



Getting Started with Oracle on Docker

Sean Scott



25 years working with Oracle technology

UTOUG Board :: RAC SIG Board

Oracle OpenWorld :: Collaborate/IOUG :: Regional UG

RAC/MAA :: DR/HA :: TFA/AHF :: Exadata/ODA

Upgrades :: Migration :: Cloud :: Automation

DevOps :: Infrastructure as Code

Containers :: Virtualization





docker.
ORACLE

Running Oracle on Docker



docker
ORACLE

“Don’t use Docker for
Databases!”



“Don’t use Docker for Databases!”

★ Oracle Support for Database Running on Docker (Doc ID 2216342.1)

APPLIES TO:

Oracle Database - Enterprise Edition - Version 12.1.0.2 and later
Oracle Database - Standard Edition - Version 12.1.0.2 and later
Linux x86-64

PURPOSE

Clarify Oracle's support of Oracle Database running on Docker

SCOPE

For customers running Oracle Database (single instance configuration) in Docker containerized environments.

DETAILS


Oracle plans to certify the latest versions of Oracle Database to run in Docker containers which are built and supported with Oracle Linux as the host.

Additionally, Oracle does support customers running Oracle Database (single instance) in Docker containers running on Oracle Linux 7 with UEK4 (and later) or Red Hat Enterprise Linux 7. Docker binaries are available in the Addons channel for Oracle Linux. Details on Installation can be found in Chapter 2 of the [Oracle Linux 7 Docker Users Guide](#). Other details on support for Docker on Oracle Linux can be found in [Support for Docker Running on Oracle Linux \(Doc ID 1921163.1\)](#).

Oracle Database running in a Real Application Clusters (RAC) configuration in Docker containers is supported for Test and Development environments. The detailed steps to build and run Oracle RAC on Docker can be found [here](#). The prerequisite patch to successfully install Oracle RAC on Docker can be downloaded from [OTN](#).

“Don’t use Docker for Databases!”

↳ You Retweeted


 **Markus Michalewicz**
@KnownAsMarkus

Happy to announce that [#Oracle](#) RAC on [#Docker](#) is now available with full production [#support](#).

See the latest [@OracleMAA](#) blog post for more information & a brief outlook on future projects:

[blogs.oracle.com/maa/post/oracl...](https://blogs.oracle.com/maa/post/oracle-rac-on-docker-now-with-full-production-support)

/CC [@OracleRACpm](#) [@oraclemaasig](#) [@oracleace](#)



blogs.oracle.com
Oracle RAC on Docker - Now with Full Production Support
Oracle Database 21c (21.3) comes with full production support for Oracle RAC on Docker!

8:53 PM · Oct 5, 2021 · Twitter Web App



Learning the Terminology

(Images, containers & builds: oh my!)

Terminology...

- Images
- Containers
- Builds
- Dockerfiles
- Stateless/Stateful
- Ephemeral/Persistent
- Layers
- Overlay Filesystems
- Volumes
- Port Mappings
- Rootless



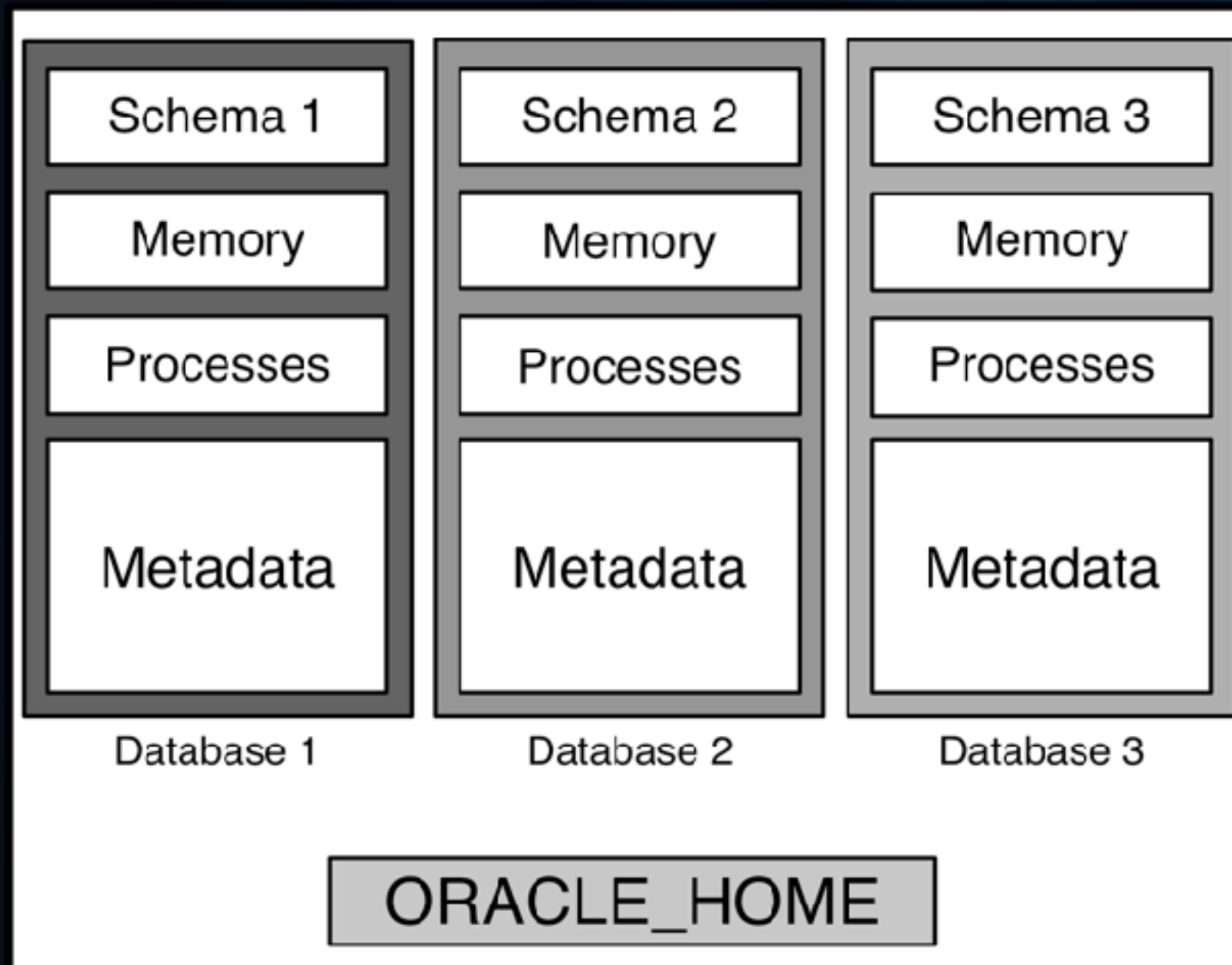
docker.
ORACLE

What is a Container?

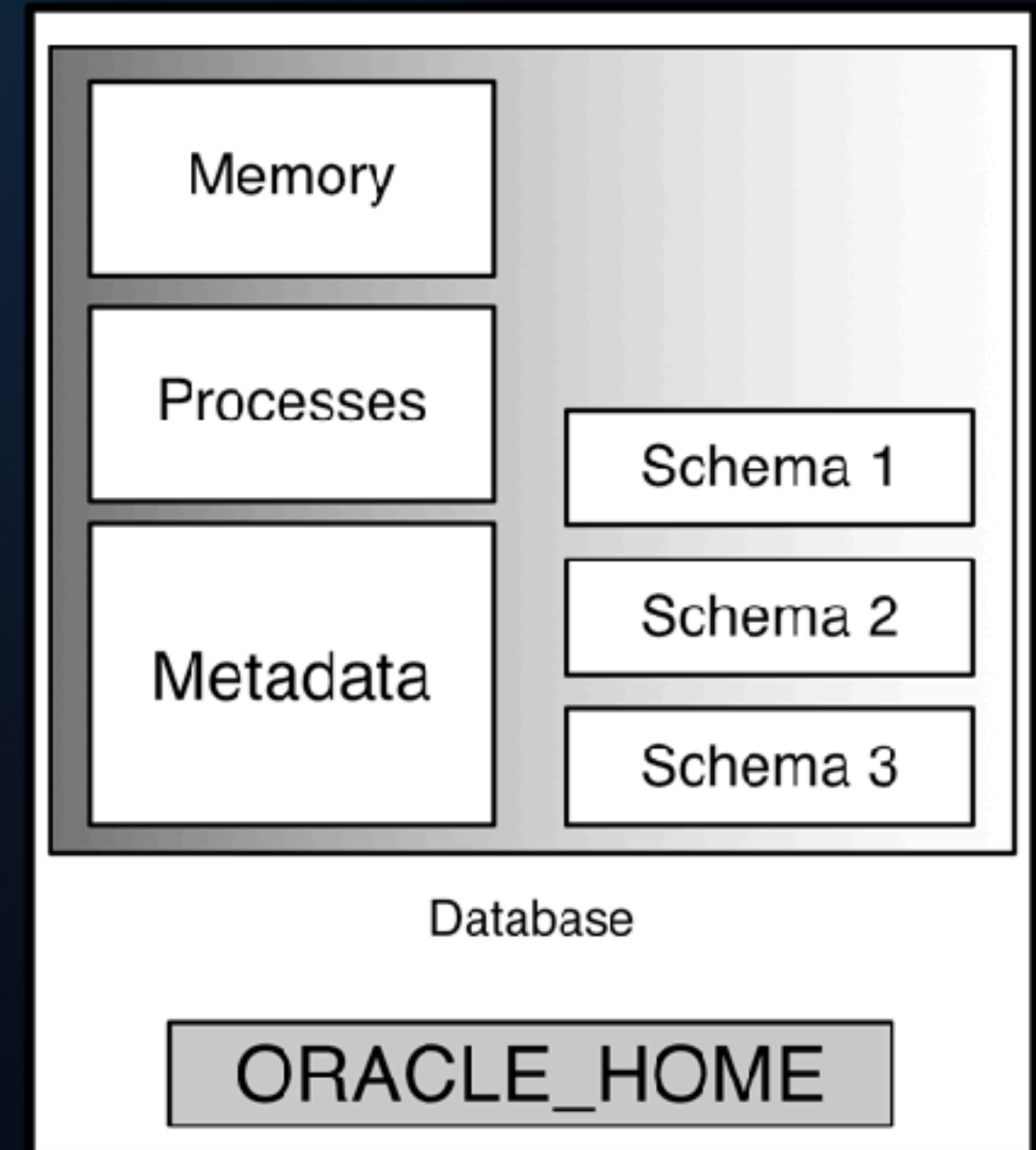
Containers are like Virtual Machines, right?



Three ORACLE_HOME vs Three Schemas

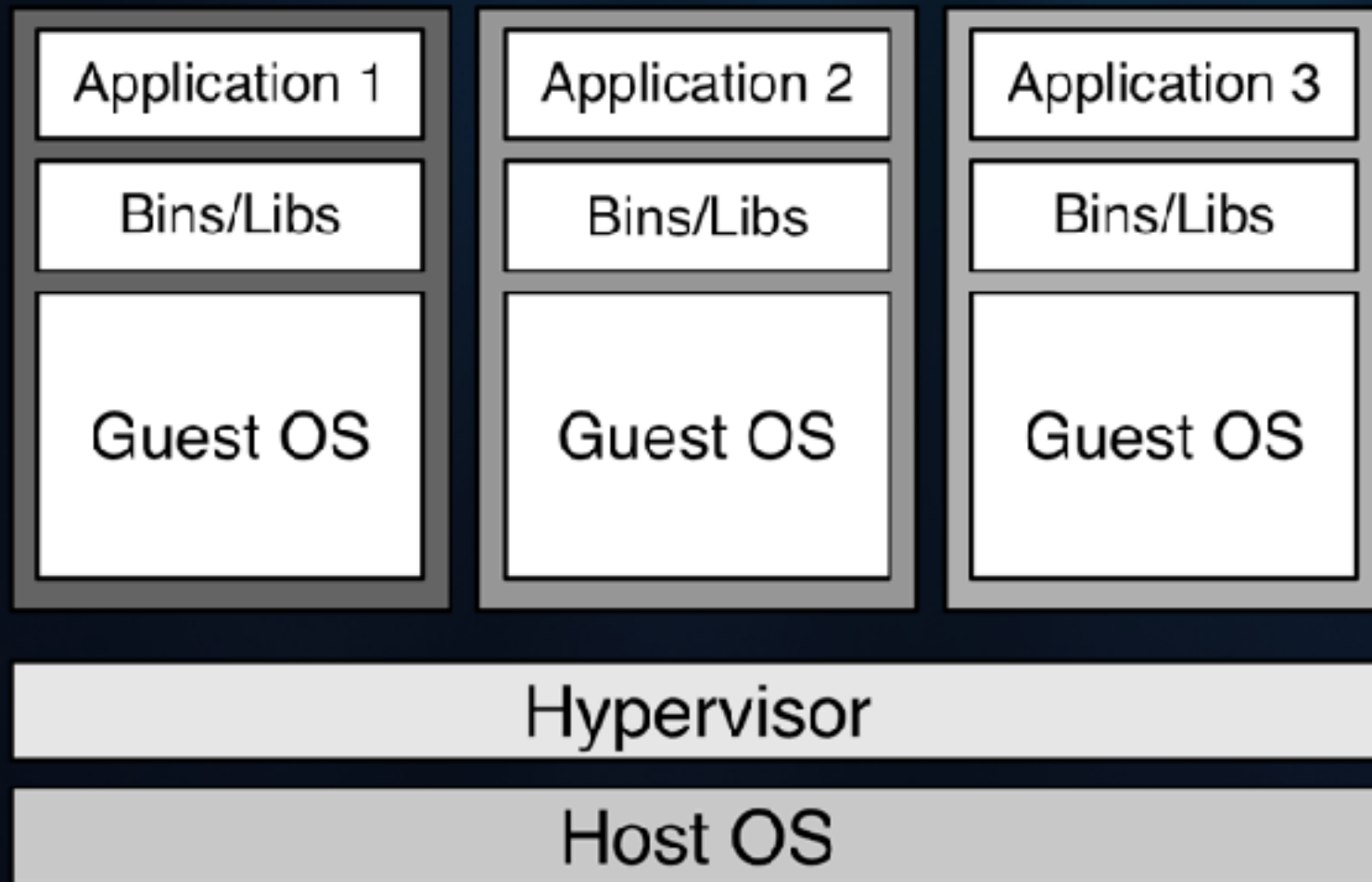


Three databases

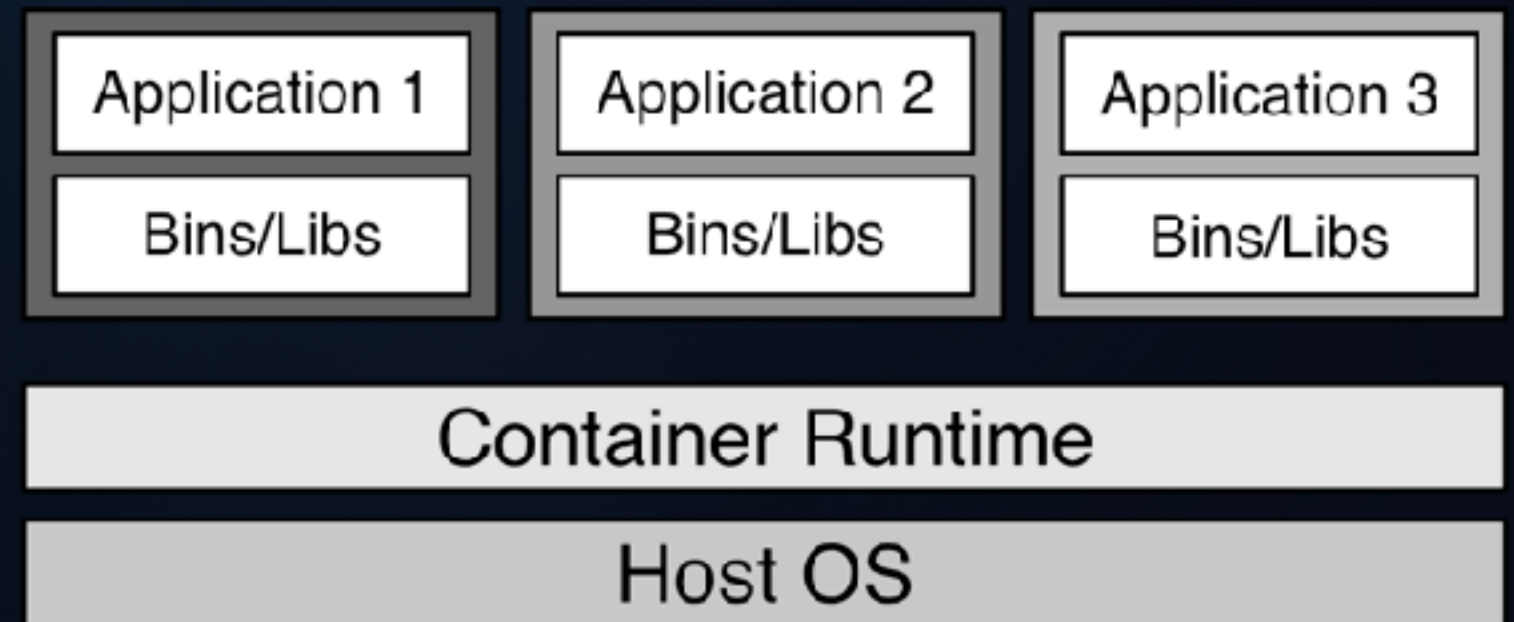


Three schemas

Virtual Machines vs. Containers

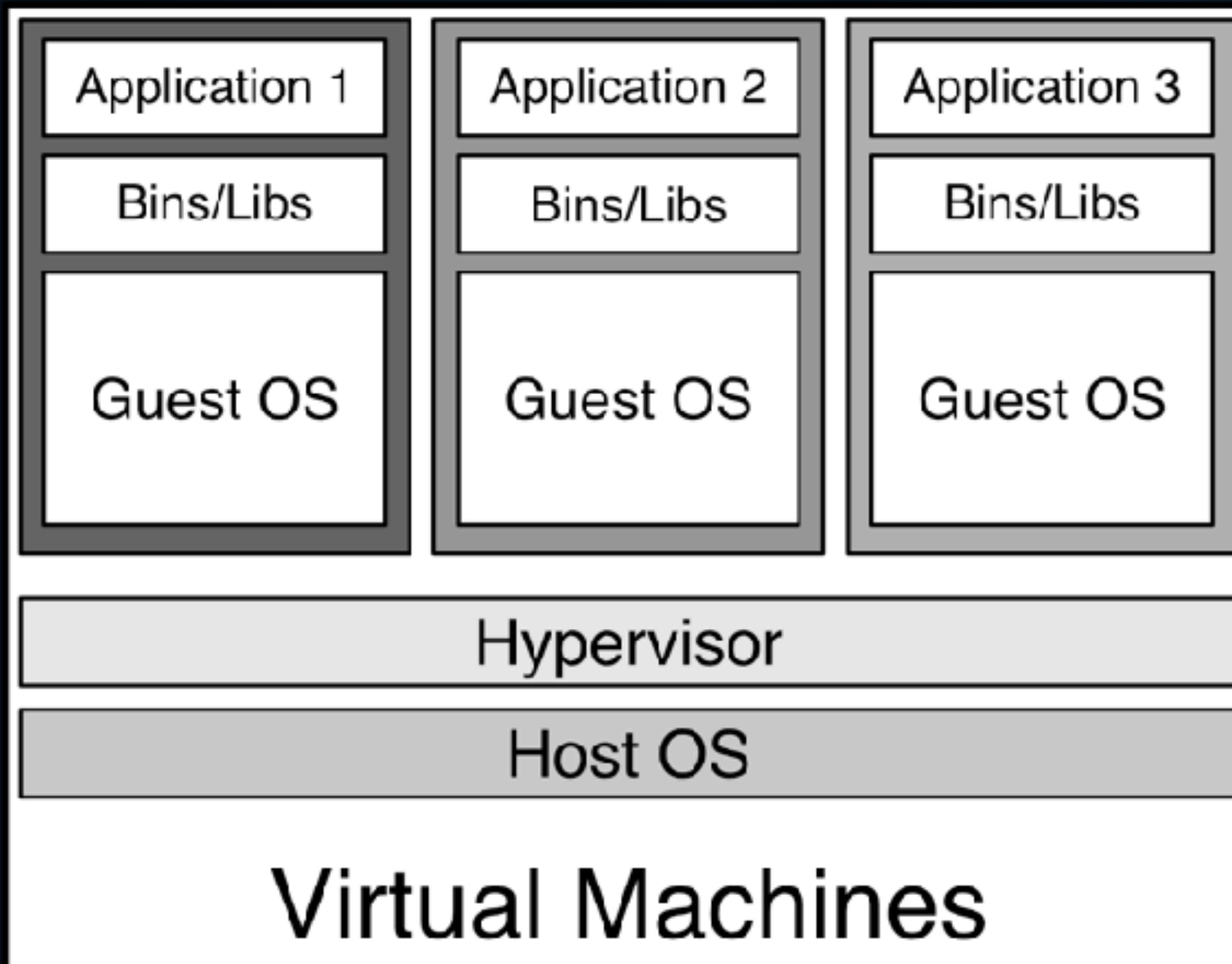


Three full operating systems

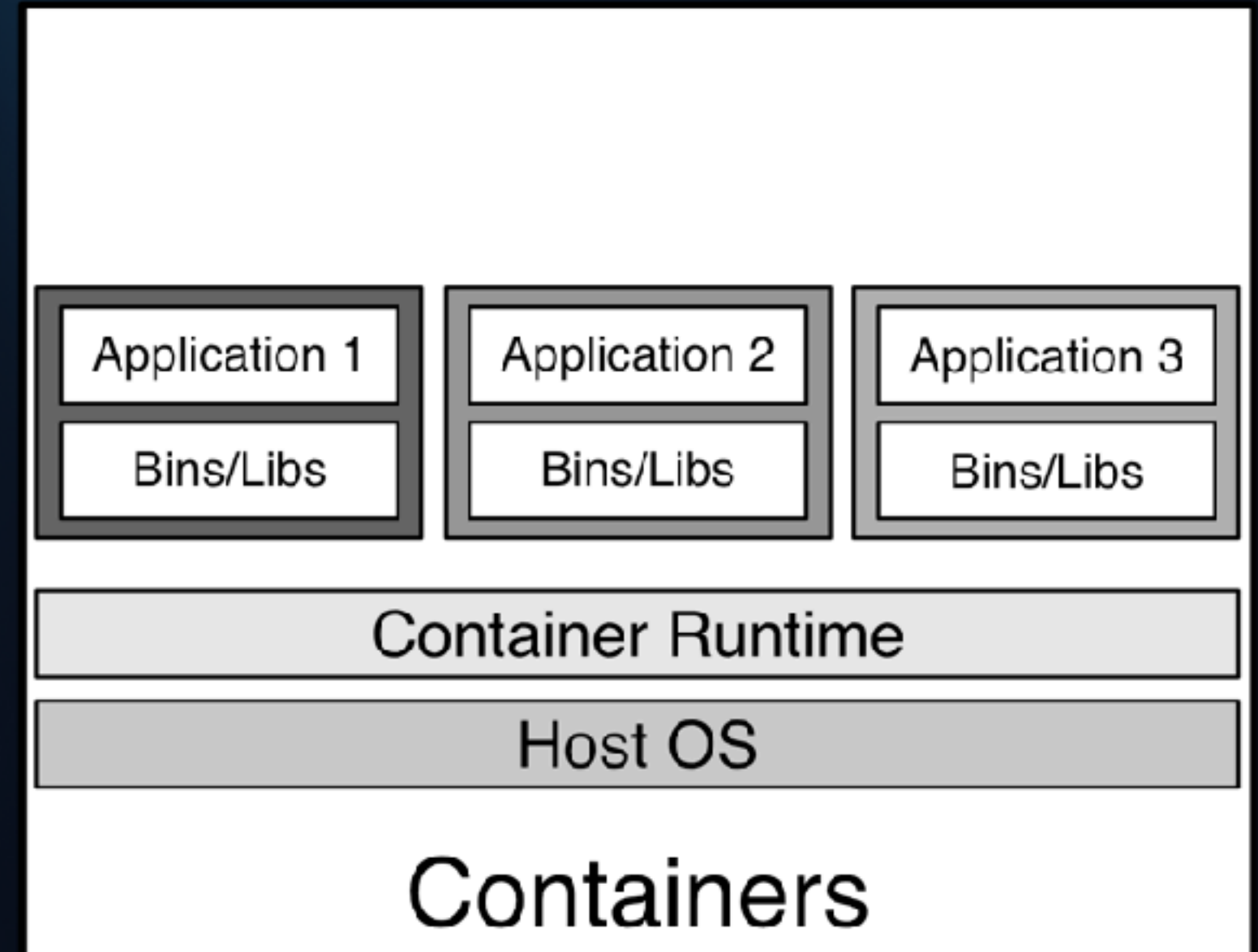


Three application/executable directories

Virtual Machines vs. Containers



Three full operating systems



Three application/executable directories

Containers:

- No hypervisor
- No operating system
- No boot/init process
- Processes on a host
- Isolation via namespaces
- Minimal dependencies

Virtual Machines:

- Hypervisor
- Operating system
- Must boot/init to use
- Virtualized on host
- Isolation through VM
- “Full” OS/filesystem



docker.
ORACLE

A Container?
From Scratch?

A Container From Scratch - unshare

Run a program with some namespaces unshared from parent
Namespaces:

- User
- Network
- Unix Time Sharing (UTS)
- Inter-Process Communication (IPC)
- Process Identifier (PID)
- Mount

A Container From Scratch - unshare

```
unshare --user --map-root-user --net --uts --fork \  
  --mount --mount-proc --pid --ipc \  
  chroot /home/opc/alpine /bin/ash
```

Build Your Own Containers

Linux containers in 500 lines of code:

<https://blog.lizzie.io/linux-containers-in-500-loc.html>

Boker—Docker in 100 lines of bash:

<https://github.com/p8952/bocker>

YouTube - Search “OracleSean”

Docker internals demos

Oracle on Docker



docker.
ORACLE

Containers are Fast



docker.
ORACLE

What is an Image?

Image:

- A set of directories, files
- Some JSON instructions
- Stateless
- “Run” an image to get a container

Containers:

- A Linux process
- Stateful
- Persistent
- The result of “running” an image

Seed Database / Template / Image

- Not a DB/document
- Standard (stateless)
- Create from seed/template
- DB/document changes don't change seed/template
- Many DB/documents can use a seed/template

Pluggable Database / Document / Container

- Independent of other PDB/documents
- Save, edit data (persistent)
- Drop DB, delete document (ephemeral)
- Use to create new seed/templates



Images are Stateless
(Can't be Changed)



Containers are Stateful
(Can be Changed)



docker.
ORACLE

Oracle on Docker: Why?

Container Advantages

Portable

Versionable

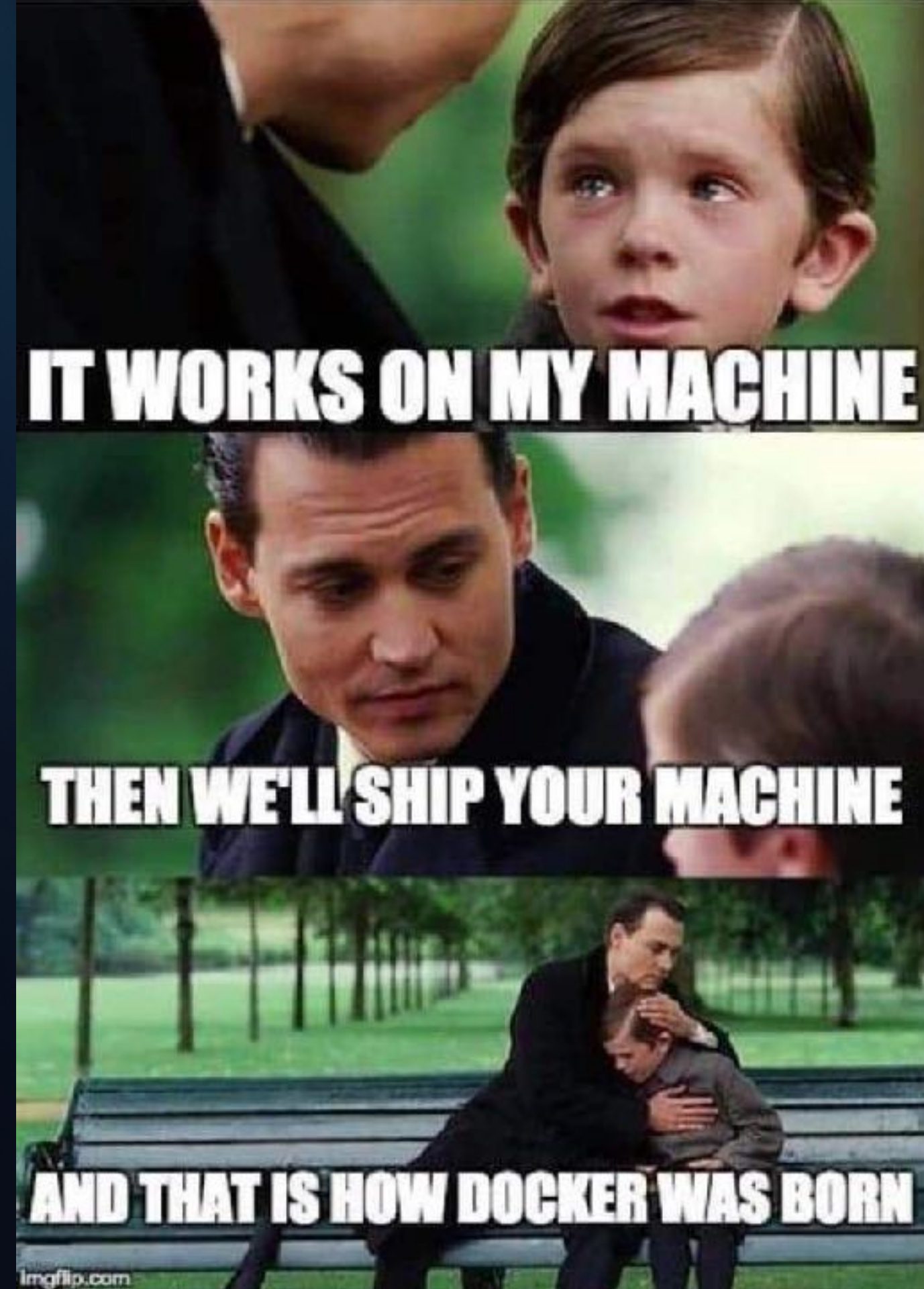
Distributable



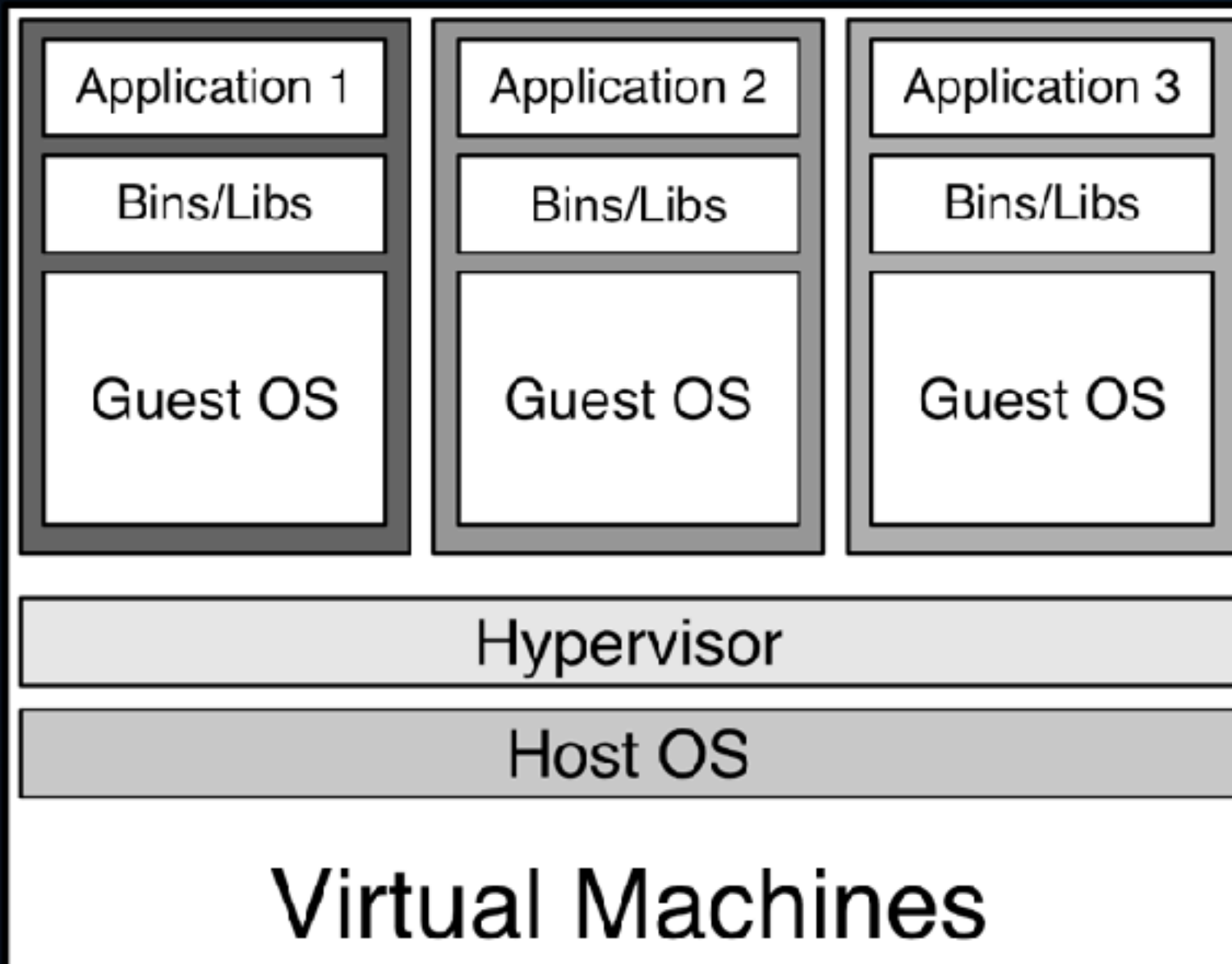
Container Advantages

Images built on <MY OS> will work on:

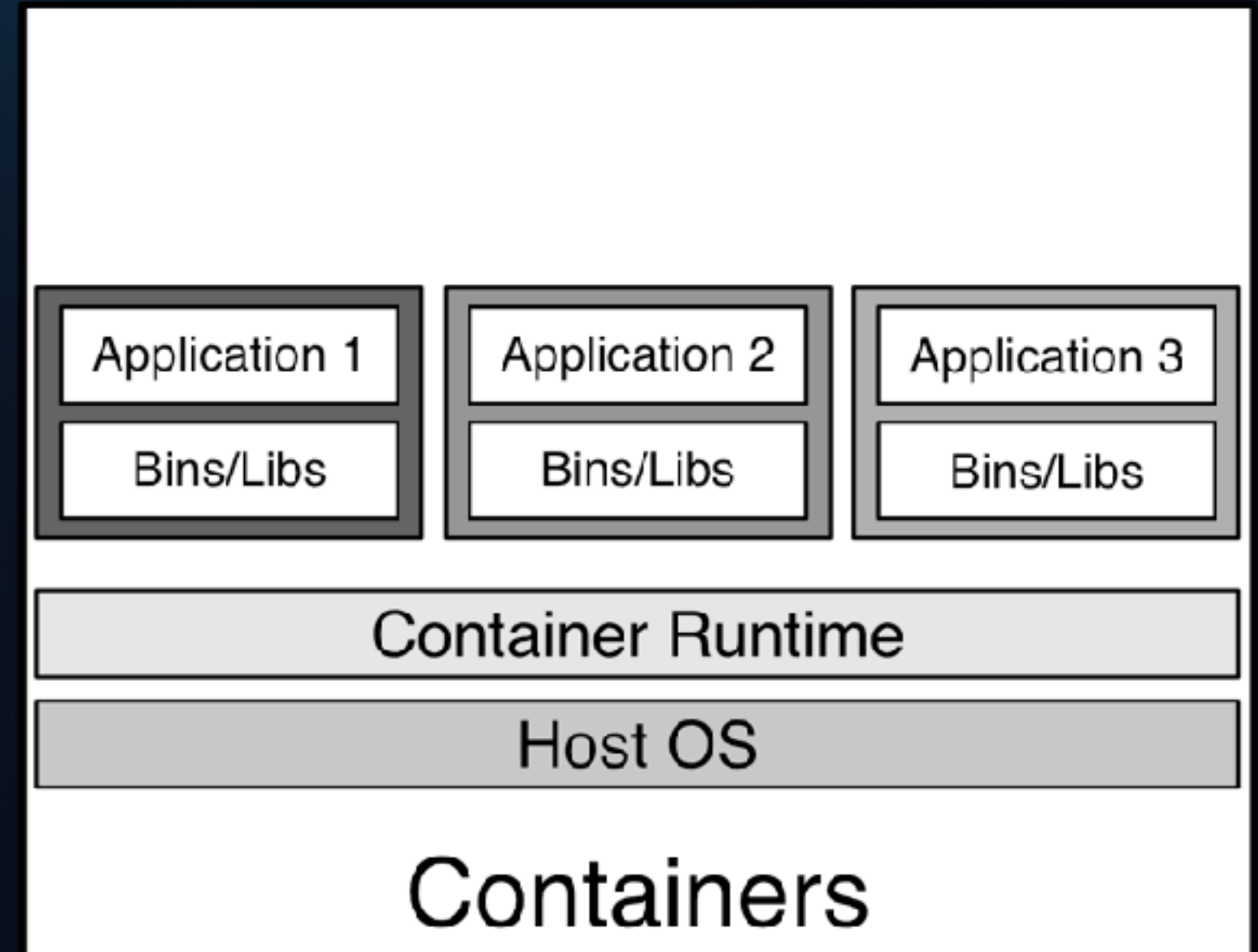
- Windows
- Mac
- Linux
- OCI
- Azure
- Docker
- Podman
- Compose
- Kubernetes
- AWS
- GCP



Virtual Machines vs. Containers



Three full operating systems



Three application/executable directories

Run Oracle on Docker to...

Eliminate dependencies

Develop SQL, PL/SQL, scripts

Hack, experiment, learn

Performance tuning

Test without infrastructure

Validate patches, upgrades

Build labs, training environments



What's possible?

Vanilla databases

Multitenant/Legacy

Data Guard

Far Sync

Real Application Clusters

Sharding

...and much, much more!



Vagrant, etc. may be better for...

Oracle Enterprise Manager

Real Application Clusters

AHF/TFA

ASM





docker.
ORACLE

Run Oracle on Docker



docker.
ORACLE



Run a Database Container

```
docker run -d oracle/database:19.3.0-ee
```

Run a Useful Database Container

```
docker run -d \  
  --name o193 \  
  -p 11521:1521 \  
  -v ~/oradata/o193:/opt/oracle/oradata \  
  -e ORACLE_SID=o193 \  
  oracle/database:19.3.0-ee
```

Run a Useful Database Container

```
docker run -d \  
  --name o193 \  
  -p 11521:1521 \  
  -v ~/oradata/o193:/opt/oracle/oradata \  
  -e ORACLE_SID=o193 \  
  oracle/database:19.3.0-ee
```

Run a Useful Database Container

```
docker run -d \  
  --name o193 \  
  -p 11521:1521 \  
  -v ~/oradata/o193:/opt/oracle/oradata \  
  -e ORACLE_SID=o193 \  
  oracle/database:19.3.0-ee
```

Run a Useful Database Container

```
docker run -d \  
  --name o193 \  
  -p 11521:1521 \  
  -v ~/oradata/o193:/opt/oracle/oradata \  
  -e ORACLE_SID=o193 \  
  oracle/database:19.3.0-ee
```

Run a Useful Database Container

```
docker run -d \  
  --name o193 \  
  -p 11521:1521 \  
  -v ~/oradata/o193:/opt/oracle/oradata \  
  -e ORACLE_SID=o193 \  
  oracle/database:19.3.0-ee
```


Run a Useful Database Container

```
docker run -d \  
  --name o193 \  
  -p 11521:1521 \  
  -v ~/oradata/o193:/opt/oracle/oradata \  
  -e ORACLE_SID=o193 \  
  oracle/database:19.3.0-ee
```



Environment Options

Customize Database Creation

Customize Oracle database creation

```
docker run ... \  
  -e ORACLE_PWD=<password>  
  -e ORACLE_SID=<sid>  
  -e ORACLE_PDB=<pdbname>  
  -e ORACLE_CHARACTERSET=<characterset>  
  -tz=<timezone>
```



docker.
ORACLE

Explore the Container

“Login” to the Container

```
docker exec -it oracledb bash
```



docker.
ORACLE

Connect Clients

Connect to Docker from SQL Developer

The screenshot shows the 'Oracle Docker Lab' connection configuration in SQL Developer. The 'User Info' tab is active, showing 'system' as the username and a masked password. The 'Details' tab is also visible, showing '158.123.45.6' as the hostname, '11521' as the port, and 'orclcdb' as the service name. Red arrows point from external text labels to these specific fields.

Name	Oracle Docker Lab	Color	
Database Type	Oracle		
User Info Proxy User			
Authentication Type	Default		
Username	system	Role	default
Password	<input checked="" type="checkbox"/> Save Password	
Connection Type	Basic		
Details Advanced			
Hostname	158.123.45.6		
Port	11521		
<input type="radio"/> SID			
<input checked="" type="radio"/> Service name	orclcdb		

Compute Public
IP Address

Mapped Port

ORACLE_SID



Building Images



docker.
ORACLE

DIY Steps

Install Docker Desktop/Docker Engine

Docker Desktop

- Free for Windows and Mac
- <https://docs.docker.com/get-docker/>



Docker Engine

- Free for Linux
- <https://docs.docker.com/engine/>

<https://www.oracle.com/database/technologies/oracle-database-software-downloads.html>

Download Oracle database installation files for Linux x86-64

19.3

Name	Download
Microsoft Windows x64 (64-bit)	 ZIP (2.9 GB)
Linux x86-64	 ZIP (2.8 GB)  RPM (2.5 GB)

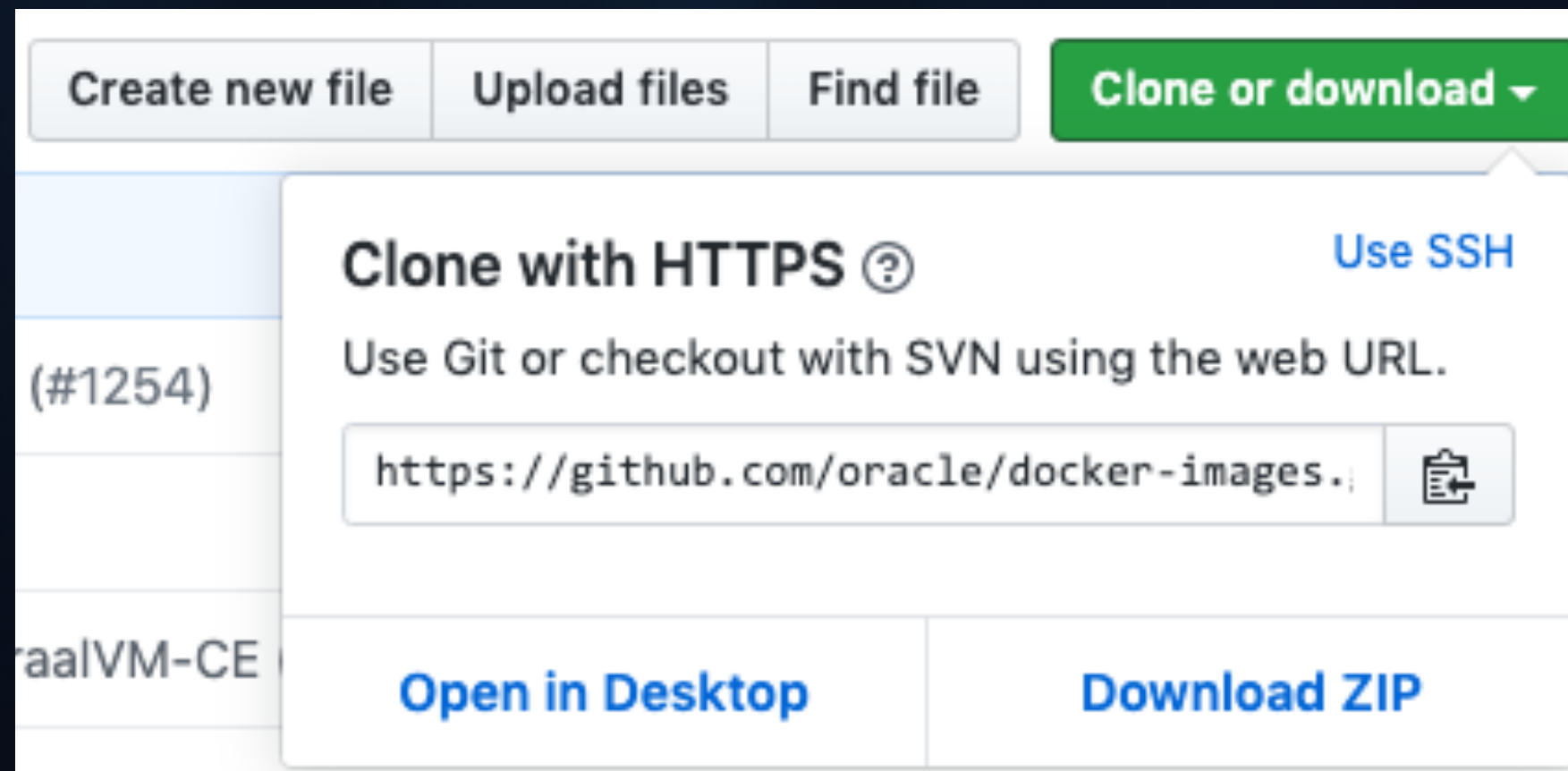
https://github.com/oracle/docker-images

The screenshot shows the GitHub repository page for 'oracle / docker-images'. At the top left, the repository name is displayed with a computer icon. To the right, there are buttons for 'Watch' (with a dropdown arrow), '395' stars, and 'Unstar'. Below this is a navigation bar with tabs for 'Code' (selected), 'Issues' (66), 'Pull requests' (8), 'Actions', 'Security' (0), and 'Insights'. The main content area contains the text: 'Official source for Docker configurations, images, and examples of Dockerfiles for Oracle products and projects' followed by a blue link: '<http://developer.oracle.com/containers>'.

Prepare the Repository

- Download and unzip -or-
- Git clone from GitHub:

```
git clone https://github.com/oracle/docker-images ~/docker-images
```



Copy Database Install Files to the Repository

```
cp ~/Downloads/LINUX.X64_193000_db_home.zip \  
~/docker-images/OracleDatabase/SingleInstance/dockerfiles/19.3.0
```

Do not unzip!

Build an Image

```
cd ~/docker-images/OracleDatabase/SingleInstance/dockerfiles  
./buildContainerImage.sh -v 19.3.0 -e
```

Wait...





Customizing Images

Customizing Images

`docker-images/OracleDatabase/SingleInstance/dockerfiles/<version>`

- Dockerfile
- `setupLinux.sh`
- `installDBBinaries.sh`
 - `db_inst.rsp`
- `startDB.sh`
- `runOracle.sh`
- `createDB.sh`
 - `dbca.rsp.tmp`



docker.
ORACLE

Pull an Image

<https://container-registry.oracle.com/>

Welcome to the Oracle Container Registry

Easy access to Oracle products for use in Docker containers

Browse Containers

Browse containers by product category such as database, java, middleware, and more!



Container Services



Container Services (Developer)



Database

FBI WARNING

Federal law provides severe civil and criminal penalties for the unauthorized reproduction, distribution and exhibition of copyrighted motion pictures and videotapes.

(Title 17, United States Code, Sections 501 & 506)

The Federal Bureau of Investigations investigates allegations of criminal copyright infringement.

(Title 17, United States Code, Section 506)

```
docker pull totallylegit/oracle-database:22q
```



docker.
ORACLE

Build as a Stack in OCI

Create an Oracle Cloud Free Tier Account

<https://signup.cloud.oracle.com>

Oracle Cloud Free Tier

Get started with...

Always-free access to essential services:

- Autonomous Database
- Virtual machines
- Object storage

Plus, \$300 of credits for 30 days to use on even more services:

- Container Engine for Kubernetes
- Analytics Cloud
- Data Integration

Account information

Country/Territory

First Name Last Name

Email

I am human

hCaptcha
Privacy - Terms

Verify my email

Access the Lab

Login to your OCI account

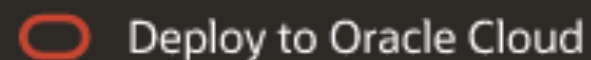
Navigate to <https://github.com/oraclesean/docker-lab>

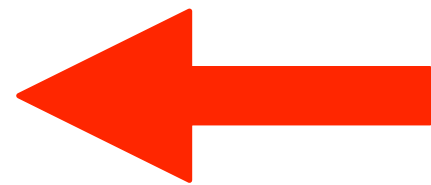
Click on the "Deploy to Oracle Cloud" button

Deploy on OCI

To deploy this lab on Oracle Cloud Infrastructure, [create a new Always Free Tier account](#), log in to an existing account still within the trial period or an account with enough credits to create a VM.Standard2.1 instance or higher.

Then, click on the "Deploy to Oracle Cloud" button:

 Deploy to Oracle Cloud



Prepare Your Lab Environment

Accept the Terms of Use

Welcome!

You're here because you clicked a button to deploy cloud resources, using the [package](#) identified below.

Package URL: <https://github.com/oraclesean/docker-lab/releases/download/v1.0/master.zip>

I have reviewed and accept the [Oracle Terms of Use](#).

Name *Optional*

Prepare Your Lab Environment

Confirm acceptance of Oracle's Software License Agreement

Configure the variables for the infrastructure resources that this stack will create when you run the apply job for this execution plan.

License Agreement

I have reviewed and accept the Oracle Licensing Agreement.

You must agree to and accept the Oracle Standard Terms and Restrictions listed on the Oracle Container Registry (<https://container-registry.oracle.com>) prior to running this lab. By checking this box, you affirm that you have accepted the Oracle Standard Terms and Restrictions

Prepare Your Lab Environment

Generate an SSH key:

```
ssh-keygen -t rsa -b 2048 -f /path/docker-lab-key
```

```
> ssh-keygen -t rsa -b 2048 -f ~/docker-lab-key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/sean.scott/docker-lab-key.
Your public key has been saved in /Users/sean.scott/docker-lab-key.pub.
The key fingerprint is:
```

Prepare Your Lab Environment

Paste or copy the public key to OCI

```
> cat docker-lab-key.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACxrHEIDSjAEU9UWg3tWhtasydKFJ2Wrhz5y94xsEG201gIM4mFZAJqhXNLrPByZ9Vs3//8AyaIsC  
hAHEfa/60fbGQ5Gn7vMpBDISXIZBXUyUh9PVQNTbKSYGXx/UT67ejQeyfw9VQy1Rt6+rdeiQ345j0nuxDgp sean.scott@SCOTT-C02QP2DJG8WN
```

Lab Name *Optional*

SSH Public Key *Optional*

Choose SSH Key File Paste SSH Key

The public key to install on the compute for SSH access.

Customize Your Lab Environment

Select your preferred VM image
Check “Show advanced options”

Instance Shape *Optional*

VM.Standard2.2

Select a shape for the compute instance

Show advanced options?
Shows advanced options for configuring Docker installation, database image builds, and container creation.

Customize Your Lab Environment

Select “Lab” as the source

Choose the database version

Check the “Run a container” option

Advanced Settings

Docker Image Source *Optional*

Lab

Select how the Docker database image will be created: pull from Lab resources, pull from Oracle Container Registry, or build from the Oracle Docker Images repository

Database Version *Optional*

19.3.0

The database version to install

Run a container?

True causes the provisioning process to create a database container using 'docker run'. Set to false to work through the lab steps. (Adds ~20 minutes to the provisioning time)

Customize Your Lab Environment

Use or change the defaults as you prefer!

Container Name *Optional*

Name for the Docker container

Database Name *Optional*

Database SID to create

Pluggable Database (PDB) Name *Optional*

Pluggable database name to create

Host Port for TNS *Optional*

Host port for mapping TNS connections from the container port 1521

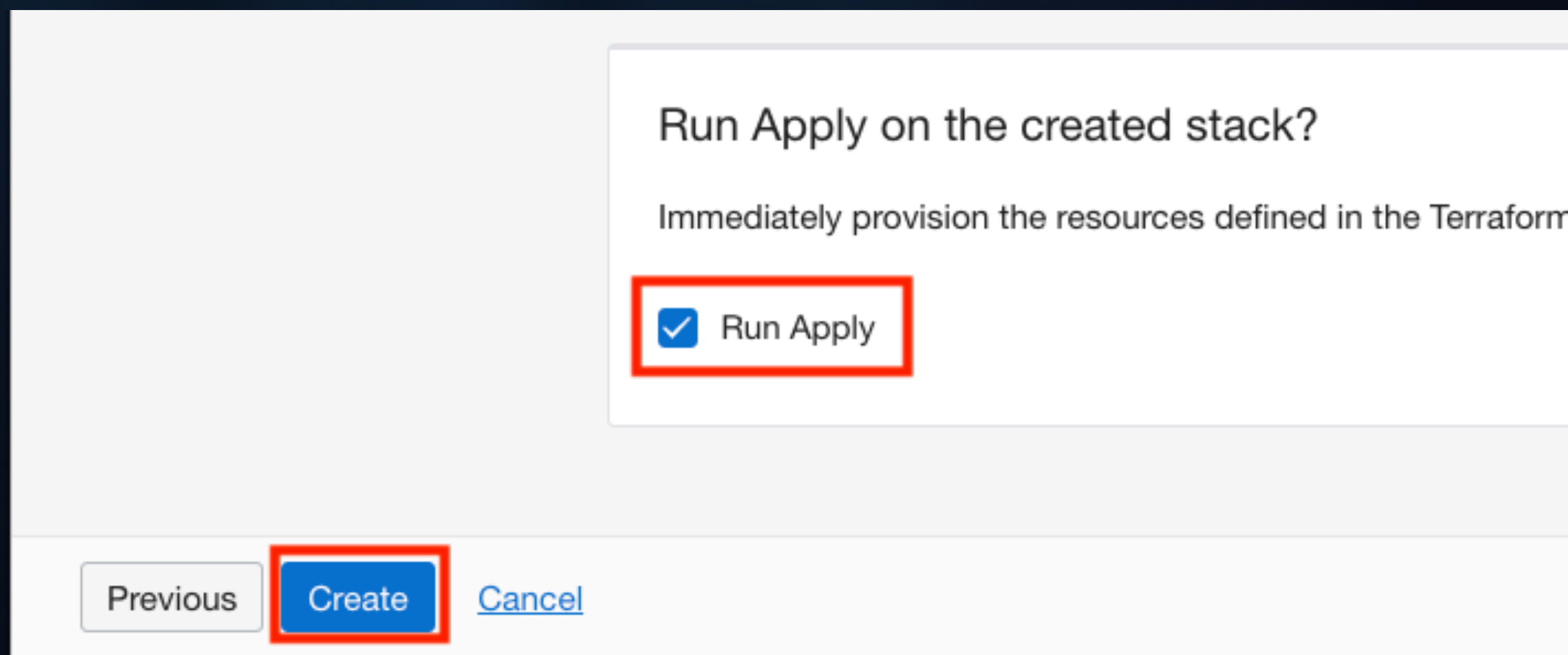
Run the Stack

Click "Next"



Make sure "Run Apply" is checked


Click "Create"




Run the Stack

Job is running:

Resource Manager » Stacks » Stack Details » Job Details



ACCEPTED

 While this job is running, only partial logs are available. You can get a complete log when the job is finished.

ormjob20210920003207

[Edit Job](#) [Download Terraform Configuration](#) [Cancel Job](#) [Add Tags](#)

Job Information

Tags

OCID: ...tdhtwq [Show](#) [Copy](#)

Job Type: Apply

State: ● Accepted

Start Time: Mon, Sep 20, 2021, 00:32:07 UTC

Compartment: [REDACTED]

Plan Job ID: Automatically approved

Working Directory: terraform

End Time: N/A

Run the Stack

Job is finished:

Resource Manager » Stacks » Stack Details » Job Details

ormjob20210920003207

[Edit Job](#) [Download Terraform Configuration](#) [Download Terraform State](#) [Add Tags](#)

RMJ

SUCCEEDED

Job Information **Tags** **Application Information**

OCID: ...tdhfwq [Show](#) [Copy](#) **Compartment:** [REDACTED]

Job Type: Apply **Plan Job ID:** Automatically approved

State: ● Succeeded **Working Directory:** terraform

Start Time: Mon, Sep 20, 2021, 00:32:07 UTC **End Time:** Mon, Sep 20, 2021, 00:33:29 UTC

Resources Logs

Get the Lab IP Address

Under the Application Information tab:

The screenshot shows the Oracle Cloud Infrastructure console interface. At the top, there are four buttons: "Edit Job", "Download Terraform Configuration", "Download Terraform State", and "Add Tags". Below these are three tabs: "Job Information", "Tags", and "Application Information", with the latter being the active tab. The main content area displays the title "Run Oracle Databases on Docker in Oracle Cloud Infrastructure" and a subtitle "A lab environment for running an Oracle Database on Docker using Always Free Oracle Cloud Infrastructure". A blue information box contains the text: "This stack provisions a new compartment, adds necessary VCN and network assets, and creates and provisions a compute instance ready to run an Oracle database in a Docker container. It optionally builds a container and prepares a database running Oracle 19.3.0 or 21.3.0." Below this is a section titled "Lab Environment Information" which contains the text "Compute Public IP address: 129.146.XXX.XXX" followed by a "Copy" link. This text is highlighted with a red rectangular border.

Connect to the Host

```
ssh -l opc -i </path/your-key-file> <IP address>
```

```
ssh -i </path/your-key-file> opc@<IP address>
```

- opc = The Oracle Cloud user account (always the same)
- Key file = The private key from the key pair uploaded to OCI
- IP Address = The address from the Application Information tab

```
ssh -l opc -i /path/docker-lab-key 129.200.100.1
```

```
ssh -i /path/docker-lab-key opc@129.200.100.1
```

Is it ready yet?

Run:

```
docker images
```

```
docker ps
```

```
[opc@docker-lab-compute ~]$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
oracle/database	21.3.0-ee	7925b37d58fd	8 days ago	8.08GB

```
[opc@docker-lab-compute ~]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
ff5ed02b9137	oracle/database:21.3.0-ee	"/bin/sh -c 'exec \$0...'"	2 days ago	Up 2 days (healthy)

Included in the stack:

~opc/alpine

Explore how containers work

~opc/docker-images

Official Oracle Docker image repo
(github.com/oracle/docker-images)

~opc/docker-lab

The repo for this lab

Included in the stack:

oracle, dba

The Oracle user and group that owns database files

/oradata

A “local” directory for database files

git, docker

yum installs the repositories



docker.
ORACLE

The oradata Volume


```
|— ORCLCDB
|  |— ORCLPDB1
|  |  |— ...
|  |— control01.ctl
|  |— pdbseed
|  |  |— ...
|  |— redo01.log
|  |— redo02.log
|  |— redo03.log
|  |— sysaux01.dbf
|  |— system01.dbf
|  |— temp01.dbf
|  |— undotbs01.dbf
|  |— users01.dbf
```

```
└─ dbconfig
   └─ ORCLCDB
      └─ listener.ora
      └─ orapwORCLCDB
      └─ oratab
      └─ spfileORCLCDB.ora
      └─ sqlnet.ora
      └─ tnsnames.ora
```

```
├── ORCLCDB
│   ├── ORCLPDB1
│   │   ├── ...
│   │   ├── control01.ctl
│   │   ├── pdbseed
│   │   │   ├── ...
│   │   ├── redo01.log
│   │   ├── redo02.log
│   │   ├── redo03.log
│   │   ├── sysaux01.dbf
│   │   ├── system01.dbf
│   │   ├── temp01.dbf
│   │   ├── undotbs01.dbf
│   └── users01.dbf
```

```
└── dbconfig
    ├── ORCLCDB
    │   ├── listener.ora
    │   ├── orapwORCLCDB
    │   ├── oratab
    │   ├── spfileORCLCDB.ora
    │   ├── sqlnet.ora
    └── tnsnames.ora
```

Mapping the Data Volume

```
docker run -d \  
  --name oracledb \  
  -p 11521:1521 \  
  -v /oradata/oracledb:/opt/oracle/oradata \  
  oracle/database:21.3.0-ee
```

Database files are in:

`/oradata/oracledb`

Mapping the Data Volume

My database **GOLD IMAGE** files are in:

`/oradata/oracledb`

Including:

- Data files
- Temp files
- Redo logs
- Archive logs
- TNS configurations
- Wallets
- `/etc/oratab`
- `pfile/spfile`

“Instant” Database!

Copy the gold image to a new directory:

```
sudo su -
```

```
cp -r /oradata/oracledb /oradata/clone
```

```
chown -R oracle:dba /oradata/clone
```

Only necessary
on Linux systems

“Instant” Database!

Be sure the hidden checkpoint file exists:

```
$ ls -la /oradata/clone
```

```
drwxr-xr-x.  4 oracle dba      61 Sep 23 14:50 .
dr-xr-xr-x. 19 root    root 4096 Sep 23 14:50 ..
drwxr-xr-x.  3 oracle dba     21 Sep 23 14:50 dbconfig
drwxr-x---.  4 oracle dba   4096 Sep 23 14:50 ORCLCDB
-rw-r--r--.  1 oracle dba     0 Sep 23 14:50 .ORCLCDB.created
```

“Instant” Database!

Change *name*, *port* and *directory path*:

```
docker run -d \  
  --name clone \  
  -p 12521:1521 \  
  -v /oradata/clone:/opt/oracle/oradata \  
  oracle/database:21.3.0-ee
```

```
docker logs -f clone
```



docker
ORACLE

Use Unique Paths for /oradata

Use Unique Paths!

Don't clone a database into the same directory:

```
docker run -d \  
  --name newname \  
  -v /oradata:/opt/oracle/oradata \  
  oracle/database:21.3.0-ee
```

Mixes, overwrites database files for multiple databases!



Alternative Repos

<https://github.com/oraclesean/docker-oracle>

- Easily create any available database version/edition
 - XE: 11.2.0.2, 18.4
 - SE: 11.2.0.4
 - SE2: 12.x, 18.3, 19.x, 21.x
 - EE: 11.2.0.4, 12.x, 18.3, 19.x, 21.x
- One management script for all versions
 - Version differences managed in version specific Dockerfiles
- Add patches without any code changes
- Leverages Buildkit

<https://github.com/oraclesean/docker-oracle>

- Customize images with runtime `--build-arg`, not script edits
 - Add/remove components
 - Add/customize binaries
 - Add supplemental RPM
 - Multi-stage minimizes image size
- Respects OFA
 - Set the `ORACLE_BASE`, `ORACLE_HOME`
 - Places `oraInventory` outside `ORACLE_BASE`

<https://github.com/oraclesean/docker-oracle>

- Install databases from ZIP or RPM without code changes
- Customize RPM install (including 18.3 XE) to:
 - Allow custom SID (not stuck with XE)
 - Create container, non-container databases
 - Define PDB names
 - Create 1 or many PDB

<https://github.com/oraclesean/docker-oracle>

- Create images and extend for:
 - Upgrades (multiple `ORACLE_HOMEs`)
 - Data Guard, Far Sync
 - GoldenGate
 - Shareplex, HVR
 - Sharding
 - OEM

<https://github.com/oraclesean/docker-oracle>

- More options for creating a database during docker run
 - Managed through environment variables—not code
 - Container, non-container
 - 1 or many PDB
 - Custom PDB naming using a common prefix or list of names
 - Control all character sets
 - Pass custom database parameters



Oracle & Docker Command Reference

Docker Command Reference: List Images, Containers

- List all Images
`docker images`
- List running Processes
`docker ps`
- List all Processes
`docker ps -a`

Docker Command Reference: Run a Container

- Run a new Container in the background
`docker run -id <image to use>`
- Run a new Container & start an interactive shell
`docker run -it <image to use>`

Docker Command Modifiers: docker run

- Name a Container

```
docker run... --name <container name>
```

- Map a host port to a container port

```
docker run... -p <port>:<container port>
```

- Map a host directory to a container filesystem

```
docker run... -v <directory>:<container directory>
```

Docker Command Reference: Start, Stop a Container

- Start a Container

```
docker start <container name>
```

- Stop a Container

```
docker stop <container name>
```

Docker Command Reference: Remove a Container

- Remove a Container
`docker rm <container name>`
- Remove a running Container
`docker rm -f <container name>`
- Remove multiple Containers
`docker rm <container1> <container2>`

Docker Command Reference: Remove an Image

- Remove an Image

```
docker rmi <image name>
```

- Force-remove an Image in use by a Container

```
docker rmi -f <image name>
```

- Remove multiple Images

```
docker rmi <image1> <image2>
```

Docker Command Reference: Show Logs

- Actively tail container logs
`docker logs -f <container name>`
- Show current logs
`docker logs <container name>`

Docker Command Reference: Connect to Containers

- Connect as the default user (Usually oracle for Oracle Database Images, root for Linux Images)
`docker exec -it <container> bash`
- Connect as root
`docker exec -it -u root <container> bash`
- Connect as root may fail if no working directory is defined; in that case, supply a working directory where root's session can start
`docker exec -it -u root -w <path> <container> bash`

Docker Command Reference: Execute Commands

- Run a simple command, return to host
`docker exec -it <container> <cmd>`
- Run multiple/complex commands, return to host
`docker exec -it <container> bash -c "<cmd1>; <cmd2>"`
- Run commands non-interactively (execute in the background)
`docker exec -id <container> <cmd>`

Docker Command Reference: Execute Commands

- Run SQL*Plus (using the default Oracle environment)
`docker exec -it <container> sqlplus`
- Change the SYS/SYSTEM passwords in Oracle
`docker exec -it <container> \
/opt/oracle/setPassword <newpass>`

Docker Command Reference: Commit an Image

- Commit a Container to a new Image. Run a Container from an Image, modify it, then save the Image to use for creating Containers that include changes
`docker commit <container> <name image name>`

Oracle Database Image Creation Reference

- Build an Oracle database Image

```
cd docker-images/OracleDatabase/SingleInstance/dockerfiles
```

Copy Database installation zip file to the appropriate version subdirectory, then:

```
./buildDockerImage <edition> -v <version>
```

Edition: -e = Enterprise

-s = Standard

-x = Express

Version: 11.2.0.2, 12.1.0.2, 12.2.0.1

18.3.0, 18.4.0, 19.3.0, 21.3.0

Oracle Database Container Creation Reference

- Create a simple database container (Image must exist)
`docker run -d <image>:<tag>`
Locate the image and tag by running `docker images`
- The Image and tag will look similar to:
`oracle/database:19.3.0-ee`
- In the following command references this is reduced to
just `<image>` for simplicity

Oracle Database Container Creation Reference

- Create a database container and name it
`docker run -d --name <container name> <image>`
- Create a database container and map ports
`docker run -d --name <container name> \
-p <host port>:<container port> \
<image>`
- For example:
`docker run -d --name o193 \
-p 11521:1521 oracle/database:19.3.0-ee`

Oracle Database Container Creation Reference

- Mount /opt/oracle/oradata to a host directory

```
docker run -d --name <container name> \  
-p <port>:<container port> \  
-v <directory>:/opt/oracle/oradata \  
<image>
```

Oracle Database Container Creation Reference

- Give the database a non-default name

```
docker run -d --name <container name> \  
-p <port>:<container port> \  
-v <directory>:/opt/oracle/oradata \  
-e ORACLE_SID=<dbname>  
<image>
```


Oracle Database Container Creation Reference

- Additional naming options for -e

Multiple parameters may be supplied to `docker run`

```
-e ORACLE_PWD=<password>
```

```
-e ORACLE_SID=<sid>
```

```
-e ORACLE_PDB=<pdbname>
```

```
-e ORACLE_CHARACTERSET=<characterset>
```

```
-tz=<timezone>
```



docker.
ORACLE

Questions



oraclesean.com



<https://www.linkedin.com/in/soscott/>



[@oraclesean](https://twitter.com/oraclesean)



<https://github.com/oraclesean>



sean.scott@viscosityna.com



Search "OracleSean" on YouTube



