HENRY QUINN

INTRO TO DOCKER

ABOUT ME

- NewHaven.IO
  - Member Board of Directors
- U.S. District Court, District of Connecticut
  - Programmer, Analyst, Database Administrator
- Administrative Office of the US Courts
  - Software Developer - Temporary Duty Assignment

PLAN FOR TODAY

- What I Was Tasked With
- What Was Expected Of Me
  - VPS Overview (Virtual Private Server)
- What I Did Instead
  - Docker Overview

GET A COLDFUSION APP IN 16 COURTS IN UNDER A YEAR

- CONSTRAINTS
  - Finish The Application
  - Get It Cleared As Secure By IT Security Office
  - Requisition Servers
  - Install, Configure, And Test Applications

AO CONTROLLED VIRTUAL PRIVATE SERVERS

FROM NOTHING

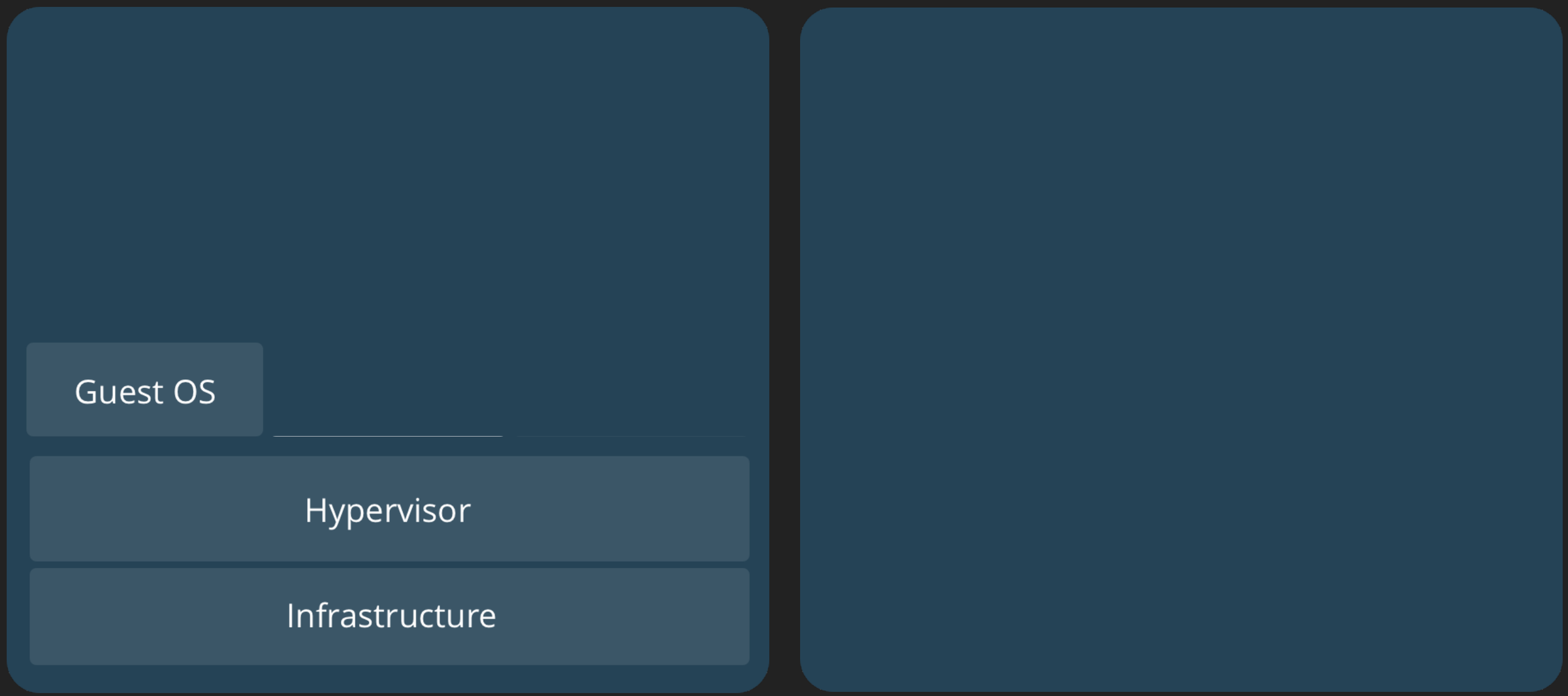
- Contact AO To Allocate Space And Install CentOS
- Log In With SSH
- Change Root Password
- Set Up Fail2Ban
- Configure Firewall
- Install And Configure Java, Apache Tomcat, ColdFusion, MySQL
- Take Periodic Snapshots
- Git Clone My Repo
- Configure Separate Directories For Each Specific Judge
- Set Up A Custom Domain
- Fly To District To Spot Check Data And Train Administrators
- Final Hand Off

PROBLEMS WITH THAT

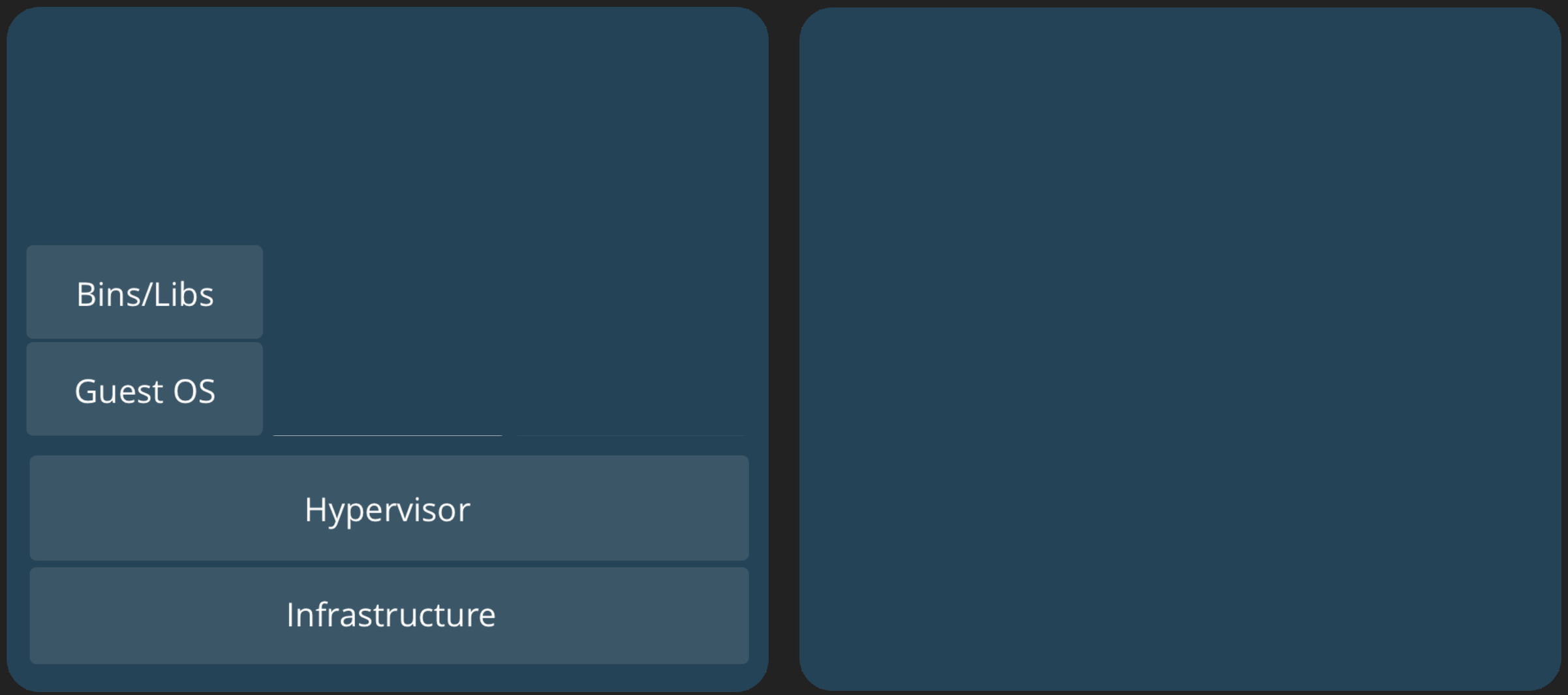
- It takes a week.
- We don’t control other districts’ hardware.
- It takes even more work to set up failover solutions.
- Setting up directories of static code per judge doesn’t make sense.
- This all doesn’t leave very much time for future development efforts.

INFRASTRUCTURE AS CODE

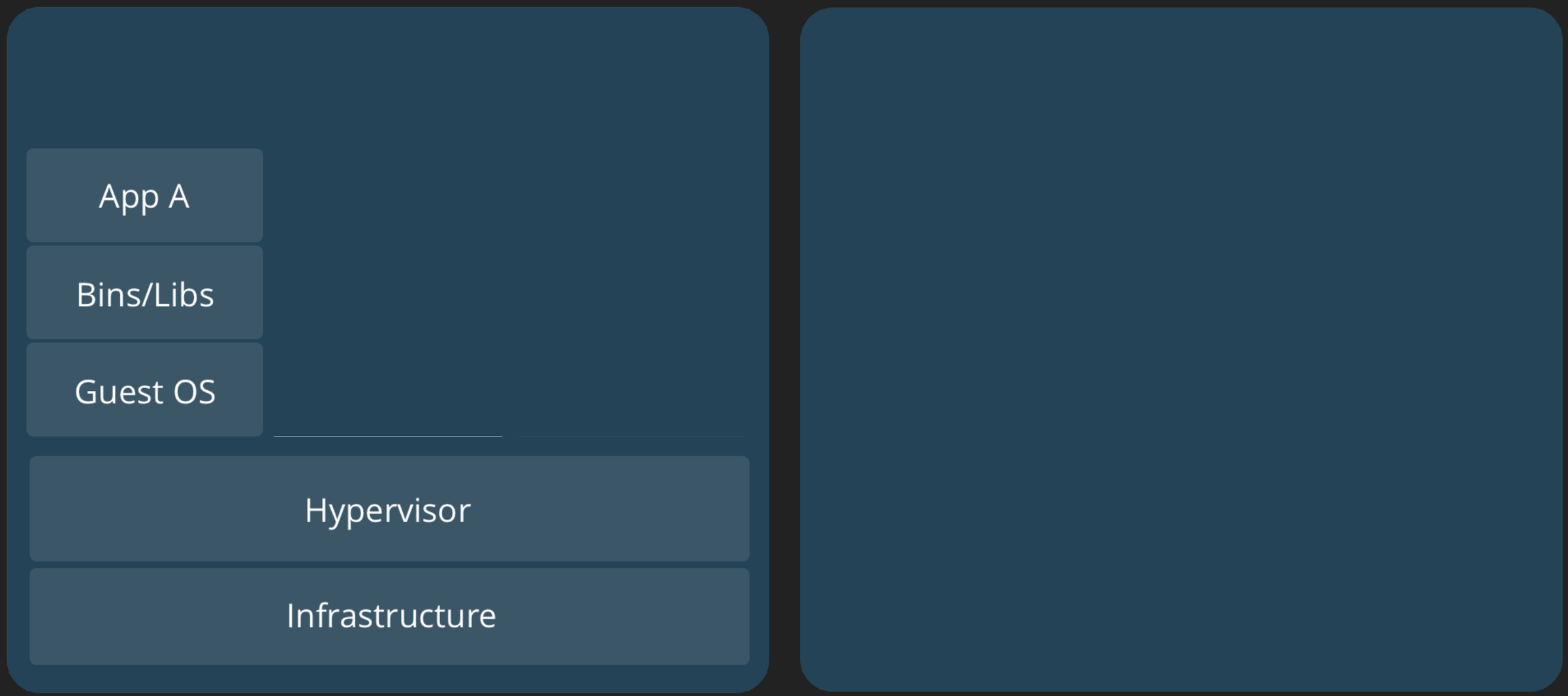
INFRASTRUCTURE AS CODE



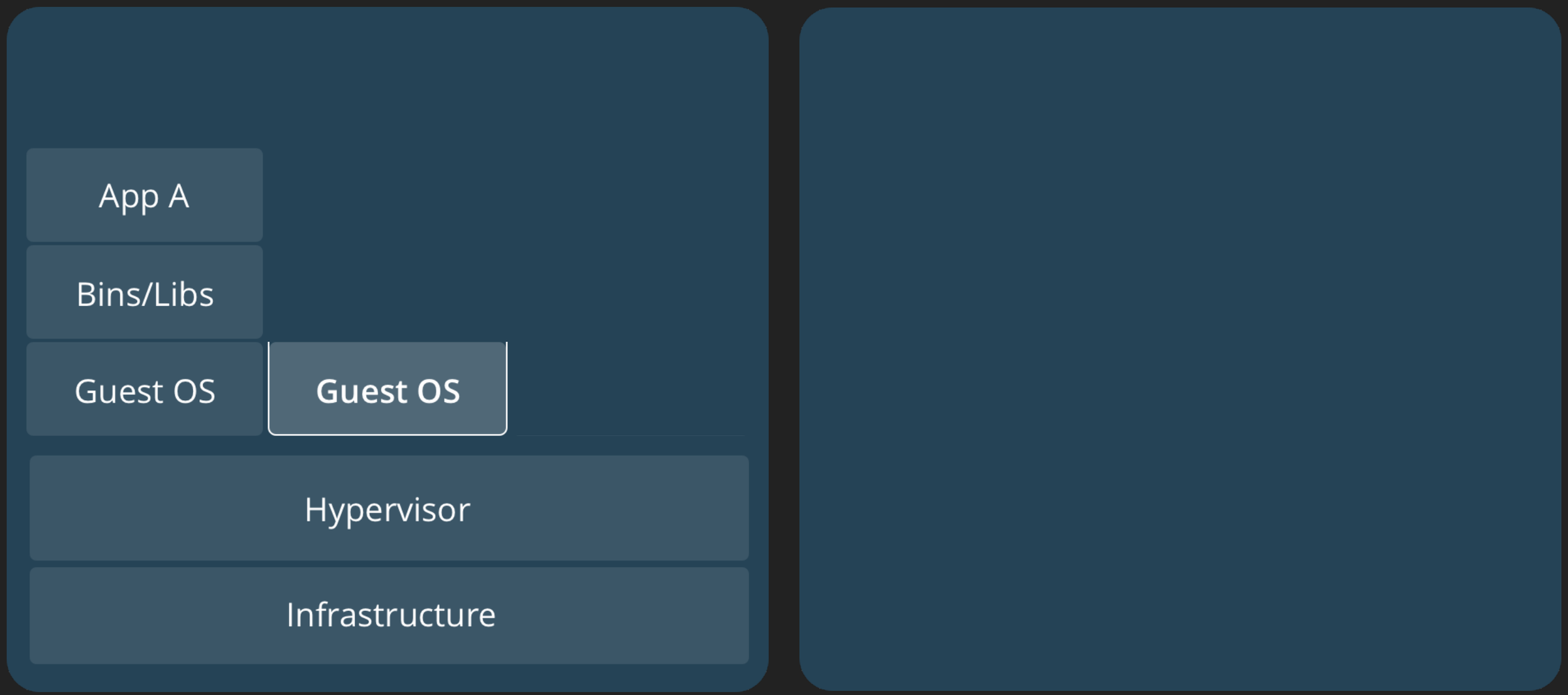
INFRASTRUCTURE AS CODE



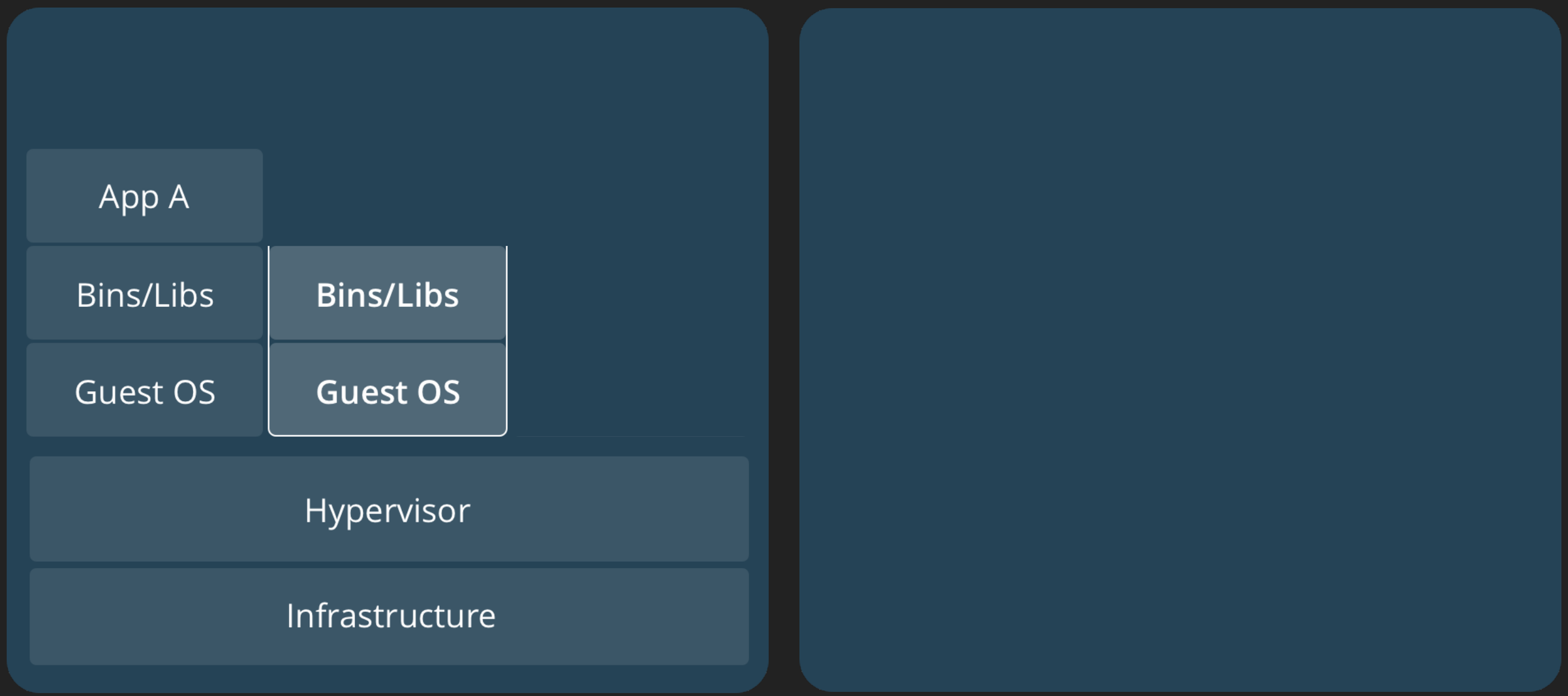
INFRASTRUCTURE AS CODE



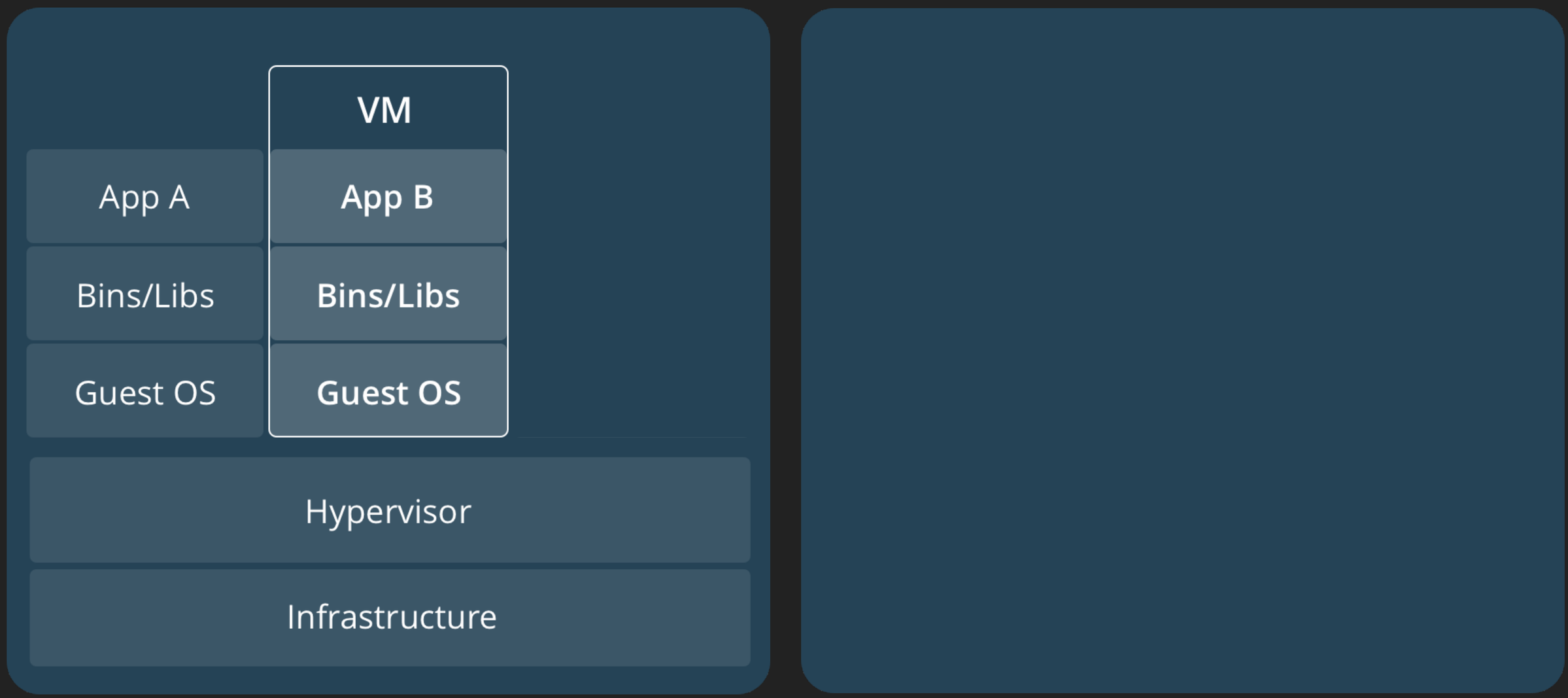
INFRASTRUCTURE AS CODE



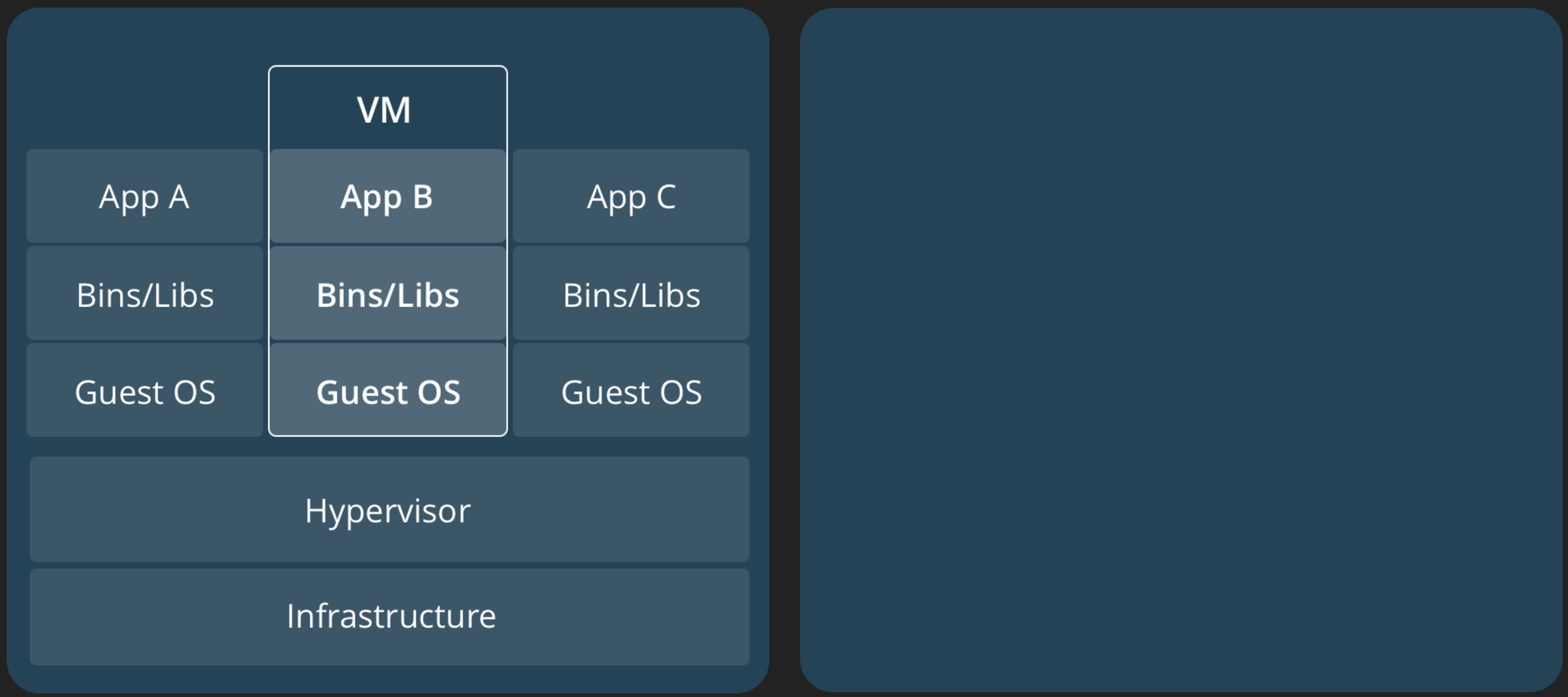
INFRASTRUCTURE AS CODE



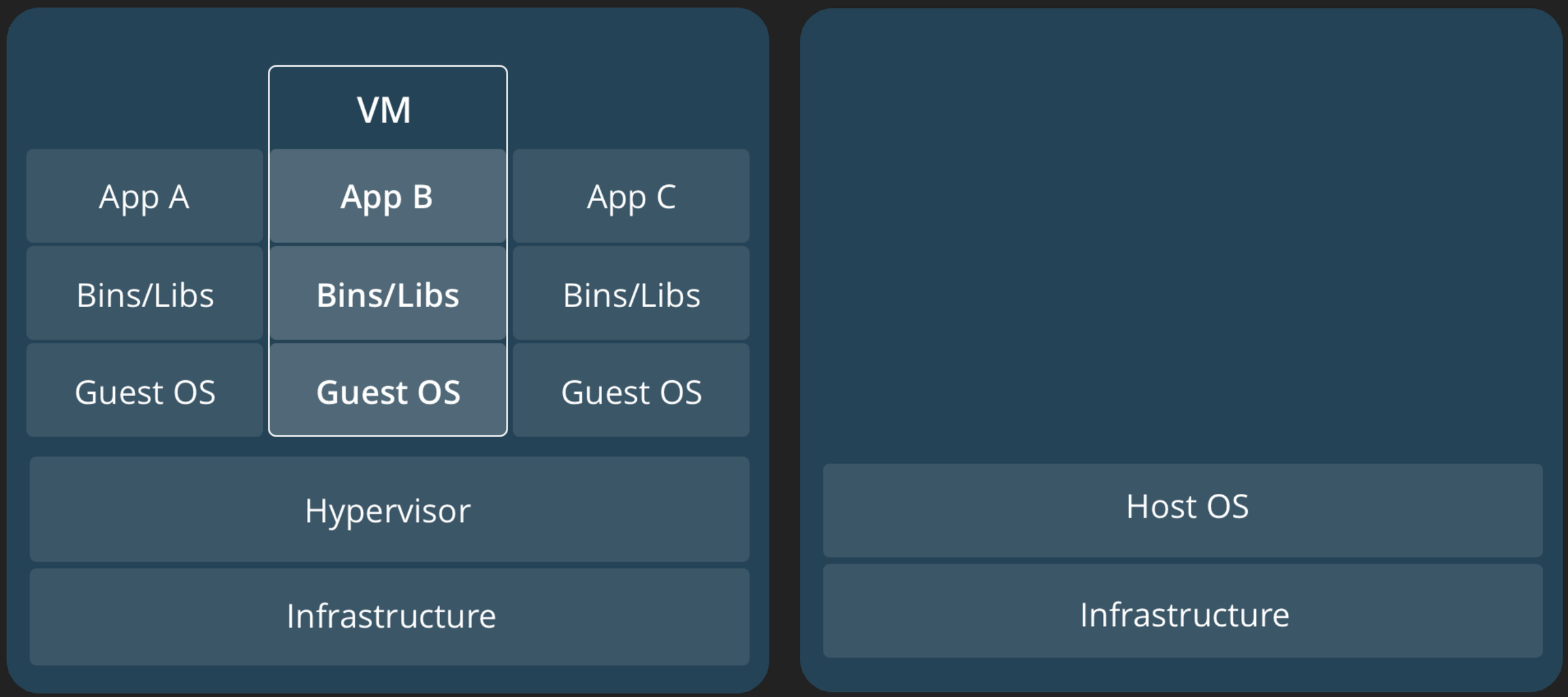
INFRASTRUCTURE AS CODE



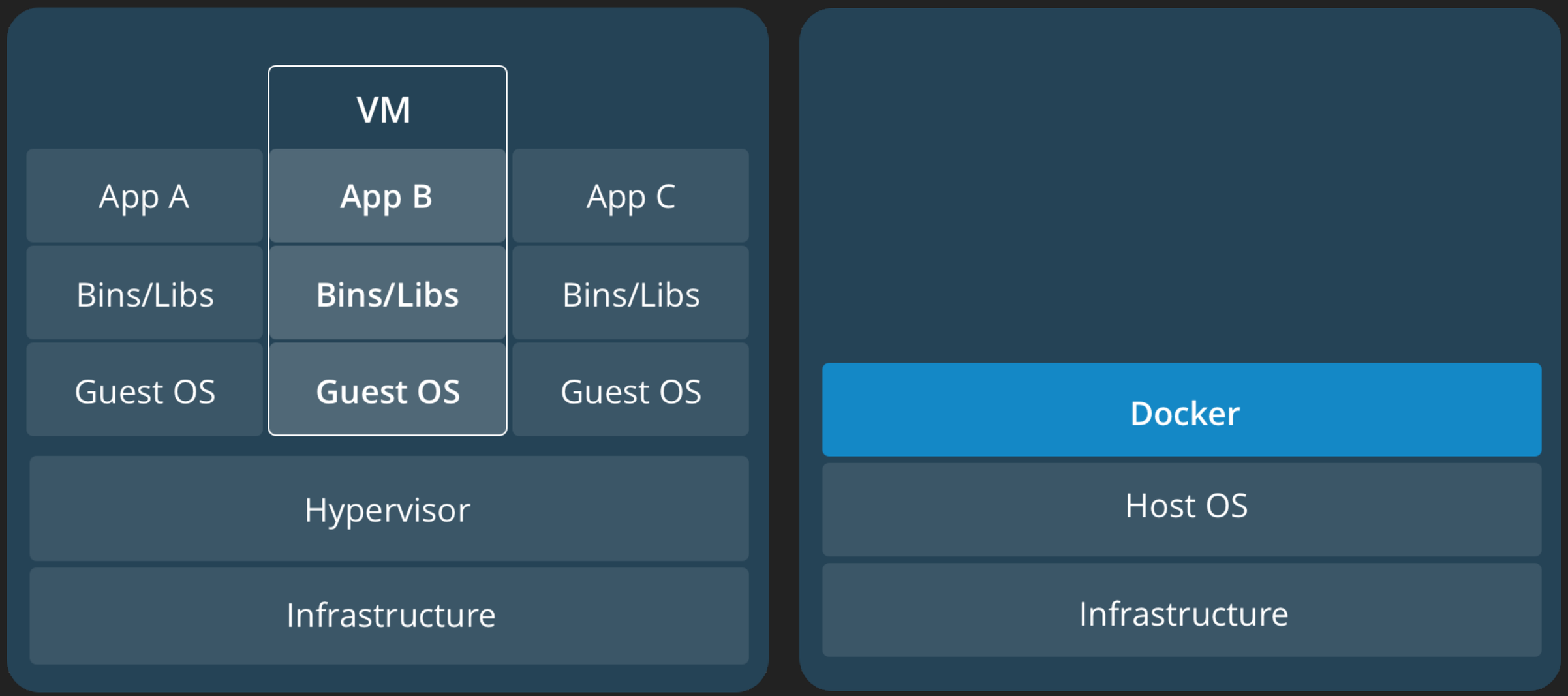
INFRASTRUCTURE AS CODE



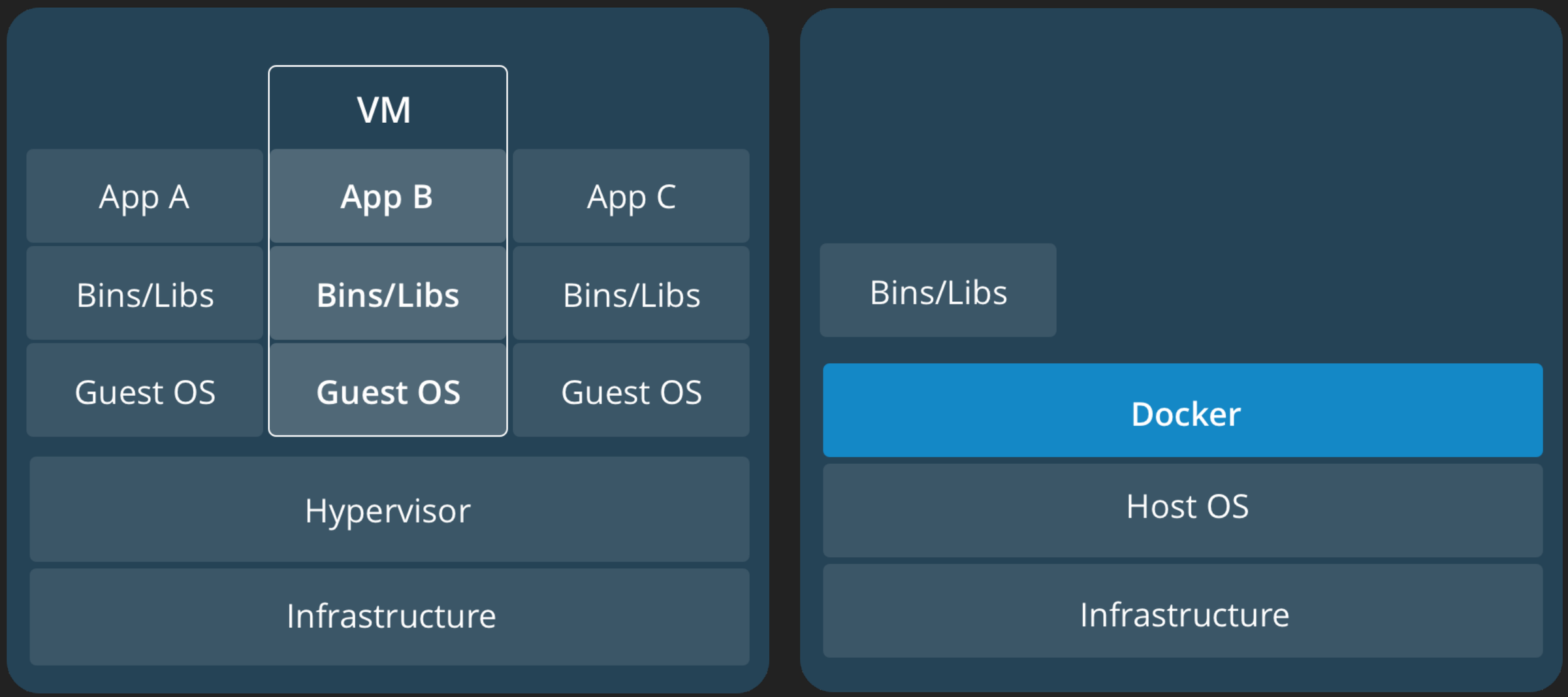
INFRASTRUCTURE AS CODE



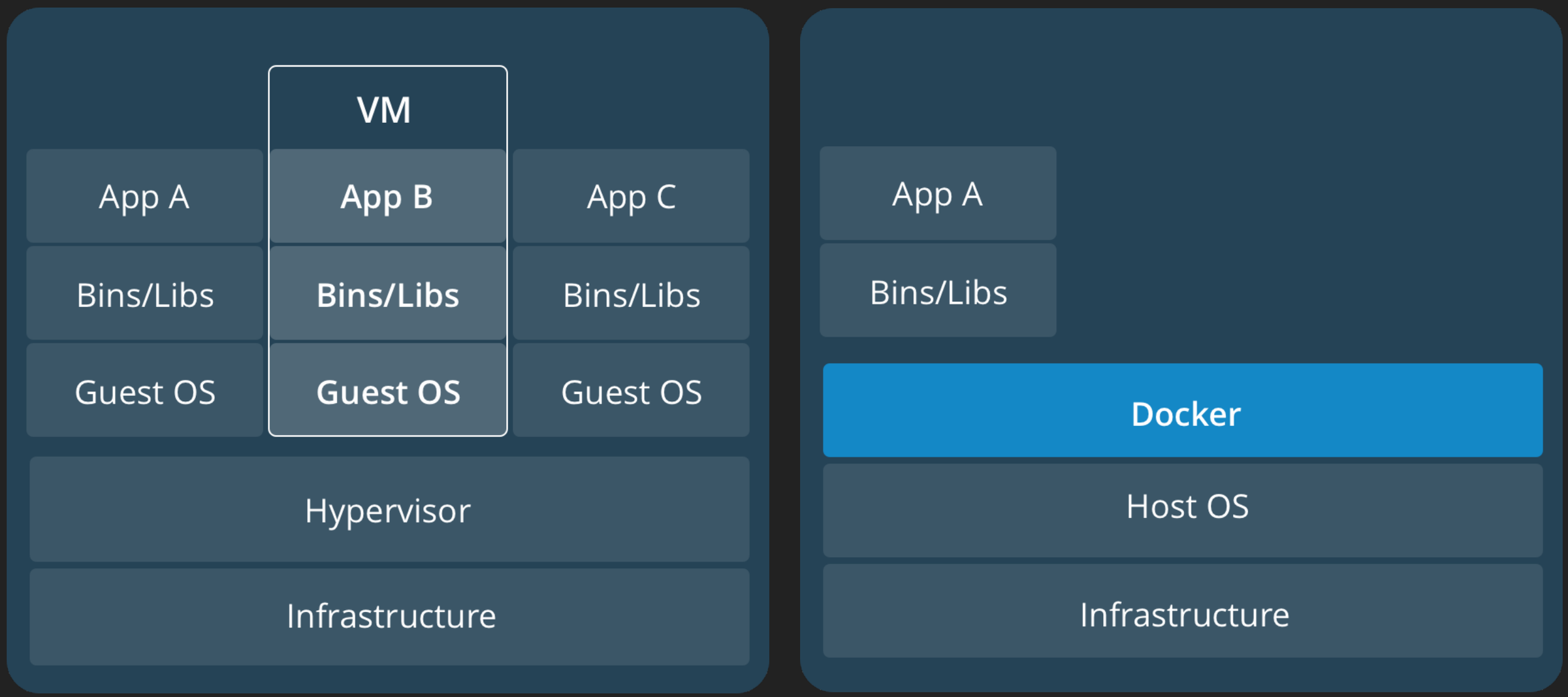
INFRASTRUCTURE AS CODE



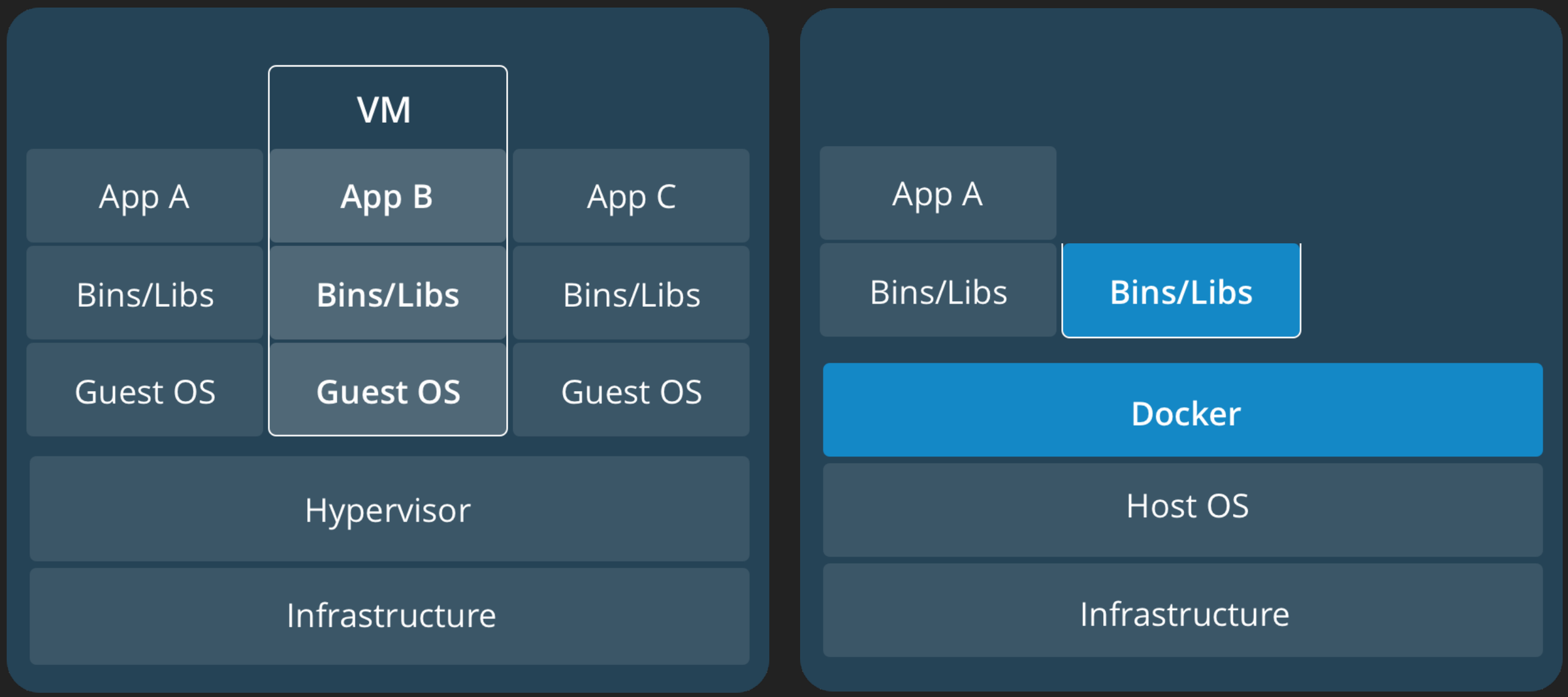
INFRASTRUCTURE AS CODE



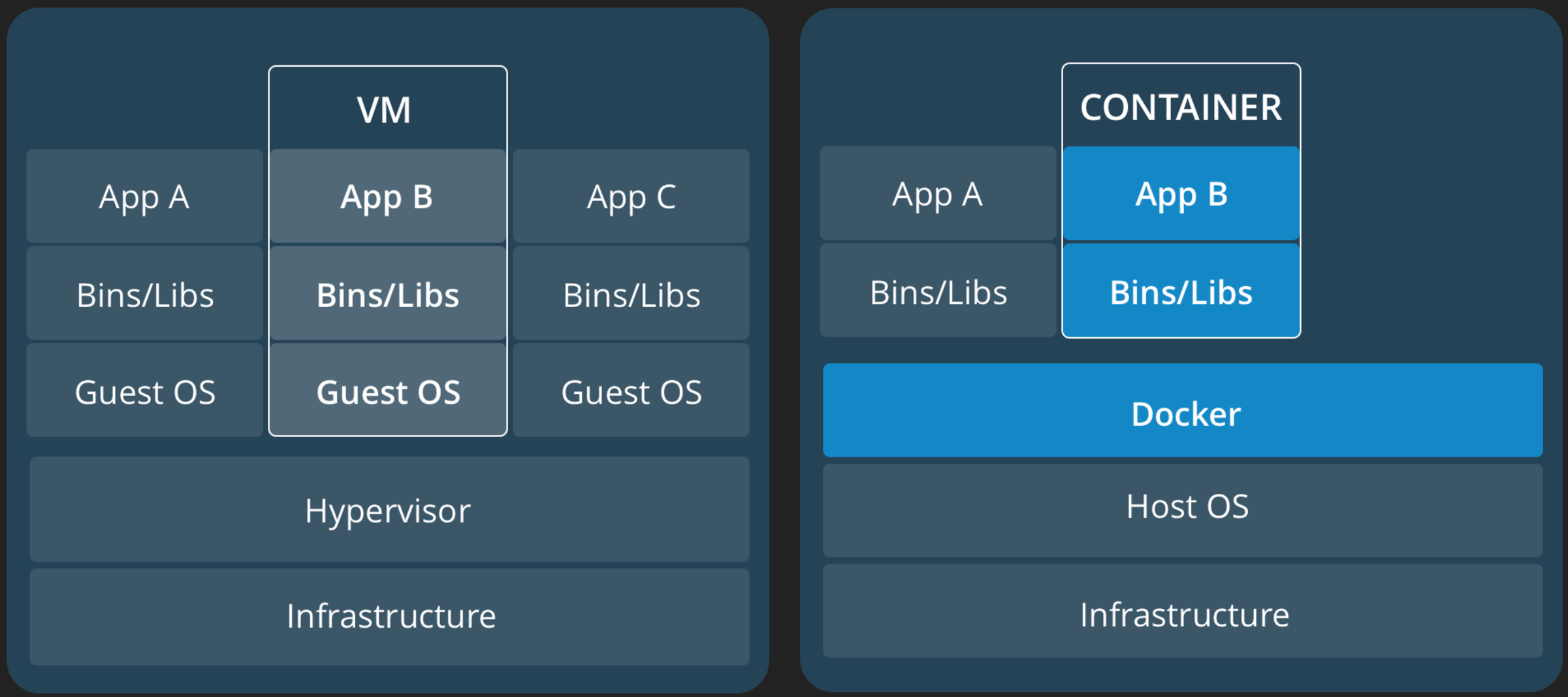
INFRASTRUCTURE AS CODE



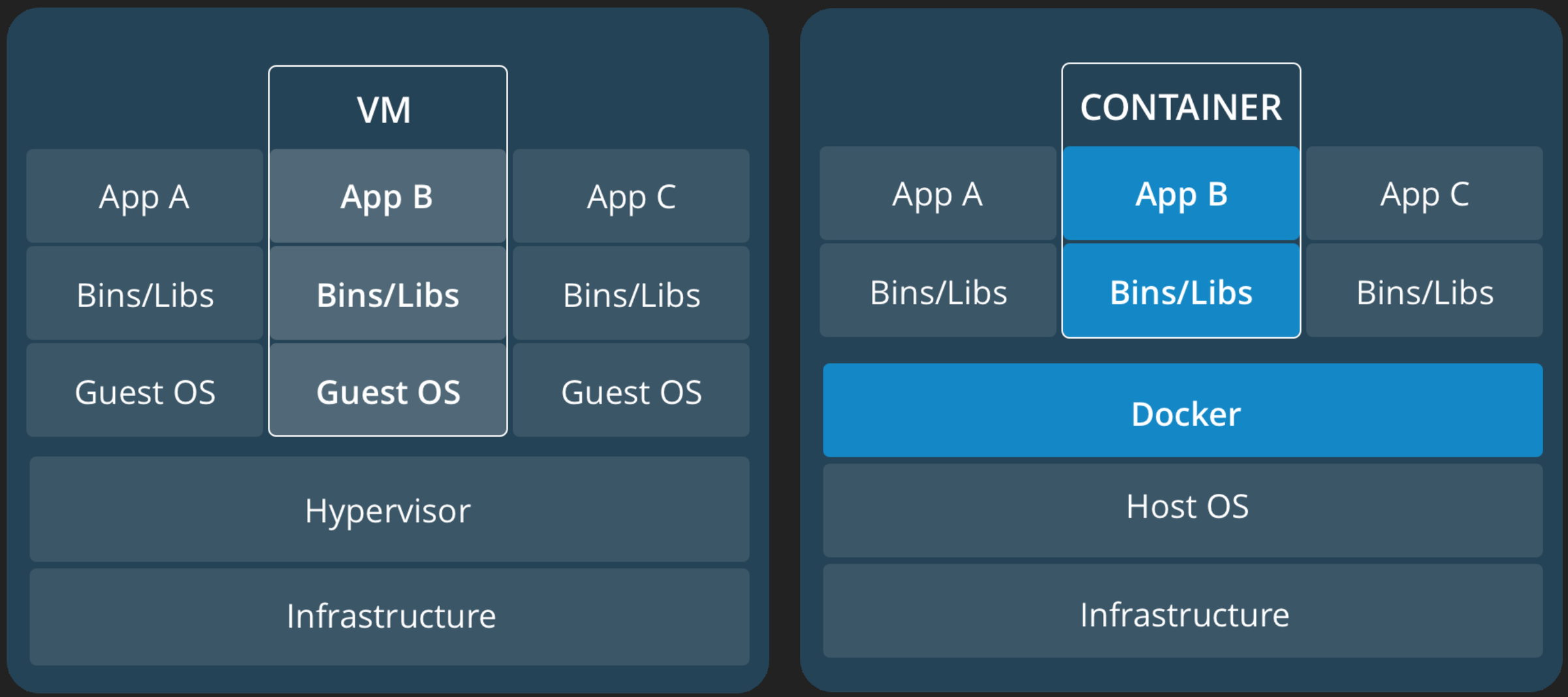
INFRASTRUCTURE AS CODE



INFRASTRUCTURE AS CODE



INFRASTRUCTURE AS CODE



BROAD OVERVIEW

- Docker is a container management engine.
- Containers are builds that run the same on every platform.
- Containers are built from images, kind of like an .ISO file.
- Images are built using Dockerfiles.
- Dockerfiles are just text files with lists of instructions on how to build, configure, and run your environments.

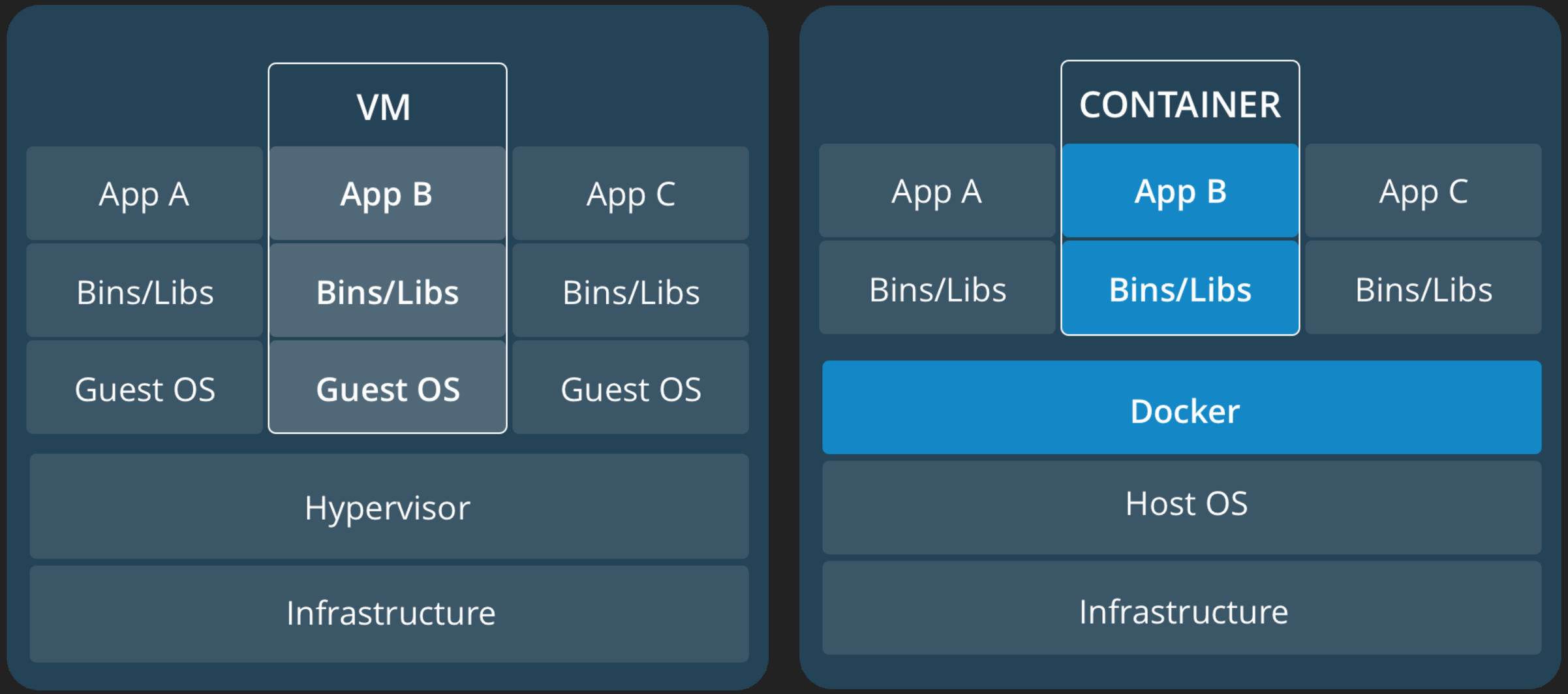
DOCKERFILES (PSEUDOCODE)

```
FROM lucee/lucee5:5.0.1.85

COPY config/lucee/setenv.sh /usr/local/tomcat/bin/

COPY config/lucee/lucee-web.xml.cfm /opt/lucee/web/

COPY config/lucee/lucee-server.xml /opt/lucee/server/
lucee-server/context/

ENV ELI_DSN_DRIVER=MySQL \
  ELI_DSN_CLASS=org.gjt.mm.mysql.Driver \
  ...

COPY code/ /var/www/
```

IMAGES

- An image is an inert, immutable file that's essentially a snapshot of a container.
- Images are created from Dockerfiles, which are made of instructions that build layers on top of a metaphorical system.
- Running "build" on a Dockerfile creates an image, while running "run" on an image spawns an instance of that image.
- A running instance of an image is called a container, of which you can have multiple based on the same image.

CONTAINERS


VM
App A
App B
App C
Bins/Libs
Bins/Libs
Bins/Libs
Guest OS
Guest OS
Guest OS
Hypervisor
Infrastructure
CONTAINER
App A
App B
App C
Bins/Libs
Bins/Libs
Bins/Libs
Docker
Host OS
Infrastructure

BUILDING AN IMAGE

- $ docker build –rm -t quinncuatro/eli-lucee .
- Boring Nerd Stuff ———–>

```
Status: Downloaded newer image for lucee/lucee5:5.0.1.85
# Executing 1 build trigger
 ---> Running in 505fd46acd64
Removing intermediate container 505fd46acd64
 ---> a37b7ca2c64f
Step 2/7 : LABEL author="Henry Quinn <henryquinniv@gmail.com>"
 ---> Running in 2034564b3357
Removing intermediate container 2034564b3357
 ---> d434d44acbd1
Step 3/7 : COPY config/lucee/setenv.sh /usr/local/tomcat/bin/
 ---> f6c501786550
Step 4/7 : COPY config/lucee/lucee-web.xml.cfm /opt/lucee/web/
 ---> c6d02a7998ab
Step 5/7 : COPY config/lucee/lucee-server.xml /opt/lucee/server/lucee-server/
context/
 ---> 7e042dc479c0
Step 6/7 : ENV ELI_DSN_DRIVER=MySQL   ELI_DSN_CLASS=org.gjt.mm.mysql.Driver
 ELI_DSN_CONNECTIONSTRING=jdbc:mysql://eli-db:3306/eli?useUnicode=true&charac
terEncoding=UTF-8&useLegacyDatetimeCode=true   ELI_DSN_DATABASENAME=eli   ELI
_DSN_USERNAME=root   ELI_DSN_PASSWORD=123456   LUCEE_TIMEZONE=America/New_Yor
k
 ---> Running in 5a20bb537337
Removing intermediate container 5a20bb537337
 ---> 677c2f84fce5
Step 7/7 : COPY code/ /var/www/
 ---> d0643da52e54
Successfully built d0643da52e54
Successfully tagged quinncuatro/eli-lucee:latest
```

USING AN IMAGE TO MAKE A CONTAINER

- $ docker images | grep eli-lucee

```
Henrys-MacBook-Pro:lucee quinncuatro$ docker images | grep eli-lucee
quinncuatro/eli-lucee   latest              d0643da52e54        2 minutes ago       643MB
```

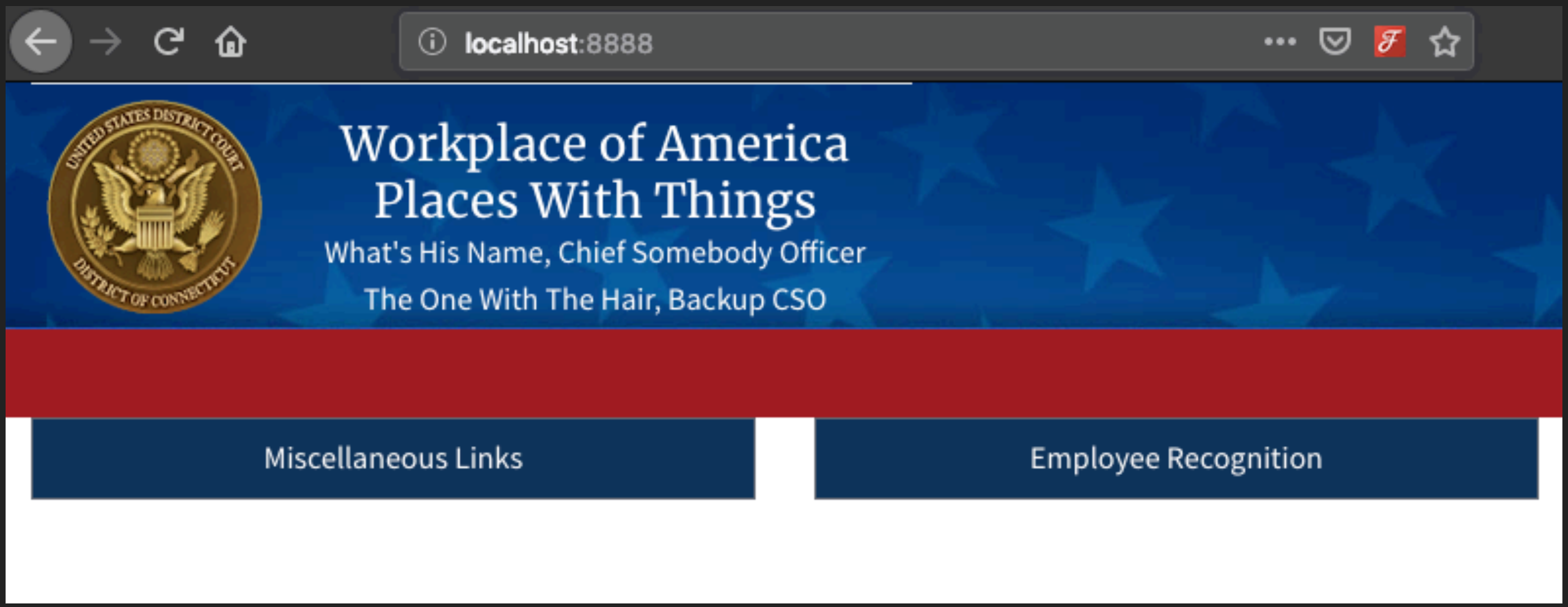
- $ docker run -d -p 127.0.0.1:8888:8888 --restart always --name eli-lucee --mount type=bind,source=$(pwd)/code,target=/var/www quinncuatro/eli-lucee:latest

```
Henrys-MacBook-Pro:lucee quinncuatro$ docker run -d -p 127.0.0.1:8888:8888 --restart always --name eli-lucee --mount type=bind,source=$(pwd)/code,target=/var/www quinncuatro/eli-lucee:latest
d306882d378cf30c1c72d4a007ae7da8d2174b449453339aff766f91355c339c
```

VERIFYING THAT THE CONTAINER IS RUNNING

- $ docker ps -a | grep eli-lucee

```
Henrys-MacBook-Pro:lucee quinncuatro$ docker ps -a | grep eli-lucee
d306882d378c        quinncuatro/eli-lucee:latest   "catalina.sh run"   2 minutes ago
    Up 2 minutes        8080/tcp, 127.0.0.1:8888->8888/tcp   eli-lucee
```



PROBLEMS WITH THAT

- It’s only one container.
- It’s not connected to a database.
- How would we even persist the database container’s data?
- If we set a database container up, how do we make the containers talk?

DOCKER COMPOSE YAML

```yaml
version: "3.5"
services:
  lucee-eli:
    build:
      context: ./lucee
      dockerfile: Dockerfile
    container_name: eli-lucee
    ports:
      - “8888:8888"
    networks:
      - eli-net
  db-eli:
    build:
      context: ./mysql
      dockerfile: Dockerfile
    container_name: eli-db
    volumes:
      - eli_dbdata:/var/lib/mysql
    networks:
      - eli-net
volumes:
  eli_dbdata:
networks:
  eli-net:
    driver: bridge
```

CREATING MULTIPLE CONTAINERS WITH DOCKER-COMPOSE

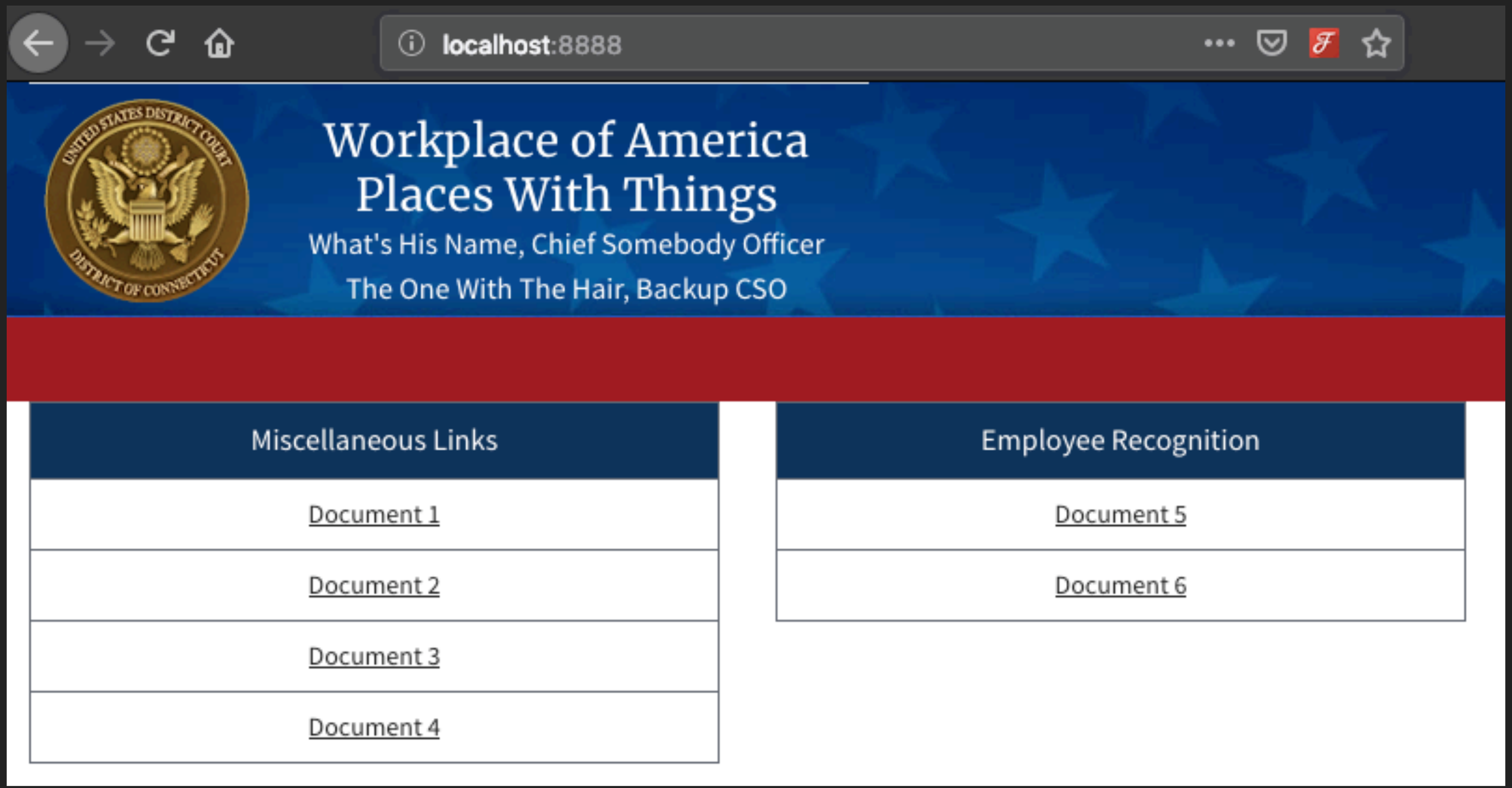
- $ docker-compose up -d

```
Henrys-MacBook-Pro:DEVC quinncuatro$ docker-compose up -d
Creating eli-db    ... done
Creating eli-lucee ... done
```

- $ docker ps -a

```
Henrys-MacBook-Pro:DEVC quinncuatro$ docker ps -a
CONTAINER ID        IMAGE                  COMMAND                  CREATED             ST
ATUS                PORTS                                NAMES
f01c4197e715        devc_lucee-eli         "catalina.sh run"        6 seconds ago       Up
 5 seconds          8080/tcp, 0.0.0.0:8888->8888/tcp     eli-lucee
f3551800c77a        devc_db-eli            "docker-entrypoint.s…"   6 seconds ago       Up
 5 seconds          3306/tcp                             eli-db
```

VERIFYING THAT THE CONTAINERS ARE RUNNING



NOW LETS ITERATE EVEN FASTER

- In spinup.sh (a bash script that executes different sets of Docker commands for me) there's a line:
- $ docker run -d -p 127.0.0.1:8888:8888 –restart always –name eli-lucee –mount type=bind,source=$(pwd)/lucee/code,target=/var/www –network eli-net devc_lucee-eli:latest
- –mount type=bind,source=$(pwd)/lucee/code,target=/var/www

PLATFORM AGNOSTIC

- Docker can generate containers that run the same no matter where they are, whether it be:
  - Windows 10/Server
  - OS X >= 10.11 (El Cap)
  - CentOS, Debian, Fedora, RHEL, SUSE, Ubuntu
  - AWS
  - Azure
  - GCE

PLATFORM AGNOSTIC

- Now, why is that a benefit?
  - You can spin your apps and data layers up REALLY fast.
  - No hand configs when trying out new tools, just “$ docker run ${whatever}” and go.
  - Develop and deploy with the exact same point releases of operating systems and dependencies.
  - Ensures that your applications are isolated and segregated.
  - Use the same image through the entire CI process.

PLAN FOR TODAY

- What I Was Tasked With
- What Was Expected Of Me
  - VPS Overview (Virtual Private Server)
- What I Did Instead
  - Docker Overview

HELPFUL LINKS (DIGITAL OCEAN)

- Digital Ocean Referral Code (Free $10 Credit)
  - https://m.do.co/c/c4539d6703fe
- Digital Ocean CentOS Guides
  - https://www.digitalocean.com/community/tutorials/initial-server-setup-with-centos-7
  - https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-centos-7

HELPFUL LINKS (DOCKER)

- Install Docker CE (Community Edition)
  - https://docs.docker.com/engine/installation/
- Getting Started
  - Mac - https://docs.docker.com/docker-for-mac/
  - PC - https://docs.docker.com/docker-for-windows/
- Docker Labs - https://github.com/docker/labs