OS For Cloud Computing

Red Hat Enterprise Linux 8

Red Hat Enterprise Linux 8

- Cloud computing and virtualization are very heavy on the OS, and this is one more reason why Linux and the open source development model are so important in today's computing environment
- Microsoft Windows Azure and Google Chrome OS are among current cloud operating systems, but Linux systems are used everywhere—the Internet, point-of-sale systems, the world's stock markets, smart TVs, in-flight entertainment systems, and most of the top supercomputers in the world run on Linux
- Your cloud instances may be based on Linux, and your private or public cloud environment is also probably based on Linux

Intelligent OS For Hybrid Cloud

- Red Hat Enterprise Linux 8 is the intelligent OS for hybrid cloud
  - Migrates workloads across environments
  - Is consistent across public, private, and hybrid cloud environments
  - Has long life-cycle commitment- a robust ecosystem of certified hardware, software, and cloud partners, which now comes with built-in management and predictive analytics

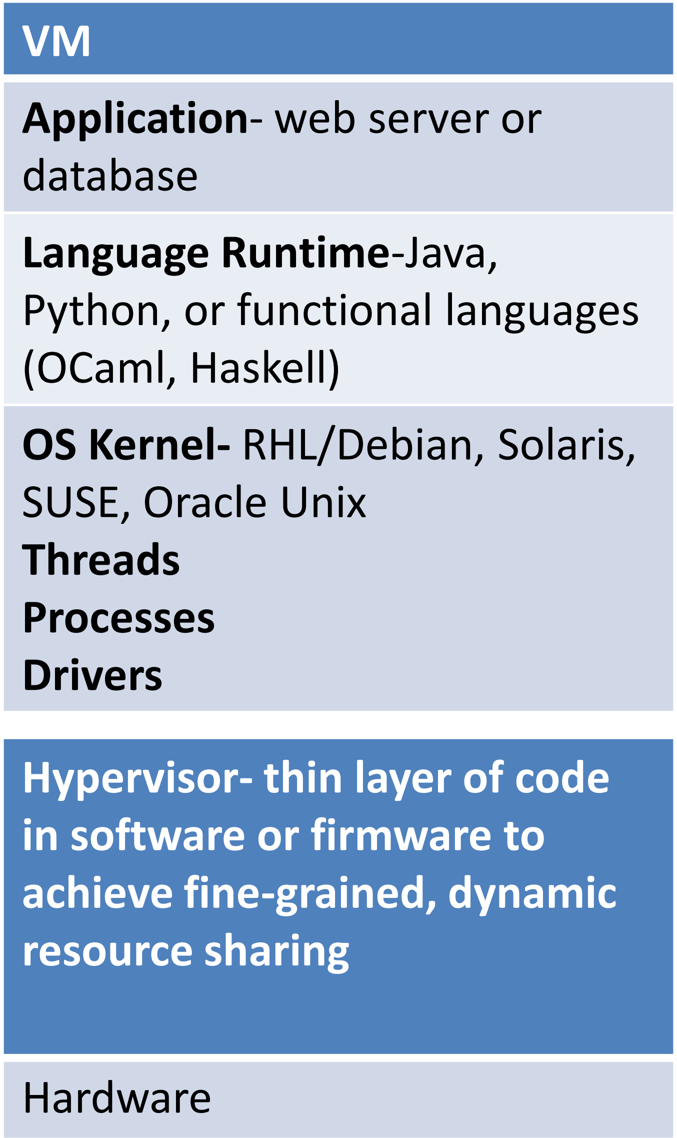
The Power Of Linux

- Red Hat was one of the first companies to recognize and identify the power of Linux; they created their own version of Linux (their own commercialization of the Linux kernel), being able to compete with Microsoft in the hardware server space
- Windows server OS was expensive and limited
- Using Linux , you could setup an Apache server, with a PHP module and a mySQL back end and be competitive with Windows technology Enterprise server and SQL server as a database
- Combining nginx with the LAMP stack (the Linux operating system, the Apache HTTP Server, the MySQL relational database management system, and the PHP programming language), Linux offered a professional product (including security)

Red Hat Linux Version 7

- With release Red Hat version 7, virtualization and cloud architecture built into the OS itself, Red Hat was ready to make the move into the virtualization market, where the big player was VMware
- Red Hat Linux 7 was capable of sandboxing processes- it means creating an application that runs in a container, a container with a namespace and control group
- Linux process- An application runs in an application thread with security policies; we can share resources inside the thread (the smallest sequence of program instructions managed independently by a scheduler- part of the OS)
- Processes inside the box can not share resources- In the Linux architecture, processes are independent; the resources available to one process are not available to another process
- There are exceptions to the rules such as the file system; it is possible with groups in Linux to restrict access to files; a process can have access to a file that can be denied to another process
- We don’t place this kind of restrictions on threads running within a process
- It’s possible to create a process in a file system directory that can run as its own independent box or container, and this independent process we can think of as our VM, or virtual box and this can be completely firewalled or sandboxed from other processes because this is inherited in the Linux architecture, which is the key point for security reason

Architecture



VM

Application- web server or database

Language Runtime-Java, Python, or functional languages (OCaml, Haskell)

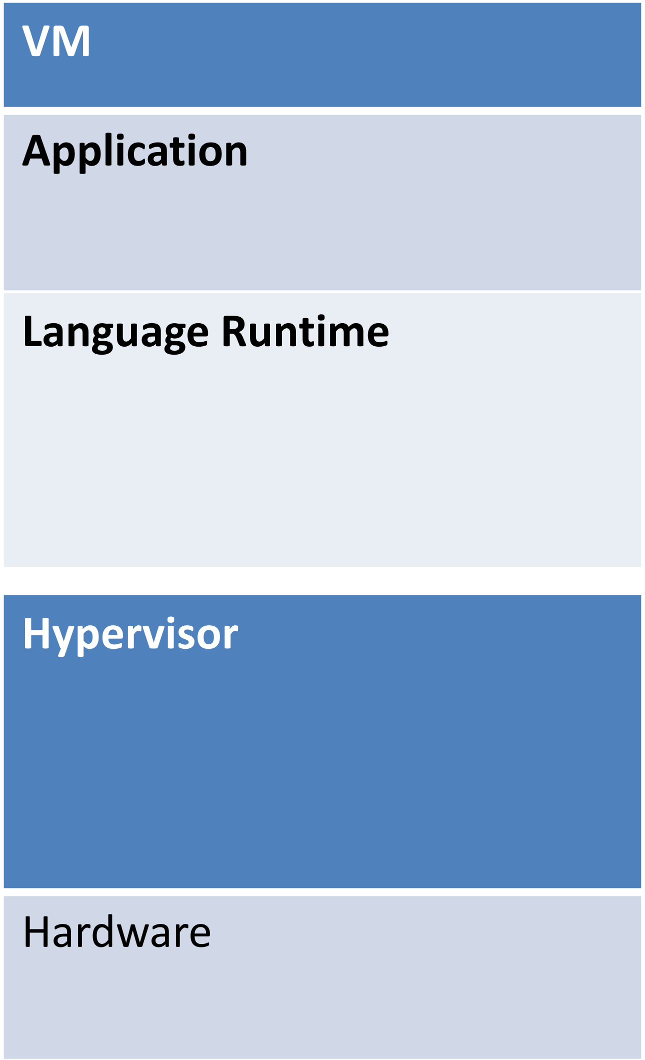
OS Kernel- RHL/Debian, Solaris, SUSE, Oracle Unix
Threads
Processes
Drivers

Hypervisor- thin layer of code in software or firmware to achieve fine-grained, dynamic resource sharing

Hardware

- Inside a file system, an independent process has been a virtualized box or VM
- Application is a Tomcat Web server **(HTTP Web server** environment in which Java code can run)
- Application can exist inside the language runtime. In this architecture, the language runtime will look to the kernel of the OS for the resources it needs for the app. These resources could include spawning threads (a function that loads and executes a new child process) or accessing native libraries, such as drivers (a common driver it would look for would be a database driver)
- In the same way runtime languages look to the OS's kernel for resources inside the VM, the VM looks for wider system resources which are made available by the hypervisor
- Just as a process will isolate the thread, a hypervisor isolates the VM
- The hypervisor is the master container, and the VM is a child container

Java OS and Advantages for the Cloud Architecture



VM

Application

Language Runtime

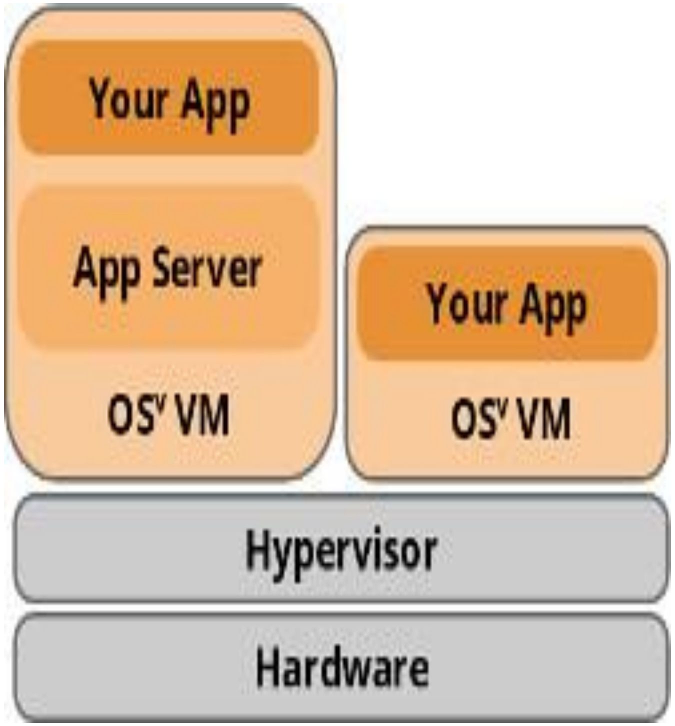
Hypervisor

Hardware

- Kernel layer has been removed
- Now, the language runtime is a Cloud OS
- Languages such as Java which create virtual machines can be considered as an OS (Java OS)
- Many other languages offer this functionality (C/C++ or Haskell)
- Application is sitting on a runtime right on top of the hypervisor
- In this schema, we are only running one application per VM
- The runtime and application can run in the same address space, simplifying things even more
- This approach is the future trend for the evolution
- The company that developed JavaOS for the cloud has partnered with Cloudius, which has attracted Salesforce and Google
- Simplifying the OS, Citrix makes one language runtime called Zen, and Cloudius make Os^{v}

OS^{v}

Running a single application in a VM



- OS^{v} is the modular unikernel designed to run Linux applications securely on micro-VMs in the cloud
- OS^{v} is built from the ground up for effortless deployment and management of micro-services and serverless applications, with superior performance
- http://osv.io/

Linux Security

- The kernel is a point of vulnerability in the system
- The VM, working with the hypervizor, can isolate application inside of the VM from the wire system
- In case of Web app, it’s possible the insertion of malicious resources in the kernel of the OS. Once malicious resources penetrate into the system, they can infest the kernel and multiply, then use it to mount attacks on the system
- Linux security is based on three characteristics : prevention, protection, and detection

Security Overview

- Linux runs by default in a lower privilege mode; a virus can’t access a lot of other parts of the system
- Linux security depends on users privileges and file permissions along with keeping the system up to date with the latest security patches
- Linux updates all software installed at once from repositories (big database or storage center) or Snappy packages (dependencies included); it doesn’t run programs in the back to listen for updates blocking system resources
- In Red Hat Enterprise 8- System-wide **cryptographic policies**, which configures the core cryptographic subsystems, covering the Transport Layer Security (TLS), Internet Protocol Security (Ipsec), Secure Shell (SSH), Domain Name System Security Extensions (DNSSEC), and Kerberos protocols, are applied by default. With the new update-crypto-policies command, the administrator can easily switch between modes: default, legacy, future, and fips (Federal Information Processing Standards). Support for **smart cards** and Hardware Security Modules (**HSM**) with **PKCS #11** (Cryptographic Token Interface Standard) is now consistent across the system

File Systems And Storage

- The **LUKS2** (Linux Unified Key Setup version 2) format replaces the legacy LUKS (LUKS1) format
- The dm-crypt disk encryption subsystem and the cryptsetup tool now uses LUKS2 as the default format for encrypted volumes

Support For Emerging Technologies

- Red Hat Enterprise Linux 8 is supported across architectures and environments, so you have a consistent and stable OS that adapts along the way to machine learning, predictive analytics, Internet of Things (IoT), edge computing, and big data workloads
- Red Hat Enterprise Linux 8 lowers the barrier to adoption of emerging technologies with support for hardware innovations like the graphics processing units (GPUs), which can assist machine learning workloads- GPU support for developing and running both traditional and containerized apps
- A graphics processing unit (GPU)- a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display
- https://computer.howstuffworks.com/graphics-card1.htm

Container Tools

- Red Hat Enterprise Linux 8 offers container tools that allow you to gear your systems to find, run, build, and share containers with other Open Container Initiative (OCI) standards-compatible tools
- Increased support for containerized applications
- **Podman** is a daemon-less (doesn’t run as a background process) command-line tool that lets you directly create and manage container images
- By letting you work with containers without the need for a runtime environment, you can more tightly control the permissions granted to software components

Dynamic Programming Languages, Web And Database Servers

- Python 3.6 is the default **Python** implementation in RHEL 8; limited support for Python 2.7 is provided. No version of Python is installed by default
- Node.js is new in RHEL. Other **dynamic programming languages** have been updated since RHEL 7: PHP 7.2, Ruby 2.5, Perl 5.26, SWIG 3.0 are now available
- The following **database servers** are distributed with RHEL 8: MariaDB 10.3, MySQL 8.0, PostgreSQL 10, PostgreSQL 9.6, and Redis 5
- RHEL 8 provides the Apache HTTP Server 2.4 and introduces a new **web server**, nginx 1.14
- Squid has been updated to version 4.4, and a new **proxy caching server** is now included: Varnish Cache 6.0

Virtualization

- A more modern PCI Express-based machine type (**Q35**) is now supported and automatically configured in virtual machines created in RHEL 8. This provides a variety of improvements in features and compatibility of virtual devices
- Virtual machines can now be created and managed using the RHEL 8 web console, also known as **Cockpit**
- The **QEMU** emulator introduces the **sandboxing** feature, which provides configurable limitations to what systems calls QEMU can perform, and thus makes virtual machines more secure

Streamline Processes

- Apps never stop, neither will the OS
- Non-Linux users using the web console, easily update apps using application streams, and stay focused on compliance and security with the built-in controls of Red Hat Enterprise Linux 8
- Use application streams to separate applications from the base OS so you can update apps without having to wait for the next major version of the operating system

A Firewall Interface Has Been Added To The Web Console

- The **Networking** page in the RHEL 8 web console now includes a **Firewall** section. In this section, users can enable or disable the firewall, as well as add, remove, and modify firewall rules
- **The web console** is now available by default
- Packages for the RHEL 8 web console, also known as Cockpit, are now part of Red Hat Enterprise Linux default repositories, and can therefore be immediately installed on a registered RHEL 8 system
- In addition, on a non-minimal installation of RHEL 8, the web console is automatically installed and firewall ports required by the console are automatically open
- **The web console is now compatible with mobile browsers-** With this update, the web console menus and pages can be navigated on mobile browser variants. This makes it possible to manage systems using the RHEL 8 web console from a mobile device.
- **Virtual Machines can now be managed using the web console-** The Virtual Machines page can now be added to the RHEL 8 web console interface, which enables the user to create and manage libvirt-based virtual machines
- **libvirt** is an open-source API, daemon and management tool for managing platform virtualization. It can be used to manage **KVM**, Xen, VMware ESXi, QEMU and other virtualization technologies. These APIs are widely used in the orchestration layer of hypervisors in the development of a cloud-**based** solution

Compilers And Development Tools

- The **GCC** compiler based on version 8.2 brings support for more recent C++ language standard versions, better optimizations, new code hardening techniques, improved warnings, and new hardware features
- Various tools for code generation, manipulation, and debugging can now experimentally handle the **DWARF5** debugging information format
- Kernel support for **eBPF** (extended Berkeley Packet Filter**)** tracing is available for some tools, such as Blind Carbon Copy (BCC), Performance Co-Pilot(PCP), and SystemTap (infrastructure to simplify the gathering of information about the running Linux kernel or userspace programs)
- The glibc libraries based on version 2.28 add support for Unicode 11, newer Linux system calls, key improvements in the DNS stub resolver, additional security hardening, and improved performance
- RHEL 8 provides OpenJDK 11, OpenJDK 8, IcedTea-Web, and various **Java** tools, such as Ant, Maven, or Scala

High Availability And Clusters

- The **Pacemaker** cluster resource manager has been upgraded to upstream version 2.0.0, which provides a number of bug fixes and enhancements
- In RHEL 8, the **pcs** (a command line tool to manage pacemaker) configuration system fully supports Corosync 3 Cluster Engine, knet (kernel network interface for efficient of packet exchange between switch and the kernel network protocol stack), and node names

Linux Kernel

- Think of Linux like the engine in a car, the car being the entire OS
- Red Hat Enterprise 8- The extended Berkeley Packet Filtering (**eBPF)** feature enables the user space to attach custom programs onto a variety of points (sockets, trace points, packet reception) to receive and process data. This feature is available as a **Technology Preview**
- Red Hat Enterprise 8- BPF Compiler Collection (**BCC**), a tool for creating efficient kernel tracing and manipulation programs, is available as a **Technology Preview**
- **Technology Preview** features are not fully supported, may not be functionally complete, and are not suitable for deployment in production

Ready Available OS

- Many versions, called distros that you can easily setup
- There are 100s available distribution designed specifically for multiple user sharing, virtualization, etc, all based on a Linux kernel, just with different tool sets packaged into them and different default settings already setup. Most of these were originally made by taking a more common distribution, adding only the needed tools and making presets, then saving it as a new install image for ease of future installation on new hardware
- **Fedora**
  - Specific for running firewall (a server version of Ubuntu works there)
  - Distros (distributors) for network attached storage (NAS), a backup server
  - Web server
  - Fedora makes an old machines with little HD (1Gb) came to life & there is many themes
- Based on Fedora 28 and the upstream kernel 4.18, Red Hat Enterprise Linux 8.0 provides users with a stable, secure, consistent foundation across hybrid cloud deployments with the tools needed to support traditional and emerging workloads

Common Administrative Commands In Red Hat Enterprise Linux 8

Task	Command
View system profile	cat /proc/cpuinfo
Synchronize time and date	/etc/chrony.conf
Configure SSH	/etc/ssh/ssh_config
List all services	find /etc/systemd/ /usr/lib/systemd/
Start/stop service	systemctl start/stop *name*.service
View logs	/var/log
Schedule/batch tasks	cron
Configure GRUB bootloader	/etc/default/grub
View hardware configured	lshw
View kernel version	rpm -q kernel

Software Management

- The **Yellowdog Updater, Modified** (**YUM**) package manager is now based on the **Dandified YUM** (**DNF**) technology and it provides support for modular content, increased performance, and a well-designed stable API for integration with tooling

Task	Command
Install a module	yum module install module_name
View info on a module	yum module info module_name
View a module's streams	yum module info module_name
Change module streams	yum module remove module_name:stream yum module reset module:stream yum module install module:new_stream
List available modules	yum module list

Networking

- The nftables framework replaces iptables in the role of the default network packet filtering facility
- The firewalld daemon now uses nftables as its default backend
- Support for **IPVLAN** virtual network drivers that enable the network connectivity for multiple containers has been introduced
- The eXpress Data Path (**XDP**), XDP for Traffic Control (**tc**), and Address Family eXpress Data Path (**AF_XDP**), as parts of the extended Berkeley Packet Filtering (**eBPF)** feature, are available as **Technology Preview** (not fully supported, may not be functionally complete, and are not suitable for deployment in production)

Networking Commands In Red Hat Enterprise Linux 8 (nftables framework)

Task	Command
Configure name resolution	/etc/hosts /etc/resolv.conf nmcli con mod
View network interface info	nmcli dev show (not ifconfig anymore) ip addr teamdctl brctl bridge
Configure network interface	/etc/sysconfig/network-scripts/ifcfg-* nmcli con [add|mod|edit] nmtui nm-connection-editor
Configure routes	ip route add nmcli nmtui nm-connection-editor /etc/sysconfig/route-*iface*
Configure firewall	firewall-cmd firewall-config nftables
View ports/sockets	ss lsof netstat pcp-dstat --socket

Ease of Installing Programs

- Linux software is installed in predetermined directories and can not be moved or installed elsewhere /dev, /user/bin, /user/applications and runs on the system
- Linux used /dev for drivers and treats all storage devices and network storage as directories along with virtual devices. Storage devices can be mounted anywhere within the file structure "mount /dev/sdc1 /mnt"

Popularity

- Three quarters of the Internet run on Linux
- Even Windows uses Linux to run the networking of it’s cloud offerings https://www.wired.com/2015/09/microsoft-using-linux-run-cloud/. It turned to Linux to build its own switch software
- The Space X Falcon 9 rocket (a two-stage-to-orbit medium lift launch vehicle) and the International Space Station (habitable artificial satellite in low Earth orbit) both use Linux
- Also, VMware products included a Linux kernel to start with, vmkernel. The Linux kernel was the primary virtual machine, invoked by the service console (the vmkernel was running on the bare computer, and the Linux-based service console ran as the first virtual machine). VMware dropped development of ESX at version 4.1, and now uses ESXi, which does not include a Linux kernel

Reference List

- https://access.redhat.com/
- http://www.eduonix.com/courses
- https://computer.howstuffworks.com/graphics-card1.htm
- http://osv.io/
- https://blog.containership.io/k8svsdocker/