GCP Pe TENT STORAGE O BERNETES

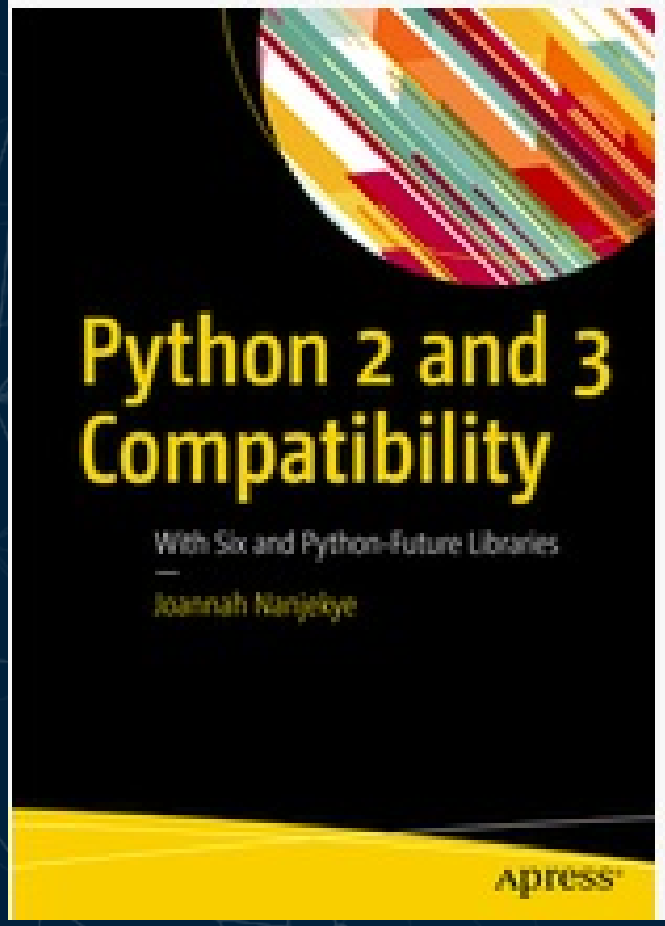
SETTING UP PERSISTENT VOLUMES ON YOUR CONTAINERS

Joann

About Me

- Joannah Nanjekye
- Software Engineer
- Aeronautical Engineer
- FOSS (Open source) Contributor.
- Pypy with IBM.

Author...



Python 2 and 3 Compatibility

With Six and Python-Future Libraries

Authors: **Nanjekye, Joannah**

Download source code

Coming soon..

Deploying and Managing Python Applications with Kubernetes By Apress.

Get some of my container talks

1. Ruby in Containers (EuRuKo 2018, Vienna, Austria) - Best Practices for implementing images.

2. Deploying and Managing Python with Kubernetes (PyconZA, Joburg, 2018).

3. Unit Testing your Docker Images with the Google Container Test Framework (DevFest Nairobi, 2018)

Agenda

- What are containers.
- Their Orchestration with Kubernetes.
- GCE Persistent Storage on Kubernetes.

Sys Admins

Mean while ...Developers

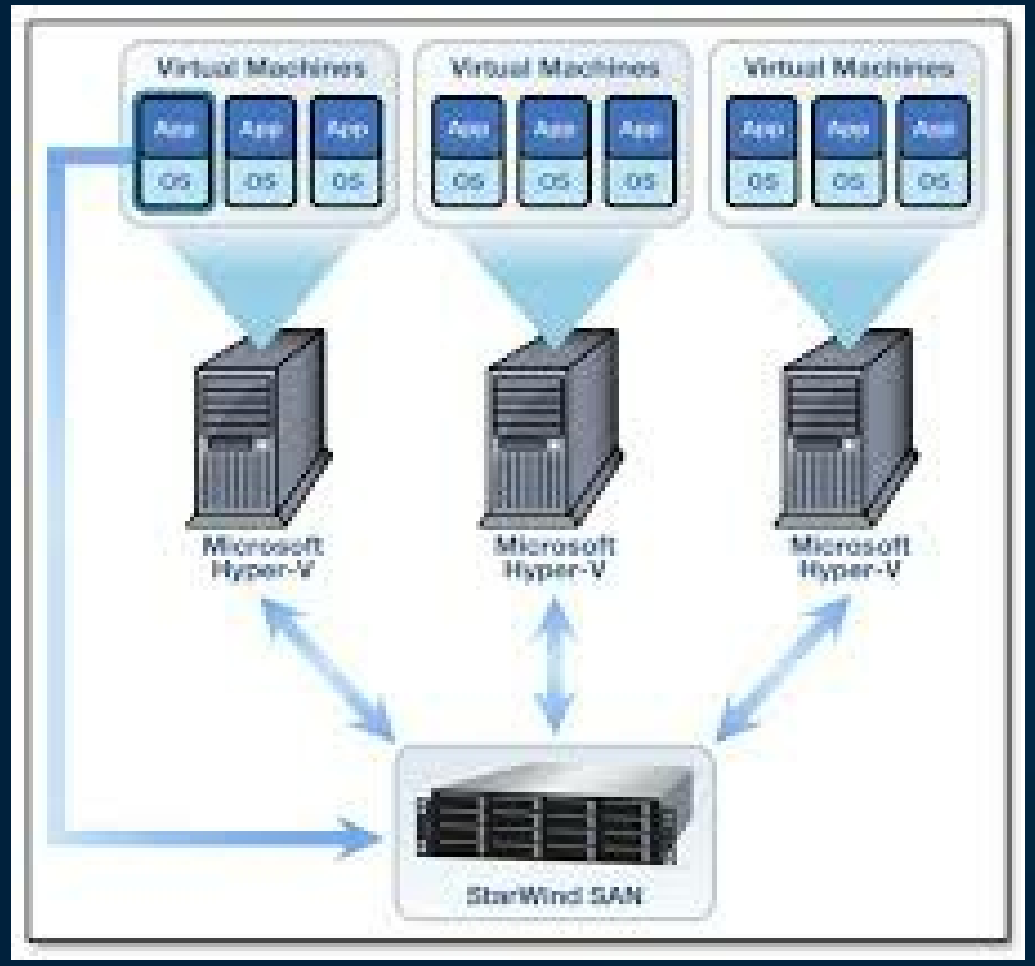
Deploying Apps Has come a long way...

- Physical Machines
- Virtual Machines
- Containers

Physical Machines

- Typically one application per host

Virtual Machines

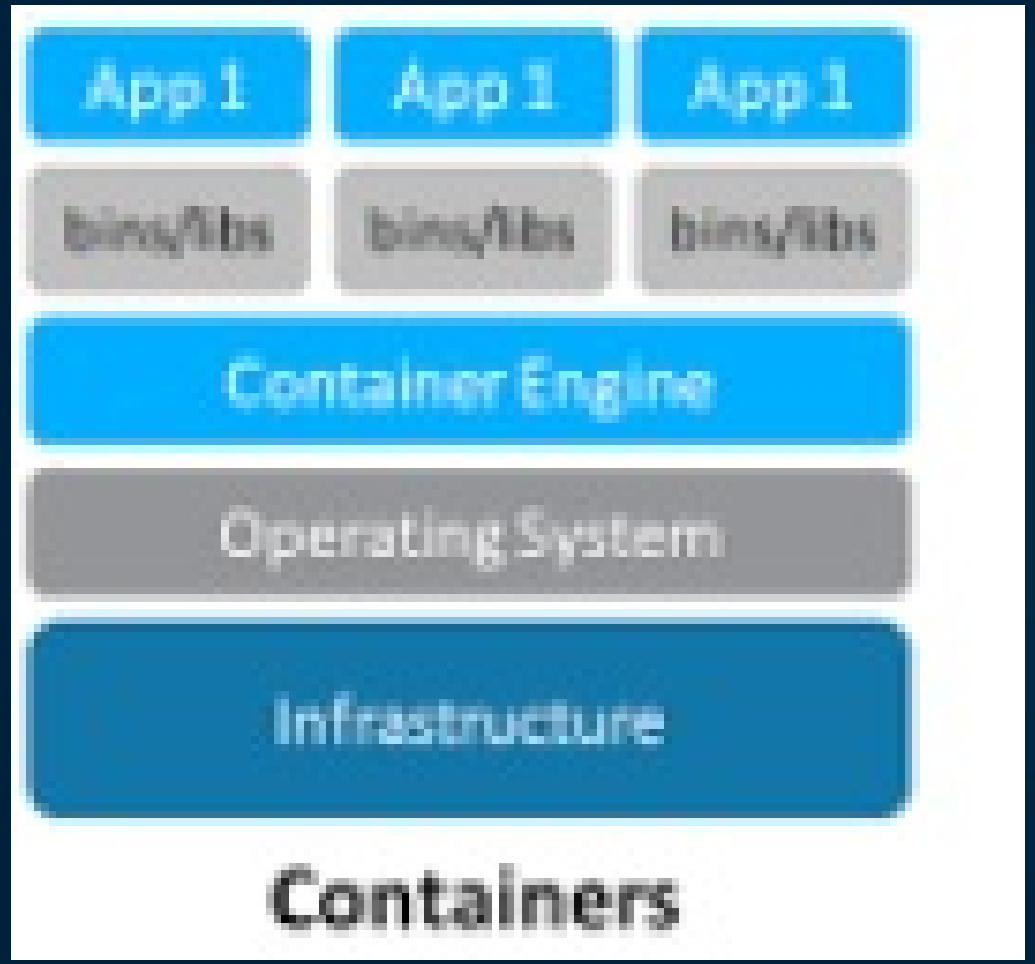


- Virtualize the hardware.
- Run multiple applications on same hardware.

CONTAINERS

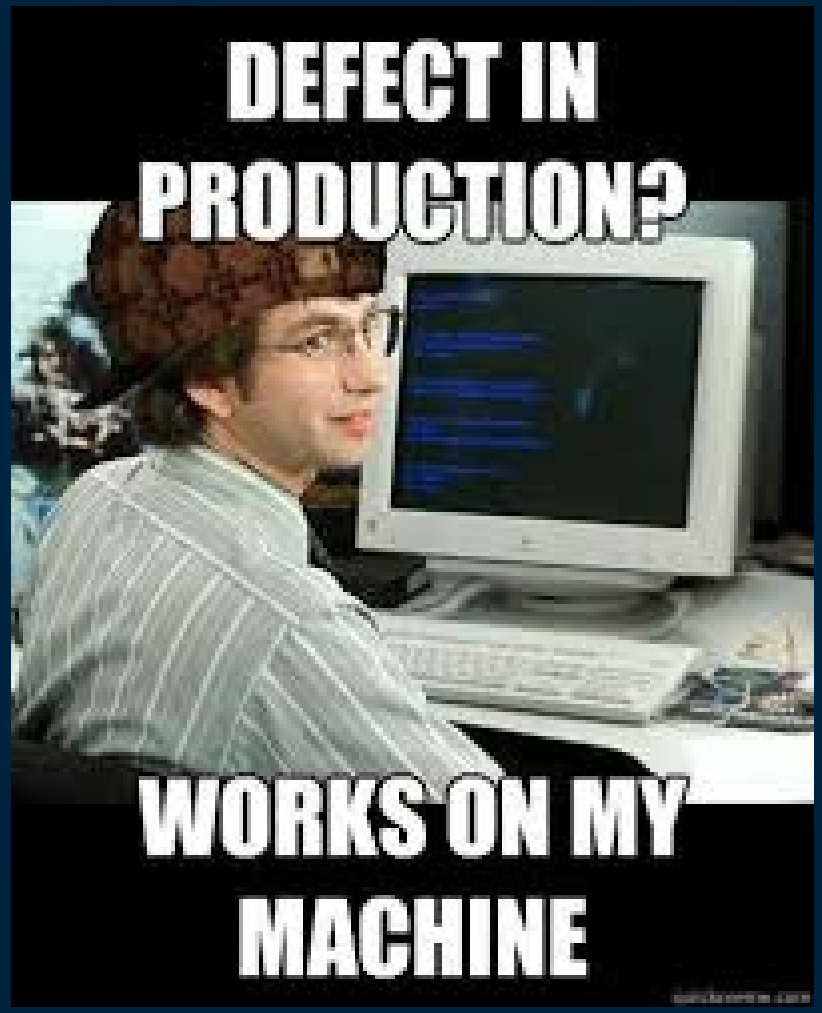
- Logical Packaging for applications.
- To run abstracted from the environment.

Virtualize the OS



- Run a container runtime on host operating system.
- Better performance due to absence of guest operating systems

Why...



Most importantly...

Containers usher us into using better platforms like kubernetes.

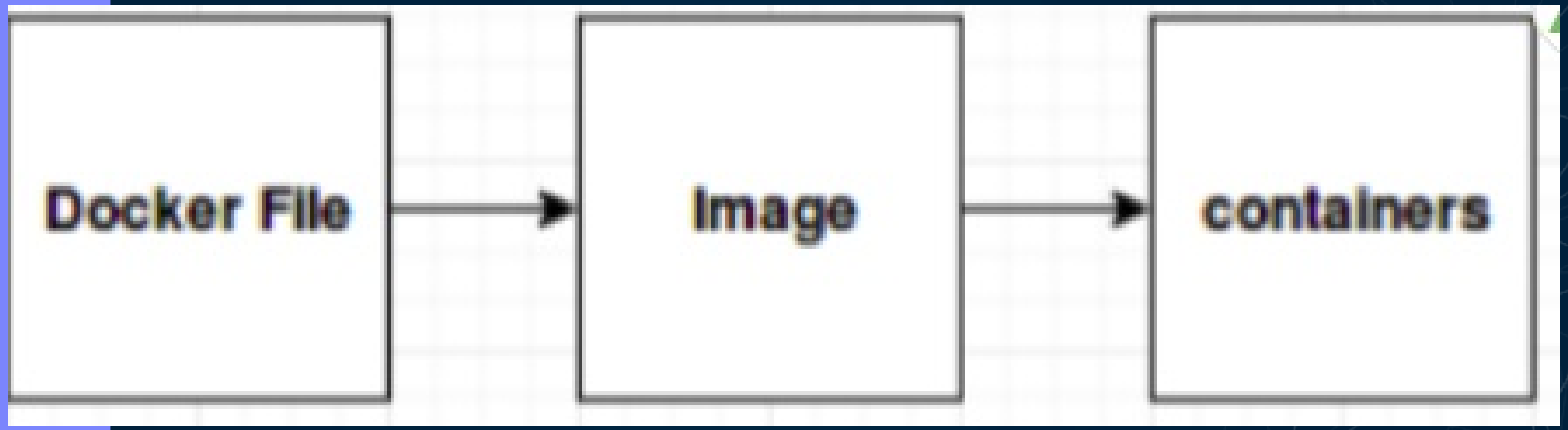
But also...

They are the most spoken Language of the cloud.

To a developer

- Productivity improved.
- Portability.
- Ease of scaling.

Containers in a nut shell



Practically : Given a basic app

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello  World!'

if __name__ == '__main__':
    app.run(host='0.0.0.0')
```

Docker file

```
FROM gliderlabs/alpine:3.2

RUN apk-install python

COPY requirements.txt .
RUN apk --update add --virtual build-dependencies py-pip \
  && pip install -r requirements.txt \
  && apk del build-dependencies

COPY . /code
WORKDIR /code

EXPOSE 5000
ENTRYPOINT ["python", "myapp.py"]
```

Build and run the image

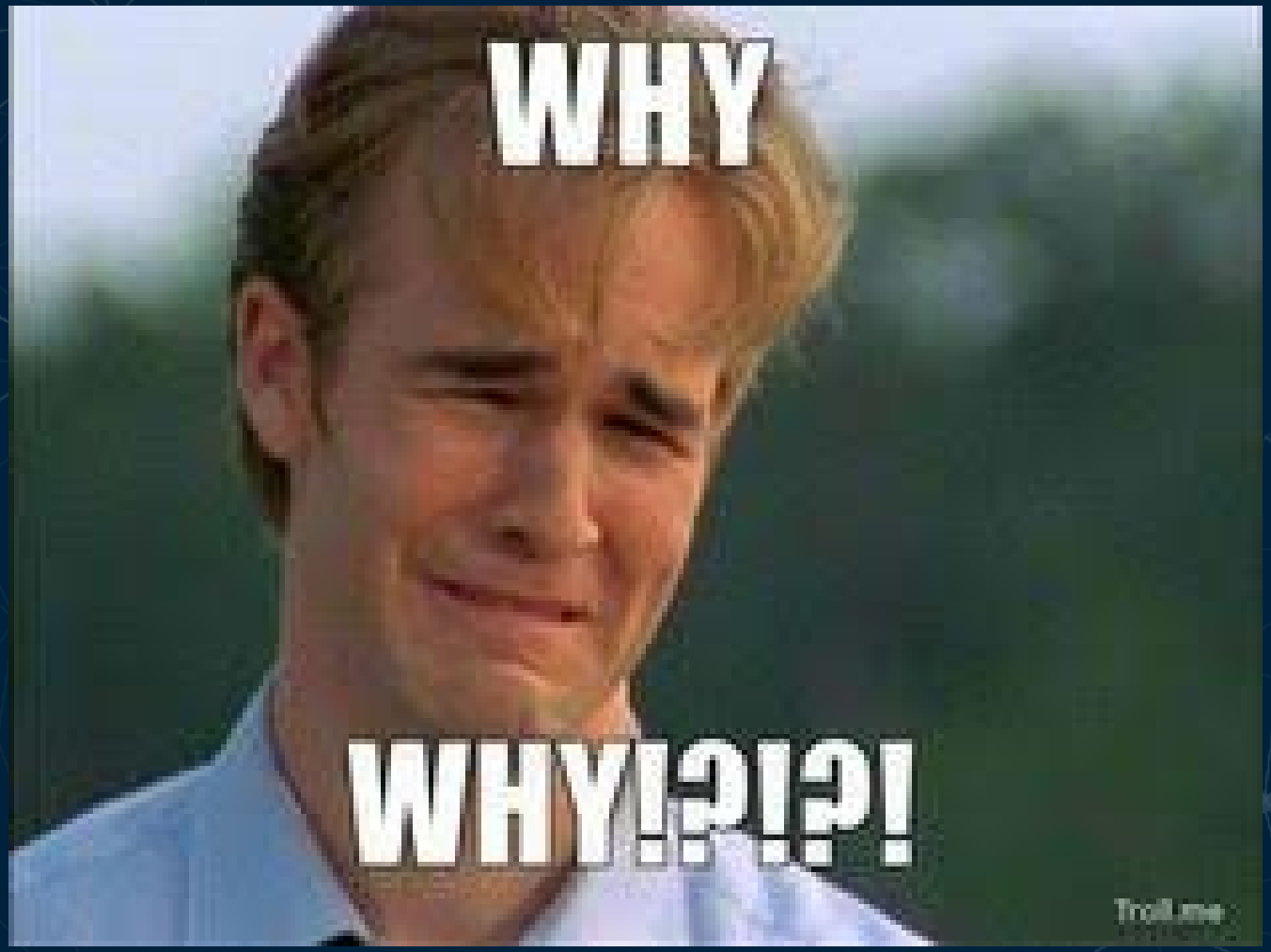
```
docker build -t docker-image .
```

```
docker run -p 4000:80 docker-image
```

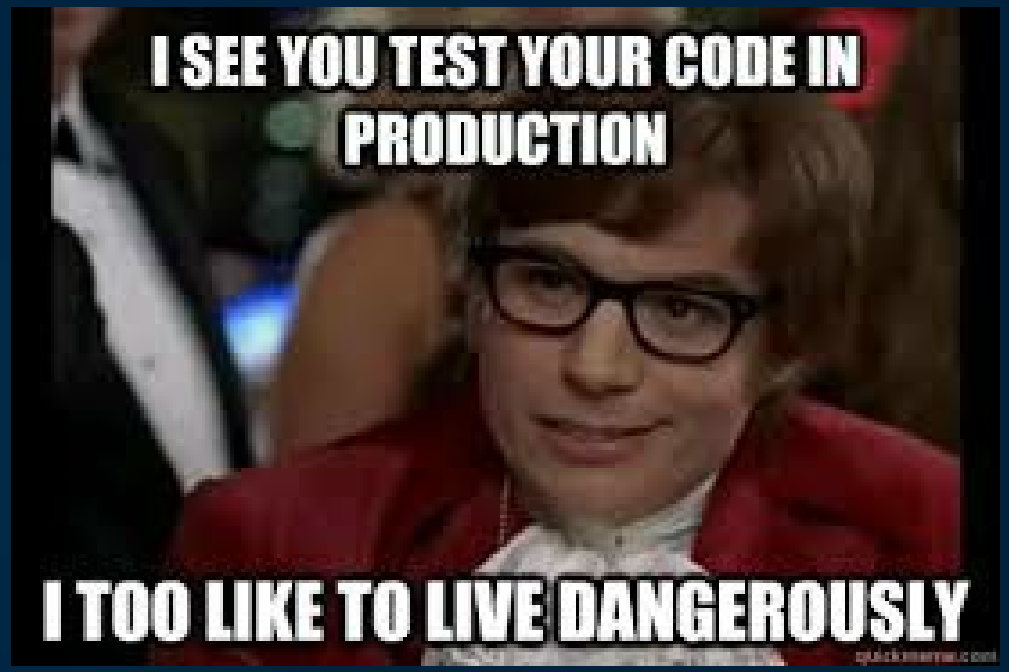
Best practices

- Create small images
  - Performance
  - Security
- Unit test container images

Testing Container Images....



- Like Code, Containers also need to be tested before being deployed to Production.



How... stand on the shoulders of others

- The Container Structure Test Framework.
- Released Earlier this year.
- Being used by Google to test all of their team's released containers

Types of tests supported;

- **Command Tests** - Run a command inside your container image and verify the output or error it produces
- **File Existence Tests** - Check existence of a file in a given location in the image's filesystem
- **File Content Tests** - Check contents and metadata of a file in the filesystem
- A unique **Metadata Test** - to verify configuration and metadata of the container itself

Requirements

- The container structure test binary or Docker image.
- An image to test against.
- A .yaml or .json file with tests to run inside the container.

For Example...

```
commandTests:
  - name: "python installation"
    command: "which"
    args: "python"
    expectedOutput: ["/usr/bin/python"]

fileExistenceTests:
  - name: "requirements file"
    path: "/code/requirements.txt"
    shouldExist: true

fileContentTests:
  - name: "requirements file"
    path: "/code/requirements.txt"
    expectedContents: ['.*flask.*']
```

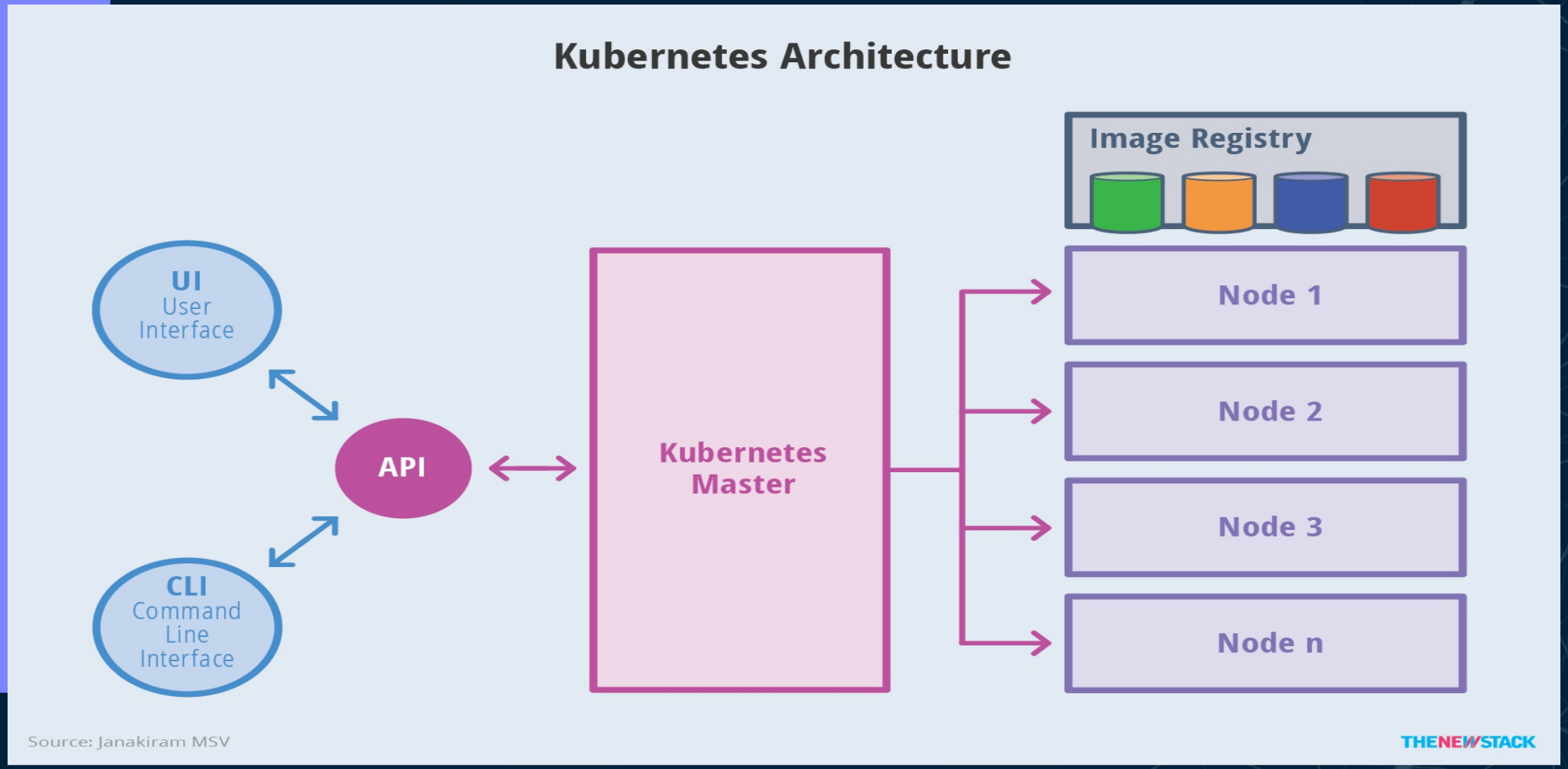
Containers : in summary

- Containers are a packaging for applications.
- Giving us portability, increased productivity and scaling.
- Create a container by creating a Dockerfile, build the image using instructions from this file, and run it to start the container.

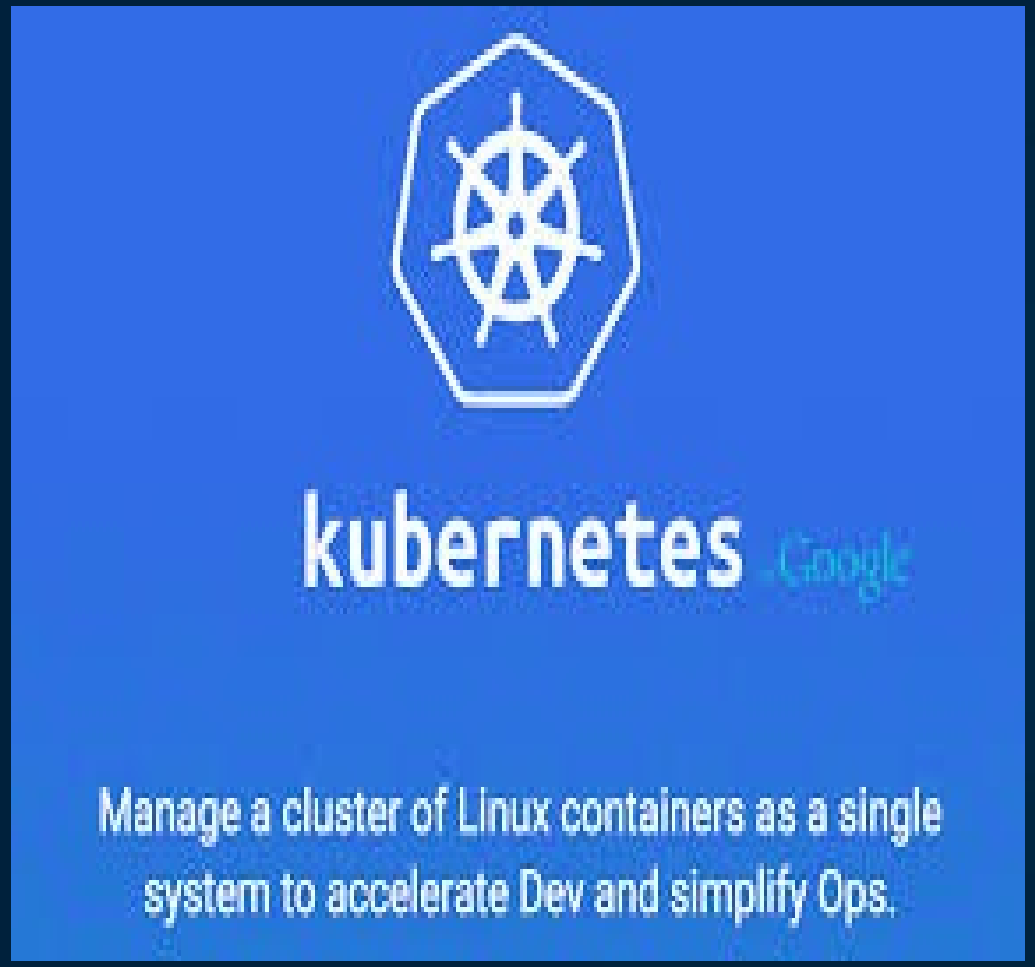
The last mile

- What happens when we have many containers?
- How do we monitor them?
- What about Scaling ?

Kubernetes to the rescue....



Kubernetes



To help manage;

- Deployments
- Monitoring
- Scaling

Deployment

- Specify an image from which to create a container.
- Deployment criteria i.e RAM, CPU, storage

Monitoring

- Health check
- Auto recovery

Scaling

- Cluster scaling.
- Horizontal pod scaling.

Deploy to Kubernetes: Requirements

- A container image for your application.
- A deployment.
- A service to expose your deployment

DEPLOYMENT

Its a yaml file.

kubectl create deployment.yaml

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: k8s_python_sample_code
  namespace: k8s_python_sample_code
spec:
  replicas: 1
  template:
  metadata:
    labels:
    k8s-app: k8s_python_sample_code
  spec:
    containers:
    - name: k8s_python_sample_code
      image: k8s_python_sample_code:0.1
      imagePullPolicy: "IfNotPresent"
      ports:
      - containerPort: 5035
      volumeMounts:
        - mountPath: /app-data
          name: k8s_python_sample_code
   volumes:
       - name: <name of application>
         persistentVolumeClaim:
           claimName: appclaim1
```

Service

Its a yaml file

kubectl create service.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
  k8s-app: k8s_python_sample_code
  name: k8s_python_sample_code
  namespace: k8s_python_sample_code
spec:
  type: NodePort
  ports:
  - port: 5035
  selector:
  k8s-app: k8s_python_sample_code
```

Kubernetes : in summary

- Kubernetes gives us;
  - Scaling
  - Monitoring
  - Zero upgrade down times

State in Containers

- Files
- Volumes
- Persistent Volumes

SCOPE

Storage	Life time
Container File system	container
Volume	Pod
Persistent Volume	Cluster

Steps in using persistent Storage

- Create a volume.
- Create a persistent volume claim to claim resources from a persistent volume.
- Mount the claim on the deployment.

Kubernetes persistent volumes

- Provisioned statically or Dynamically.
- Created with a specific filesystem.
- Has a specific size
- Has unique characteristics such as volume IDs and a name.

Create a GCE volume: using GC client

```
gcloud compute disks create --size=500GB
--zone=us-central1-a pg-data-disk
```

Create a GCE volume: using config

```
#volume.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: dump
  namespace: web
spec:
  capacity:
    storage: 100Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  storageClassName: slow
  gcePersistentDisk:
    pdName: "data-dump-disk"
    fsType: "ext4"
```

Create the image

```
kubectl apply -f volume.yaml
```

Create a claim

```
#claim.yaml

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: dump
  namespace: web
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: slow
```

Create the claim

```
kubectl apply -f claim.yaml
```

Mount claim on the deployment

```
...
  volumeMounts:
    - name: dump
      mountPath: /loklak_server/data
volumes:
  - name: dump
    persistentVolumeClaim:
      claimName: dump
```

Full sample

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: k8s_python_sample_code
  namespace: k8s_python_sample_code
spec:
  replicas: 1
  template:
  metadata:
    labels:
    k8s-app: k8s_python_sample_code
  spec:
    containers:
    - name: k8s_python_sample_code
      image: k8s_python_sample_code:0.1
      imagePullPolicy: "IfNotPresent"
      ports:
      - containerPort: 5035
      volumeMounts:
        - mountPath: /app-data
          name: k8s_python_sample_code
   volumes:
       - name: <name of application>
         persistentVolumeClaim:
           claimName: appclaim1
```

Summary

- Containers can have state.
- Use Persistent storage to achieve this state.
  - Create a volume.
  - Create a claim for the volume.
  - Mount the volume on the container.

References

- Https://portworx.com/tutorial-kubernetes-persistent-volumes/
- https://matthewpalmer.net/kubernetes-app-developer/articles/kubernetes-volumes-example-nfs-persistent-volume.html
- Https://blog.fossasia.org/tag/persistent-volume-claim/
- Http://kubernetesbyexample.com/volumes/
- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux_atomic_host/7/html/getting_started_with_kubernetes/get_started_provisioning_storage_in_kubernetes

Questions!!