



ROBIN SYSTEMS

Containerizing your Existing Enterprise
Applications

Adeesh Fulay
Director of Products

 @AdeeshF

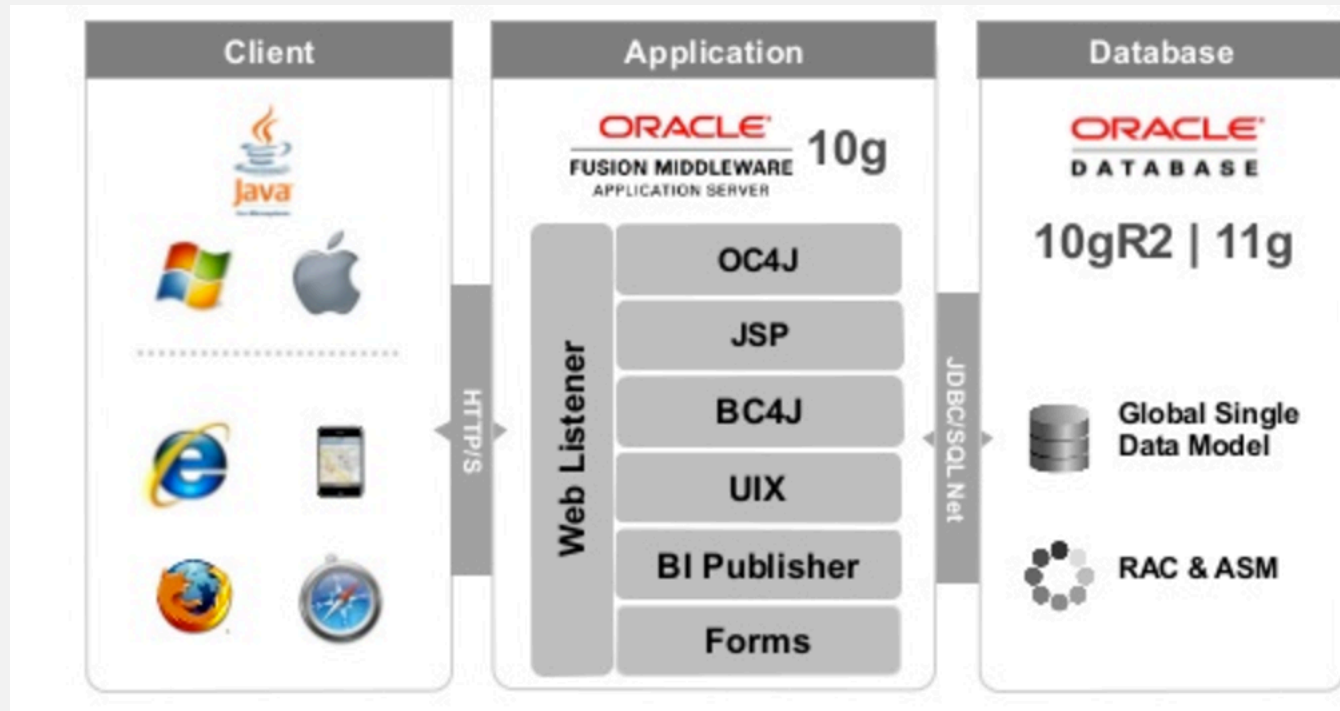
 adeesh@robinsystems.com

Agenda

- › Characteristics of Enterprise Applications
- › Current Business & IT Demands
- › Why Containers?
- › Path to Containerization
- › Robin Solution and Demos
- › Summary

Characteristics of Existing Enterprise Application

Oracle EBS 12.1 Suite / Custom Application



- > Typical 3 Tier architecture
- > Java or .NET based
- > Single logical Application Server and Database
- > Various libraries as components, called via in-memory function calls
- > Limited scale-out, primarily scale up
- > Each tier
 - > managed by a specialist team
 - > deployed on different infrastructure
 - > has specialized tooling for backup, clone, restore, patching, failover, DR, etc
 - > has specialized operational processes



UI Experts



App Admins



DBAs

Current Business & IT Demands

Business Goals



Accelerate Time to Market



Drive Innovation



Reduce Cost

IT Goals

- > Reduce time to move app from dev->test->prod
- > Automate frequent operations
- > Deliver cloud experience on on-prem infrastructure
- > Orchestrate deployment of new complex distributed apps
- > Rapid snapshot, clone, restore for data
- > Let developers use their preferred tools
- > Reduce h/w footprint
- > Improve server utilization w/o sacrificing performance
- > Migrate to commodity h/w instead of expensive SAN storage

Why Containers?



Stateless



Stateful



Data Apps



Enterprise Apps

- > Break out of dependency hell
- > Build, ship, deploy anywhere

- > Lightweight, high-performance “virtualization”
- > Multi-tenancy with bare-metal performance
- > Higher consolidation density than VMs
- > Simplify ops by eliminating OS or VM sprawl

Path to Containerization for Enterprise Applications

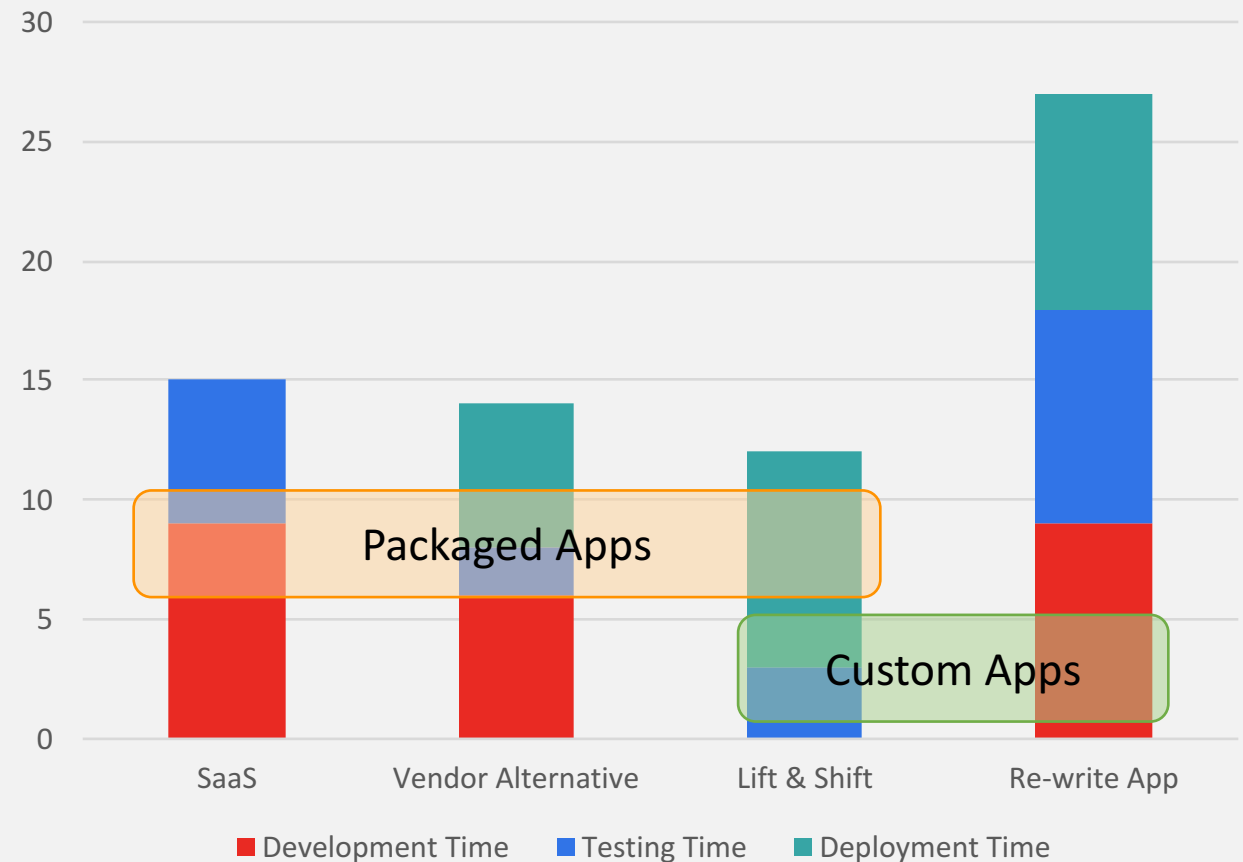
- > Vendor alternative or a SaaS alternative
 - > Works only for packaged apps
 - > May require re-coding of customizations
- > Re-write App
 - > Expensive and time consuming
 - > Disruptive change



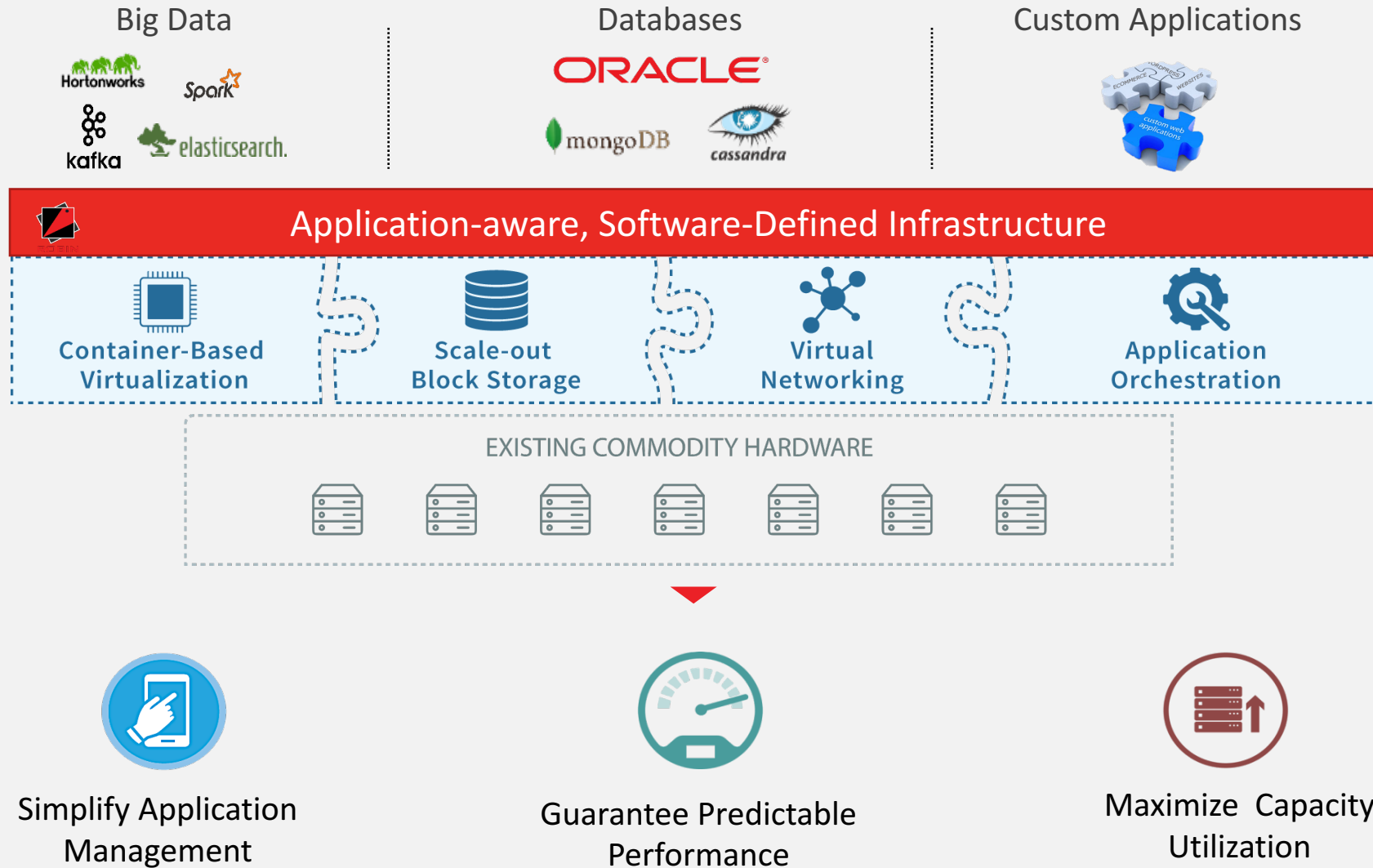
Lift & Shift

- > Break app into components, containerize, & deploy together
- > Best approach for existing apps

Effort vs Containerization Path

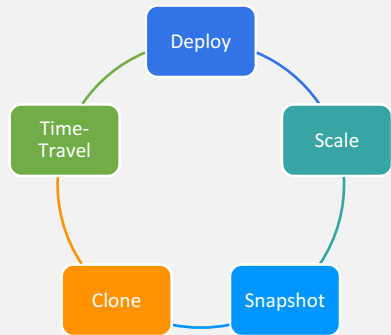


What is Robin?

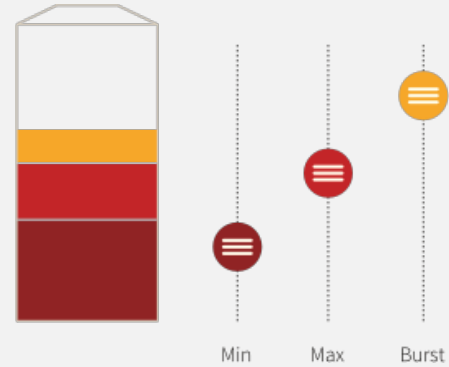


How Robin Works

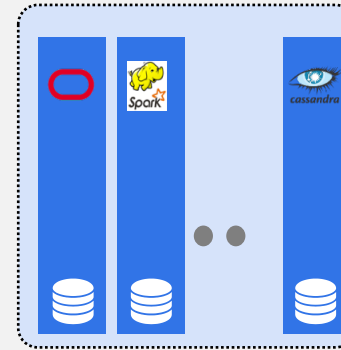
Dramatically Agile, Simple Application Management



Elastic, Guaranteed QoS

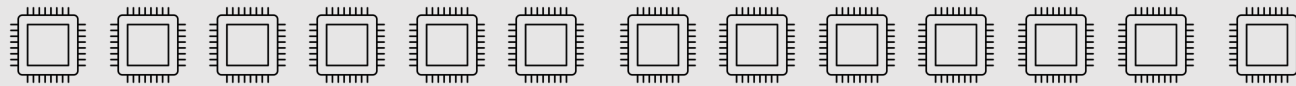


Consolidate with Bare Metal Performance



Fluid, Elastic, High Performance Infrastructure Platform

ROBIN CONTAINER-BASED COMPUTE PLANE



APPLICATION ORCHESTRATION + QoS GUARANTEE + DATA LIFECYCLE MANAGEMENT

ROBIN CONTAINER-AWARE, BLOCK STORAGE



EXISTING COMMODITY HARDWARE

← **Application-defined Compute**

← Cloud-extend Compute & Storage

← **Application-defined Storage**



ROBIN FABRIC CONTROLLER



The Robin Advantage

Business Goals



Accelerate Time to Market



Drive Innovation



Reduce Cost

Robin Provides

- > Push-button deployment of full stack
- > Integrated app & infrastructure management
- > Cloud-like agility on bare metal hardware
- > Simplified orchestration of app stacks & data pipelines
- > Instant snapshots & clones for the entire app stack
- > Dependency freedom via containers
- > 2-3x reduction in hardware
- > Guaranteed performance for all apps
- > Reduction in VM & OS sprawl

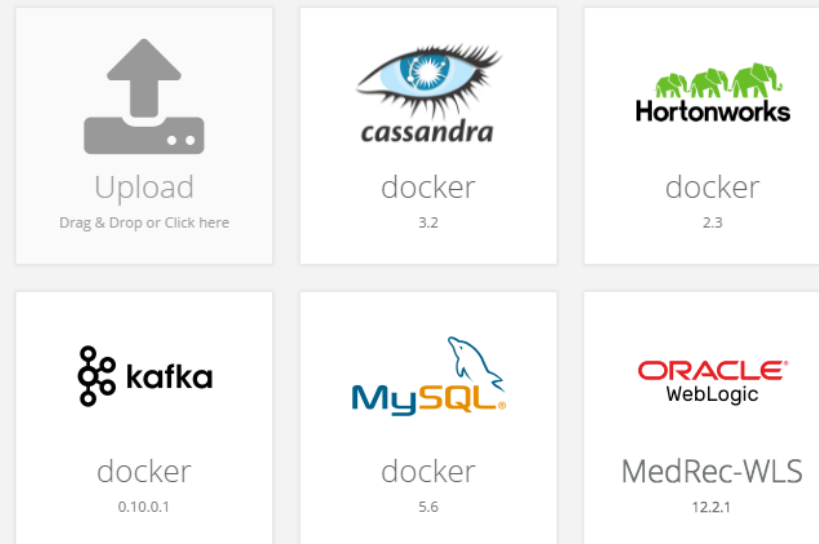
Application-Aware Infrastructure

```
name: MedRec WebLogic application
version: 12.2.1.0
icon: oracle_weblogic.png
roles: [oracledb, medrec_admin_server]
serialize: true
snapshot: enabled
clone: enabled
clonemode: unfenced

oracledb:
  name: Oracle Database
  multinode: false
  description: Runs oracle database server
  image:
    name: oracledb-medrec
    version: "12.1.0.2"
    engine: lxc
  compute:
    memory: 4096M
    cpu:
      reserve: false
      cores: 2
  storage:
    - type: root_fs
      media: ssd
      path: /
      size: 50G
  vnodehooks:
    postcreate: 'bash start_db.sh hostname={{HOSTNAME}}'
    poststart: 'bash start_db.sh hostname={{HOSTNAME}}'
```

- > Describe an application's needs in a YAML file called an **application manifest**
- > Bundle manifest file along with automation hooks into an **application bundle**

1-Click Application Deployment

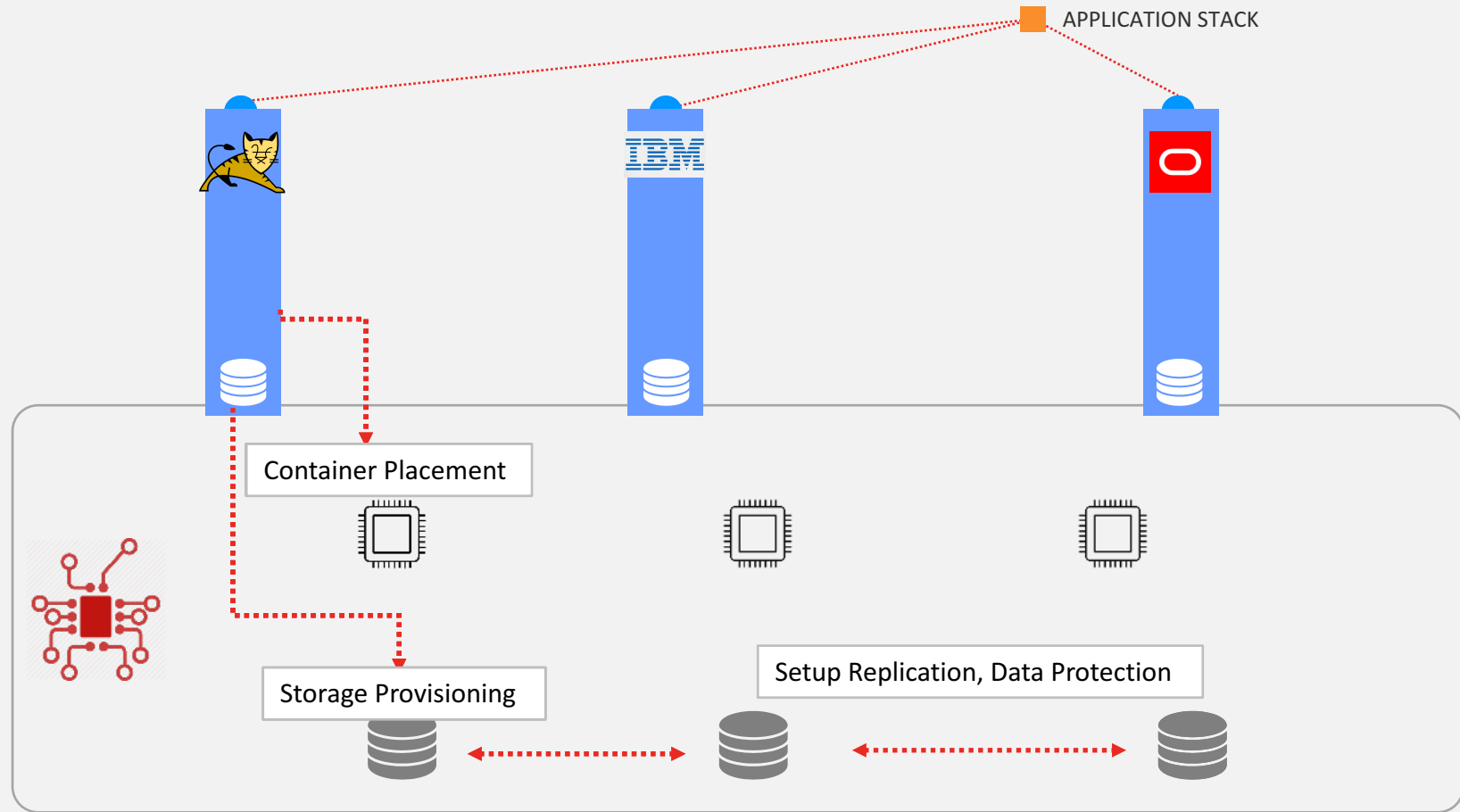




Demo: Full Stack Deployment

Robin Does All The Heavy Lifting For You

- > Container Placement
- > Storage Provisioning
- > Network Configuration

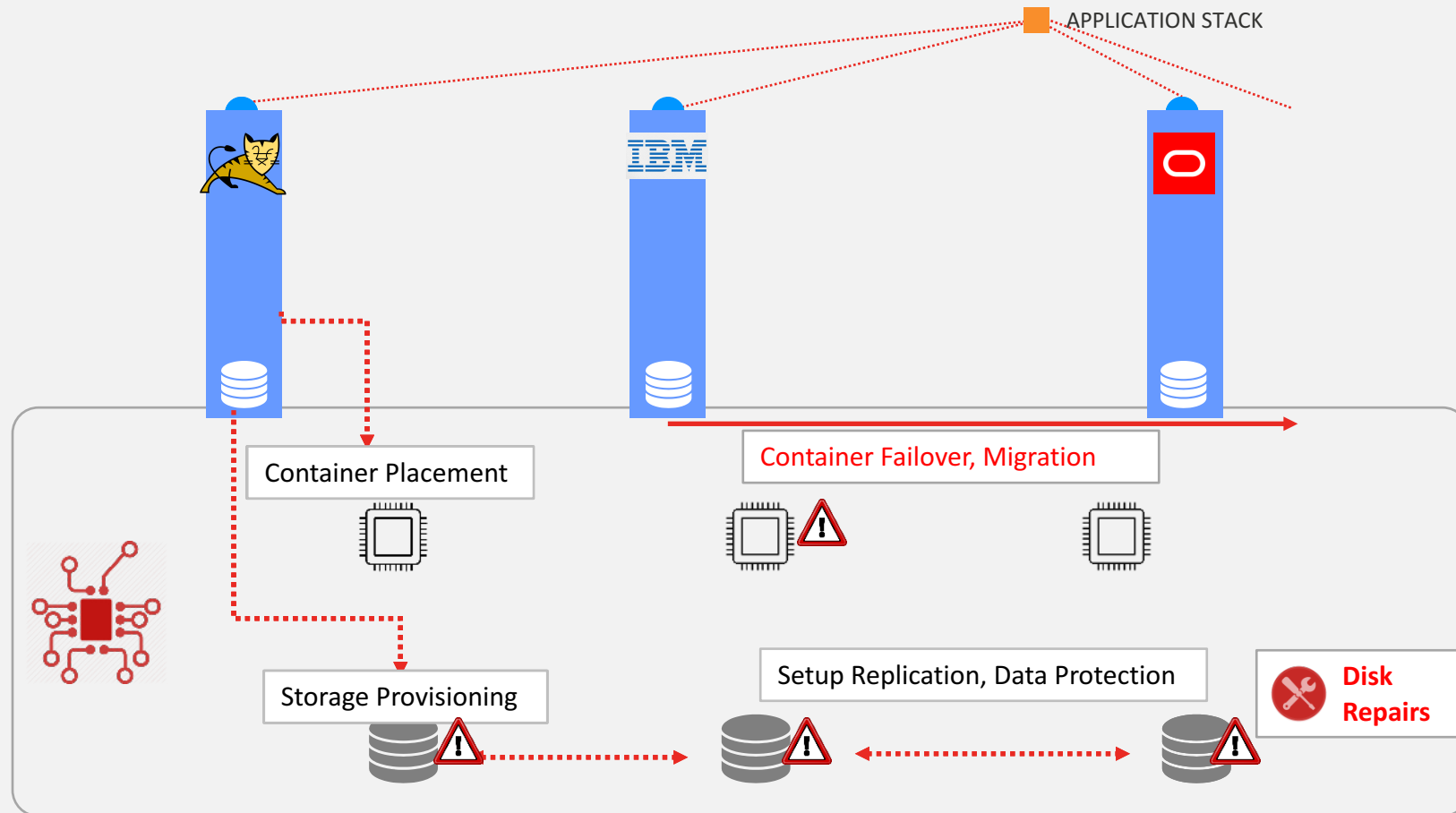


Robin Does All The Heavy Lifting For You

- > Container Placement
- > Storage Provisioning
- > Network Configuration

- > Instant Failover, even on bare metal
- > Manual node evacuation for planned maintenance (e.g. OS upgrade)

- > Enterprise grade data protection



Snapshot, Clone, and Restore Running Applications

Need to test a change?

Corruption or data loss?



Prod App



Prod App



Prod App



Prod App



Prod App



Prod App



Full Application Stack

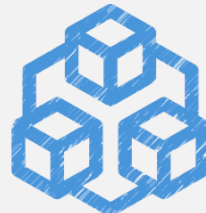
Snapshots

OS + App software + configuration + data

Lot more than just storage snapshots



Instant App Clone



Virtual Test App



Prod Scale Test Env in Minutes



Instant Restore

Erased data files
Restore backup
Restart app





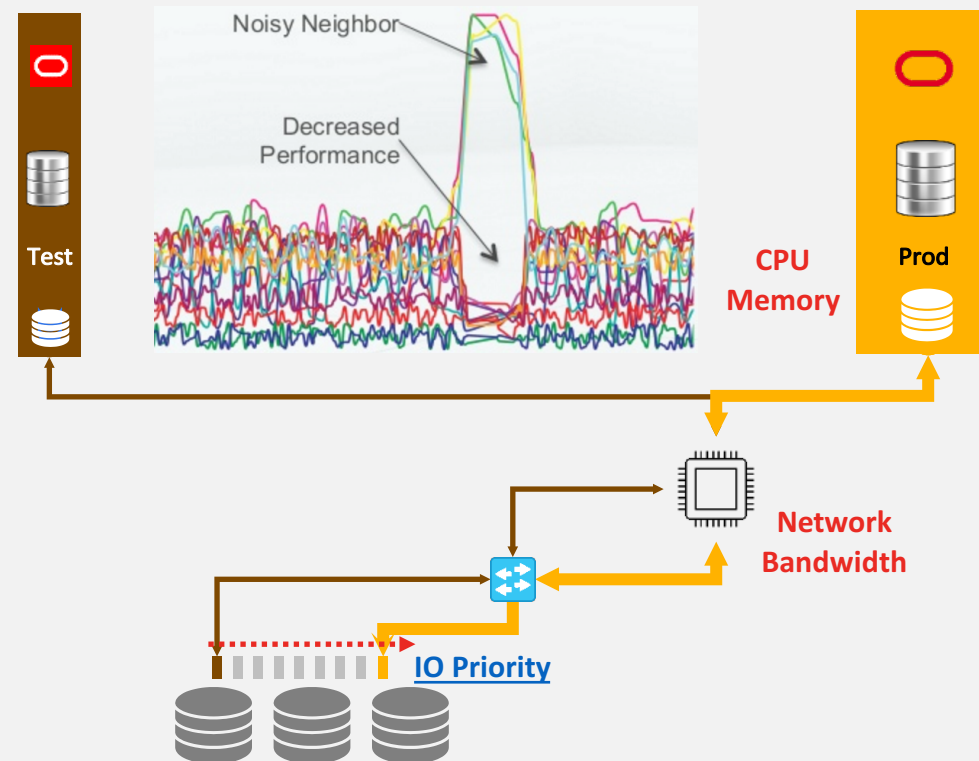
Demo: Snapshot, Clone, Restore

Consolidate Workload with Predictable Performance

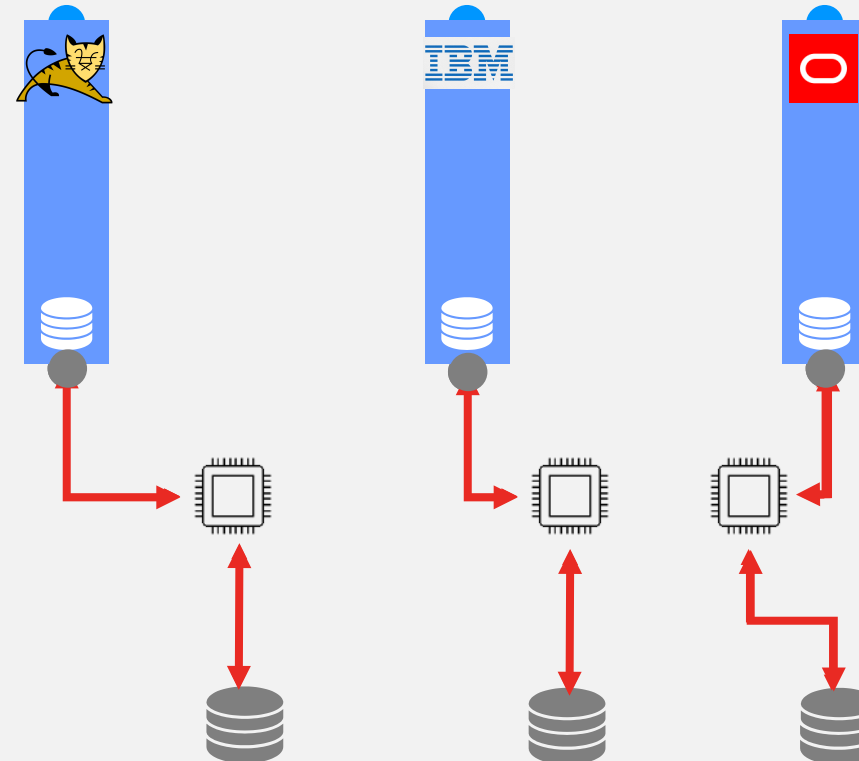
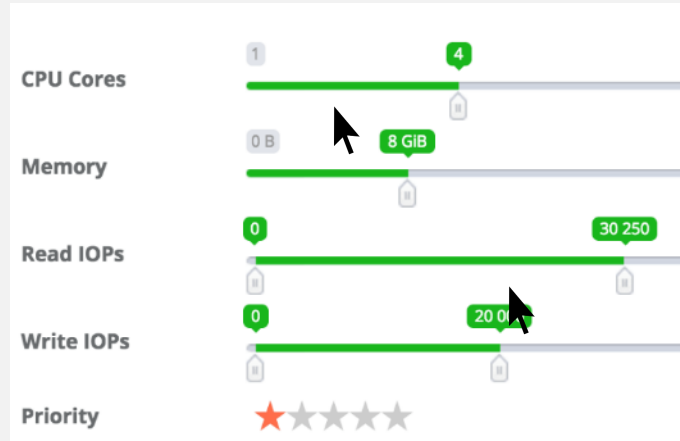
- > **Only** product that can guarantee MAX and MIN IOPS per container, per App
- > Performance isolation between Apps
- > Thin Clones don't slow down Master App

Without Robin:
Clone slows down
Prod App

With Robin:
Perfect isolation between
Clone and Production App



Elastically Scale Applications Up and Down



Just scaling up CPU share does not increase throughput

Robin enables Scaling Up BOTH Compute and Storage Resources

- > **Only** product that enables “Scale Up”
 - > 1-Click add CPU/Storage resources
 - > Maintain same #Nodes
- > On-demand and Instant
 - > No data redistribution overhead
 - > No need to stop the cluster
 - > Ideal as quick fix for temporary demand



Demo: Quality of Service

Summary

Current State



Best Performance **OR** Best Utilization



Best Performance **AND** Best Utilization

Complex Application Stack Management



Dramatically **Simple** Application Management

Unpredictable Application Performance



Predictable App-to-Spindle Performance

Painful Infrastructure Management



100% Application-Driven, **"Invisible" Infrastructure**

Stalled, Frustrated Developers



Agile Developers, Dev-Ops Harmony





THANK YOU

 www.robinsystems.com

 info@robinsystems.com
