



Rise of
Containers



Developing
Containers



Container
Tools

APIs



Microservices

Polyglot

Distillation

Fin



Kubernetes Challenge: Assembling a Multi-Component Application

- A simple voting application
- Run commands to start up a multi-microservice app

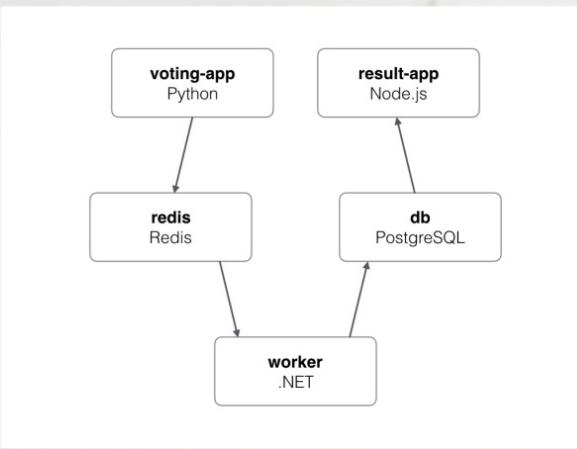


SCENARIO

Kubernetes Skills: Assembling a Multi-Component Application

Challenge your skills assembling a multi-component application containing web interfaces, microservices,...

learning.oreilly.com/scenarios
Search for "**assembling**"



Container

A standard unit of software

that packages up code and
all its dependencies

so the application runs
quickly and reliably

from one computing
environment to another.

- Docker



Container

A standard unit of software
that packages up code and
all its dependencies
so the application runs
quickly and reliably
from one computing
environment to another.

- Docker

```
FROM bowl:1.0.0
RUN apt-get install \
h2o \
pebbles \
plant
COPY fish /bin
CMD ["fish", "swim"]
```

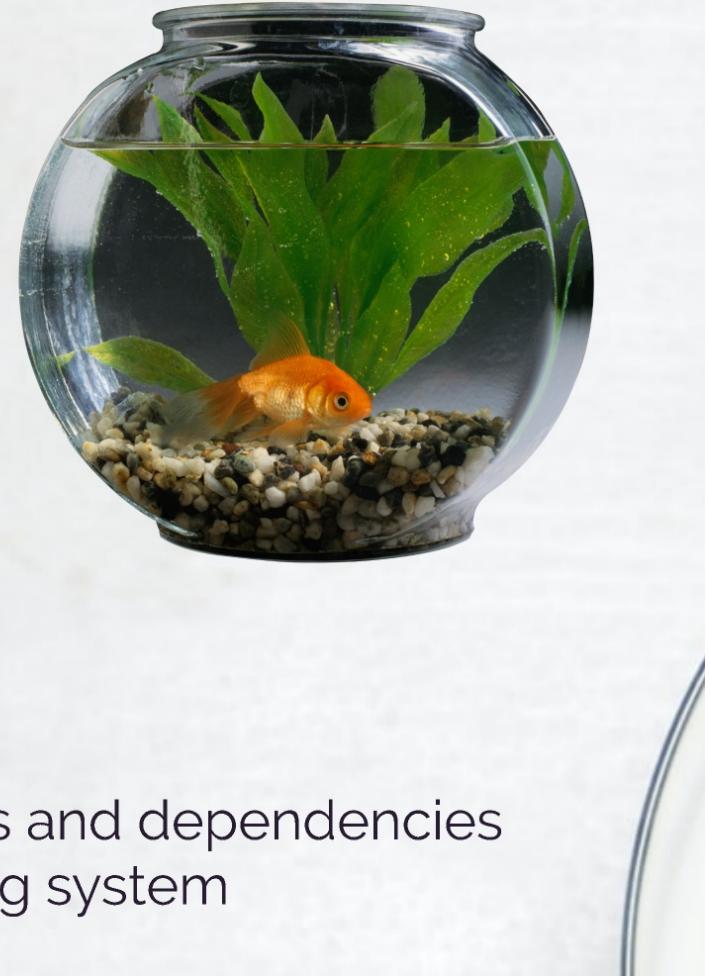


Container

A standard unit of software
that packages up code and
all its dependencies
so the application runs
quickly and reliably
from one computing
environment to another.

- Docker

```
FROM bowl:1.0.0  
  
RUN apt-get install \  
    h2o \  
    pebbles \  
    plant  
  
COPY fish /bin  
  
CMD ["fish", "swim"]
```



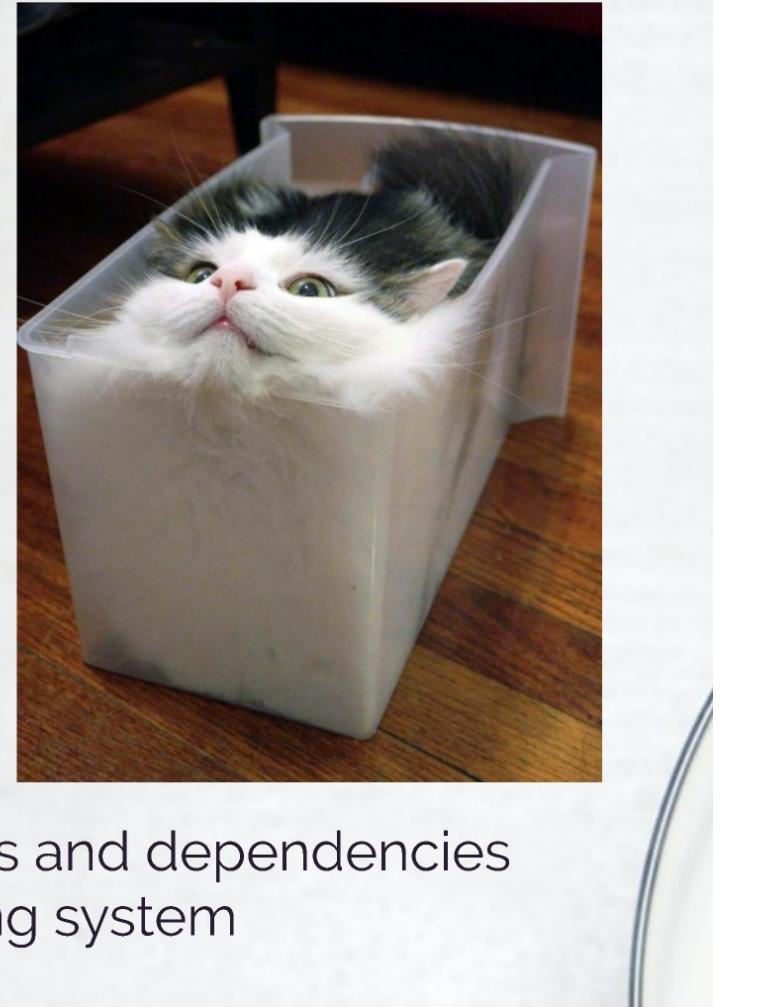
- Design time: Cohesive full stack of apps and dependencies
- Runtime: Boxed process on an operating system

Container

A standard unit of software
that packages up code and
all its dependencies
so the application runs
quickly and reliably
from one computing
environment to another.

- Docker

```
FROM bowl:1.0.0  
  
RUN apt-get install \  
    h2o \  
    pebbles \  
    plant  
  
COPY fish /bin  
  
CMD ["fish", "swim"]
```



- Design time: Cohesive full stack of apps and dependencies
- Runtime: Boxed process on an operating system

Why Containers?

Why would you make your life more complicated?

Polyglot

Disposable

Separation of concerns

Control your infrastructure

Smaller, independent units

Fast scaling

Why Containers?

Why would you make your life more complicated?

Polyglot

Disposable

Separation of concerns

Control your infrastructure

Smaller, independent units

Fast scaling

Replication:

Production abhors singletons

Why Containers?

New reality for service delivery

Why would you make your life more complicated?

- Polyglot
- Disposable
- Separation of concerns
- Control your infrastructure
- Smaller, independent units
- Fast scaling

- Replication:
Production abhors singletons

Why Containers?

Why would you make your life more complicated?

Polyglot
Disposable

Separation of concerns
Control your infrastructure
Smaller, independent units
Fast scaling

Replication:
Production abhors singletons

New reality for service delivery



<https://12factor.net>

Why Containers?

Why would you make your life more complicated?

- Polyglot
- Disposable
- Separation of concerns
- Control your infrastructure
- Smaller, independent units
- Fast scaling

Replication:
Production abhors singletons

New reality for service delivery



<https://12factor.net>



Layers



Infrastructure as code

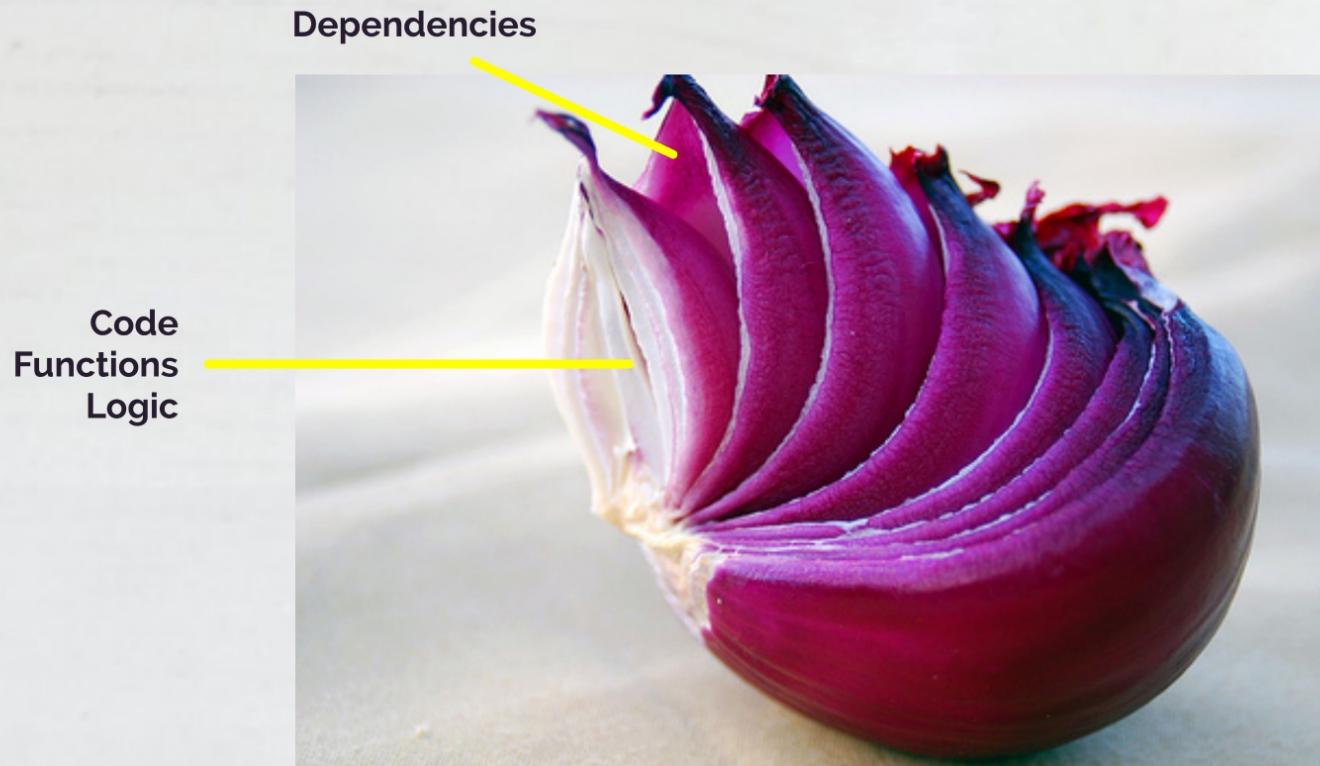
Layers

Code
Functions
Logic



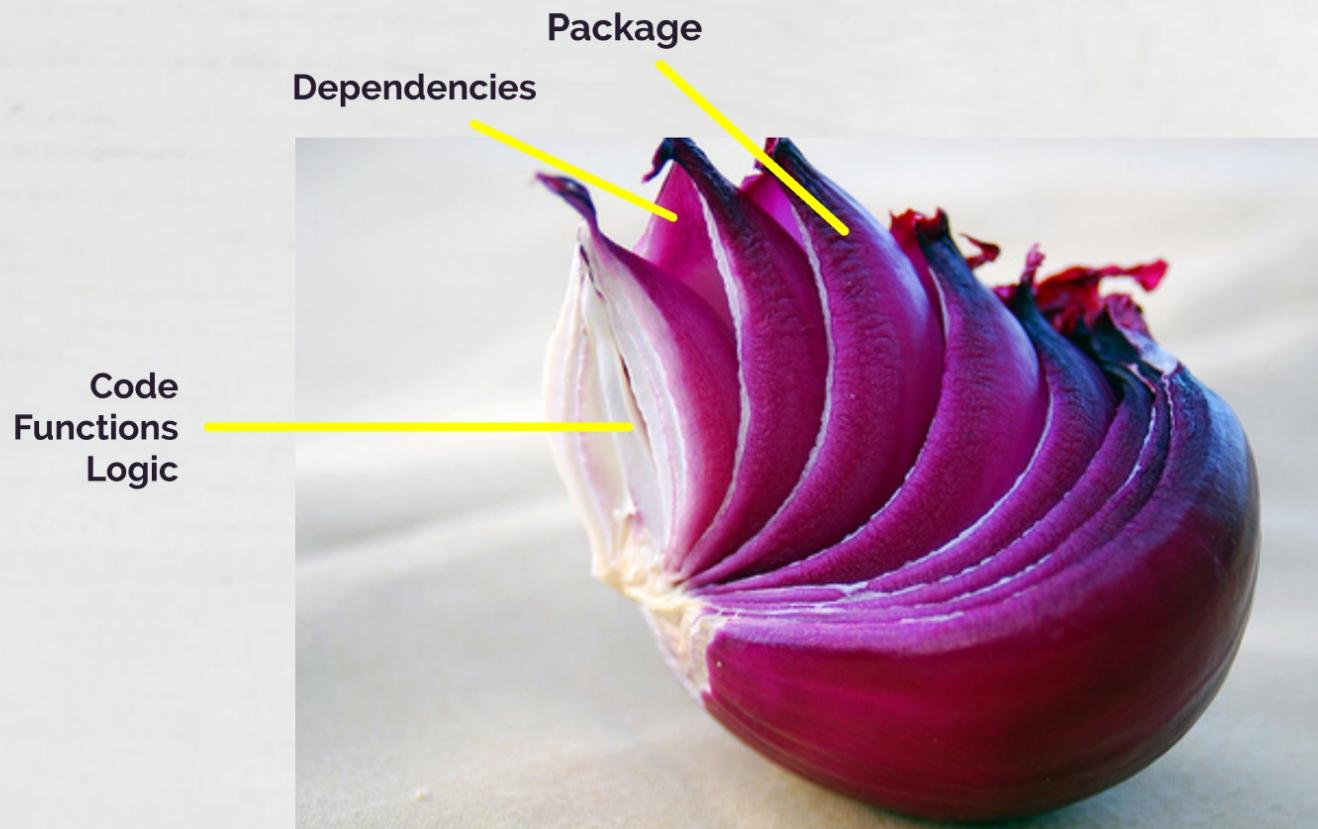
Infrastructure as code

Layers



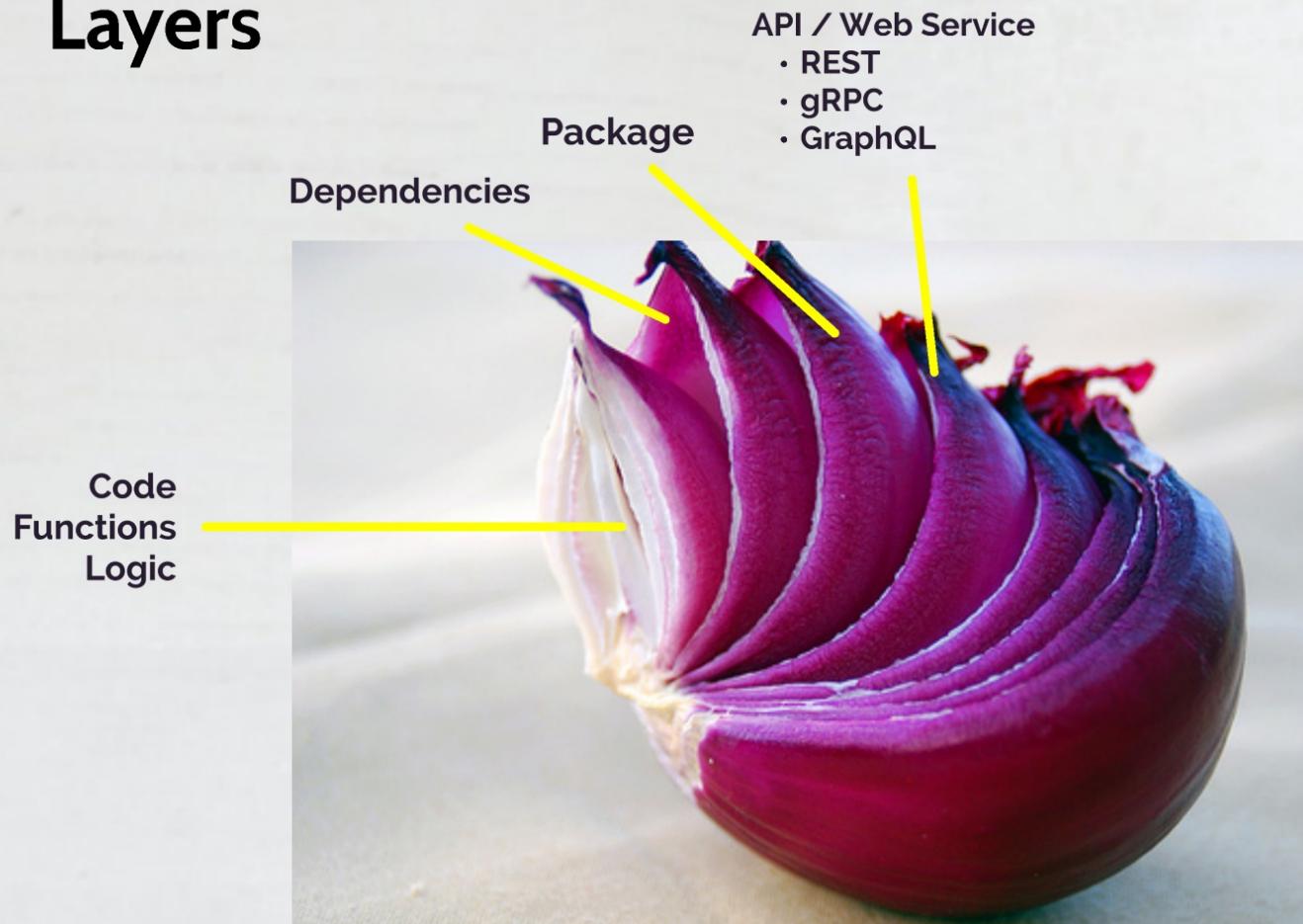
Infrastructure as code

Layers



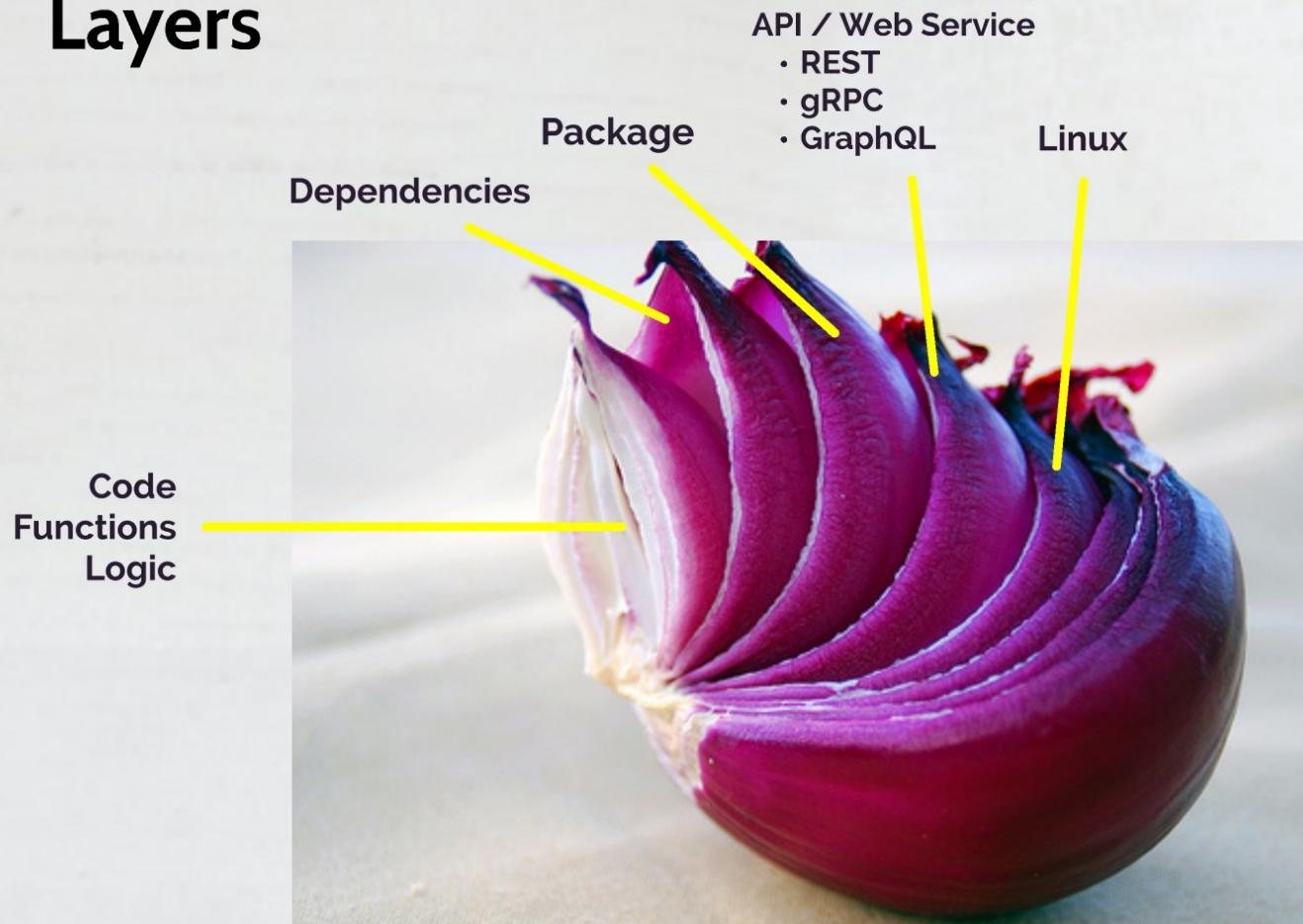
Infrastructure as code

Layers



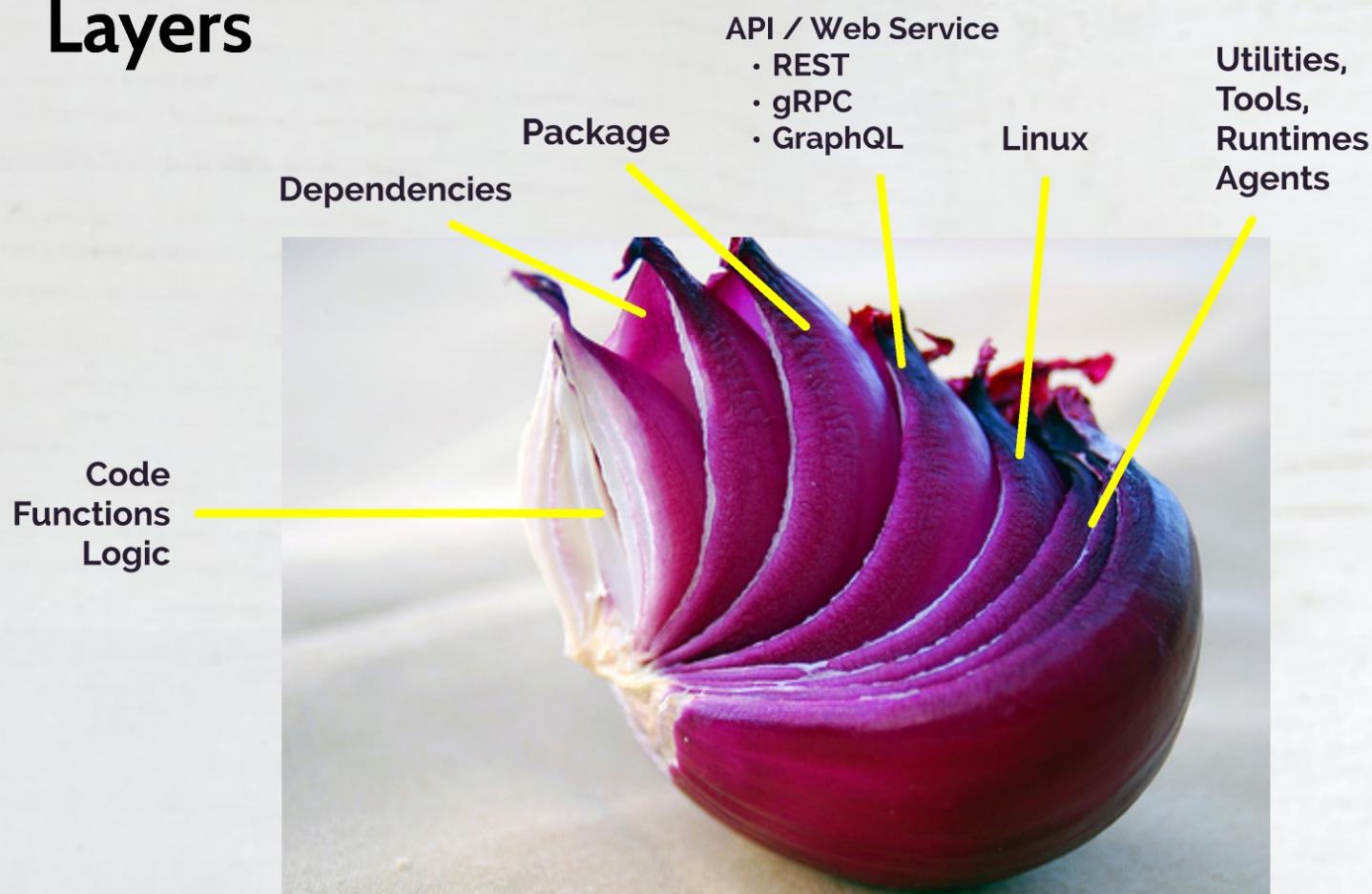
Infrastructure as code

Layers



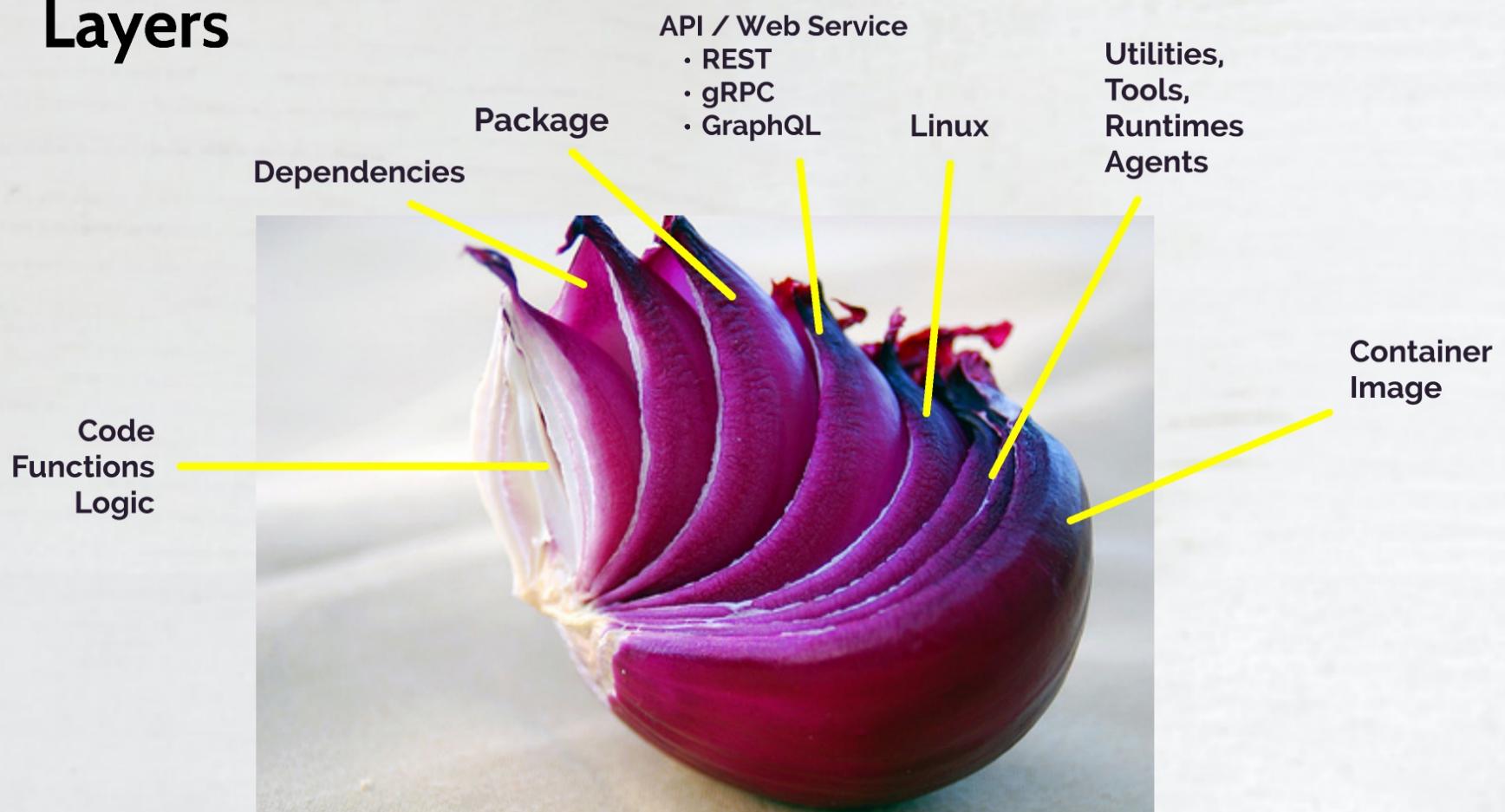
Infrastructure as code

Layers



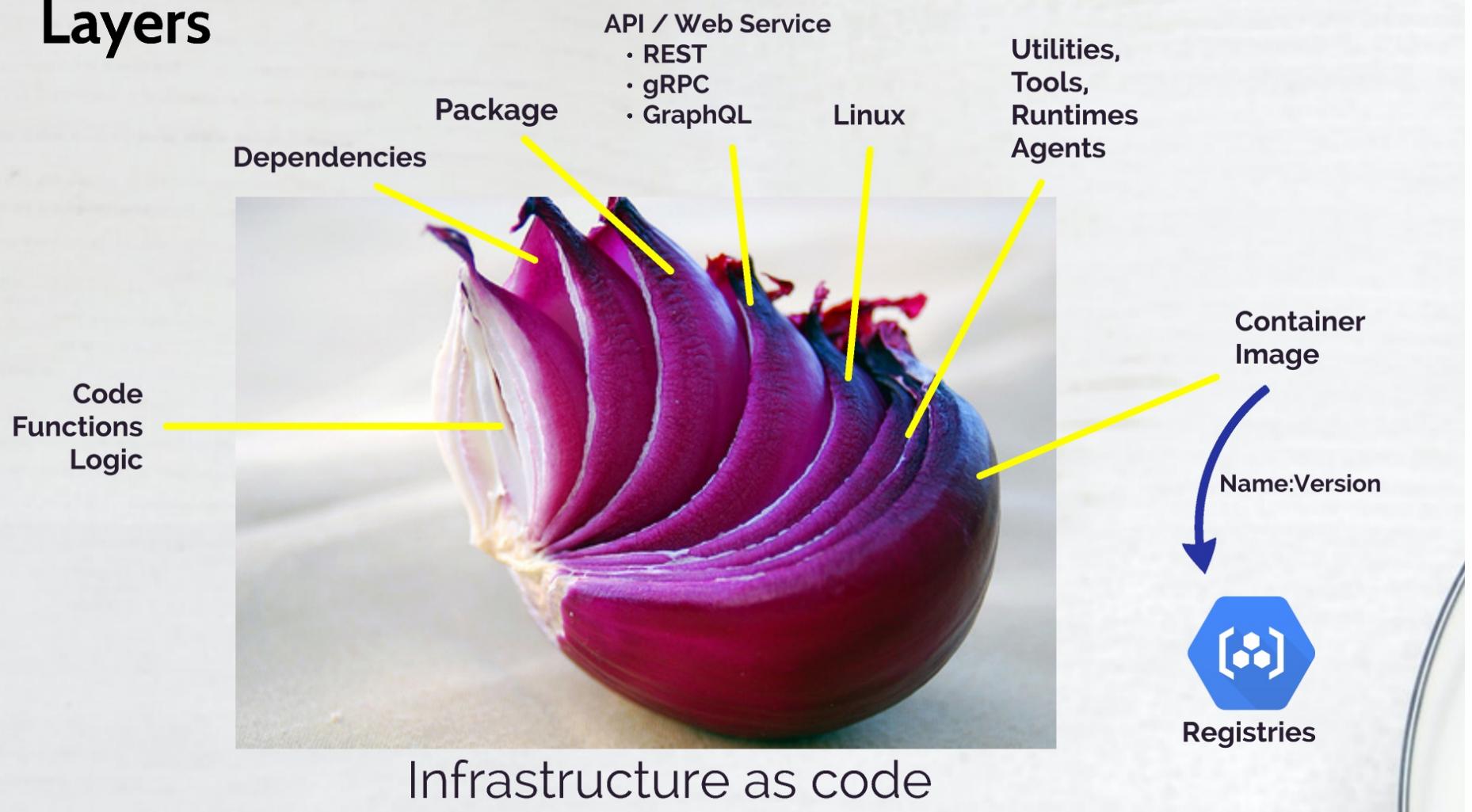
Infrastructure as code

Layers

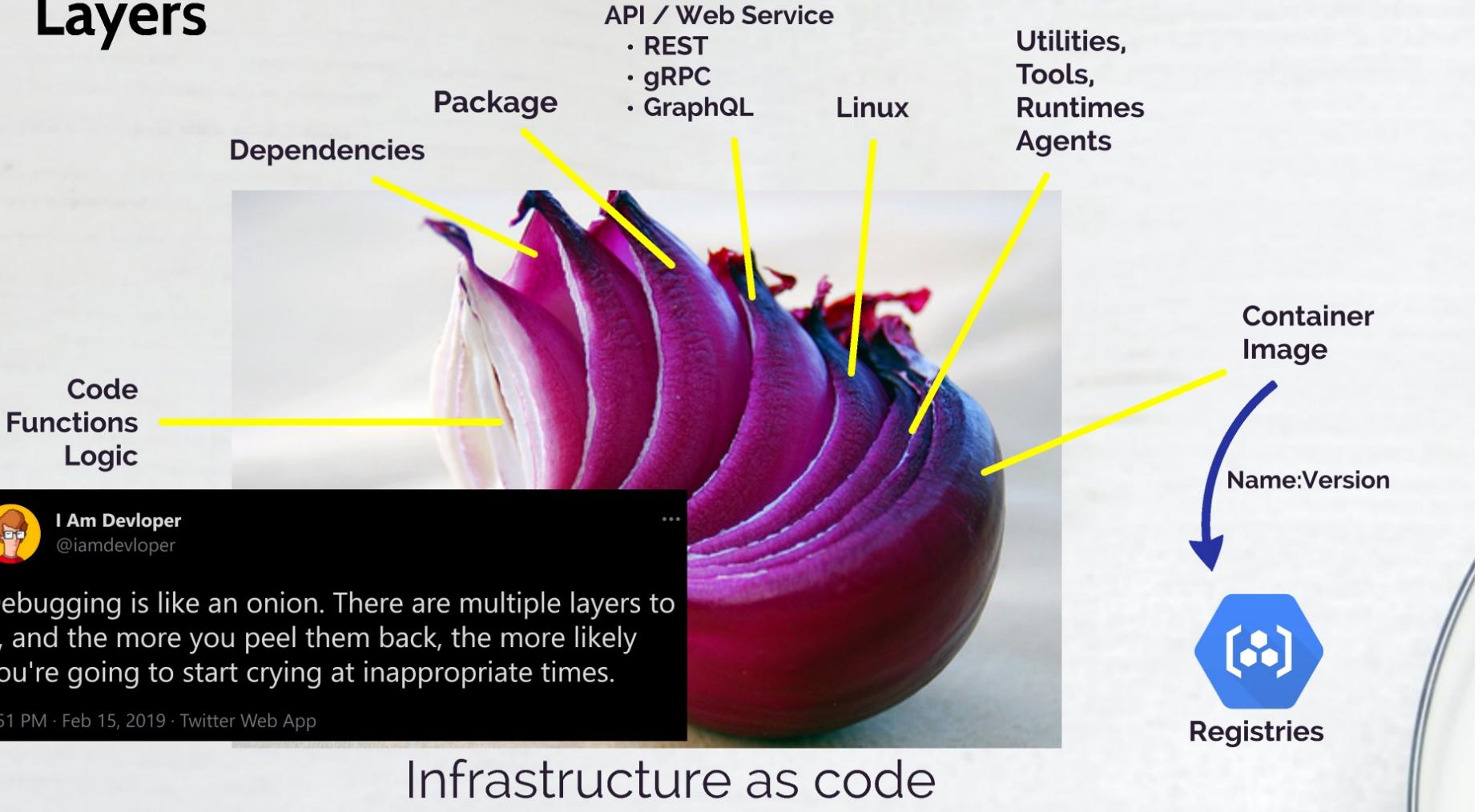


Infrastructure as code

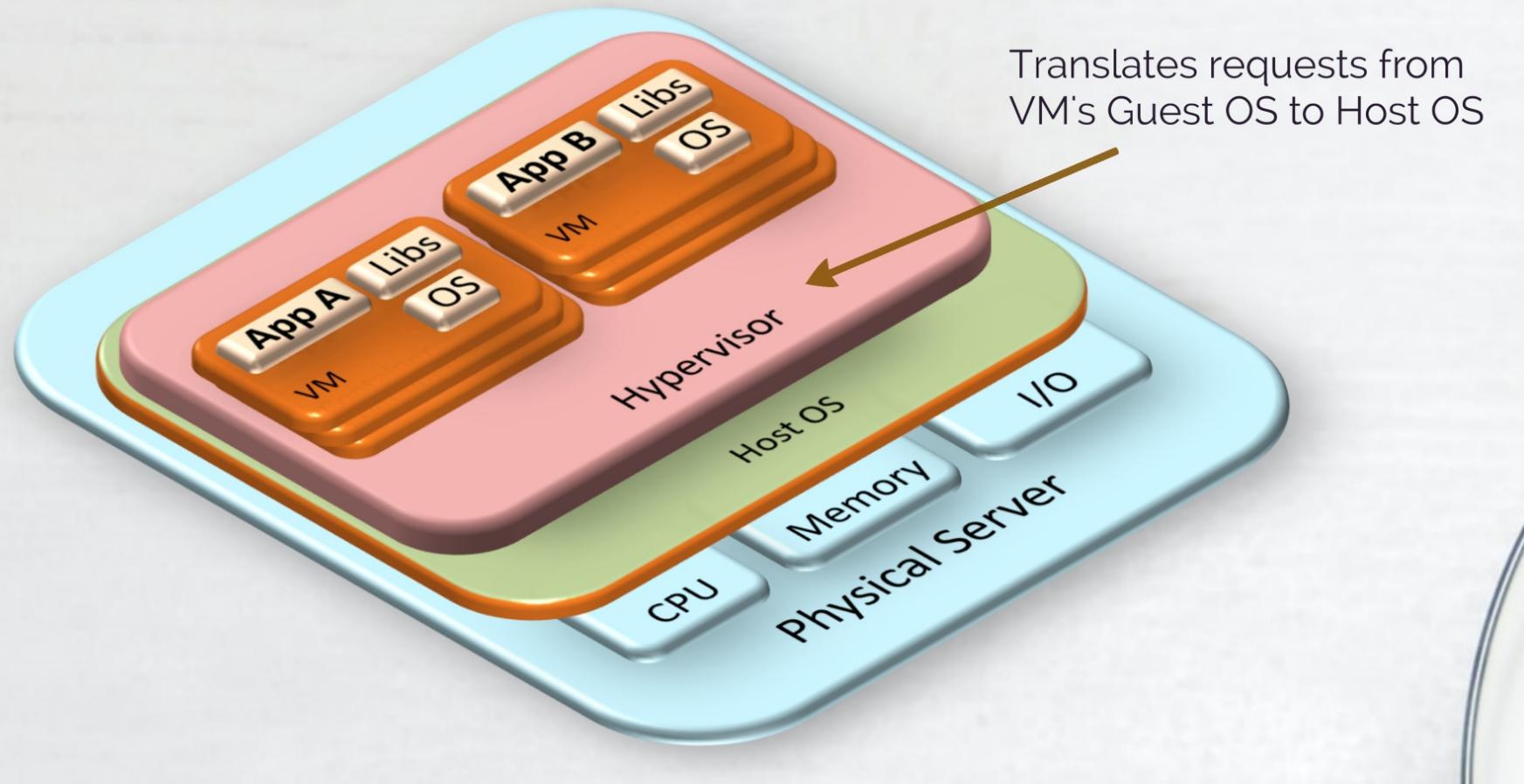
Layers



Layers



Before Containers, We had Virtual Machines



Container Architecture

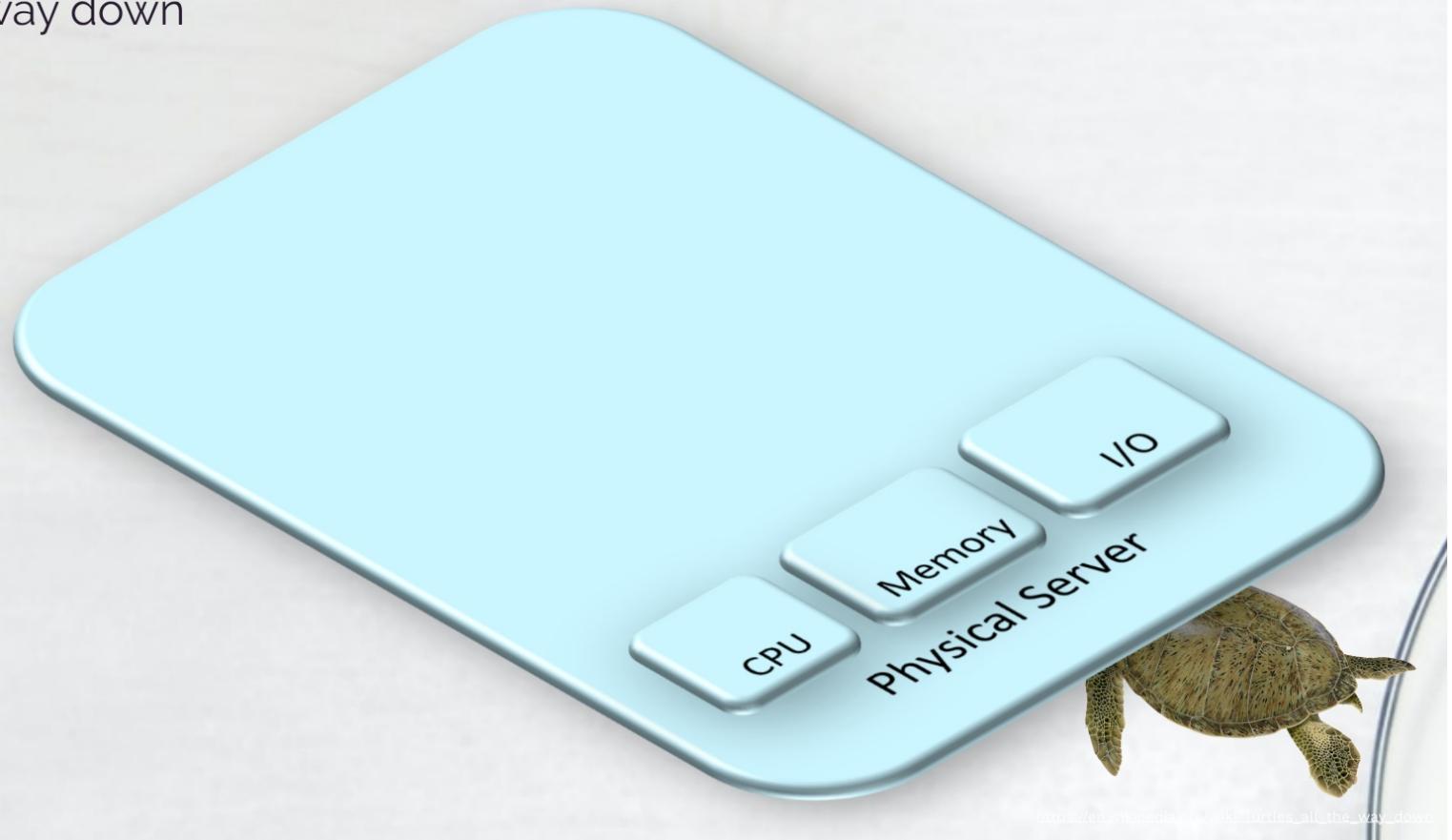
Turtles all the way down



https://zenorocha.medium.com/turtles_all_the_way_down

Container Architecture

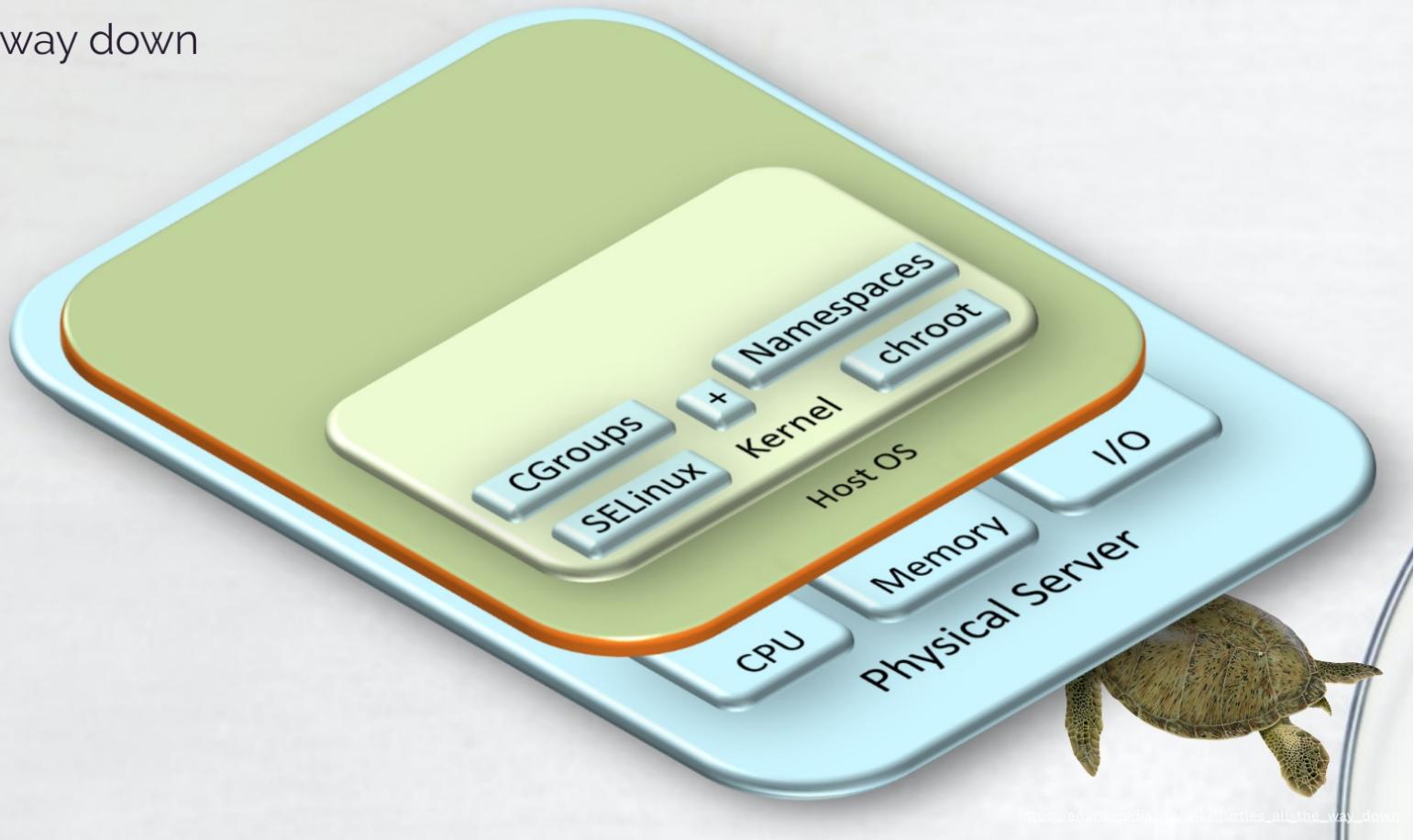
Turtles all the way down



https://zenorocha.medium.com/turtles_all_the_way_down

Container Architecture

Turtles all the way down

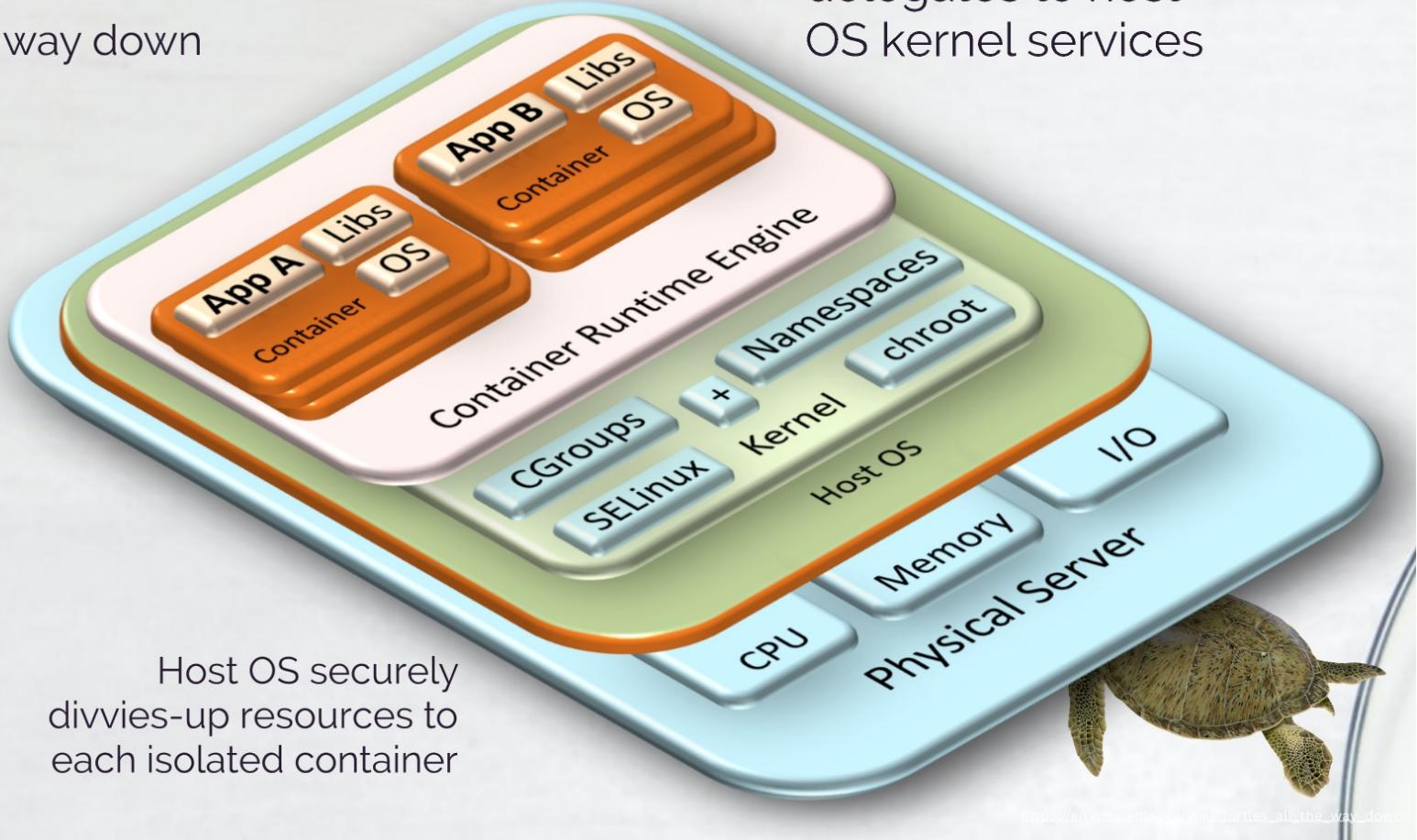


https://zenorocha.medium.com/turtles_all_the_way_down

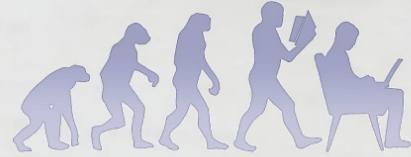
Container Architecture

Turtles all the way down

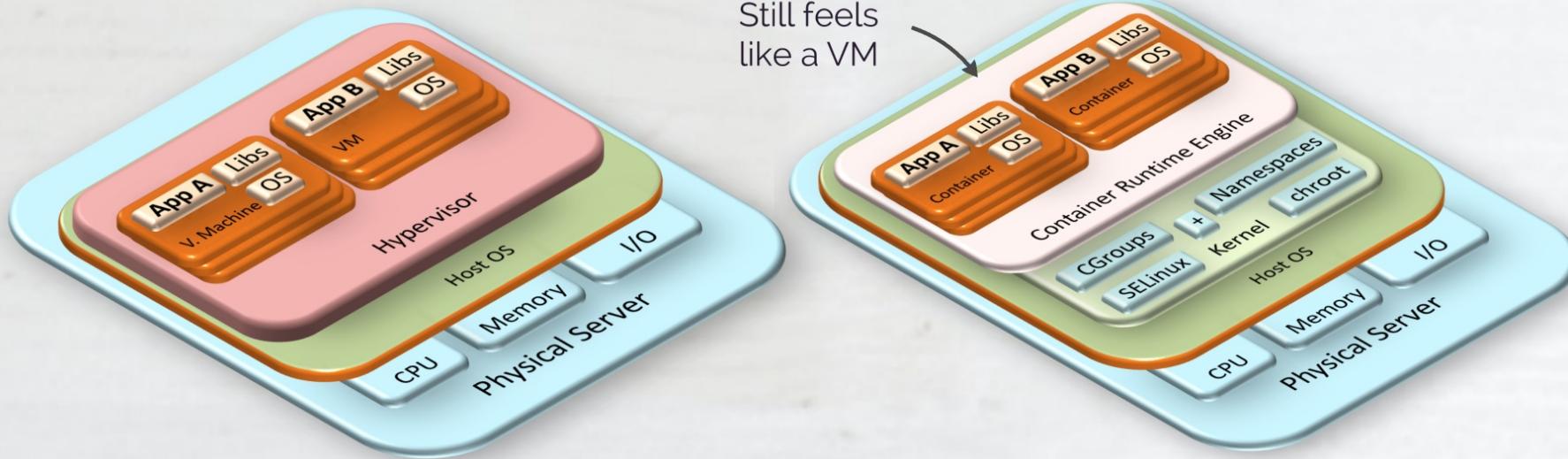
OS in container
delegates to host
OS kernel services



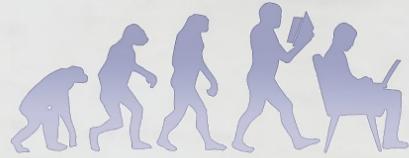
Evolution: VMs



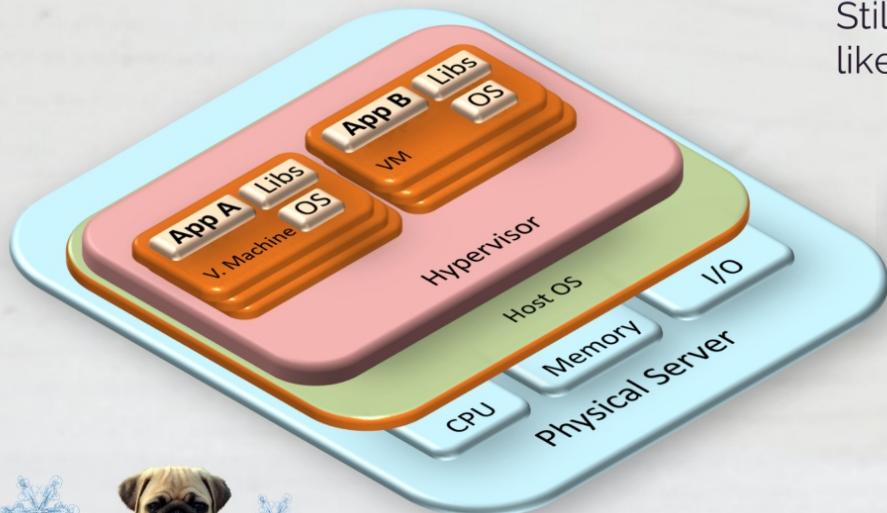
to Container Runtimes



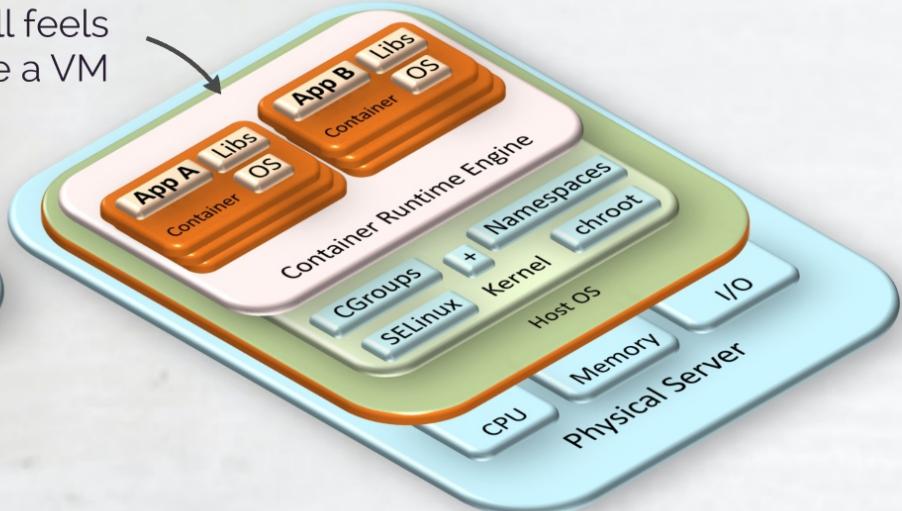
Evolution: VMs



to Container Runtimes



Still feels
like a VM

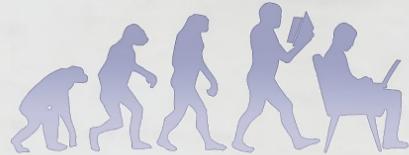


Unique servers,
snowflakes GBs -
minutes to start

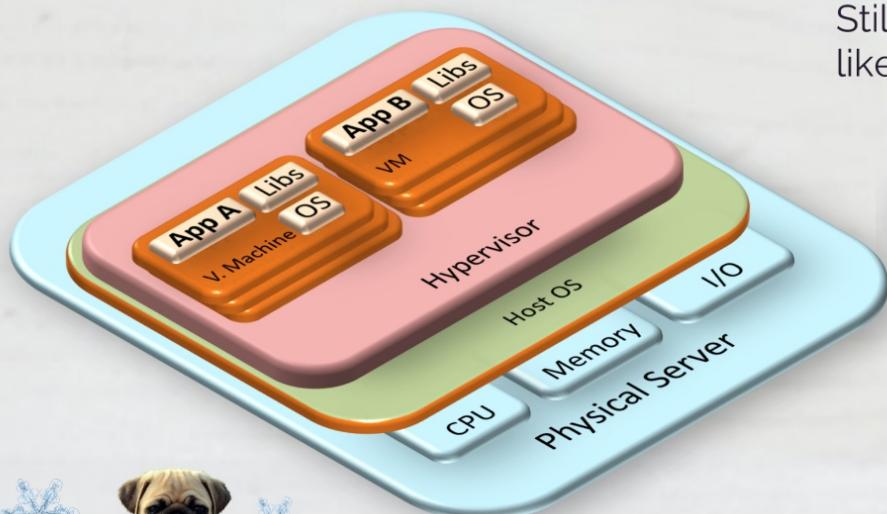
<https://martinfowler.com/bliki/SnowflakeServer.html>



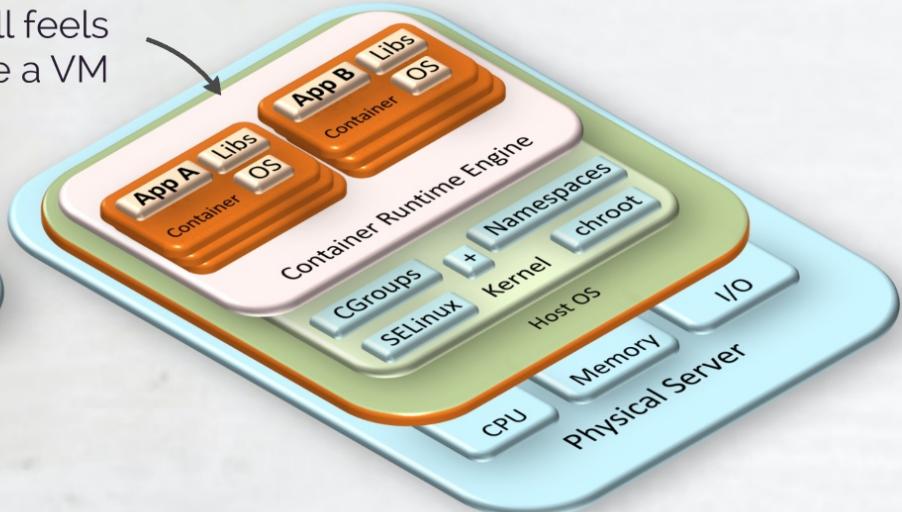
Evolution: VMs



to Container Runtimes



Still feels like a VM



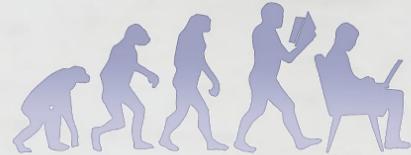
Unique servers,
snowflakes GBs -
minutes to start

<https://martinfowler.com/bliki/SnowflakeServer.html>

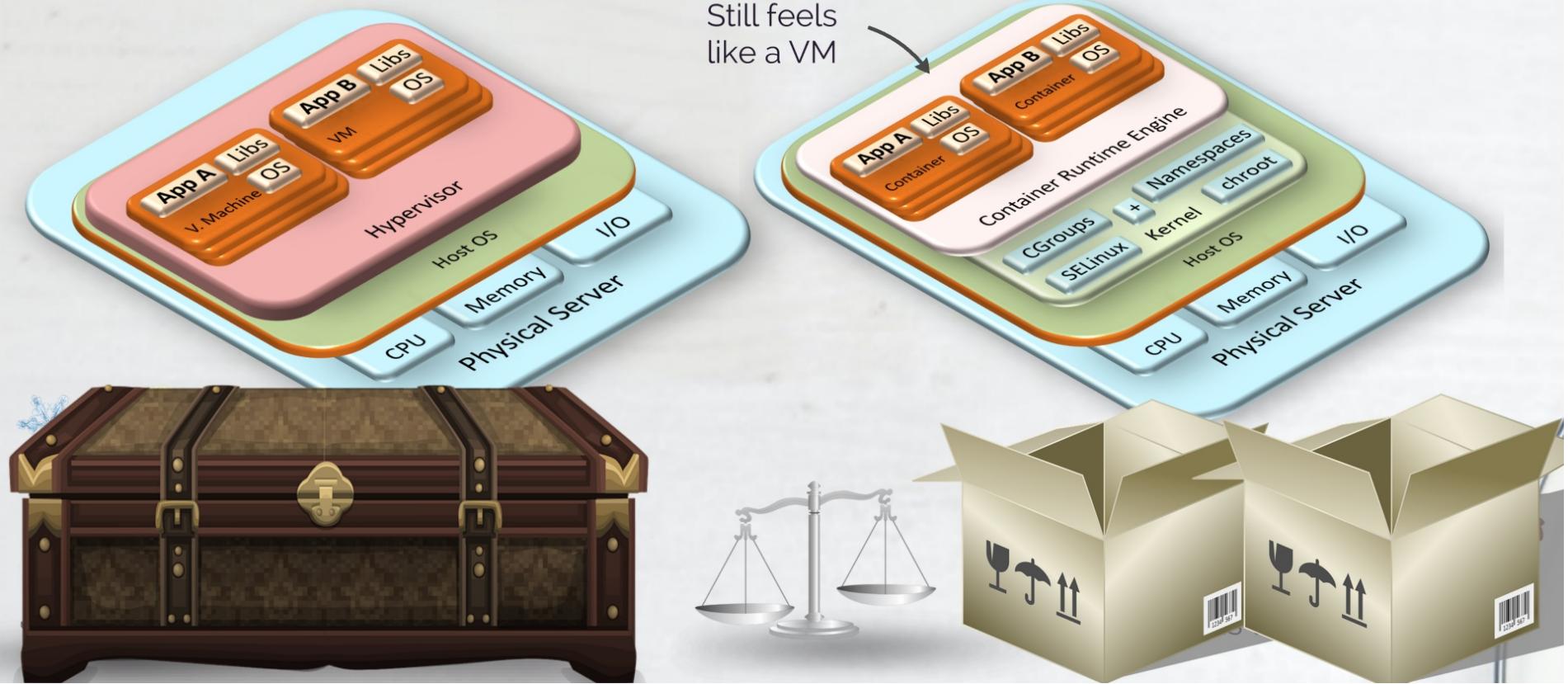


Disposable, ephemeral IPs
MBs - seconds to start

Evolution: VMs



to Container Runtimes



Kernel Sleight of Hand for Containers



Kernel Sleight of Hand for Containers

chroot

Limit where
you run



Kernel Sleight of Hand for Containers

chroot

Limit where
you run



Security-Enhanced Linux

Limit what
you can access

Kernel Sleight of Hand for Containers

chroot

Limit where
you run



Control Groups

Limit what you
can use

Security-Enhanced Linux

Limit what
you can access

Kernel Sleight of Hand for Containers

chroot

Limit where
you run



Namespaces

Limit what
you can see

Control Groups

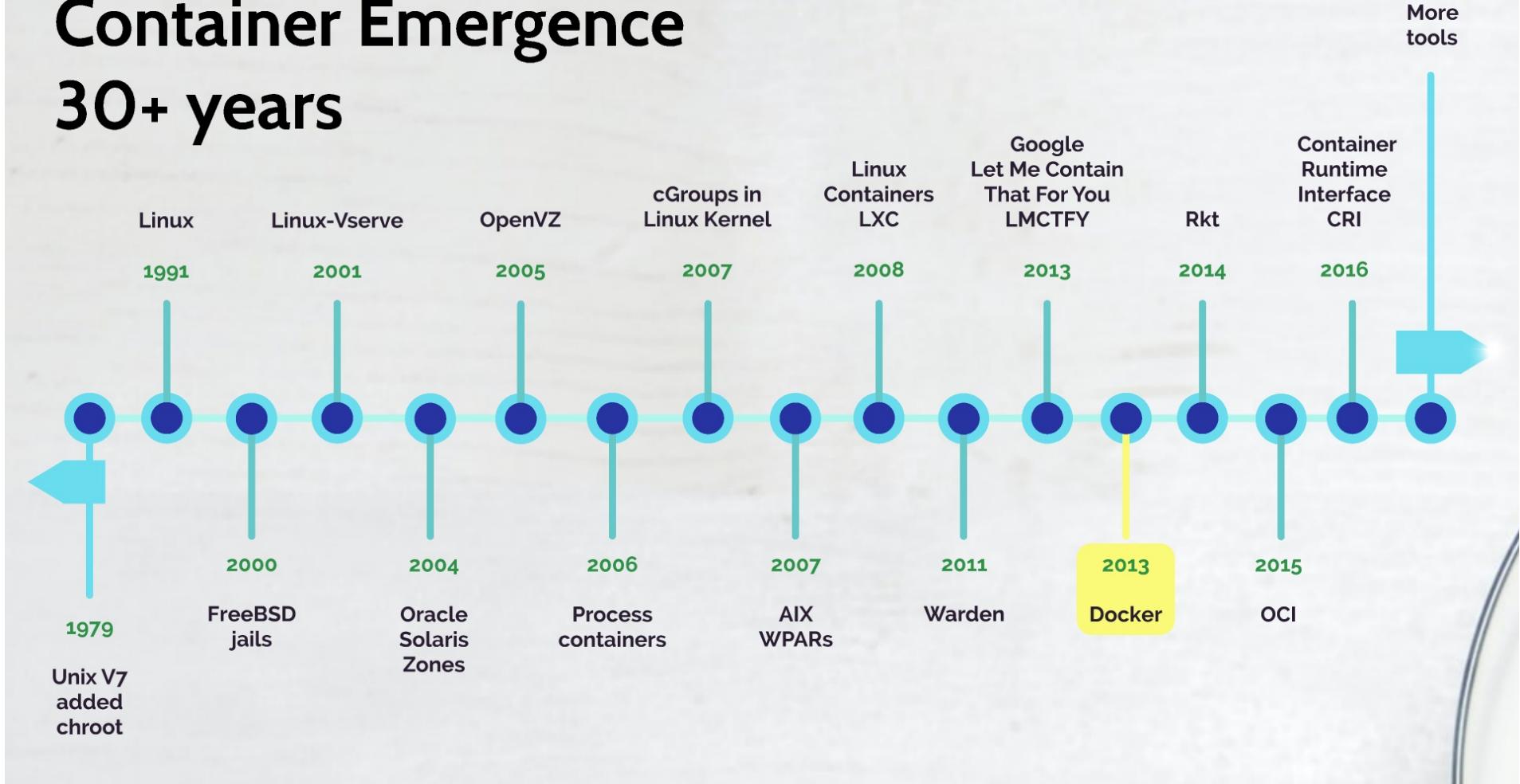
Limit what you
can use

Security-Enhanced Linux

Limit what
you can access

Container Emergence

30+ years



Terminology

A few common terms

Container Image

A file of bytes following the container image format that defines the container. Stored in a repository. Often composed of multiple layers.

Image Layer

Container images are typically many layers of images. Each layer defines distinct and reusable container definitions. FROM, COPY, ADD and RUN add new layers

Container Engine

Process on the Host OS that accepts requests to pull images, calls the container runtime and cleans up after containers terminate.

Registry

Library or datastore of multiple repositories. Hosts and manages the repo files and manifests (add, remove, list, security).

Repository

Listing of one or more versions of a container along with its images layers and manifests. Follows established image manifest schema.

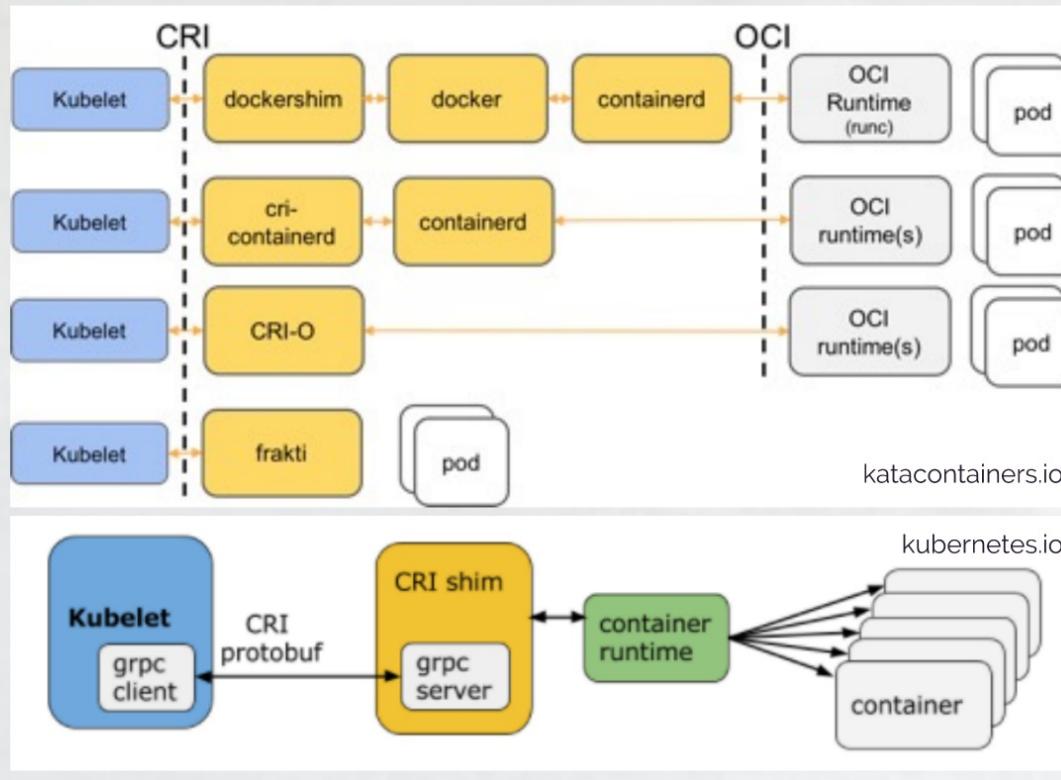
Tag

A label that uniquely identifies multiple container images in a single repository.

Base Image

Most containers build upon base container images layers, and build more layers on top of them. Encourages reuse.

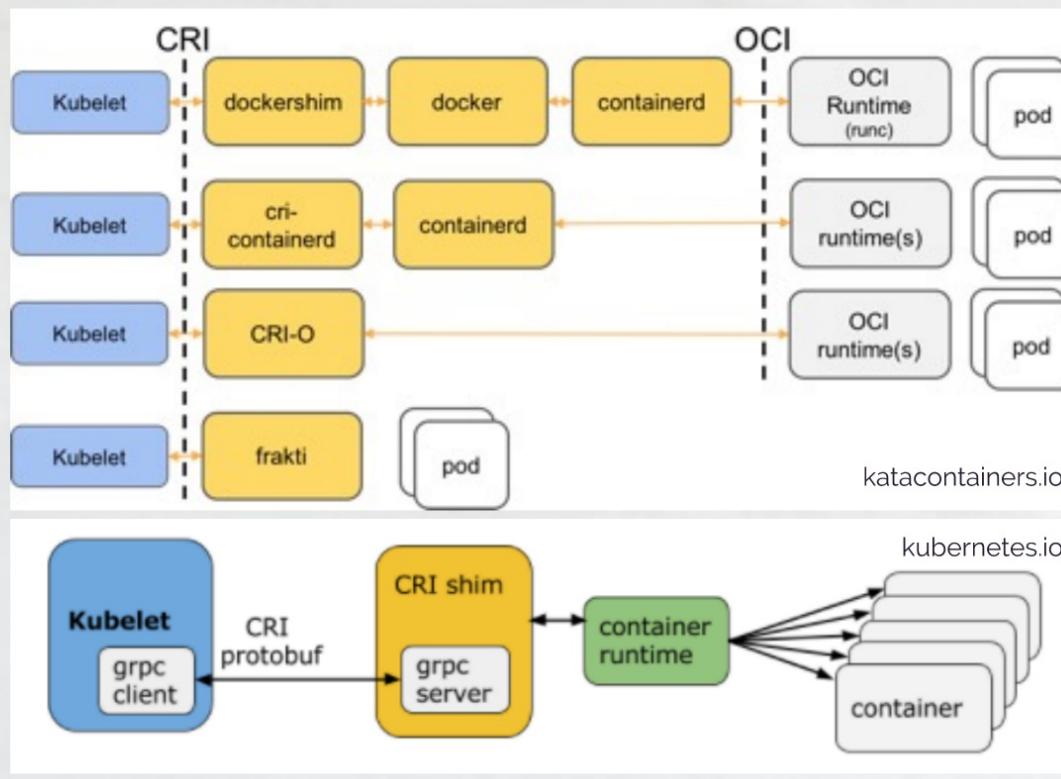
CRI and OCI



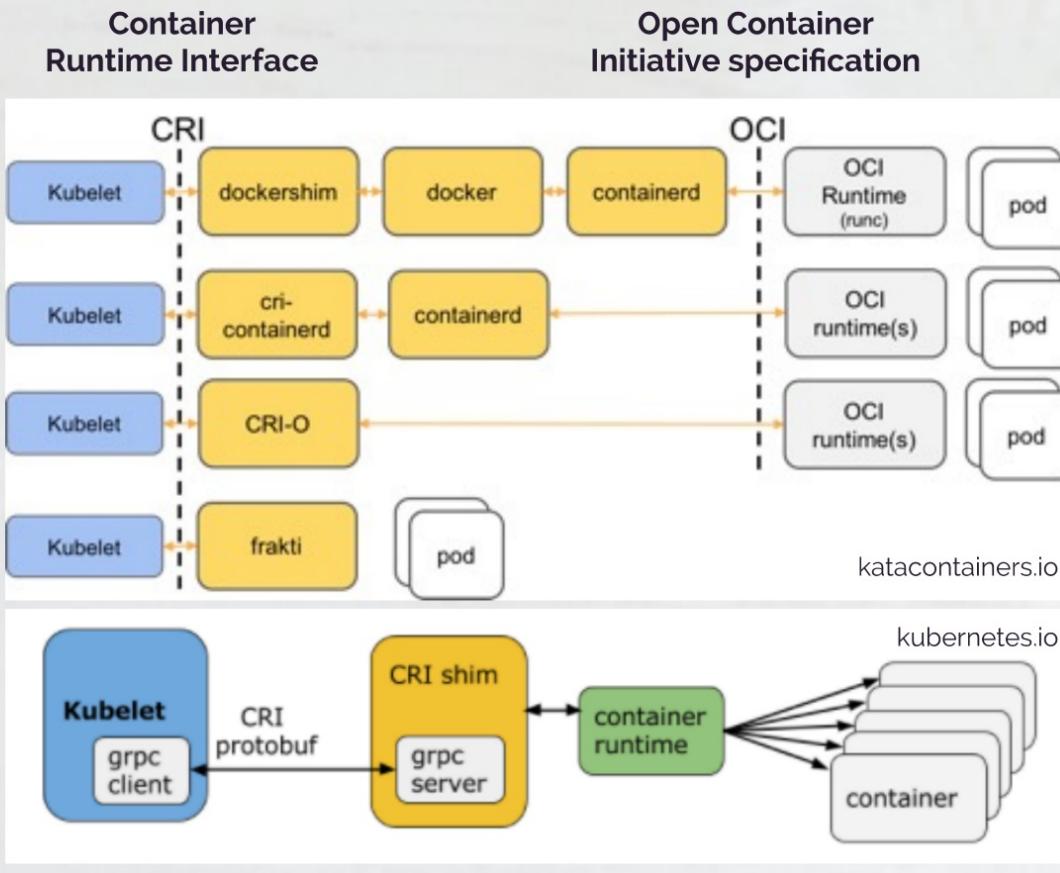
CRI and OCI



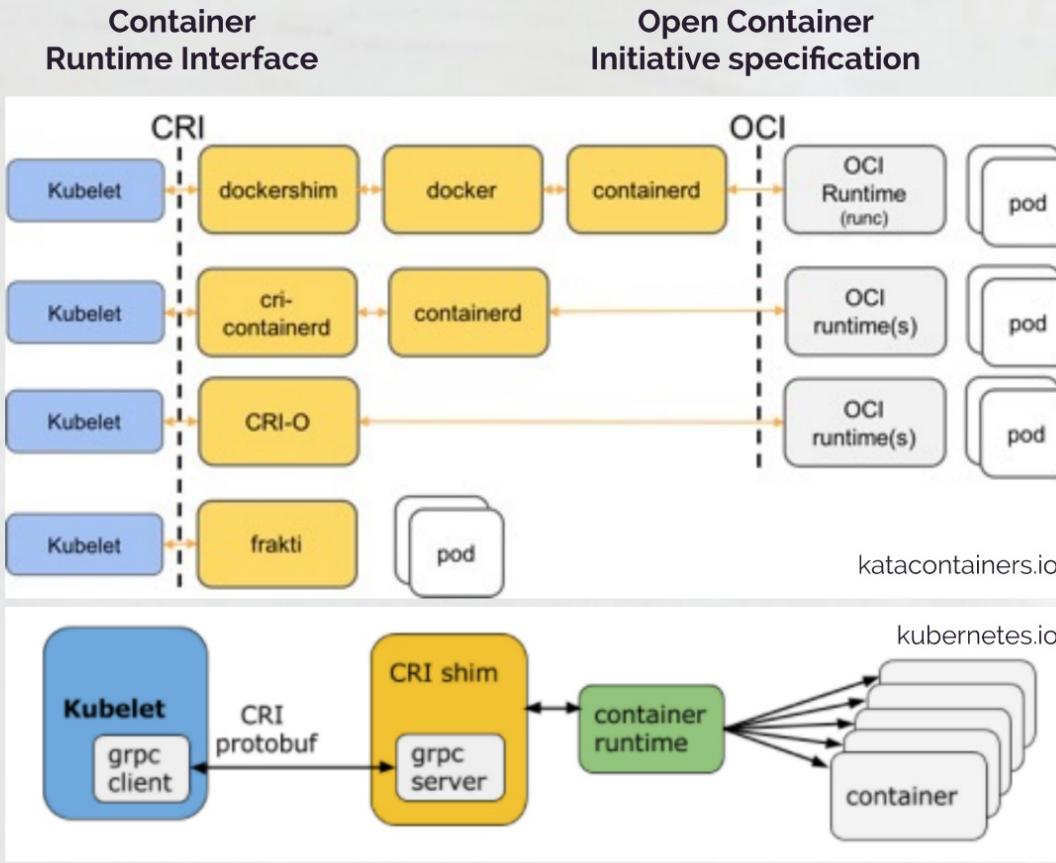
Container
Runtime Interface



CRI and OCI



CRI and OCI



OCI container runtimes with Kubelet CRI

- Docker (nix Q4 2020)
- cri-o (OpenShift)
- **cri-containerd** (default)
 - runc
 - gVisor
 - Kata containers
 - Firecracker
 - Nabla containers
- crun (c based)
- Rktlet (CRI for Rkt)
- rktnetes
- Frakti
- Singularity
- * Virtual Kubelet

<https://landscape.cncf.io/category=container-runtime&format=card-mode&grouping=category>



Rise of Containers



APIs



Polyglot

Distillation

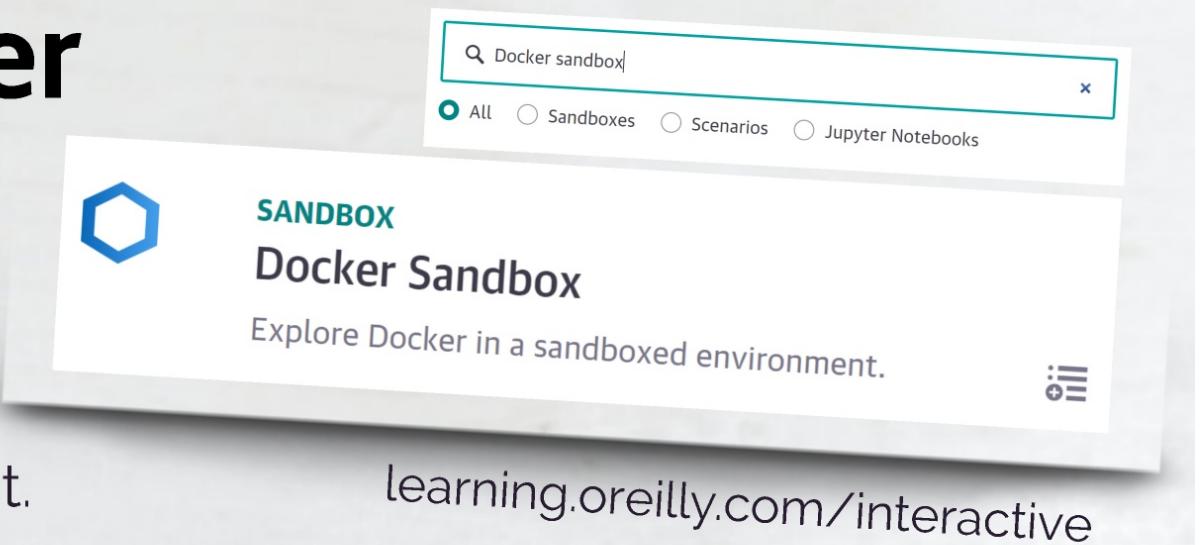
Fin



Hello Docker

Docker CLI orientation

Sandboxes for safe,
fast, and consistent
places to experiment.



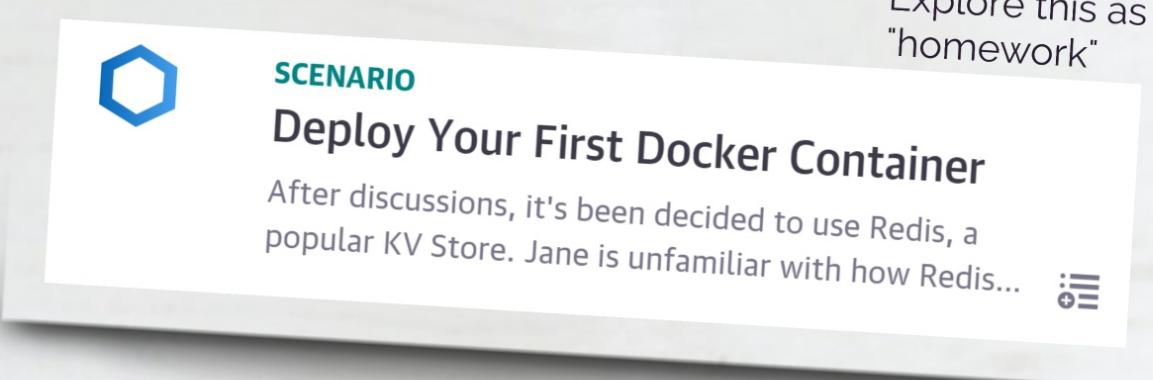
Experiment with docker cli tool:

```
$ docker build, push, pull, run, save, images, ps, search...
```



redis Container

In-memory data structure store
Common container for cloud native



Explore this as "homework"

SCENARIO

Deploy Your First Docker Container

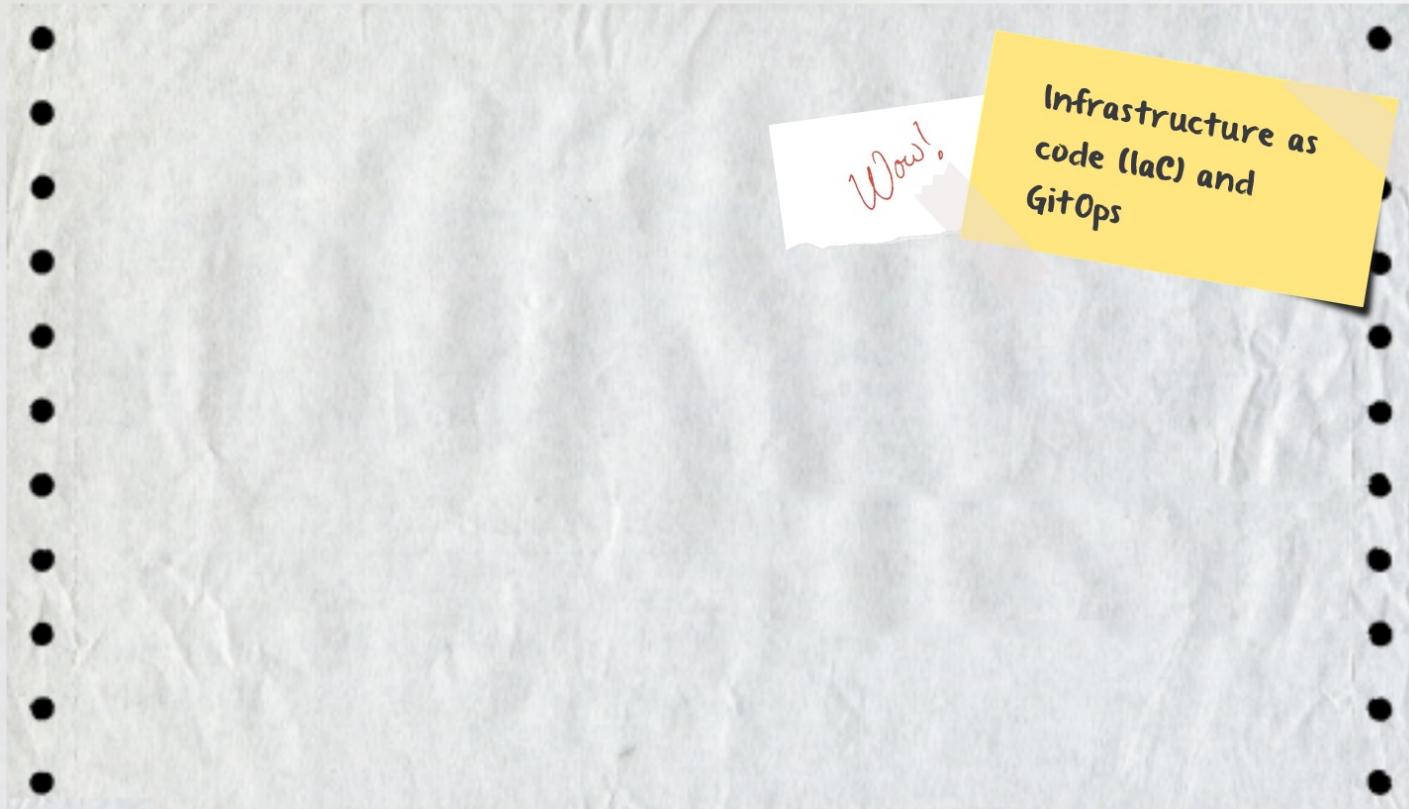
After discussions, it's been decided to use Redis, a popular KV Store. Jane is unfamiliar with how Redis... 

<https://bit.ly/3nhP5ku>

Experiment

- Container versions
- Foreground / background execution
- Inspection and logs
- Exposing ports
- Mounts to persistent storage
- Jumping into container's shell

Full Payloads with Few Lines



Full Payloads with Few Lines



```
FROM nginx  
ADD html /usr/share/nginx/html/
```

Wow!

Infrastructure as code (IaC) and GitOps

Full Payloads with Few Lines



```
FROM nginx  
ADD html /usr/share/nginx/ht
```

```
FROM postgres:9.5.3  
ADD data /data/  
RUN chmod -R 777 /data  
ADD sql /docker-entrypoint-initdb.d/
```

Wow!

Infrastructure as code (IaC) and GitOps

Full Payloads with Few Lines

NGINX



PostgreSQL



```
• FROM nginx  
• ADD html /usr/share/nginx/html/  
  
• FROM postgres:9.5.3  
• ADD data /data/  
• RUN chmod -R 777 /data  
• ADD sql /docker-entrypoint-initdb.d/  
  
• FROM java:openjdk-8-jre-alpine  
• ADD processor.jar processor.jar  
• ENTRYPOINT ["java", "-Dserver.port=9001", "-jar", "processor.jar"]  
• EXPOSE 9001
```

Wow!

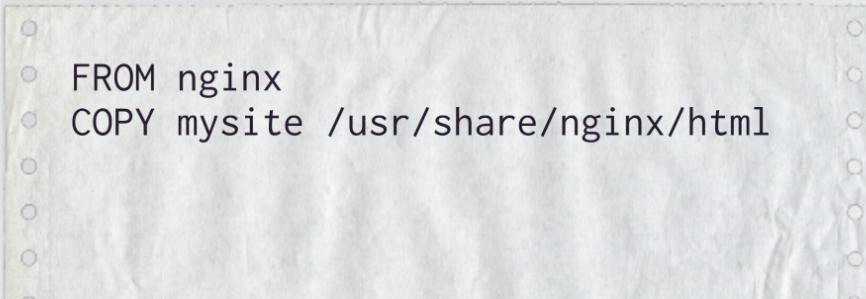
Infrastructure as
code (IaC) and
GitOps

Defining Containers

Domain specific language of Dockerfiles

- Dockerfile is infrastructure-as-code
 - Comments start with #
 - Instructions ALL CAPS
- Instruction set is small
- First instruction is FROM

Your web server with 2-line Dockerfile



Defining Containers

Domain specific language of Dockerfiles

- Dockerfile is infrastructure-as-code
 - Comments start with #
 - Instructions ALL CAPS
- Instruction set is small
- First instruction is FROM

Your web server with 2-line Dockerfile

```
FROM nginx
COPY mysite /usr/share/nginx/html
```

Dockerfile Instruction Set (17)

Common	Secondary
FROM	LABEL
RUN	EXPOSE
COPY	ENV
CMD	ADD
ENTRYPOINT	VOLUME
	USER
	WORKDIR
	ARG
	ONBUILD
	STOP SIGNAL
	HEALTHCHECK
	SHELL

NGINX Container



SCENARIO

Deploy a Static HTML Website as Container

In this scenario, you'll learn how to create a Docker image for running a static HTML website using...



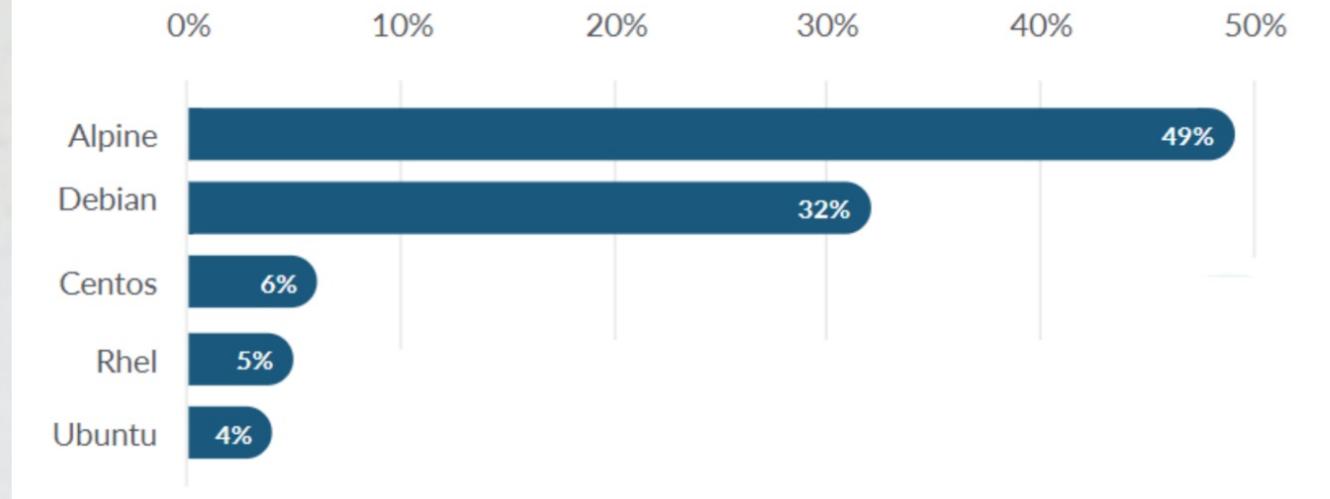
<https://bit.ly/3kmYFkb>

- Open source web serve
- Can be used as reverse proxy, load balancer, mail proxy, and HTTP cache
- Common for cloud native apps

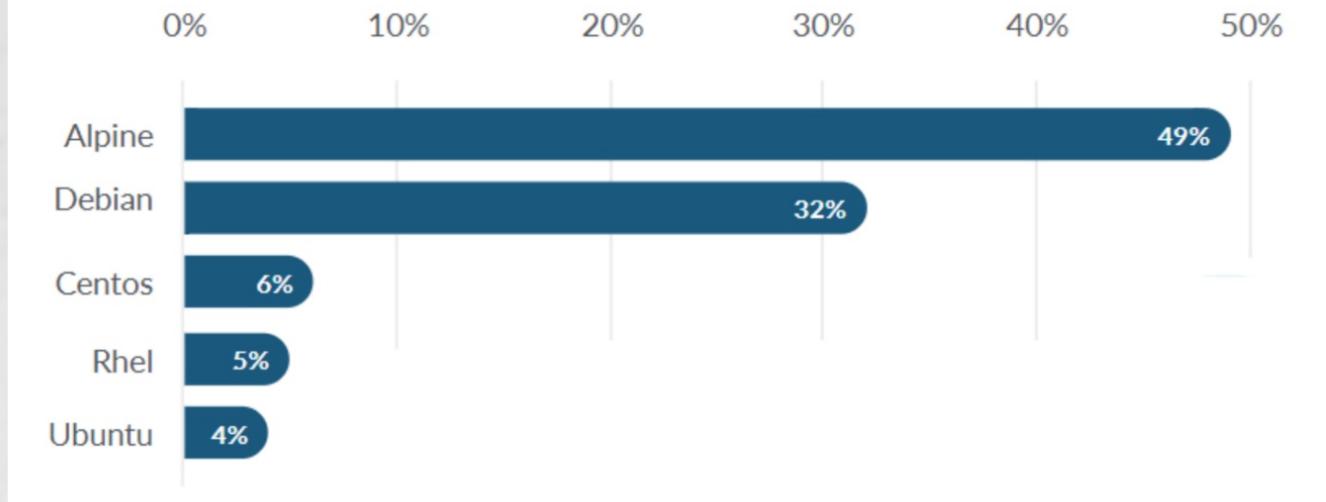
Experiment:

- Define Dockerfile
- Load HTML content into container
- Build, push, run your container

What are the top image distros?



What are the top image distros?

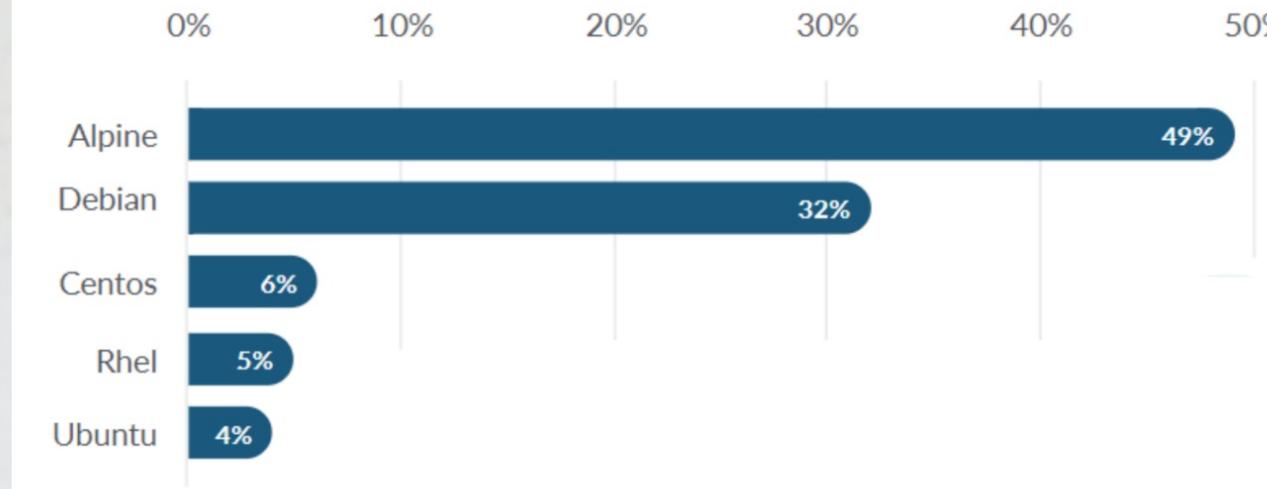


Average image size: **376 MB**

Images that run as root: **58%**

High/critical vulnerabilities: **53%**

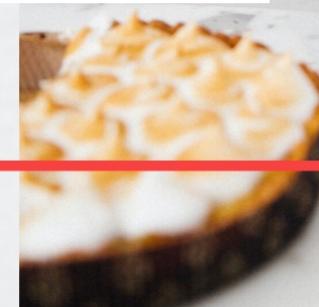
What are the top image distros?



Average image size: **376 MB**

Images that run as root: **58%**

High/critical vulnerabilities: **53%**



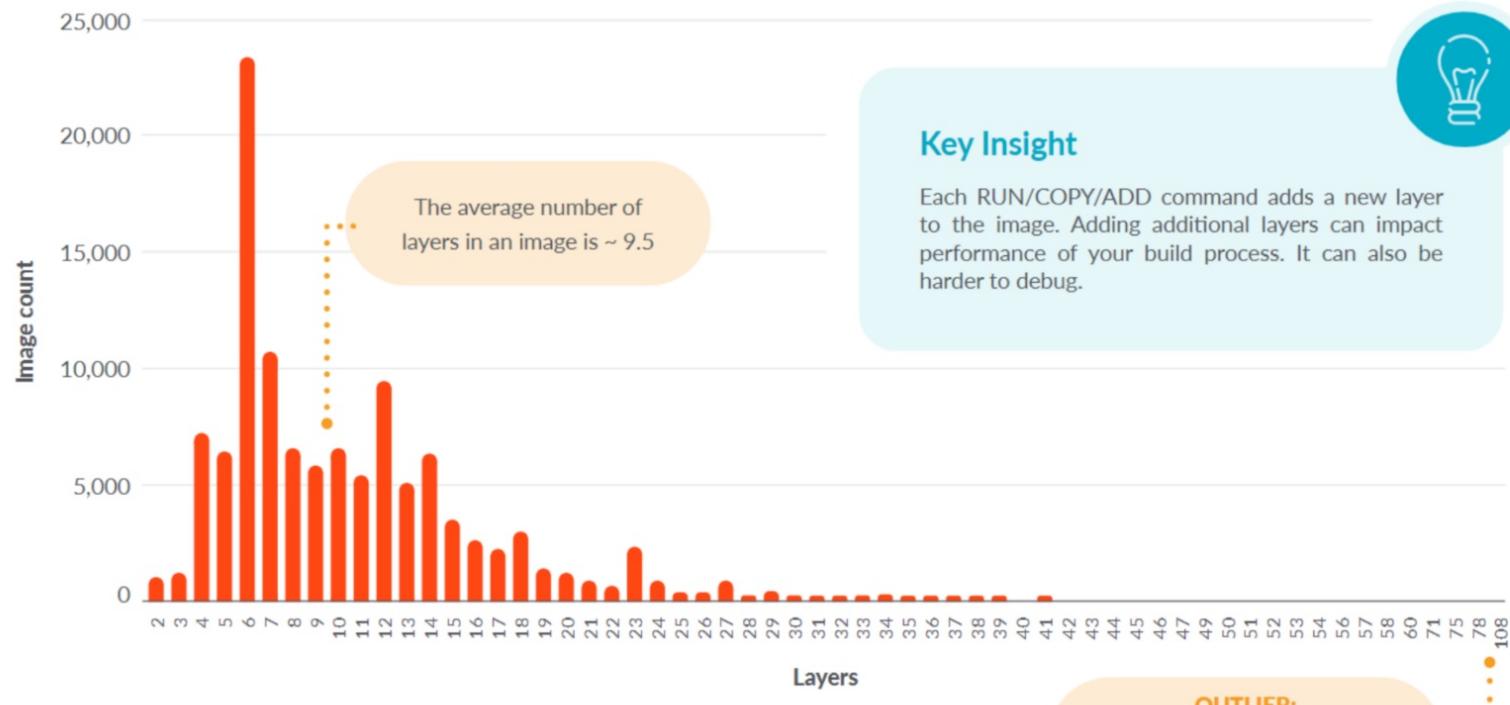
Stop doing that.
Distillation!

How many layers are part of an image?

Image Count vs. Layers



INFOGRAPHIC. 2020 Container Security Snapshot



Tar of Tars



Home of Original **Muffuletta**
Central Grocery and Deli
New Orleans, Louisiana
(Really, they came from Sicily)

Tar of Tars

```
$ docker pull redis
Using default tag: latest
latest: Pulling from library/redis
8d691f585fa8: Pull complete
8ccd02d17190: Pull complete
4719eb1815f2: Pull complete
200531706a7d: Pull complete
eed7c26916cf: Pull complete
e1285fcc6a46: Pull complete
Digest:
sha256:fe80393a67c7058590ca6b6903f64e35b50fa411b0496f604a85c526fb5bd2d2
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest

$ docker save --output redis.tar redis
```



Home of Original **Muffuletta**
Central Grocery and Deli
New Orleans, Louisiana
(Really, they came from Sicily)

Tar of Tars

```
$ docker pull redis
Using default tag: latest
latest: Pulling from library/redis
8d691f585fa8: Pull complete
8ccd02d17190: Pull complete
4719eb1815f2: Pull complete
200531706a7d: Pull complete
eed7c26916cf: Pull complete
e1285fcc6a46: Pull complete
Digest:
sha256:fe80393a67c7058590ca6b6903f64e35b50fa411b0496f604a85c526fb5bd2d2
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
```

```
$ docker save --output redis.tar redis
```

```
$ tar -tf redis.tar
1167eb15dc99e9c437519ce5dfcf46fa776b5be1b1acb86636b0c434d266eec/
  VERSION
  json
  layer.tar
3084577582f46c3659e22d20aa16607803299745d87d4f4d36bc9a632bbcdaec/
  799674d5f31766be4c5b95433657c01959c242160a38cf6ccdbd5af46cf31f2/
  bad490e5568e6fad2311dc0217b79408eb4cd61b56f9d6b22ad6b96179241525/
  f9c316ad0bbad0d22843076db519a1fc385b305dc908fd7c7027125e1e458e3/
  fd359d877a550df140b3480605e4e56d0aa245ad2d6a6ba4f1af70b62fd7549e/
  manifest.json
  repositories
```



Home of Original **Muffuletta**
Central Grocery and Deli
New Orleans, Louisiana
(Really, they came from Sicily)

Decomposing an Image File



learning.oreilly.com/interactive
Search for "**decomposing**"

Demystify containers:

- Understand container image format and layers
- Construct and run an OCI container without Docker
- Introduction to *Dive* tool





Rise of Containers



Developing Containers



Container Tools

APIs



Microservices

Polyglot

Distillation

Fin



OCI Images without Docker

Why can't I just use Docker CLI?

1. I want to run CI/CD pipelines *on* Kubernetes
2. Docker build inside engine does not play well
3. Need **daemonless** and **rootless**



Katacoda

[https://www.katacoda.com/
courses/containers-without-docker](https://www.katacoda.com/courses/containers-without-docker)

OCI Container Image Builders

- **Umoci** - OCI, reference implementation
 - **Kaniko** - Google
 - **Buildah/podman** - Red Hat
 - **BuildKit** - From Docker Moby Project
 - **Buildpacks** - CNCF (Pivotal, Heroku)
 - **PouchContainer** - Alibaba
 - **K3c** - From Rancher Labs, experimental
 - **FTL** - "Faster than light", depends on Bazel install
 - **Bazel** - Container Image Rules, Google
-
- **Img** - From Jess Frazelle
 - **orca-build** - Low activity

Decision fatigue with cloud native architectures



Katacoda

[https://www.katacoda.com/
courses/containers-without-docker](https://www.katacoda.com/courses/containers-without-docker)

BuildKit

Next gen "docker build"

Optional with Docker
CLI 18.06+, enable with:

```
$ DOCKER_BUILDKIT=1  
$ docker build .
```

- Based on Moby from Tõnis Tiigi
- Moby upstreamed from Docker Inc, April 2017
- Replaces Moby engine internals of default build features
- Faster build - efficient caching and parallelism
- Build OCI images from Dockerfile
- Build Buildpacks (Heroku and Cloud Foundry)
- Rootless
- Daemonless (forking daemon and then killed after a build)

Img

Genuine Tools

We are the home of quality software with a focus on simplicity, usability, security and minimalism.

gt

No bullshit, no politics, all genuine. Made by hackers with love that were just having fun.



Container image builder for Dockerfile and OCI

- Wraps Moby BuildKit library
- Concurrent, cache-efficient
- Docker like CLI (build, tag, push, pull, save, etc)
- Rootless - builds container images as unprivileged user
- Daemonless

Jessie Frazelle

```
$ img build -t product.com/my-app .
$ img push product.com/my-app
$ img save product.com/my-app | docker load
```

| Sorry, <https://github.com/genuinetools/img>
This site is not active.



- Google
- OCI images
- Understands Dockerfile
- Daemonless build inside Kubernetes Pod
- Runs as root, but in a protected container



- Google
- OCI images
- Understands Dockerfile
- Daemonless build inside Kubernetes Pod
- Runs as root, but in a protected container

Example: Build containers from a job

<https://harthoover.com/using-kaniko-for-container-builds-on-kubernetes/>

```
apiVersion: batch/v1
kind: Job
metadata:
  name: kanikojob
  namespace: kanikotest
spec:
  completions: 1
  template:
    metadata:
      name: kanikojob
      namespace: kanikotest
    spec:
      restartPolicy: Never
      initContainers:
        - name: init-clone-repo
          image: alpine/git
          args:
            - clone
            - --single-branch
            - --
            - https://github.com/repo/code.git
            - /context
      volumeMounts:
        - name: context
          mountPath: /context
      containers:
        - name: kaniko
          image: gcr.io/kaniko-project/executor:latest
          args: ["--dockerfile=/context/Dockerfile",
                 "--context=/context",
                 "--destination=registry/repo/container:latest"]
          volumeMounts:
            - name: context
              mountPath: /context
            - name: registry-creds
              mountPath: /root/
      volumes:
        - name: registry-creds
          projected:
            sources:
              - secret:
                  name: docker-secret
                items:
                  - key: .dockerconfigjson
                    path: docker/config.json
        - name: context
          emptyDir: {}
```

```
apiVersion: batch/v1
kind: Job
metadata:
  name: kanikojob
  namespace: kanikotest
spec:
  completions: 1
  template:
    metadata:
      name: kanikojob
      namespace: kanikotest
    spec:
      restartPolicy: Never
      initContainers:
```

```
namespace: kanikotest
spec:
  restartPolicy: Never
  initContainers:
    - name: init-clone-repo
      image: alpine/git
      args:
        - clone
        - --single-branch
        - --
        - - https://github.com/repo/code.git
        - /context
  volumeMounts:
    - name: context
      mountPath: /context
  containers:
```

```
    mountPath: /context
  containers:
    - name: kaniko
      image: gcr.io/kaniko-project/executor:latest
      args: ["--dockerfile=/context/Dockerfile",
             "--context=/context",
             "--destination=registry/repo/container:latest"]
    volumeMounts:
      - name: context
        mountPath: /context
      - name: registry-creds
        mountPath: /root/
  volumes:
    - name: registry-creds
      projected:
        sources:
          - secret:
              name: docker-secret
              items:
                - key: .dockerconfigjson
                  path: .docker/config.json
            - name: context
              emptyDir: {}
```

```
    mountPath: /context
  containers:
    - name: kaniko
      image: gcr.io/kaniko-project/executor:latest
      args: ["--dockerfile=/context/Dockerfile",
             "--context=/context",
             "--destination=registry/repo/container:latest"]
    volumeMounts:
      - name: context
        mountPath: /context
      - name: registry-creds
        mountPath: /root/
  volumes:
    - name: registry-creds
      projected:
        sources:
          - secret:
              name: docker-secret
              items:
                - key: .dockerconfigjson
                  path: .docker/config.json
            - name: context
              emptyDir: {}
```



- Google
- OCI images
- Understands Dockerfile
- Daemonless build inside Kubernetes Pod
- Runs as root, but in a protected container

Example: Build containers from a job

<https://harthoover.com/using-kaniko-for-container-builds-on-kubernetes/>

```
apiVersion: batch/v1
kind: Job
metadata:
  name: kanikojob
  namespace: kanikotest
spec:
  completions: 1
  template:
    metadata:
      name: kanikojob
      namespace: kanikotest
    spec:
      restartPolicy: Never
      initContainers:
        - name: init-clone-repo
          image: alpine/git
          args:
            - clone
            - --single-branch
            - --
            - https://github.com/repo/code.git
            - /context
      volumeMounts:
        - name: context
          mountPath: /context
      containers:
        - name: kaniko
          image: gcr.io/kaniko-project/executor:latest
          args: ["-dockerfile=/context/Dockerfile",
                 "-context=/context",
                 "-destination=registry/repo/container:latest"]
          volumeMounts:
            - name: context
              mountPath: /context
            - name: registry-creds
              mountPath: /root/
      volumes:
        - name: registry-creds
          projected:
            sources:
              - secret:
                  name: docker-secret
                  items:
                    - key: dockerconfigjson
                      path: docker/config.json
            - name: context
              emptyDir: {}
```

Example: Build container in Jenkins step

```
stages {
    stage('Checkout') {
        steps {
            git 'https://github.com/ioostvdp/cat.git'
        }
    }
    stage('Build') {
        steps {
            container('golang') {
                sh './build-go-bin.sh'
            }
        }
    }
    stage('Make Image') {
        environment {
            PATH = "/busybox:$PATH"
        }
        steps {
            container(name: 'kaniko', shell: '/busybox/sh') {
                sh ''#!/busybox/sh
                /kaniko/executor -f 'pwd'/Dockerfile.run -c 'pwd' --cache=true --destination=index.docker.io/caladreas/cat
            }
        }
    }
}
```

Example: Build container in Jenkins step

```
stages {
    stage('Checkout') {
        steps {
            git 'https://github.com/joostvdg/cat.git'
        }
    }
    stage('Build') {
        steps {
            container('golang') {
                sh './build-go-bin.sh'
            }
        }
    }
    stage('Make Image') {
        environment {
            PATH = "/busybox:$PATH"
        }
        steps {
            container(name: 'kaniko', shell: '/busybox/sh') {
                sh '''#!/bin/sh
/kaniko/executor -f `pwd`/Dockerfile.run -c `pwd` --cache=true --destination=index.docker.io/caladreas/cat
''''
            }
        }
    }
}
```



- Google
- OCI images
- Understands Dockerfile
- Daemonless build inside Kubernetes Pod
- Runs as root, but in a protected container

Example: Build containers from a job

<https://harthoover.com/using-kaniko-for-container-builds-on-kubernetes/>

```
apiVersion: batch/v1
kind: Job
metadata:
  name: kanikojob
  namespace: kanikotest
spec:
  completions: 1
  template:
    metadata:
      name: kanikojob
      namespace: kanikotest
    spec:
      restartPolicy: Never
      initContainers:
        - name: init-clone-repo
          image: alpine/git
          args:
            - clone
            - --single-branch
            - --
            - https://github.com/repo/code.git
            - /context
      volumeMounts:
        - name: context
          mountPath: /context
      containers:
        - name: kaniko
          image: gcr.io/kaniko-project/executor:latest
          args: ["-dockerfile=/context/Dockerfile",
                 "-context=/context",
                 "-destination=registry/repo/container:latest"]
          volumeMounts:
            - name: context
              mountPath: /context
            - name: registry-creds
              mountPath: /root/
          volumes:
            - name: registry-creds
              projected:
                sources:
                  - secret:
                      name: docker-secret
                    items:
                      - key: dockerconfigjson
                        path: docker/config.json
            - name: context
              emptyDir: {}
```

Example: Build container in Jenkins step

```
stages {
    stage('Checkout') {
        steps {
            git 'https://github.com/ioostvdp/cat.git'
        }
    }
    stage('Build') {
        steps {
            container('golang') {
                sh './build-go-bin.sh'
            }
        }
    }
    stage('Make Image') {
        environment {
            PATH = "/busybox:$PATH"
        }
        steps {
            container(name: 'kaniko', shell: '/busybox/sh') {
                sh ''#!/busybox/sh
                /kaniko/executor -f 'pwd'/Dockerfile.run -c 'pwd' --cache=true --destination=index.docker.io/caladreas/cat
            }
        }
    }
}
```

Java OCI Container Builders



Containerize your Java application.

- Builds optimized OCI images for **Java** apps
- Does not use docker or daemon
- Avoid mastery of Docker best-practices.
- Maven, Gradle, and Java library.



Special mention: <https://github.com/bmuschko/gradle-docker-plugin>

Ben Muschko Docker Gradle Plugin

This repository contains a list of tutorials to get started with Knative Build.

Hello, World

Learn how to install Knative Build, run a simple Build and inspect its result.

OPEN IN GOOGLE CLOUD SHELL

Docker Build with Kaniko

Learn how to build a Docker image from a Dockerfile, using [Kaniko](#).

OPEN IN GOOGLE CLOUD SHELL

Spring Boot Java application with Jib

Learn how to build a Spring Boot application without Docker, using [Jib](#).

OPEN IN GOOGLE CLOUD SHELL

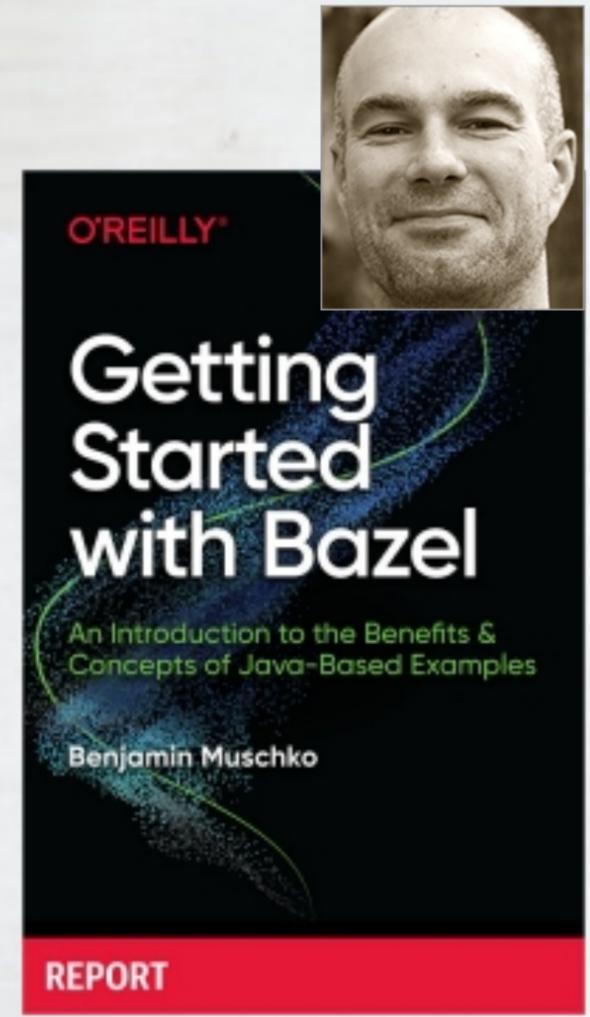
<https://github.com/GoogleCloudPlatform/knative-build-tutorials>



Bazel

- High level build language
- From Google Blaze project
- Parallels with Make, Ant, Maven, Gradle
- Faster build speed
- Artifact idempotency
(https://en.wikipedia.org/wiki/Deterministic_system)
- Creates OCI images

```
container_image(  
    name = "app",  
    # References container_pull from WORKSPACE (above)  
    base = "@java_base//image",  
    files = ["//java/com/example/app:Hello_deploy.jar"],  
    cmd = ["Hello_deploy.jar"]  
)
```





podman



buildah



skopeo

CLI: Daemonless,
rootless replacement
for docker

CLI: Build OCI
containers

CLI: Transfer
container images

```
$ alias docker=podman
```

Daniel Walsh



redhat.



The screenshot shows a blue header with the Katacoda logo (a cartoon cat) and the title "Running Containers with Podman". Below the header, there is descriptive text: "Based on Runc and Libpod, Podman launches OCI-based Containers via a familiar CLI". A large orange button at the bottom right is labeled "Start Scenario".

[https://www.katacoda.com/
courses/containers-without-docker/
running-containers-with-podman](https://www.katacoda.com/courses/containers-without-docker/running-containers-with-podman)

Source to Image



Accepts polyglot "src" files



Source to Image



git



Accepts polyglot "src" files



s2i



Produces Distilled
OCI Containers

Source to Image



Accepts polyglot "src" files

s2i



Produces Distilled
OCI Containers

```
$ s2i build https://github.com/openshift/ruby-hello-world centos/ruby-25-centos7 test-ruby-app
$ docker run --rm -i -p :8080 -t test-ruby-app

$ s2i build --ref=10.x --context-dir=helloworld \
  https://github.com/wildfly/quickstart \
  openshift/wildfly-101-centos7 test-jee-app
$ docker run --rm -i -p 8080:8080 -t test-jee-app
```

<https://github.com/openshift/source-to-image>



Buildpacks.io



CLOUD NATIVE
COMPUTING FOUNDATION



VMware Tanzu



Accepts polyglot "src" files

```
$ git clone https://github.com/paketo-buildpacks/samples
$ cd samples/demo-apps/app-source

$ pack build paketo-demo-app --builder paketobuildpacks/builder:base

$ docker run -d -p 8080:8080 -e PORT=8080 paketo-demo-app
```



Produces Distilled
OCI Containers

<https://paketo.io/>

Pipelines Steps as Containers



Knative

API for on-cluster container

- Eventing
- Serving



API for on-cluster container

• Building

More Toolbox Items

GraalVM™

Updating changes



S K A F F O L D

kSYNC

Debugging



<https://kubernetes.io/blog/2018/05/01/developing-on-kubernetes>

CLOUD NATIVE
COMPUTING FOUNDATION

<https://landscape.cncf.io>



Rise of Containers



Developing Containers



Container Tools

APIs



Microservices

Polyglot

Distillation

Fin



Common Protocol Choices

Typically synchronous

- **REST**: Common, easy
- **GraphQL**: Query language
- **gRPC**: Fast, Protocol Buffers, OSI layer 7
- **Websockets**: Stateful, 2-way comm
- **RSocket**: RPC from Netflix, OSI layer 5/6



REST API Framework

Across many languages

Example

[https://api.github.com/
users/dchen1107/repos](https://api.github.com/users/dchen1107/repos)



*Dawn Chen (Photo:
Google)*

Senior staff software
engineer at Google Cloud

REST API Framework

Across many languages



- Specification and language describing REST APIs.
- Based on verbs GET, POST, PUT, DELETE

Example

[https://api.github.com/
users/dchen1107/repos](https://api.github.com/users/dchen1107/repos)



Dawn Chen (Photo:
Google)

Senior staff software
engineer at Google Cloud



SWAGGER

- Tools for implementing specification
- Codegen generates API client libraries (SDK generation), server stubs, and documentation

SpringFox

Visualize REST APIs

The screenshot shows a web browser displaying the SpringFox API documentation for a "full-petstore-api (/v2/a)" endpoint. The browser's address bar shows "localhost:8080/springfox/". The main content area is titled "Api Documentation" and includes sections for "getStudent", "getStudentByCountry", and "pet".

getStudent
Show/Hide | List Operations | Expand Operations

getStudentByCountry
Show/Hide | List Operations | Expand Operations
DELETE /getStudentByCountry/{country} Get specific Student By Country in the System
GET /getStudentByCountry/{country} Get specific Student By Country in the System

Response Class (Status 200)
OK

Model Example Value

```
{  
    "cls": "string",  
    "country": "string",  
    "name": "string"  
}
```

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
country	<input type="text" value="Required"/>	country	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Springfox petstore API

Lore Ipsum is simply dummy text of the printing and typesetting i since the 1500s, when an unknown printer took a galley of type and centuries, but also the leap into electronic typesetting, remaining es Letraset sheets containing Lore Ipsum passages, and more recent versions of Lore Ipsum.

Created by springfox
[Apache License Version 2.0](#)

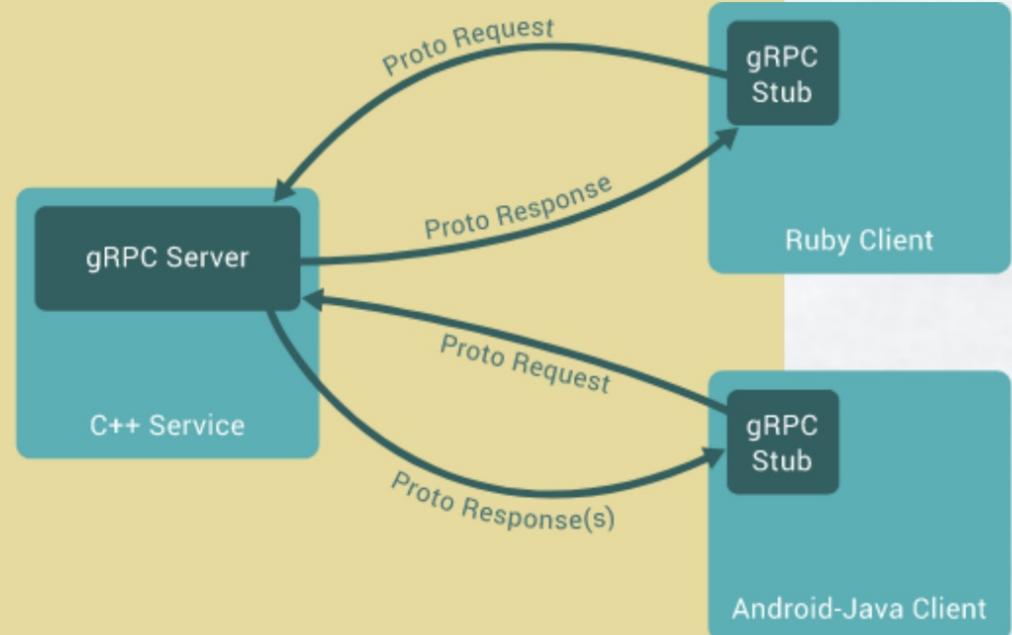
pet

Method	Path
POST	/api/pet
PUT	/api/pet
GET	/api/pet/findByStatus
GET	/api/pet/findByTags

gRPC · Remote Procedure Calls

CLOUD NATIVE
COMPUTING FOUNDATION

when
bytes
matter

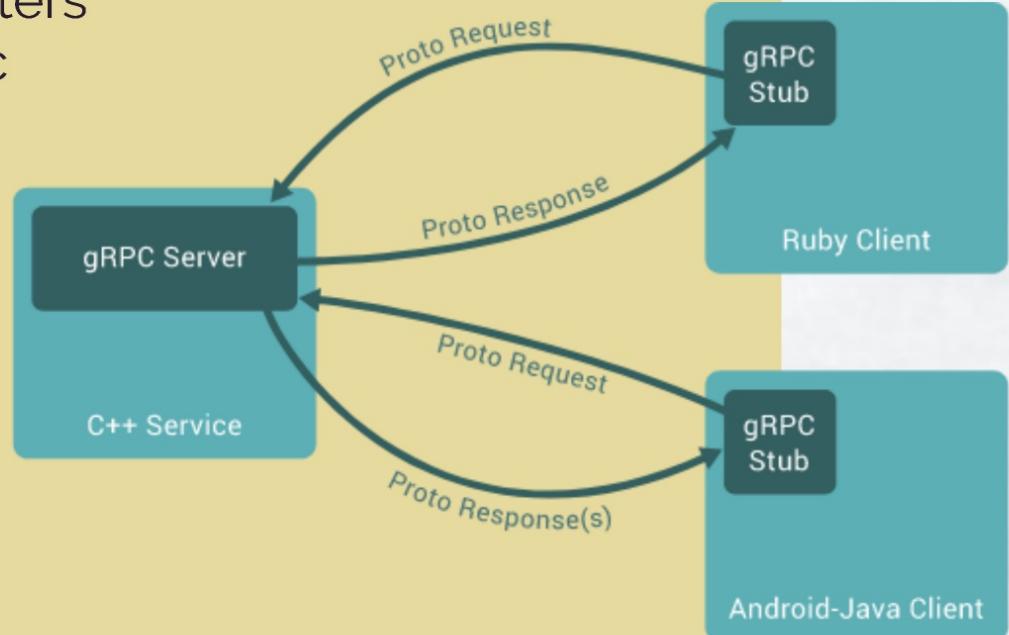


gRPC · Remote Procedure Calls

CLOUD NATIVE
COMPUTING FOUNDATION

when
bytes
matter

- **Protocol Buffers** (binary data stream)
- Optimized - each byte matters
- Performance for high traffic
- Coupling for performance
- Google APIs are gRPC
- Language agnostic
- OSI layer 7



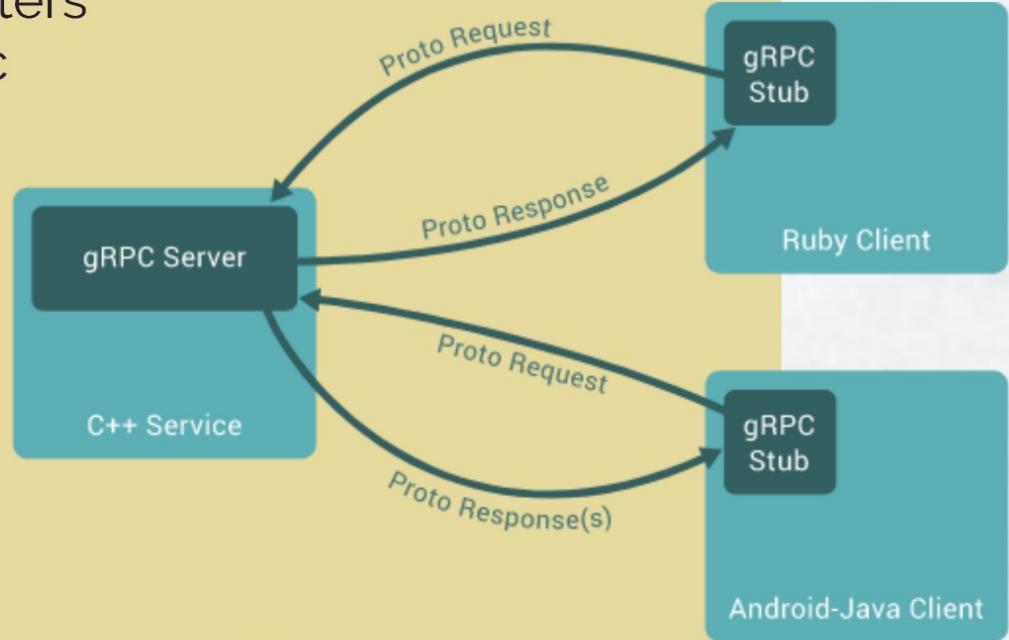
gRPC · Remote Procedure Calls

CLOUD NATIVE
COMPUTING FOUNDATION

when
bytes
matter

- **Protocol Buffers** (binary data stream)
- Optimized - each byte matters
- Performance for high traffic
- Coupling for performance
- Google APIs are gRPC
- Language agnostic
- OSI layer 7

C++, Java/JMV, Android Java,
Kotlin, Objective-C, Python, Ruby,
Go, C#, Node.js, Dart, PHP, Web



gRPC · Remote Procedure Calls

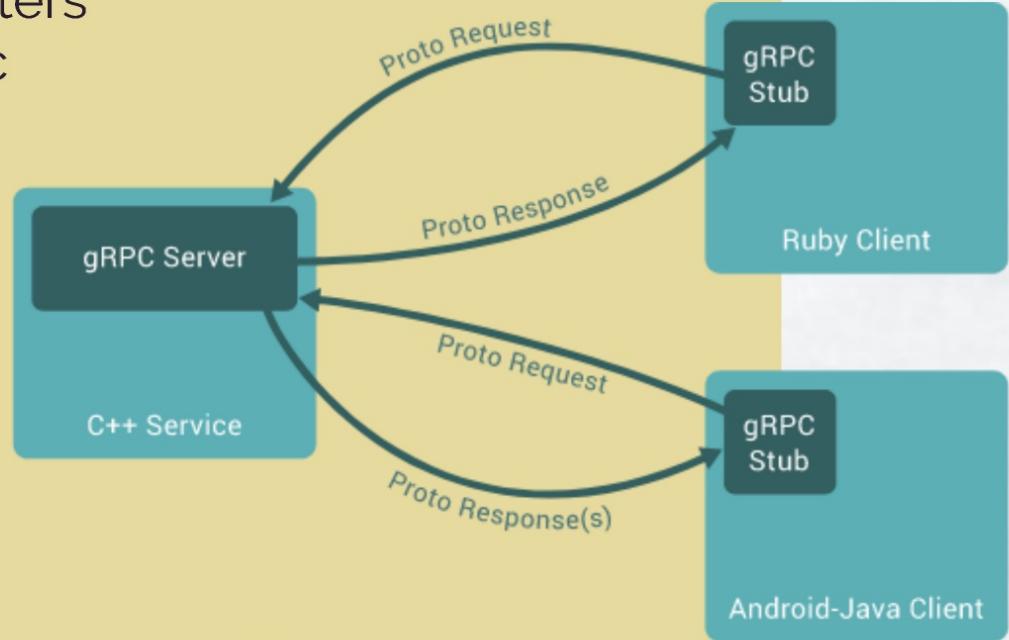
CLOUD NATIVE
COMPUTING FOUNDATION

when
bytes
matter

- **Protocol Buffers** (binary data stream)
- Optimized - each byte matters
- Performance for high traffic
- Coupling for performance
- Google APIs are gRPC
- Language agnostic
- OSI layer 7

C++, Java/JMV, Android Java,
Kotlin, Objective-C, Python, Ruby,
Go, C#, Node.js, Dart, PHP, Web

RPC forms have been difficult in past
with CORBA, DCOM, RMI and SOAP



gRPC · Remote Procedure Calls

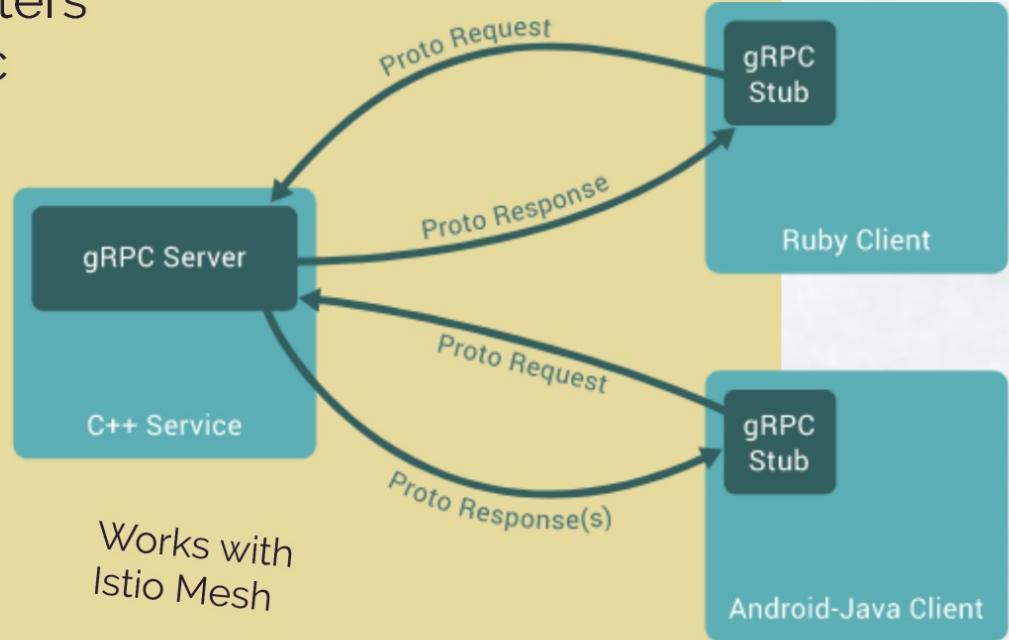
CLOUD NATIVE
COMPUTING FOUNDATION

when
bytes
matter

- **Protocol Buffers** (binary data stream)
- Optimized - each byte matters
- Performance for high traffic
- Coupling for performance
- Google APIs are gRPC
- Language agnostic
- OSI layer 7

C++, Java/JMV, Android Java,
Kotlin, Objective-C, Python, Ruby,
Go, C#, Node.js, Dart, PHP, Web

RPC forms have been difficult in past
with CORBA, DCOM, RMI and SOAP





GraphQL

Data Query Language

Open sourced by

1

Send structure
to the server

2

Server responds with
populated structure

```
{  
  user(id: 4802170) {  
    id  
    name  
    isViewerFriend  
    profilePicture(size: 50) {  
      uri  
      width  
      height  
    }  
    friendConnection(first: 5) {  
      totalCount  
      friends {  
        id  
        name  
      }  
    }  
  }  
}
```

```
{  
  "data": {  
    "user": {  
      "id": "4802170",  
      "name": "Lee Byron",  
      "isViewerFriend": true,  
      "profilePicture": {  
        "uri": "cdn://pic/4802170/50",  
        "width": 50,  
        "height": 50  
      },  
      "friendConnection": {  
        "totalCount": 13,  
        "friends": [  
          {  
            "id": "305249",  
            "name": "Stephen Schwink"  
          },  
          {  
            "id": "3108935",  
            "name": "Nathaniel Roman"  
          }  
        ]  
      }  
    }  
  }  
}
```

1

Send structure
to the server

```
{  
  user(id: 4802170) {  
    id  
    name  
    isViewerFriend  
    profilePicture(size: 50) {  
      uri  
      width  
      height  
    }  
    friendConnection(first: 5) {  
      totalCount  
      friends {  
        id  
        name  
      }  
    }  
  }  
}
```

2

Server responds with
populated structure

```
{  
  "data": {  
    "user": {  
      "id": "4802170",  
      "name": "Lee Byron",  
      "isViewerFriend": true,  
      "profilePicture": {  
        "uri": "cdn://pic/4802170/50",  
        "width": 50,  
        "height": 50  
      },  
      "friendConnection": {  
        "totalCount": 13,  
        "friends": [  
          {  
            "id": "305249",  
            "name": "Stephen Schwink"  
          },  
          {  
            "id": "3108935",  
            "name": "Nathaniel Roman"  
          }  
        ]  
      }  
    }  
  }  
}
```

1

Send structure
to the server

```
{  
  user(id: 4802170) {  
    id  
    name  
    isViewerFriend  
    profilePicture(size: 50) {  
      uri  
      width  
      height  
    }  
    friendConnection(first: 5) {  
      totalCount  
      friends {  
        id  
        name  
      }  
    }  
  }  
}
```

2

Server responds with
populated structure

```
{  
  "data": {  
    "user": {  
      "id": "4802170",  
      "name": "Lee Byron",  
      "isViewerFriend": true,  
      "profilePicture": {  
        "uri": "cdn://pic/4802170/50",  
        "width": 50,  
        "height": 50  
      },  
      "friendConnection": {  
        "totalCount": 13,  
        "friends": [  
          {  
            "id": "305249",  
            "name": "Stephen Schwink"  
          },  
          {  
            "id": "3108935",  
            "name": "Nathaniel Roman"  
          }  
        ]  
      }  
    }  
  }  
}
```

Data shape is define by requester

Structure is hierarchical

Strongly typed

Independent of storage structures

Version free



GraphQL

Data Query Language

Open sourced by f

1 Send structure to the server

```
{  
  user(id: 4802170) {  
    id  
    name  
    isViewerFriend  
    profilePicture(size: 50) {  
      uri  
      width  
      height  
    }  
    friendConnection(first: 5) {  
      totalCount  
      friends {  
        id  
        name  
      }  
    }  
  }  
}
```

2 Server responds with populated structure

```
{  
  "data": {  
    "user": {  
      "id": "4802170",  
      "name": "Lee Byron",  
      "isViewerFriend": true,  
      "profilePicture": {  
        "uri": "cdn://pic/4802170/50",  
        "width": 50,  
        "height": 50  
      },  
      "friendConnection": {  
        "totalCount": 13,  
        "friends": [  
          {  
            "id": "305249",  
            "name": "Stephen Schwink"  
          },  
          {  
            "id": "3108935",  
            "name": "Nathaniel Roman"  
          }  
        ]  
      }  
    }  
  }  
}
```



Rise of Containers



APIs



Polyglot

Distillation

Fin



Two Delicious Architectures

Containers

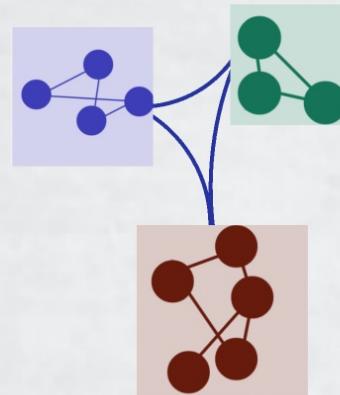


Two Delicious Architectures

Containers



Microservices

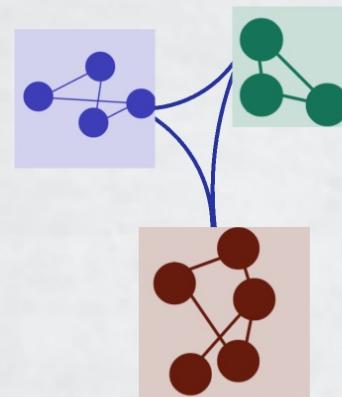


Two Delicious Architectures

Containers



Microservices



Tasty Combination

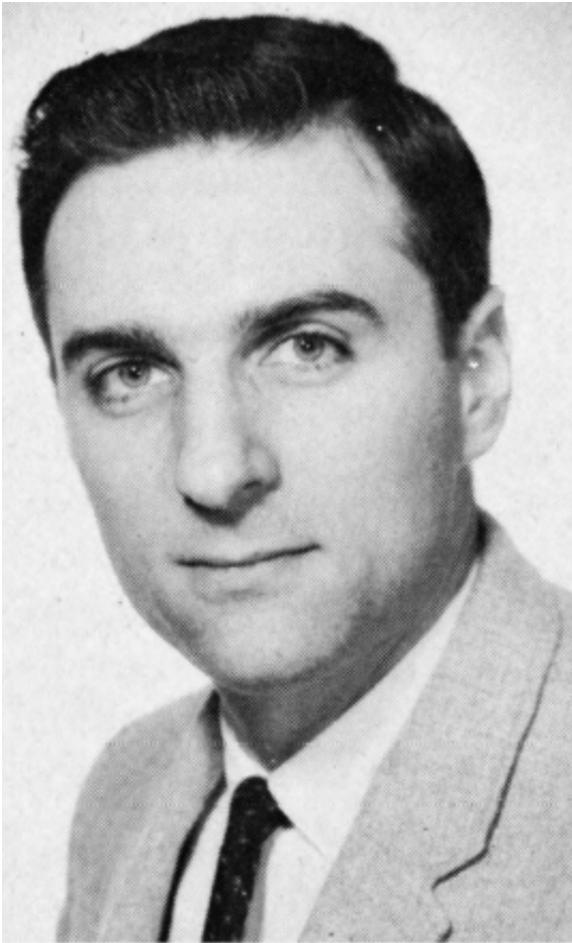


Microservices

- Apps as suites of independently deployable services
- CS 101: High cohesion, low coupling
- Often own small datastore
- Scaling at a micro level
- Do one thing and do it well
- Sometimes they feel like OO classes
- Monolith antimatter

- Unix philosophy

[https://martinfowler.com/
articles/microservices.html](https://martinfowler.com/articles/microservices.html)



Team organization

"Any organization that designs a system (defined more broadly here than just information systems) will inevitably produce a design whose structure is a copy of the organization's communication structure."

— Melvin E. Conway, 1968

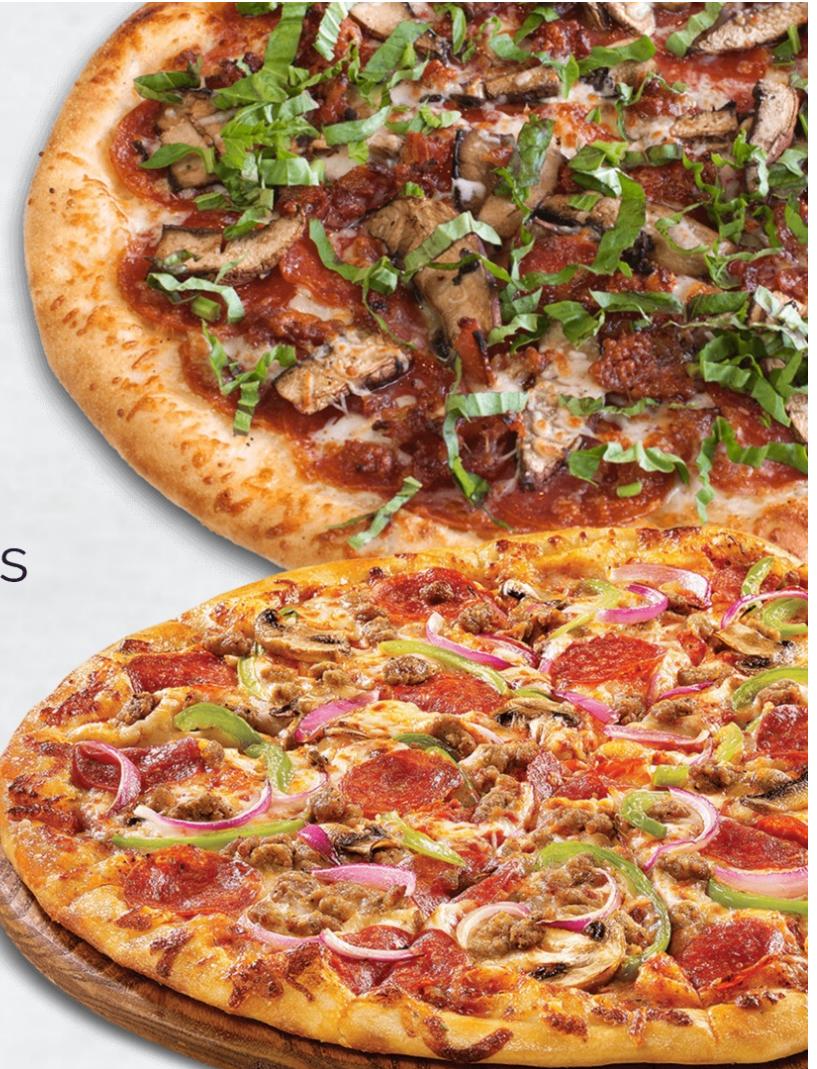
Each team continuously delivers solutions, independently

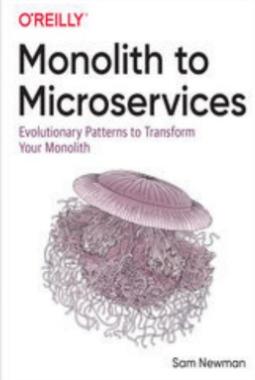
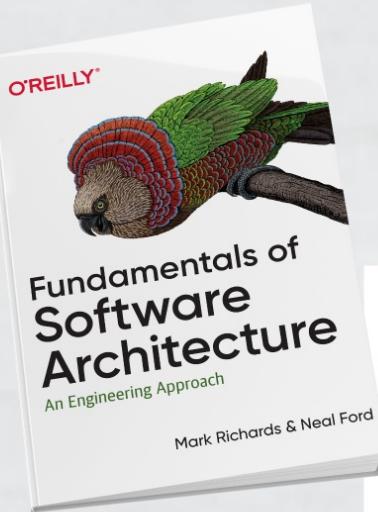
Team

Communication and size

Jeff Bezos

- No matter how large your company gets, individual teams shouldn't be larger than what two pizzas can feed.
- API Mandate: All teams only communication through APIs, else you are fired.





Mark
Richards



Neal
Ford

space-based architecture

	layered monolith	microkernel	microservices	service-based	event-driven	space-based
agility	★	★★★	★★★★★	★★★★★	★★★★★	★★★★
deployment	★	★★★	★★★★★	★★★★★	★★★	★★★
testability	★★	★★★	★★★★★	★★★★★	★★	★
performance	★★★	★★★	★★	★★★	★★★★★	★★★★★
scalability	★	★	★★★★★	★★★	★★★★★	★★★★★
elasticity	★	★	★★★★★	★★	★★★	★★★★★
simplicity	★★★★★	★★★★★	★	★★★	★	★
fault-tolerance	★	★	★★★★★	★★★★★	★★★★★	★★★
evolvability	★	★★★	★★★★★	★★★★★	★★★★★	★★★
total cost	★★★★★	★★★★★	★	★★★★★	★★★	★★

space-based architecture

	layered monolith	microkernel	microservices	service-based	event-driven	space-based
agility	★	★★★	★★★★★	★★★★★	★★★★★	★★★★
deployment	★	★★★	★★★★★	★★★★★	★★★	★★★
testability	★★	★★★	★★★★★	★★★★★	★★	★
performance	★★★	★★★	★	★★★	★★★★★	★★★★★
scalability	★	★	★★★★★	★★★★	★★★★	★★★★★
elasticity	★	★	★★★★★	★★	★★★	★★★★★
simplicity	★★★★★	★★★★★	★	★★★	★	★
fault-tolerance	★	★	★★★★★	★★★★★	★★★★★	★★★
evolvability	★	★★★	★★★★★	★★★★★	★★★★★	★★★
total cost	★★★★★	★★★★★	★	★★★★★	★★★	★★★

space-based architecture

	layered monolith	microkernel	microservices	service-based	event-driven	space-based
agility	★	★★★	★★★★★	★★★★★	★★★★★	★★★★
deployment	★	★★★	★★★★★	★★★★★	★★★	★★★
testability	★★	★★★	★★★★★	★★★★★	★★	★
performance	★★★★	★★★	★★★	★★★	★★★★★	★★★★★
scalability	★	★	★★★★★	★★★★	★★★★★	★★★★★
elasticity	★	★	★★★★★	★★★	★★★	★★★★★
simplicity	★★★★★	★★★★★	★	★★★	★	★
fault-tolerance	★	★	★★★★★	★★★★★	★★★★★	★★★
evolvability	★	★★★	★★★★★	★★★★★	★★★★★	★★★
total cost	★★★★★	★★★★★	★	★★★★★	★★★	★★★

space-based architecture

	layered monolith	microkernel	microservices	service-based	event-driven	space-based
agility	★	★★★	★★★★★	★★★★★	★★★★★	★★★★
deployment	★	★★★	★★★★★	★★★★★	★★★	★★★
testability	★★	★★★	★★★★★	★★★★★	★★	★
performance	★★★★	★★★	★★★ 	★★★	★★★★★	★★★★★
scalability	★	★	★★★★★	★★★★	★★★★★	★★★★★
elasticity	★	★	★★★★★	★★★	★★★	★★★★★
simplicity	★★★★★	★★★★★	★★★ 	★★★	★	★
fault-tolerance	★	★	★★★★★	★★★★★	★★★★★	★★★
evolvability	★	★★★	★★★★★	★★★★★	★★★★★	★★★
total cost	★★★★★	★★★★★	★	★★★★★	★★★	★★★

space-based architecture

	layered monolith	microkernel	microservices	service-based	event-driven	space-based
agility	★	★★★	★★★★★	★★★★★	★★★★★	★★★★
deployment	★	★★★	★★★★★	★★★★★	★★★	★★★
testability	★★	★★★	★★★★★	★★★★★	★★	★
performance	★★★★	★★★	★★★ 	★★★	★★★★★	★★★★★
scalability	★	★	★★★★★	★★★★★	★★★★★	★★★★★
elasticity	★	★	★★★★★	★★★	★★★	★★★★★
simplicity	★★★★★	★★★★★	★★★ 	★★★	★	★
fault-tolerance	★	★	★★★★★	★★★★★	★★★★★	★★★
evolvability	★	★★★	★★★★★	★★★★★	★★★★★	★★★
total cost	★★★★★	★★★★★	★★★ 	★★★	★★★	★★★

Choices

For distributed system, the number of ecosystems and sub-choices can be daunting.

Microservices design decisions:

- Language
- Frameworks within language
- API type
- Build tools
- Data stores
- Messaging
- Observability







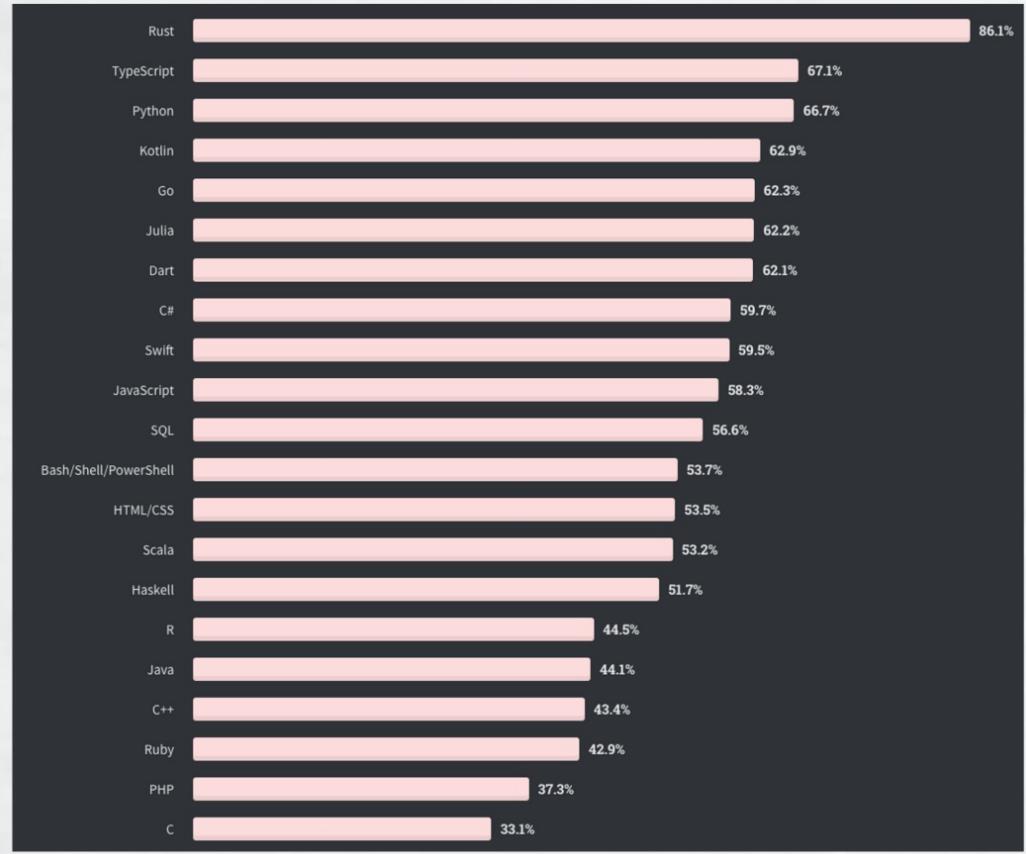
"There are only two kinds of languages: the ones people complain about and the ones nobody uses."
- Bjarne Stroustrup

Most Loved Languages

- StackOverflow Dev Survey 2020

Languages offer
frameworks and
communities
to support
microservices.

Which is best for you?



Language Determines Microframeworks

Microframework is "minimalistic" web application framework

Sample frameworks for languages

Bottle, Flask, Falcon for **Python**

Camping, Sinatra for **Ruby**

Express.js for **Node.js**

Scalatra for **Scala**

Lumen, Silex, Slim for **PHP**

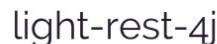
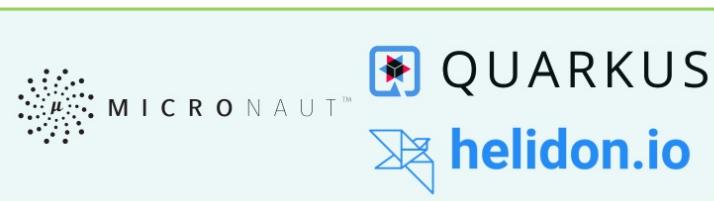
Ktor for **Kotlin**

Tonic for **Rust**

Revel, Gin, Martini, Gorilla, Buffalo, Goji for **Go**

Java Microframeworks

Minimalistic web application frameworks



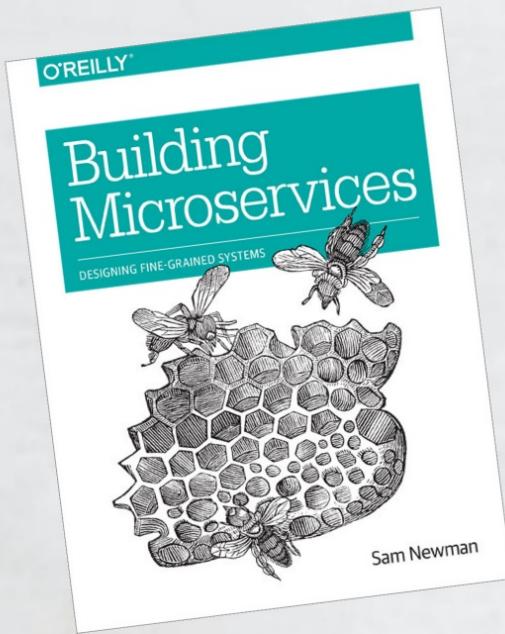
more...

Features often provided

- Tooling
- Libraries
- Observability *
- Debugging *
- Connectivity *
- APIs: REST, gRPC, QueryML

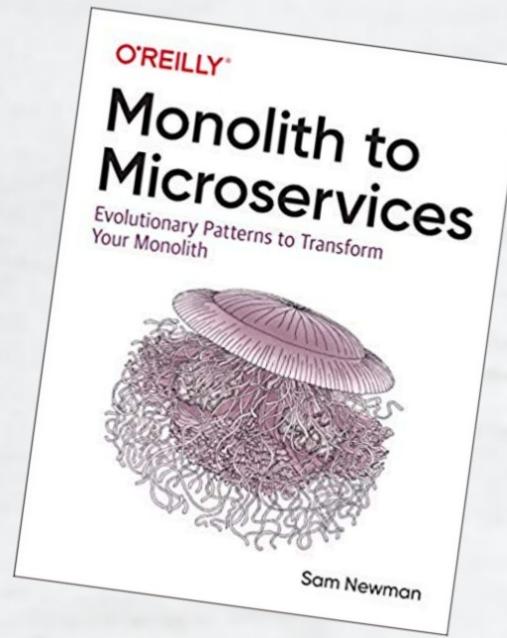
* Although a mesh can also help

Recommended Study



Sam Newman

Independent technology consultant
focusing in Microservices, Cloud and
Continuous Delivery

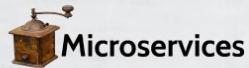




Rise of Containers



APIs



Polyglot

Distillation

Fin





Multi-stage Dockerfile

Produce sleek images without cruft
of intermediate build artifacts.

```
FROM golang:1.7.3 as build
WORKDIR /go/src/github.com/alexellis/href-counter/
RUN go get -d -v golang.org/x/net/html
COPY app.go .
RUN CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o app .

FROM alpine:latest
RUN apk --no-cache add ca-certificates
WORKDIR /root/
COPY --from=build /go/src/github.com/alexellis/href-counter/app .
CMD ["../app"]
```

New to Docker 17.05, May 2017

 Good pattern for GraalVM

Produce sleek images without cruft
of intermediate build artifacts.



```
FROM golang:1.7.3 as build
WORKDIR /go/src/github.com/alexellis(href-counter/
RUN go get -d -v golang.org/x/net/html
COPY app.go .
RUN CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o app .

FROM alpine:latest
RUN apk --no-cache add ca-certificates
WORKDIR /root/
COPY --from=build /go/src/github.com/alexellis(href-counter/app .
CMD ["../app"]
```

Produce sleek images without cruft
of intermediate build artifacts.

learning.ore
Search fo

```
FROM golang:1.7.3 as build
WORKDIR /go/src/github.com/alexellis/href-counter/
RUN go get -d -v golang.org/x/net/html
COPY app.go .
RUN CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o app .

FROM alpine:latest
RUN apk --no-cache add ca-certificates
WORKDIR /root/
COPY --from=build /go/src/github.com/alexellis/href-counter/app .
CMD ["../app"]
```

Multi-stage Dockerfile

Produce sleek images without cruft of intermediate build artifacts.



SCENARIO

Create Optimized Docker Images using Multi-Stage Builds

Create Optimized Docker Images using Multi-Stage Builds



learning.oreilly.com/scenarios
Search for "multi-stage"

```
FROM golang:1.7.3 as build
WORKDIR /go/src/github.com/alexellis/href-counter/
RUN go get -d -v golang.org/x/net/html
COPY app.go .
RUN CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o app .

FROM alpine:latest
RUN apk --no-cache add ca-certificates
WORKDIR /root/
COPY --from=build /go/src/github.com/alexellis/href-counter/app .
CMD ["../app"]
```

New to Docker 17.05, May 2017

💡 Good pattern for GraalVM



- Google's "easier" C++
- Build simple, reliable, and efficient software
- When every byte counts

Gin Web
Framework



SCENARIO

Kubernetes Containers: Go Apps in Containers on Kubernetes

Packaging Go applications in containers



learning.oreilly.com/scenarios
Search for "**go apps**"

- **Write** microservice with Go
- **Package** Go microservice in container
- **Run** Go microservice on Kubernetes



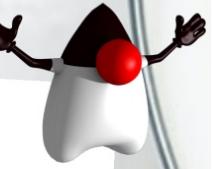
Most popular
framework for
Java based
webservices



SCENARIO

Kubernetes Fundamentals: Spring Boot to Kubernetes

From a blank slate to a Spring Boot app running on
Kubernetes



learning.oreilly.com/scenarios
Search for "**spring boot to**"

- Generate Spring Boot skeleton
- Add basic REST endpoints
- Define multi-stage Dockerfile
- Build jar and container
- Push container to private registry
- Start app on Kubernetes
- Run your application



gRPC



SCENARIO

Kubernetes Containers: Node.js to Kubernetes

Connect two Node.js-based microservices with gRPC and run them on Kubernetes

learning.oreilly.com/scenarios
Search for "**node.js to**"

- Compile small Node.js web and server application
- Node.js application into a distilled container
- Push/pull from private registry on Kubernetes
- Connect applications via gRPC
- Run app on Kubernetes



Rise of Containers



APIs



Polyglot

Distillation

Fin



Java to Container

How hard could it be?

Dockerfile:



Linux

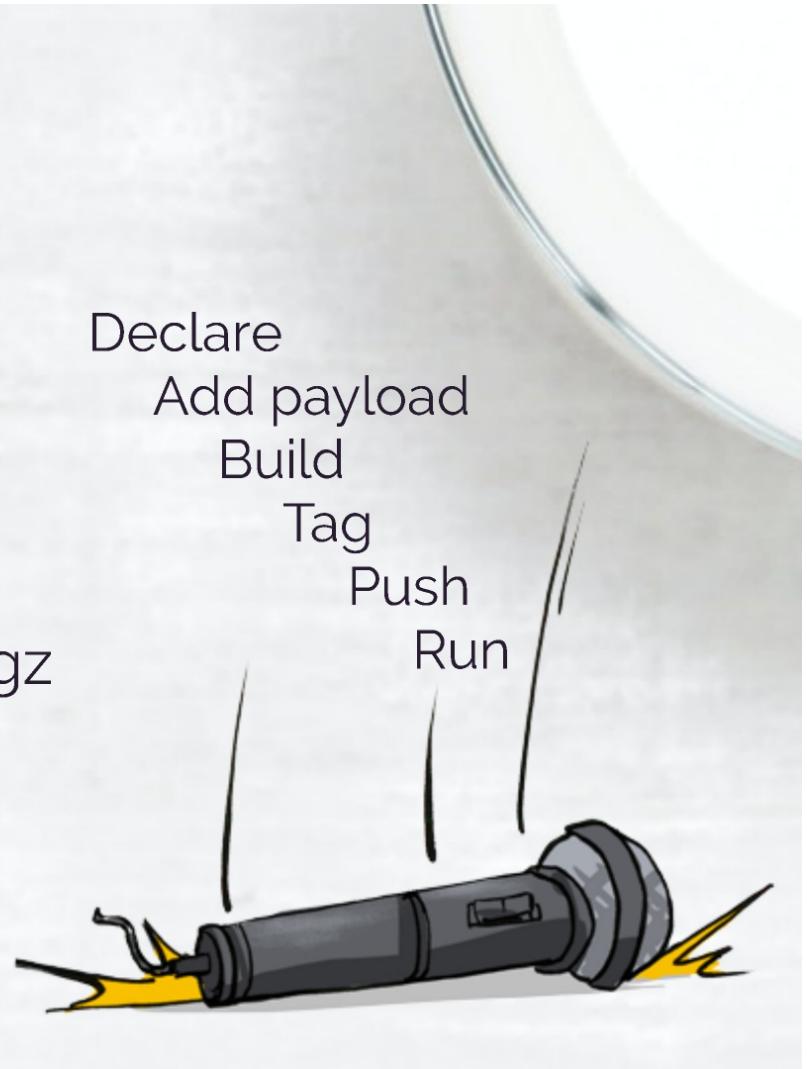


openjdk-8_linux-x64_bin.tar.gz



HelloWorld.class

Declare
Add payload
Build
Tag
Push
Run



Distillation Pattern

Distilling to get just the right taste, in a container



Distillation Pattern

Distilling to get just the right taste, in a container



PeatOps 😊

Distillation Pattern

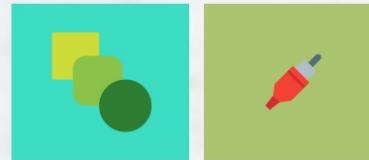
8 Factors

Distillation Pattern

8 Factors

High cohesion

All features in the container are well defined, and used.
"Do one thing and do it well." -- Unix



Low coupling

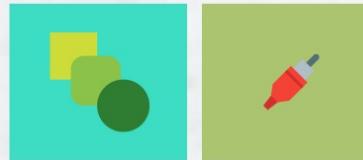
Exposed interfaces are minimal, simple, and stable.

Distillation Pattern

8 Factors

High cohesion

All features in the container are well defined, and used.
"Do one thing and do it well." -- Unix



Immutable

Container not capable of or susceptible to change.



Low coupling

Exposed interfaces are minimal, simple, and stable.

Idempotent

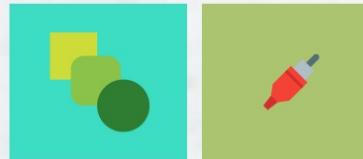
Multiple instances all behave the same, over time.

Distillation Pattern

8 Factors

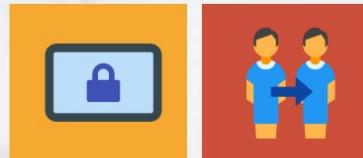
High cohesion

All features in the container are well defined, and used.
"Do one thing and do it well." -- Unix



Immutable

Container not capable of or susceptible to change.



Attack phobic

Security attack vectors are resisted as potential surfaces have been minimized.



Low coupling

Exposed interfaces are minimal, simple, and stable.

Idempotent

Multiple instances all behave the same, over time.

Small images

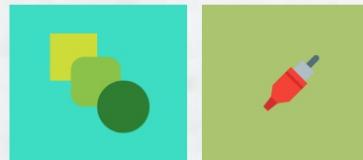
Less is more. Reduce I/O pressure. Images are revised, stored, transported, downloaded and scaled.

Distillation Pattern

8 Factors

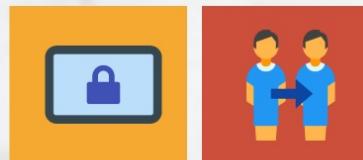
High cohesion

All features in the container are well defined, and used.
"Do one thing and do it well." -- Unix



Immutable

Container not capable of or susceptible to change.



Attack phobic

Security attack vectors are resisted as potential surfaces have been minimized.



Rapid startup

Containers are embraced as ephemeral, fragile, expendable, and scalable cattle.



Low coupling

Exposed interfaces are minimal, simple, and stable.

Idempotent

Multiple instances all behave the same, over time.

Small images

Less is more. Reduce I/O pressure. Images are revised, stored, transported, downloaded and scaled.

Resource frugality

Every bit and CPU tick is a limited resource as containers are scaled throughout your datacenter.



Minify and Secure Containers

Java application images:

from ubuntu:14.04 - 743.6 MB => 100.3 MB

Node.js application images:

from ubuntu:14.04 - 432MB => 14MB (by 30.85X)

from debian:jessie - 406MB => 25.1MB (16.21X)

from node:alpine - 66.7MB => 34.7MB (1.92X)

from node:distroless - 72.7MB => 39.7MB (1.83X)

Python application images:

from ubuntu:14.04 - 438MB => 16.8MB (by 25.99X)

from python:2.7-alpine - 84.3MB => 23.1MB (by 3.65X)

from python:2.7.15 - 916MB => 27.5MB (by 33.29X)

from centos:7 - 647MB => 23MB (by 28.57X)

from centos/python-27-centos7 - 700MB => 24MB (29.01X)

from python2.7:distroless - 60.7MB => 18.3MB (by 3.32X)

Base Containers

Many choices, open projects

Common examples

- Alpine
 - Debian
 - Centos
 - Ubuntu
 - Fedora
-
- Distroless
 - Scratch

Notable Freemiums, Frenemies



Minimal Ubuntu, 29MB



Universal Base Image (UBI)



zulu-openjdk

Base Containers

Many choices, open projects

Common examples

- Alpine
 - Debian
 - Centos
 - Ubuntu
 - Fedora
-
- Distroless
 - Scratch

Notable Freemiums, Frenemies



Minimal Ubuntu, 29MB



Universal Base Image (UBI)



zulu-openjdk

Bake your own base

Don't you have enough to do? Security, performance, patches.... oh my

C standard library

libc

- Library of types, functions, macros, operating system services with resource management
- Kernel calls
- CPU, Memory, I/O
- Language ports link to it

C standard library

libc

- Library of types, functions, macros, operating system services with resource management
- Kernel calls
- CPU, Memory, I/O
- Language ports link to it



Two common c libs:



- | | |
|-------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">• In Debian and other common distros• More complete, yet perhaps "bloated" | <ul style="list-style-type: none">• Several distributions including Alpine• Partially binary compatible with glibc• Lightweight, optimized for embedded |
|-------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

C standard library

libc

- Library of types, functions, macros, operating system services with resource management
- Kernel calls
- CPU, Memory, I/O
- Language ports link to it

Langs, like **Java**,
port and validate
on different OSs
with libc variants.

Your app
JRE
C++
libc
OS

Two common c libs:



**GNU C
Library (glibc)**



musl libc

- In Debian and other common distros
- More complete, yet perhaps "bloated"
- Several distributions including Alpine
- Partially binary compatible with glibc
- Lightweight, optimized for embedded





Small Linux for Distillation Pattern

- 5MB
- Security-oriented
- Lightweight
- musl libc and busybox
- Popular because of size
- Some find musl c slower



Small Linux for Distillation Pattern

- 5MB
- Security-oriented
- Lightweight
- musl libc and busybox
- Popular because of size
- Some find musl c slower



OpenJDK lacked support for Alpine after 8u212-jre-alpine3.9, but has reappeared in 13-alpine3.9.



Distilling Containers

Efficient packaging Java into containers



SCENARIO

Kubernetes Fundamentals: Distilled JRE Apps in Containers

Applying the Distillation Pattern to Containers



Discover steps to distill Java app container to its minimum size.

Smallest image size we can achieve, with Java, running on Linux?

Let's find out...



Distilling Containers

Efficient packaging Java into containers



SCENARIO

Kubernetes Fundamentals: Distilled JRE Apps in Containers

Applying the Distillation Pattern to Containers



learning.oreilly.com/scenarios
Search for "**distill**"

Discover steps to distill Java app container to its minimum size.

Smallest image size we can achieve, with Java, running on Linux?

Let's find out...



Rise of Containers



APIs



Polyglot

Distillation

Fin



Thank you

Veni, Vidi, Didici

I came, I saw, I learned



Containers now preferred packaging for cloud native
Kubernetes helps you navigate complexities of distributed computing

Polyglot · Images · Registries · Distillation · Deployment



dijure.com



[jonathan.johnson
@dijure.com](mailto:jonathan.johnson@dijure.com)



[twitter.com/
javajonjohn](https://twitter.com/javajonjohn)



[github.com/
javajon](https://github.com/javajon)



[linkedin.com/in/
javajon](https://linkedin.com/in/javajon)

Kubernetes Challenge: Assembling a Multi-Component Application

- A simple voting application
- Run commands to start up a multi-microservice app

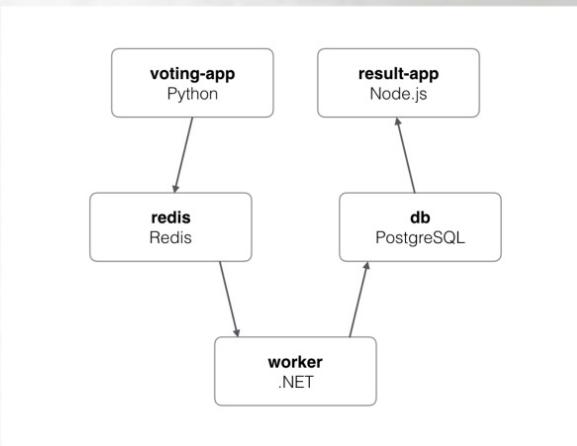


SCENARIO

Kubernetes Skills: Assembling a Multi-Component Application

Challenge your skills assembling a multi-component application containing web interfaces, microservices,...

learning.oreilly.com/scenarios
Search for "**assembling**"





Rise of Containers



APIs



Polyglot

Distillation

Fin

