ROBIN
SYSTEMS

Containerizing Oracle: Not Thinking About It Yet ? You Should Be !!!

ABOUT ME

- Over 19 years of experience across Databases, & big data applications
- Director of Products, Robin Systems
  - Virtualizing Big data and databases with bare-metal performance
- Previously
  - Product manager– Oracle Multitenant, Database Diagnostics and Tuning packs
  - Principal Systems Consultant - Performance Services team

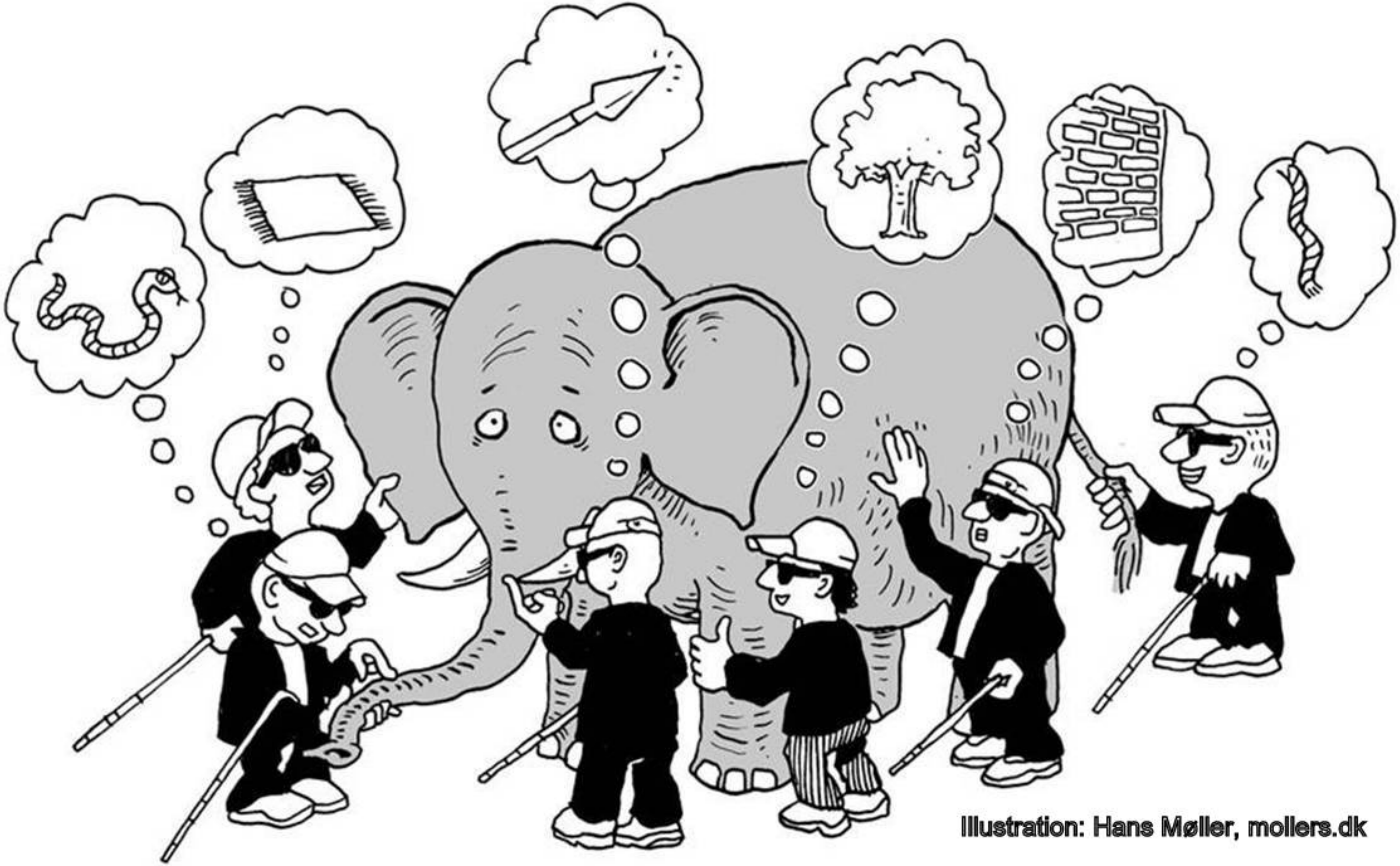


Illustration: Hans Møller, mollers.dk

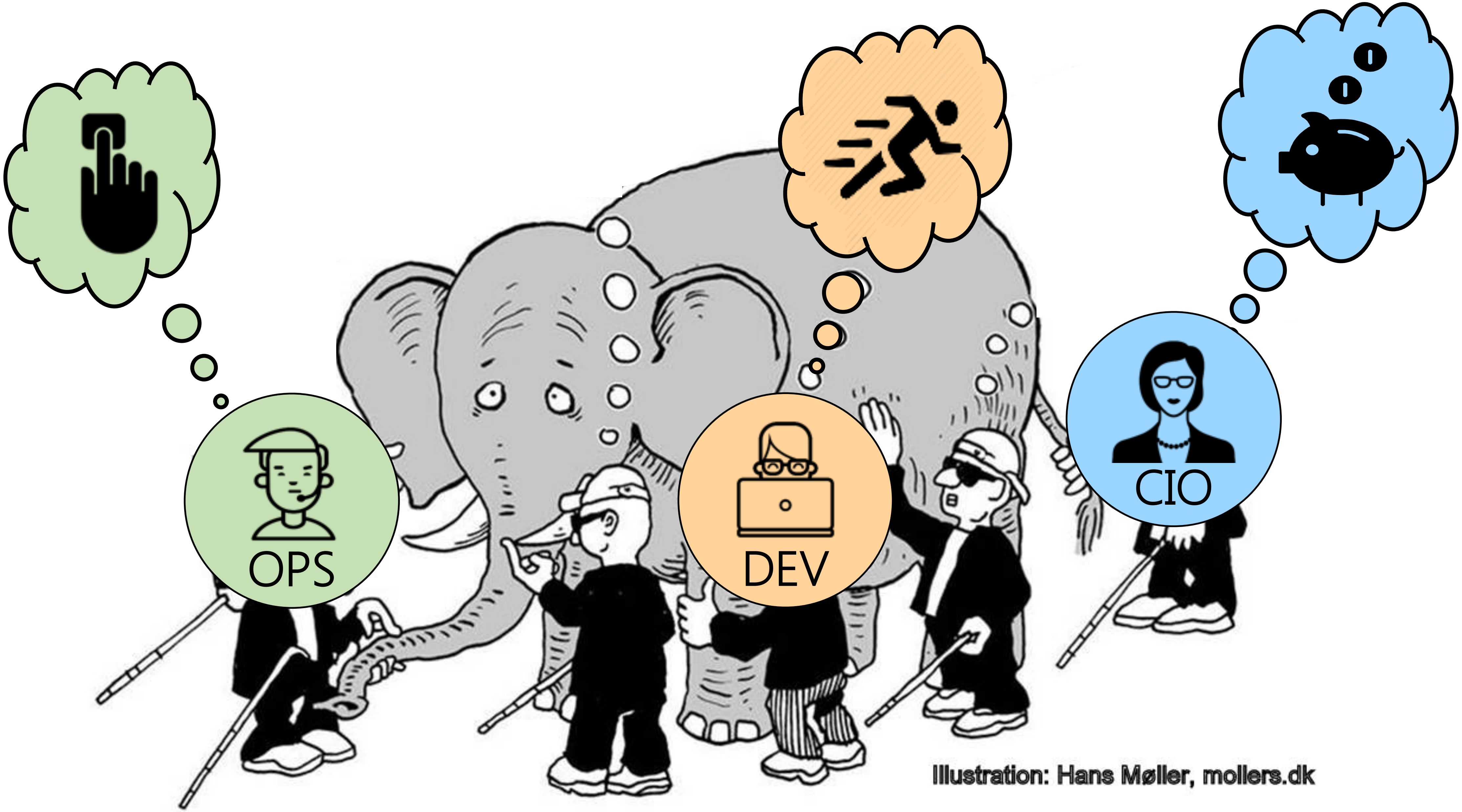
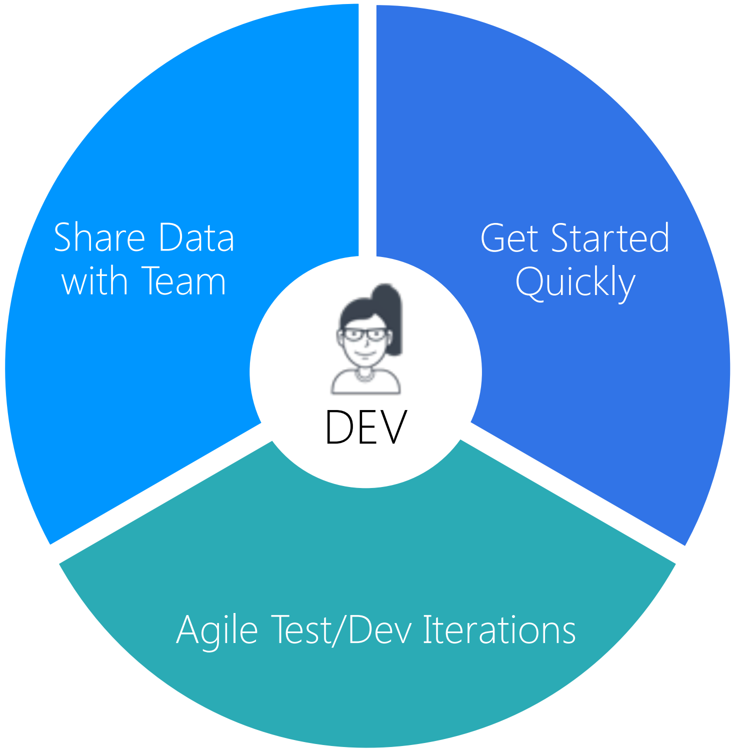
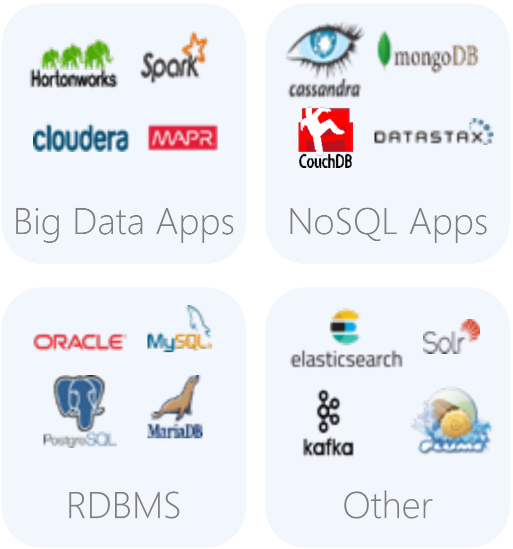
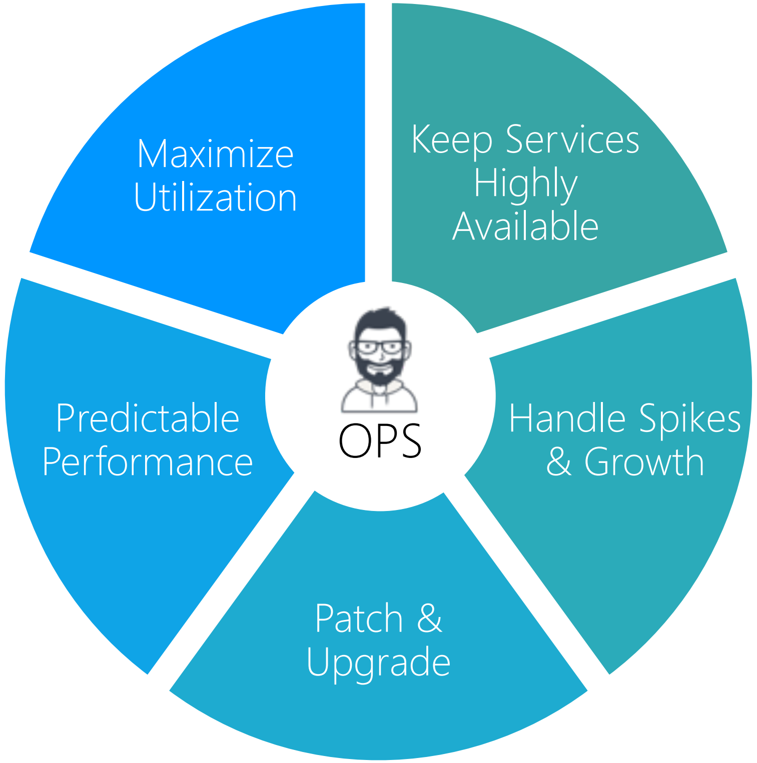


Illustration: Hans Møller, mollers.dk

CHALLENGES IN MANAGING DATA-DRIVEN APPLICATIONS







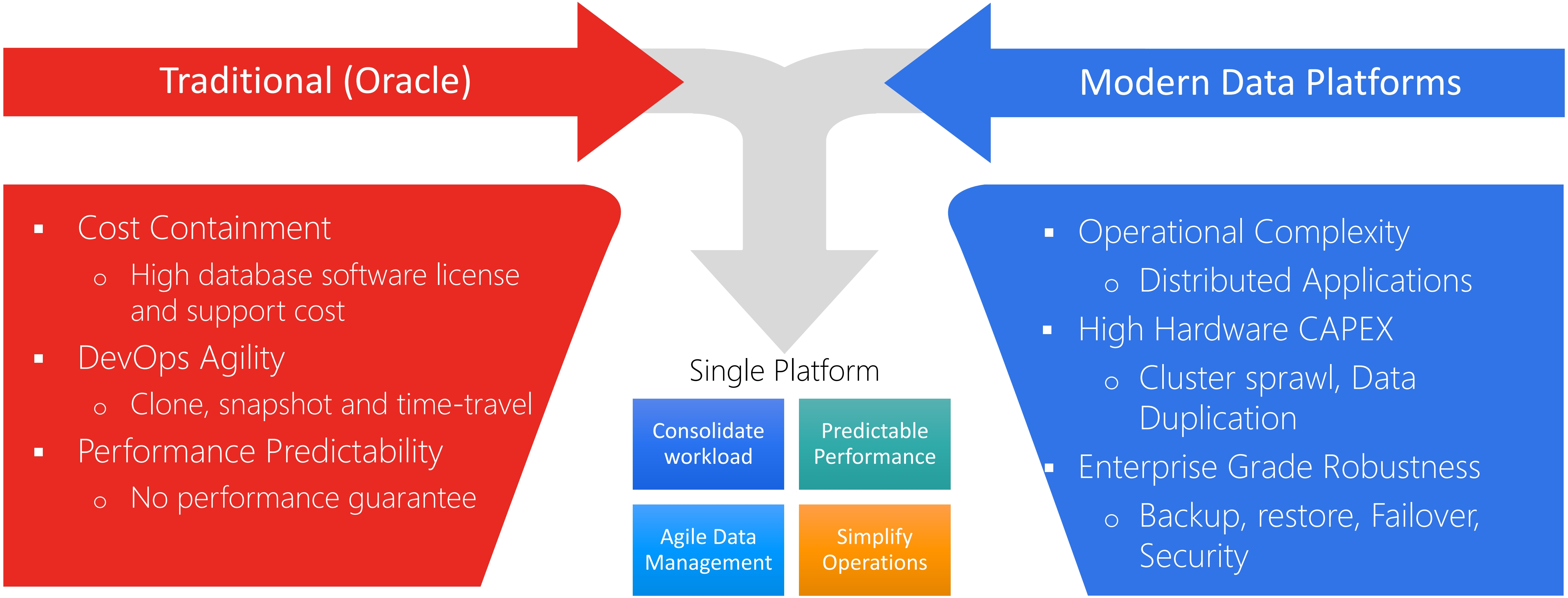
Infrastructure isn't aware of applications

Infrastructure
is the bottleneck

App Diversity/Complexity & Budget
is the bottleneck

EVOLVING DATA MANAGEMENT LANDSCAPE AND CHALLENGES

Enterprises need an unified infrastructure platform for all data applications



Traditional (Oracle)

- Cost Containment
  - High database software license and support cost
- DevOps Agility
  - Clone, snapshot and time-travel
- Performance Predictability
  - No performance guarantee

Modern Data Platforms

- Operational Complexity
  - Distributed Applications
- High Hardware CAPEX
  - Cluster sprawl, Data Duplication
- Enterprise Grade Robustness
  - Backup, restore, Failover, Security

Single Platform

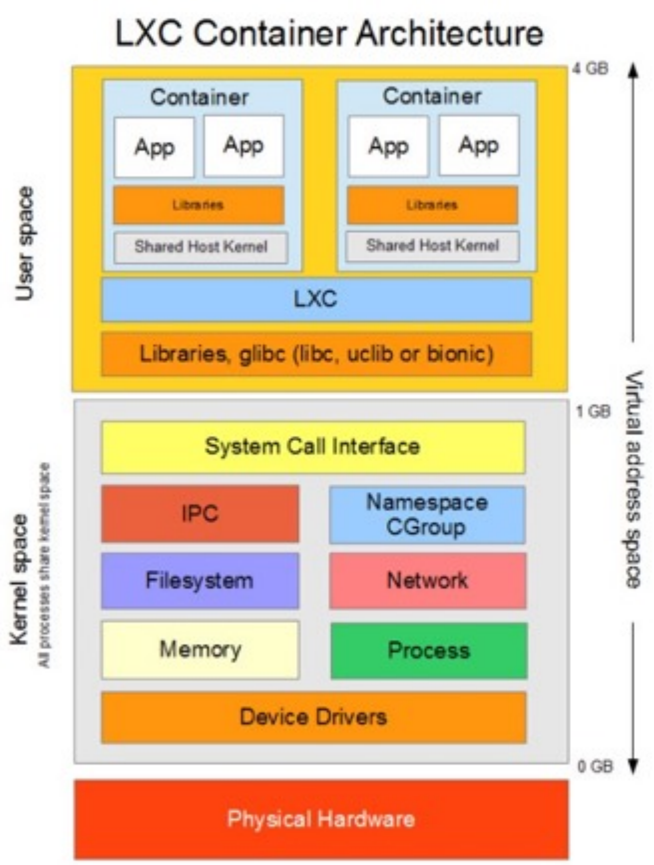
Consolidate workload	Predictable Performance
Agile Data Management	Simplify Operations

EXISTING INFRASTRUCTURE CHOICES ARE INADEQUATE

Bare-Metal	VMs	Purpose-Built Appliances	Containers
Inefficient	Performance Overhead	Very Expensive	?
Complex	VM, DB software Sprawl	Application specific	

WHAT ARE CONTAINERS ?

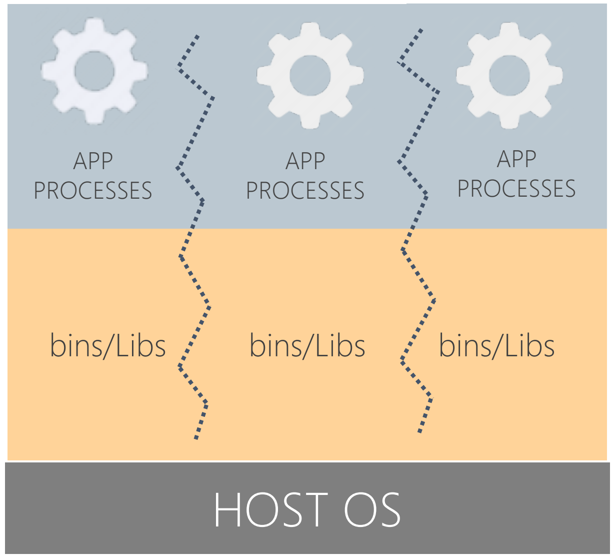
WHAT IS A LINUX CONTAINER ?



- Next Generation of Virtualization technology
- OS-based, and delivers bare metal performance
- Operationally very similar to Virtual Machines
  - Dedicated IP address, file systems, separate user space, etc.
- Agile
  - Containers can be launched in seconds
- Better hardware consolidation
  - 10x or more higher consolidation benefits
- Eliminate OS duplication and application software sprawl

DEPLOYMENT CHOICES

BARE METAL



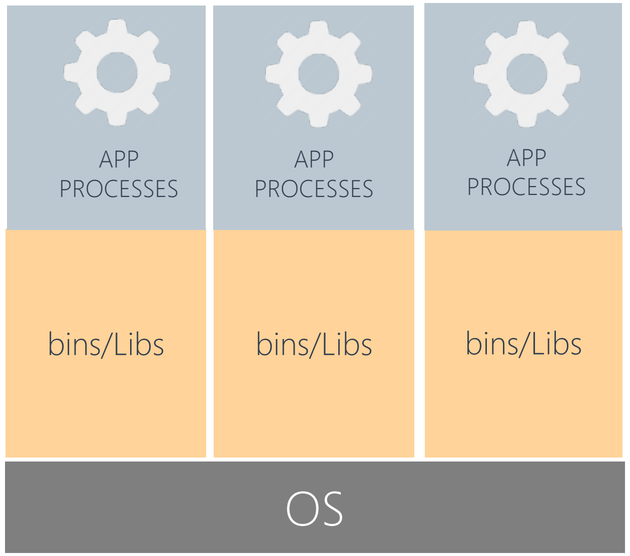
APP PROCESSES | APP PROCESSES | APP PROCESSES

bins/Libs | bins/Libs | bins/Libs

HOST OS

- No Isolation 👎
- 👍 No Performance overhead
- Not Portable

CONTAINERS



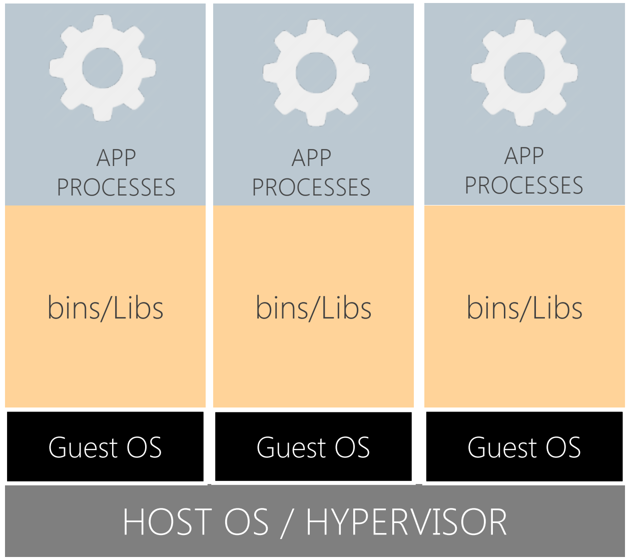
APP PROCESSES | APP PROCESSES | APP PROCESSES

bins/Libs | bins/Libs | bins/Libs

OS

- 👍 Run Time Isolation
- 👍 No Performance overhead
- Portable

VIRTUAL MACHINE



APP PROCESSES | APP PROCESSES | APP PROCESSES

bins/Libs | bins/Libs | bins/Libs

Guest OS | Guest OS | Guest OS

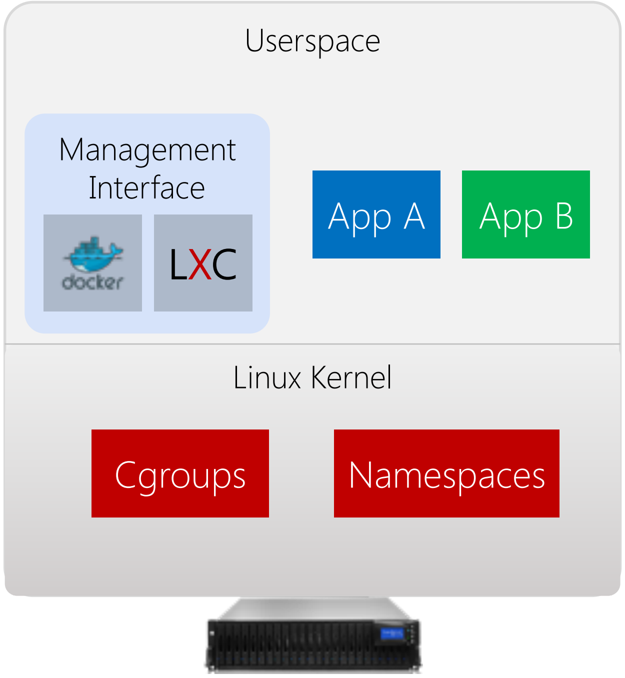
HOST OS / HYPERVISOR

- 👍 Full Isolation
- Performance overhead 👎
- Partially Portable

UNDERSTANDING CONTAINERS

Each container has:

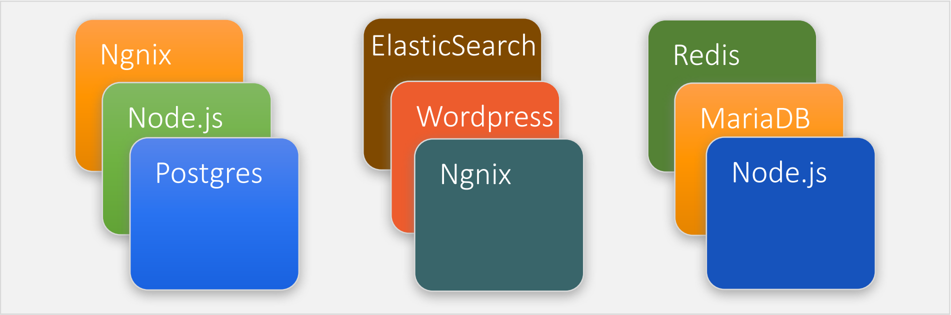
1. Its own network interface (and IP address)
2. Its own file system
3. Isolation (security)
   - Container A & B can't harm (or even see) each other
   - Uses Linux kernel's "**namespaces**" for this
4. Isolation (resource usage)
   - Soft & hard quotas for CPU, RAM and IO
   - Uses Linux kernel's "**cgroups**" for this



Wait, this looks like a virtual machine!
So, what's the difference?

TYPES OF CONTAINERS

APPLICATION CONTAINERS



Ngnix

Node.js

Postgres

ElasticSearch

Wordpress

Ngnix

Redis

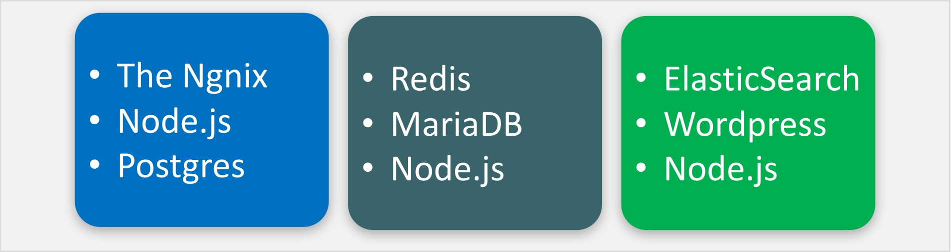
MariaDB

Node.js

- Each container runs a single application (single concern per container philosophy)
- Most popular container format. Example – Docker
- Requires applications to be repackaged and reconfigured to work with Docker image format
- Patch/Upgrade entails replacing container image

Great for Modern Applications

SYSTEM CONTAINERS



- The Ngnix
- Node.js
- Postgres

- Redis
- MariaDB
- Node.js

- ElasticSearch
- Wordpress
- Node.js

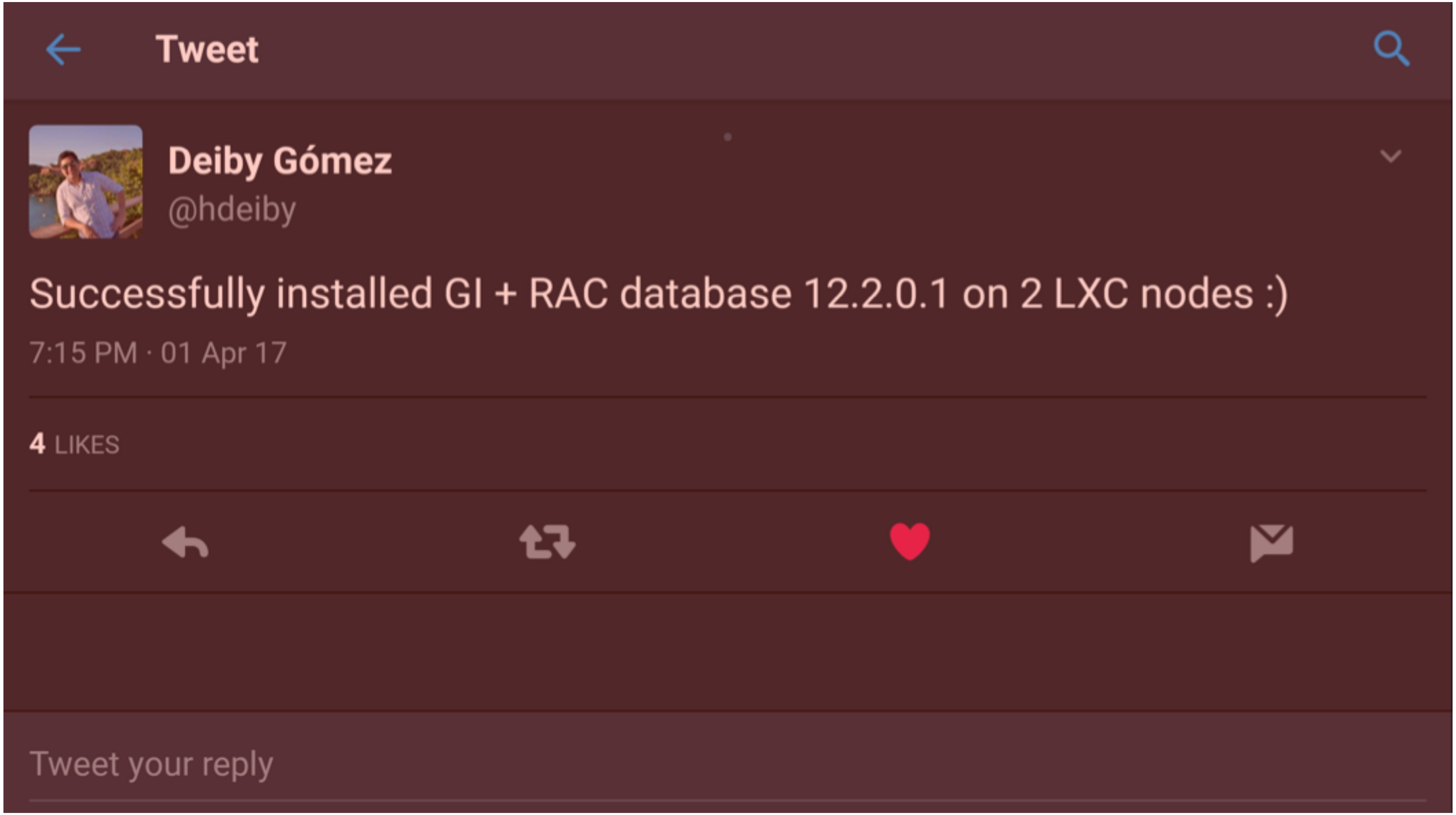
- Each container runs an entire service stack (multiple applications per container)
- Meant to be used as lightweight VM. Examples – LXC, OpenVZ, Solaris Zones
- No need to repackage applications in any special way
- Supports in-place patch/upgrade & SSH access

Great for Traditional Applications

Blog on LXC vs Docker: https://robinsystems.com/blog/containers-deep-dive-lxc-vs-docker-comparison/

ORACLE CONTAINER SUPPORT POLICY

- "Starting with Oracle Database 12c Release 1 (12.1.0.2), Linux Containers are supported on Oracle Linux 7 and Oracle Linux 6 and certified on Linux x86-64 systems".
- Oracle Database is ***certified*** on Oracle Linux, Red Hat Enterprise Linux and SUSE Enterprise Linux in LXC.
- Oracle Database is ***supported*** ONLY on Oracle Linux UEK in LXC.
- OS requirements for supported Oracle DB in Linux Containers:
  - Oracle Linux 7: 3.8.13-98.el7uek.x86_64 (or later)
  - Oracle Linux 6: 3.8.13-98.el6uek.x86_64 (or later)



Tweet

Deiby Gómez
@hdeiby

Successfully installed GI + RAC database 12.2.0.1 on 2 LXC nodes :)

7:15 PM · 01 Apr 17

4 LIKES

ORACLE SUPPORTS BOTH DOCKER AND LXC

Official Oracle on Docker repository:
https://github.com/oracle/docker-images

- General
  - OracleJava
  - OpenJDK
- Database
  - RDBMS
  - MySQL
  - NoSQL
- Middleware
  - Glassfish
  - WebLogic
  - Coherence
  - Tuxedo
  - HTTP Server

WELCOME TO THE

Oracle Container Registry

The Oracle Container Registry provides Oracle tested and approved Docker and other container images for Oracle products.

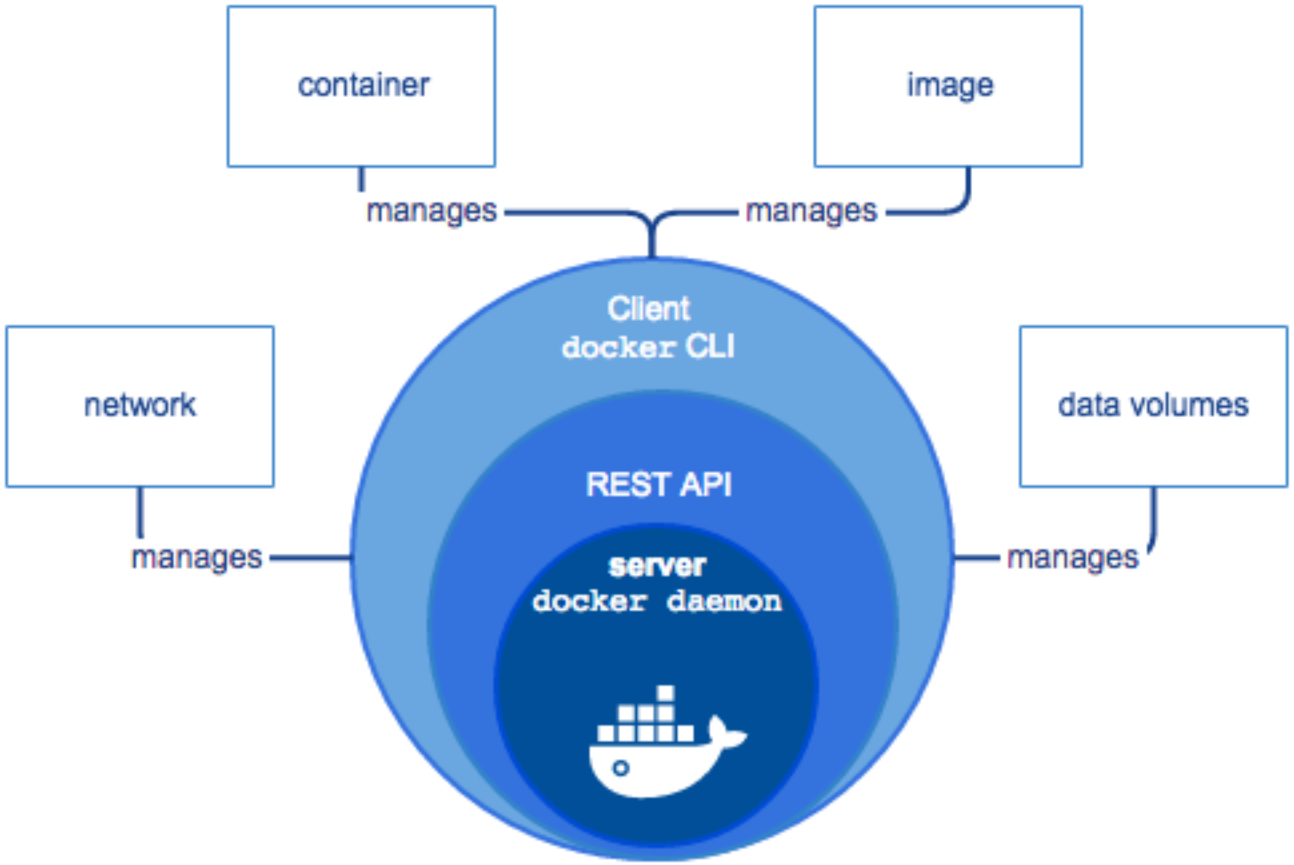
You must accept licensing terms before using Docker or other tools to pull images into your local registry or environment.

Please sign in to continue.

	Oracle Database	Oracle RAC (CRS + ASM)
LXC (Test & Production)	✓	✓
Docker (Dev Only)	✓	✗

LXC Support: http://www.oracle.com/technetwork/database/virtualizationmatrix-172995.html

DOCKER ENGINE

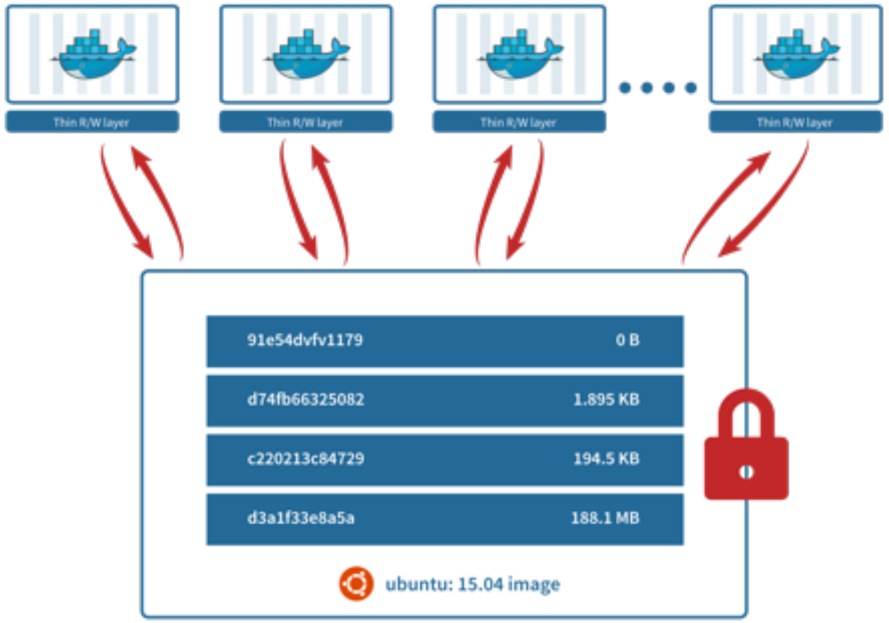


Docker Engine: Responsible for managing networking, images, containers, volumes, plugins, orchestration, etc

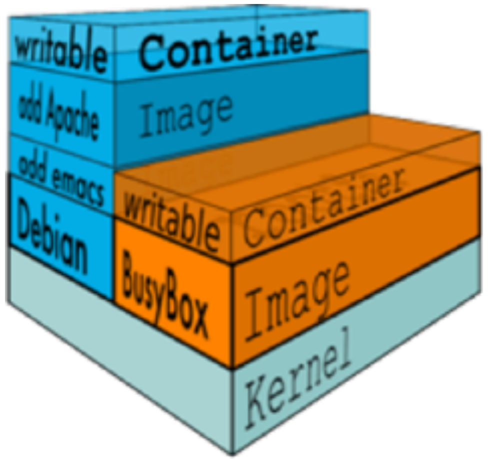
Install Docker Engine on your laptops and servers

IMAGES

- Images are made up of multiple r/o layers
- Containers are a thin r/w layer on top
- Layers are shared by multiple images
- Storage drivers
  - AUFS (Ubuntu, OSX)
  - Device mapper (RHEL, CentOS)
  - BTRFS (Oracle Linux)
  - Overlay, Overlay2
  - ZFS
- Layer default location: /var/lib/docker

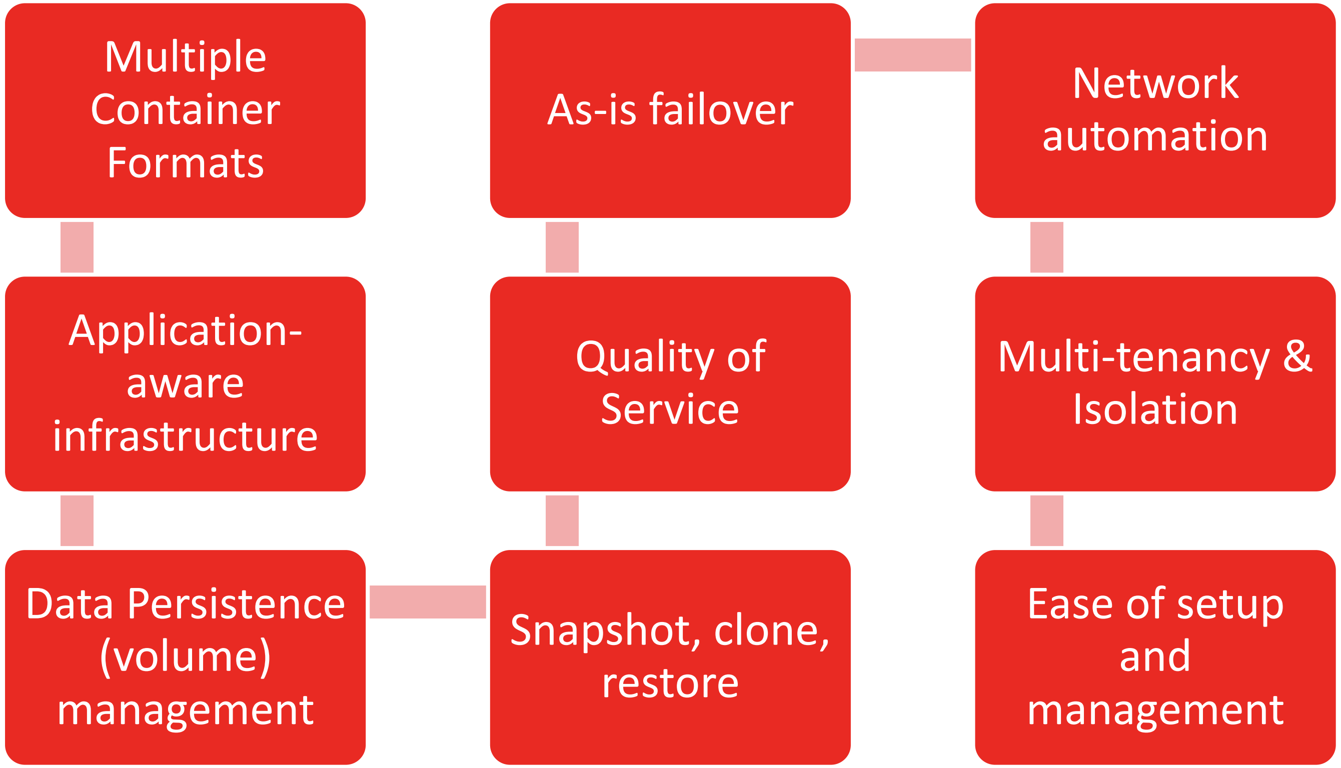


91e54dvfv1179	0 B
d74fb66325082	1.895 KB
c220213c84729	194.5 KB
d3a1f33e8a5a	188.1 MB



THE DATABASE CONTAINERIZATION PLATFORM CHECKLIST

Guiding Principle:
Do what's right for the application/database!



Multiple Container Formats

Application-aware infrastructure

Data Persistence (volume) management

Snapshot, clone, restore

Quality of Service

As-is failover

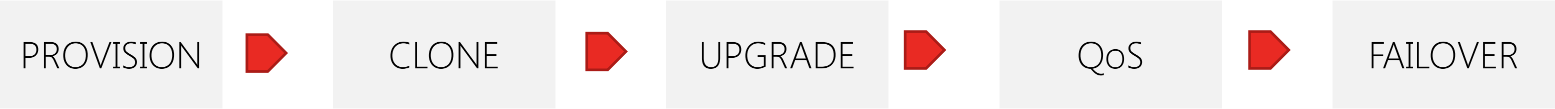
Network automation

Multi-tenancy & Isolation

Ease of setup and management

Original Blog: http://www.robinsystems.com/blog/stateful-database-containerization-platform-checklist/

Robin for Oracle



PROVISION → CLONE → UPGRADE → QoS → FAILOVER

Let Applications drive the Infrastructure

Embed Application Lifecycle Management Primitives in
Compute, Network and Storage Infrastructure

Hortonworks Spark
cloudera MAPR
Big Data Apps

cassandra mongoDB
CouchDB DATASTAX
NoSQL Apps

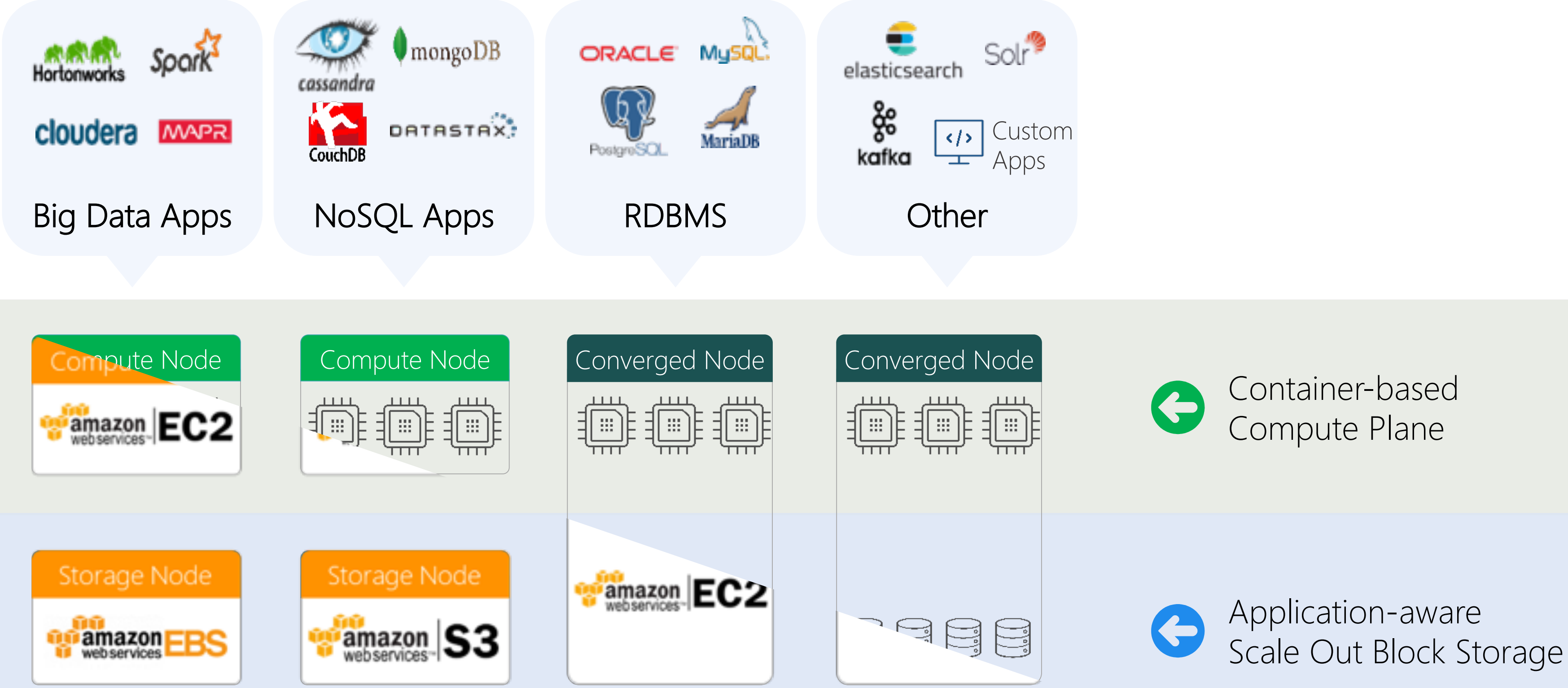
ORACLE MySQL
PostgreSQL MariaDB
RDBMS

elasticsearch Solr
kafka Custom Apps
Other

Containers + Integrated Scale-Out Block Storage + Integrated Networking + Application-aware Workflow Manager

HOW ROBIN IS DEPLOYED

PUSH BUTTON APPLICATION LIFECYCLE MANAGEMENT	
1	DEPLOY
2	MANAGE
3	DATA SHARING
4	QoS
5	SCALE
6	PERFORMANCE
7	SECURITY



Big Data Apps

NoSQL Apps

RDBMS

Other

Custom Apps

Compute Node

Compute Node

Converged Node

Converged Node

Storage Node

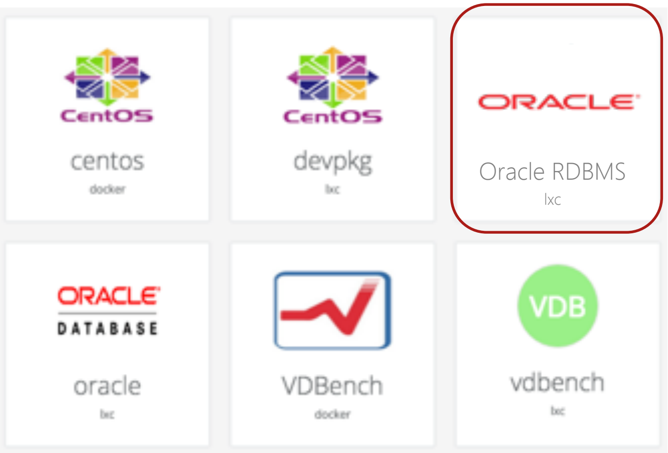
Storage Node

Container-based Compute Plane

Application-aware Scale Out Block Storage

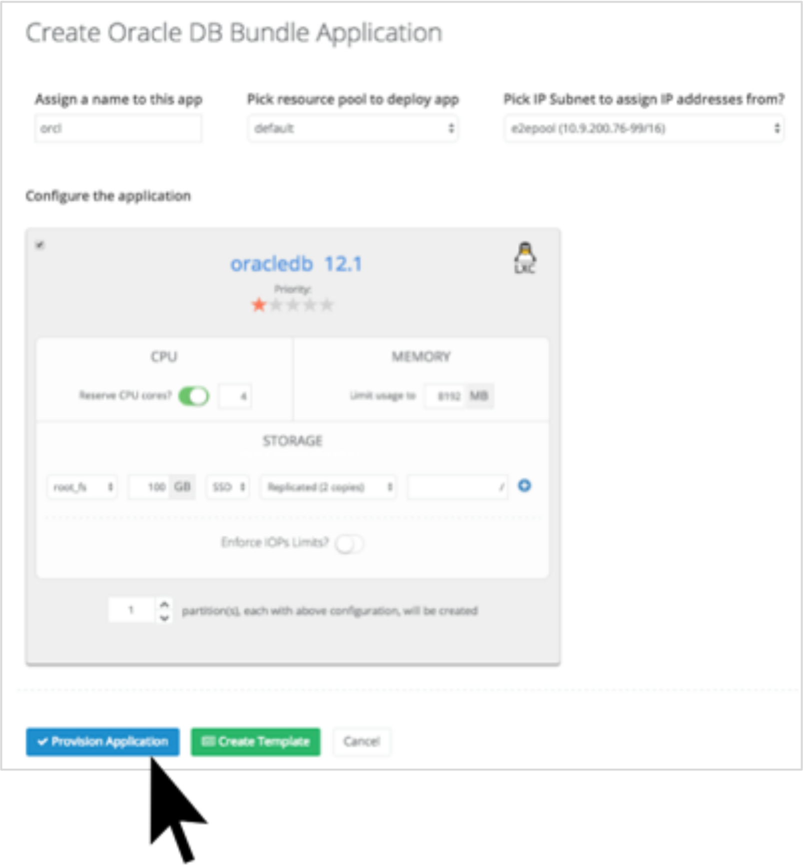
DEPLOY ANY APPLICATION WITH PUSH OF A BUTTON

Describe Application needs in a YAML file and upload to Robin AVP as an "Application Bundle"



Select application from uploaded bundles

1



Create Oracle DB Bundle Application

Assign a name to this app: orcl

Pick resource pool to deploy app: default

Pick IP Subnet to assign IP addresses from?: e2epool (10.9.200.76-99/16)

Configure the application

oracledb 12.1

Priority: ★☆☆☆☆

CPU

Reserve CPU cores? ● 4

MEMORY

Limit usage to 8192 MB

STORAGE

root_fs | 100 GB | SSD | Replicated (2 copies) | /

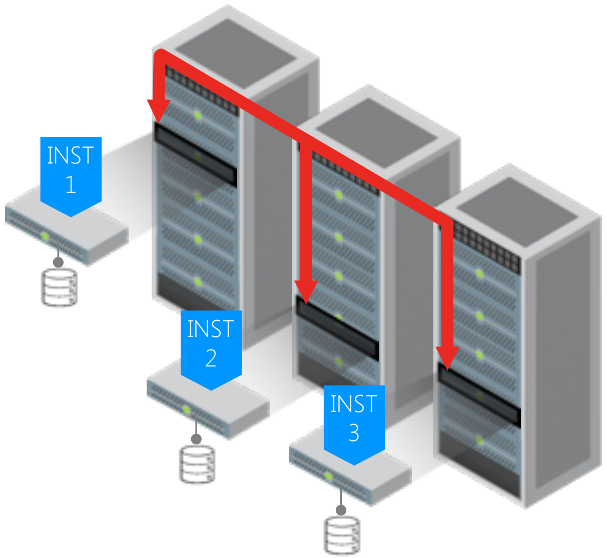
Enforce IOPs Limits? ○

1 partition(s), each with above configuration, will be created

Provide required inputs

2

1. Robin determines best resources to deploy the application
   (System utilization, isolation policies, licensing restrictions, etc)
2. Storage volumes are allocated based app needs
   (Media type, QoS, Affinity rules)
3. Open vSwitch is programmed to network all components
4. App-specific hooks are executed to customize application

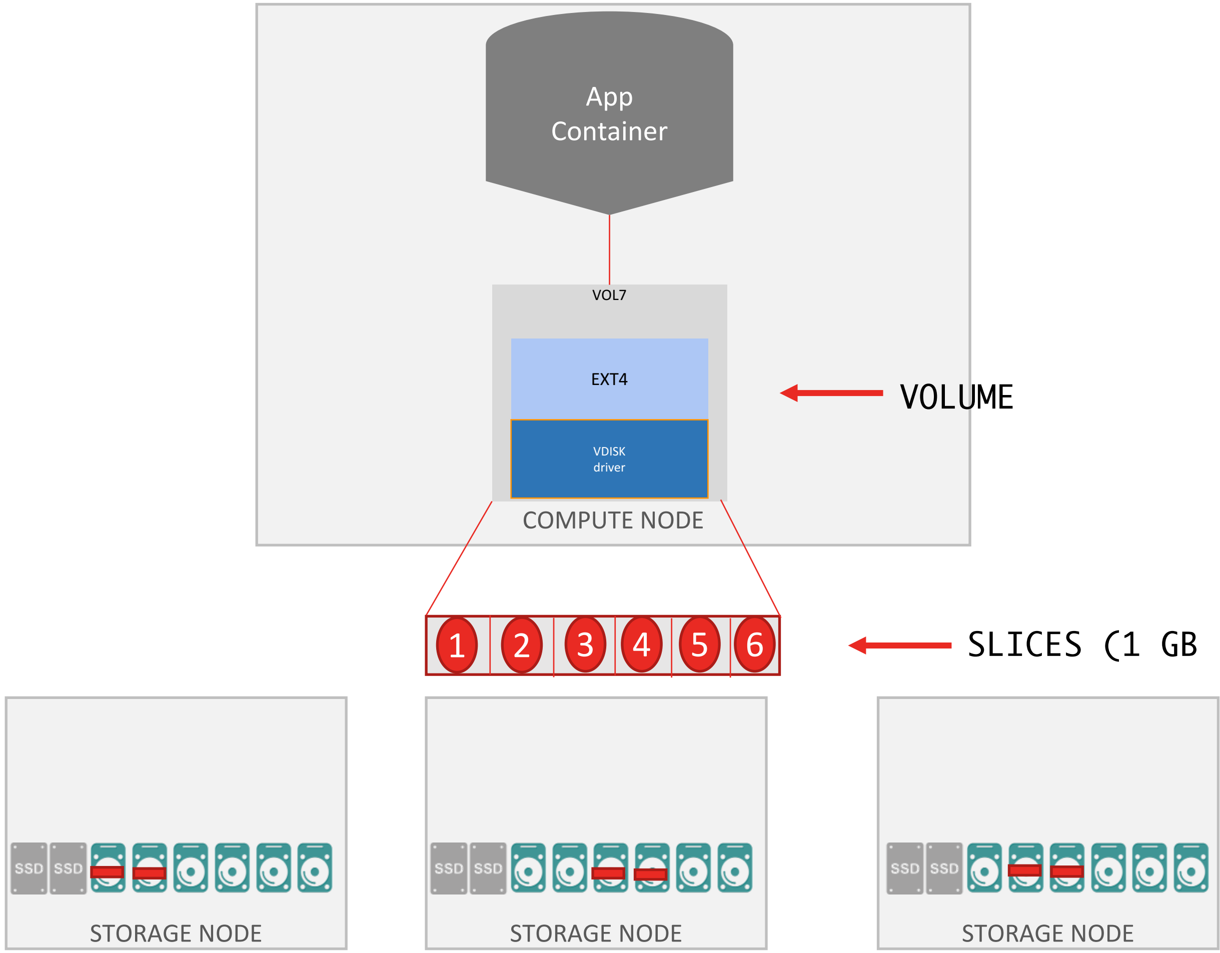


Application is brought online in minutes

3

PROVISION | ONBOARD | CLONE | QoS | FAILOVER

STORAGE OVERVIEW

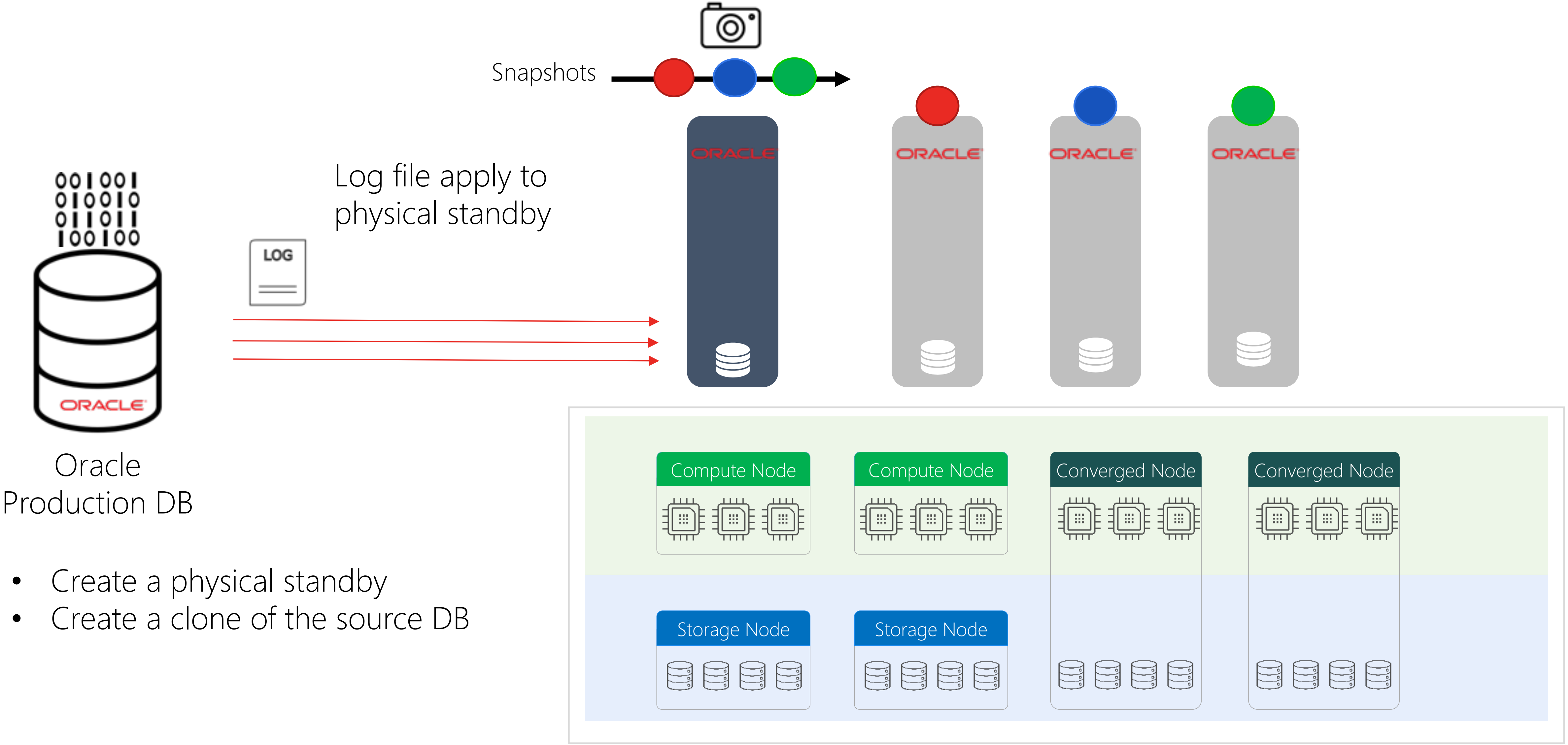


VOLUME

SLICES (1 GB CHUNKS)

SLICES ARE
BUILT ON 32MB
SEGMENTS

ONBOARDING: HOW DOES IT WORK ?

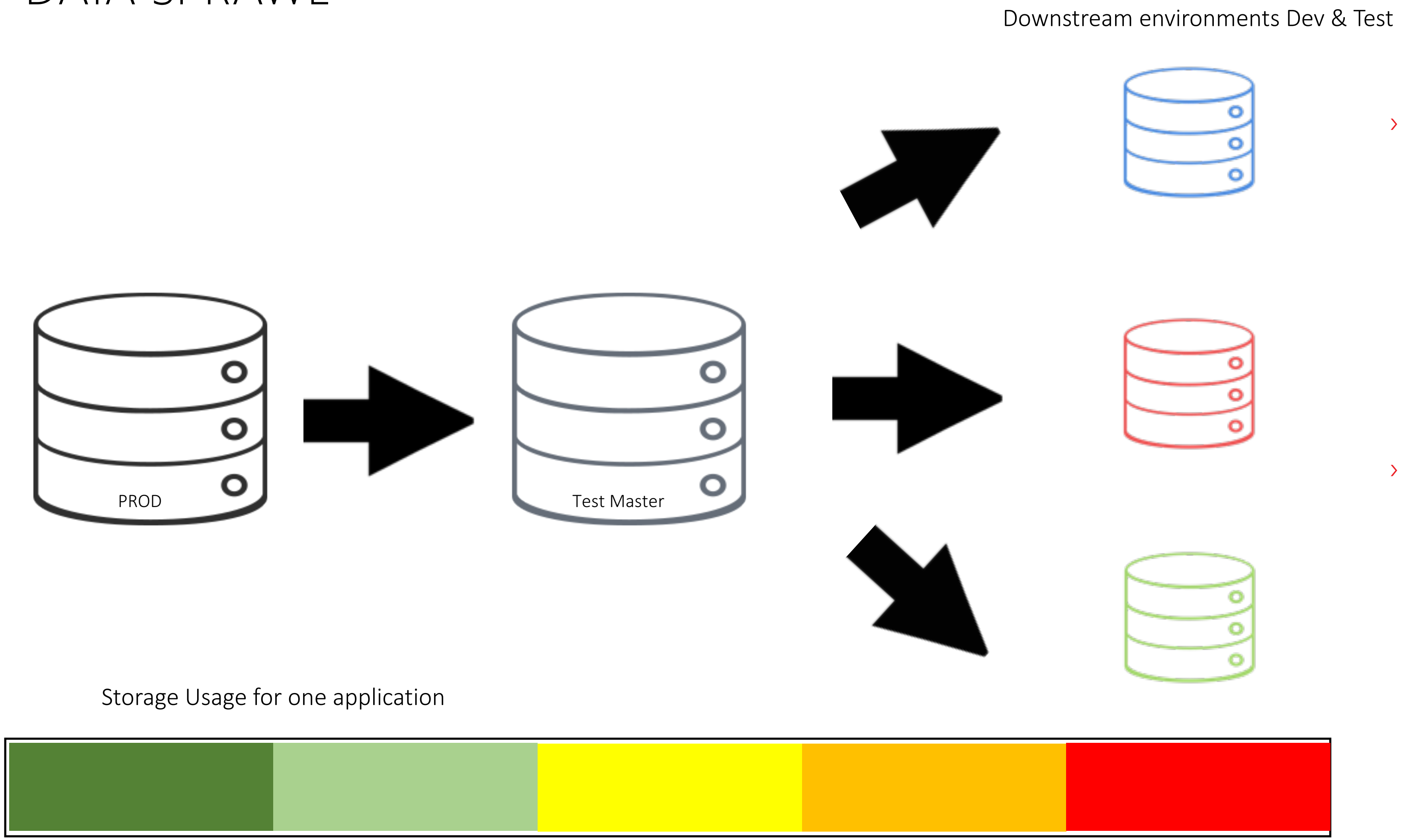


- Create a physical standby
- Create a clone of the source DB

PROVISION | ONBOARD | CLONE | QoS | FAILOVER

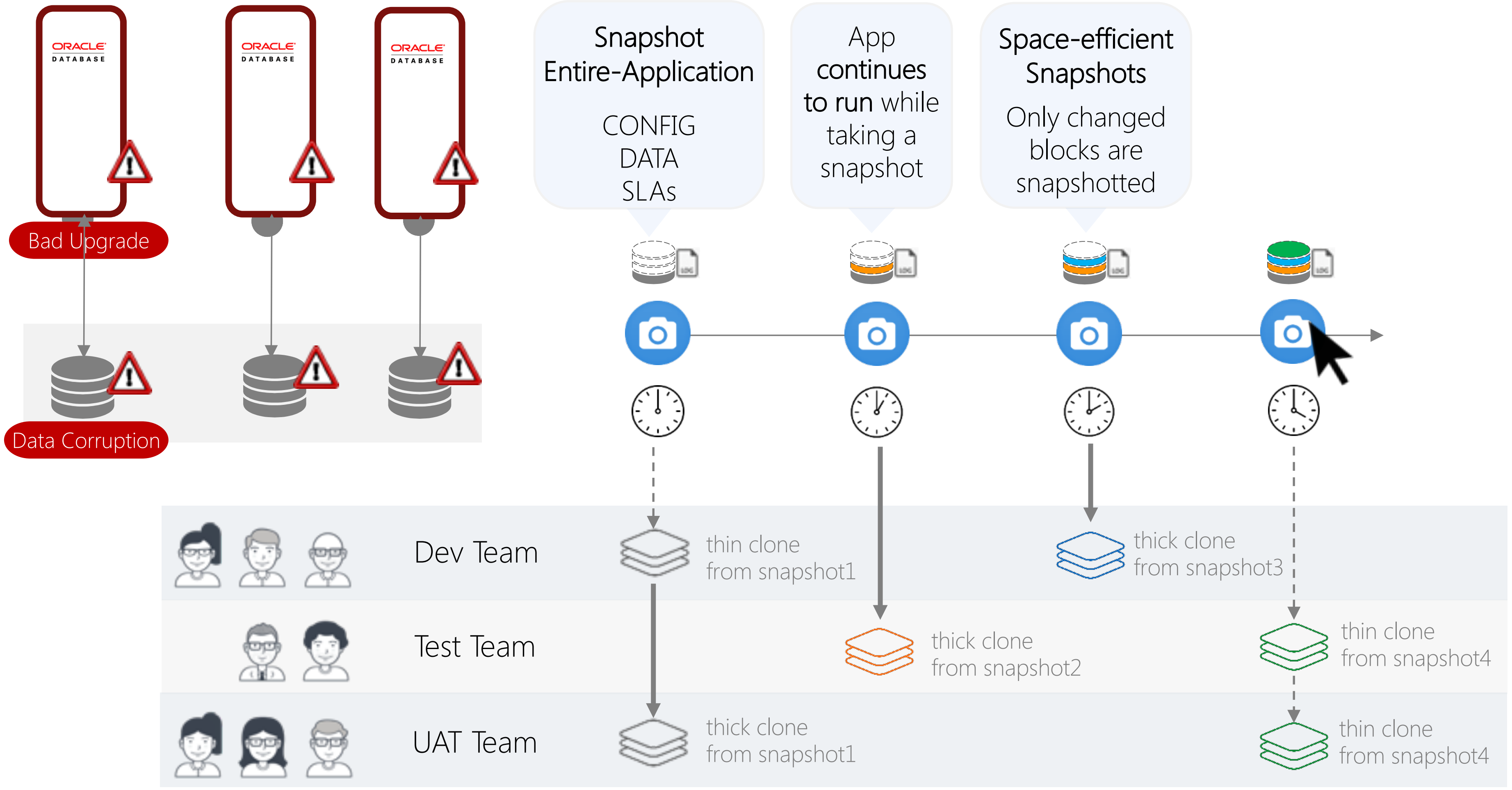
ROBIN

DATA SPRAWL


Downstream environments Dev & Test
PROD
Test Master
Storage Usage for one application

- Full Clones of your production database
  - Time consuming
  - Results in Data Sprawl
  - Create Test master to avoid performance penalty
- What if , you create as many environments but not pay the storage penalty ?

MANAGE DATA SHARING (SNAPSHOTS AND CLONES)

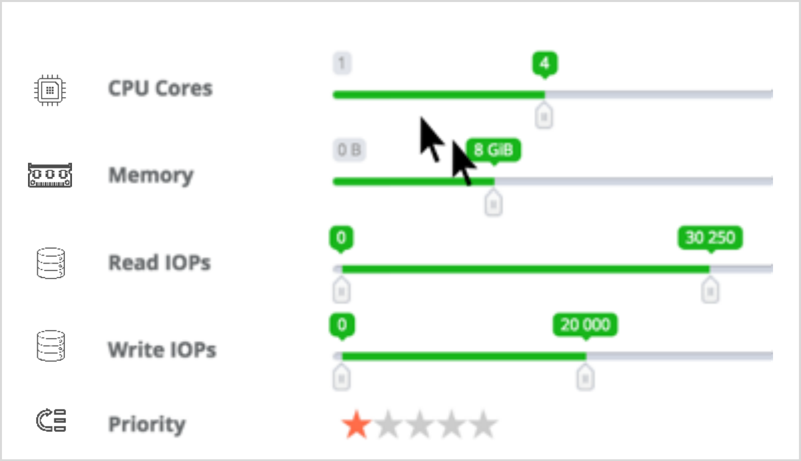

ORACLE DATABASE
ORACLE DATABASE
ORACLE DATABASE
Bad Upgrade
Data Corruption
Snapshot Entire-Application
CONFIG
DATA
SLAs
App continues to run while taking a snapshot
Space-efficient Snapshots
Only changed blocks are snapshotted
Dev Team
thin clone from snapshot1
thick clone from snapshot3
Test Team
thick clone from snapshot2
thin clone from snapshot4
UAT Team
thick clone from snapshot1
thin clone from snapshot4

Common Push-button Snapshot & Clone **Workflow for all Apps** to

App Time Travel
Create Test/Dev Copies
Validate Upgrades
Protect Data/Backup

PROVISION | ONBOARD | **CLONE** | QoS | FAILOVER

HANDLING TRANSIENT SPIKES USING INSTANT SCALE-UP



CPU Cores 1 – 4

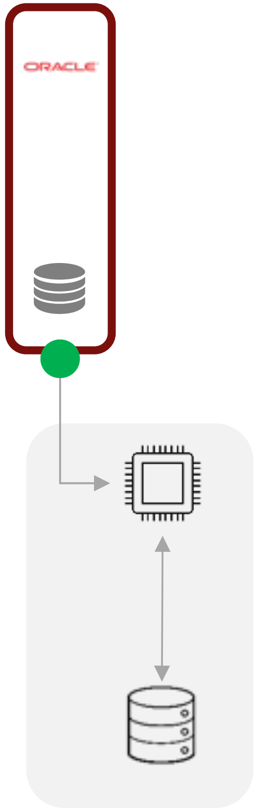
Memory 0 B – 8 GiB

Read IOPs 0 – 30 250

Write IOPs 0 – 20 000

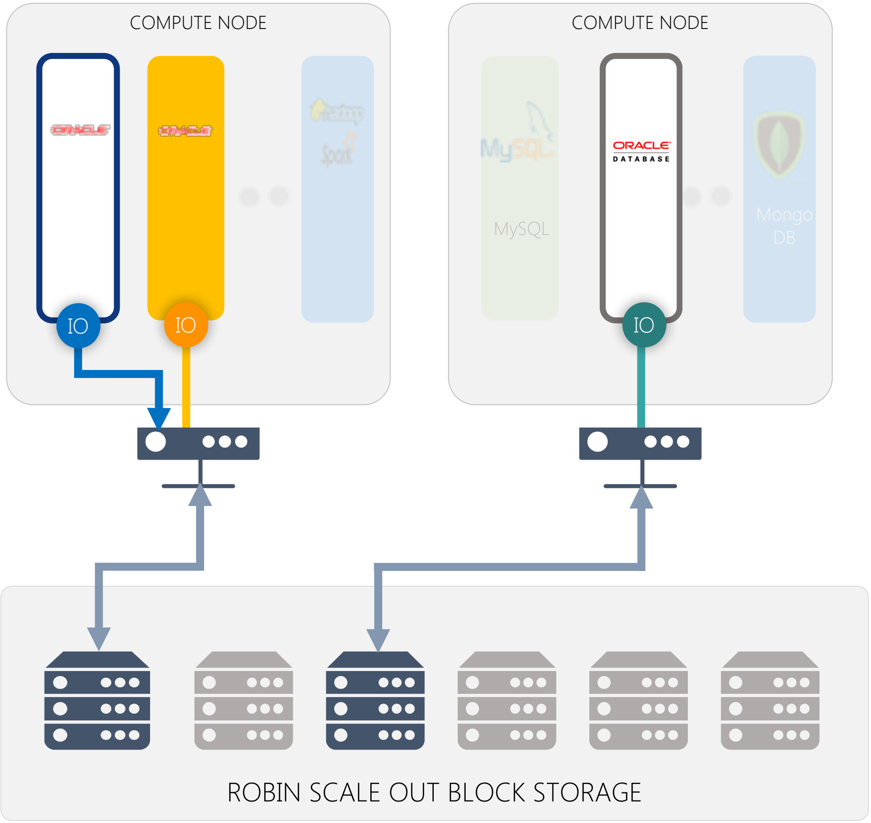
Priority ★☆☆☆☆

Oracle DB – Scale UP



- ✓ On-demand Instant Scale-up to handle transient spikes
  - No data movement overhead
  - No need to stop the cluster
  - Ideal to meet temporary or seasonal demand

END-TO-END QUALITY OF SERVICE



MAX IOPS enforced here to throttle usage

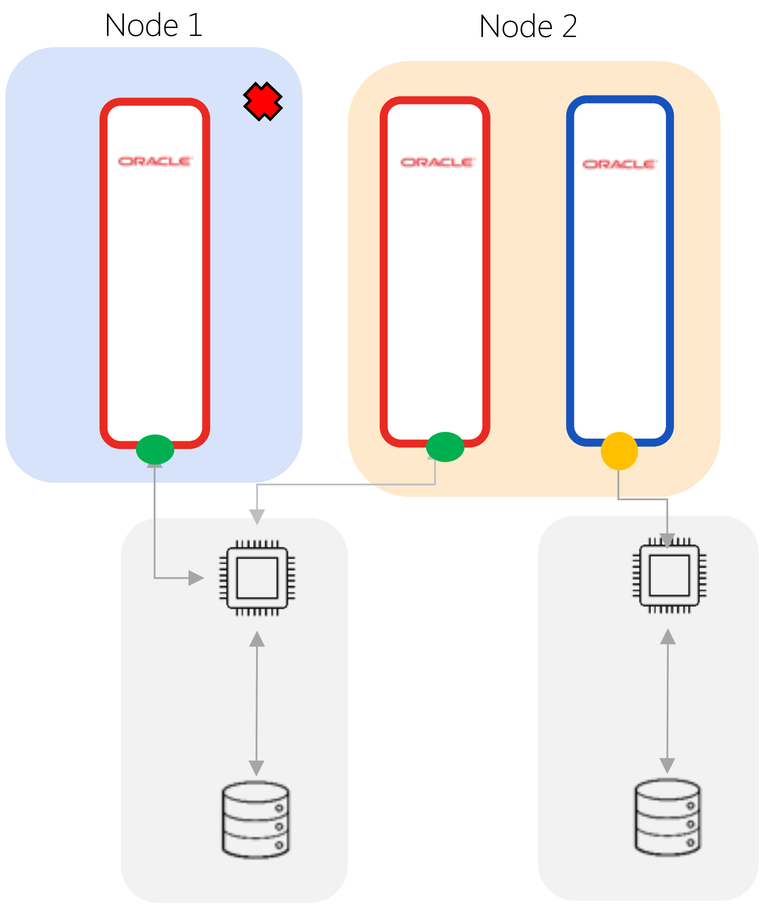
MIN IOPS & IO PRIORITIES enforced here to deliver predictability

Robin AVP controls end-to-end IO pipeline (App-to-Disk) making it the only product to truly enforce QoS

PROVISION | ONBOARD | CLONE | QoS | FAILOVER

AUTOMATIC FAILOVER

- Robin automatically detects container failure
- Robin Application monitor brings up an Oracle container image in seconds in another compute node
- No need to move storage. Robin automatically mounts the storage volumes to the new container



CONSOLIDATION OPTIONS FOR ORACLE DATABASES

Options → Criteria ↓	Virtual Machine	Oracle 12c Multitenant	Containers on Robin
Performance overhead	Significant Hypervisor layer, Guest OS	Complicated Shared Redo logs, Noisy neighbors problem	Negligible Completely independent, no hypervisor
Availability	High One VM doesn't impact another one	Medium CDB shutdown takes down all PDBs with it	High Just like VMs
Isolation	Excellent	Good Shared buffer cache	Excellent
Performance predictability	Poor Cannot cap IOPS at the hypervisor layer	Good IOPS control only available on Exadata	Excellent. Built into the platform
Agility	Good	Excellent	Excellent
Manageability	High OS sprawl	High Challenges in getting patching window	Low No OS sprawl, no additional licenses

https://robinsystems.com/product-solutions/robin-oracle/

ROBIN

DEMO

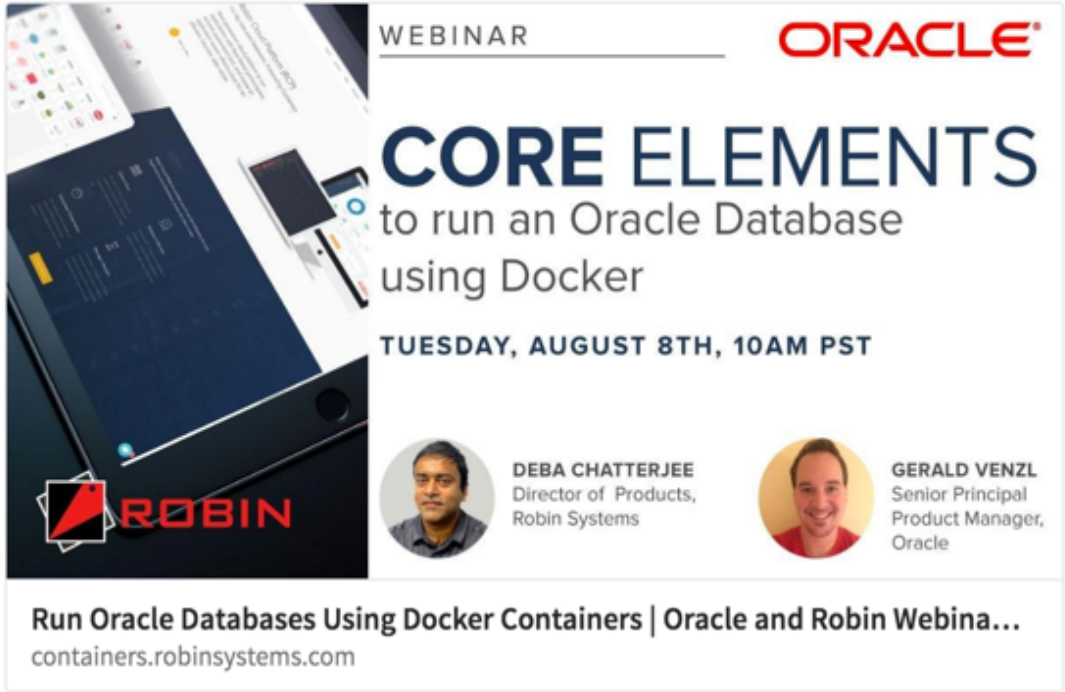
- Oracle Database lifecycle management with Docker on Robin
- IOPS management of Oracle

JOINT WEBINAR – ROBIN SYSTEMS & ORACLE – TUESDAY 8TH AUGUST, 10 AM PST

See Docker Benefits In Action

Consolidate, Agility, QoS

In this joint webinar by Robin Systems and Oracle Corporation, we will go over the essentials that you need to run the Oracle database inside a Docker container. We will also explore the core elements required to use containers to consolidate databases without compromising performance, while guaranteeing isolation and no manageability changes.



Run Oracle Databases Using Docker Containers | Oracle and Robin Webina...

containers.robinsystems.com

THANK YOU

🌐 www.robinsystems.com ✉ info@robinsystems.com