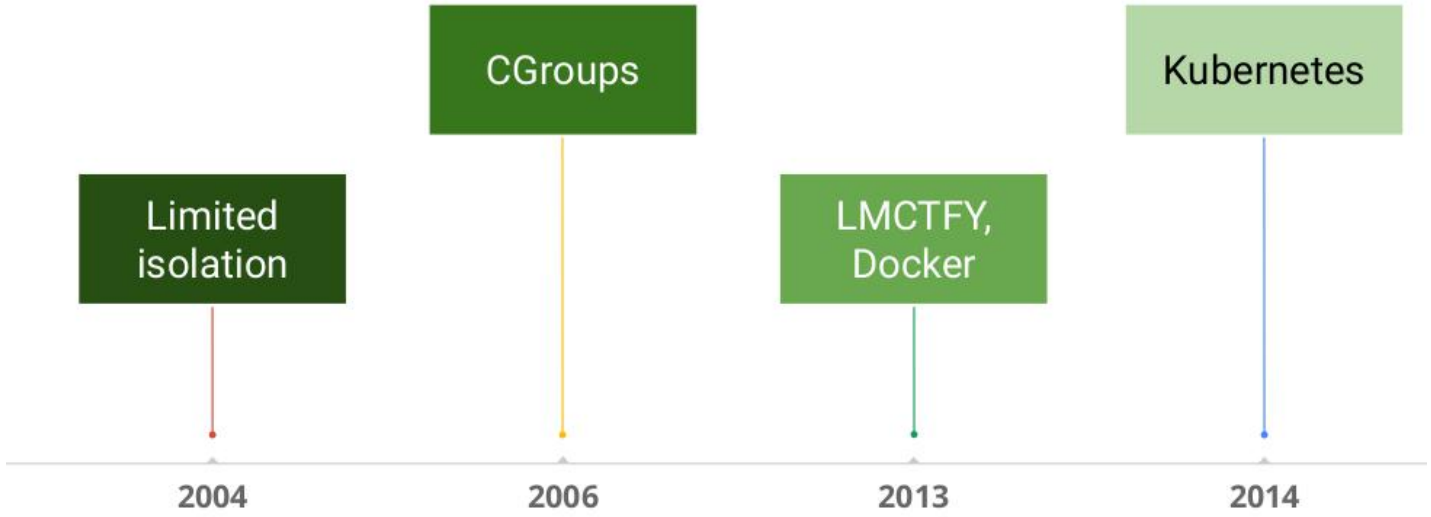
Deployment ~months
Not portable
Low utilization

Deployment ~days (mins)
Hypervisor-specific
Low isolation, Tied to OS

Deployment ~mins (sec)
Portable
Very efficient

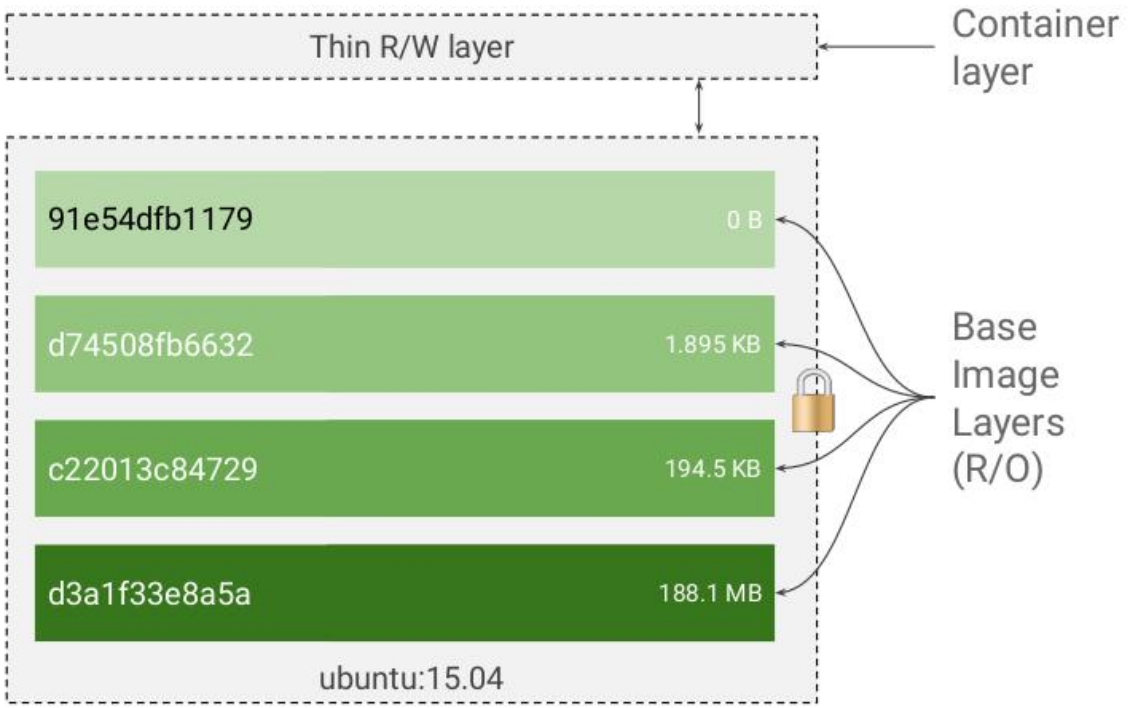
CONTAINER HISTORY

Google has been developing and using containers to manage its applications for 12 years



CONTAINERS' LAYERED FILE SYSTEM

Containers use a layered file system with only the top layer writable

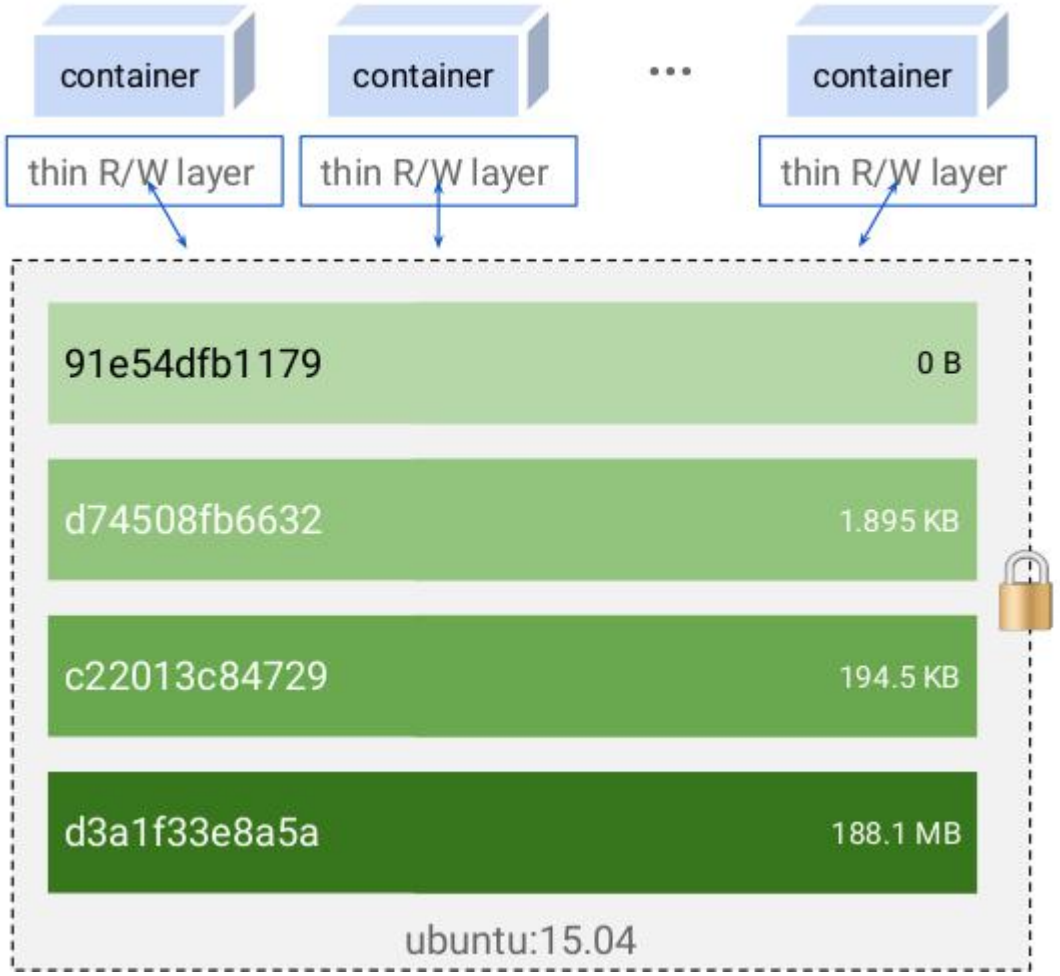


Dockerfile

```
FROM ubuntu:15.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

SMALLER SHARED IMAGES

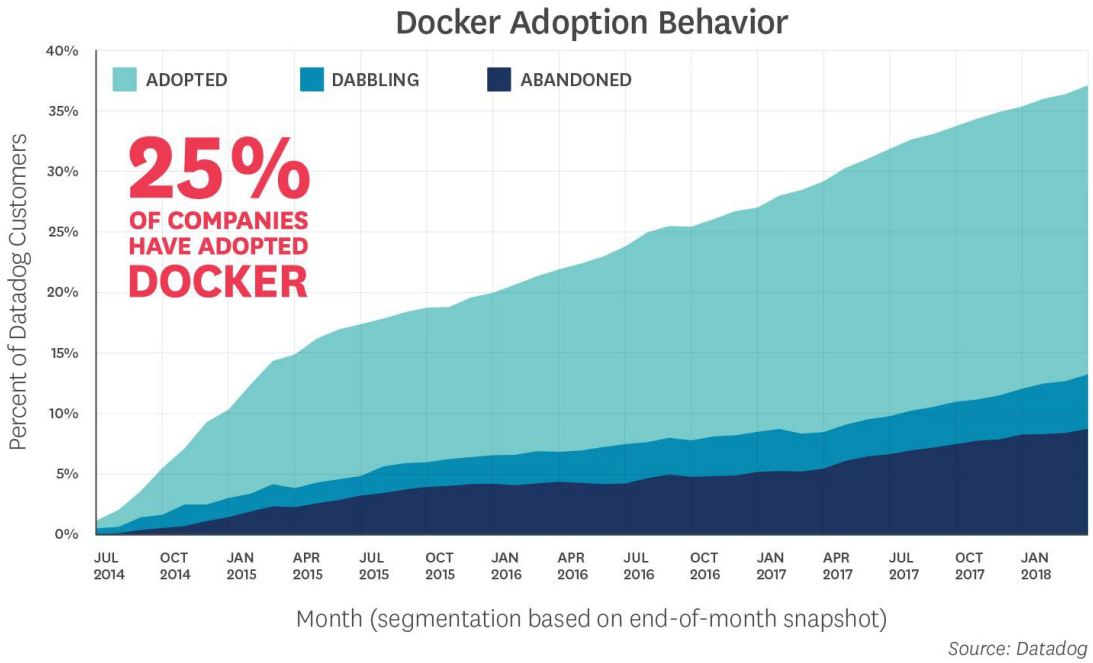
Containers promote smaller shared images



DOCKER

CONTAINERS
DOCKER
KUBERNETES
SERVICES, LABELS, SELECTORS
VOLUMES
KUBERNETES ON GCP - GKE
KUBERNETES ON AWS
KUBERNETES ON AZURE

DOCKER ADOPTION



Source: Datadog

Source: DataDog

DOCKERIZING EXAMPLE - CODE

This application listens on port `8080` and reply

`python web-server.py`

```
1  import tornado.ioloop
2  import tornado.web
3  import socket
4  dependencies
5  class MainHandler(tornado.web.RequestHandler):
6  def get(self):
7  self.write("Hostname: " +
8  socket.gethostname())
9  def make_app():
10 return tornado.web.Application([
11 (r"/", MainHandler),
12 ])
13 if __name__ == "__main__":
14 app = make_app()
15 listening on a port
16 app.listen(8080)
17 tornado.ioloop.IOLoop.current().start()
```

CONTAINERIZE IT WITH DOCKER

```
$> docker build -t
py-web-server .
$> docker run -d py-web-server
```

```
FROM library/python:3.6.0-alpine
RUN pip install tornado
ADD web-server.py /web-server.py
CMD ["python", "/web-server.py"]
```

```
You can also do stuff like:
$> docker images
$> docker ps
$> docker logs <container id>
$> docker stop py-web-server
```

USE GOOGLE DOCKER REGISTRY

```
docker build -t gcr.io/$PROJECT_ID/py-web-server:v1 .

gcloud docker -- push gcr.io/$PROJECT_ID/py-web-server:v1

docker run -d -p 8080:8080 --name py-web-server  \
gcr.io/$PROJECT_ID/py-web-server:v1
```

build a container image

push it to a registry

run it

CONTAINERS ON GCP

App Engine supports Docker containers as a custom runtime

Google Container Registry: private container image hosting on GCS with various CI/CD integrations

Compute Engine supports containers, including managed instance groups with Docker containers

The most powerful choice is a container orchestrator

QUIZ

Docker Desktop is an app for building and sharing containerized apps and microservices available on which of the following operating systems?

- A. macOS only
- B. Linux only
- C. Windows, macOS, and Windows Subsystem for Linux (WSL)

QUIZ

Which is correct Docker command to rebuild a container image?

- A. `docker rebuild`
- B. `docker compile`
- C. `docker build`

QUIZ

Which of the following sentences describe a container image the best?

- A. A container image is a read-only portable package that contains software and may include an operating system.
- B. A container image is a set of commands that builds a container.
- C. A container image is a read-only portable package that contains software.

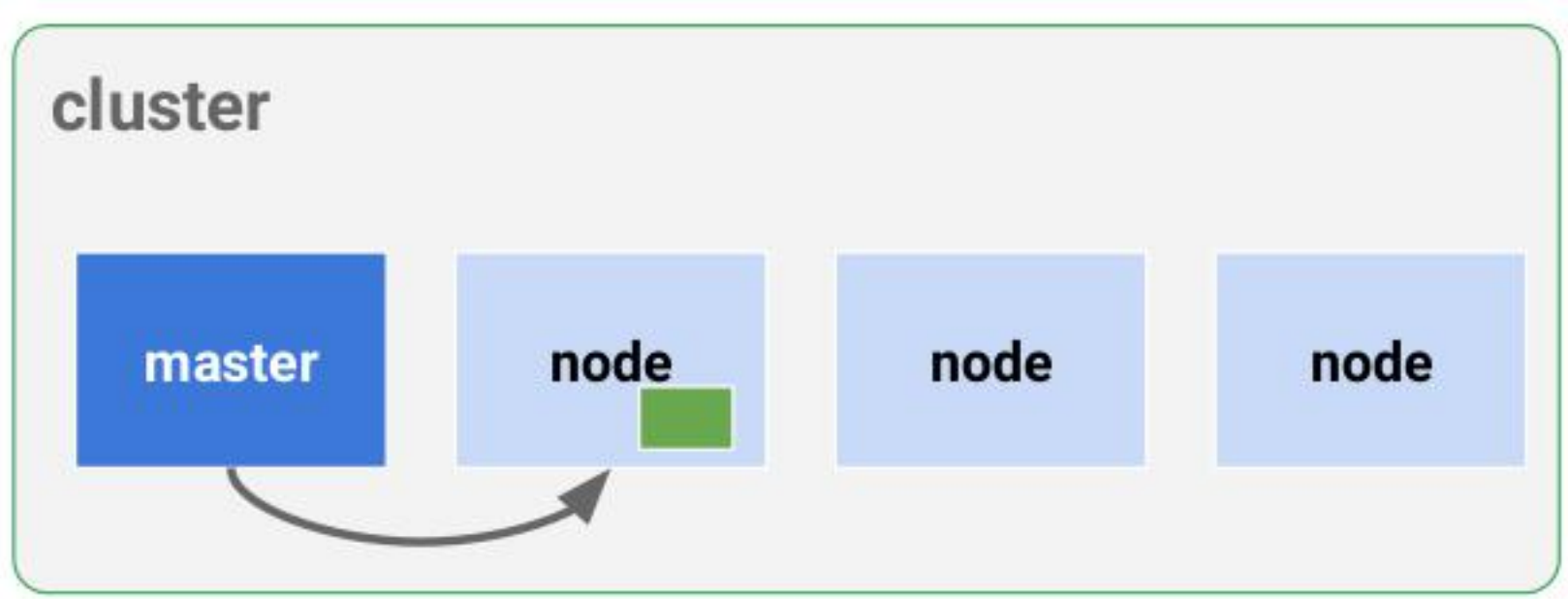
KUBERNETES

CONTAINERS
DOCKER
KUBERNETES
SERVICES, LABELS, SELECTORS
VOLUMES
KUBERNETES ON GCP - GKE
KUBERNETES ON AWS
KUBERNETES ON AZURE

WHAT IS KUBERNETES (K8)

Kubernetes manages jobs

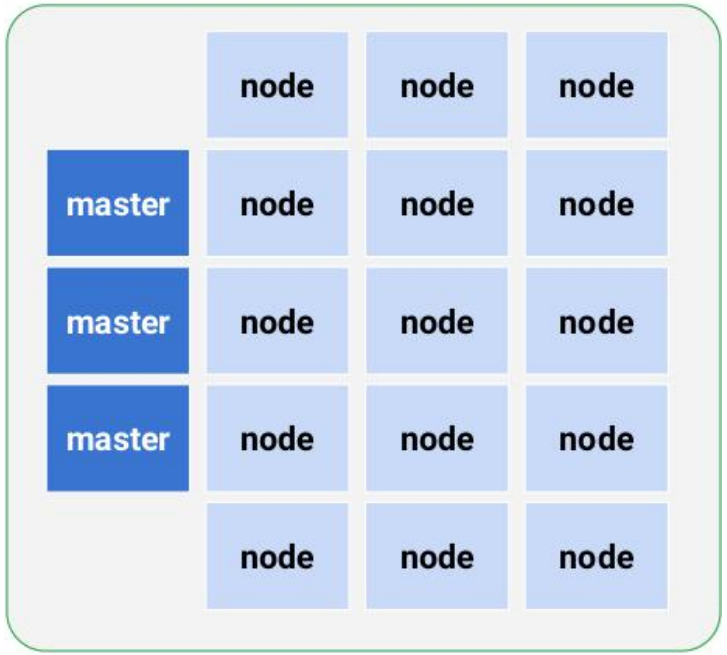
Jobs run containers on nodes



THE REAL WORLD

In a real ecosystem, a cluster may have 1000s of nodes and multiple masters.

Regional clusters have masters and nodes spread across 3 zones



PARTICIPANT PRESENTATION

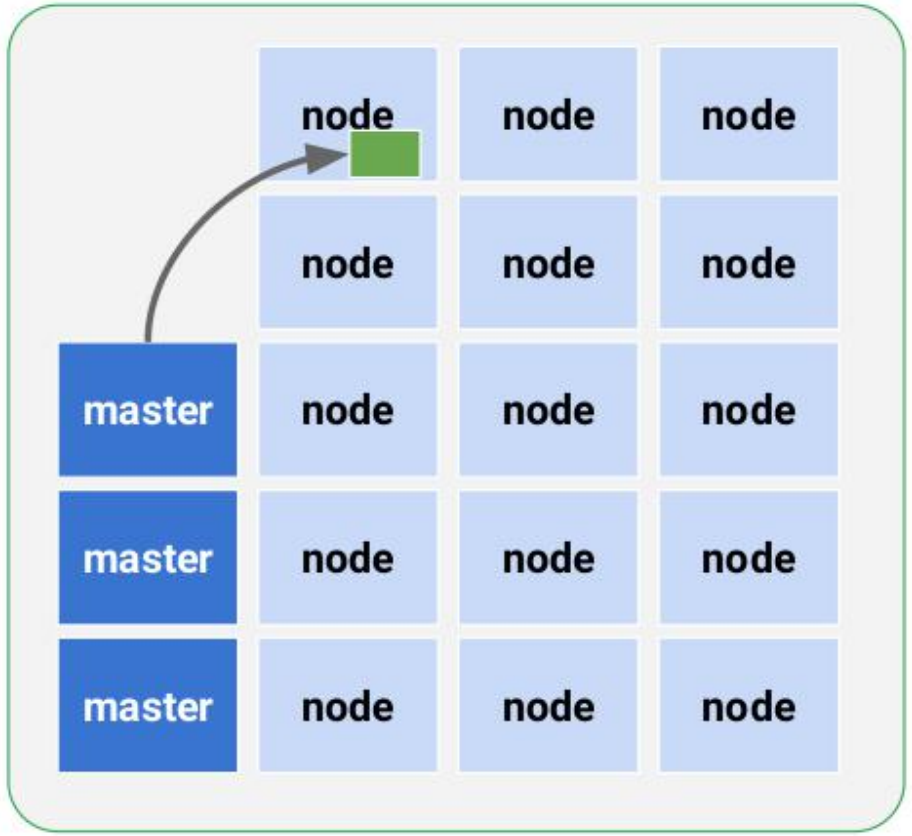
Kubernetes use cases presented by the participants

Karl Kornel

Stanford team

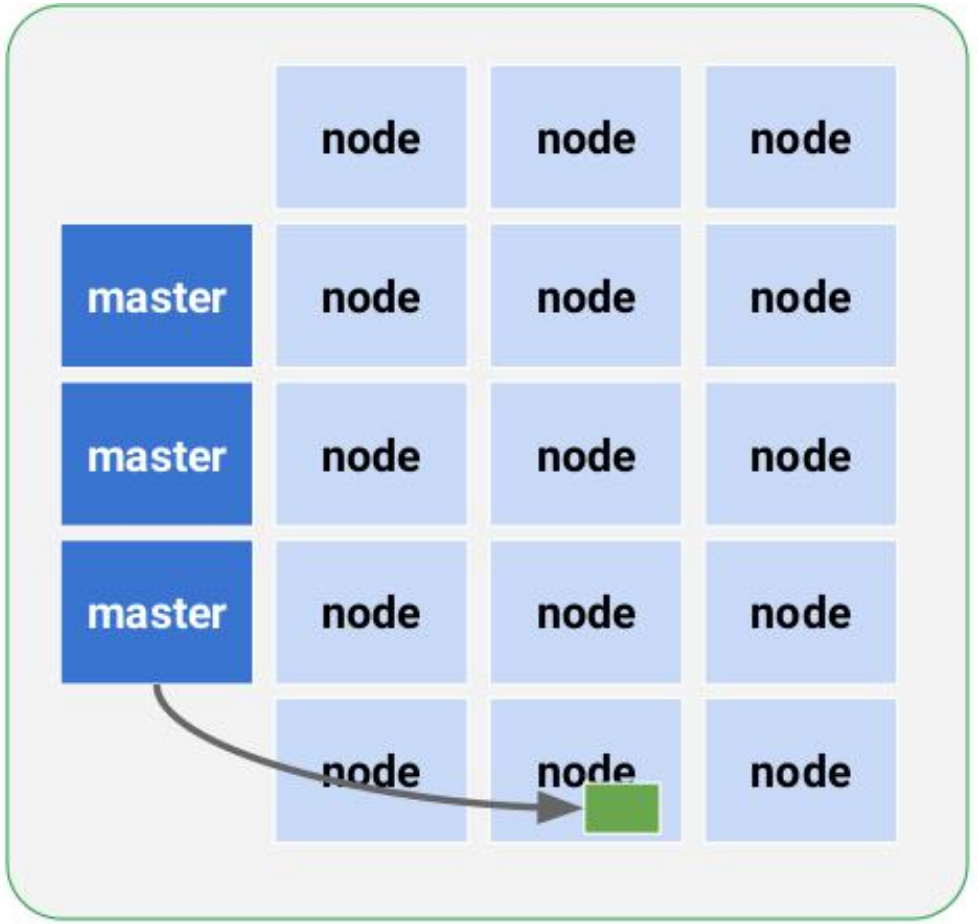
MASTERS SCHEDULE JOBS

Masters schedule jobs on nodes based on load



AND THEY CAN MOVE

And they can move them as needed to match resources



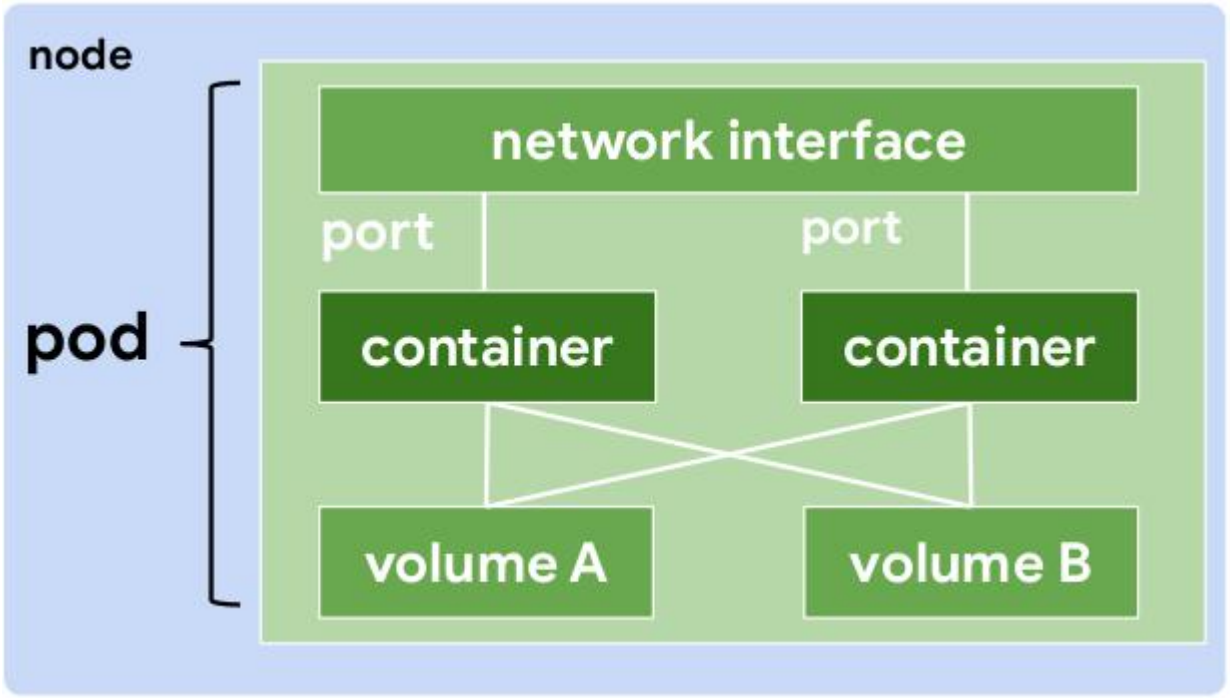
POD

A job is called a Pod.

It's analogous to a VM

It can run multiple containers

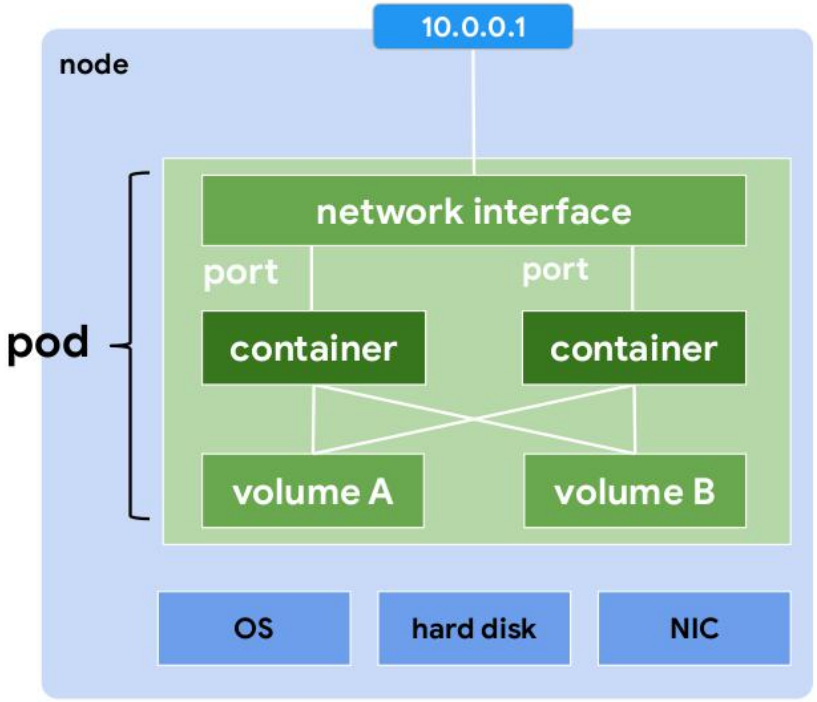
It shares networking and storage separate from the node



UNDER THE POD

Underneath the pod you have

- the node's hardware,
- OS, and
- NIC
- And a node IP

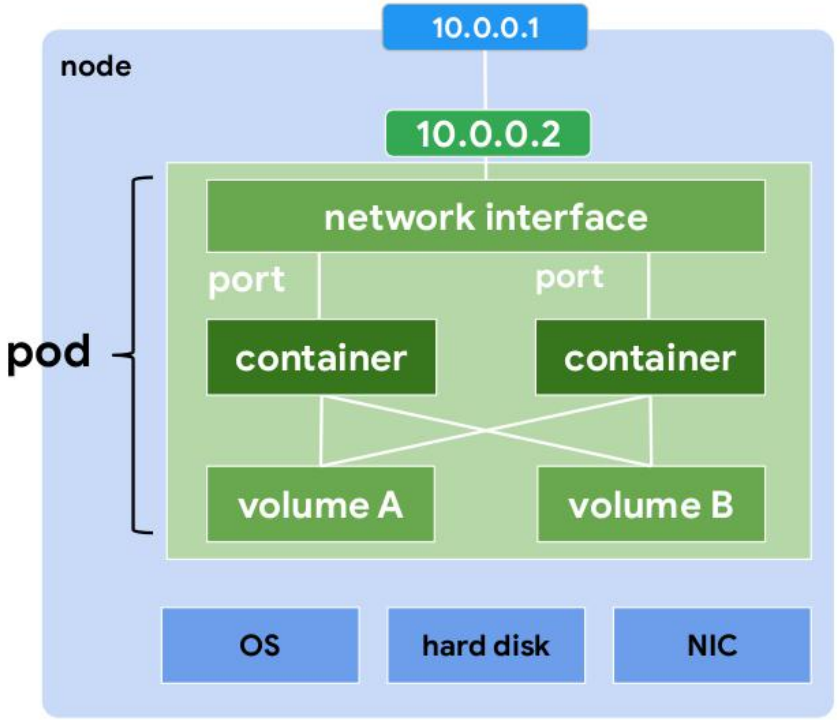


INSIDE THE POD

Inside you have the pod with its containers

It gets its own network namespace with unique IP and set of ports

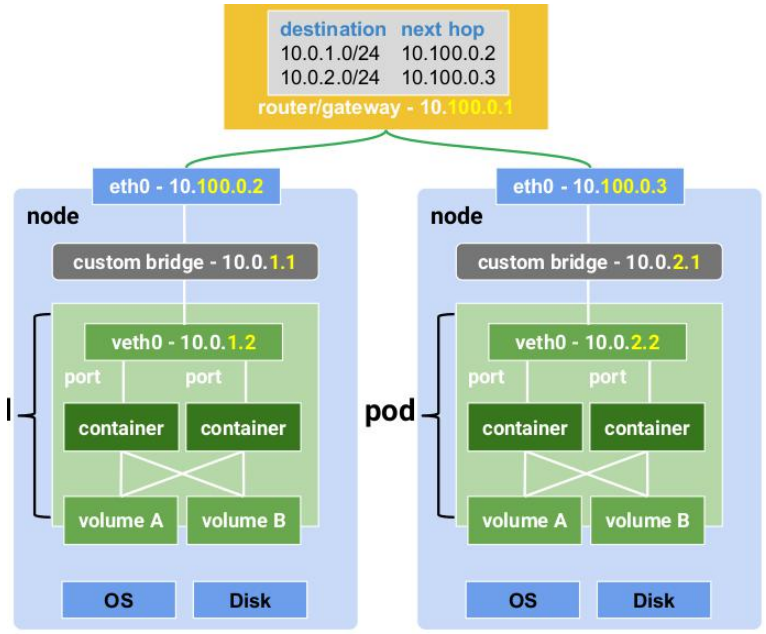
Data is stored in volumes in memory or persistent disks



GKE NETWORKING

Networking between pods in different nodes can be managed in a master routing table or other means.

GKE uses iptables in nodes and port forwarding for fewer points of failure (not shown).



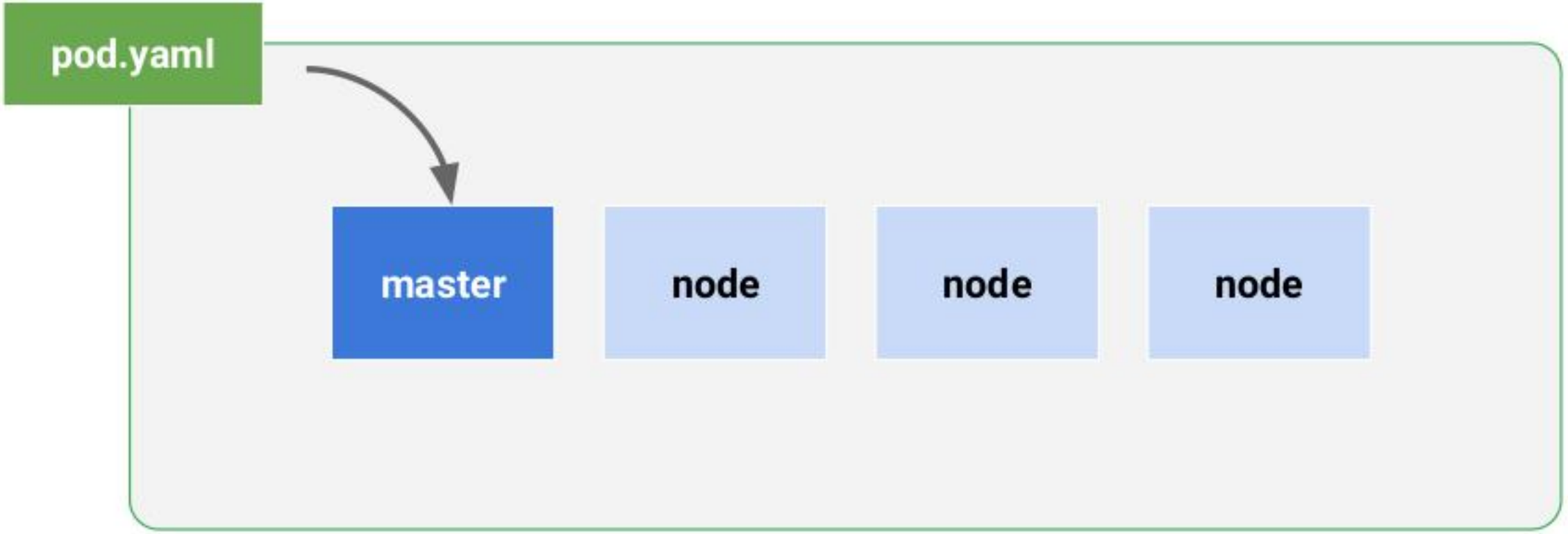
POD IN YAML

You define a pod with a YAML file

```
1  apiVersion: v1
2  kind: Pod
3  metadata:
4  name: my-app
5  spec:
6  containers:
7  - name: my-app
8  image: my-app
9  - name: nginx-ssl
10 image: nginx
11 ports:
12 - containerPort: 80
13 - containerPort: 443
```

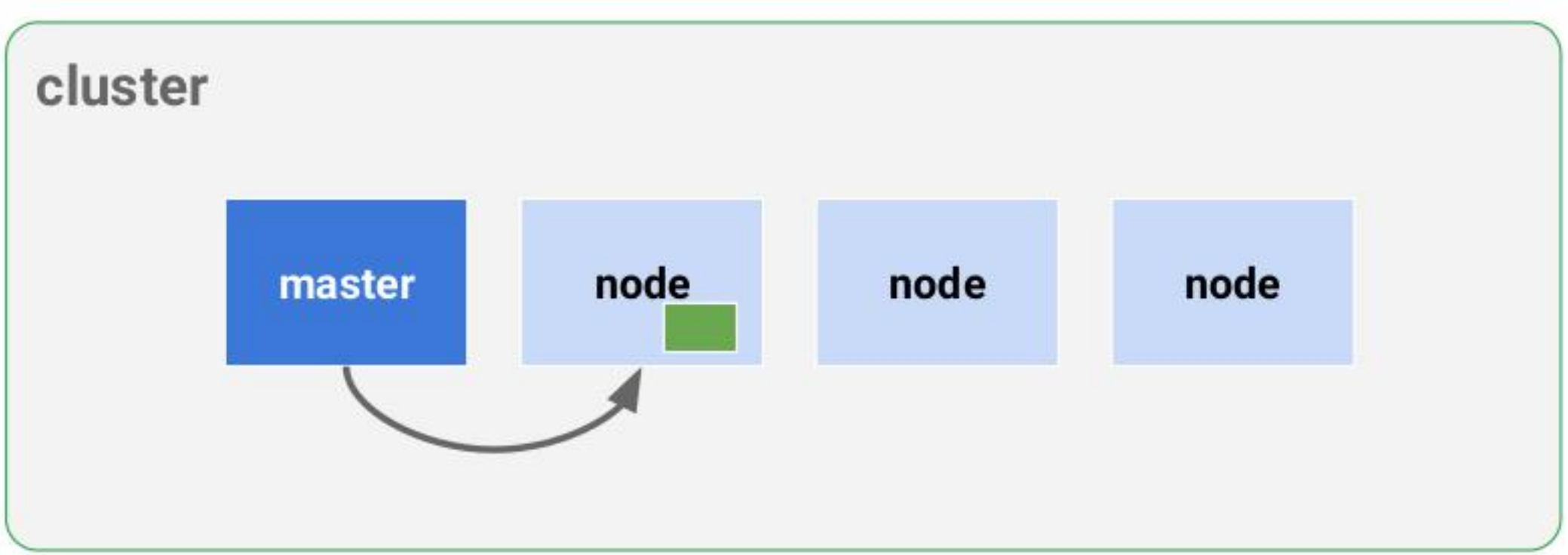
YAML UPLOAD

You upload the YAML file to the master



MASTER CREATES A POD

And the master creates a pod on your set of nodes



POD COMPOSITION

A pod file is composed of several parts; for example...

```
apiVersion: v1
kind: Pod
metadata:
  name: my-app
spec:
  containers:
  - name:  my-app
    image: my-app
  - name:  nginx-ssl
    image: nginx
    ports:
    - containerPort: 80
    - containerPort: 443
```

API version

pod resource

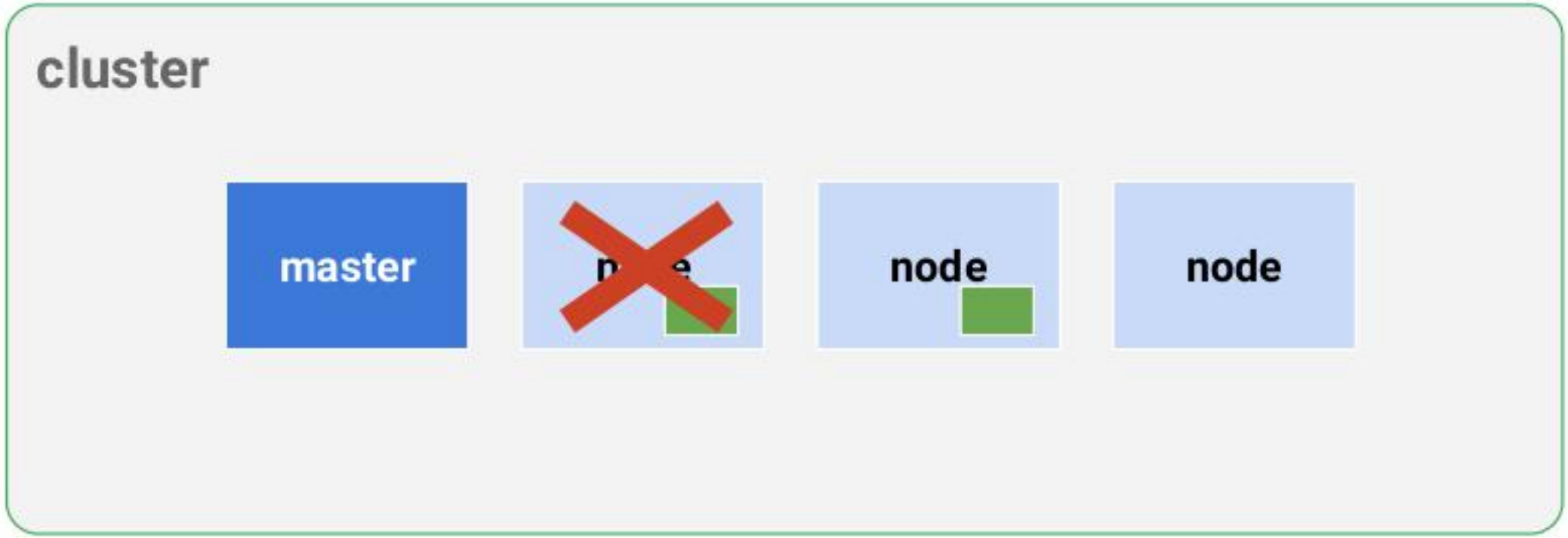
pod name

two containers

NGINX front end on two ports

MULTIPLE PODS

A deployment ensures that N pods are running in a cluster at any given time



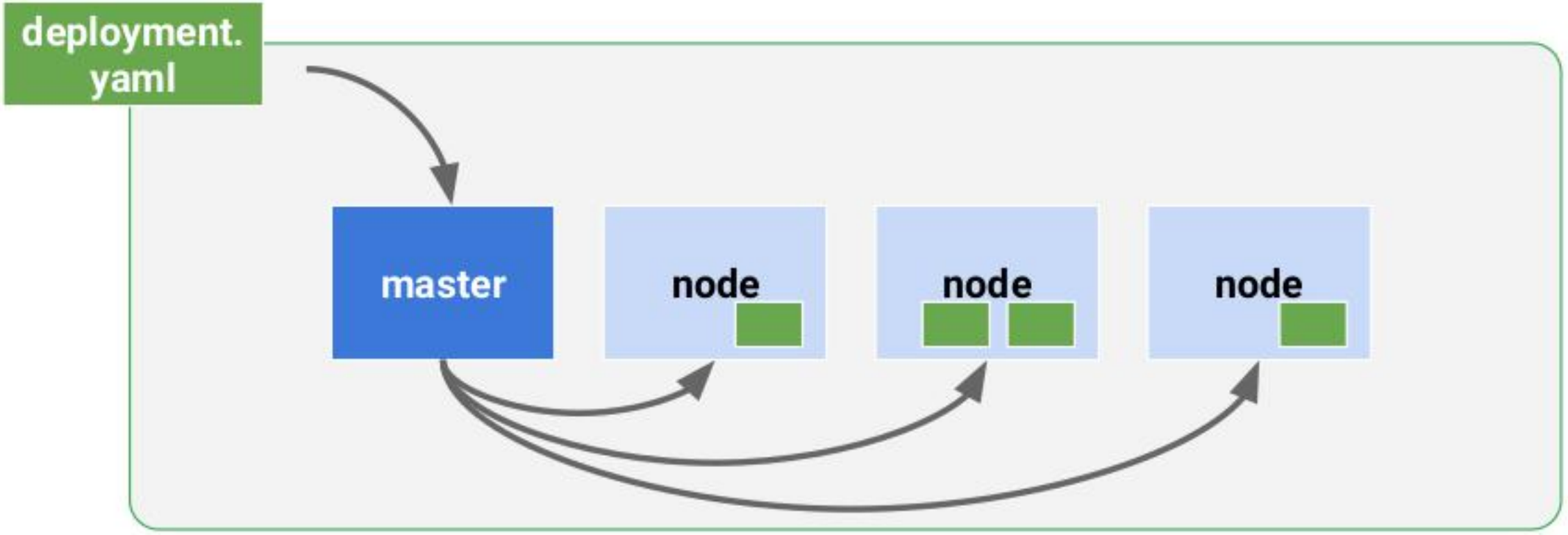
YAML DEPLOYMENT

You define a deployment with a YAML file

```
kind: Deployment ——— deployment resource
apiVersion: v1.1
metadata:
  name: frontend ——— deployment name
spec:
  replicas: 4 ——— replicas
  selector: ——— pod selector
    role: web role=web
  template:
    metadata:
      name: web
      labels: ——— pod label
        role: web role=web
    spec:
      containers: ——— containers
      - name:  my-app
        image: my-app
      - name:  nginx-ssl
        image: nginx
        ports:
        - containerPort: 80
        - containerPort: 443
```

RUNNING THE PODS

You upload the YAML file to the master, and the scheduler decides where to run the pods



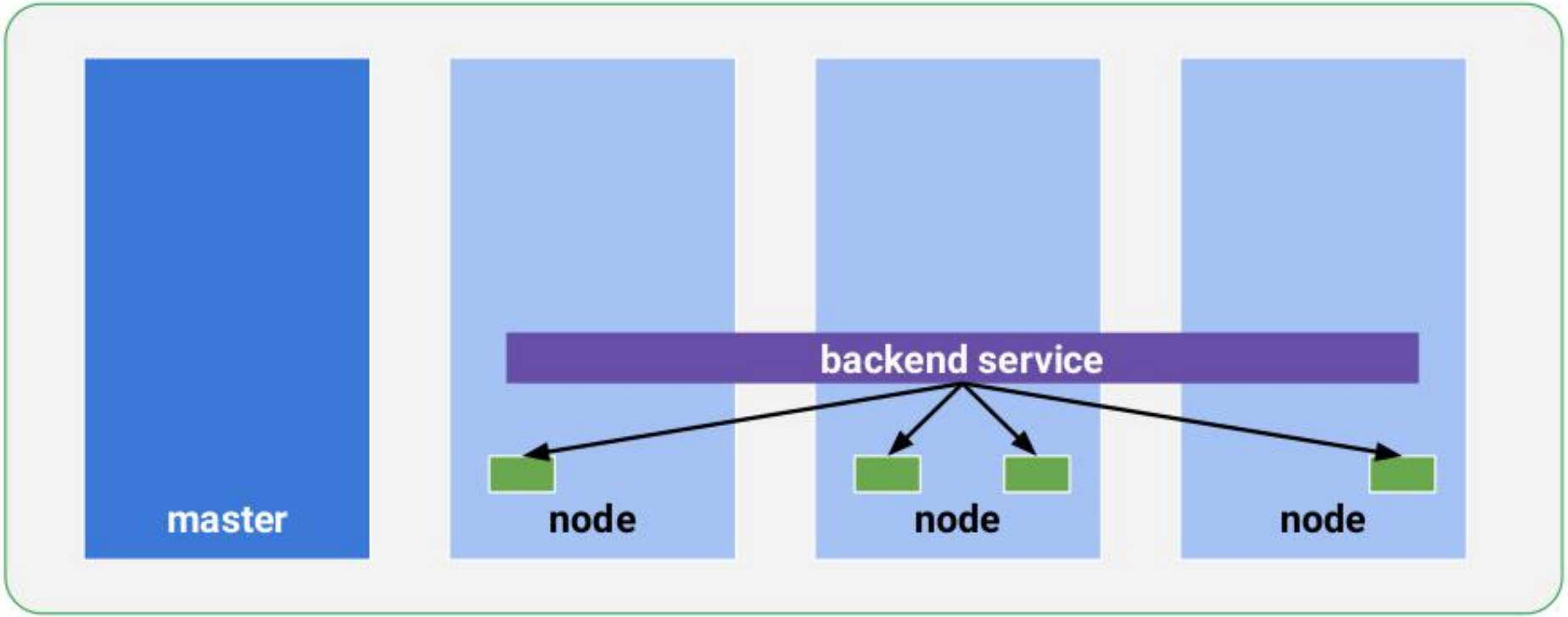
SERVICES, LABELS, SELECTORS

CONTAINERS
DOCKER
KUBERNETES
SERVICES, LABELS, SELECTORS
VOLUMES
KUBERNETES ON GCP - GKE
KUBERNETES ON AWS
KUBERNETES ON AZURE

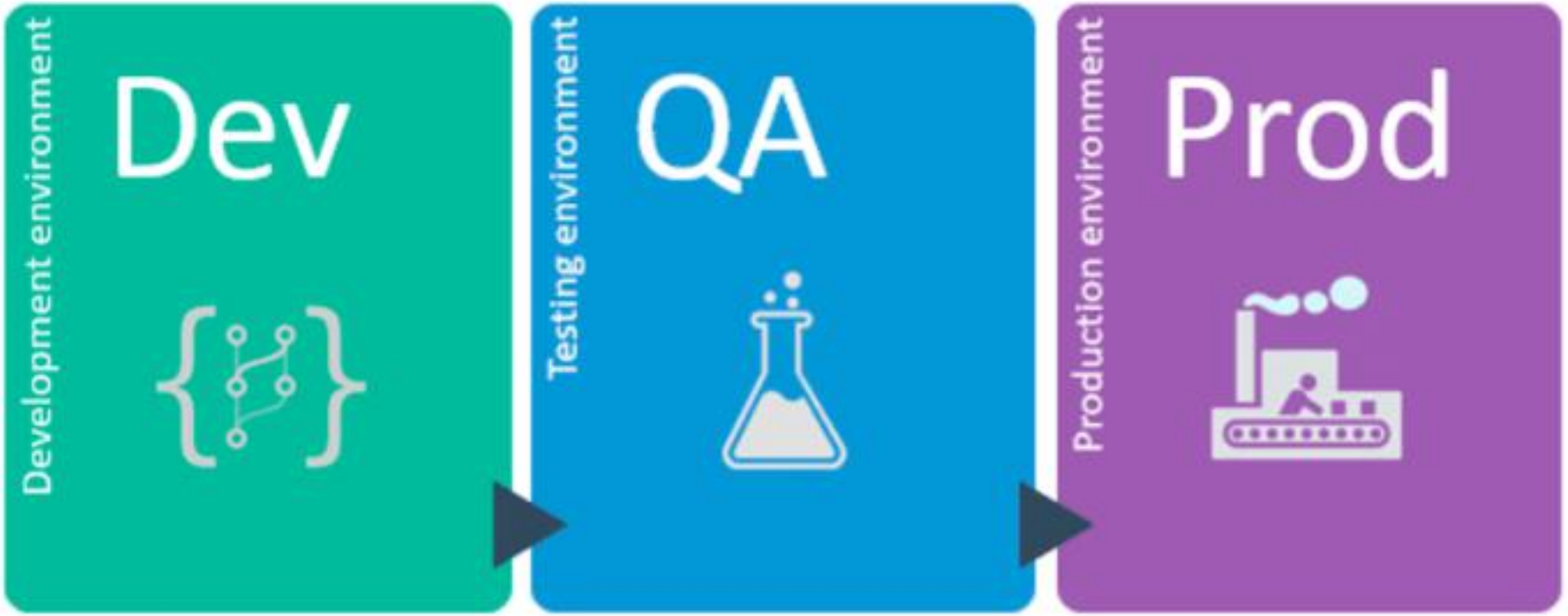
HOW PODS TALK TO EACH OTHER

SERVICE

A service assigns a fixed IP to your pod replicas and allows other pods or services to communicate with them



ENVIRONMENTS

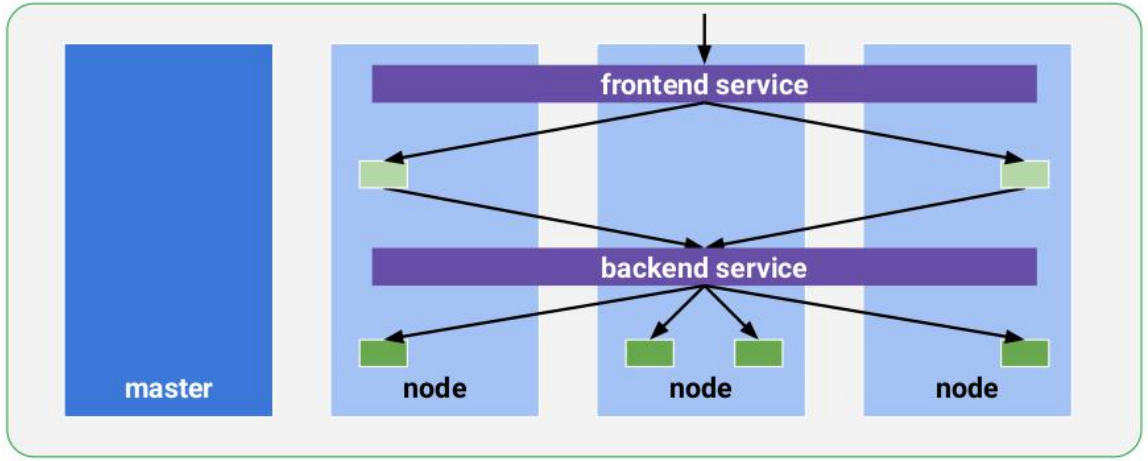


ENVIRONMENTS

Environment/Tier Name	Description
Environment/Tier Name	Description
Local	Developer's desktop/workstation
Development/Trunk	Development server acting as a sandbox where unit testing may be performed by the developer
Integration	CI build target, or for developer testing of side effects
Testing/Test/QC/Internal Acceptance	The environment where interface testing is performed. A quality control team ensures that the new code will not have any impact on the existing functionality and tests major functionalities of the system after deploying the new code in the test environment.
Staging/Stage/Model/Pre-production/External-Client Acceptance/Demo	Mirror of production environment
Production/Live	Serves end-users/clients

MULTIPLE SERVICES

You can have multiple services with different configurations and features



To participants:

- Can you give application examples that would be relevant to your area of research?

YAML, AGAIN

You define a service with a YAML file

```
kind: Service          ——— resource
apiVersion: v1
metadata:
  name: web-frontend
spec:
  ports:               ——— ports
  - name: http
    port: 80
    targetPort: 80
    protocol: TCP
  selector:            ——— pod selector
    role: web
  type: LoadBalancer   — type is load
                         balancer
```

LABELS ARE METADATA

Labels are metadata you can assign to any API object and represent identity

They are

- The only grouping mechanism for pods
- Search by selectors

EXAMPLE

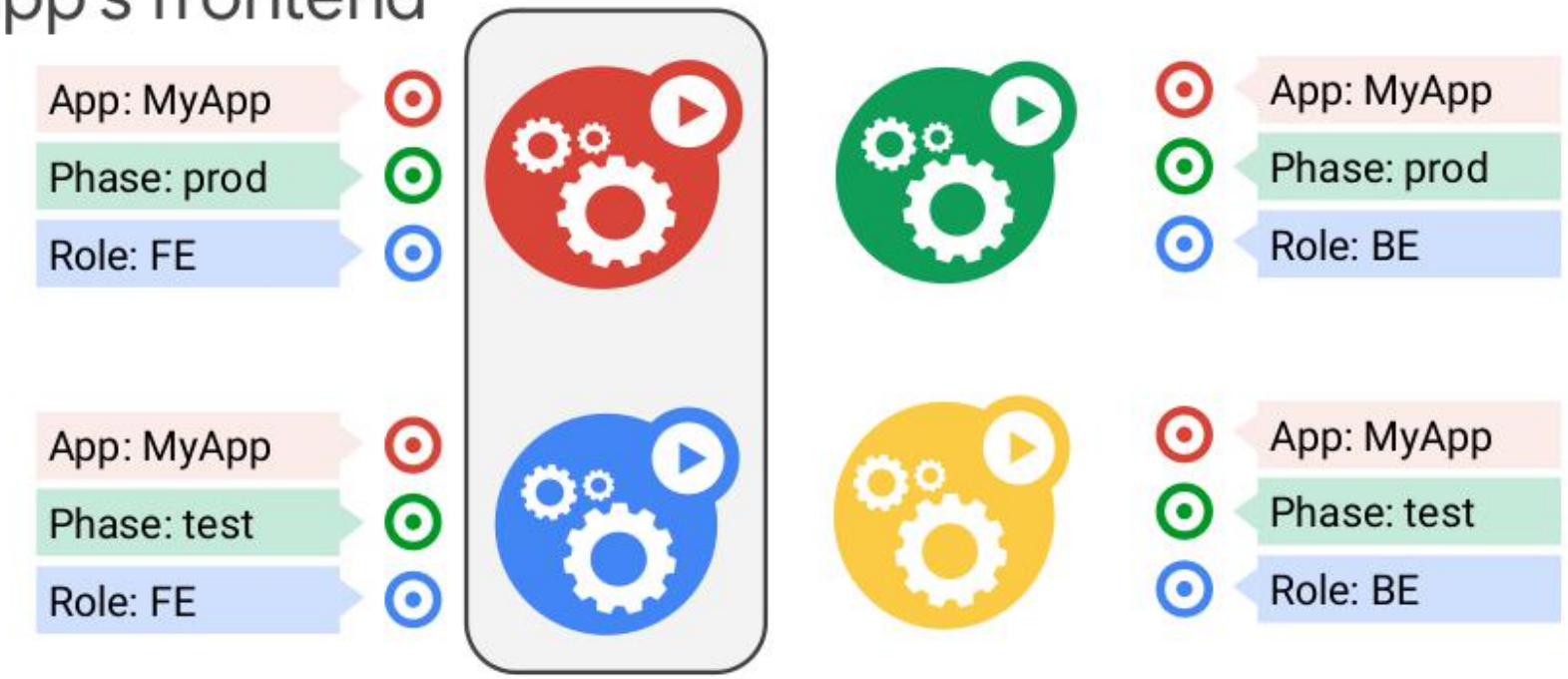
This example has four pods and three labels

QUERY FOR LABELS

You can query for labels that map to a value like the entire app

SEARCH WITH MULTIPLE LABELS

Or narrow your search with multiple labels like your app's frontend



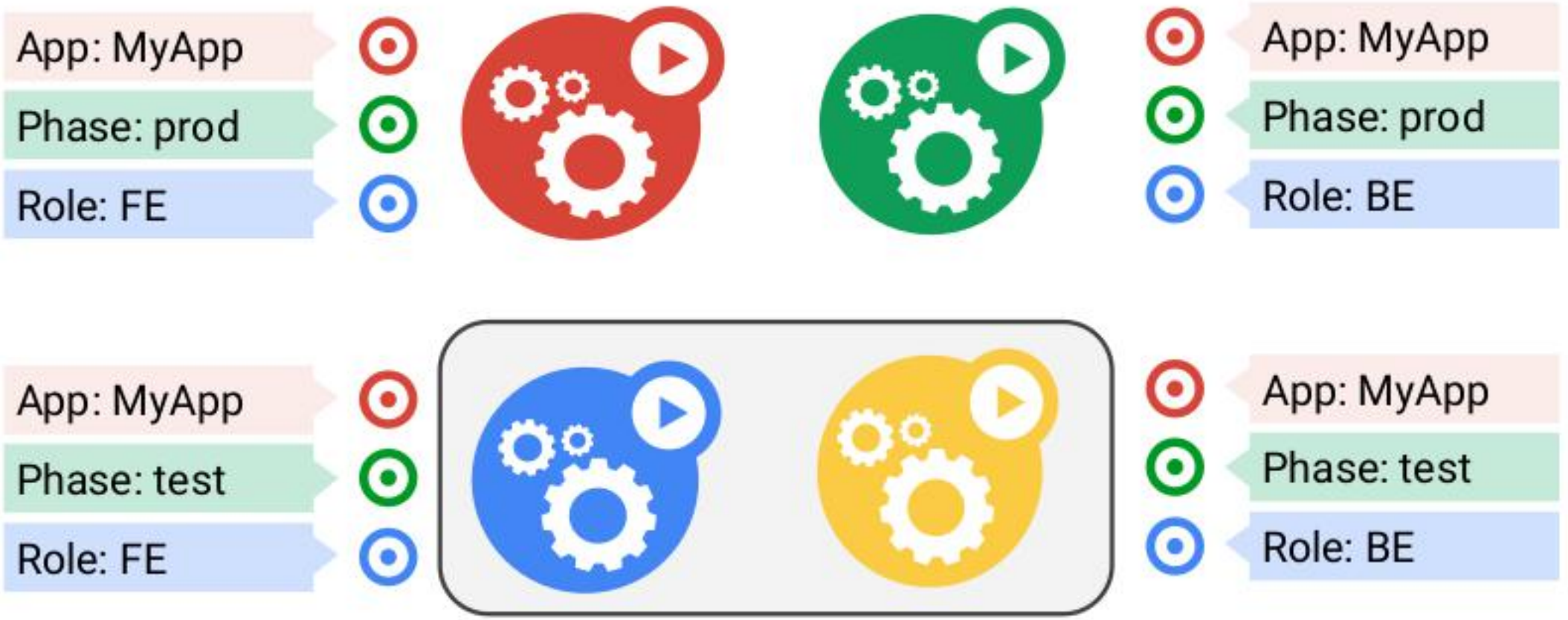
App = MyApp, Role = FE

SEARCH BACK-END

You can search your app's backend

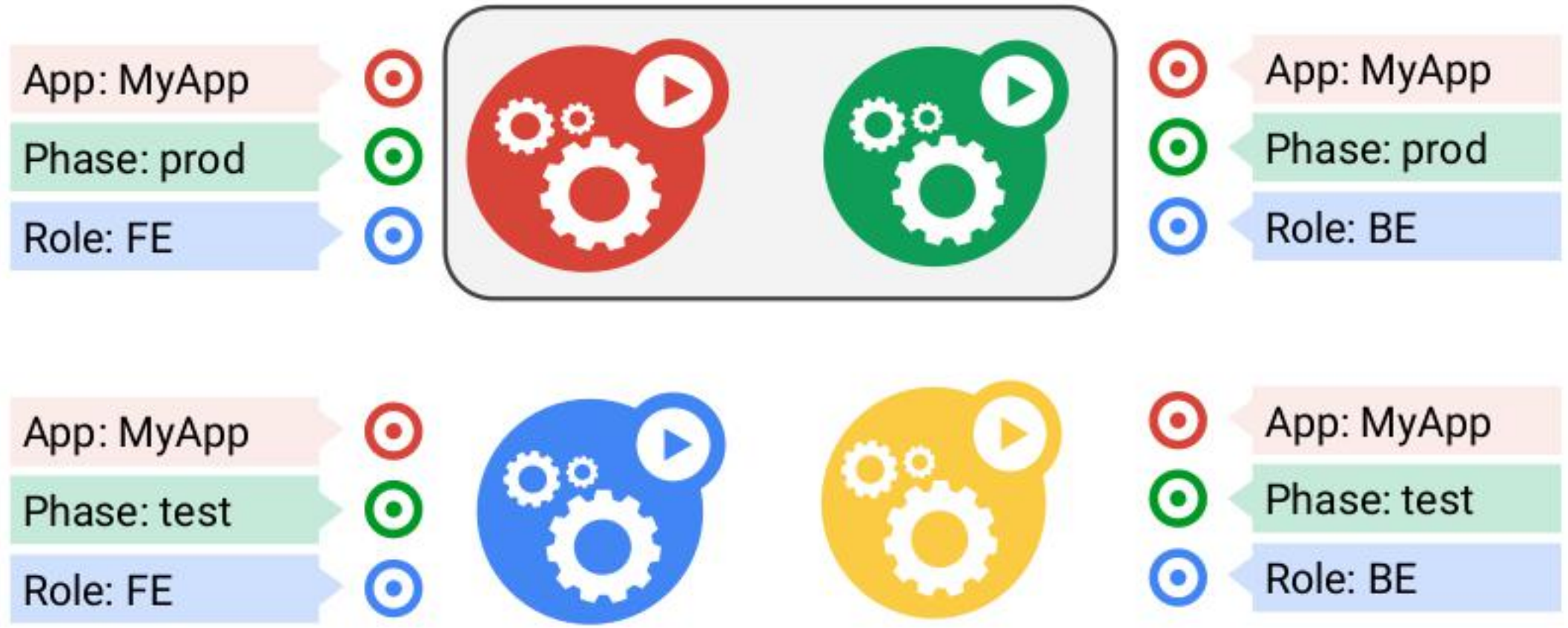
SEARCH TEST PHASE

You can search your app's test phase



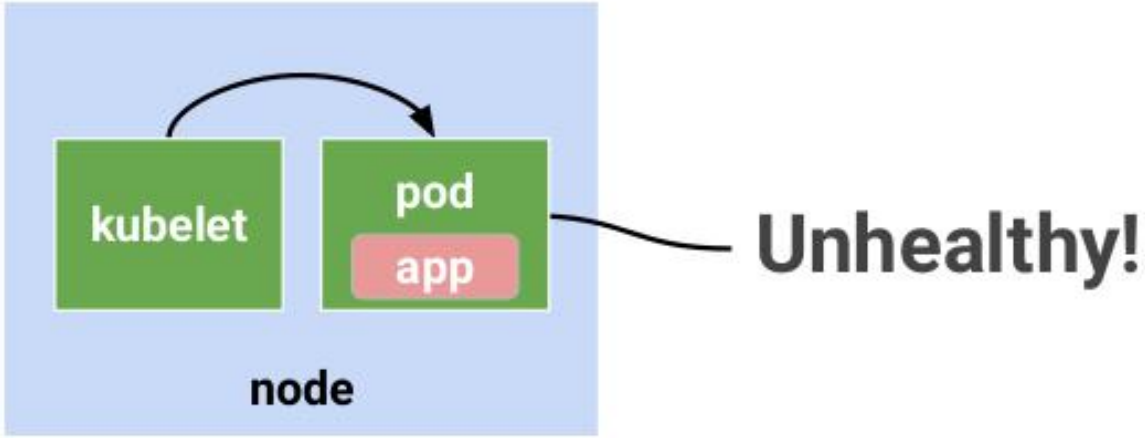
SEARCH PRODUCTION RELEASE

You can search your app's production release



KUBELET

Kubelet checks whether the pod is alive and healthy; if it gets a negative response or no reply...

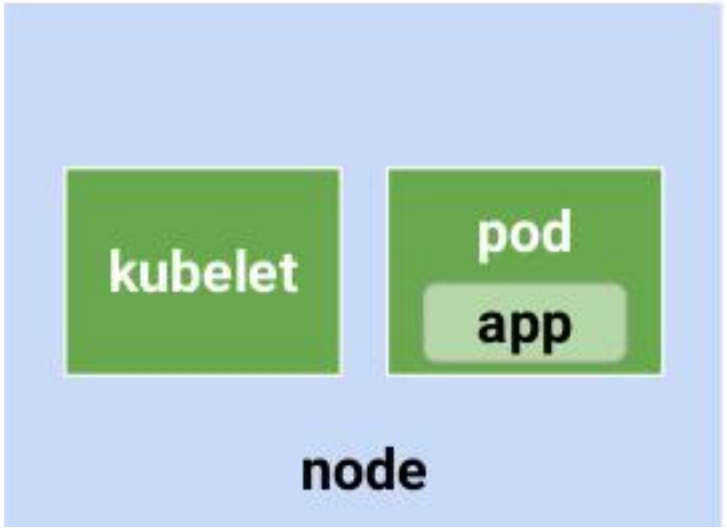


When to use

- In a decomposed compute model
- In a workflow chain where one piece fails for some reason

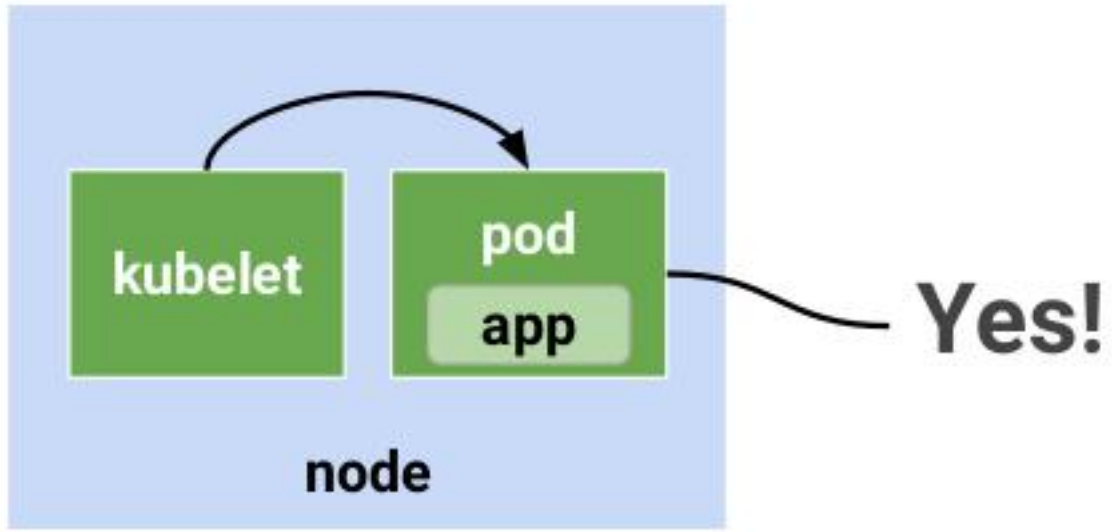
POD RESTART

Kubelet restarts the pod



KUBELET CONTINUES

And continues until it gets a healthy reply



VOLUMES

CONTAINERS
DOCKER
KUBERNETES
SERVICES, LABELS, SELECTORS
VOLUMES
KUBERNETES ON GCP - GKE
KUBERNETES ON AWS
KUBERNETES ON AZURE

KUBERNETES VOLUMES

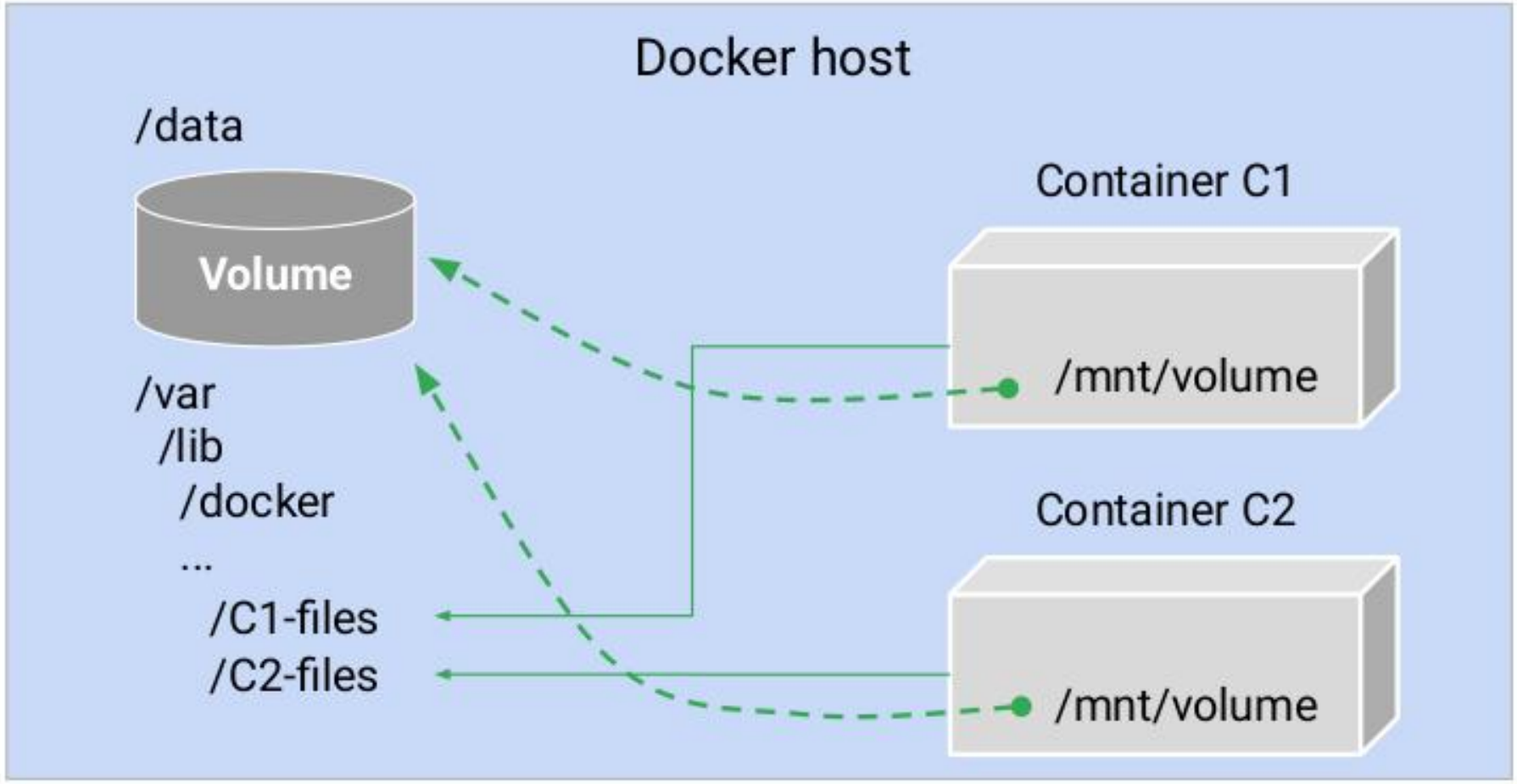
Now let's look at how to get data into your pod and store data persistently using Kubernetes volumes.

Even though a pod isn't meant to be persistent, its data may be.

Docker also has a concept of volumes though it is somewhat looser and less managed than a Kubernetes volume.

DOCKER VOLUMES

Docker provides data storage for containers, but volumes do not provide sharing between containers or lifecycle management



DISCUSSION

Do you remember different storage tiers might be considered here?

Price/performance tradeoff

Would you know where to look for the slide?

QUIZ

Which App Engine environment runs your application in Docker containers on Google Compute Engine virtual machines?

- A. App Engine Standard Environment
- B. App Engine Flexible Environment

QUIZ

Kubernetes Engine's cluster autoscaler automatically add or remove nodes from the cluster as required

- A. False
- B. True

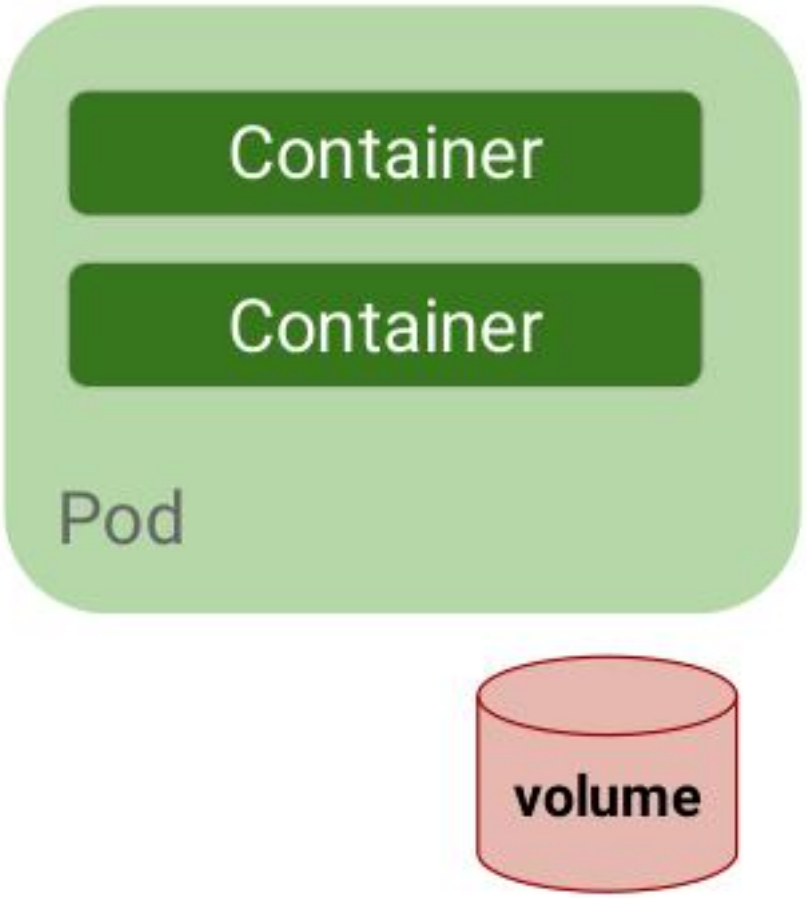
QUIZ

Care should be taken when creating node pools in multizone clusters because node pools are automatically replicated to those zones

- A. False
- B. True

KUBERNETES VOLUMES

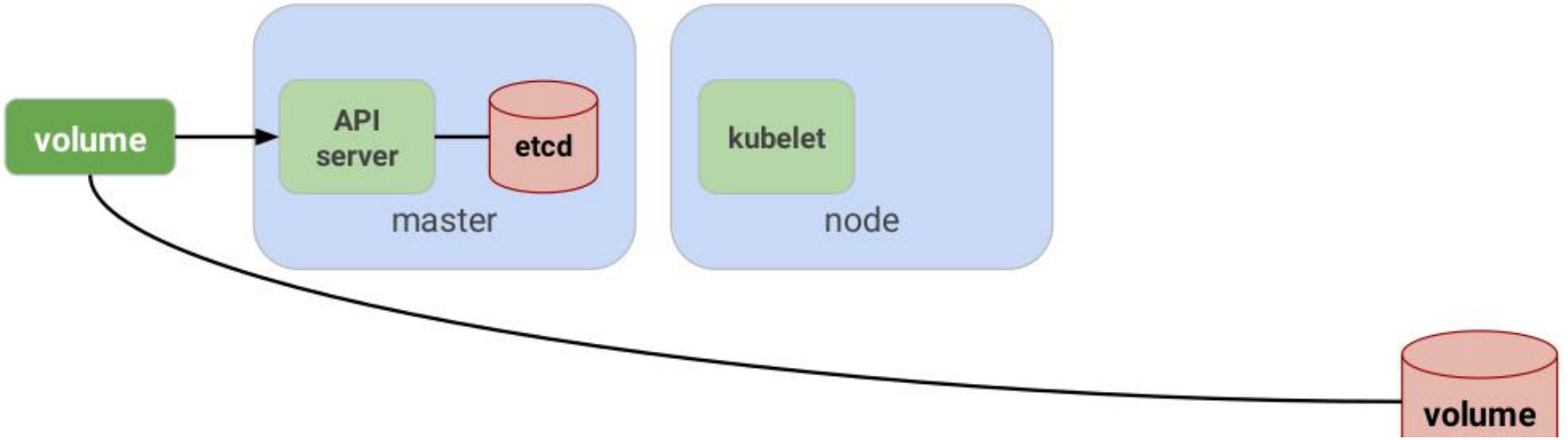
Kubernetes volumes allow containers in pods to share data and be stateful



A VOLUME IS A DIRECTORY

A volume is just a directory, and how it gets created depends on its type

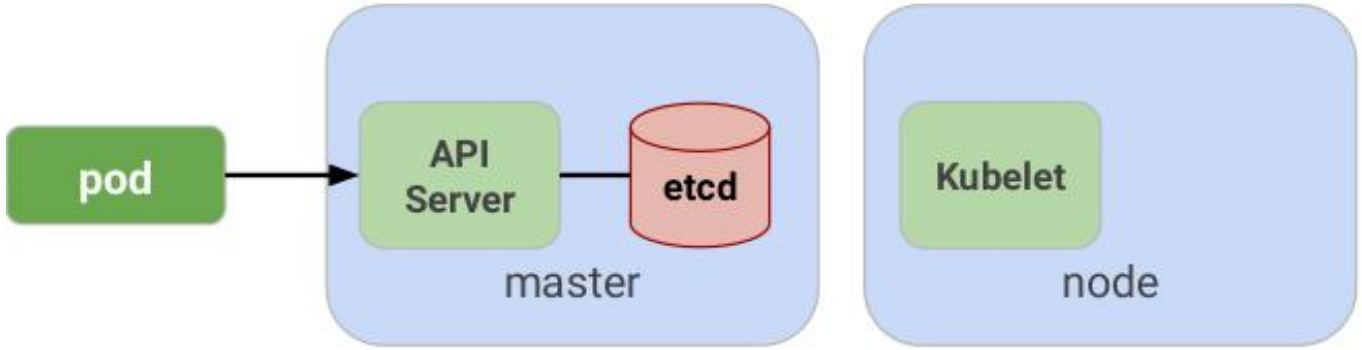
```
1 $> kubectl create <volume>
```

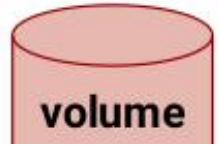


POD CONSUMES THAT DATA

Then you create a pod that consumes that data

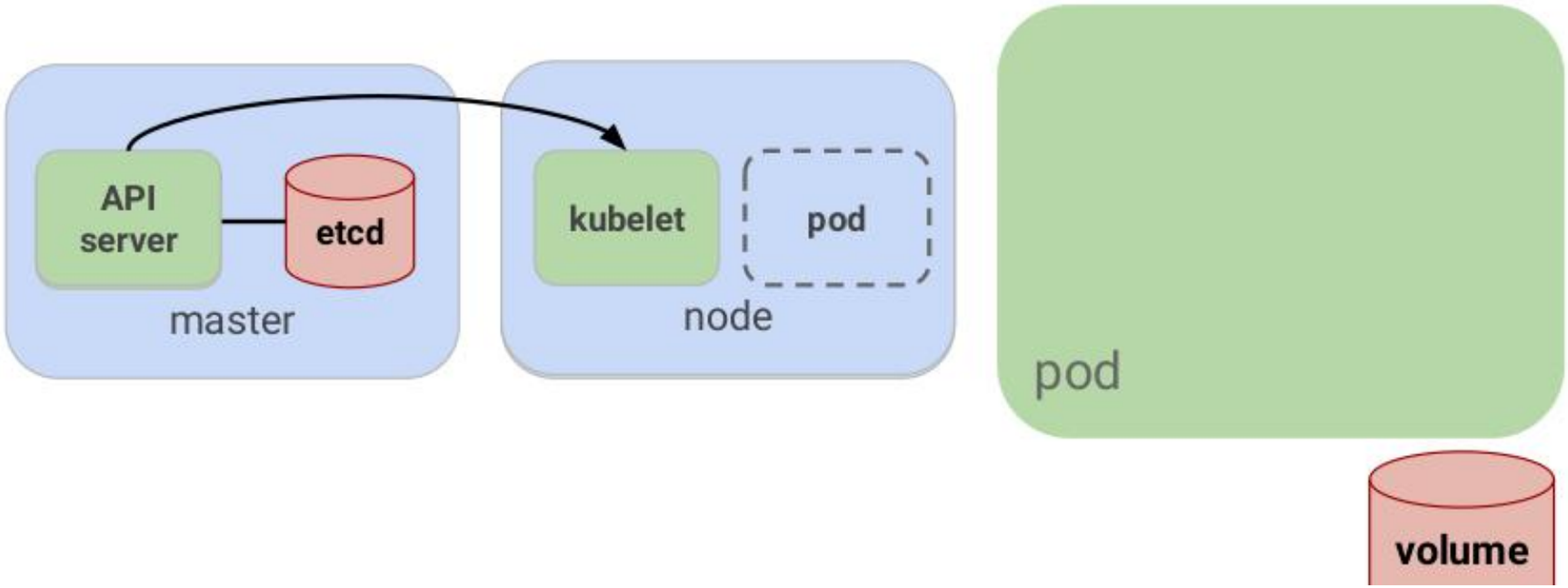
```
1 $> kubectl create -f pod.yaml
```





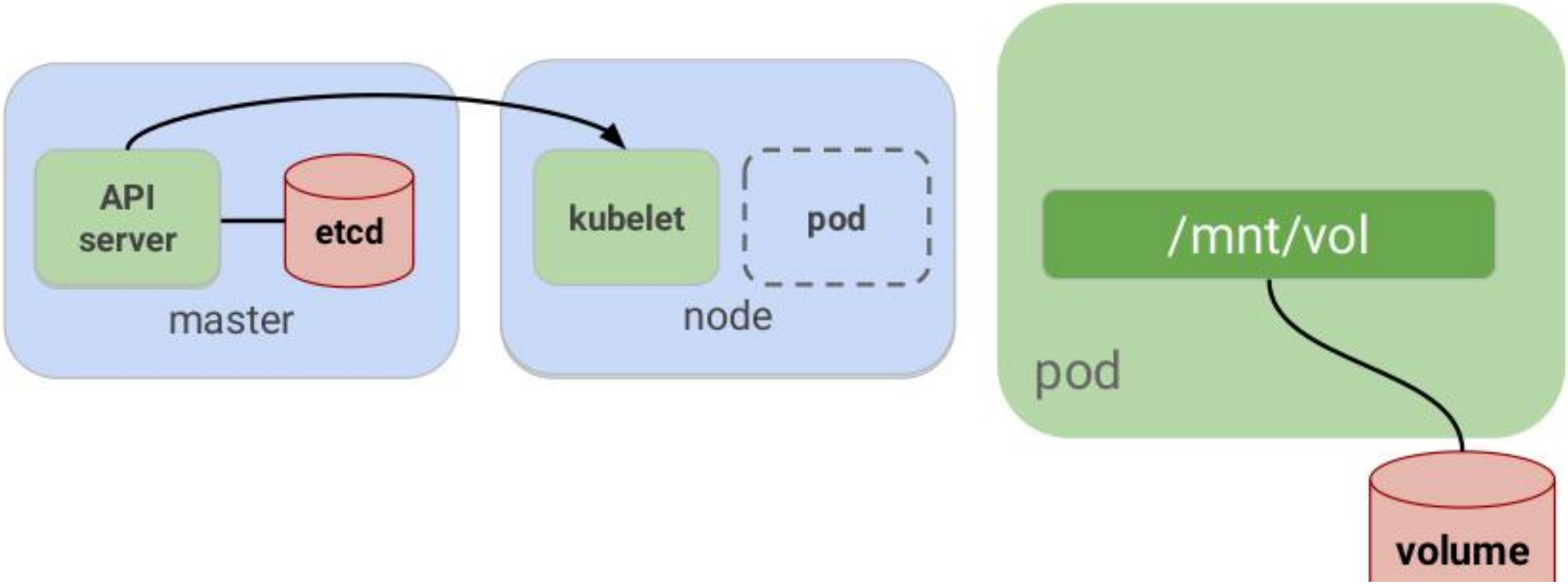
VOLUME ATTACHED TO THE POD

The volume is attached to the pod and made available to containers before they are brought online



DATA ACCESS

Once the volume is attached, data can be mounted into a container's file system



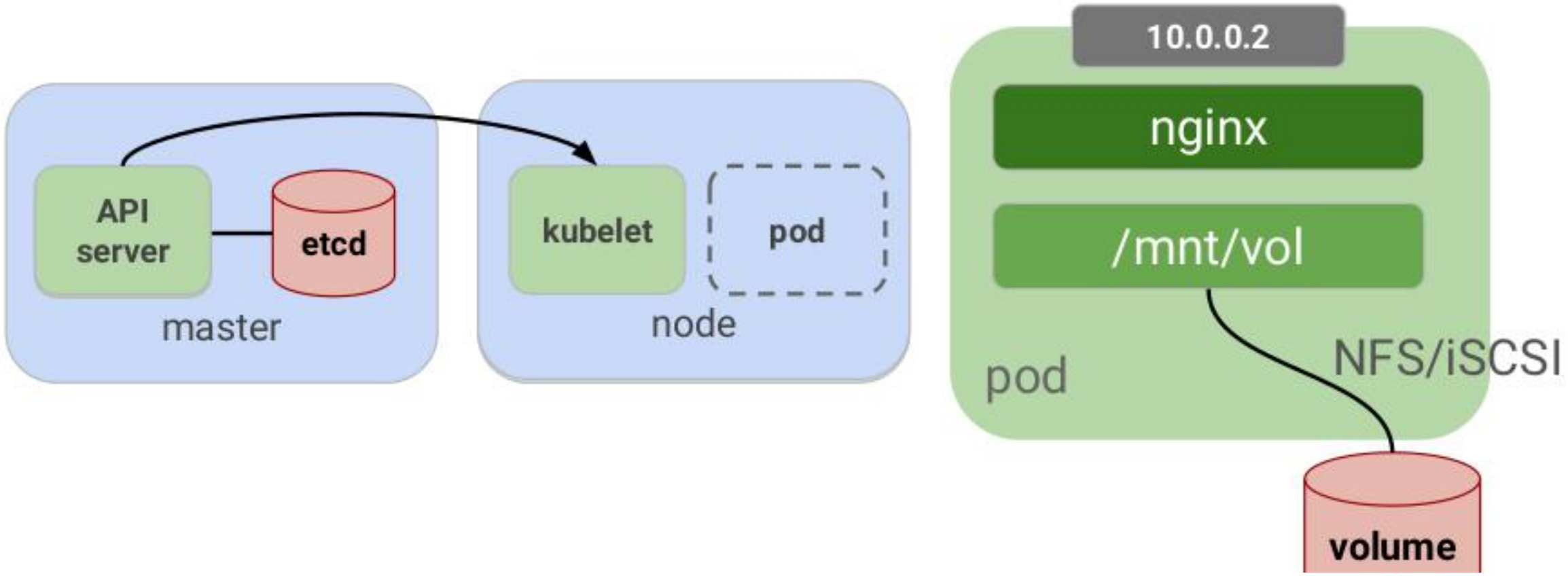
DISCUSSION

Do you think it possible to have multiple volumes?

- For example, with different performance characteristics) mounted in a single container?
- Fast data and slow data?

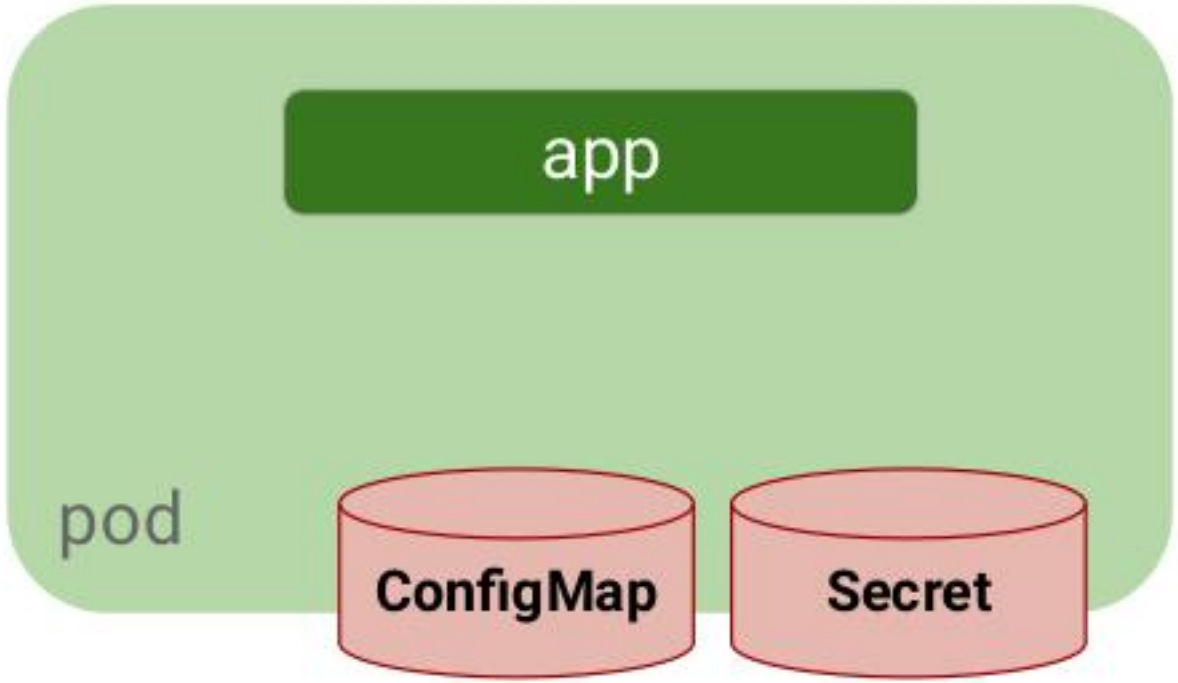
CONTAINER READS DATA

Then the container is run and can get the mounted data



VOLUME LIFECYCLE

Some volumes share the lifecycle of their pod



KUBERNETES ON GCP - GKE

CONTAINERS
DOCKER
KUBERNETES
SERVICES, LABELS, SELECTORS
VOLUMES
KUBERNETES ON GCP - GKE
KUBERNETES ON AWS
KUBERNETES ON AZURE

GCP COMPUTE AND PROCESSING OPTIONS

	Compute Engine	Container Engine ***	App Engine Standard and Flexible	Cloud Functions
Support language	Any	Any	Java, Python, Go, NodeJS	Triggers
Usage model	IaaS	IaaS, PaaS	PaaS	Microservices
	Server	Cluster	Autoscaling managed servers	Serverless
Primary use case	General workloads	Container workloads	Scalable web applications Mobile backend applications	Lightweight Event Actions

GOOGLE CONTAINER ENGINE (GKE)

Fully-managed service

- Kubernetes software maintained
- SLA

Docker format containers

Autoscaling (CPU or memory)

Stackdriver logging and monitoring

Cloud VPN integration

- Hybrid cloud and on premise solutions

Cloud IAM integration

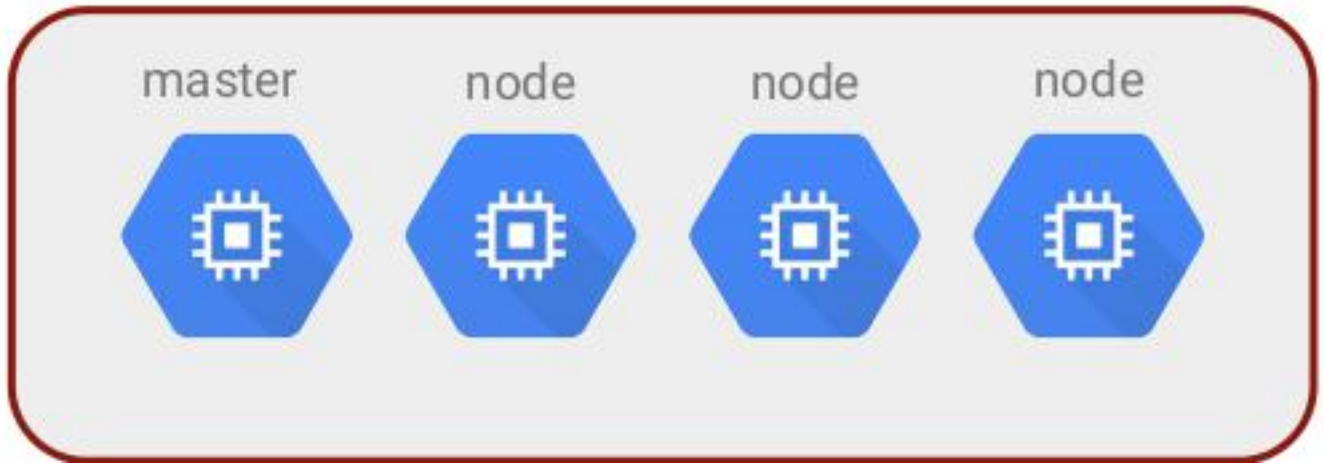
Google Kubernetes Engine

CONTAINER CLUSTER

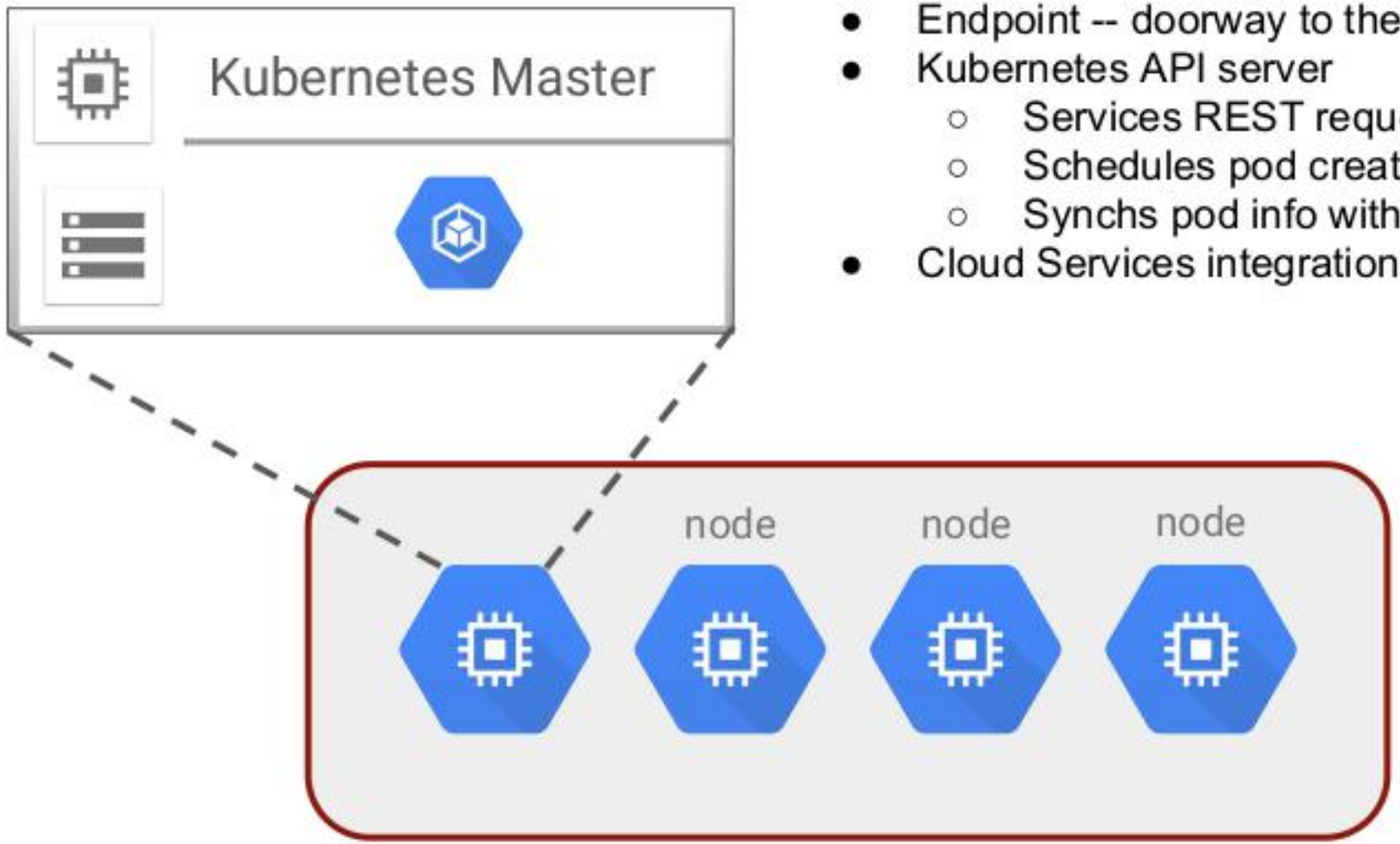
Each node runs:

- Docker runtime
- Kubelet agent
    - Manages scheduled Docker containers

Network proxy

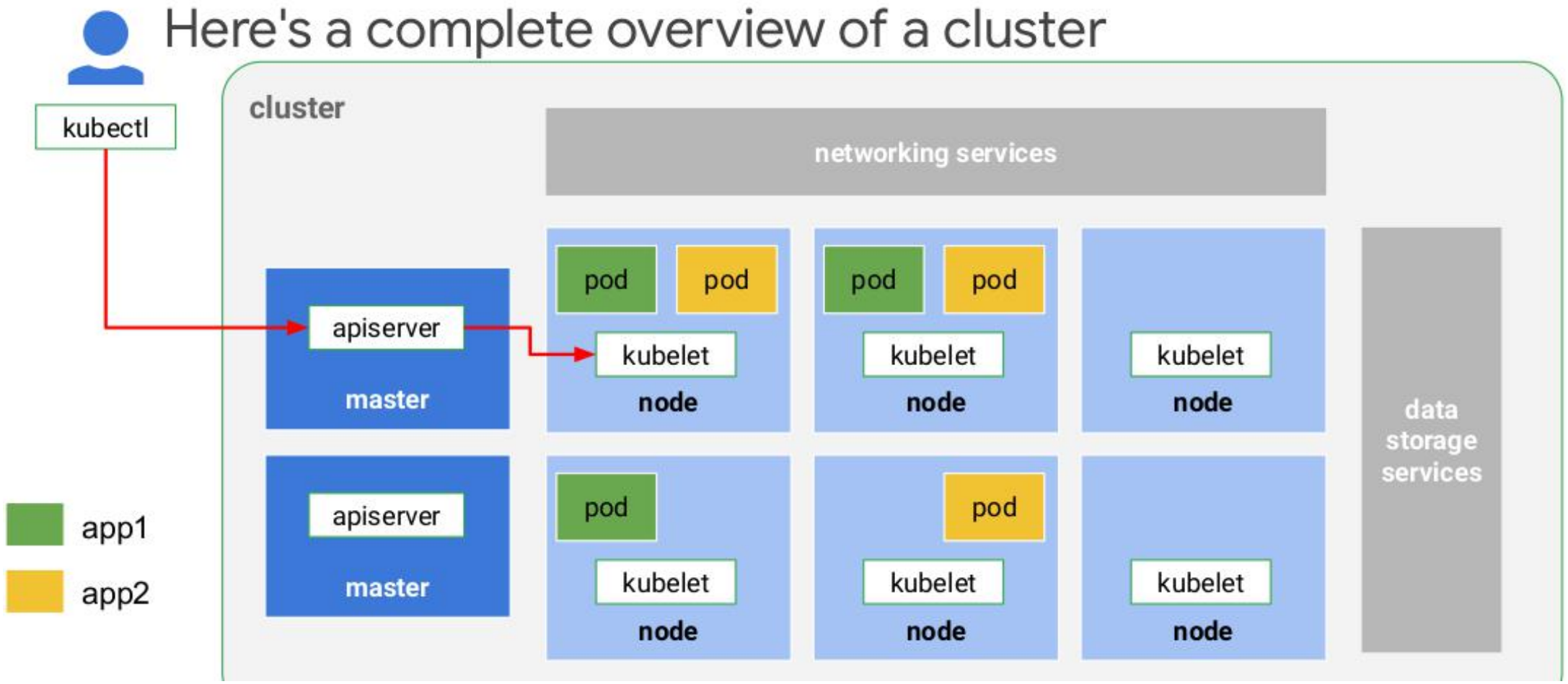


KUBERNETES MASTER ENDPOINT



- Endpoint -- doorway to the cluster
- Kubernetes API server
  - Services REST requests
  - Schedules pod creation/deletion on nodes
  - Synchs pod info with service info
- Cloud Services integration

CLUSTER VIEW

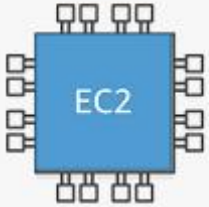


KUBERNETES ON AWS

CONTAINERS
DOCKER
KUBERNETES
SERVICES, LABELS, SELECTORS
VOLUMES
KUBERNETES ON GCP - GKE
KUBERNETES ON AWS
KUBERNETES ON AZURE

AMAZON ELASTIC CONTAINER REGISTRY

RUNNING KUBERNETES ON AWS

IF YOU WANT TO...	CONSIDER USING	
Fully manage your Kubernetes deployment. Provision and run Kubernetes on your choice of powerful instance types.	Amazon EC2	
Run Kubernetes without needing to provision or manage master instances and etcd.	Amazon EKS	
Store, encrypt, and manage container images for fast deployment.	Amazon ECR	

AMAZON EKS

Amazon Elastic Kubernetes Service (Amazon EKS)

- Start, run, and scale Kubernetes applications in the AWS cloud or on-premises.
- Amazon EKS helps you provide highly-available and secure clusters
- Automates key tasks such as patching, node provisioning, and updates

EKS runs upstream Kubernetes

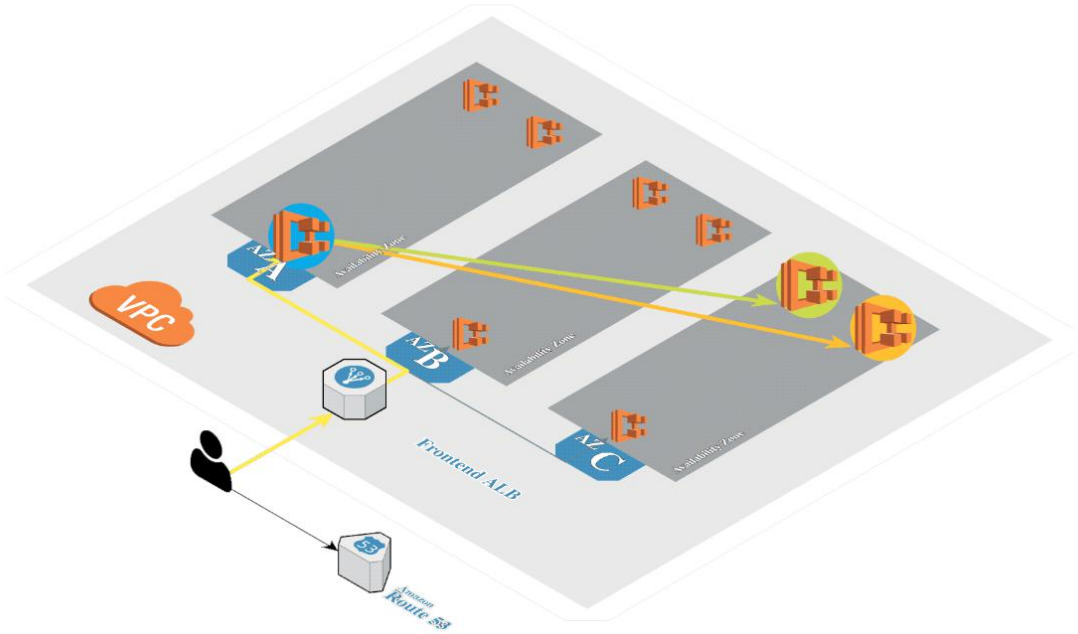
- Certified Kubernetes conformant
- You can easily migrate any standard Kubernetes application to EKS without needing to refactor your code.

Amazon EKS Distro

- Host and operate your Kubernetes clusters on-premises and at the edge
- AWS Outposts
- AWS Wavelength
- Amazon EKS Anywhere is coming in 2021.

EKS WORKSHOP

multiple ways to configure VPC, ALB, and EC2 Kubernetes workers and Amazon Elastic Kubernetes Service



CREATE USER

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	workshop
AWS access type	AWS Management Console access - with a password
Console password type	Custom
Require password reset	No
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AdministratorAccess

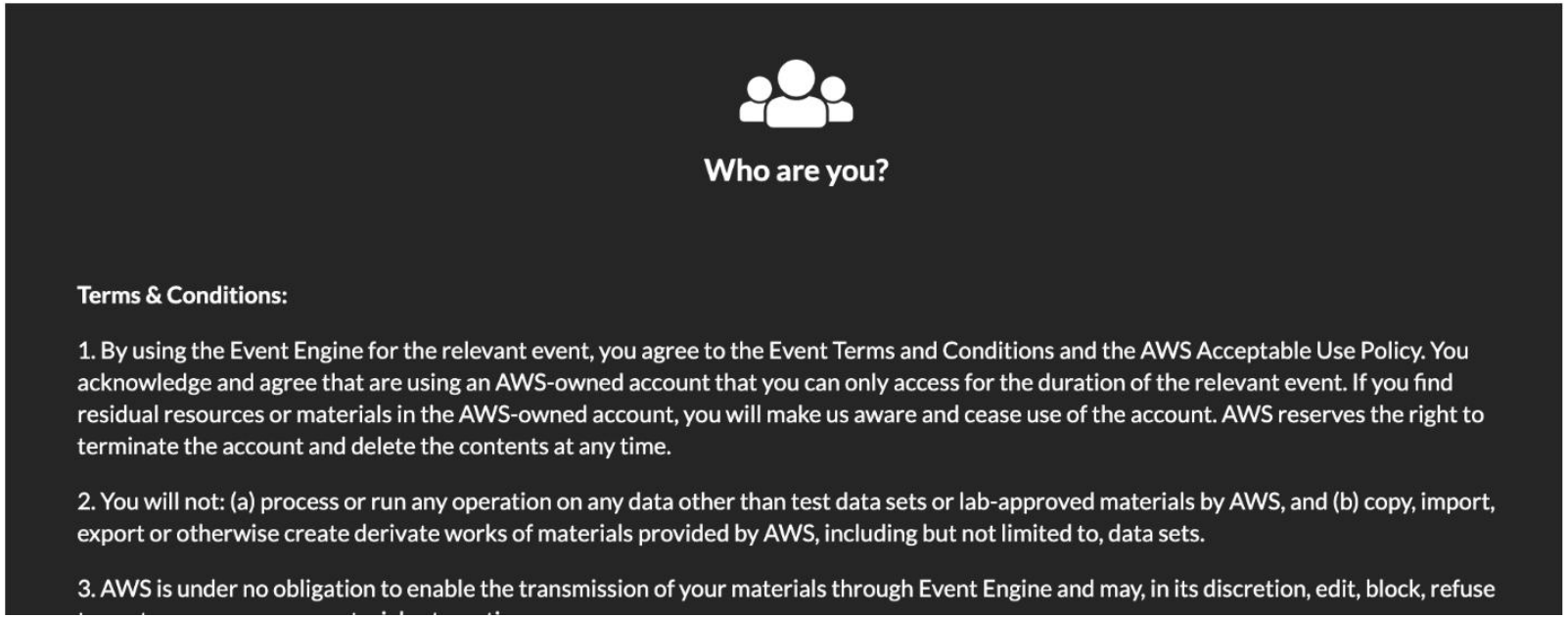
LOGIN TO AWS WORKSHOP PORTAL

The workshop creates an AWS account and a Cloud9 environment

Login to AWS Workshop Portal

This workshop creates an AWS account and a Cloud9 environment. You will need the **Participant Hash** provided upon entry, and your email address to track your unique session.

Connect to the portal by clicking the button or browsing to https://dashboard.eventengine.run/. The following screen shows up.

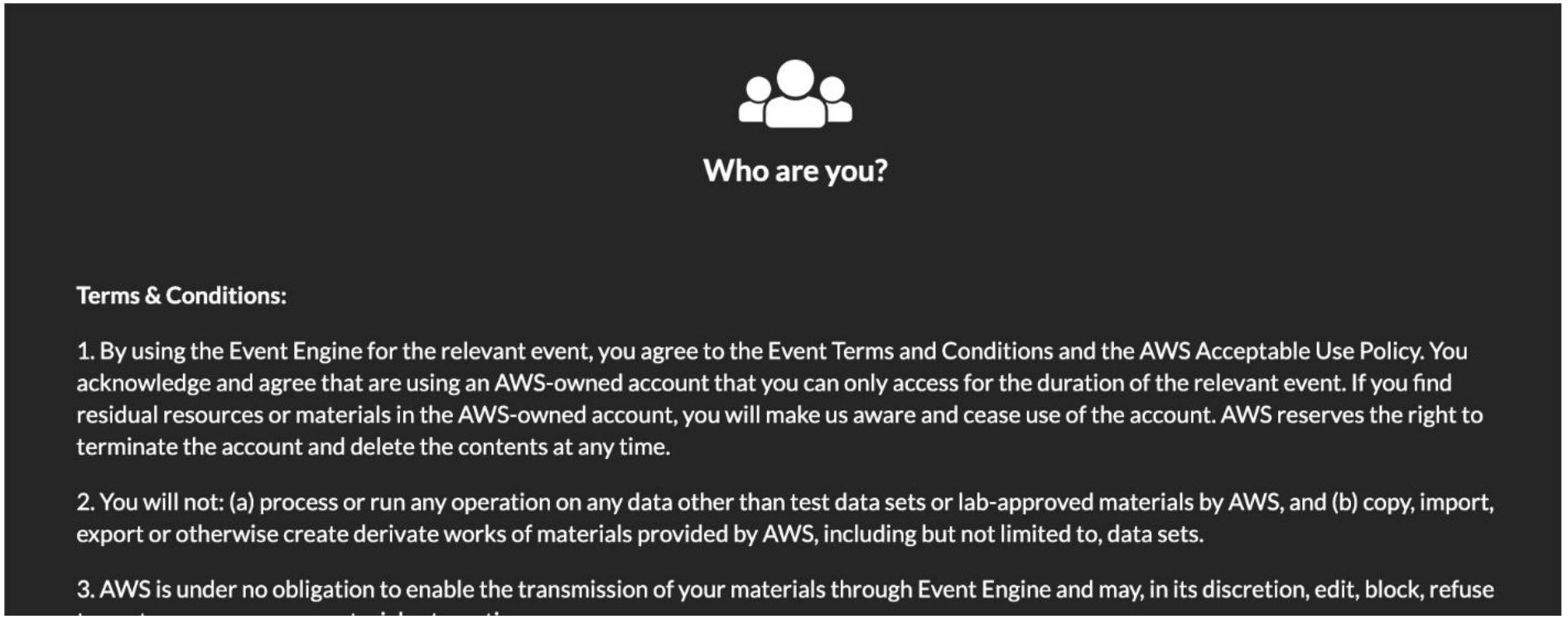


LAUNCH CLOUD9

Login to AWS Workshop Portal

This workshop creates an AWS account and a Cloud9 environment. You will need the **Participant Hash** provided upon entry, and your email address to track your unique session.

Connect to the portal by clicking the button or browsing to https://dashboard.eventengine.run/. The following screen shows up.



IN CLOUD9

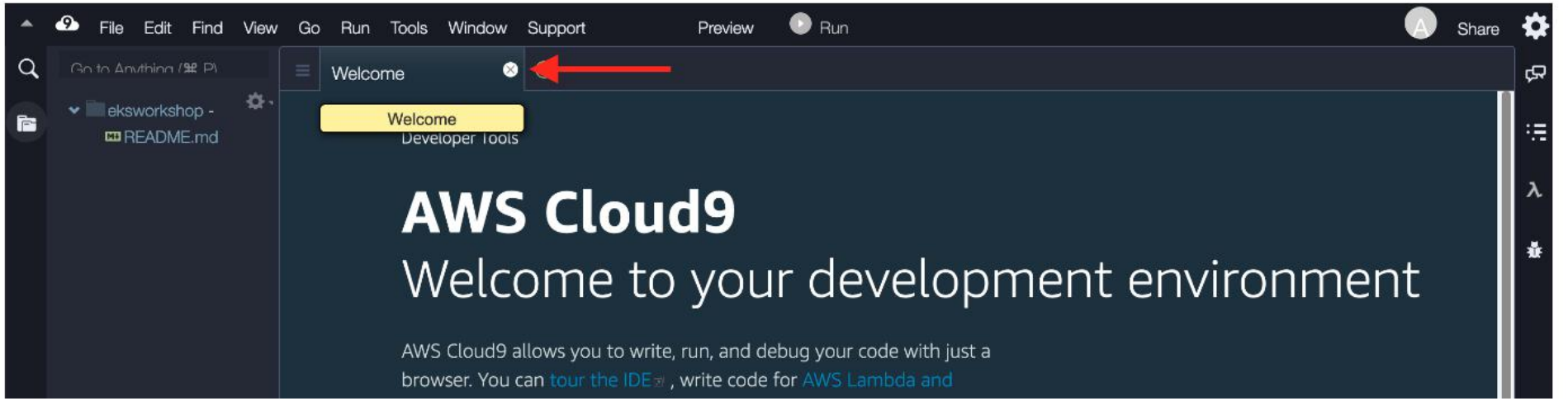
Oregon | Ireland | Ohio | Singapore

Create a Cloud9 Environment: https://us-west-2.console.aws.amazon.com/cloud9/home?region=us-west-2

- Select **Create environment**
- Name it **eksworkshop**, click Next.
- Choose **t3.small** for instance type, take all default values and click **Create environment**

When it comes up, customize the environment by:

- Closing the **Welcome tab**



ON COMMAND-LINE

```
sudo curl --silent --location -o /usr/local/bin/kubectl \
   https://amazon-eks.s3.us-west-2.amazonaws.com/1.17.11/2020-09-18/bin/linux/amd64/kubectl

sudo chmod +x /usr/local/bin/kubectl
```

Install kubectl

Update awscli

Install jq, envsubst (from GNU gettext utilities) and bash-completion

Install yq for yaml processing

set the AWS Load Balancer Controller version

Create the role `eksworkshop-admin`

CREATE AN EKS CLUSTER

Test the cluster

```
eksctl create cluster -f eksworkshop.yaml
```

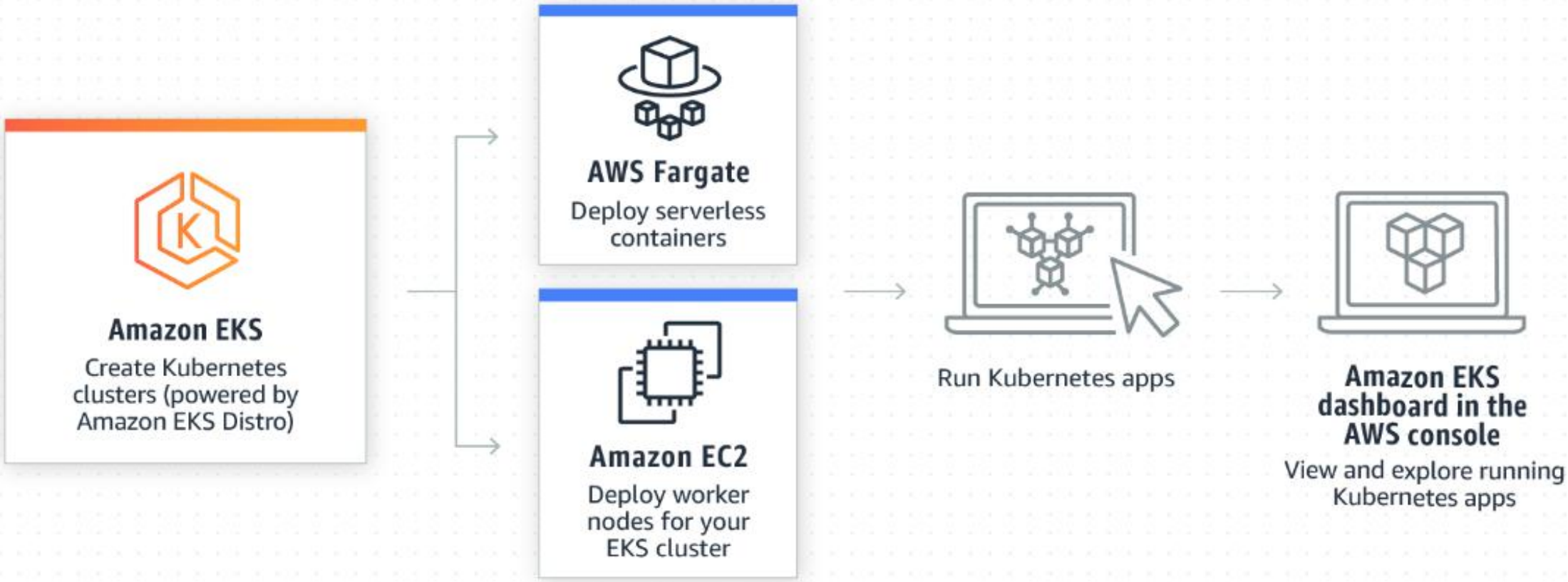
```
kubectl get nodes
```

Export the Worker Role Name for use throughout the workshop

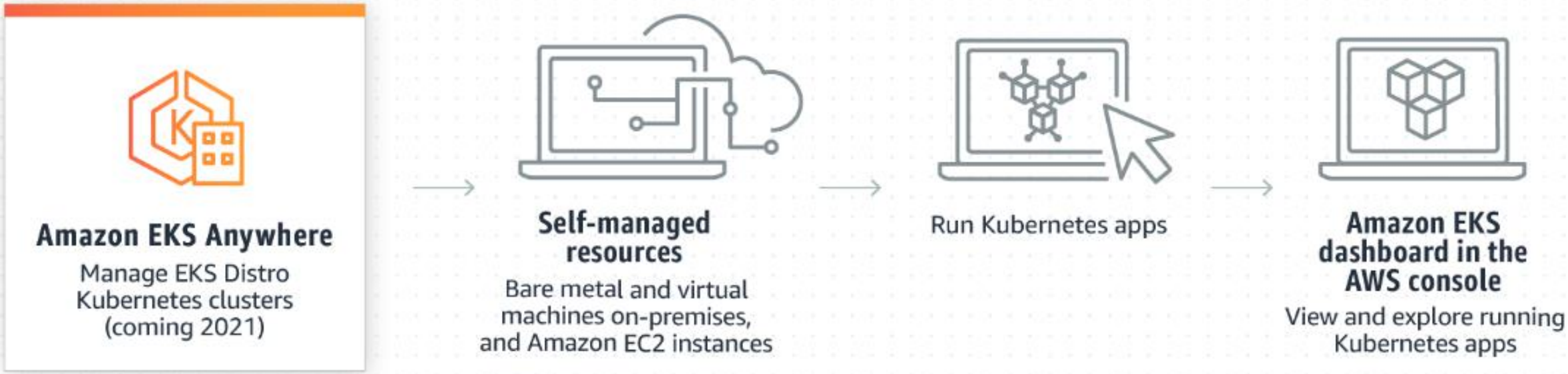
Congratulations!

You now have a fully working Amazon EKS Cluster that is ready to use!

DEPLOY APPLICATIONS WITH AMAZON EKS



DEPLOY APPLICATIONS WITH AMAZON EKS

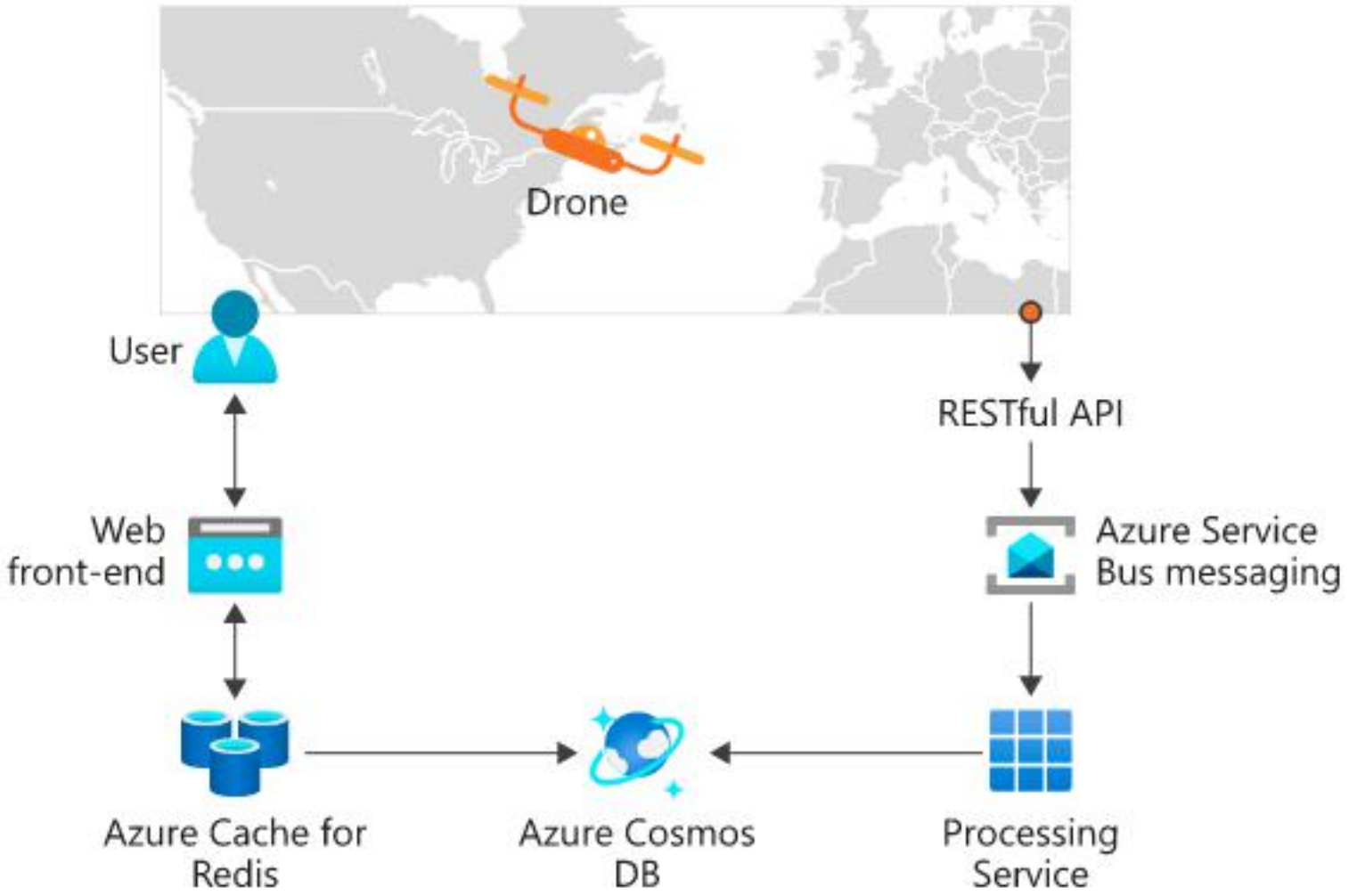


KUBERNETES ON AZURE

CONTAINERS
DOCKER
KUBERNETES
SERVICES, LABELS, SELECTORS
VOLUMES
KUBERNETES ON GCP - GKE
KUBERNETES ON AWS
KUBERNETES ON AZURE

AZURE CONTAINER REGISTRY

EXAMPLE OF KUBERNETES USE ON AZURE



AZURE KUBERNETES SERVICE (AKS)

Elastic provisioning of capacity without the need to manage the infrastructure, and with the ability to add event-driven autoscaling and triggers through KEDA

DISCUSSION

Does anyone know of an example of autoscaling support from Research applications?

Outside of very big science?

(Sounds fairly fancy for a typical research project, but perhaps useful in sensor network applications, or where instruments can burst out data and events can be used to scale up data processing?)

AZURE KUBERNETES SERVICE (AKS)

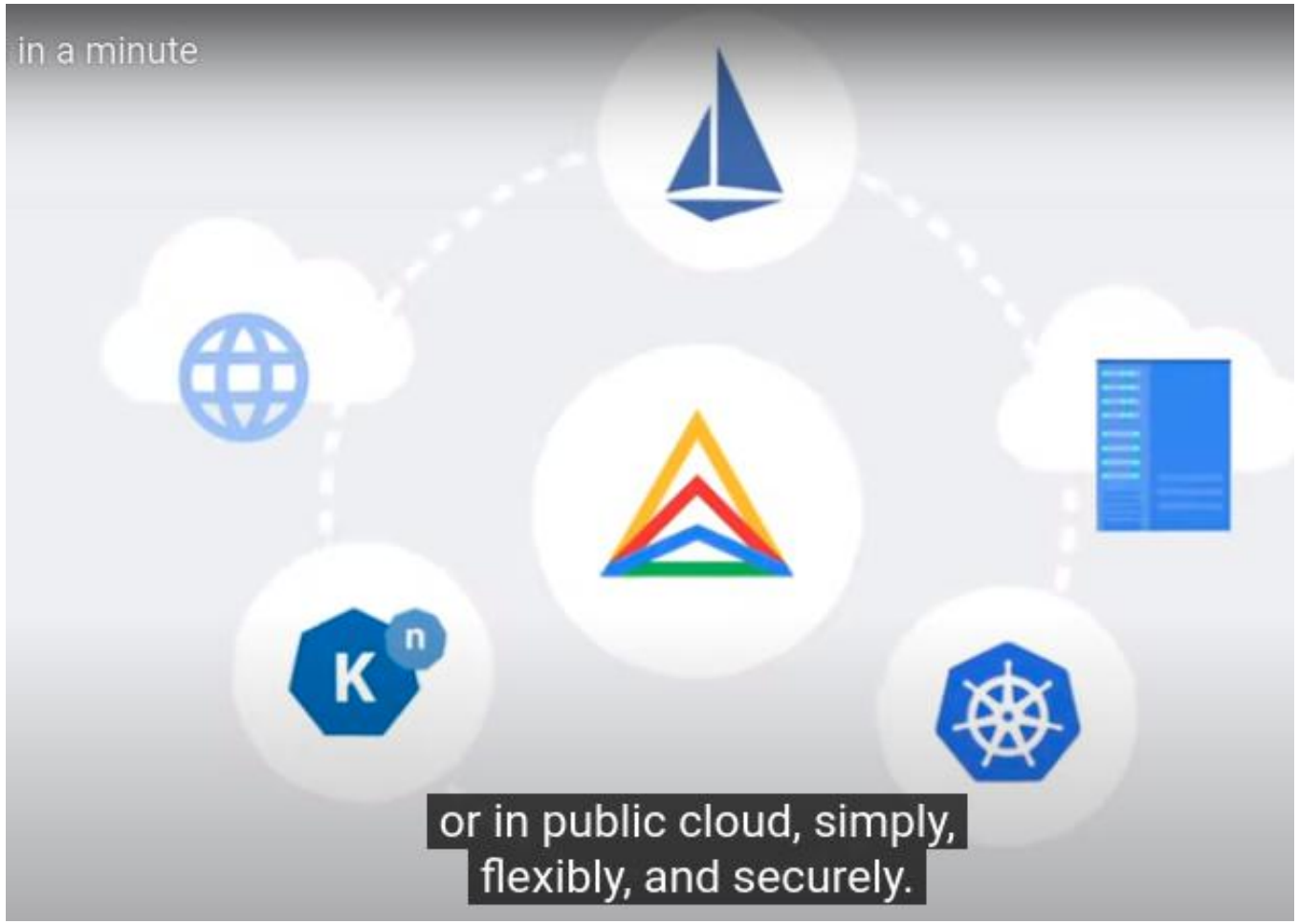
"Faster end-to-end development experience" (MSFT) through Visual Studio Code Kubernetes tools, Azure DevOps, and Azure Monitor

AZURE KUBERNETES SERVICE (AKS)

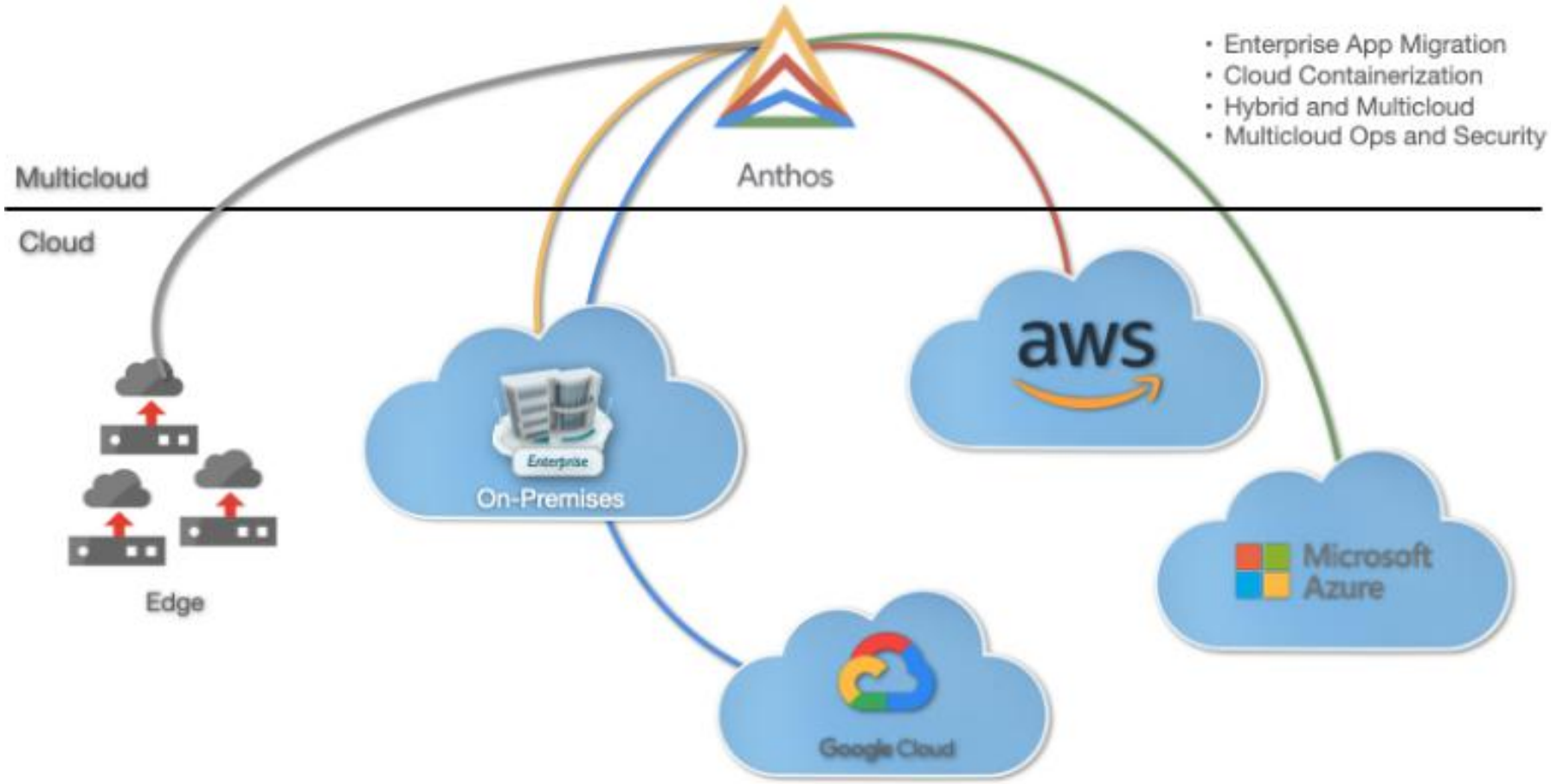
Advanced identity and access management using Azure Active Directory, and dynamic rules enforcement across multiple clusters with Azure Policy

- Single sign-on to any cloud app
- Enforce Multi-Factor Authentication with SaaS
- Works with multiple platforms and devices
- Integrate with on-premises Active Directory
- "Enterprise" (large production) Scale and SLA
- To discuss - can you think of relevant use cases?

ANTHOS - MULTICLOUD FROM GOOGLE

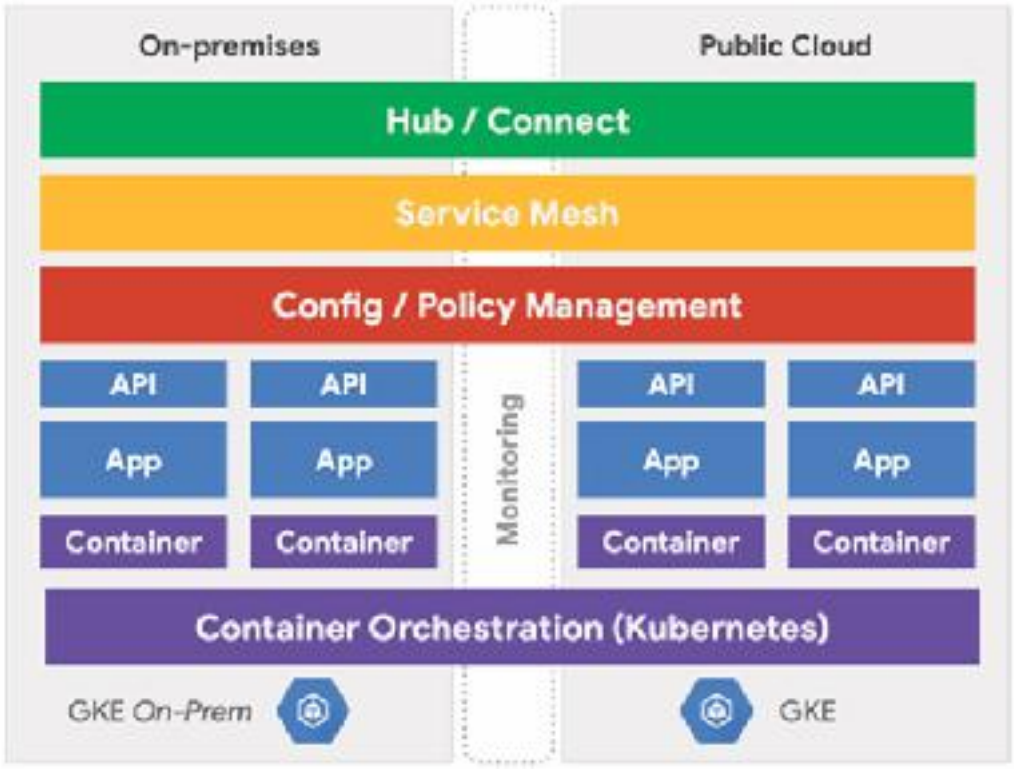


WHAT IS ANTHOS?



(Image from ZDnet)

ANTHOS AT A GLANCE



Anthos works on AWS and is in preview on Azure

Anthos will be covered in more detail in the next section

CONGRATS ON COMPLETION