NYOUG FALL 2016

ORACLE LINUX CONTAINER UPDATE

PRESENTER INFORMATION

- Gilbert Standen
- Principal Solutions Architect, Robin Systems:
- http://www.robinsystems.com
- Creator of Orabuntu-LXC:
- https://github.com/gstanden/orabuntu-lxc
- Creator of nandydandyoracle blog:
- https://sites.google.com/site/nandydandyoracle
- Presenter, NYOUG, 2007, 2008, 2014, 2016
- Presenter, AUSOUG, "Oracle Communities with 20:20 Foresight", 2006, 2007
- Presenter, (Violin Memory All Flash Arrays), Oracle Open World, 2014

CREDITS AND RECOGNITION

- Margarett Roy, Executive Director, NYOUG
- My mother, Mary Elizabeth “Betty” Standen (nee Bell)
- My father, who believed you can be/do anything
- The woman to whom I am married, Dr. Yelena Belyaeva-Standen
- The Robin Systems team

ROBIN SYSTEMS AT CASSANDRA SUMMIT.

ROBIN SYSTEMS WILL BE AT ORACLE OPEN WORLD NEXT WEEK!
PLEASE VISIT US AT OOW FOR FUN, PROGRESS, AND PRIZES!

ROBIN

What is Happening with Containers Lately?

Application-Defined Containerization

Software-defined Intelligent Infrastructure
Containers + Application-aware Storage + Networking + Application Orchestration

SIMPLIFY
Application Management
Increase Agility

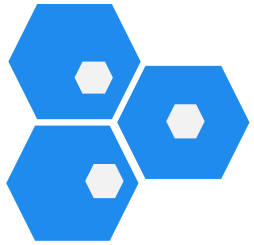
GUARANTEE
Performance and QoS

MAXIMIZE
Capacity Utilization
Reduce Cost

info@robinsystems.com

ROBIN

Why Containers?



Stateless

Stateful

Data Apps

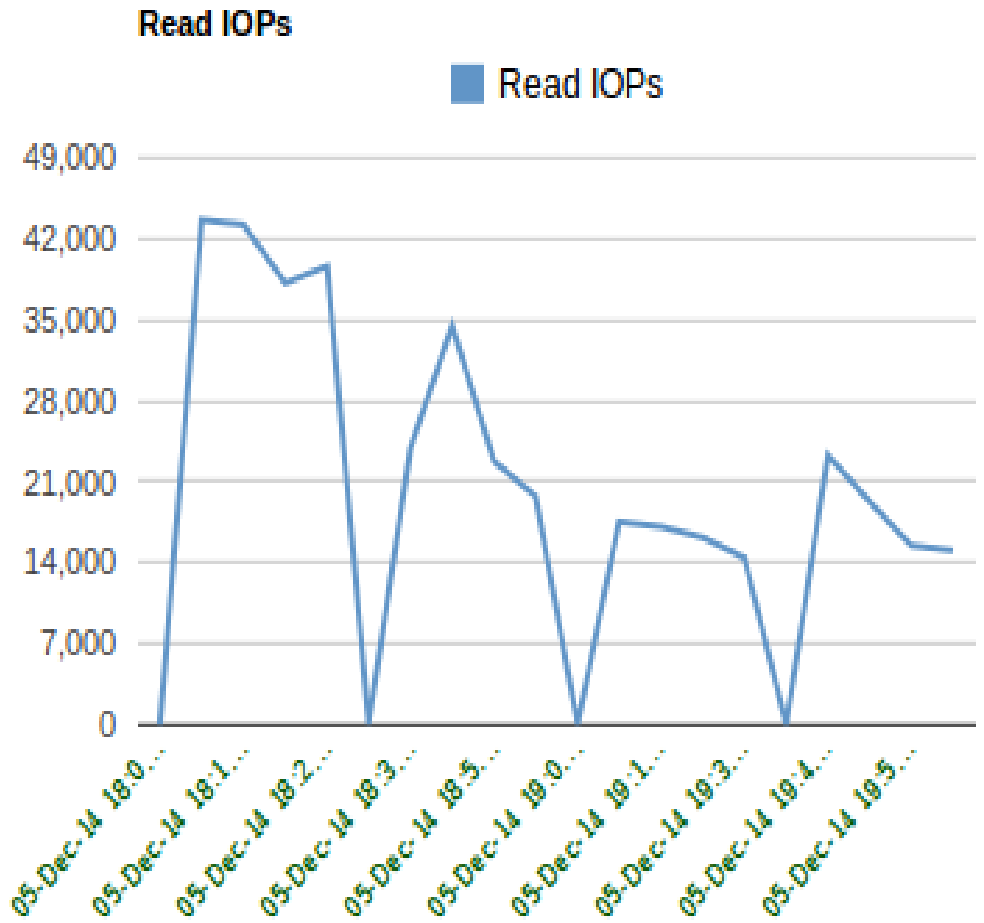
Enterprise Apps

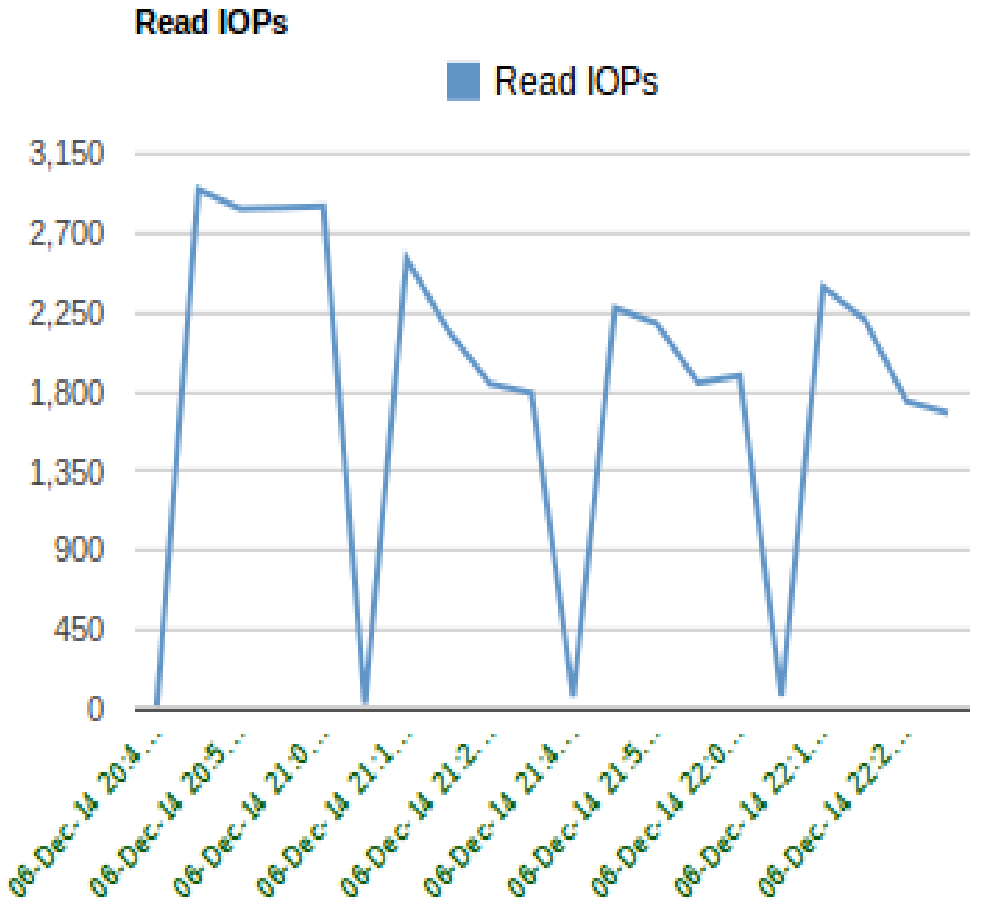
- Break out of dependency hell
- Build, ship, deploy anywhere
- Manageable
- Bare-metal performance
- Pure Linux skillsets

- Lightweight, high-performance "virtualization"
- Multi-tenancy with bare-metal performance
- Higher consolidation density than VMs
- Simplify ops by eliminating OS or VM sprawl
- Just a standard Linux feature, available with all major Linux distributions

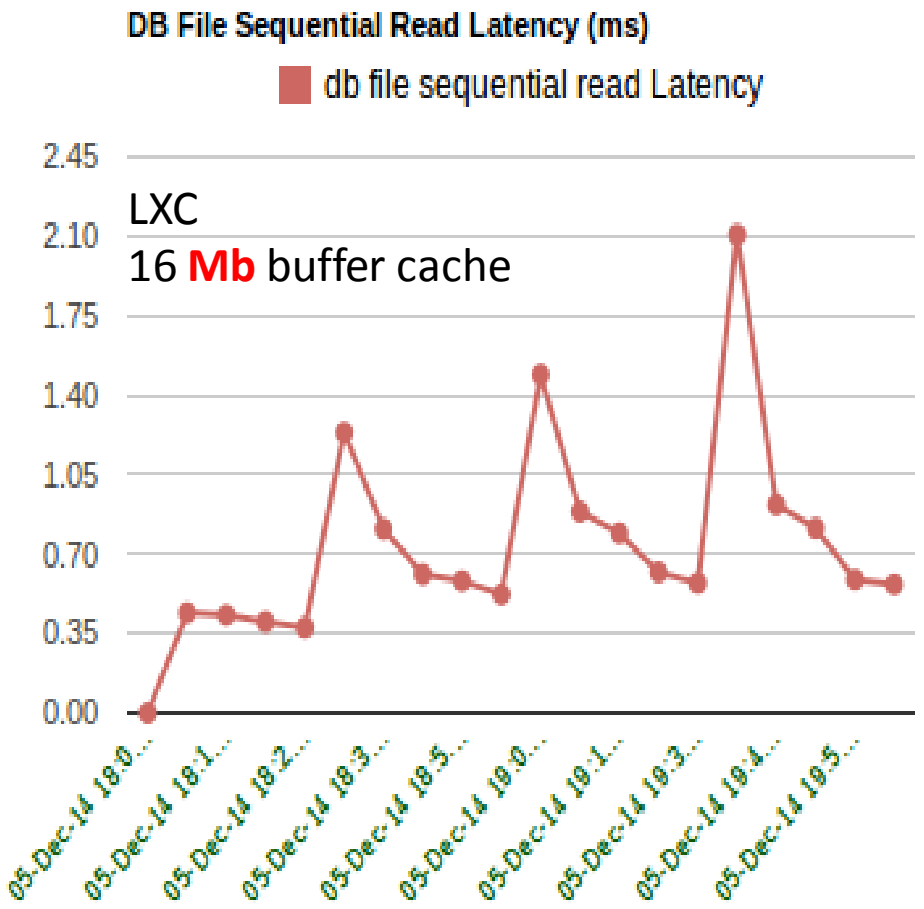
CONTAINERS ARE THE RIGHT VEHICLE FOR THE JOB!

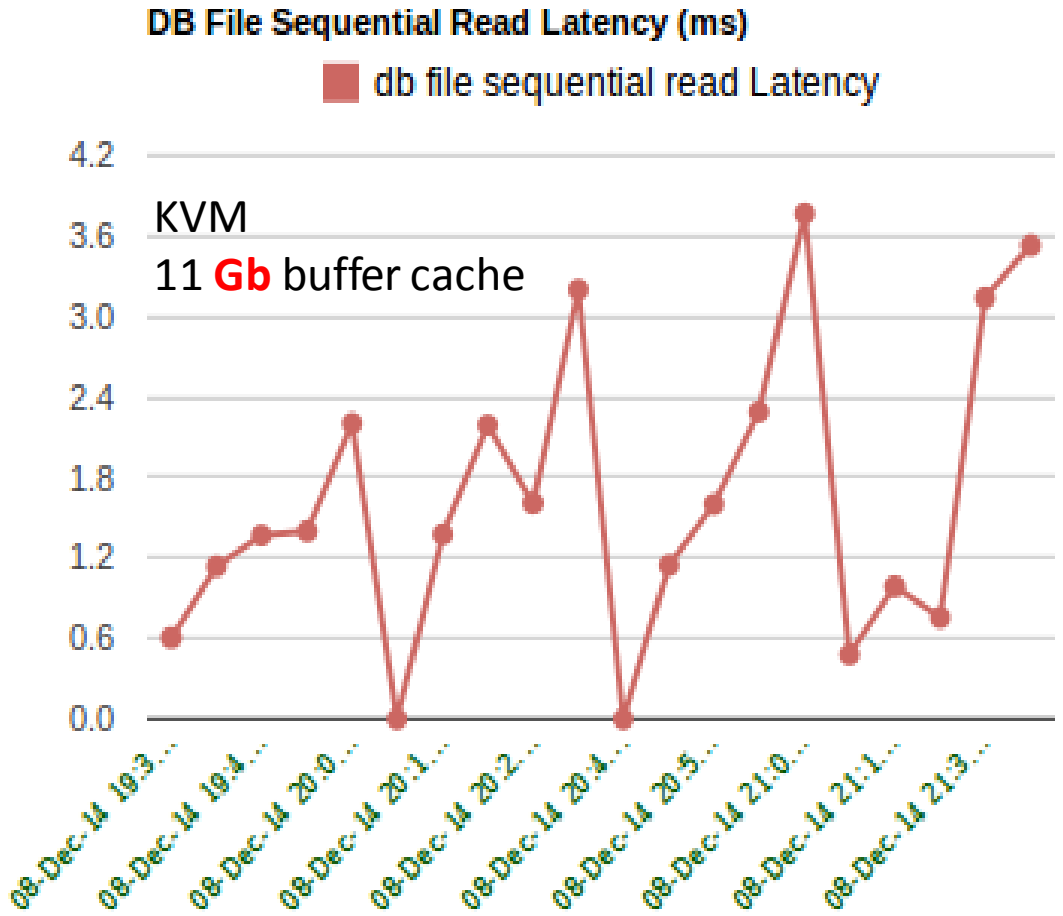
DB READ IOPS PERFORMANCE (LXC VS KVM)



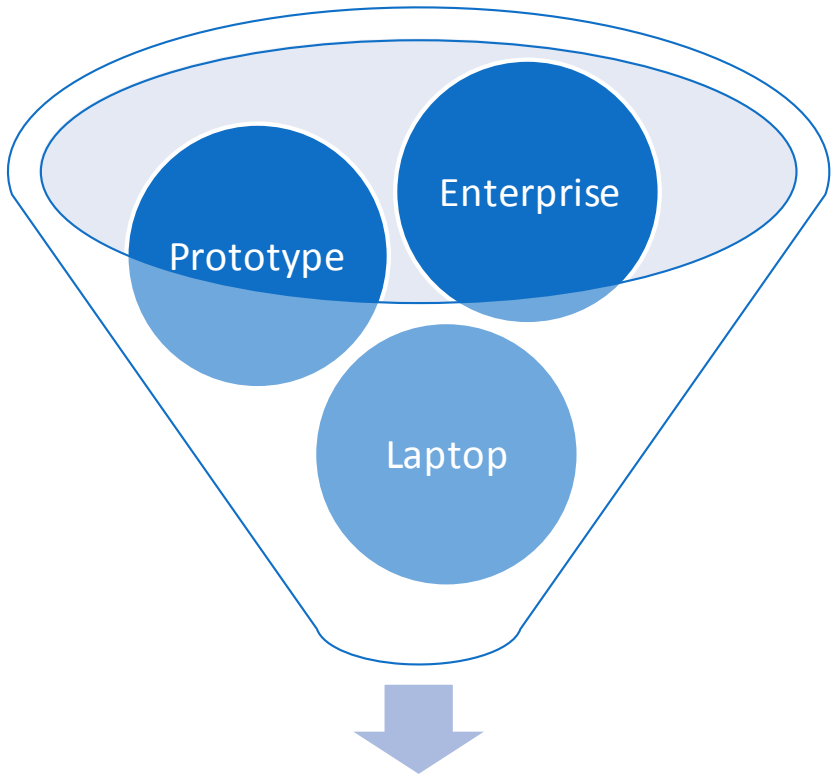


DB SEQUENTIAL READ LATENCY (LXC VS KVM)





THE LINUX CONTAINER CONVERGENCE PHENOMENON



Linux Containers

- Linux Containers are ideal for enterprise, because they fully leverage the power/expense of high-speed computing equipment
- Linux Containers, and the necessary soft infrastructure of DNS, DHCP, and OpenvSwitch can run on your desktop or laptop in exactly the same way they will run in the enterprise. Also, since desktop is a performance-limited environment, Linux Containers are ideal there too.

Container Advantages

Container-Based, Application-Centric
Server and Storage Virtualization Software

- *Consolidate* Databases, Big Data Apps on Bare Metal to avoid Hypervisor Performance Overhead
- *Reduce* VM sprawl, software licensing costs by deploying multiple applications per machine
- *Deploy applications 10x faster*, make applications portable
- *Optimize* data capacity & performance with application-driven storage management

info@robinsystems.com

ROBIN

Did you know that…

Actually, HoHo's *are* Better!

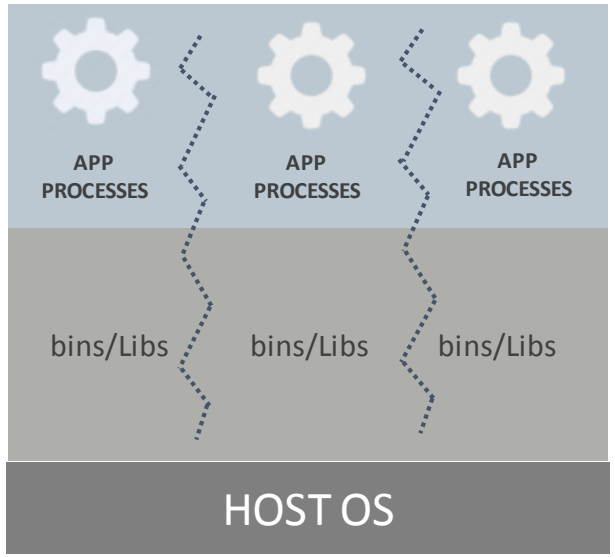
WHAT ARE CONTAINERS ?

info@robinsystems.com

ROBIN

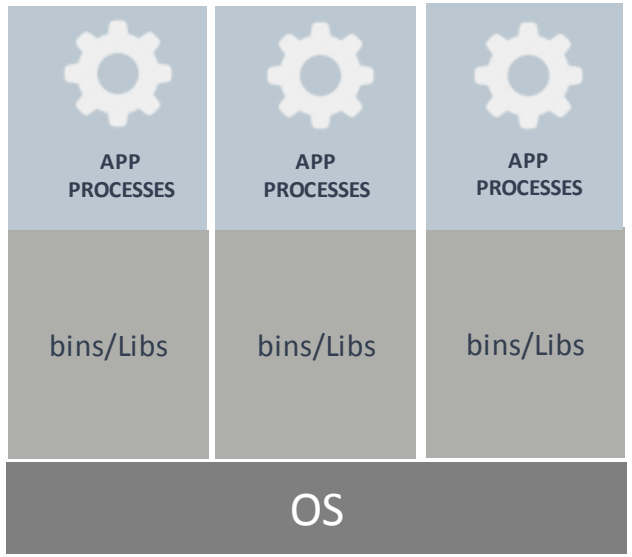
Deployment choices

BARE METAL



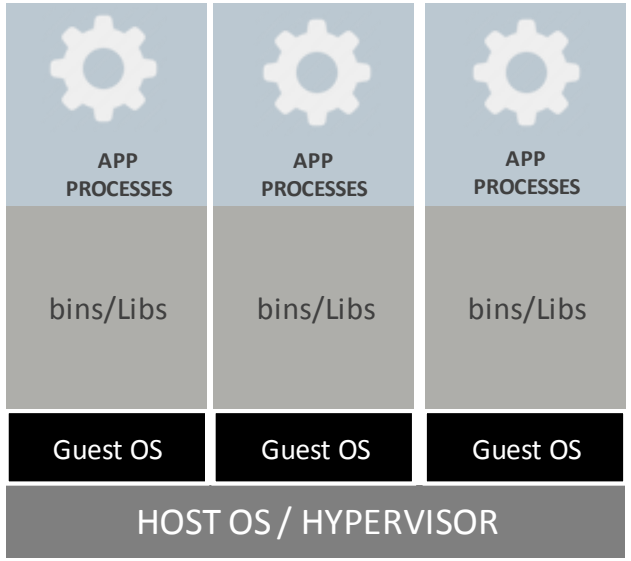
- No Isolation
- No Performance overhead
- Not Portable

CONTAINERS



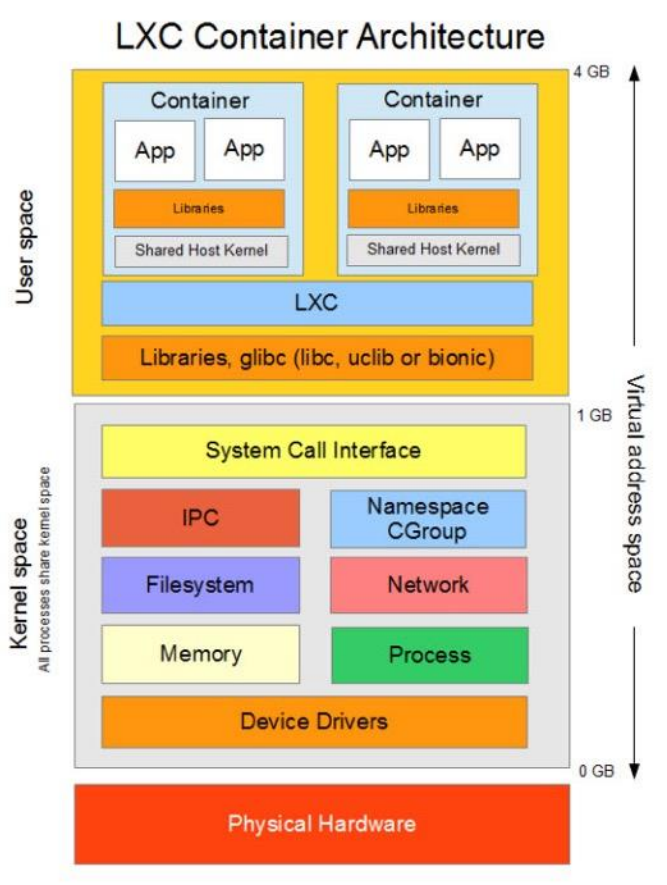
- Run Time Isolation
- No Performance overhead
- Boots from init / Portable
- No virtualized hardware

VIRTUAL MACHINE



- Full Isolation
- Performance overhead
- Partially Portable

What is a Linux Container ?



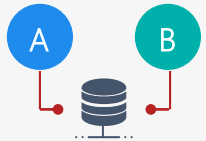
Linux Containers (LXC) is an operating system-level virtualization method for running multiple isolated Linux systems (containers) on a single control host (LXC host).

- Not a virtual machine
- Provides a virtual environment
- Own CPU, memory, block I/O, network etc.

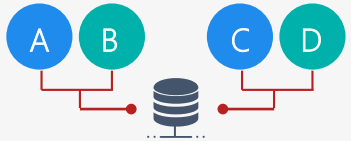
info@robinsystems.com

Containers & Data Apps: What is Missing?

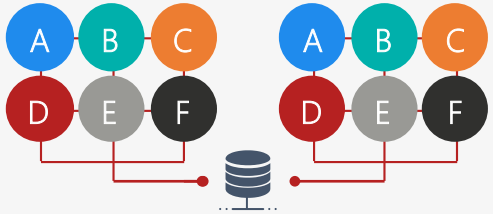
Apps on Bare Metal
(1995-2005)



Apps on Virtualized HW
(2005-2015)



Apps Containerized
(2015-2025)



Need 10-100x more performance/scale from Storage

- The whole point is to pack 10-100x more containers than VMs on your hardware
- Milliseconds response time to start/stop containers vs. minutes for VMs
- Each app in each container can bind to multiple data volumes – HDD, SSD, Flash
- Traditional storage systems can't handle this scale, deployment agility or diversity

Need End-to-End performance isolation and QOS control

- Containers only provide CPU and Memory isolation – what about IO and Storage?

Need to extend application lifecycle management simplicity to data

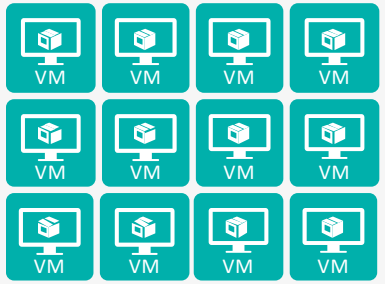
- Quick provisioning, snapshot, cloning, time-travel

info@robinsystems.com

ROBIN

Container Ecosystem

Pre-Container Era



Almost always deployed on VMs

Stateless Application Layer

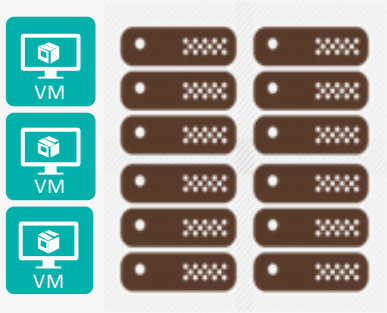
Web Server
Apache, Tomcat, Nginx, IIS

Application Framework
Ruby on Rails, PHP, Java

Container Era

docker

Containerization Platform for Stateless Apps



Mostly Bare Metal

Stateful Application (Data) Tier

Databases
RDBMS, NoSQL

Analytics
Hadoop, Spark

Search
ELK, Solr

In-Memory
Redis, Memcache

?

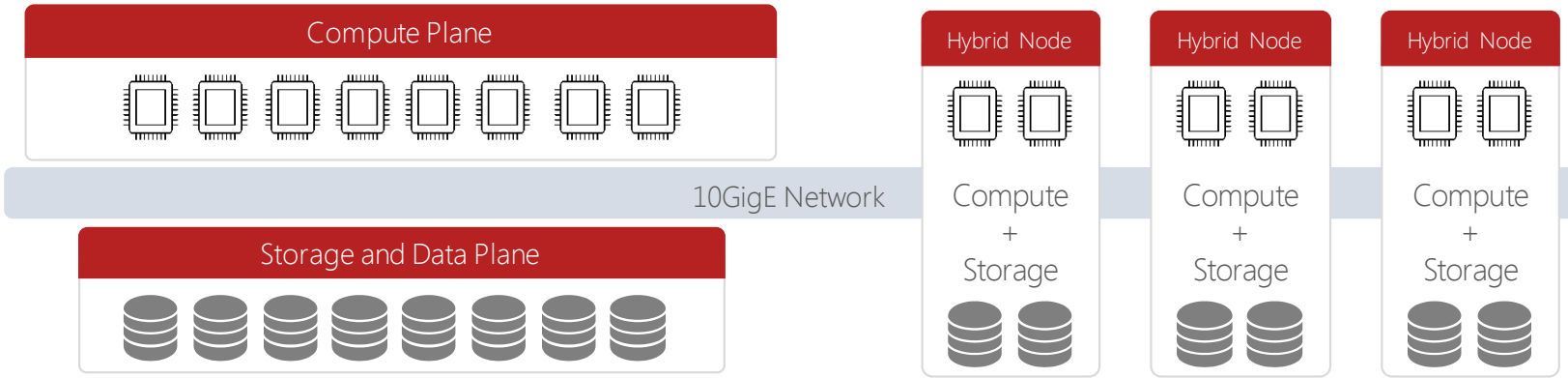
Hardware Deployment Choices

Decoupled Compute and Storage

- Right size hardware for compute and storage layers
- Scale compute and storage independently
- Centralize data protection, security, governance
- Enable data sharing

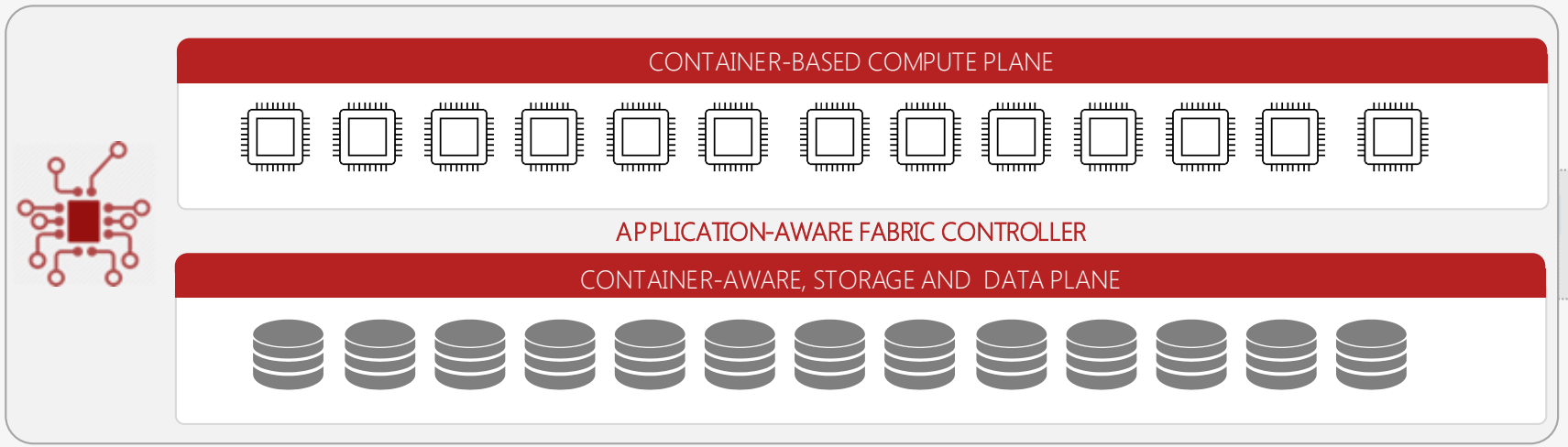
Collocated Compute and Storage

- Hardware management simplicity
- Data locality and affinity to avoid network overload



CONTAINERS RUN ON YOUR OWN COMMODITY HARDWARE

Application-Defined Data Center Software



Application-defined Compute

Cloud-extend
Compute & Storage

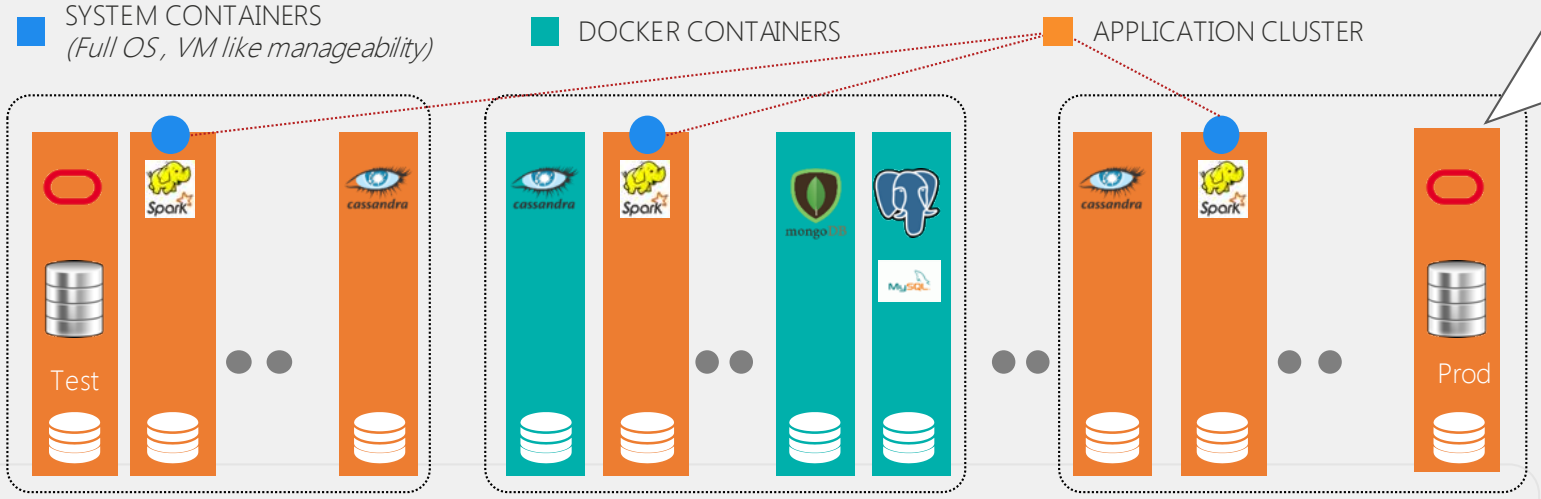
Application-defined Storage

RUNS ON YOUR OWN COMMODITY HARDWARE

info@robinsystems.com

ROBIN

Consolidate Applications



- Pack more applications per machine
- Complete performance isolation
- 2-3x hardware, rack space, power savings
- 100x faster application deployment, scaling
- Seamless portability

CONTAINER-BASED COMPUTE PLANE

Application-defined Compute

APPLICATION-AWARE FABRIC CONTROLLER

Cloud-extend
Compute & Storage

ROBIN CONTAINER-AWARE, STORAGE AND DATA PLANE

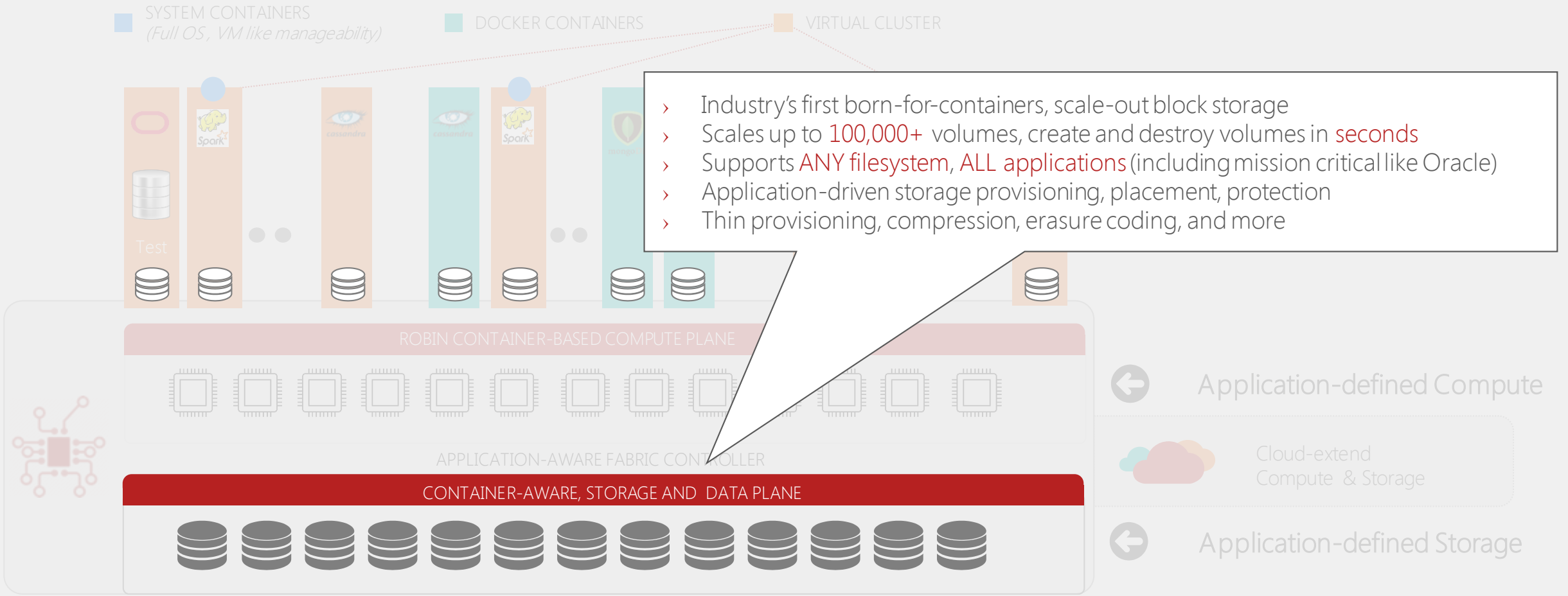
Application-defined Storage

RUNS ON YOUR OWN COMMODITY HARDWARE

info@robinsystems.com

ROBIN

100% Application-Driven, "Invisible" Storage

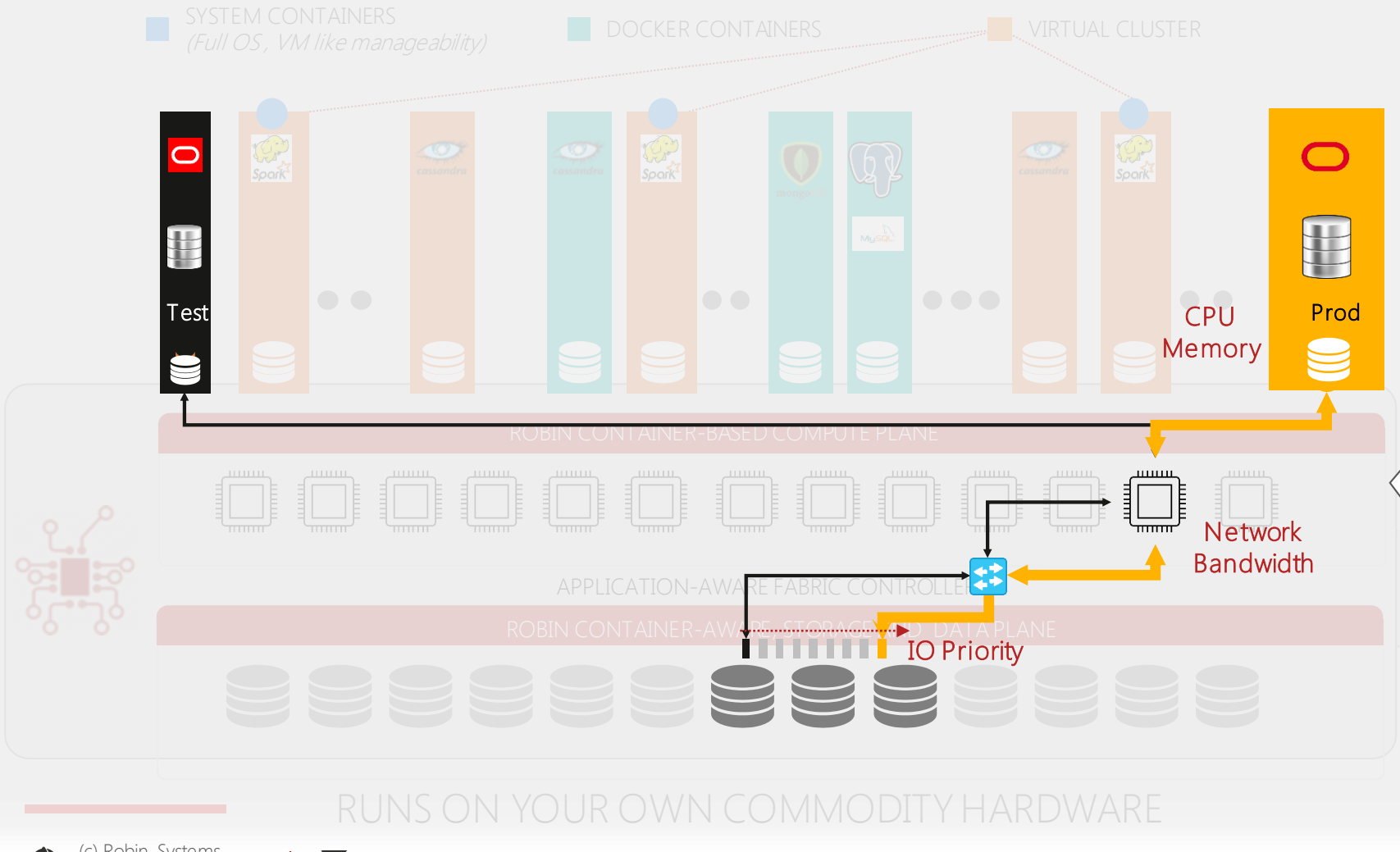


- Industry's first born-for-containers, scale-out block storage
- Scales up to 100,000+ volumes, create and destroy volumes in seconds
- Supports ANY filesystem, ALL applications (including mission critical like Oracle)
- Application-driven storage provisioning, placement, protection
- Thin provisioning, compression, erasure coding, and more

RUNS ON YOUR OWN COMMODITY HARDWARE

info@robinsystems.com

App-to-Spindle QoS Guarantee



- Control resources & enforce priority at each layer – compute, network, storage
- Divide resources among clusters, applications according to business priorities
- Guarantee MAX and MIN IOPS at the container-level

Application-defined Storage

info@robinsystems.com

ROBIN

Ever Wonder…

Toblerone chocolate triangles the kind you see in airports always at the Duty Free store are transported very efficiently by the Swiss on Special Trains…Exactly the same way that LXC Linux Containers also operate with maximum efficiency because everything is at bare metal performance – network, compute, storage.

How the Swiss Transport Toblerone…Efficiently

Toblerone in VMs

Toblerone in LXC

LXC for Oracle

info@robinsystems.com

ROBIN

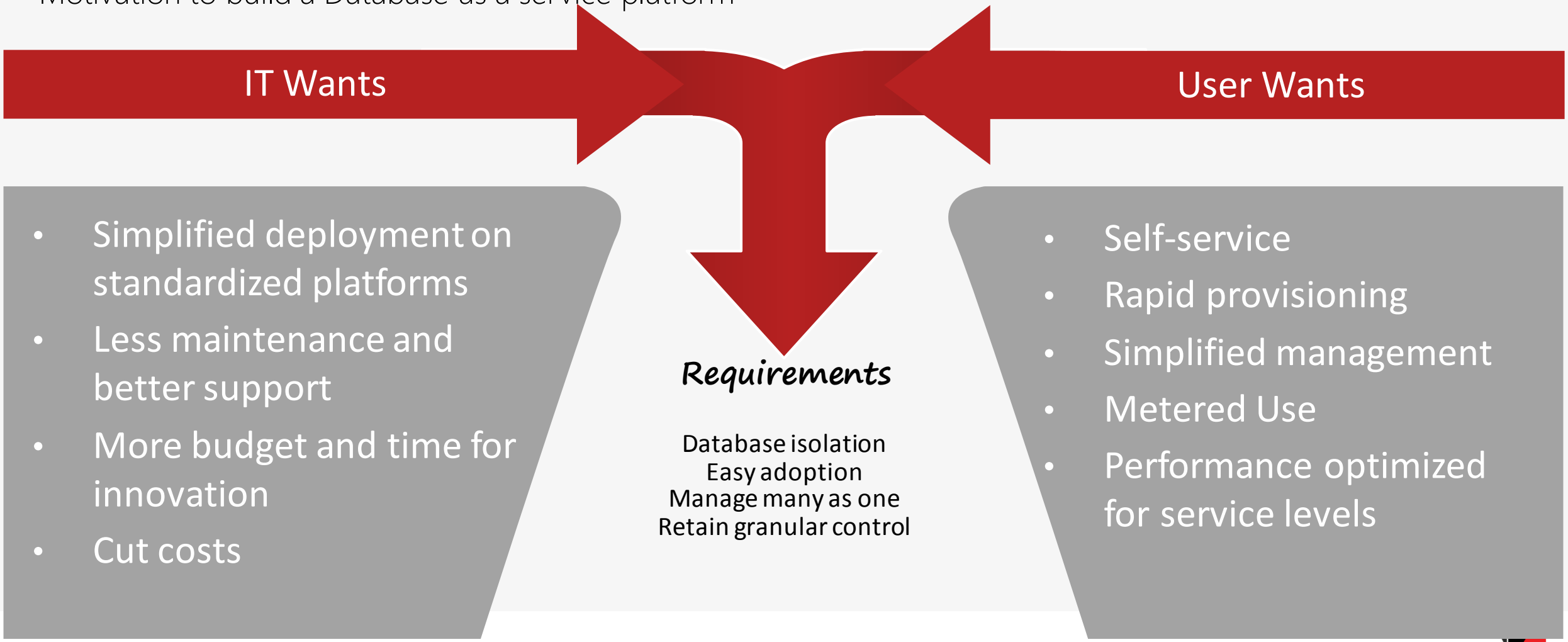
Ease of Provisioning

info@robinsystems.com

ROBIN

Business Drivers

Motivation to build a Database as a service platform

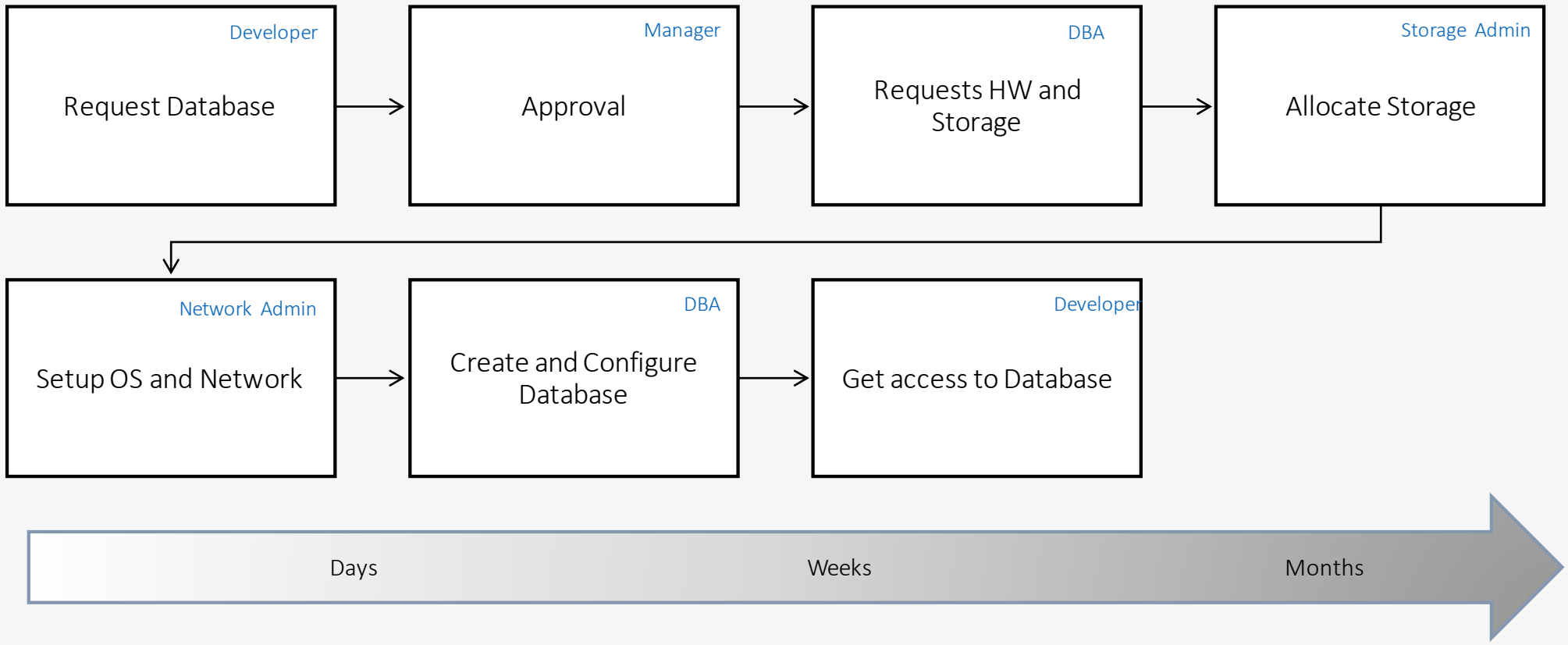


- Simplified deployment on standardized platforms
- Less maintenance and better support
- More budget and time for innovation
- Cut costs

- Self-service
- Rapid provisioning
- Simplified management
- Metered Use
- Performance optimized for service levels

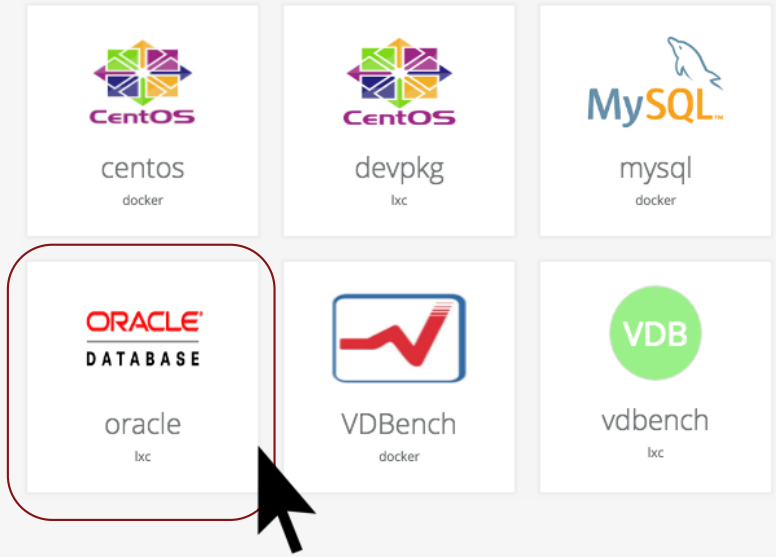
info@robinsystems.com

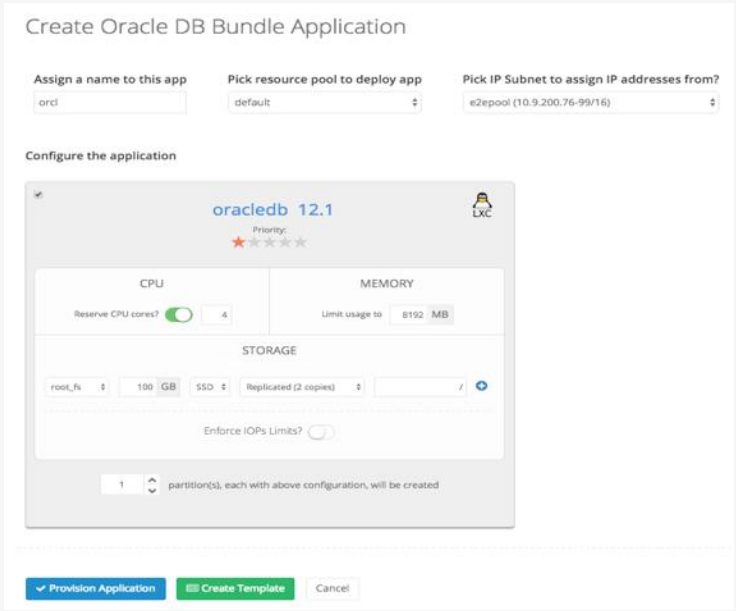
Traditional Deployment

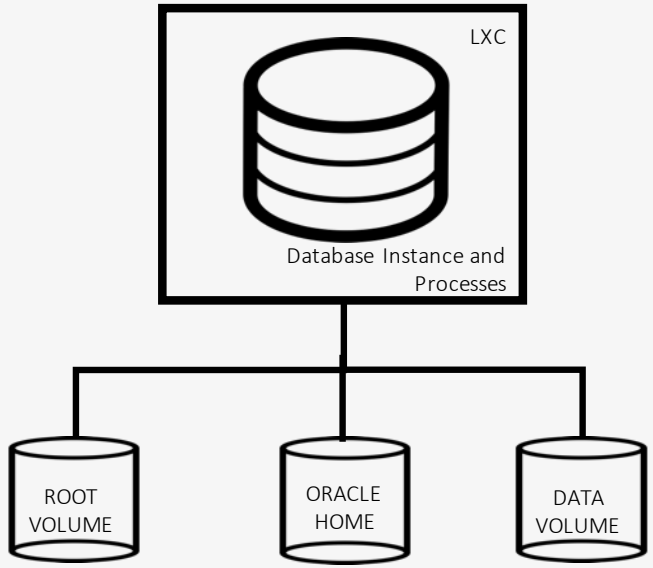


info@robinsystems.com

Containers Simplify Deployment







Select an image from the Software Library

Provide the required inputs

Provision your database

info@robinsystems.com

How does this work internally ? (1)

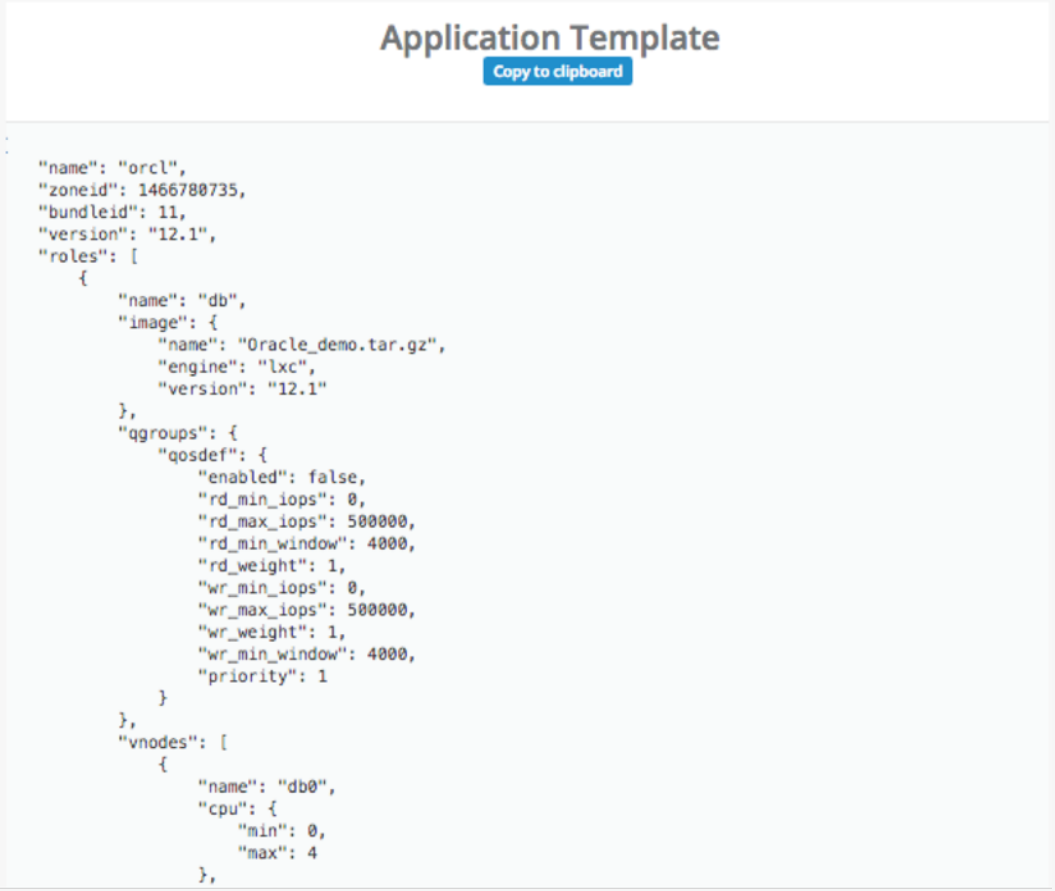
MANIFEST FILE

```
name: Oracle DB Bundle
version: "12.1"
description: My organization's Oracle dev environment
website: http://www.mycompany.com
icon: oracle_logo.gif
roles: [db]

db:
    name: oracledb
    version: "12.1"
    icon: oracle_logo.gif
    multinode: true
    description: Runs oracle database server
    image: Oracle_demo.tar.gz
    engine: lxc
    cpu:
        reserve: false
        cores: 8

hooks:
    setup: "bash setup.sh"
    start: "bash start.sh"
    stop: "bash stop.sh"
```

- Manifest file describes the Oracle image
  - References to a tar ball Oracle image
  - Reserve CPU
  - Define Storage Mounts
- Define Hooks
  - Setup
    - Create database based on user inputs
  - Startup
    - Allows starting up database
    - Perform any operation needed at each startup
  - Stop
    - Stops the database

info@robinsystems.com

How does this work internally ? (2)

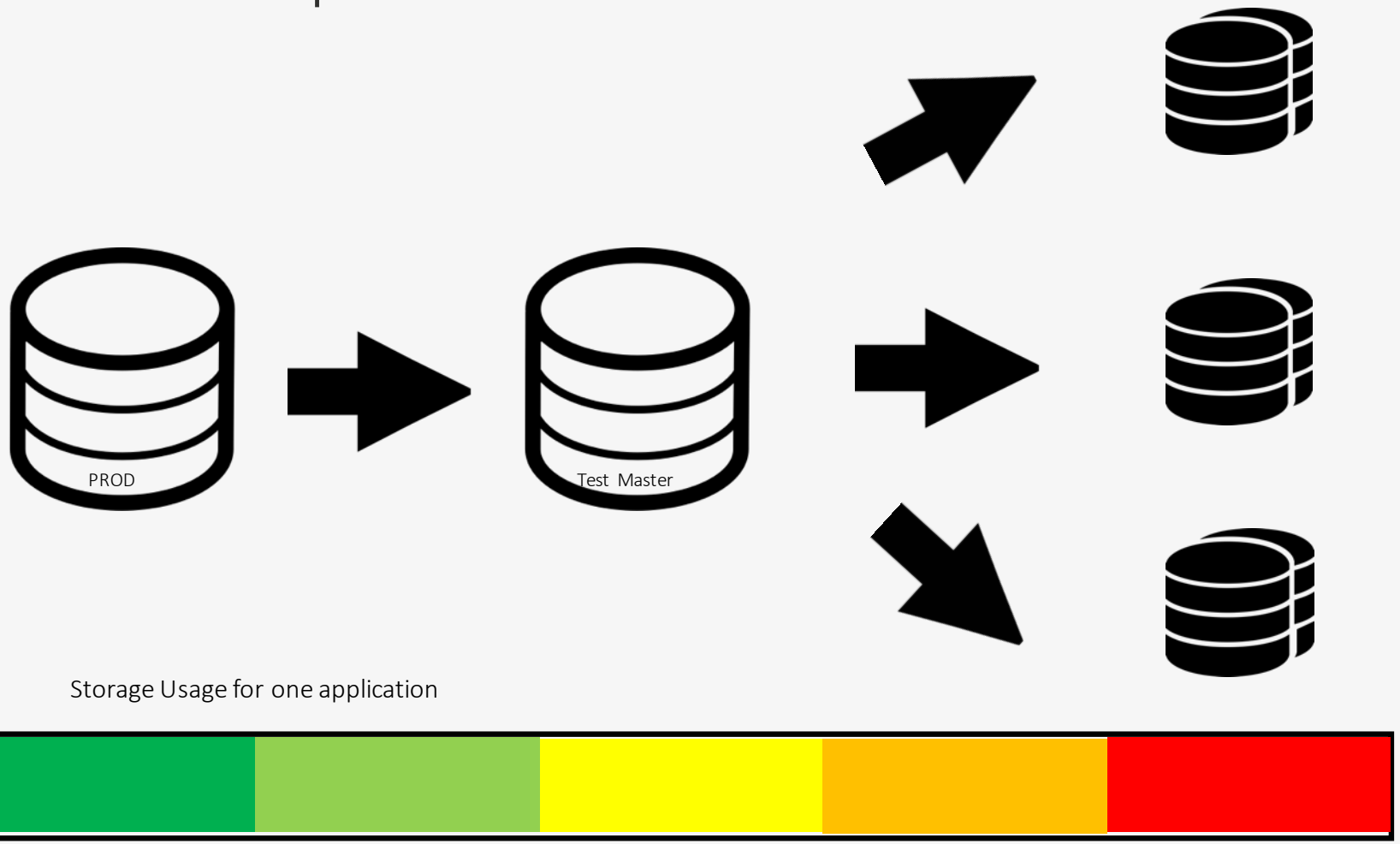


- When user input is provided
  - Template file is generated with all required inputs
  - Template file can be reused to create similar databases from the command line
  - User can always customize by editing this template

Cloning

info@robinsystems.com

Data Sprawl

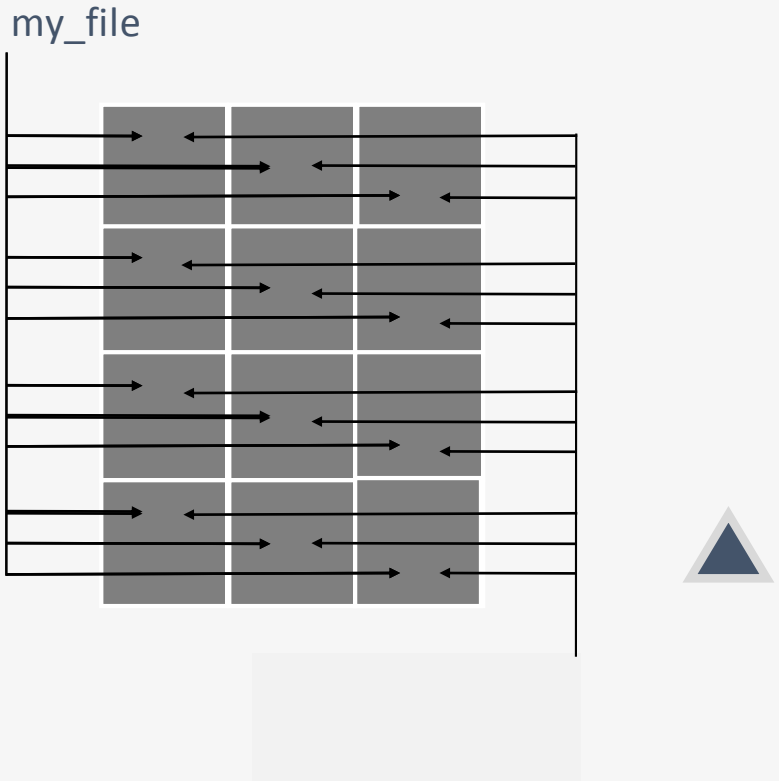


- Full Clones of your production database
  - Time consuming
  - Results in Data Sprawl
  - Create Test master to avoid performance penalty
- What if , you create as many environments but not pay the storage penalty ?

Storage Usage for one application

Copy-on-Write Basics

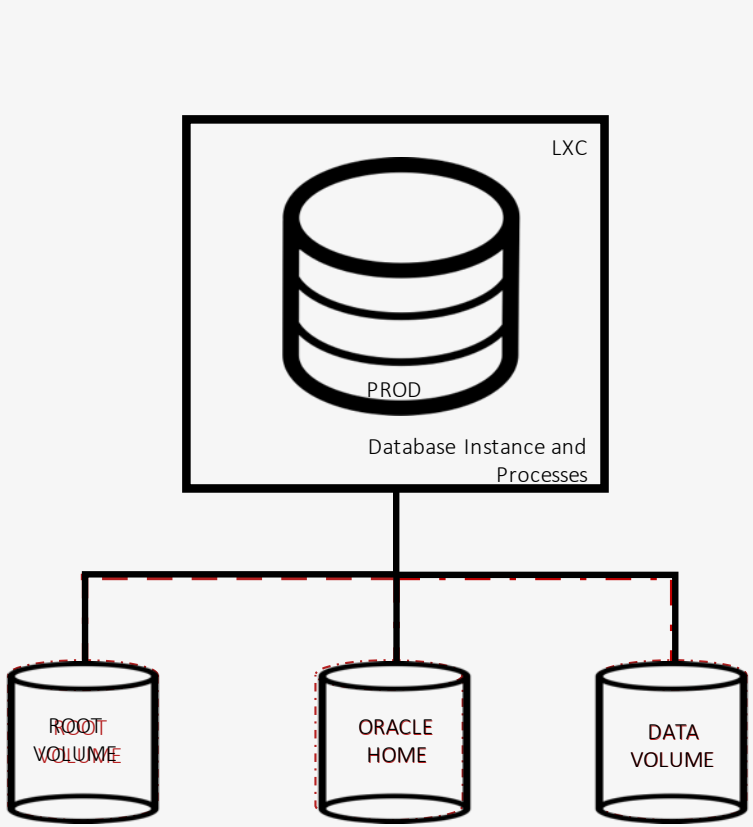
Much more efficient storage of substantially similar files

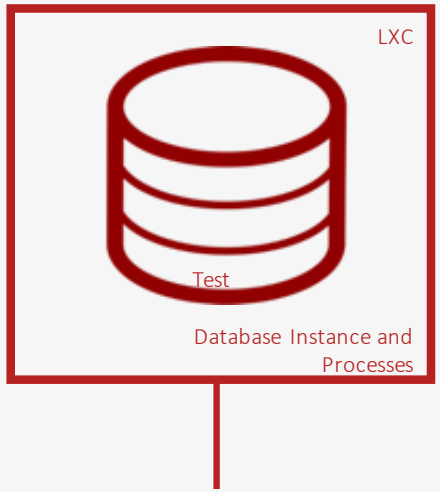


- File is collection of data blocks
- And headers
- Copy file requires only copy of header
  - Much more efficient storage
  - Minimal IO to create copy
- Only copy block when changed

```
cp my_file my_file_copy
```

Thin clone using LXC Containers





Create a clone of the database

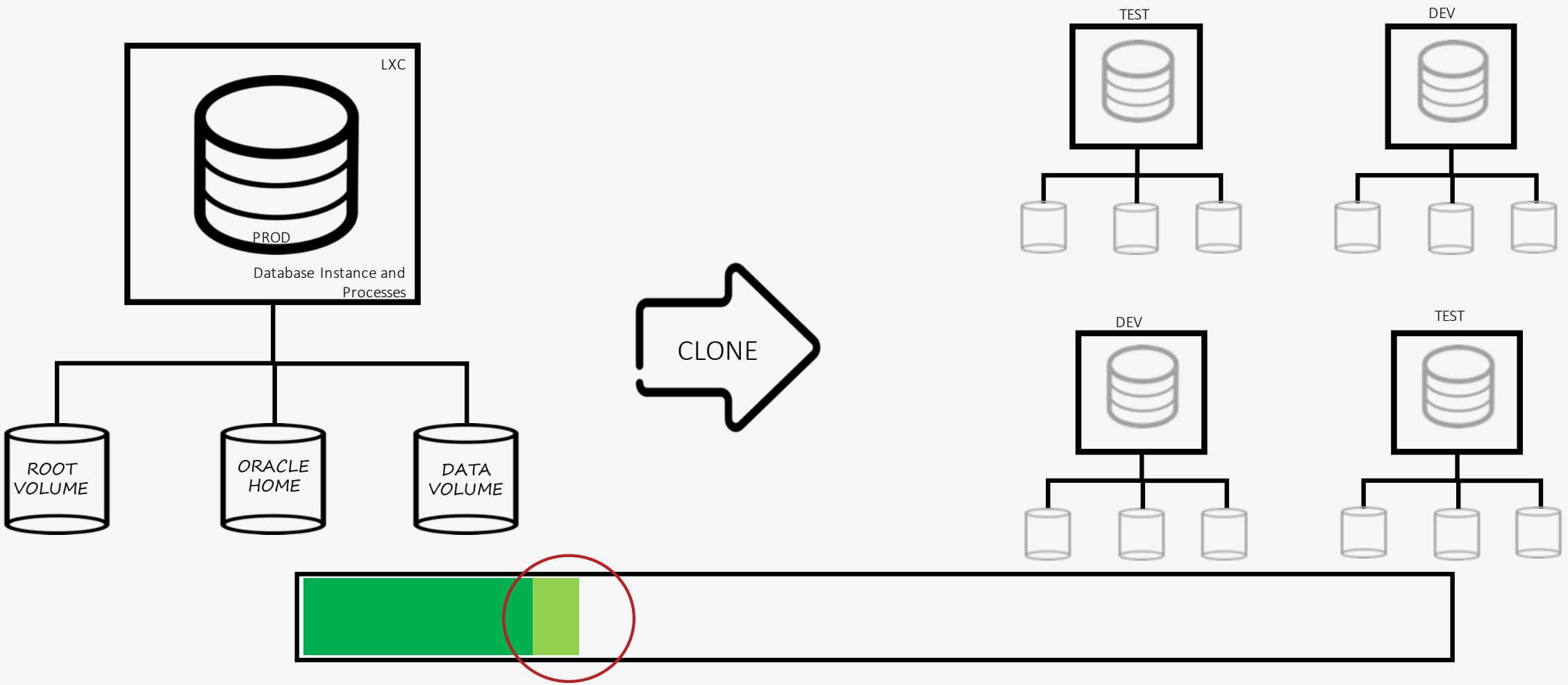
Time consistent copies with minimum space usage

Point in time snapshot of all the volumes

info@robinsystems.com

ROBIN

Multiple Clone copies with fraction of storage



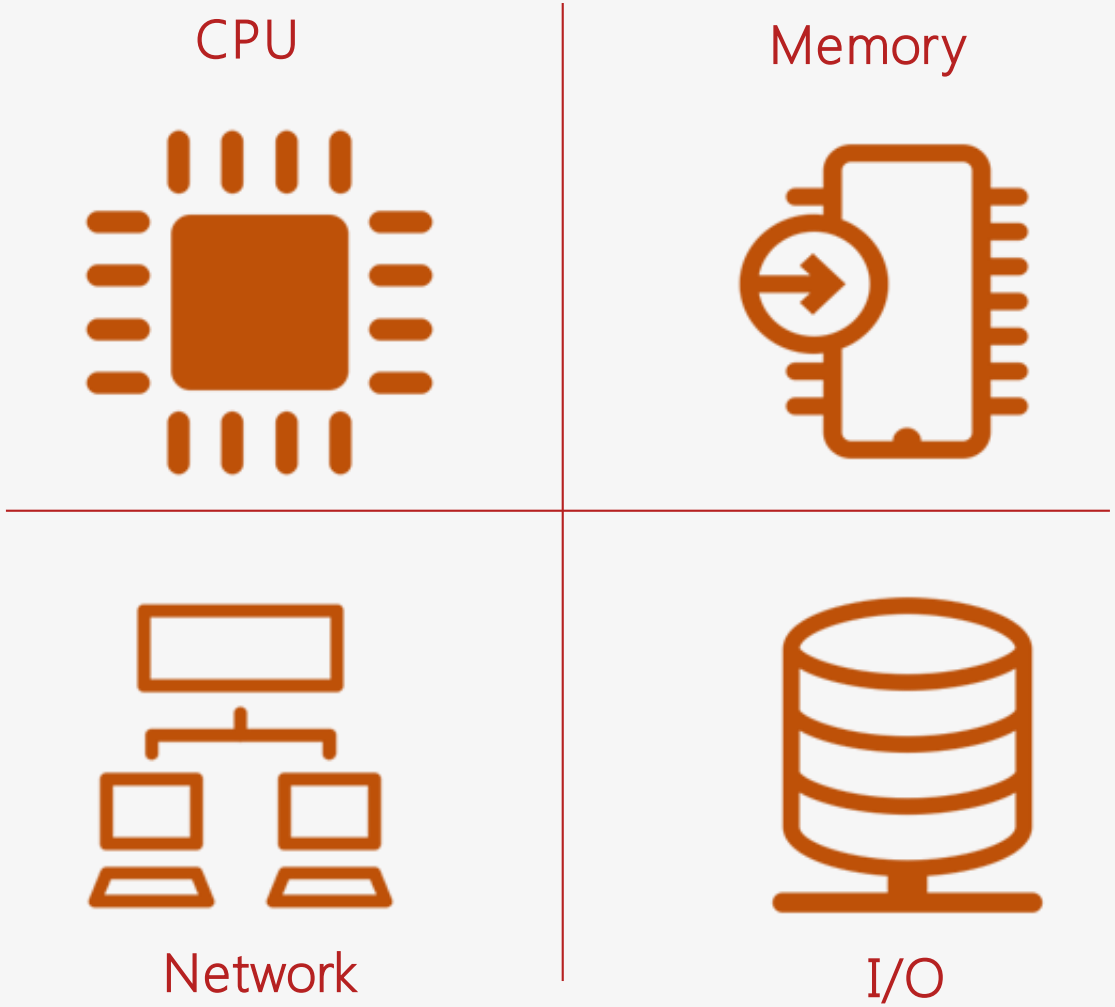
info@robinsystems.com

Resource Management

info@robinsystems.com

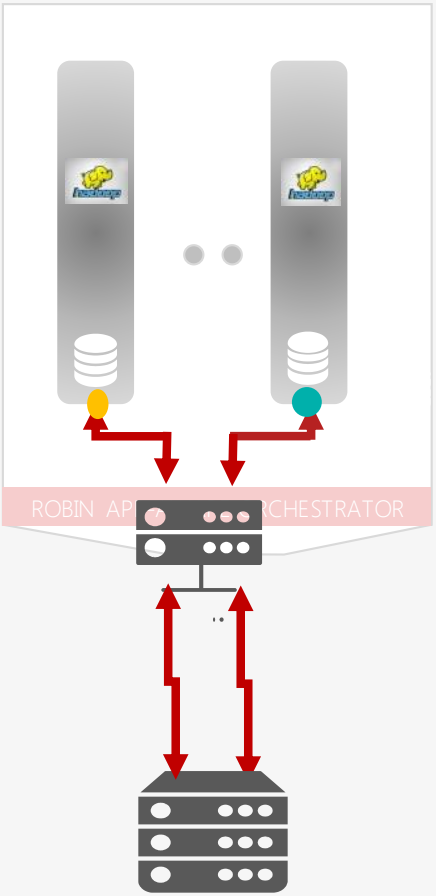
Clustering Challenges

- Noisy neighbor problem



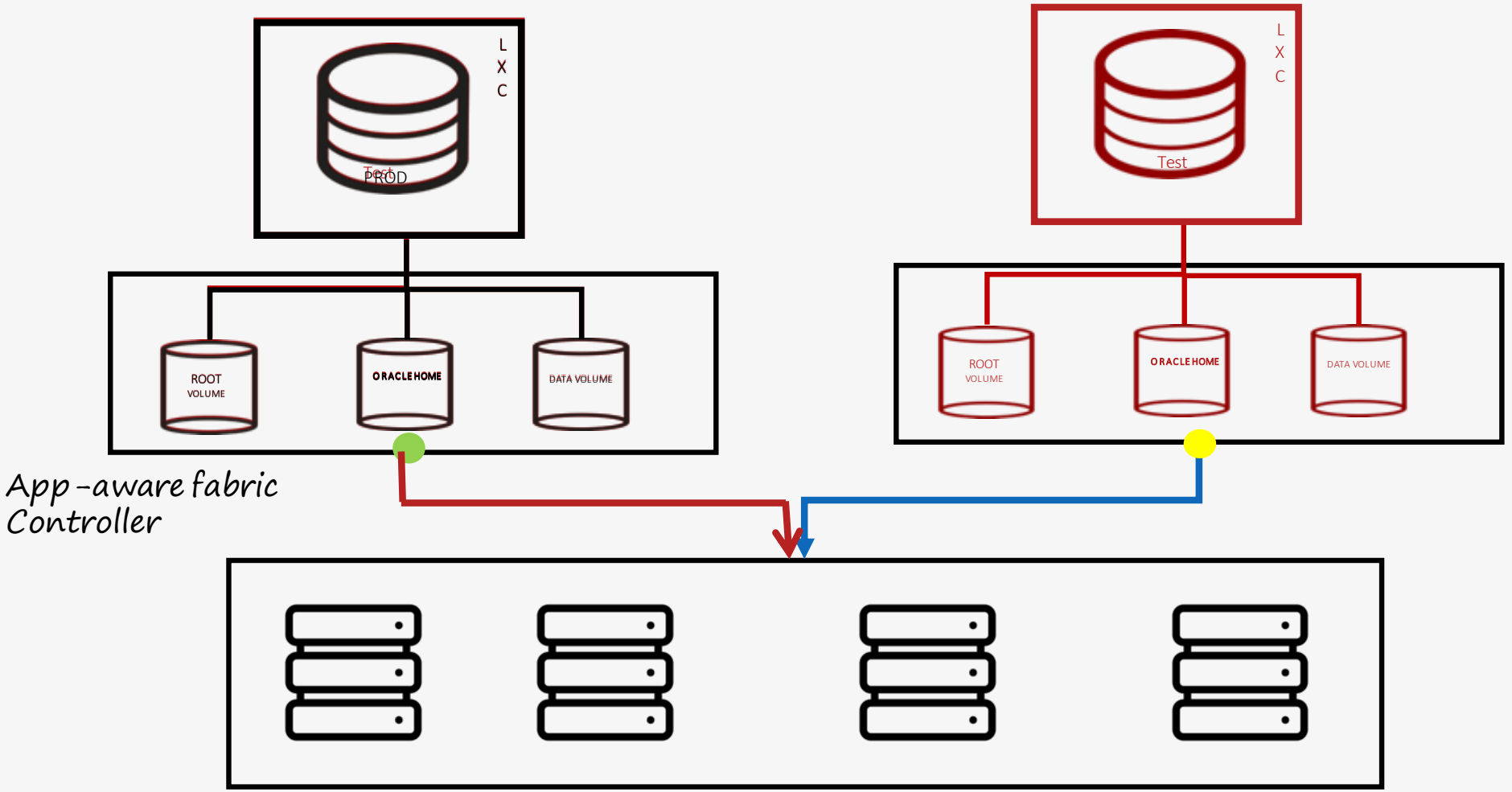
LXC Container QoS

- Resource Pools
  - Partitioning of resources into "Pools" of compute and storage nodes
  - Enables physical isolation for a set of applications
- Resource Reservation
  - Carve out a slice of CPU, memory, IO for mission-critical applications
- Application Shares
  - Ensure best performance for highest priority applications
- Read and Write IO
  - Max IOPS Cap → Prevent a single application from hogging resources
  - Min IOPS Guarantee → Predictable application performance



info@robinsystems.com

Thin clone directly from production



- IO resource management lets you clone directly from Production
- No need to create a test instance
- Application aware fabric controller allocates two separate I/O channels which guarantees I/O for you production environment

info@robinsystems.com

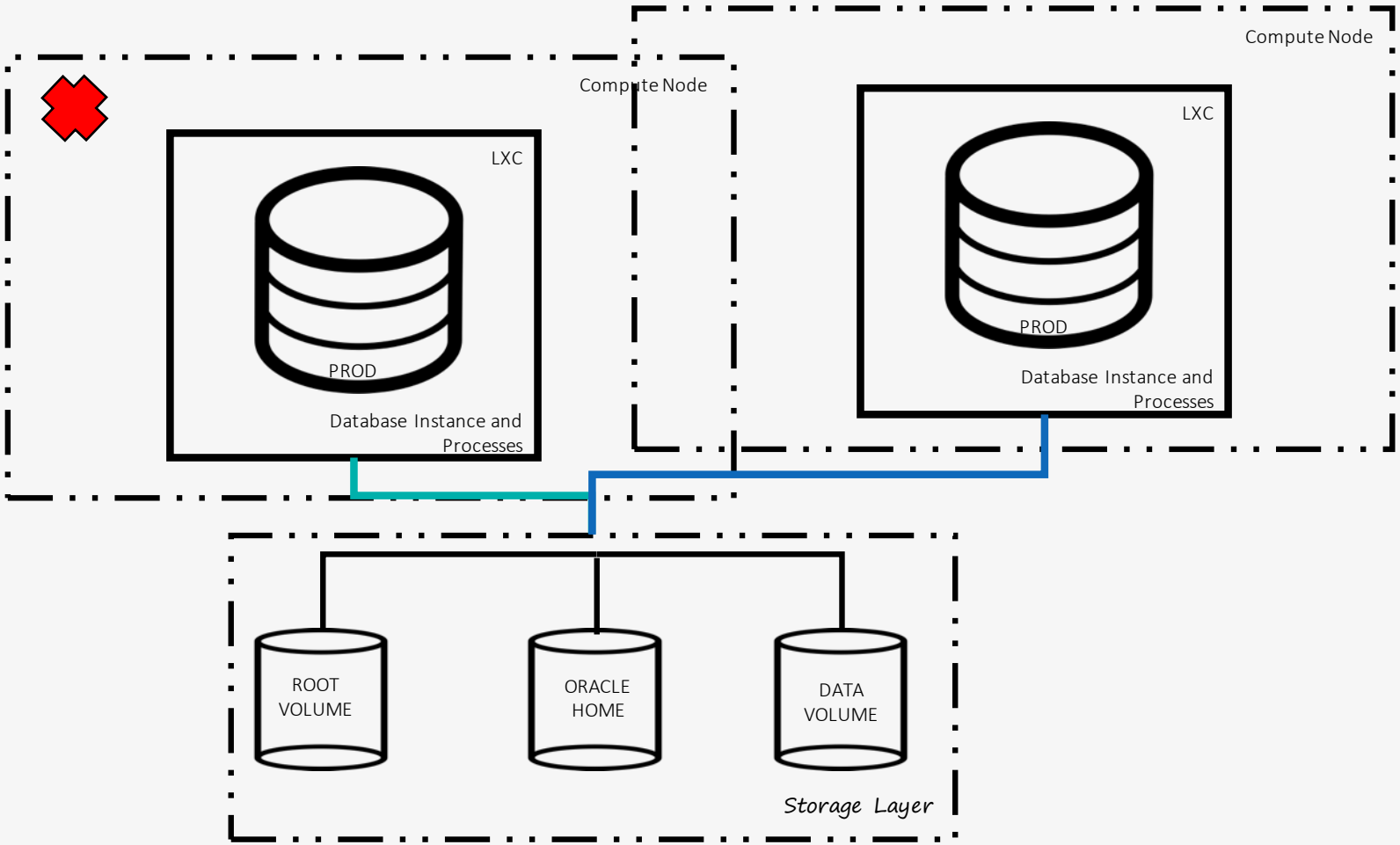
ROBIN

Automated Failover

info@robinsystems.com

ROBIN

Automatic Failover



- Automatically detects container failure (extensions)
- Application monitor brings up an Oracle container image in seconds in another compute node (extensions)
- No need to move storage. Automatically mount of the storage volumes to the new container

info@robinsystems.com

LXC Oracle Support and Certification

Let’s look at the current support picture for running Oracle databases including Oracle 12c RAC on Oracle Grid Infrastructure in LXC Linux containers.

Oracle Certification vs. Support

Oracle Certification

- Oracle Certification means that specific tested combinations of OS and Oracle database software installed in the certified configuration will run properly; patches, updates, etc. specific to that combo are not guaranteed to be provided.

Oracle Support

- Oracle support contracts are purchased from Oracle, and means updates, install patches, and logged issues with Oracle Support can be submitted via the SR process and will be worked and resolved.

Oracle DB 12.1 **Supported** on LXC on UEK

- "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)

Patch Requirements for Oracle DB in LXC

- You must install patch 20920711 prior to enabling RT (real-time processes) for LXC Linux Containers.
- This patch is required because the container does not reboot upon cssd failure, but rather the cssd failure will cause the physical host of the LXC containers to itself reboot.

Enable Real-Time Processes for Linux Containers

- Use the --privileged=RT" switch when creating an LXC Linux Container for Oracle Grid Infrastructure 12c.
- lxc-create -n node_name -t oracle -B [btrfs|ext4] -- --release=6.latest --privileged=rt
- Note that both ext4 and btrfs filesystems are supported.
- Generally btrfs is considered to be much better suited to LXC
- However ext4 can also be used.
- https://twitter.com/BBCLondonNews/status/773502781671997440/photo/1

Filesystem Choices – BTRFS is Better

- The typical choices are BTRFS or ext4
- I have personally used ext4 in all my work to-date
- The choice of ext4 is not necessarily the best choice.
- BTRFS is generally regarded as a better choice for LXC containers
- BTRFS automatically creates LXC containers in subvolumes
- In an ext4 or xfs file system an lxc container clone would take the same amount of space as the original container.
- In a BTRFS file system a subvolume clone does not take up an extra space apart from the differences between the 2 containers

More on BTRFS Filesystem…

- Generally BTRFS allows live snapshots of running containers.
- However database workloads cannot be snapshotted in this way.

Hard Limit Value Error in the Prerequisite Screen During the Installation

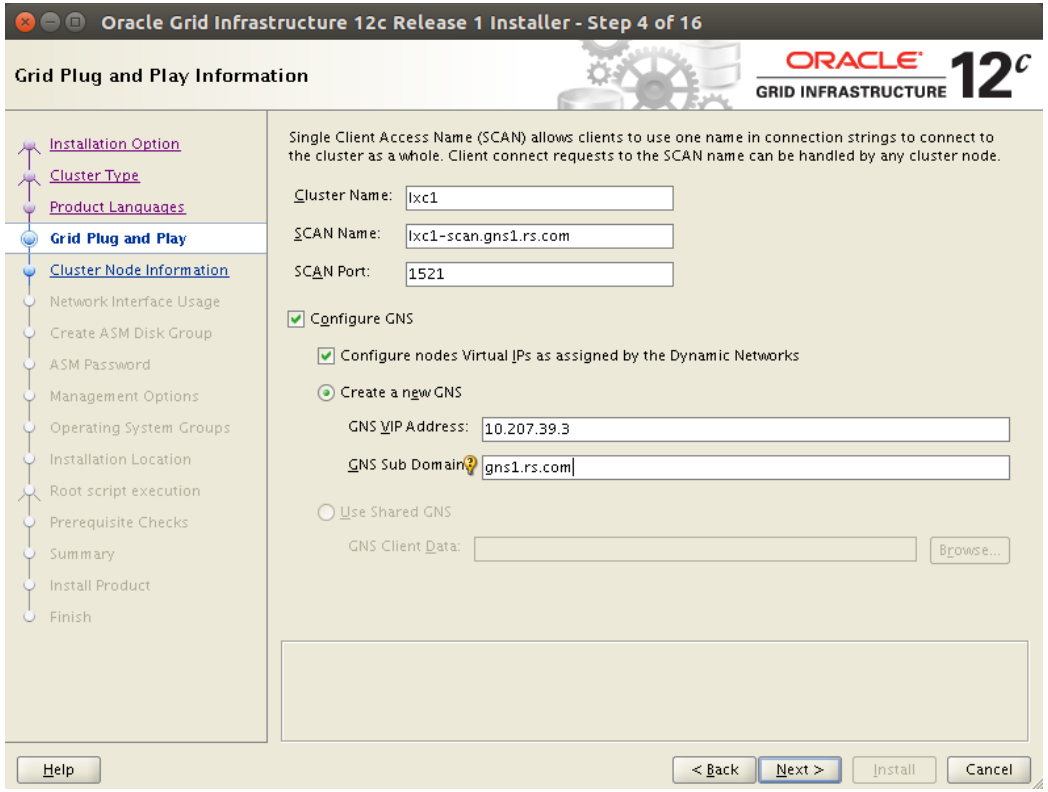
- Hard Limit Value Error in the Prerequisite Screen During the Installation
- "You must not fix the hard limit value using the Fix and Check Again button. Instead, manually add the hard limit value 65536 to the entry etc/security/limits.conf on the host, log out and log in again, and then proceed with the installation. This may help to overcome any issue when the user tries to connect to the Linux container."
- This issue is tracked with Oracle bug 20683209.
- OK so all of the above is what the Oracle Database Release Notes for Oracle 12c DB Say…What Does it Boil Down to Really?……

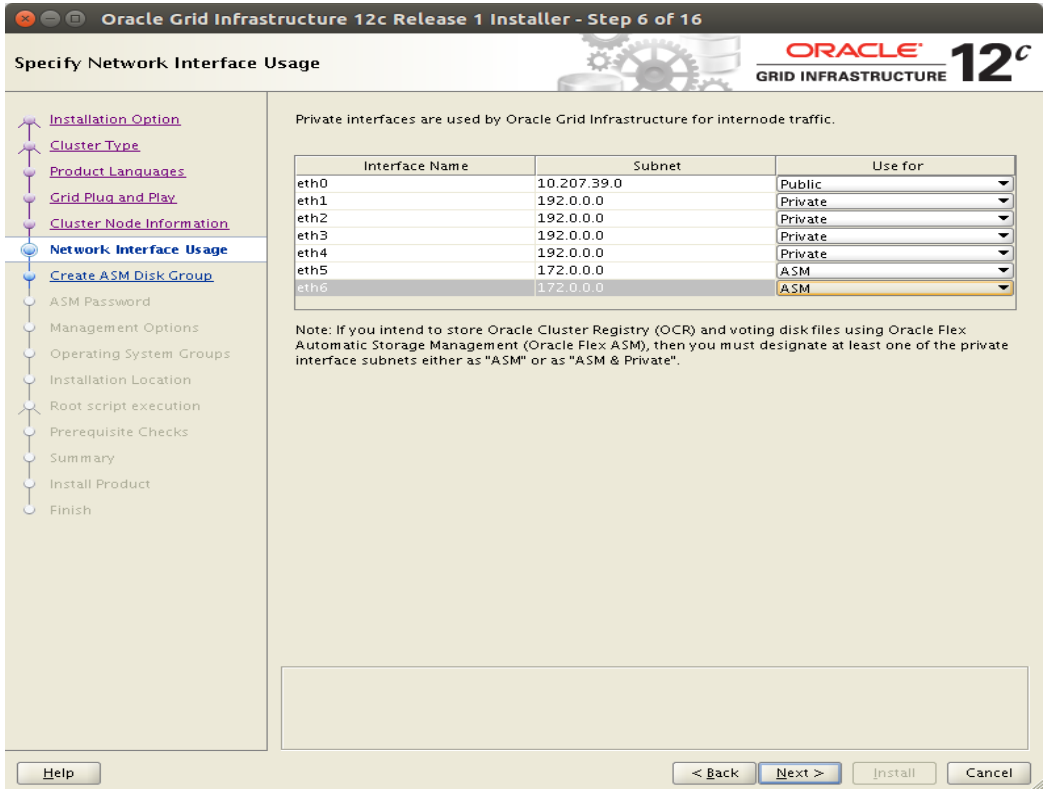
Example limits.conf

```
# Oracle

  grid     soft   nofile    1024
  grid     hard   nofile   65536
  grid     soft   nproc    16384
  grid     hard   nproc    16384
# grid     soft   stack    10240
# grid     hard   stack    32768

  oracle   soft   nofile    1024
  oracle   hard   nofile   65536
  oracle   soft   nproc    16384
  oracle   hard   nproc    16384
# oracle   soft   stack    10240
# oracle   hard   stack    32768
```

- You may find that some settings in the /etc/security/limits.conf file will suddenly cause it to be impossible to login as that user (e.g. as the linux “oracle” and/or “grid” user). If you run into this problem where the “oracle” or “grid” user cannot login to the container (nor “su –” from root) limits.conf settings are usually the culprit.

Notable GI installer screens – 12c RAC on LXC





Practical guide to setting up GNS for LXC

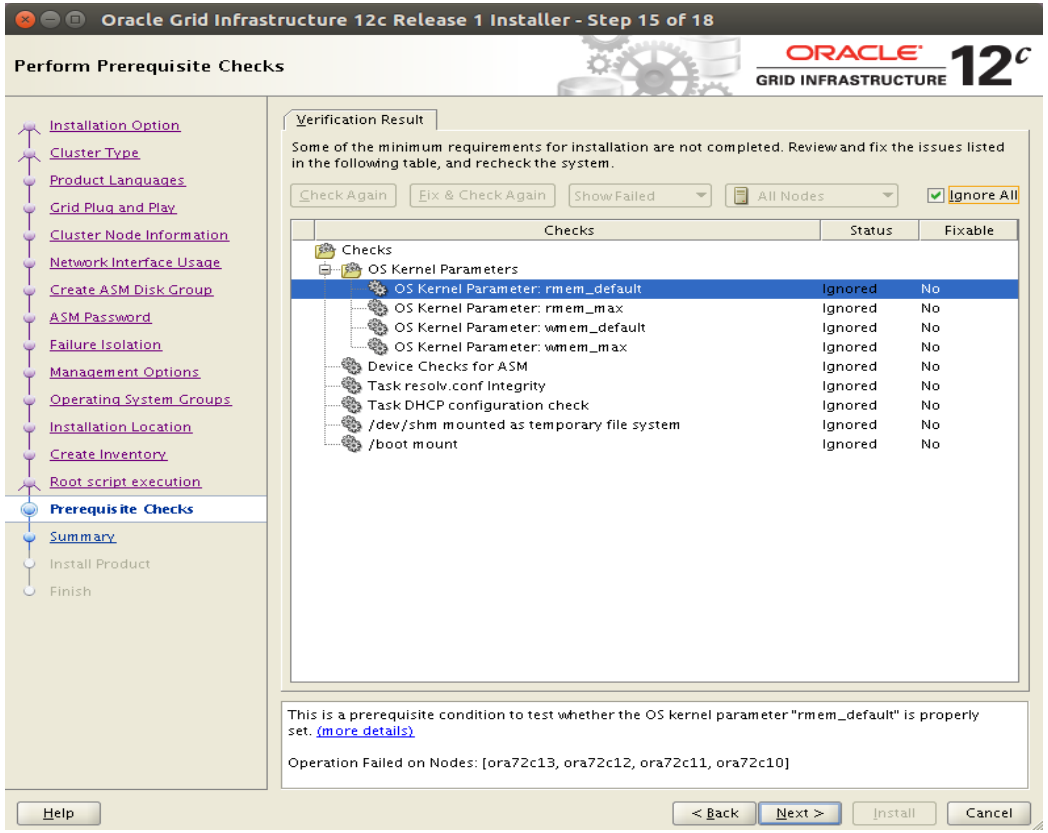
GNS forward ZONE FILE ENTRIES

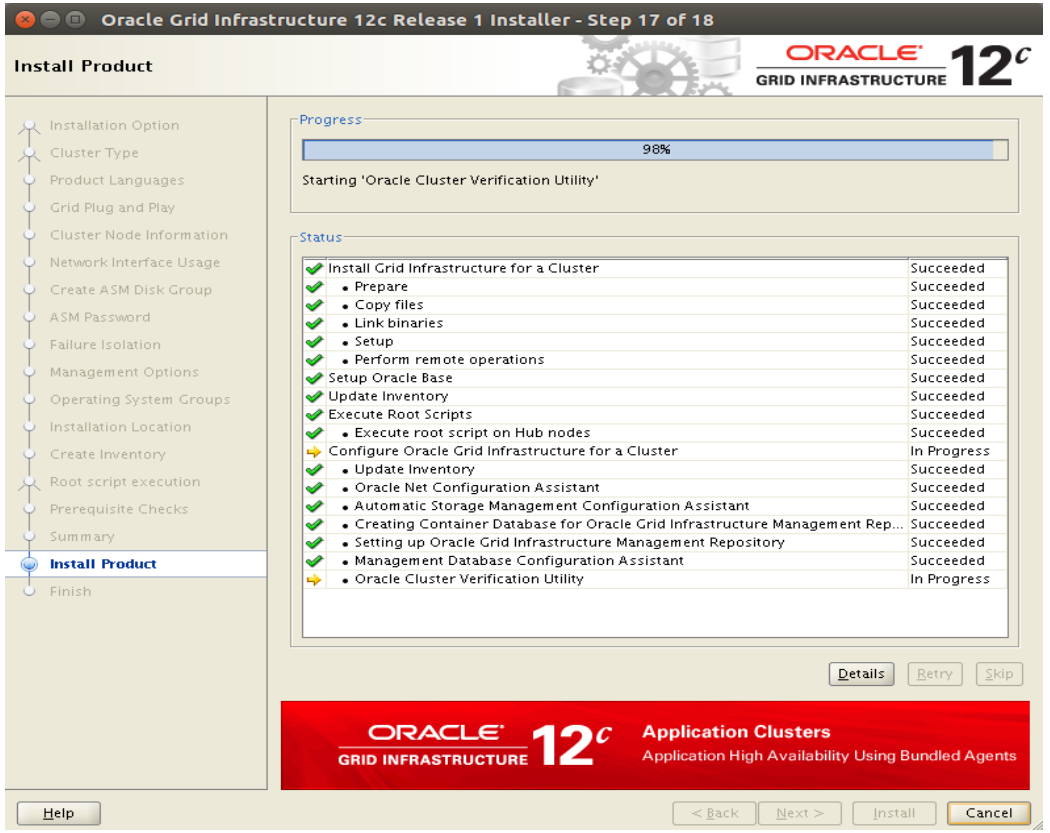
```
$ORIGIN .
$TTL 86400      ; 1 day
rs.com             IN SOA    stlns01.rs.com. postmaster.rs.com. (
                  1512181475 ; serial
                  60         ; refresh (1 minute)
                  1800       ; retry (30 minutes)
                  604800     ; expire (1 week)
                  86400      ; minimum (1 day)
                  )
            NS    stlns01.rs.com.
$ORIGIN rs.com.
_sflow._udp         TXT    "txtvers=1" "polling=20" "sampling=512"
            SRV    0 0 6343 stlns01
$TTL 3600     ; 1 hour
centos-72a             A    10.207.39.14
            TXT    "00abb790500dc92de5c1914108e81e721c"
DESKTOP-LNGFATI        A    10.207.39.22
            TXT    "3121c1dc8a886e646035acf23ae64c1e06"
$TTL 86400     ; 1 day
gns1                  NS   lxc1-gns-vip
lxc1-gns-vip          A    10.207.39.3
$TTL 3600     ; 1 hour
ora72c10              A    10.207.39.10
            TXT    "003166812569484256b4222c5d89c1647f"
$TTL 86400     ; 1 day
stlns01               A    10.207.39.1
vnode-39-100          A    10.207.39.100
```

Gns reverse zone file entires

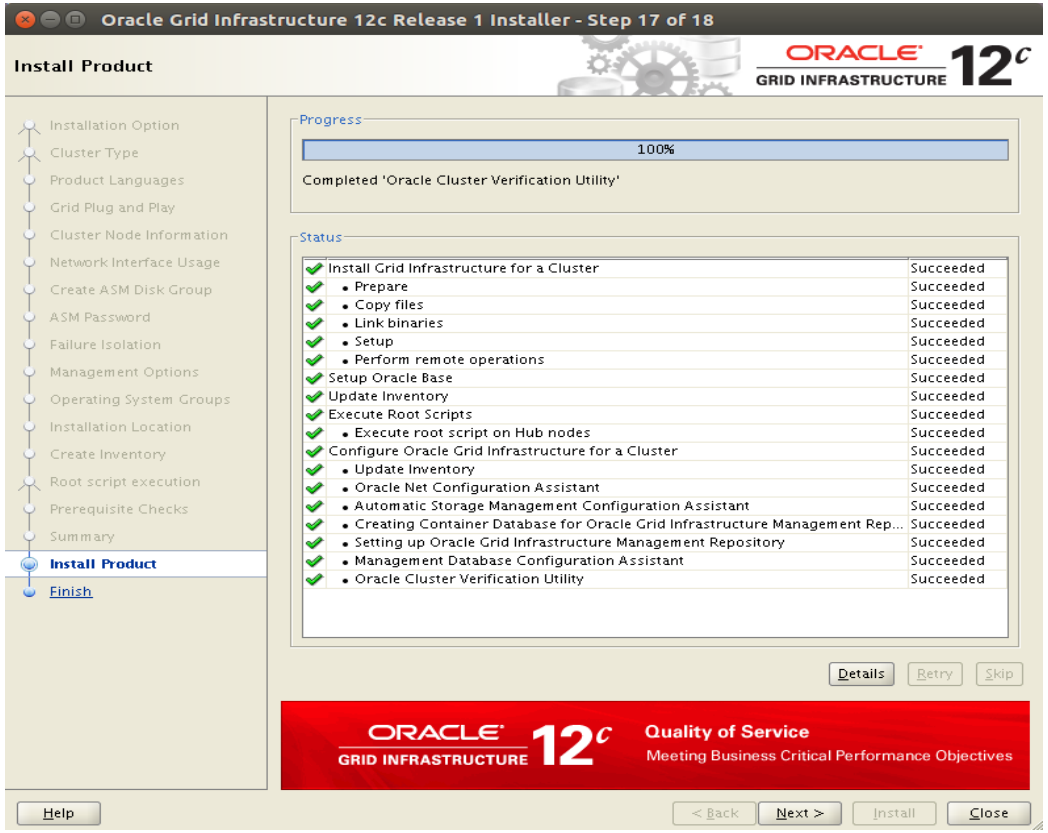
```
$ORIGIN .
$TTL 86400      ; 1 day
39.207.10.in-addr.arpa     IN SOA    stlns01.rs.com.
postmaster.rs.com. (
                  1512181475 ; serial
                  60         ; refresh (1 minute)
                  1800       ; retry (30 minutes)
                  604800     ; expire (1 week)
                  86400      ; minimum (1 day)
                  )
             NS    stlns01.rs.com.
$ORIGIN 39.207.10.in-addr.arpa.
1             PTR    stlns01.rs.com.
$TTL 3600     ; 1 hour
10            PTR    ora72c10.rs.com.
100           PTR    vnode-39-100.rs.com.
11            PTR    ora72c11.rs.com.
12            PTR    ora72c12.rs.com.
13            PTR    ora72c13.rs.com.
14            PTR    centos-72a.rs.com.
22            PTR    DESKTOP-LNGFATI.rs.com.
$TTL 86400    ; 1 day
3             PTR    lxc1-gns-vip.rs.com.
39.207.10.in-addr.arpa     NS    stlns01.rs.com.
stlns01             A    10.207.39.1
```

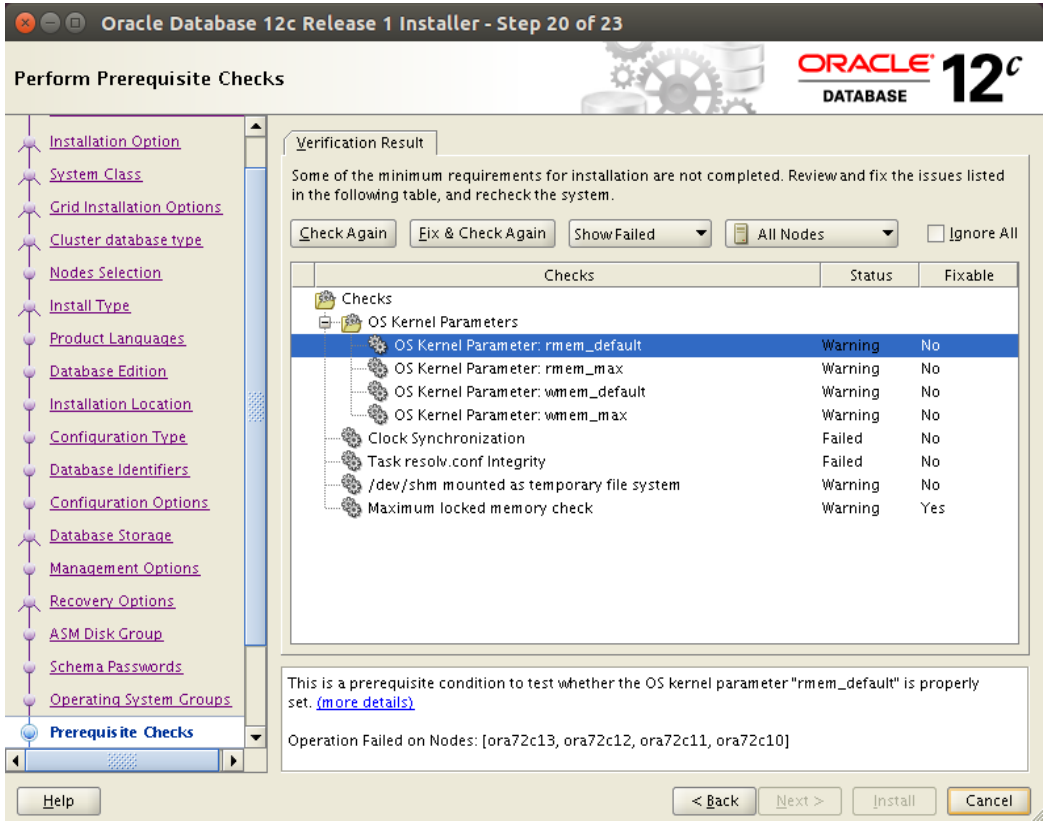
Notable GI installer screens – 12c rac on LXC



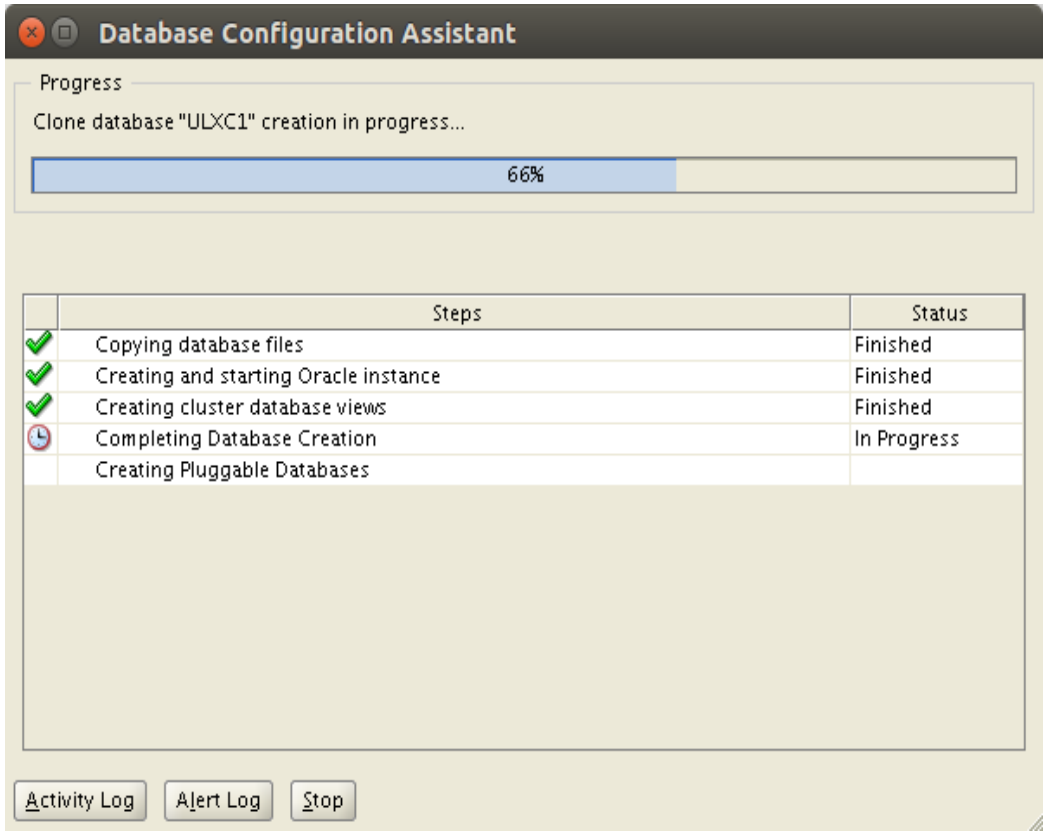


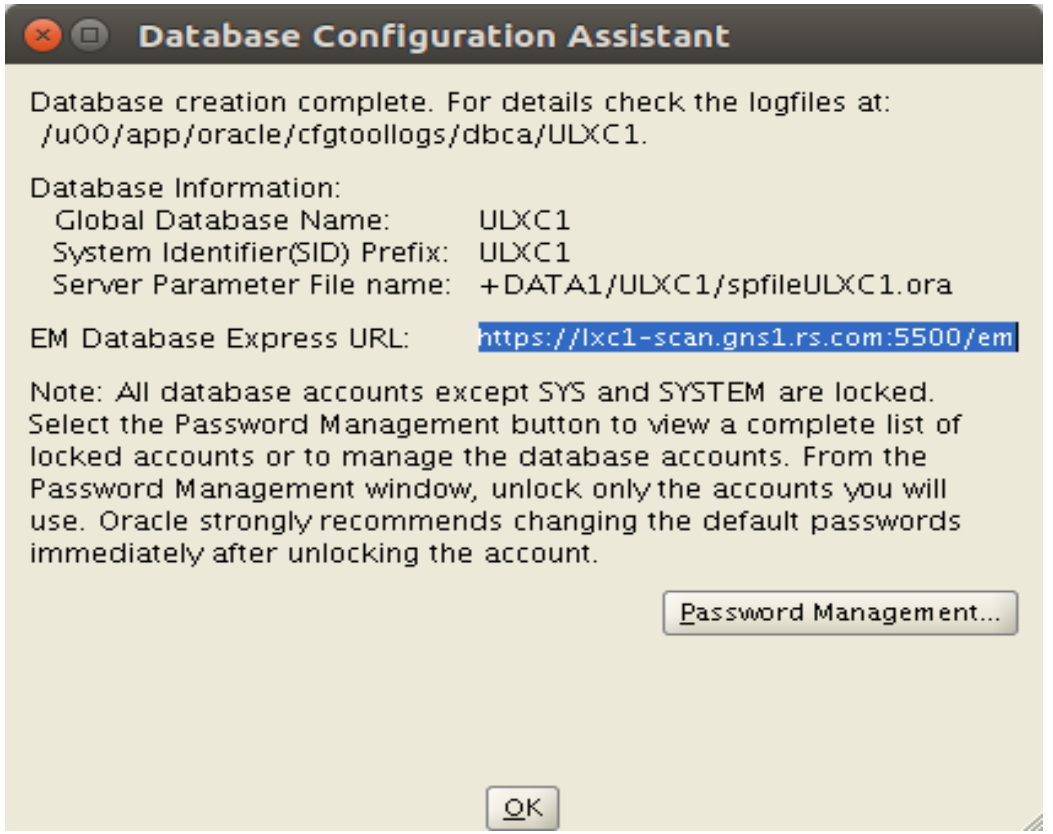
Notable GI installer screens – 12c rac on LXC



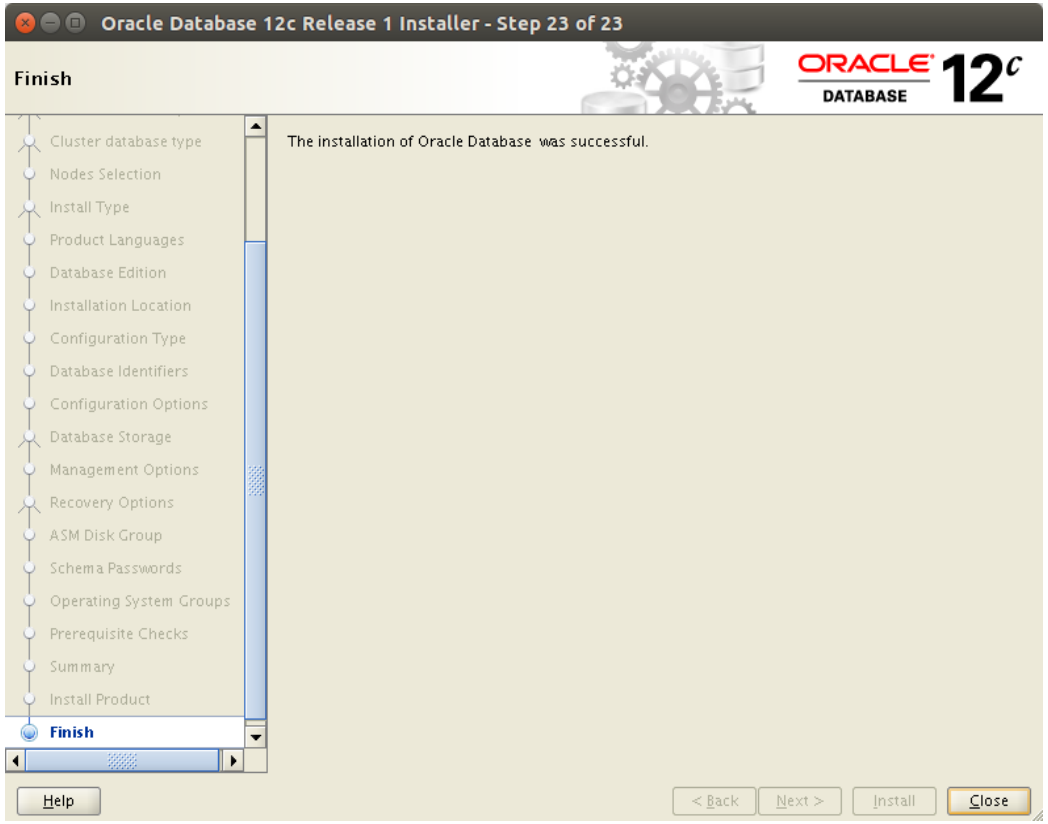


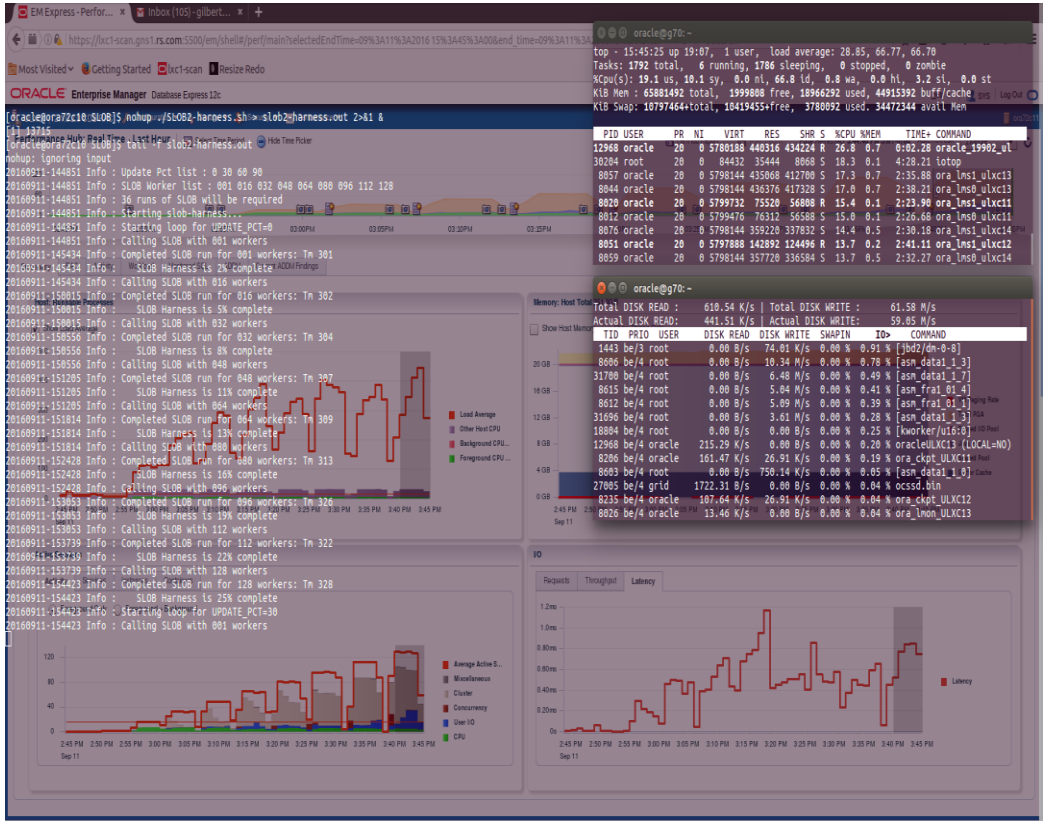
Notable GI installer screens – 12c RAC on LXC





Notable GI installer screens – 12c rac on LXC





Seven Advantages of Linux Containers

- Reduction in deployment (creation) time
- Reduction in bootup time (boot starts at init)
- Performance improvement (bare metal performance)
- Manageability improvement (files can be edited at host level)
- Consolidation of Enterprise to “Standard” (non-hypervisor) generic Linux skillsets
- Linux Containers deliver all the efficient resource utilization
- No hypervisor, no virtualized hardware, bare-metal performance

ROBIN

APPLICATION DEFINED DATA CENTER