

ORACLE®

Ksplice & Linux Containers Overview

Presenter: Anuj Verma

Title: Senior Pre-Sales Consultant



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract.

It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



ORACLE®

Ksplice Overview

Use Ksplice which ever way fits your needs

What is Ksplice?

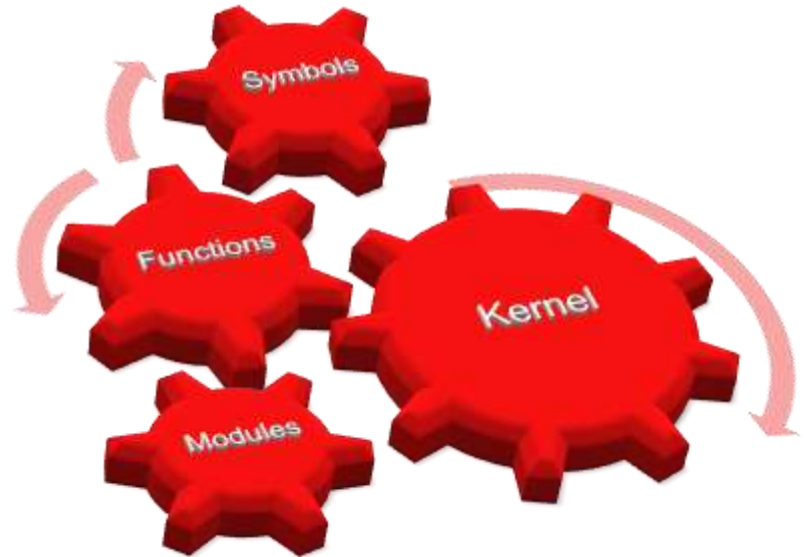
Compliance, Security and Integrity without Downtime

- Founded in 2008, acquired by Oracle in 2011
 - Over 700 customers across multiple industries
 - Creator of innovative zero downtime updates for Linux
- Ksplice technology applies updates without rebooting
 - No-Reboot Benefits:
 - Improve Security - apply security patches without reboot
 - Reliability (Compliance) - update the whole system
 - Lower Op-Ex costs - no weeknight or weekend upgrades
 - Active Support Fixes - applied to relevant system state

Zero-downtime Kernel Updates

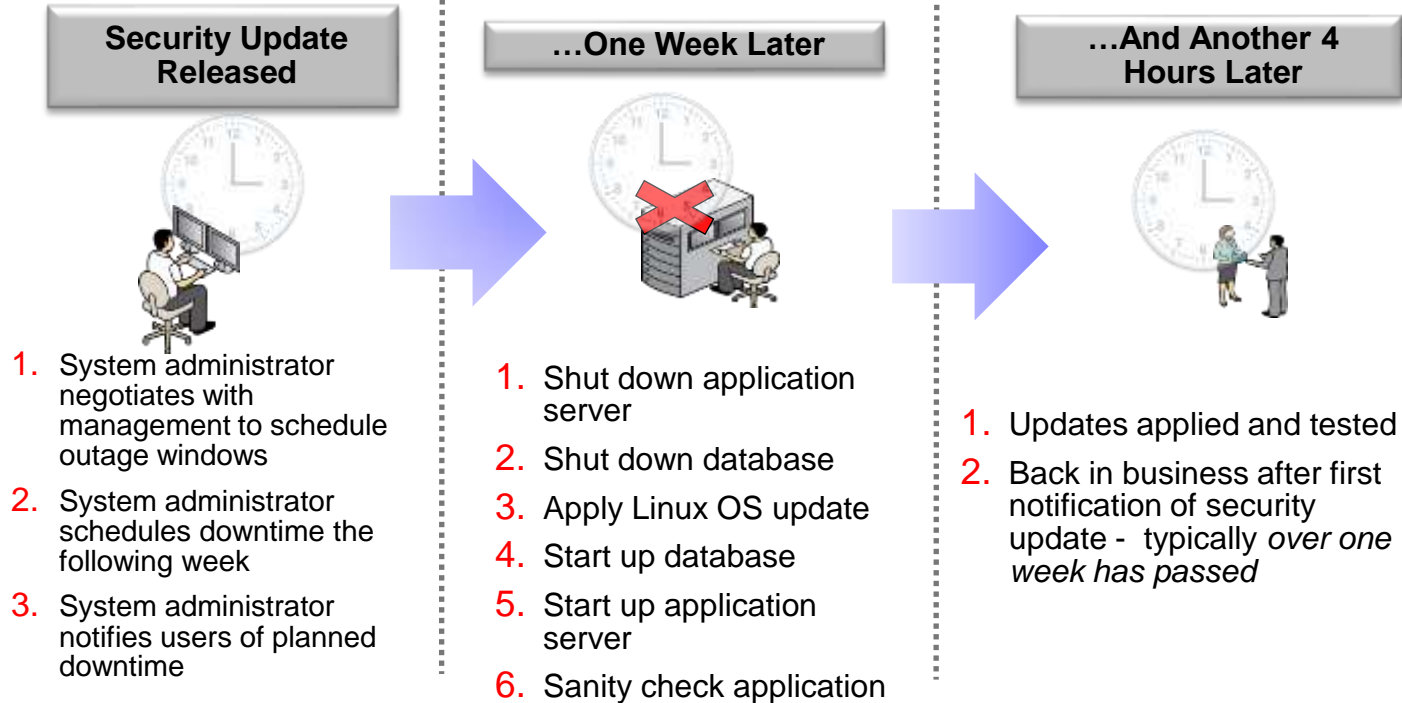
Oracle Ksplice is available with Oracle Linux Premier Support

- Oracle Ksplice capabilities are extensive
 - Capable of patching a variety of kernel issues
 - Easy to apply and rollback updates
 - Simple, flexible tools and options for installing updates
 - Proven track record in providing stable updates for production systems



Why use Ksplice?

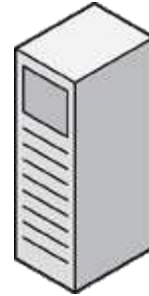
Reboots are Disruptive, incur Downtime and cause Delays



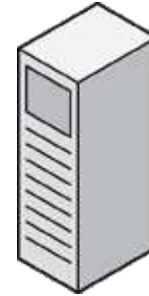
Why use Ksplice ?

Debug with Zero-Downtime

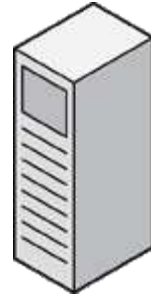
- Production diagnostics is time consuming and has a high impact to users and applications
- Ksplice can assist in root cause kernel analysis by:
 - Install debug kernel
 - Apply hot fix
 - Apply final patch



DEBUG



HOTFIX



FINAL

Zero-Downtime and No Reboot

Experience

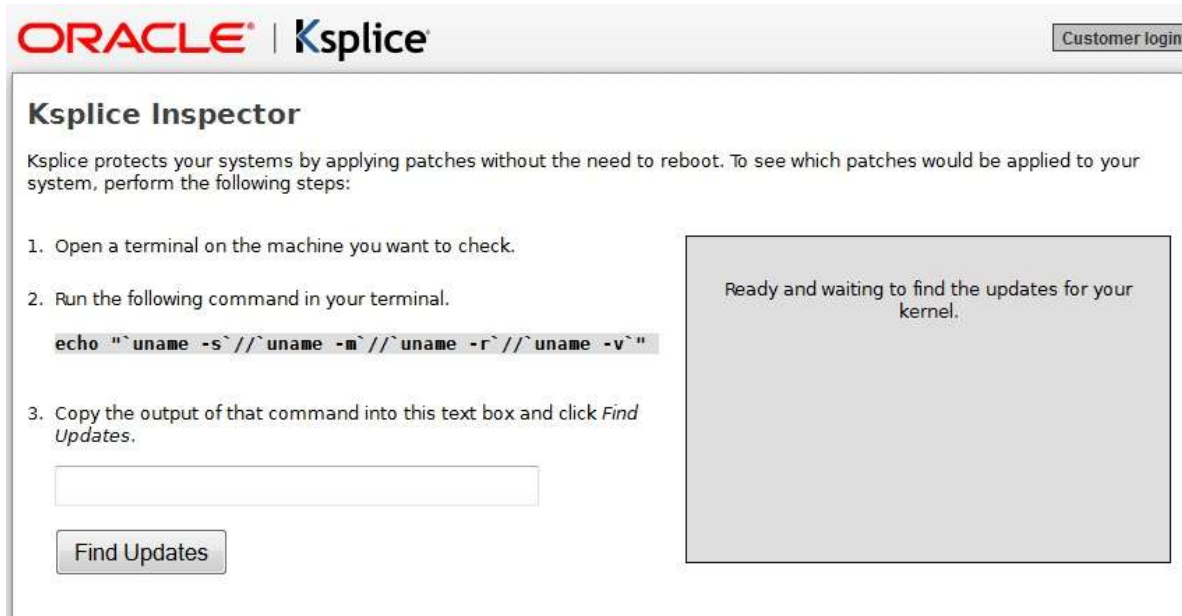
Oracle Ksplice - providing zero-downtime kernel updates since 2007

- Ksplice has provided zero-downtime kernel updates for a number of years and has a proven track record in delivering quality patches under this model
- Development tools and infrastructure are in place to track kernel updates and provide updates for a number of platform releases and kernel versions
 - Oracle Linux 5, Oracle Linux 6, Red Hat Enterprise Linux 5 and 6*
 - This includes the various kernels released over the course of the product lifecycle, including the Unbreakable Enterprise Kernel and the Red Hat Compatible Kernel
 - Fedora and Ubuntu
- Ksplice production patches released: **Over 1 million and counting!**
 - kGraft and kpatch: **0**

Ksplice Inspector

Is your System Compliant? Find out: <https://ksplice.oracle.com/inspector>

- Validate the patch level of your kernel; Apply the patches you need



ORACLE | **Ksplice** Customer login

Ksplice Inspector

Ksplice protects your systems by applying patches without the need to reboot. To see which patches would be applied to your system, perform the following steps:

1. Open a terminal on the machine you want to check.
2. Run the following command in your terminal.

```
echo "`uname -s` / "`uname -m` / "`uname -r` / "`uname -v`"
```
3. Copy the output of that command into this text box and click *Find Updates*.

Ready and waiting to find the updates for your kernel.

Installing Ksplice

Simple, Safe and Swift

- Go to ULN < <http://linux.oracle.com> >
 - Log in using your SSO and Click on “Register Ksplice” in the upper right hand corner
 - Confirm receipt of your ksplice “ACCESS KEY”
- 60-second install process - download and execute the ksplice uptrack installation script

```
wget -N https://www.ksplice.com/uptrack/install-uptrack  
sh install-uptrack <ACCESS KEY>
```
- You now have a “yum like” repository for ksplice

```
uptrack-upgrade -y
```
- Done! No initial reboot

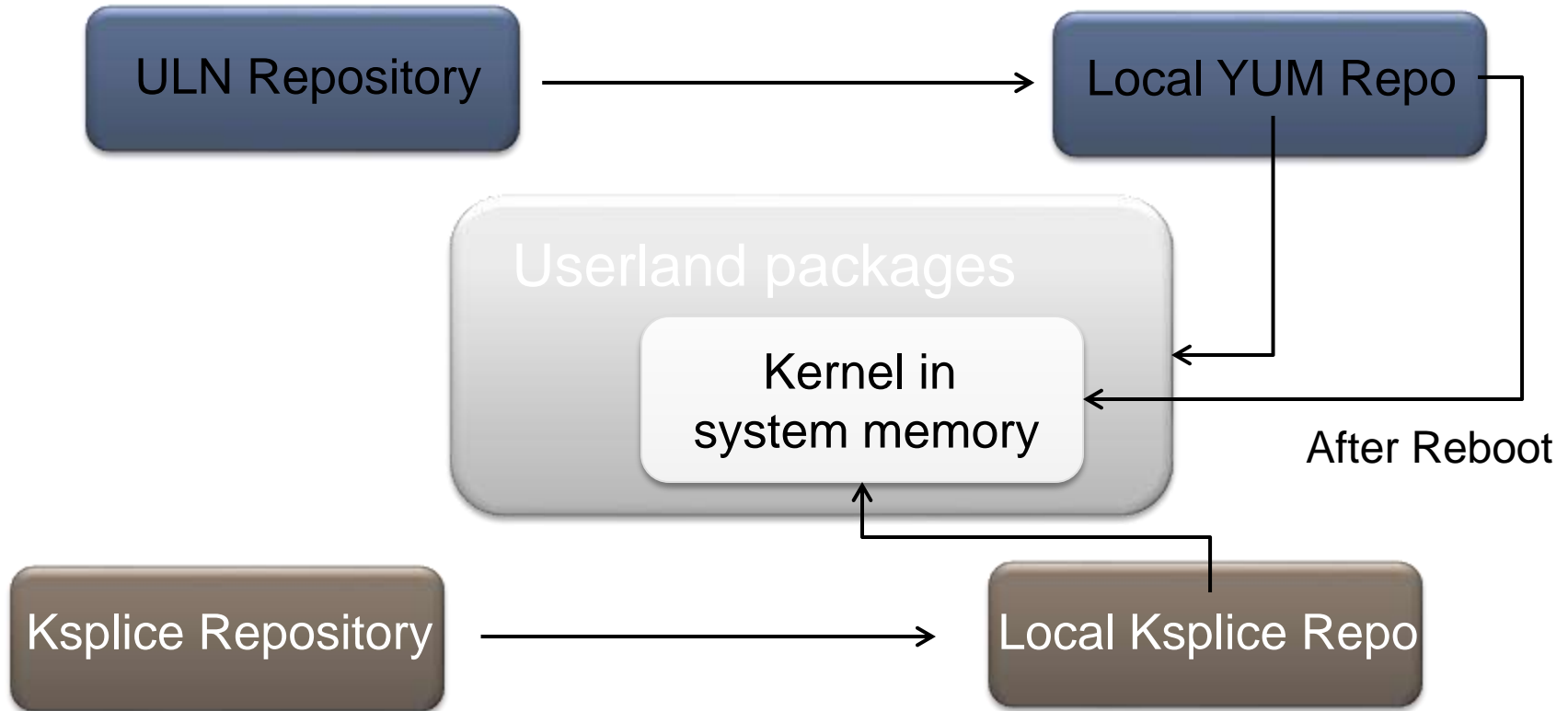
Dynamic kernel patching:

How it works - Modifying the runtime kernel

Patching Oracle Linux System using YUM

- Oracle Linux patches are applied as RPMs from the ULN
- Patches are categorized as:
 - Security [CVE]
 - Errata (bugs)
 - Features/Enhancements (simple updates)
- There are basically two kinds of patches
 - Userland
 - Can be applied any time and does not require a reboot
 - Kernel
 - Can be installed any time, but to take effect requires a reboot
 - What if you need to immediately patch the kernel without rebooting?

Patching an Oracle Linux System



Always Accessible Systems

No-Reboot Benefits Summary

- Improved Availability
 - Apply critical updates and security patches without rebooting
 - Eliminate downtime and disruption for users and customers
 - Update while applications are running
 - Coordinate Ksplice with support hot patches
- Enhanced Security
 - Reduce windows of vulnerability
 - Increase compliance
- Reduced Operational Cost
 - Eliminate unplanned maintenance downtime
 - No more long nights and weekends spent rebooting servers for kernel updates
 - No need to coordinate with system users about outages caused by reboots



NEW: Ksplice 30 Day Free Trial for RHEL

RHEL Ksplice Trial

- Available to anyone with RHEL 5 or RHEL 6
- Simple registration
- One minute installation of Uptrack client
- <http://ksplice.oracle.com/rhel-signup>



Sign in | Linux Security Updates Without a Reboot | Ksplice Uptrack

<http://www.ksplice.com/rhel-signup>

ORACLE | Ksplice

Free 30-day Ksplice trial for Red Hat Enterprise Linux

You are seconds away from trying out Ksplice zero downtime kernel updates for your RHEL 5 or 6 system!

Take back your weekend and say goodbye to lengthy maintenance windows for kernel updates. With Ksplice, you can install kernel updates while the system is running. Stay secure and compliant without the hassle.

But don't take our word for it, try it now with our **30-day free trial**. Once you experience Ksplice, you'll never want to go back. Ksplice is a standard feature of Oracle Linux Premier Support, and when your trial ends, we'll be happy to help you make the switch.



Create your account

Account details

First name*

Last name*

Company*

Email*

Phone

Choose how you'll sign into your account

Username*

Password*

Confirm*

I agree to the [terms of service](#).

Simplest installation ever
Experience zero downtime updates without rebooting even once to setup. This trial works on the RHEL kernel you are already running, and it usually takes one minute to install.

Works in your environment
Ksplice complements your existing infrastructure. It works alongside your and your existing tools. Its job is keep the kernel up to date, in memory, and it is highly scriptable, with command-line tools and an API.

Rollback capability
Instantly roll back any Ksplice update, rebootlessly. Removing Ksplice is just as easy as installing it, so why not try it now?

Questions? Please email ksplice-support_us@oracle.com

Copyright © 2011, Oracle Corporation and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. [Legal](#)



ORACLE®

Linux Containers (LXC) Overview

Operating System Level Virtualization

Agenda

- What are Linux Containers?
- Containers vs. Virtualization
- Control Groups
- Resource Isolation
- LXC
- Templates
- Use Cases
- Final Comments

What Are Linux Containers?

- Containers are a way to separate and isolate processes or groups of processes in Oracle Linux
- Containers takes cgroups resource management and adds POSIX file capabilities to implement process and network isolation
- There are two types of containers:
 - Application
 - System

What Are Linux Containers? (Cont.)

- Application containers are designed to hold and run a single application:
 - Has its own process and network isolation from the rest of the system
 - Only requires the minimal set of libraries needed to run the application
- System containers are designed to have a complete isolated copy of the entire operating system available:
 - Lighter-weight than running a hypervisor, such as Oracle VM or VirtualBox
 - All containers share the same kernel and hardware resources
 - Can even run Oracle Linux 5 in a container on Oracle Linux 6

Containers vs. Virtualization

- Containers are often cited as a form of virtualization
 - Depends on the definition of “virtualization”
 - More like partitioning or isolation
- Biggest difference – no hypervisor is used
 - Uses process isolation and other features to separate resources
 - Still “bound” to the host – kernel version is the same
 - Still “bound” to the hardware – no way to change what appears
- Inside, containers “feel” like a VM
- Outside, containers “look” like a system process

Containers vs. Virtualization (Cont.)

Advantages of Using Containers

- Setup and teardown times are *FAST!*
 - Especially when combined with Btrfs
 - Makes for easy “disposable” environments
- Less resources required
 - Useful when all you want to run is Linux
- Ability to introspect from the host system
 - “Peer into” containers from the controlling host

Containers vs. Virtualization (Cont.)

Advantages of Using Virtualization

- Variability of workloads
 - Many operating systems available, including Windows
 - Completely isolated from the underlying hardware, so can be moved easily
- Independence from the host
 - Does not share with the host system
 - Each guest “owns” its own resources
 - Can be kept unaware that it is being virtualized

LXC

Lightweight and fast resource isolation

- Moved from technology preview to fully supported feature in UEK R3
- Operating system-level virtualization
- Isolates individual services or a complete Linux operating system from other services running on the same host
- Allows direct and fair distribution of CPU and I/O-resources

LXC (Cont.)

Command Line Tool Set For Containers

- Setup a container (root file system and config)
`lxc-create`
- Boot the container (by default, you get a console)
`lxc-start`
- Attach a console (if you started in background)
`lxc-console`
- Shutdown the container
`lxc-stop`
- Destroy the container (tear-down the file system, etc.)
`lxc-destroy`

Templates

Container Creation Short-Cuts

- When creating a container, a template or template script is used to populate the container file system:
 - At the base level, a template can be a simple archive (tarball)
 - Template scripts can be used to clone existing templates or file systems (*Fast with Btrfs!*) or to install the container from a yum repository
- Containers can also be created/cloned from Oracle VM Template file system images

Use cases

- Running Oracle Linux 5 and Oracle Linux 6 in parallel
- Run an Oracle Linux 5 container on an Oracle Linux 6 system that is using UEK R3, even though URK R3 is not supported for Oracle Linux 5
- Run an i386 (32-bit) container on a x86-64 (64-bit) kernel
 - Note: you cannot run a 64-bit container on a 32-bit kernel
- Running many copies of application configurations on the same system – example many LAMP stacks
- Sandbox environments for Dev & Test
- Creating containers where each appears to have its own IP address

Final Comments

- Linux Containers provide another tool for maximizing the use of an Oracle Linux system
- Typically you can expect 5-10x density over virtualization on the same hardware – plus, no manager is required
- Setup and tear-down times make containers an easy way to share resources with minimal effort

ORACLE®