



# Run Oracle Database in Containers

Rocky Mountain Oracle User Group  
Training Days 2023  
February 15, 2023



Sean Scott

**Oracle ACE Pro  
Managing Principal Consultant**

- @oraclesean
- linkedin.com/in/soscott/
- sean.scott@viscosityna.com



Sean Scott

**Oracle ACE Pro  
Managing Principal Consultant**

@oraclesean

[linkedin.com/in/soscott/](https://www.linkedin.com/in/soscott/)

[sean.scott@viscosityna.com](mailto:sean.scott@viscosityna.com)

**DATABASE RELIABILITY ENGINEERING ·· DEVOPS & AUTOMATION  
HIGH AVAILABILITY ·· BUSINESS CONTINUITY ·· DISASTER RECOVERY  
MODERNIZATION ·· OBSERVABILITY ·· ENGINEERED SYSTEMS**

AutoUpgrade ·· Zero Downtime Migrations ·· Patching

Real Application Clusters ·· Data Guard ·· Sharding

Docker/Containers ·· Terraform ·· Ansible

Exadata ·· Oracle Database Appliance

AHF ·· TFA ·· GIMR ·· CHA

# Oracle on Docker

Running Oracle Databases in Linux Containers

# Oracle on Docker

Running Oracle Databases in  
Linux Containers

—  
Sean Scott

apress®

Download a free sample chapter:  
<https://oraclesean.com>

# Viscosity's Oracle ACEs

## The Oracle ACE Program

The Oracle ACE Program recognizes and rewards individuals for their contributions to the Oracle community.



**Charles Kim**  
CEO | Co-Founder  
 @racdba  




**Rich Niemiec**  
Chief Innovation Officer  
 @richniemiec  




**Craig Shallahamer**  
Applied AI Scientist  
 @orapub  




**Sean Scott**  
Principal Consultant  
 @oraclesean  




**Gary Gordhamer**  
Principal Consultant  
 @ggordham  


# Viscosity Pillars and Delivery Models



## DATA

Oracle & SQL Server PostGres  
Performance Tuning  
Data Replication  
Data Warehousing Analytics  
Data Integration  
ERP Blue Prints  
Database Upgrades



## APPS

APEX  
EBS  
Web/Mobile Apps  
.Net and C#  
E-Business Suite  
SAAS/PAAS



## CLOUD

Azure Gold Partner  
Cloud Migrations  
Engineered Systems  
Oracle Cloud Partner  
Google Partner  
AWS Partner Hybrid Cloud

Workshops

Assessments

Proof of  
Concepts

Training

Turnkey  
Projects

Managed  
Services



viscosityna.com



@ViscosityNa



## Oracle License Management

Get the most out of your Oracle investment



### Apps

Oracle APEX  
SaaS/PaaS  
Custom Development



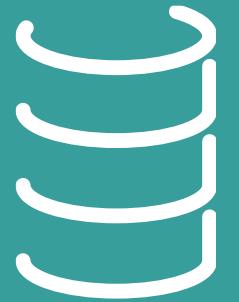
### Professional Services

Where you need it most



### Staff Aug

Workforce Capacity on Demand



### DBA Services

Remote and On-site



### ZERO DOWNTIME Migrations



### Performance Health Checks

How's it running?



### On-Call Support Managed Services



viscosityna.com



@ViscosityNa



Day 1 of RMOUG Training Days 2023: February 15, 2023  
Denver, CO



8am - 12pm MST  
Room 107

**Hands-on Workshop:**  
**Get Started With Oracle on Containers**  
Presented by: **Sean Scott**



1 - 5pm MST  
Room 107

**Hands-on Workshop:**  
**A Hands-on Introduction To Applied Machine Learning  
For The DBA**  
Presented by: **Craig Shallahamer**



viscosityna.com



@ViscosityNa



Day 2 of RMOUG Training Days 2023: February 16, 2023  
Denver, CO



8:30 - 9:30am MST  
Room 200 B + Virtual

[Machine Learning to Solve Business Issues](#) | Presented by: **Rich Niemiec**



8:30 - 9:30am MST  
Room 105 + Virtual

[Top 21 New Features of Oracle Database 21c and 19c](#) | Presented by: **Charles Kim**



11:20am - 12:20pm MST  
Room 109

[10 Things Every DBA Needs to do That Has Nothing to do With Databases](#)

Presented by: **Gary Gordhamer**



1. 1:35 - 2:35pm MST  
2. 2:40 - 3:40pm MST  
Room 200 B + Virtual

[Terraform on Oracle Cloud: A Primer for DBAs](#) | Presented by: **Sean Scott**



4:15 - 5:15pm MST  
Room 110

[AI/ML To The Rescue: How IT Operations Got Out of a Monitoring Mess](#)

Presented by: **Craig Shallahamer**



viscosityna.com



@ViscosityNa



Day 3 of RMOUG Training Days 2023: February 17, 2023  
Denver, CO



8:30 - 9:30am MST  
Room 110

**Practical Applications of Machine Learning |** Presented by: **Rich Niemiec**



1:35 - 2:35pm MST  
Room 112

**Top 5 Considerations for Migrating Databases to Oracle Cloud**

Presented by: **Charles Kim**



2:45 - 3:45pm MST  
Room 104 + Virtual

**Oracle SQL Profiles -- In Depth |** Presented by: **Gary Gordhamer**



2:45 - 3:45pm MST  
Room 107

**From Oracle Professional to Data Engineer in 8 Short Steps**

Presented by: **Jerry Ward**



# Workshop Preparations

# Create an Oracle Cloud Free Tier Account

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

The screenshot shows the Oracle Cloud Free Tier sign-up process. On the left, a sidebar lists "Always-free access to essential services" including Autonomous Database, Virtual machines, Object storage, and "Plus, \$300 of credits for 30 days to use on even more services" including Container Engine for Kubernetes, Analytics Cloud, and Data Integration. The main "Account information" section contains fields for Country/Territory, First Name, Last Name, and Email. Below these is a CAPTCHA challenge with a checkbox labeled "I am human" and a "Verify my email" button.

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

Verify my email



# Access the Lab

Login to your OCI account

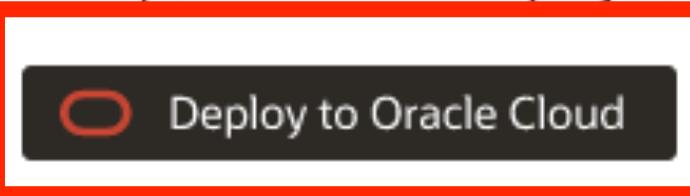
Navigate to <https://github.com/oraclesean/RMOUG23>

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:



# 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 AAAAB3NzaC1yc2EAAAQABAAQCrHEIDSjAEU9UWg3tWhtasydKFJ2Wrhz5y94xsEG201gIM4mFZAjQhXNLrPByZ9Vs3//8AyaIsChAHEfa/60fbGQ5Gn7vMpBDISXIzBXUyUh9PVQNTbKSYGXx/UT67ejQeyfw9VQylRt6+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

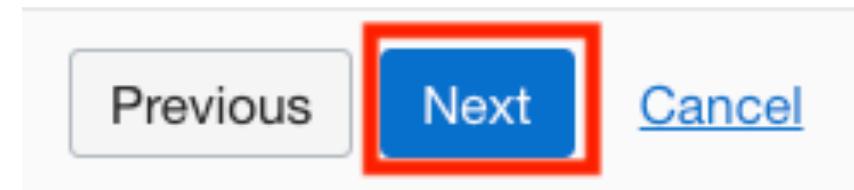


Show advanced options?

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

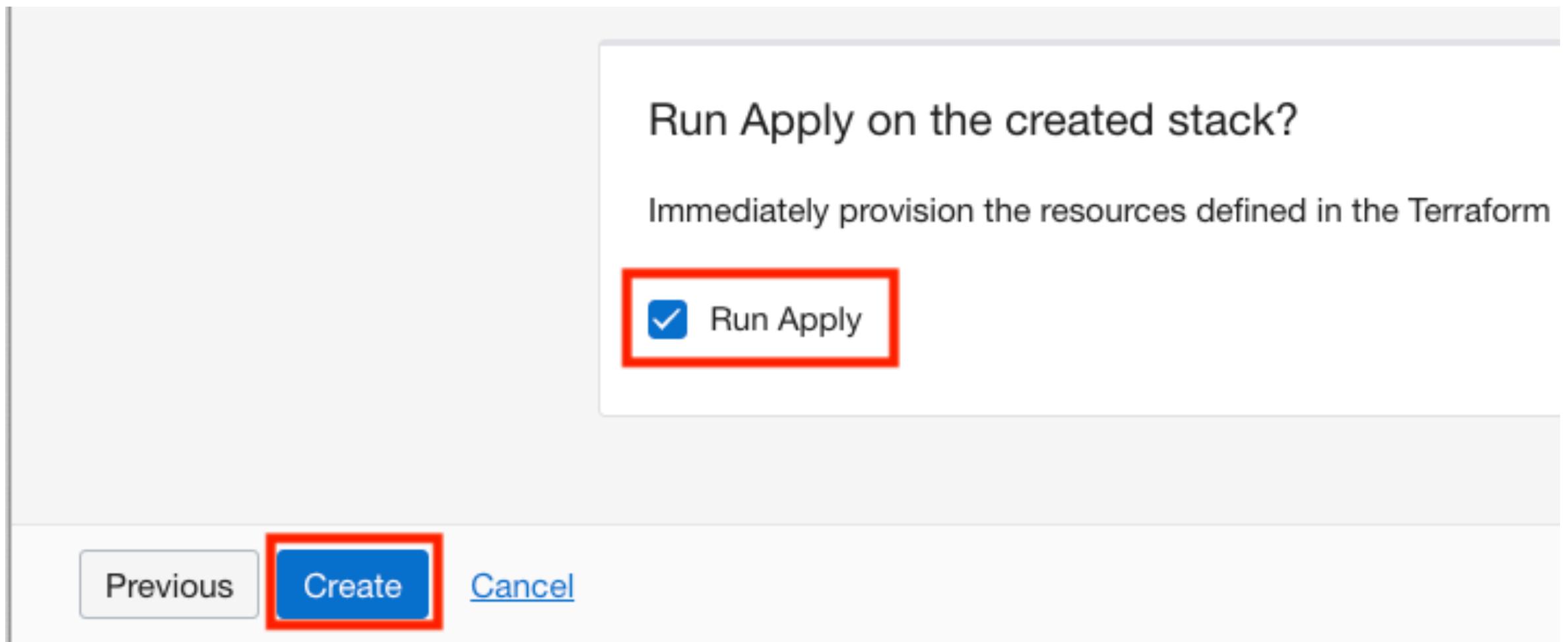
# Run the Stack

Click “Next”



Make sure “Run Apply” is checked

Click “Create”



# Run the Stack

Job is running:

Resource Manager > Stacks > Stack Details > Job Details

The screenshot shows the Oracle Cloud Resource Manager Job Details page. On the left, there's a large orange box with the letters 'RMJ'. Below it, the word 'ACCEPTED' is displayed in orange. The main content area has a blue info icon with a white 'i' followed by a message: 'While this job is running, only partial logs are available. You can get a complete log when the job is finished.' The job ID 'ormjob20210920003207' is prominently displayed. Below the job ID are four buttons: 'Edit Job', 'Download Terraform Configuration', 'Cancel Job' (which is red), and 'Add Tags'. Underneath these buttons are two tabs: 'Job Information' (selected) and 'Tags'. The 'Job Information' section contains the following details:

- OCID:** ...tdhtwq [Show](#) [Copy](#)
- Job Type:** Apply
- State:** ● Accepted
- Start Time:** Mon, Sep 20, 2021, 00:32:07 UTC

The 'Tags' section is currently empty. To the right of the job information, there are two columns of details:

- Compartment:** [empty]
- 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

The screenshot shows the Oracle Cloud Resource Manager Job Details page. On the left, there is a large green box with the letters "RMJ" in white. Below it, the word "SUCCEEDED" is displayed in green. The main title of the job is "rmjob20210920003207". Below the title are four buttons: "Edit Job", "Download Terraform Configuration", "Download Terraform State", and "Add Tags". Underneath these buttons, there are three tabs: "Job Information", "Tags", and "Application Information". The "Application Information" tab is highlighted with a red border. To the right of the tabs, there are several details listed: OCID: ...tdhfwq (with "Show" and "Copy" links), Compartment: (empty), Job Type: Apply, Plan Job ID: Automatically approved, State: ● Succeeded, Working Directory: terraform, Start Time: Mon, Sep 20, 2021, 00:32:07 UTC, and 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 a user interface for managing a cloud job. 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 'Application Information' being the active tab. The main content area has a 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 info 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.' At the bottom, there is a section titled 'Lab Environment Information' with a red box around the text 'Compute Public IP address: 129.146.XXX.XXX [Copy](#)'.

Edit Job Download Terraform Configuration Download Terraform State Add Tags

Job Information Tags Application Information

Run Oracle Databases on Docker in Oracle Cloud Infrastructure

A lab environment for running an Oracle Database on Docker using Always Free Oracle Cloud Infrastructure

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.

Compute Public IP address: 129.146.XXX.XXX [Copy](#)

# 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
```

# 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://github.com/oracle/docker-images>

Download the Oracle Docker images repository:

The screenshot shows the GitHub repository page for 'oracle / docker-images'. The page includes the repository name, a 'Watch' button (395), an 'Unstar' button, and navigation links for Code, Issues (66), Pull requests (8), Actions, Security (0), and Insights. A descriptive text block below the navigation links states: 'Official source for Docker configurations, images, and examples of Dockerfiles for Oracle products and projects'. Below this text are several blue-colored tags: docker, oracle, oracle-database, oracle-linux, dockerfile, and oracle-commercial.

oracle / docker-images

Watch 395 Unstar

Code Issues 66 Pull requests 8 Actions Security 0 Insights

Official source for Docker configurations, images, and examples of Dockerfiles for Oracle products and projects

<http://developer.oracle.com/containers>

docker oracle oracle-database oracle-linux dockerfile oracle-commercial

# Prepare the Repository

Create new file   Upload files   Find file   **Clone or download ▾**

. (#1254)

GraalVM-CE

**Clone with HTTPS** ⓘ   **Use SSH**

Use Git or checkout with SVN using the web URL.

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

**Open in Desktop**   **Download ZIP**



# Pull a Database Image

Pull the pre-prepared database image

Verify the image was downloaded successfully

```
docker pull phx.ocir.io/axh0z5q052rw/oracle:19.3.0
```

```
docker images
```

# Access the Lab Scripts

<https://github.com/oraclesean/oracle-container-lab>

# Optional (for building images)

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)

# Optional (for building images)

```
cp ~/Downloads/LINUX.X64_193000_db_home.zip \  
~/docker/docker-images/0oracleDatabase/SingleInstance/dockerfiles/19.3.0
```

Do not unzip!

```
cd ~/docker/docker-images/0oracleDatabase/SingleInstance/  
dockerfiles
```





Run Oracle in Containers?

# “Don’t use Docker for Databases!”

"Containers are for developers, not production!"

"Containers can't/shouldn't support  
complex applications like Oracle!"

"Containers are ephemeral and  
can't be trusted with important data!"



## 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](#).



Markus Michalewicz  
@KnownAsMarkus

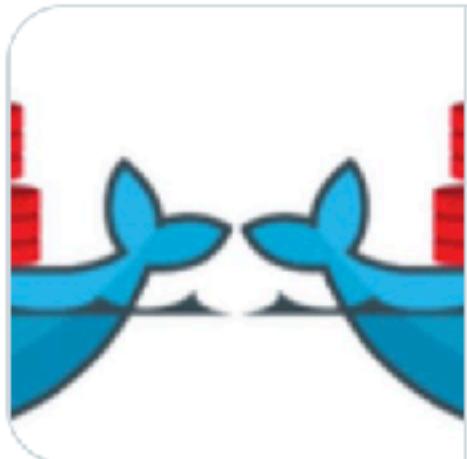
...

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-database-21c-comes-with-full-production-support-for-oracle-rac-on-docker)

/CC @OracleRACpm @oraclemaasig @oracleace



[blogs.oracle.com](https://blogs.oracle.com)

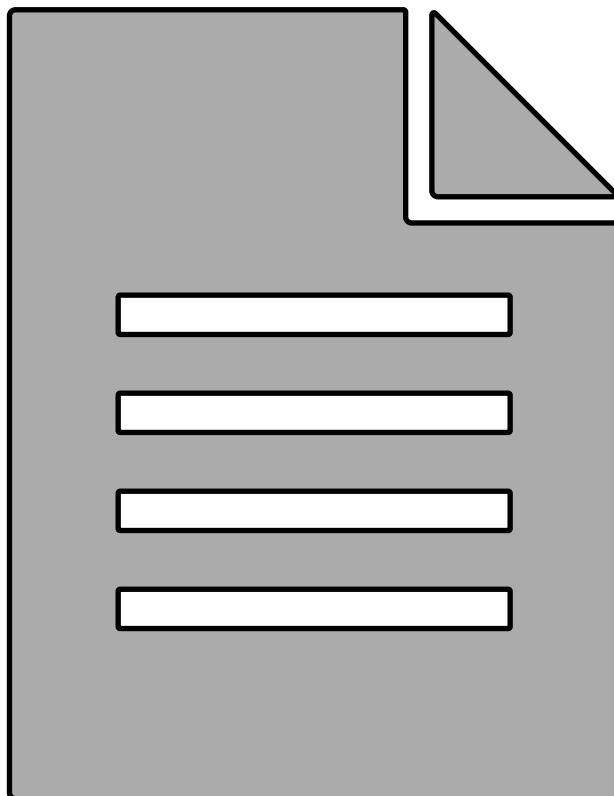
Oracle RAC on Docker - Now with Full Production Support  
Oracle Database 21c (21.3) comes with full production support for Oracle RAC on Docker!

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



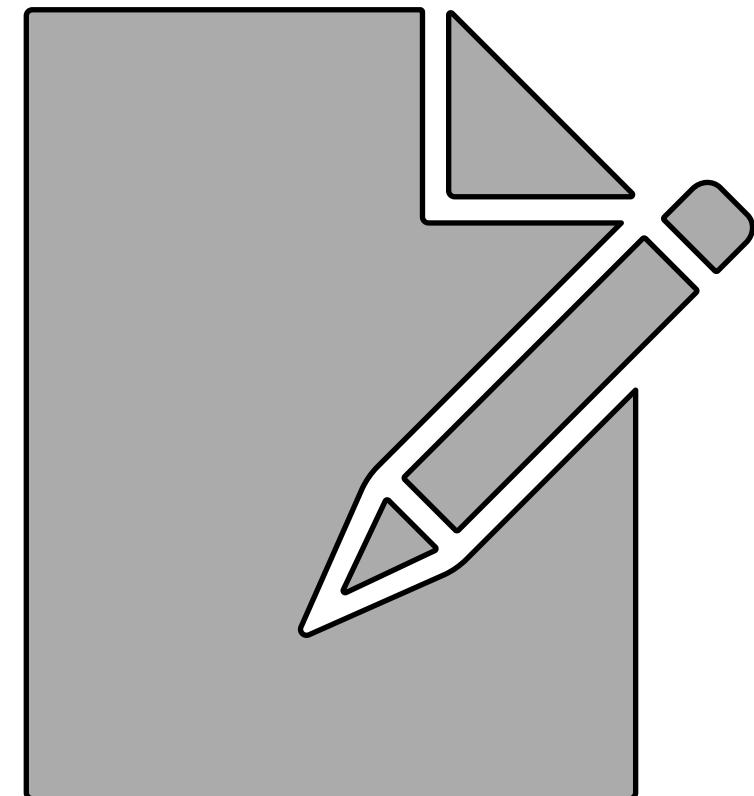
"Databases are Special™"

# "Databases are Special™"

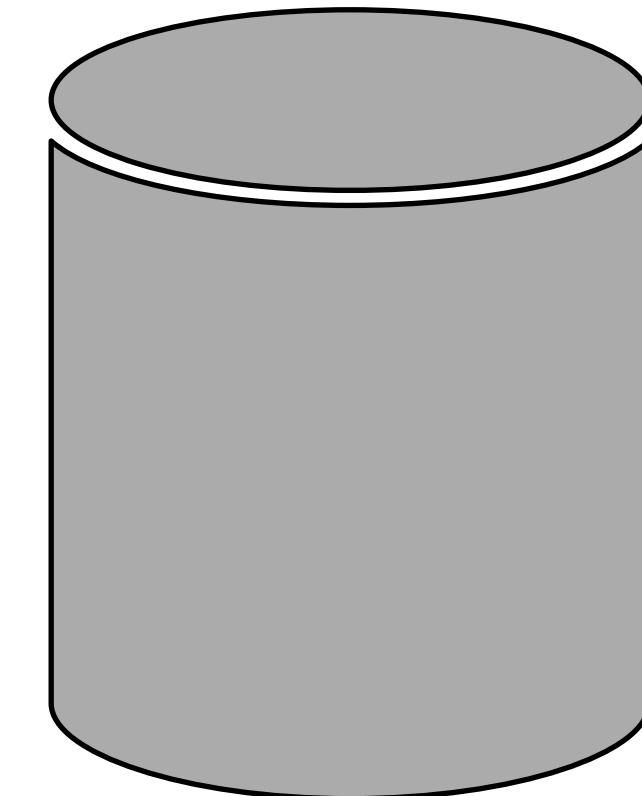


Software

Not a Database

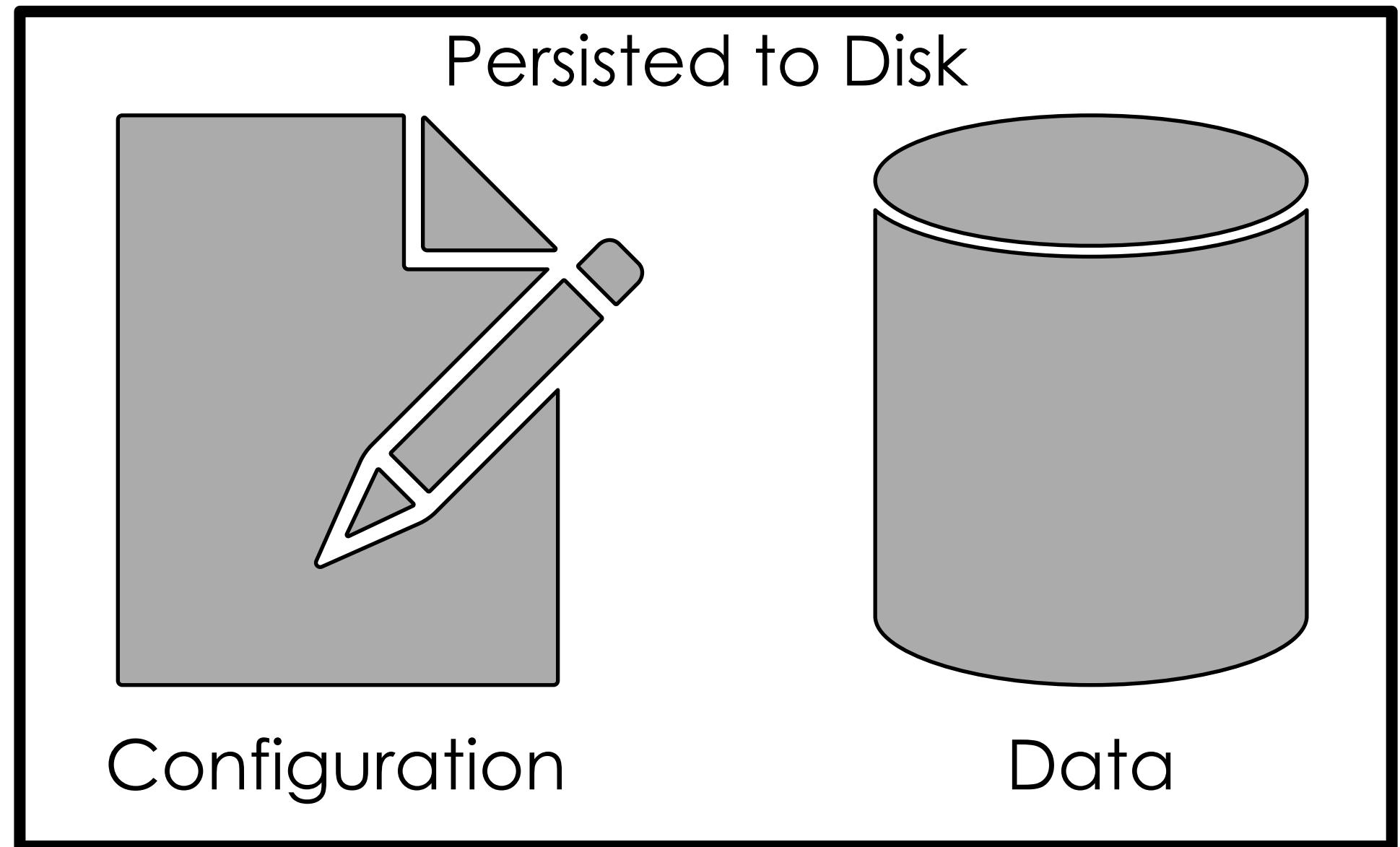
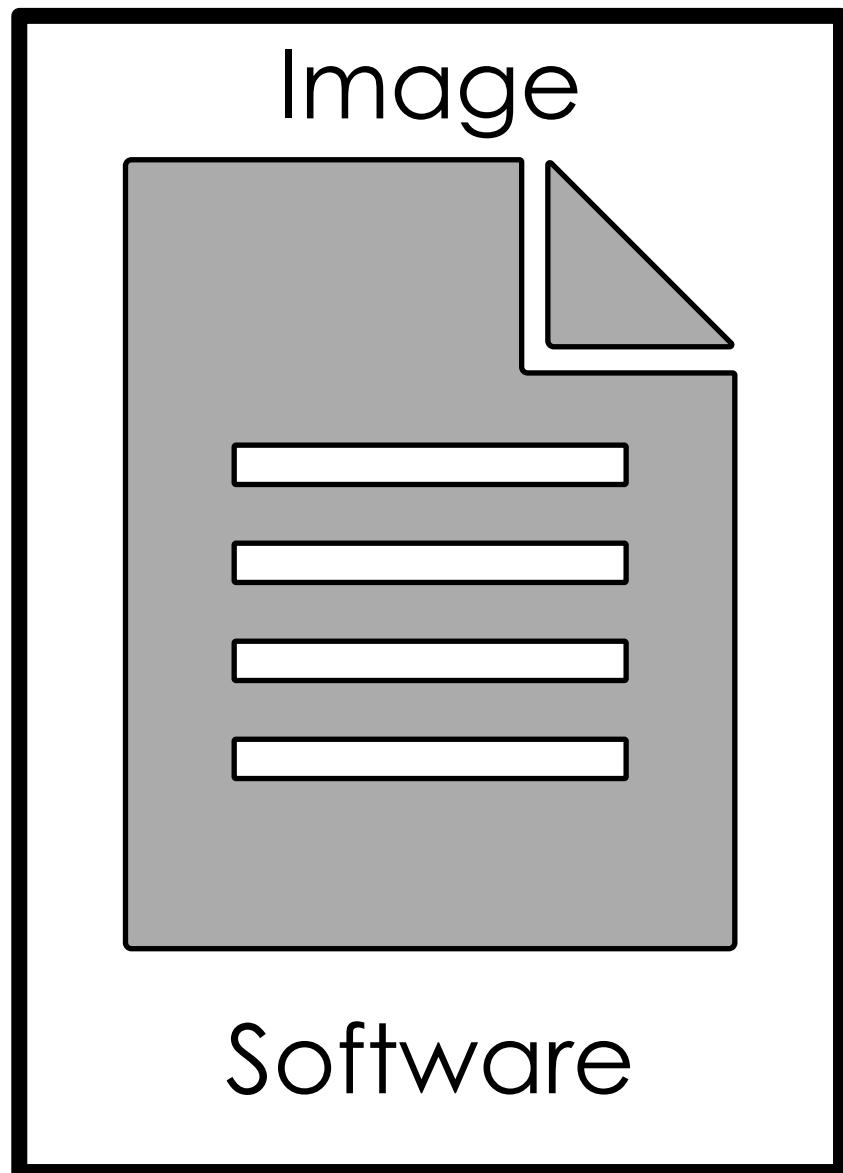


Configuration



Data

# "Databases are Special™"





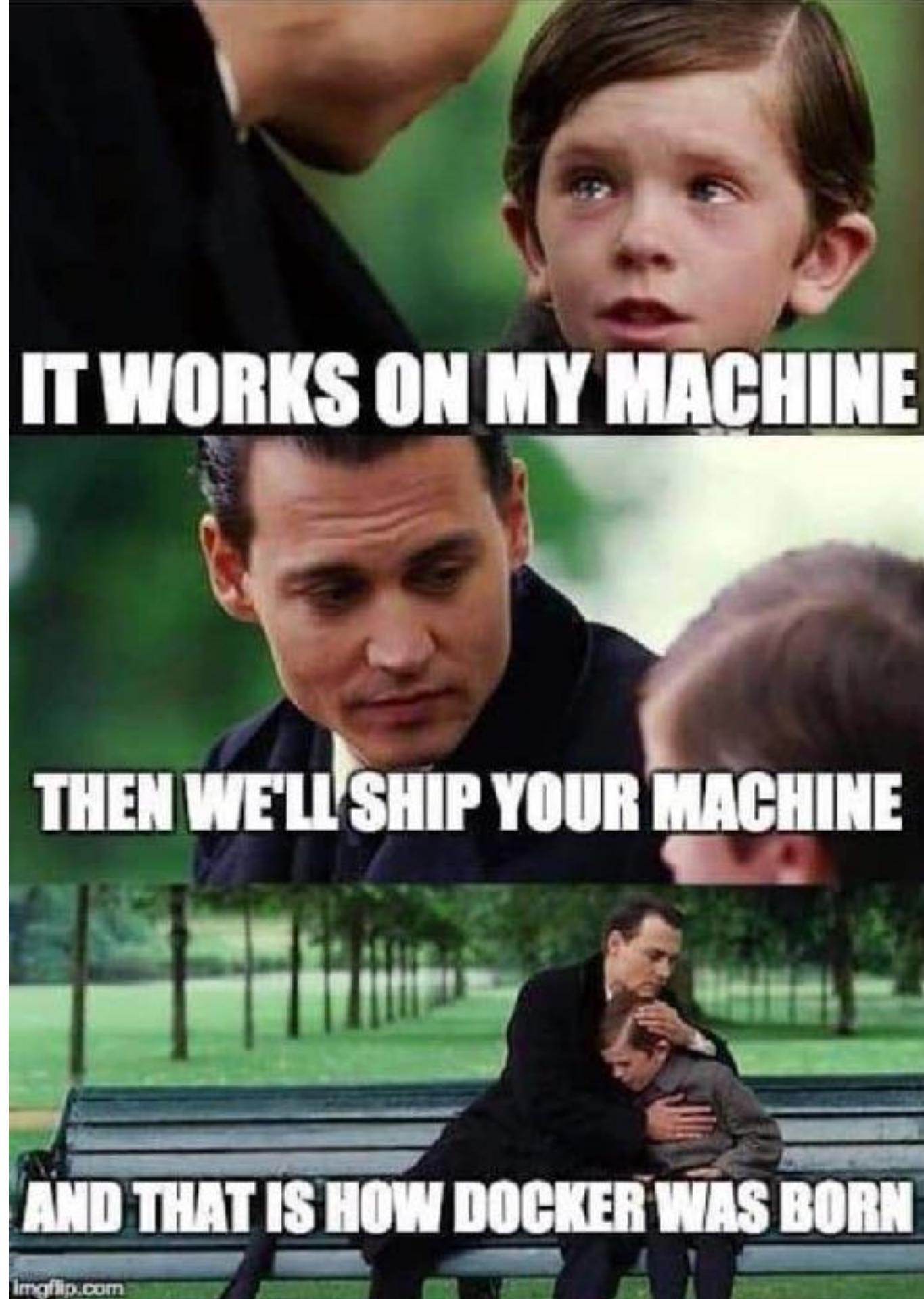
Why?

## Images are Infrastructure as Code

- Manage through Version Control
- Reliable and repeatable

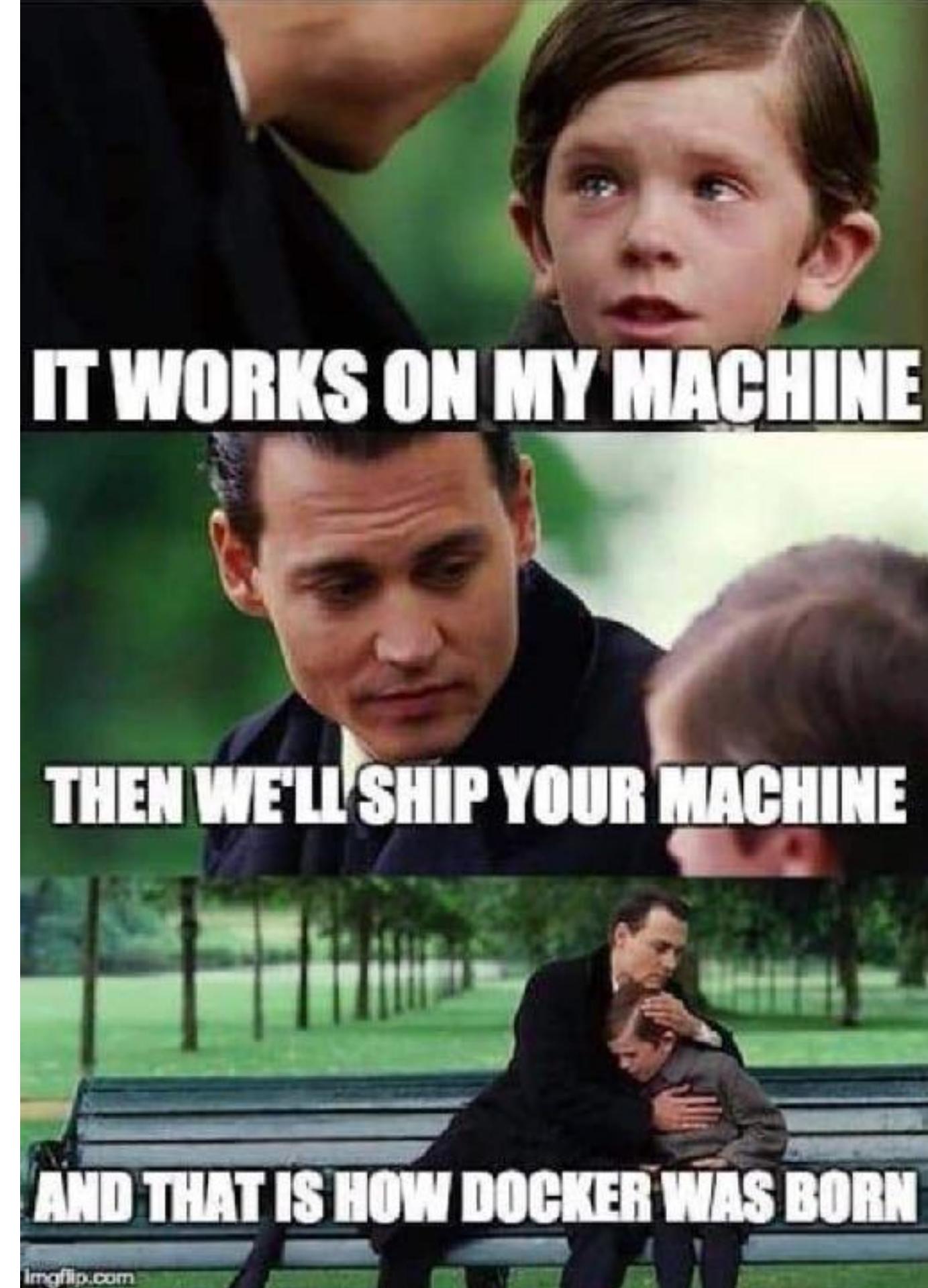
## Images are Artifacts

- Portable
- Distributable



Images built on <MY OS> work on:

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



# 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!



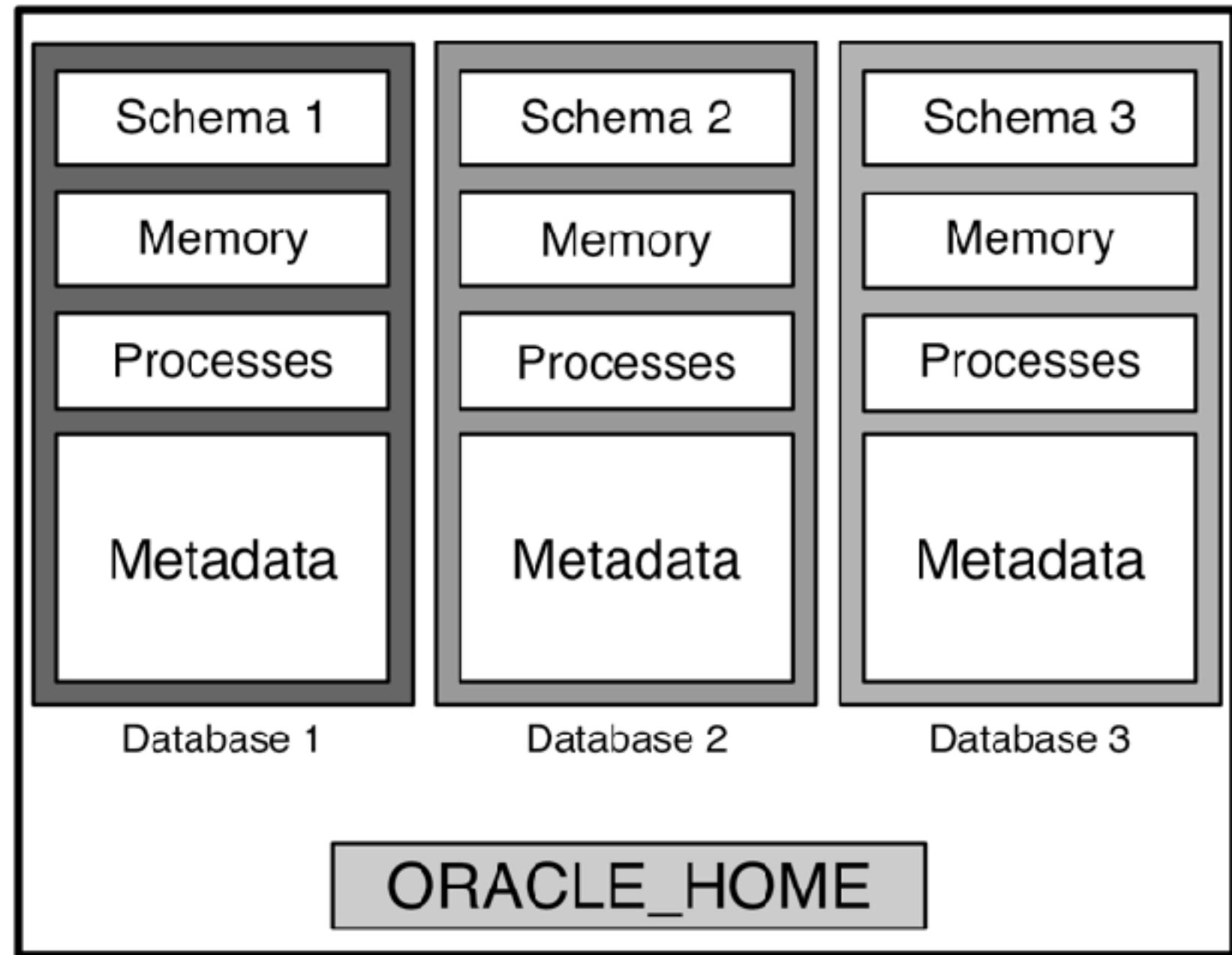
Containers are Fast

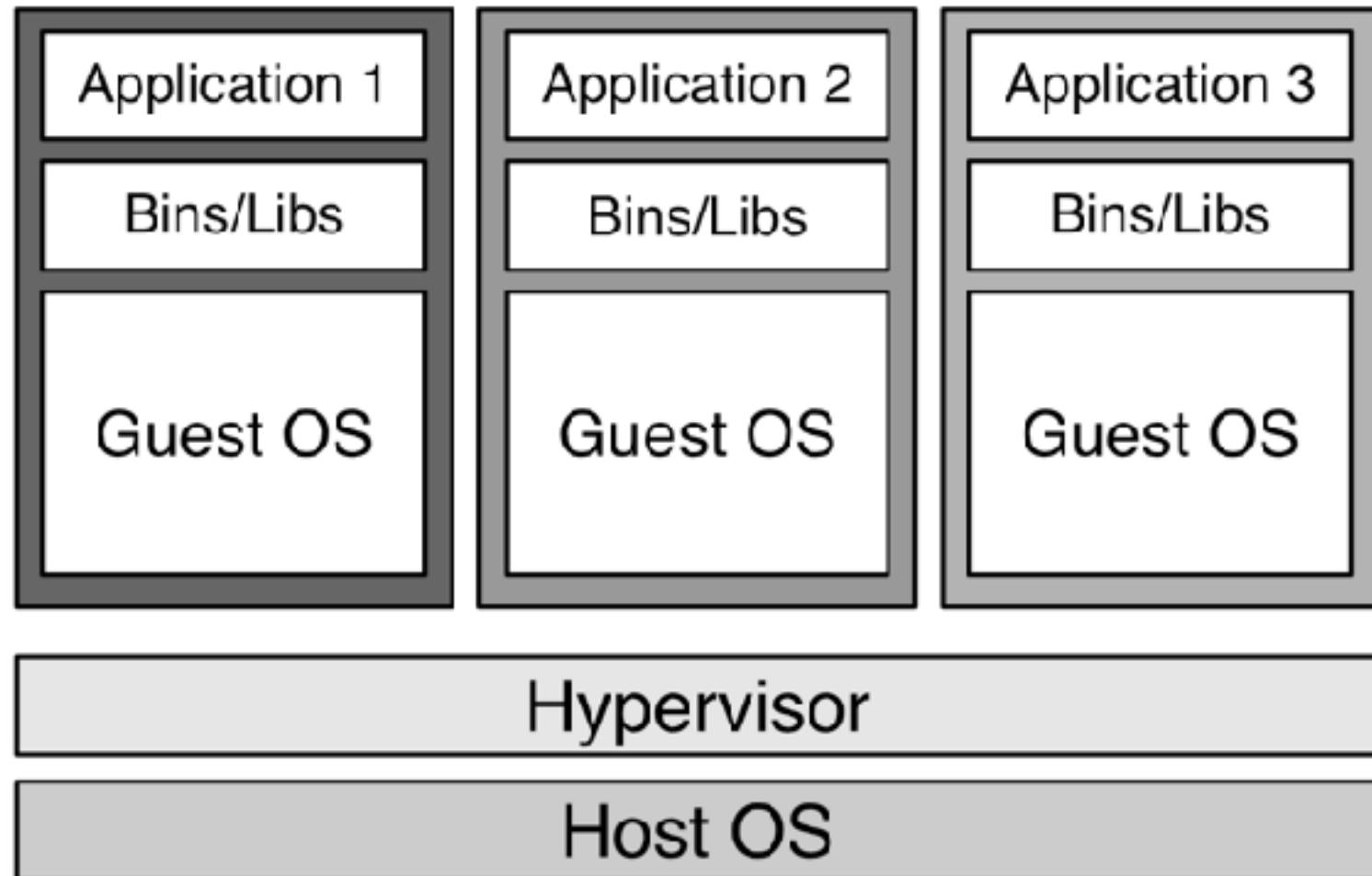


Containers are... uh...  
what are they?

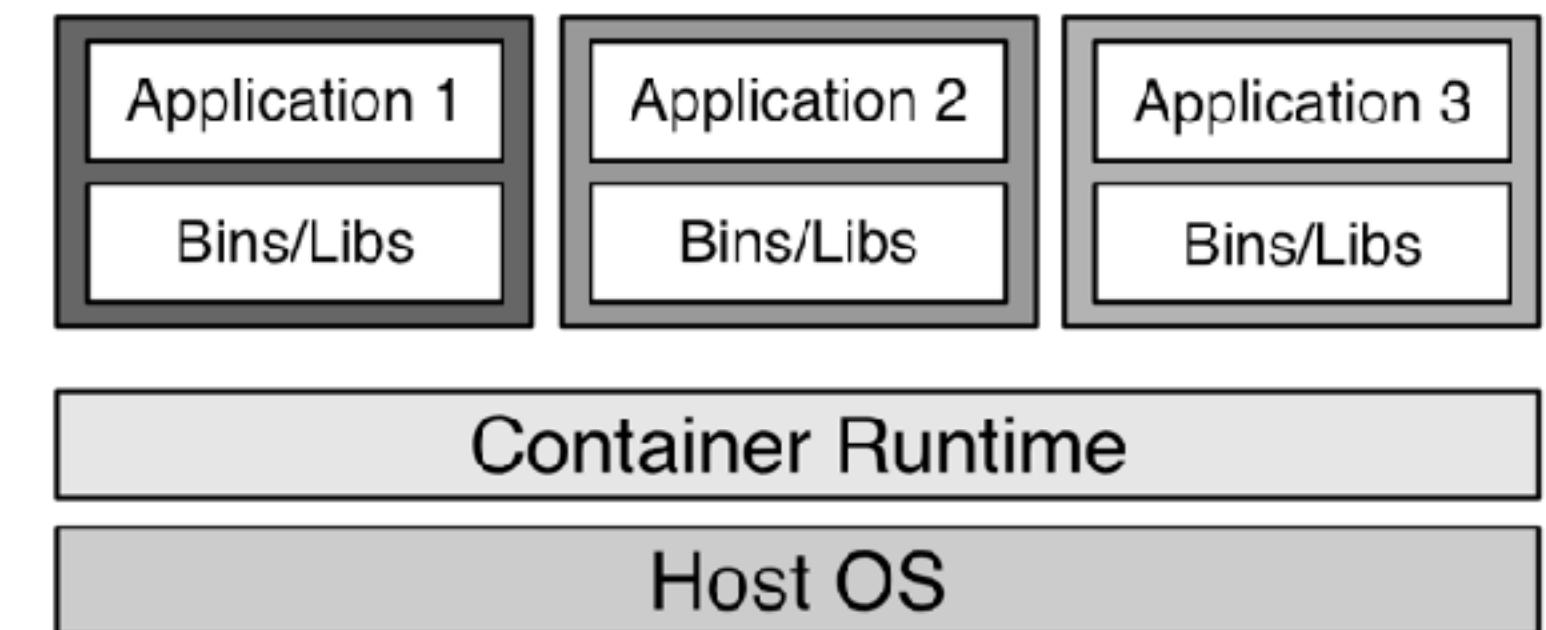
Containers are like lightweight Virtual Machines, right?



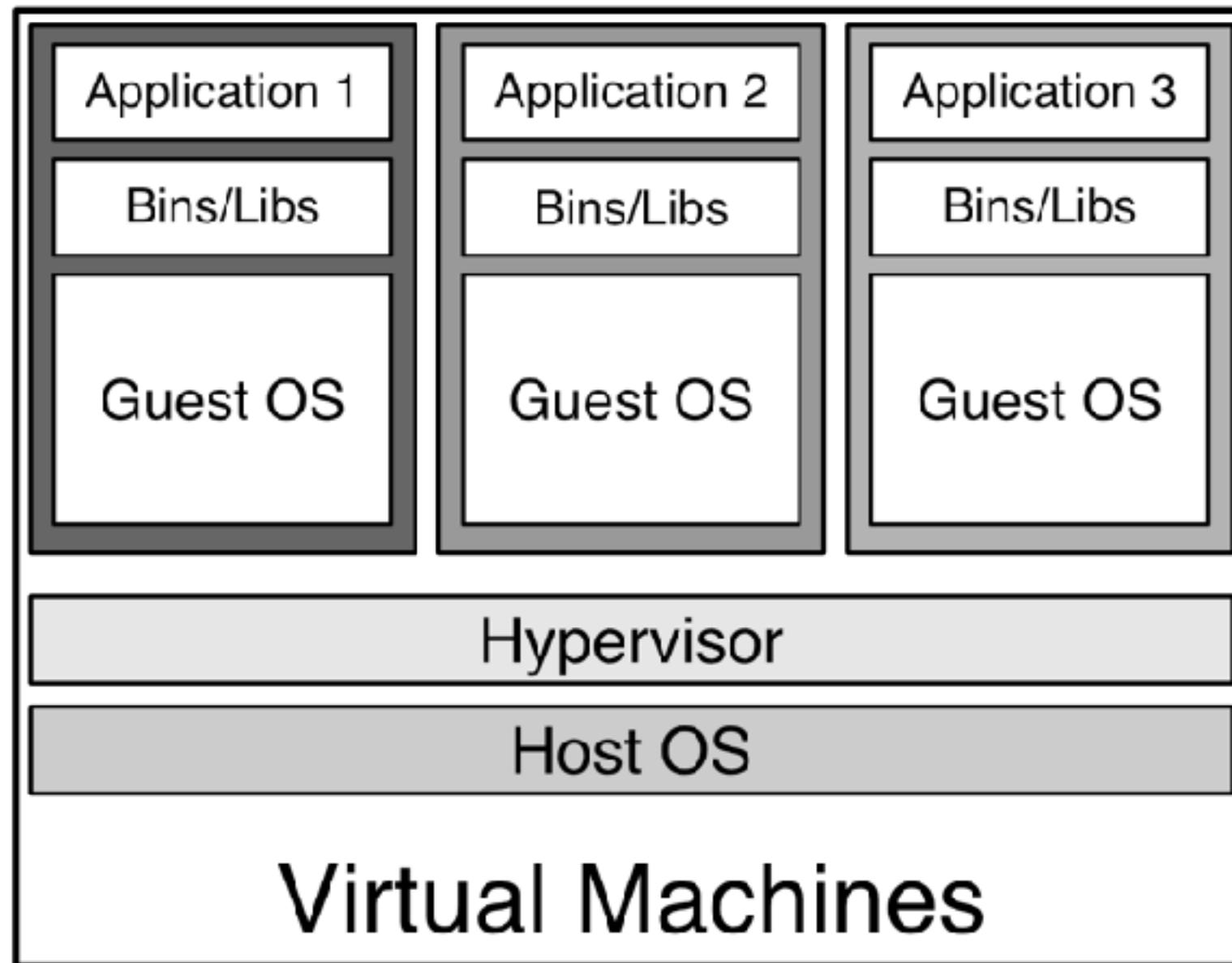




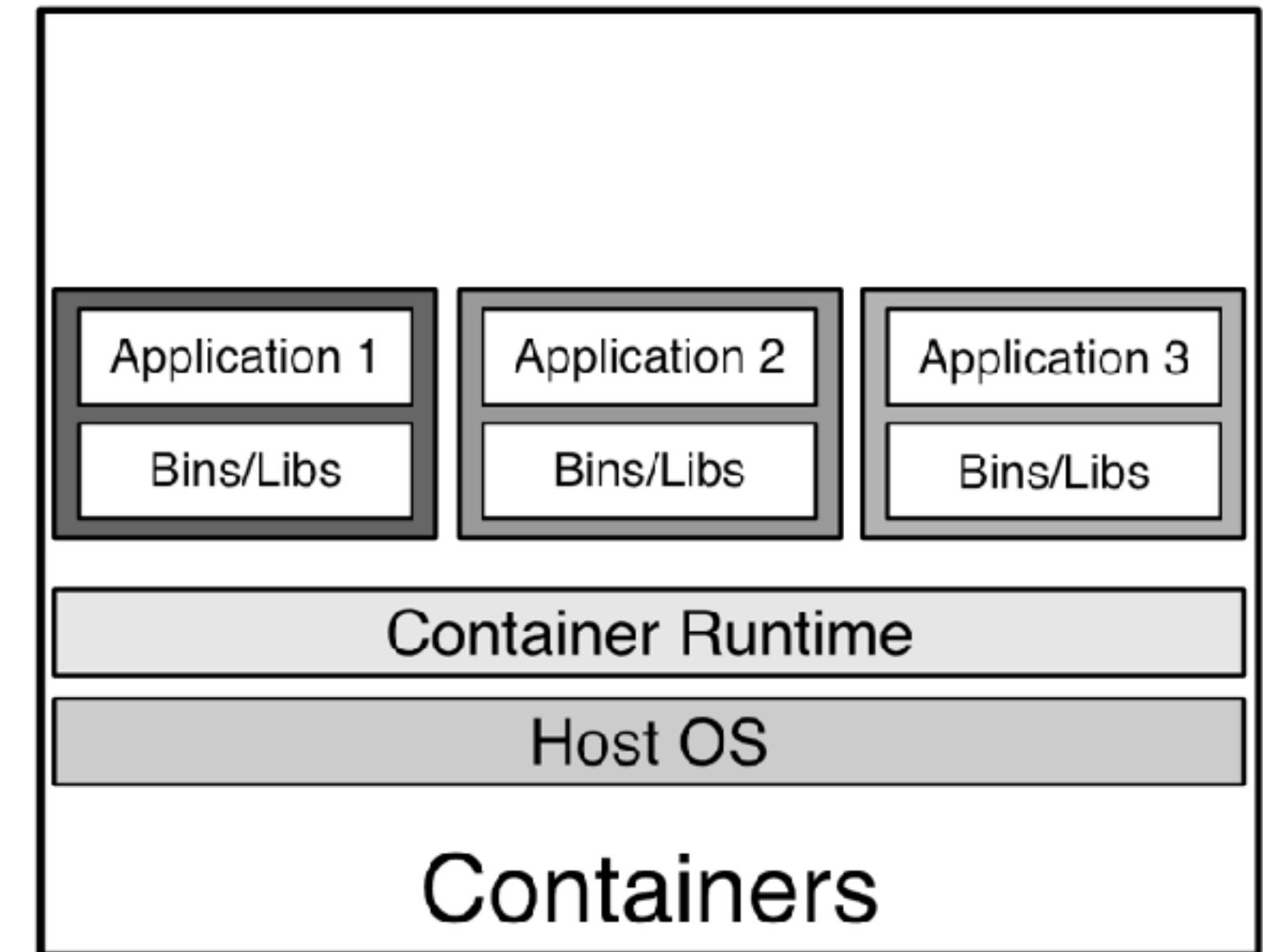
Three full operating systems



Three application/executable directories



Three full operating systems



Three application/executable directories

# 13 Oracle Databases on a Laptop

#	dps	NAMES	IMAGE	PORTS	NETWORKS	STATE
		ORCL112	oraclesean/db:11.2.0.4-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running
		ORCL121	oraclesean/db:12.1-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running
		ORCL122	oraclesean/db:12.2-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running
		ORCL1914	oraclesean/db:19.14-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running
		ORCL19151	oraclesean/db:19.15.1-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running
		ORCL19152	oraclesean/db:19.15.2-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running
		ORCL1915	oraclesean/db:19.15-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running
		ORCL19161	oraclesean/db:19.16.1-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running
		ORCL1916	oraclesean/db:19.16-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running
		ORCL1917	oraclesean/db:19.17-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running
		ORCL216	oraclesean/db:21.6-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running
		ORCL217	oraclesean/db:21.7-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running
		ORCL218	oraclesean/db:21.8-EE	1521/tcp, 5500/tcp, 8080/tcp	oracle-db	running

2018 MacBook Pro 15", 2.2GHz 6-core Intel Core i7, 16GB, 1TB SSD

**Virtual Machines:** Bootable OS, often heavy

A 100-page book of games & puzzles

**Containers:** Support one application or service

A sheet of paper with a Tic-Tac-Toe grid



vo · cab · u · lar · y

**Image  
Layer  
Container  
Union Filesystem**

**Stateful/Stateless  
Immutable  
Ephemeral  
Persistent**

**Images** include a minimal *Filesystem*  
& Metadata needed by a Service.

The *Filesystems* in **Images** are  
constructed in **Layers**.

19c Preinstall RPM + 19.3 DB Binary

Oracle Linux 7

**Image Layers** are Modular and Reusable.

19.15.1

19.15.2

19.16.1

19.16.2

19.15 RU

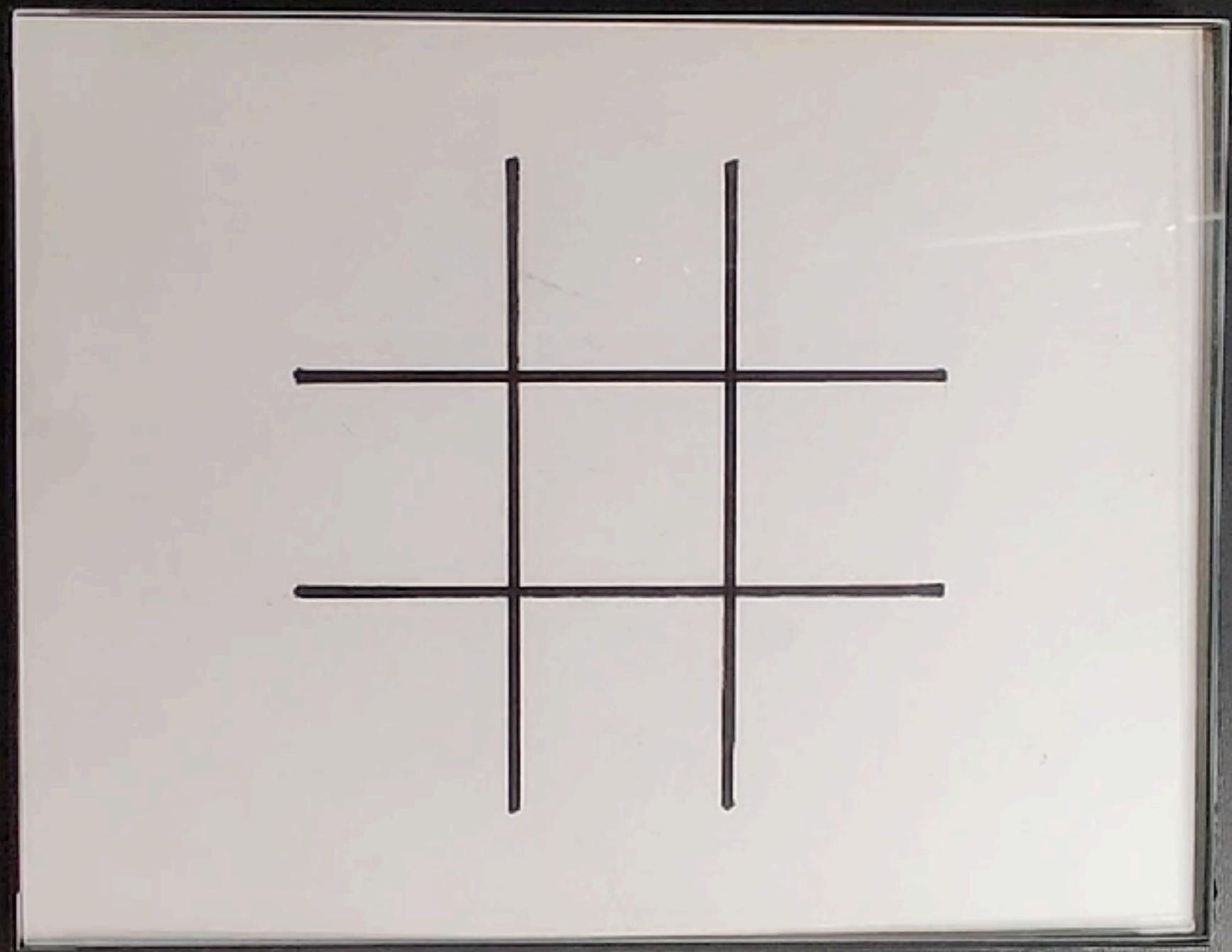
19.16 RU

19c Preinstall RPM + 19.3 DB Binary

Oracle Linux 7

*Running an **Image** starts a **Container**  
as an *Isolated Process* on its host.*

**Physically, Containers are a Layer  
superimposed onto an Image.**



An **Image** is the game board and rules;

**Containers** host and persist games;

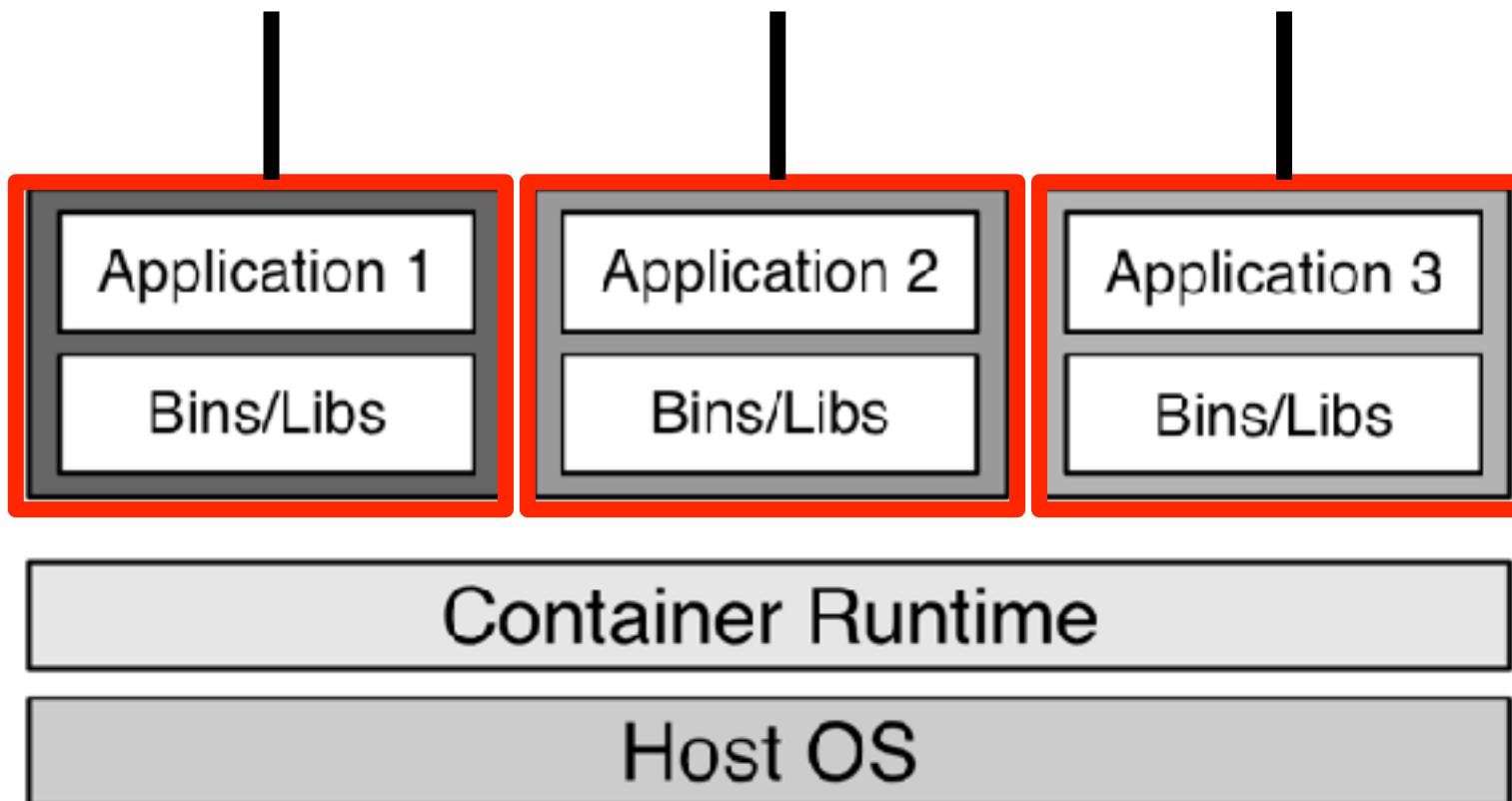
The superimposed **Layer** saves moves;

Deleting a **Container** removes its **Layer**.

# **Multiple Containers share one Image.**

# How much space do three containers use?

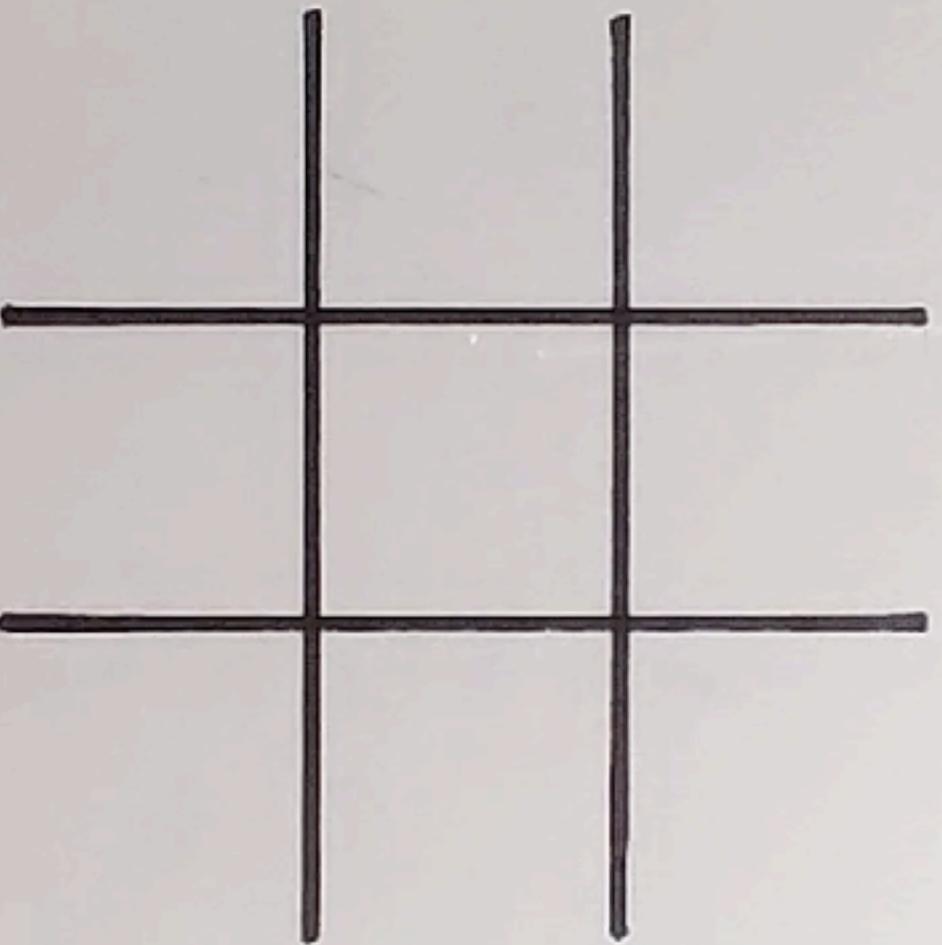
500MB 500MB 500MB



$$\begin{array}{r} 500\text{MB} \\ + 500\text{MB} \\ + 500\text{MB} \\ \hline 1,500\text{MB} \end{array}$$

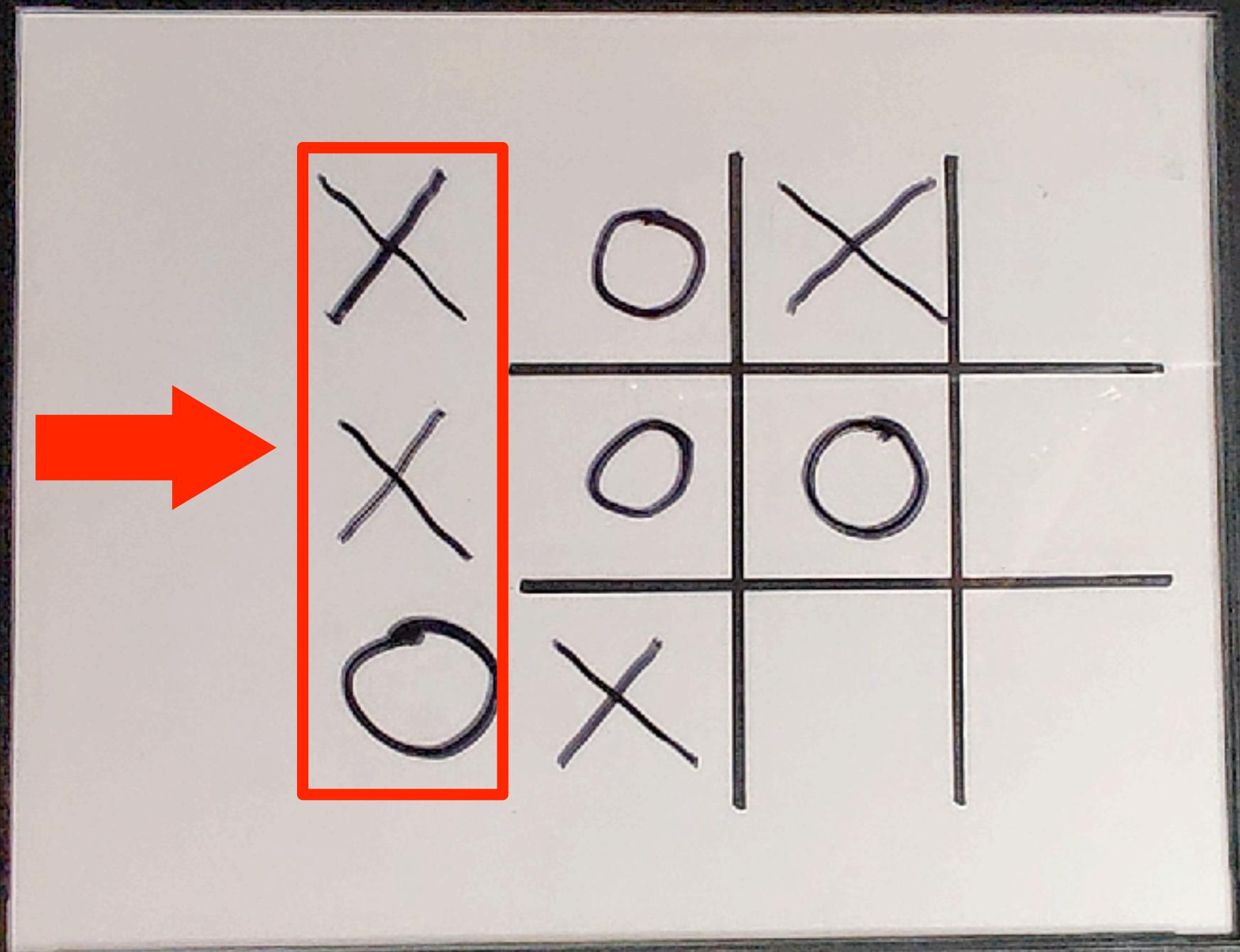
**Images** are **Immutable** and *never change.*

# Stalemate!



# Changing Shift in the `makeRight` the game!

Moves fall off the  
game board:



**Images** are **Immutable** and can't persist state  
information. They are **Stateless**.

**Containers are Stateful. They Persist  
state in their run-time Layer.**

Deleting a **Container** removes its  
run-time **Layer**. They are **Ephemeral**

**Images** = Static (Templates, Seed PDB)

**Containers** = Living (Documents, PDBs)



Run Oracle on Docker

TV LAND

IT'S ABOUT TIME



## Building Images

# 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

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>

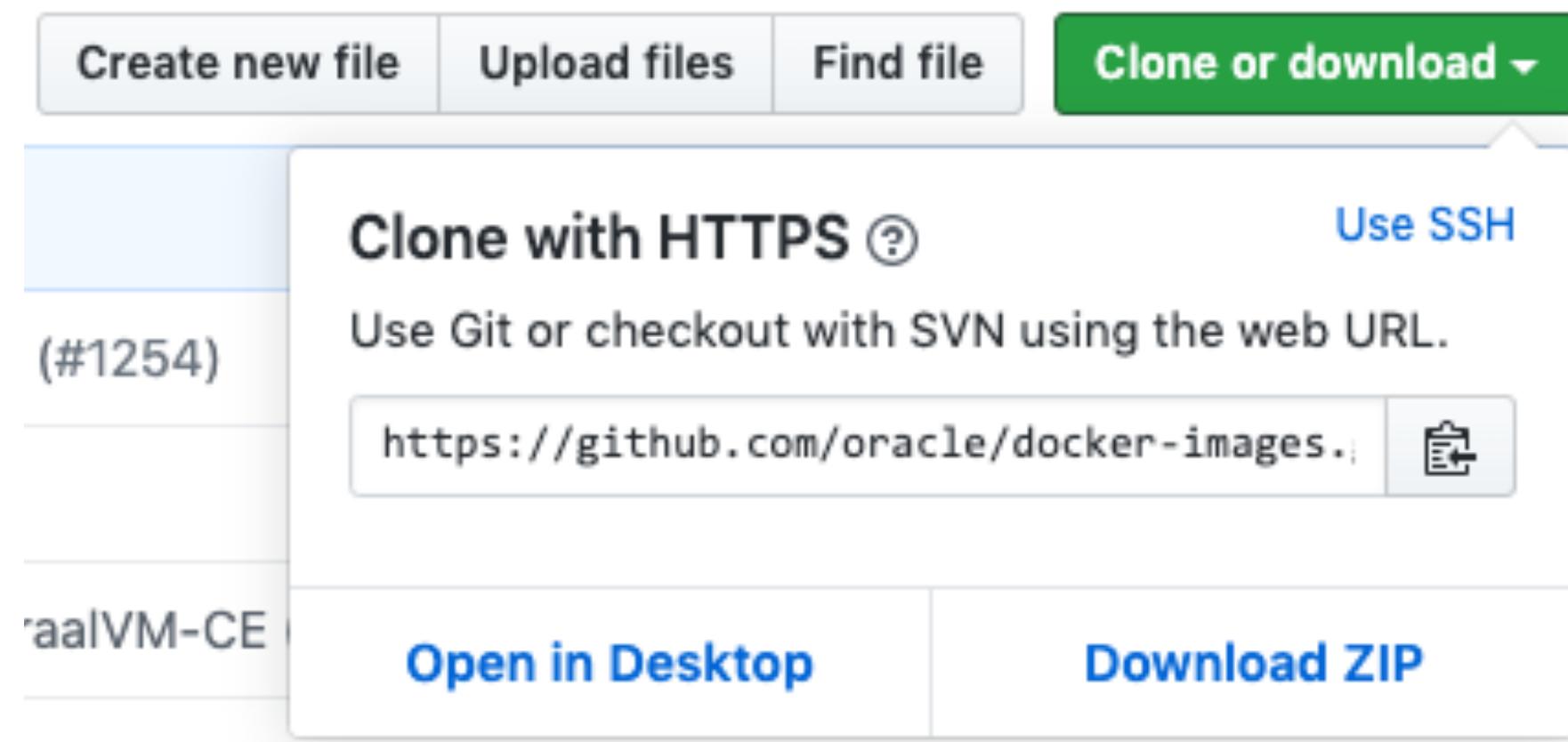
The screenshot shows the GitHub repository page for 'oracle / docker-images'. The repository has 395 stars. The navigation bar includes links for Code (selected), Issues (66), Pull requests (8), Actions, Security (0), and Insights.

Official source for Docker configurations, images, and examples of Dockerfiles for Oracle products and projects  
<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/0oracleDatabase/SingleInstance/dockerfiles  
./buildContainerImage.sh -v 19.3.0 -e
```





Pull an Image

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

Welcome to the Oracle Container Registry

Easy access to Oracle products for use in Docker containers

Search Oracle Container Registry

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 not-malware/oracle-database:24x



Run a Container

# Run a Database Container

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

# Run a Useful Database Container

```
docker run -d \
    --name o193 \
    -p 51521:1521 \
    -v ~/oradata/o193:/opt/oracle/oradata \
    -e ORACLE_SID=o193 \
    oracle/database:19.3.0-ee
```

# Customize Database Creation

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

# “Login” to the Container

```
docker exec -it o193 bash
```

# Connect to Docker from SQL Developer

Host  
IP Address  
Mapped Port  
ORACLE\_SID

The screenshot shows the 'User Info' tab of the SQL Developer connection dialog. The 'Name' field is set to 'Oracle Docker Lab'. The 'Database Type' is 'Oracle'. The 'User Info' tab is selected, showing 'Proxy User'.

**User Info** Proxy User

Authentication Type: Default

Username: system

Password: \*\*\*\*\*

Role: default

Save Password

Connection Type: Basic

**Details** Advanced

Hostname: localhost

Port: 51521

SID

Service name: o193



## 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
        ├── orapw0RCLCDB
        ├── oratab
        ├── spfile0RCLCDB.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
        ├── orapw0RCLCDB
        ├── oratab
        ├── spfile0RCLCDB.ora
        ├── sqlnet.ora
        └── tnsnames.ora
```

# Mapping the Data Volume

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

Database files are in:

`~/oradata/o193`

# Mapping the Data Volume

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

`~/oradata/0193`

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/o193 ~/oradata/clone
```

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

Only necessary  
on Linux systems



# “Instant” Database!

Change name, port, and directory path:

```
docker run -d \
--name clone \
-p 52521:1521 \
-v ~/oradata/clone:/opt/oracle/oradata \
oracle/database:19.3.0-ee
```

```
docker logs -f clone
```

# Use Unique Paths!

Don't clone a database into the same directory:

```
docker run -d \
    --name dynamo \
    -v ~/oradata:/opt/oracle/oradata \
    oracle/database:19.3.0-ee
```

Mixes and overwrites database files of multiple databases!



Customize Images

# Customizing Images

docker-images/0oracleDatabase/SingleInstance/dockerfiles/<version>

- Dockerfile
- setupLinux.sh
- installDBBinaries.sh
  - db\_inst.rsp
- startDB.sh
- runOracle.sh
- createDB.sh
  - dbca.rsp.tmp



oraclesean.com



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



@oraclesean



<https://github.com/oraclesean>



sean.scott@viscosityna.com



[https://youtube.com/@oracle\\_sean](https://youtube.com/@oracle_sean)

