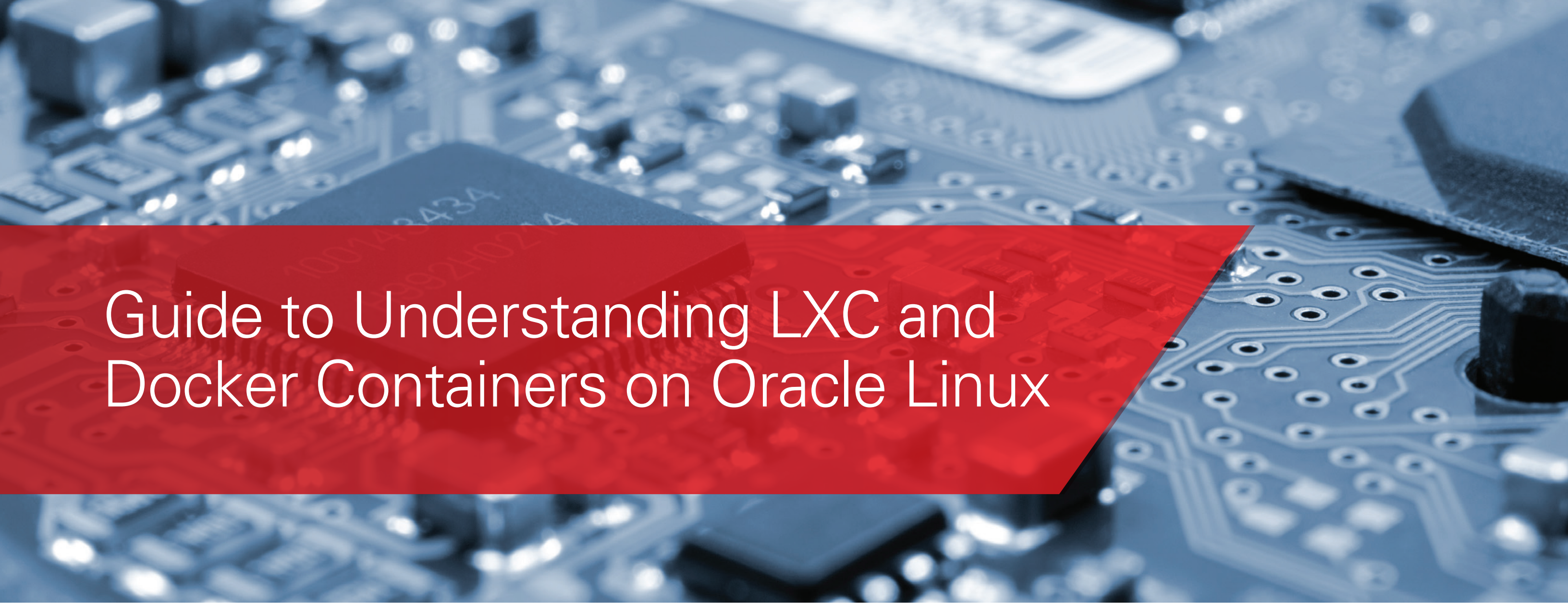

Guide to Understanding LXC and Docker Containers on Oracle Linux

Selecting Linux containers (LXC) or Docker containers for application delivery on Oracle Linux requires some thought. While each container technology has advantages, there is no "one size fits all" solution, but rather a best choice that fits the task at hand.

TABLE OF CONTENTS

CONTAINER TECHNOLOGIES: THE BASICS

How do LXC and Docker containers differ from well-established hypervisor technologies (such as Oracle VM (Xen), Microsoft Hyper-V, or VMware's ESXi)?

Table 1: Container Technology Basics

Traditional Hypervisor	LXC	Docker
▪ Implemented as an additional software layer on top of the native host OS	▪ Shares the same kernel space (with the same drivers and kernel modules) as the host OS	▪ Built using file system layers that capture the complete application environment into a virtual container ▪ Base image consists of OS binaries and libraries ▪ When modified, the new container shares the same binaries and libraries as the base
▪ Supports multiple virtual machine (VM) guests, performing hardware emulation to access underlying physical server resources	▪ Unlike a hypervisor-based virtualization solution, LXC containers support only Linux guest VMs that use the host's kernel and device drivers	▪ Containers are managed and run by the Docker Engine** ▪ A Dockerfile (a small text file) can be defined to build a container image capturing all its dependencies ▪ Docker images built from a Dockerfile can run on any host running the Docker Engine
▪ Each VM guest contains an isolated, full-blown OS instance with its own kernel and user space	▪ Each container has its own isolated user space and takes advantage of the cgroups resource management and namespace capabilities in Linux to implement resource control and isolation	▪ Unlike an LXC system container, a Docker container does not run a separate init process by default, so a single container can't easily host a multitiered stack ▪ Instead, multiple Docker containers (each running a separate service or application) should be connected to one another to construct a full multitiered stack
▪ A single physical machine can support applications running simultaneously on completely different OSs while fully sandboxing them from one another *	▪ Can be deployed in two different ways: as application or system containers, although they are primarily deployed as system containers	▪ Unlike LXC, the ability to capture an application, its environment, and all dependencies and move them across platforms is a significant benefit.
▪ The additional work of emulation imposes some degree of performance overhead	▪ Considered a lighter weight approach to creating an isolated system in comparison to conventional virtualization solutions	▪ Considered a lighter weight approach to isolating applications in comparison to conventional virtualization solutions

* As an example, Oracle VM (which uses the Xen hypervisor) can support applications that run within separate Linux, Oracle Solaris, and Microsoft Windows guest VMs.

** At this time, the Docker Engine is available for all major Linux distributions and Microsoft operating systems.

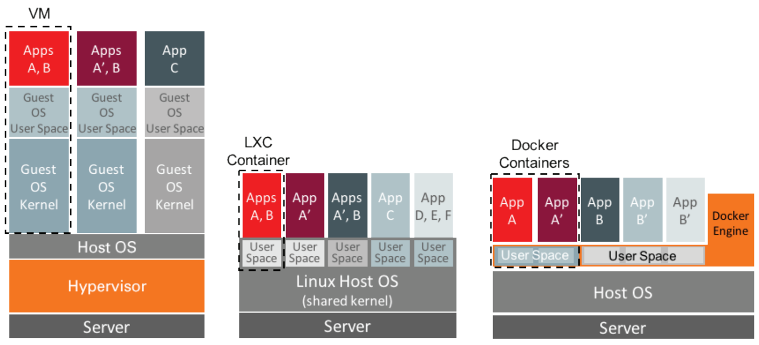

Figure 1. LXC and Docker containers versus hypervisor-based virtualization.

LXC AND DOCKER SIMILARITIES

Because they are both lightweight virtualization tools, LXC and Docker containers have some notable similarities:

- Unlike a traditional hypervisor-based approach, LXC containers don't require the duplication of a full operating system installed as a guest, and Docker containers rely on efficiently layered images.
- Both LXC and Docker containers are open source, taking advantage of private Linux namespaces to sandbox applications and Linux cgroups to enable resource allocations and limits.
- With both technologies, each container gets its own networking stack—a private network interface with a virtual bridge that allows the container to communicate while providing a separate VLAN.
- On Linux systems, both LXC and Docker containers can leverage an underlying Btrfs file system for fast snapshots, which helps to create new Docker image layers quickly or copy a root file system rapidly when you are cloning an LXC container.

LXC AND DOCKER DIFFERENCES

LXC and Docker containers also have significant differences—even beyond the prominent difference of an LXC system container supporting an init process while a Docker container does not:

- Docker containers are built from previously captured Docker images (Figure 2)
- Images can be downloaded from (as well as published to) a Docker registry, either the public Docker Hub Registry maintained by Docker, Inc. or a privately maintained local registry
- Oracle publishes images for various distribution versions of Oracle Linux 6, Oracle Linux 7, and MySQL to the Docker Hub Registry, and these can be freely downloaded and run as a base layer of a Docker container, which can be saved to build a new container image
- Docker captures all changes made to a base image (such as installing application packages, resolving dependencies, and configuring settings),
- To save space, Docker stores only changes to a base image in layers to reconstruct the full container image
- You can build and run a container image interactively in the Docker Engine, but more commonly, a Dockerfile is defined to build the container image and run the application

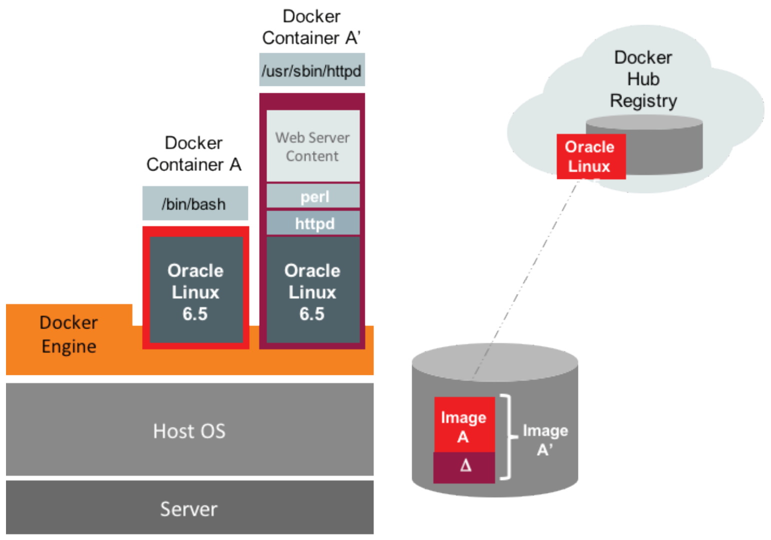

Figure 2. Docker captures changes to a container image and uses layers to efficiently reconstruct a container.

EXAMPLE USE CASES FOR DOCKER

- Docker is an application container best suited for a single application or service
- Docker containers are ideally suited for isolating a service, such as a web service, and container instances can be easily cloned to scale the service and support greater capacity

CONCLUSION

Container technologies have substantial benefits, especially given the need to isolate applications to increase service levels. Both LXC and Docker containers provide application sandboxes so that if a security flaw is exploited or a containerized application is compromised, it's unable to affect other applications and services running in other containers. Depending on your use case, there should be a lightweight LXC or Docker container technology that is a good fit.

LEARN MORE

Oracle Linux documentation (the *Oracle Linux Administrator's Solutions Guide for Release 6* or the *Oracle Linux Administrator's Guide for Oracle Linux 7*) has detailed sections on getting started with both LXC and Docker container technologies.

There are also excellent resources for learning more about Docker at http://docs.docker.com/, including information on installing Docker on different operating systems, the *Docker Engine User Guide*, and the Docker reference manual.

ORACLE

Oracle Corporation
WORLDWIDE HEADQUARTERS
500 Oracle Parkway
Redwood Shores
CA 94065
U.S.A.

WORLDWIDE INQUIRIES
Phone:
+1.650.506.7000
+1.800.ORACLE1
Fax:
+1.650.506.7200

oracle.com