Red Hat

Welcome to Section 4

Redhat System Administration II (RH134)

By: Imran Afzal
www.utclisolutions.com

Improve Command Line Productivity

Linux shell scripting is a strong tool for systems administrator to automate daily repetitive tasks

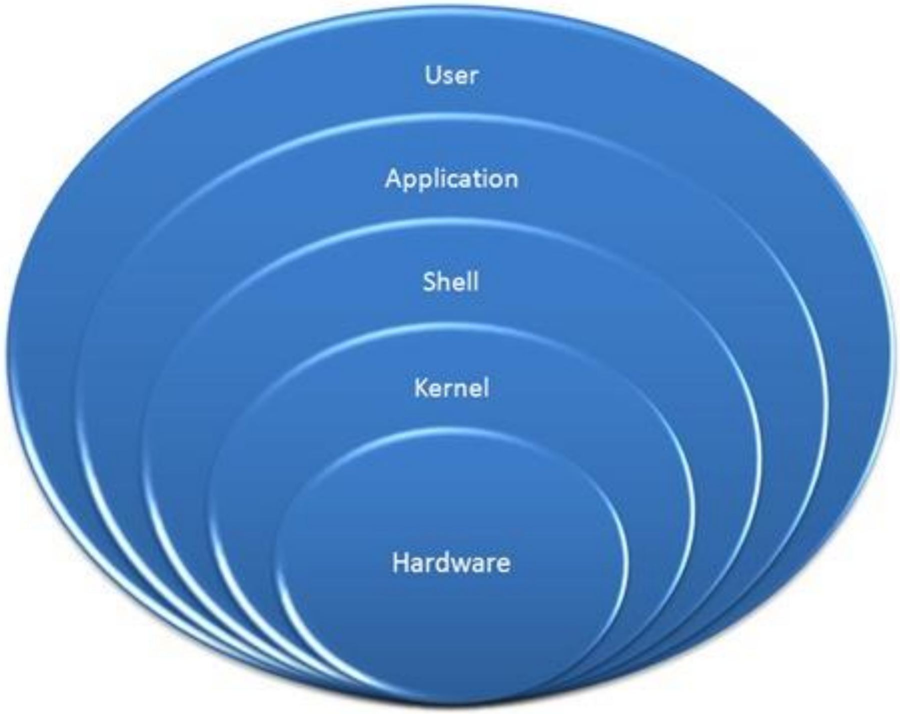
We will learn..

- Introduction to shell
- Shell scripting
- Basic shell scripts
- Input and Output of scripts
- If-then scripts
- for loop scripts

Introduction to Shell

- What is a Shell?
  - Its like a container
  - Interface between users and Kernel/OS
  - CLI is a Shell

- Find your Shell
  - **`echo $0`**
  - Available Shells "**`cat /etc/shells`**"
  - Your Shell? **`/etc/passwd`**



- Windows GUI is a shell
- Linux KDE GUI is a shell
- Linux sh, bash etc. is a shell

Improve Command Line Productivity

Shell Scripting

- What is a Shell Script?

  A shell script is an executable file containing multiple shell commands that are executed sequentially. The file can contain:

  - Shell (**`#!/bin/bash`**)
  - Comments (**`# comments`**)
  - Commands (**`echo, cp, grep`** **etc.**)
  - Statements (**`if, while, for`** etc.)

- Shell script should have executable permissions (e.g. **`-rwx r-x r-x`**)
- Shell script has to be called from absolute path (e.g **`/home/userdir/script.bash`**)
- If called from current location then **`./script.bash`**

By: Imran Afzal
www.utclisolutions.com

Improve Command Line Productivity

Shell Script – Basic Scripts

- Output to screen using "echo"

- Creating tasks
  - Telling your id, current location, your files/directories, system info
  - Creating files or directories
  - Output to a file ">"

- Filters/Text processors through scripts (**`cut, awk, grep, sort, uniq, wc`**)

Improve Command Line Productivity

if-then Scripts

- If then statement

```
If this happens      = do this
Otherwise            = do that
```

Improve Command Line Productivity

For Loop Scripts

- For loops

```
Keep running until specified number of variable
e.g: variable = 10 then run the script 10 times
OR
     variable = green, blue, red (then run the
     script 3 times for each color.
```

Improve Command Line Productivity

`grep/egrep` - Text Processors Commands

- What is grep?
  - The grep command which stands for "global regular expression print," processes text line by line and prints any lines which match a specified pattern

`grep --version OR grep --help`	=	Check version or help
`grep keyword file`	=	Search for a keyword from a file
`grep -c keyword file`	=	Search for a keyword and count
`grep -i KEYword file`	=	Search for a keyword ignore case-sensitive
`grep -n keyword file`	=	Display the matched lines and their line numbers
`grep -v keyword file`	=	Display everything but keyword
`grep keyword file | awk '{print $1}'`	=	Search for a keyword and then only give the 1st field
`ls -l | grep Desktop`	=	Search for a keyword and then only give the 1st field
`egrep -i "keyword|keyword2" file`	=	Search for 2 keywords.

By: Imran Afzal
www.utclisolutions.com

Schedule future tasks (Crontab and at)

In this lesson we will learn…

- How to schedule future tasks through **`crontab`**
- How to schedule one time ad-hoc tasks through **`at`** command

Tune System Performance

Linux system comes fined tunned by default when you install, however there are a few tweaks that can be done based on system performance and application requirements

In this lesson we will learn…

- Optimize system performance by selecting a tuning profile managed by the **`tuned`** daemon
- Prioritize or de-prioritize specific processes with the **`nice`** and **`renice`** commands

Tune System Performance

What is tuned?

- Tuned pronounced as tune-d
- **`Tune`** is for system tuning and **`d`** is for daemon
- It is **`systemd`** service that is used to tune Linux system performance
- It is installed in CentOS/Redhat version 7 and 8 by default
- **`tuned`** package name is **`tuned`**
- The **`tuned`** service comes with pre-defined profiles and settings *(List of profile will be discussed in the next page)*
- Based on selected profile the **`tuned`** service automictically adjust system to get the best performance. E.g. **`tuned`** will adjust networking if you are downloading a large file or it will adjust IO settings if it detects high storage read/write
- The tuned daemon applies system settings when the service starts or upon selection of a new tuning profile.

Tune System Performance

(`tuned` profiles)

Tuned profile	Purpose
balanced	deal for systems that require a compromise between power saving and performance
desktop	Derived from the balanced profile. Provides faster response of interactive applications
Throughput-performance	Tunes the system for maximum throughput
Latency-performance	Ideal for server systems that require low latency at the expense of power consumption
network-latency	Derived from the latency-performance profile. It enables additional network tuning parameters to provide low network latency
Network-throughput	Derived from the throughput-performance profile. Additional network tuning parameters are applied for maximum network throughput
powersave	Tunes the system for maximum power saving
oracle	Optimized for Oracle database loads based on the throughput-performance profile
virtual-guest	Tunes the system for maximum performance if it runs on a virtual machine
virtual-host	Tunes the system for maximum performance if it acts as a host for virtual machines

Tune System Performance

- Check if tuned package has been installed
  ```
  rpm -qa | grep tuned
  ```
- Install tuned package if NOT installed already
  ```
  yum install tuned
  ```
- Check **tuned** service status
  ```
  systemctl status|enable|start tuned
  systemctl enable tuned
  ```
  (To enable at boot time)
- Command to change setting for tuned daemon
  ```
  tuned-adm
  ```
- To check which profile is active
  ```
  tuned-adm active
  ```
- To list available profiles
  ```
  tuned-adm list.
  ```

Tune System Performance

- To change to desired profile
  ```
  tuned-adm profile profile-name
  ```

- Check for tuned recommendation
  ```
  tuned-adm recommend
  ```

- Turn off tuned setting daemon
  ```
  tuned-adm off
  ```

- Change profile through web console
  ```
  Login to https://192.168.1.x:9090
  Overview → Configuration → Performance profile
  ```

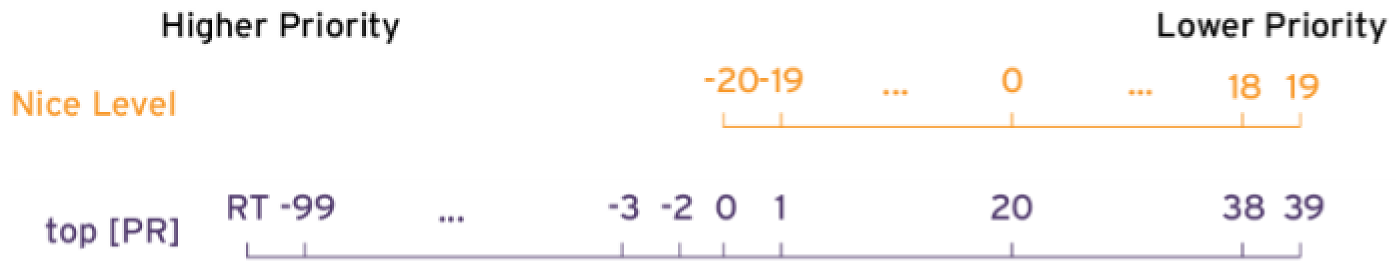
Tune System Performance

- Another way of keeping your system fine-tuned is by prioritizing processes through **`nice`** and **`renice`** command
- If a server has 1 CPU then it can execute **1** computation/process at a time as they come in *(first come first served)* while other processes must wait
- With **`nice`** and **`renice`** commands we can make the system to give preference to certain processes than others
- This priority can be set at 40 different levels
- The nice level values range from -20 (highest priority) to 19 (lowest priority) and by default, processes inherit their nice level from their parent, which is usually 0.

Tune System Performance

- To check process priority

```
top
```



Nice value is a user-space and priority PR is the process's actual priority that use by Linux kernel. In Linux system priorities are 0 to 139 in which 0 to 99 for real time and 100 to 139 for users

- Process priority can be viewed through ps command as well with the right options

```
$ ps axo pid,comm,nice,cls --sort=-nice
```

Tune System Performance (nice/renince)

- To set the process priority

```
nice -n # process-name
e.g. nice -n -15 top
```

- To change the process priority

```
renice -n # process-name
e.g. renice -n 12 PID.
```

Controlling access to files with ACL

What is ACL?

- Access control list (ACL) provides an additional, more flexible permission mechanism for file systems. It is designed to assist with UNIX file permissions. ACL allows you to give permissions for any user or group to any disc resource.

Use of ACL :

- Think of a scenario in which a particular user is not a member of group created by you but still you want to give some read or write access, how can you do it without making user a member of group, here comes in picture Access Control Lists, ACL helps us to do this trick

- Basically, ACLs are used to make a flexible permission mechanism in Linux.

- From Linux man pages, ACLs are used to define more fine-grained discretionary access rights for files and directories.

- Commands to assign and remove ACL permissions are:

  `setfacl` and **`getfacl`**

Controlling access to files with ACL

Access Control List (ACL)

List of commands for setting up ACL :

1) To add permission for user

```
setfacl -m u:user:rwx /path/to/file
```

2) To add permissions for a group

```
setfacl -m g:group:rw /path/to/file
```

3) To allow all files or directories to inherit ACL entries from the directory it is within

```
setfacl -dm "entry" /path/to/dir
```

4) To remove a specific entry

```
setfacl -x u:user /path/to/file
```
(For a specific user)

5) To remove all entries

```
setfacl -b path/to/file
```
(For all users)

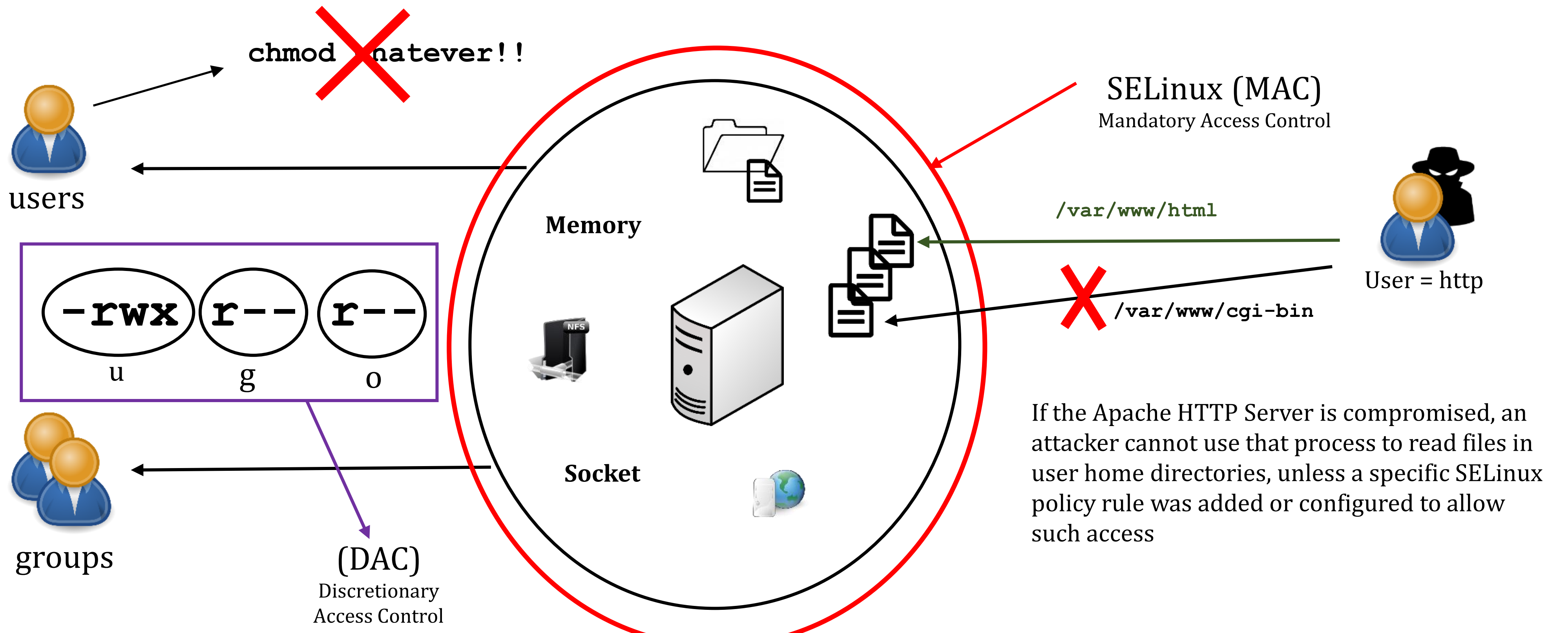
Note:

- As you assign the ACL permission to a file/directory it adds + sign at the end of the permission
- Setting w permission with ACL does not allow to remove a file

By: Imran Afzal
www.utclisolutions.com

Managing SELinux Security

- What is SELinux?
  - Security-Enhanced Linux is a Linux kernel security module that provides a mechanism for supporting access control security policies, including mandatory access controls. (*Wikipedia*)
  - It is a project of the United States National Security Agency (NSA) and the SELinux community

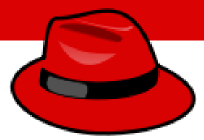


If the Apache HTTP Server is compromised, an attacker cannot use that process to read files in user home directories, unless a specific SELinux policy rule was added or configured to allow such access

By: Imran Afzal
www.utclisolutions.com

Managing SELinux Security

SELinux *(Security Enhanced Linux)*



- SELinux options?
  - Enforcing = Enabled *(enabled by default in Redhat, CentOS and Fedora)*
  - Permissive = Disabled but logs the activity
  - Disable = Disabled and not activity logs

- To check SELinux status
  - `# sestatus OR getenforce`

- SELinux setting
  - `# setenforce 0      =        Permissive/Disable`
  - `# setenforce 1      =        Enable`

- Modify SELinux config for permanent setting
  - `/etc/selinux/config`

```
SELINUX=enforcing
SELINUX=disabled
```

- Before modifying selinux config file
  - **`Create a snapshot of your VM`**

- Before rebooting create a file

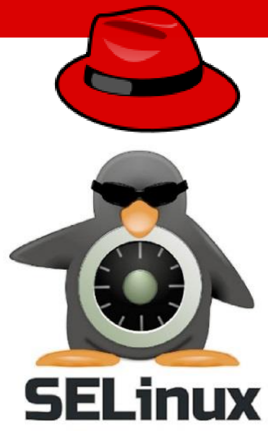
```
    /.autorelabel
```

By: Imran Afzal
www.utclisolutions.com

Managing SELinux Security

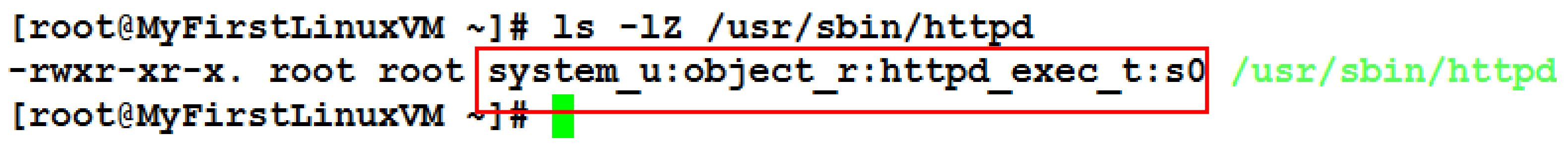
SELinux *(Security Enhanced Linux)*

User:role:
type:level



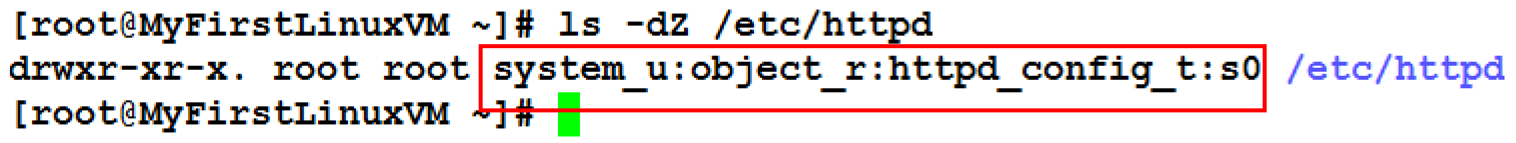
- Two main concepts of SELinux
  - Labeling
  - Type enforcement

- To list the label of a directory
  - `ls -lZ /usr/sbin/httpd`



```
[root@MyFirstLinuxVM ~]# ls -lZ /usr/sbin/httpd
-rwxr-xr-x. root root system_u:object_r:httpd_exec_t:s0 /usr/sbin/httpd
[root@MyFirstLinuxVM ~]#
```

- To list the label of a directory
  - `ls -dZ /etc/httpd`



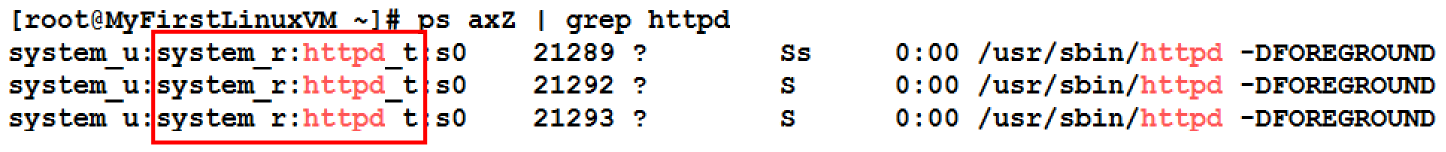
```
[root@MyFirstLinuxVM ~]# ls -dZ /etc/httpd
drwxr-xr-x. root root system_u:object_r:httpd_config_t:s0 /etc/httpd
[root@MyFirstLinuxVM ~]#
```

By: Imran Afzal
www.utclisolutions.com

Managing SELinux Security

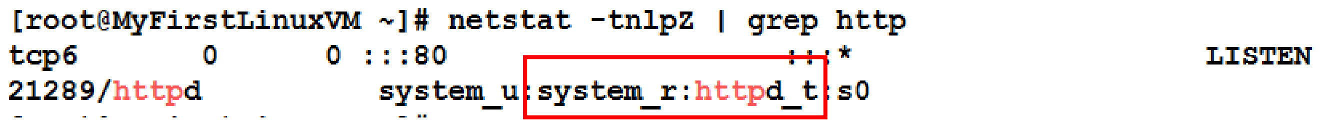
SELinux *(Security Enhanced Linux)*

- As the webserver runs its process is labeled in memory as **`httpd_t`**:
  - **`ps axZ | grep httpd`**



```
[root@MyFirstLinuxVM ~]# ps axZ | grep httpd
system_u:system_r:httpd_t:s0    21289 ?        Ss     0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    21292 ?        S      0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    21293 ?        S      0:00 /usr/sbin/httpd -DFOREGROUND
```

- The SELinux assigns the label at the socket level
  - **`netstat -tnlpZ | grep http`**



```
[root@MyFirstLinuxVM ~]# netstat -tnlpZ | grep http
tcp6       0      0 :::80                   :::*                    LISTEN
21289/httpd          system_u:system_r:httpd_t:s0
```

By: Imran Afzal
www.utclisolutions.com

Managing SELinux Security

SELinux *(Security Enhanced Linux)*

- Command to manage SELinux setting:
  - **`semanage` → to label**
    - **`login`**
    - **`user`**
    - **`port`**
    - **`interface`**
    - **`module`**
    - **`node`**
    - **`file context`**
    - **`boolean`**
    - **`permissive state`**
    - **`dontaudit`**

By: Imran Afzal
www.utclisolutions.com

Manage Basic Storage

Storage

- Local Storage
- SAN (Storage Area Network)
- NAS (Network Attached Storage)

Manage Basic Storage

Disk Partition

- Commands for disk partition
  - `df`
  - `fdisk`

Manage Basic Storage

Adding Disk and Creating Partition

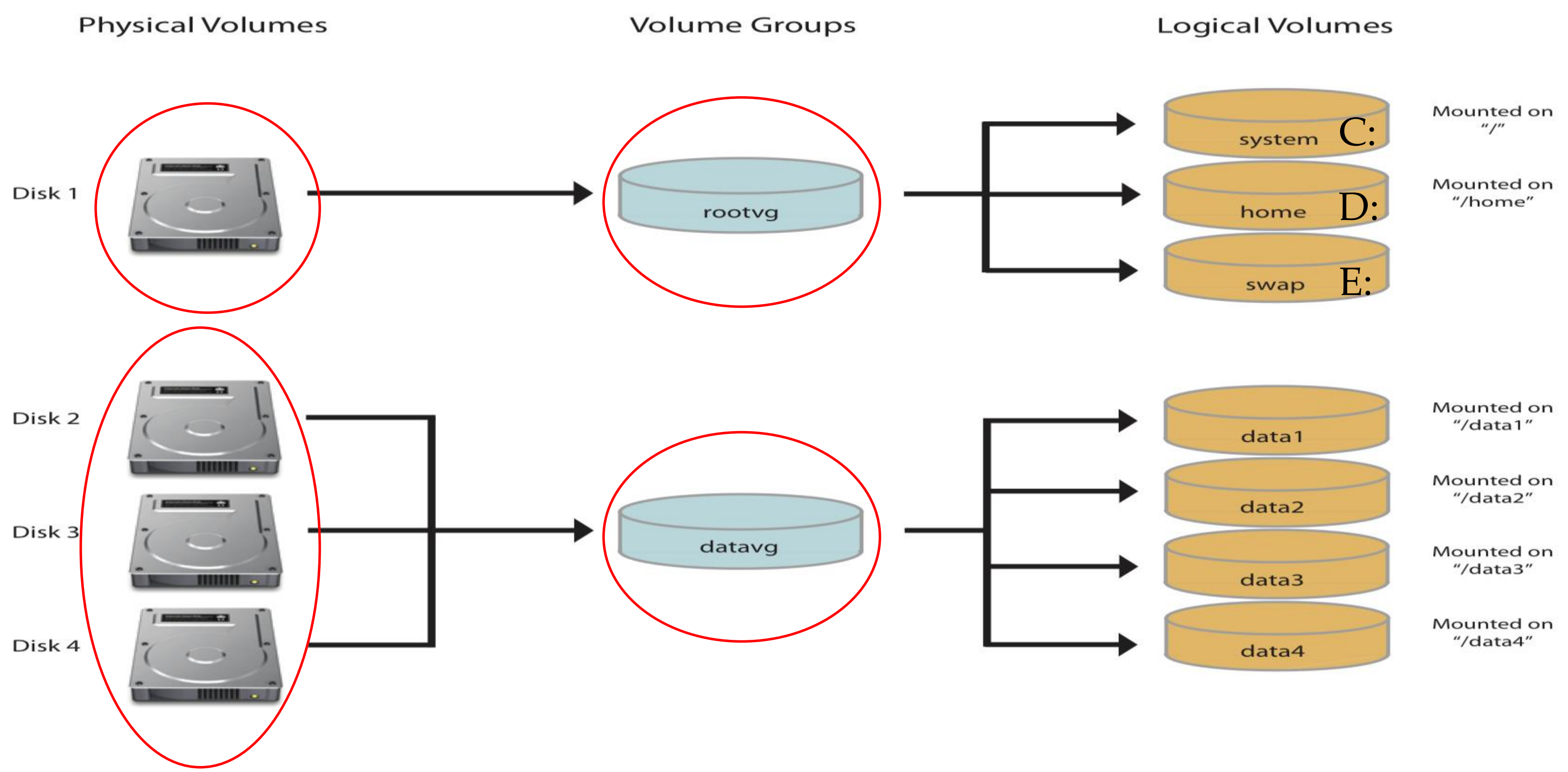
- Purpose? = Out of Space, Additional Apps etc.

- Commands for disk partition
  - **`df`**
  - **`fdisk`**

Manage Logical Volumes

Logical Volume Management (LVM)

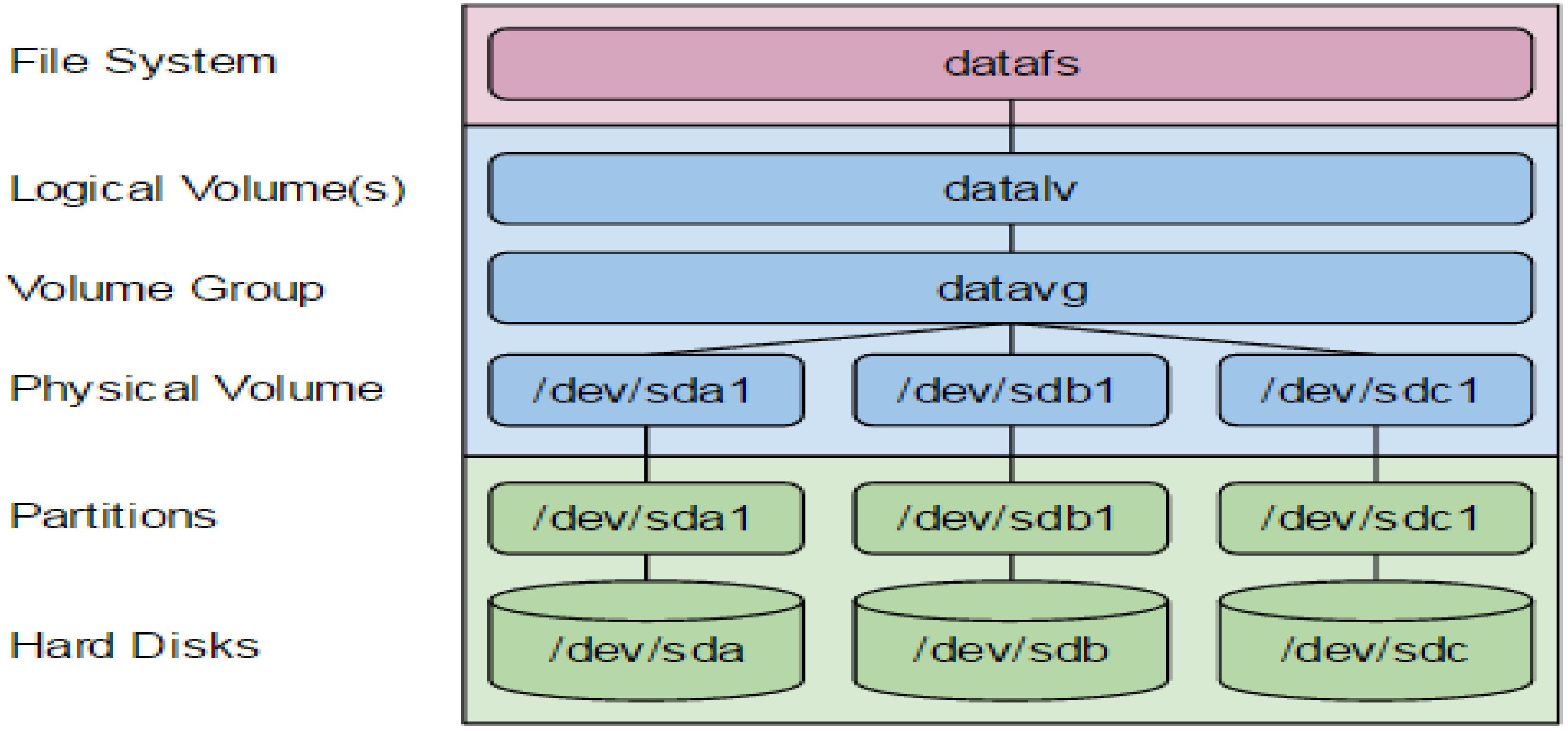
- LVM allows disks to be combined



By: Imran Afzal
www.utclisolutions.com

Manage Logical Volumes

Add Disk and Create LVM Partition



By: Imran Afzal
www.utclisolutions.com

Manage Logical Volumes

Add and Extend Disk using LVM

/oracle = 1.0G
/oracle = Full

Few Options:

- Delete older files to free up disk space
- Add new physical disk mount to /oracle2
- Create a new virtual disk and mount to /oracle2
- Or extend /oracle through LVM.

Manage Logical Volumes

Implement Advanced Storage Features

- Red Hat 8 introduces the next generation volume management solution called Stratis
- It uses thin provisioning by default
- It combines the process of creating logical volume management (LVM) and creation of filesystems into one management
- In LVM if a filesystem system gets full you will have to extend it manually whereas stratis extends the filesystem automatically if it has available space in its pool

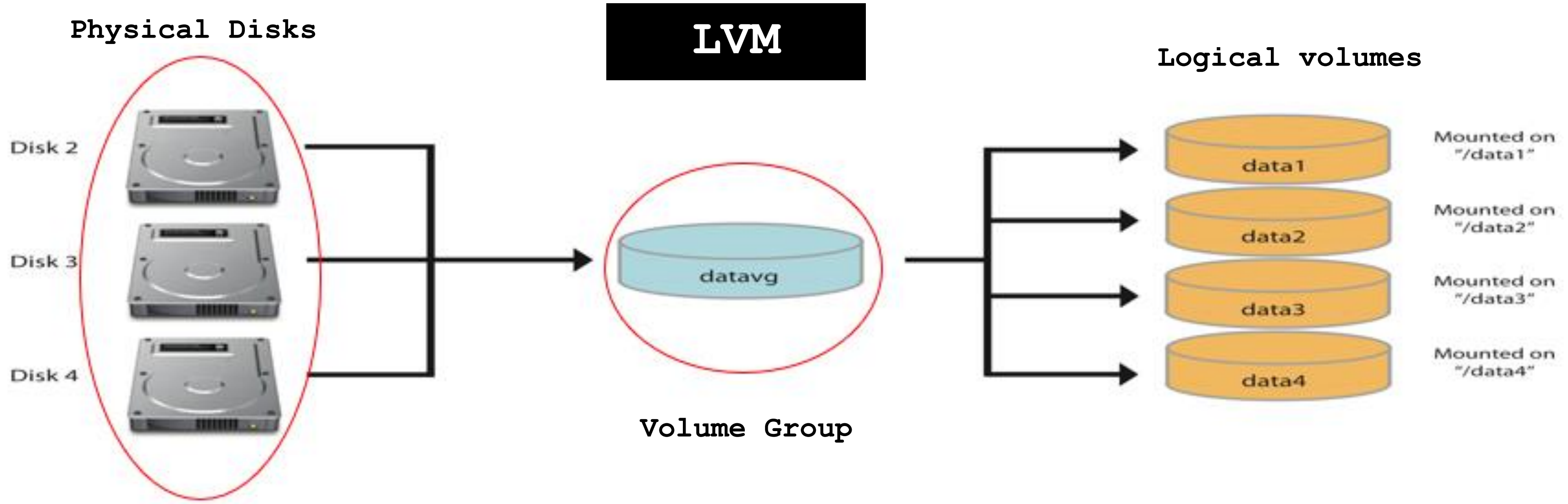
In this lesson we will learn…

- How to manage multiple storage layers using Stratis local storage management

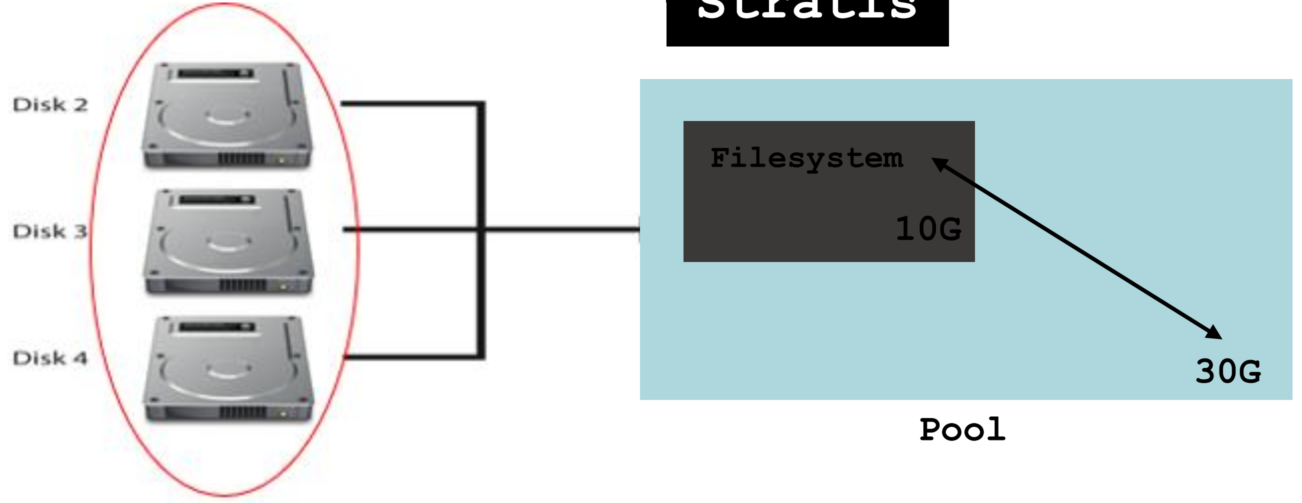
Manage Logical Volumes

Implement Advanced Storage Features

LVM



Stratis



By: Imran Afzal
www.utclisolutions.com

Implement Advanced Storage Features with Stratis

- Install Statris package
  ```
  yum/dnf install stratis-cli stratisd
  ```

- Enable and start Statris service
  ```
  systemctl enable|start stratisd
  ```

- Add 2 x 5G new disks from virtualization software and verify at the OS level
  ```
  Oracle virtualbox storage setting
  lsblk
  ```

- Create a new stratis pool and verify
  ```
  stratis pool create pool1 /dev/sdb
  stratis pool list
  ```

- Extend the pool
  ```
  stratis pool add-data pool1 /dev/sdc
  stratis pool list
  ```

Implement Advanced Storage Features with Stratis

- Create a new filesystem using stratis

```
stratis filesystem create pool1 fs1
stratis filesystem list
```

(Filesystem will start with 546 MB)

- Create a directory for mount point and mount filesystem

```
mkdir /bigdata
mount /dev/stratis/pool1/fs1 /bigdata
lsblk
```

- Create a snapshot of your filesystem

```
startis filesystem snapshot pool1 fs1 fs1-snap
stratis filesystem list
```

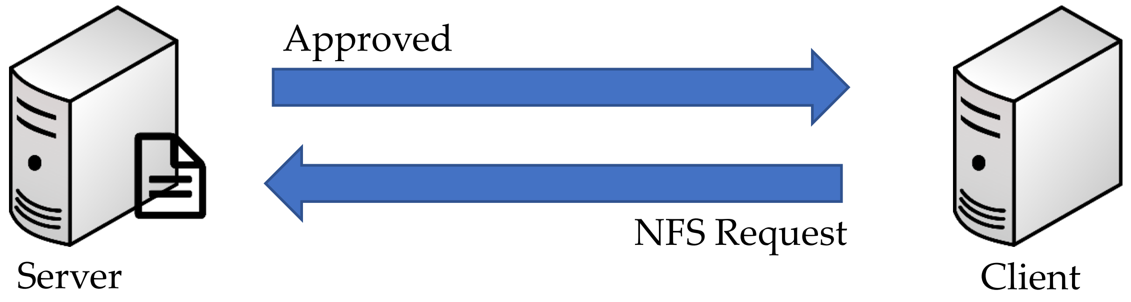
- Add the entry to /etc/fstab to mount at boot

```
UUID=“asf-0887afgdja-” /bigdata xfs defaults,x-
systemd.requires=stratisd.service 0 0
```

Access Network-Attached Storage

Network File System (NFS)

- NFS stands for Network File System, a file system developed by Sun Microsystems, Inc.
- It is a client/server system that allows users to access files across a network and treat them as if they resided in a local file directory
- For example, if you were using a computer linked to a second computer via NFS, you could access files on the second computer as if they resided in a directory on the first computer. This is accomplished through the processes of exporting (the process by which an NFS server provides remote clients with access to its files) and mounting (the process by which client map NFS shared filesystem)



By: Imran Afzal
www.utclisolutions.com

Access Network-Attached Storage

Network File System (NFS)

Steps for NFS Server Configuration

- Install NFS packages

```
# yum install nfs-utils libnfsidmap    (most likely they are installed)
```

- Once the packages are installed, enable and start NFS services

```
# systemctl enable rpcbind
# systemctl enable nfs-server
# systemctl start rpcbind, nfs-server, rpc-statd, nfs-idmapd
```

- Create NFS share directory and assign permissions

```
# mkdir /mypretzels
# chmod a+rwx /mypretzels
```

- Modify **/etc/exports** file to add new shared filesystem

```
#    /mypretzels 192.168.12.7(rw,sync,no_root_squash) = for only 1 host

# /mypretzels *(rw,sync,no_root_squash) = for everyone
```

- Export the NFS filesystem

```
# exportfs -rv
```

NFS share

IP address of client machine

Read/write

all changes to the according filesystem are immediately flushed to disk; the respective write operations are being waited for

root on the client machine will have the same level of access to the files on the system as root on the server.

By: Imran Afzal
www.utclisolutions.com

Access Network-Attached Storage

Network File System (NFS)

Steps for NFS Client Configuration

- Install NFS packages

```
# yum install nfs-utils rpcbind
```

- Once the packages are installed enable and start rpcbind service

```
# systemctl rpcbind start
```

- Make sure firewalld or iptables stopped (if running)

```
# ps –ef | egrep "firewall|iptable"
```

- Show mount from the NFS server

```
# showmount -e 192.168.1.5 (NFS Server IP)
```

- Create a mount point

```
# mkdir /mnt/kramer
```

- Mount the NFS filesystem

```
# mount 192.168.1.5:/mypretzels /mnt/kramer
```

- Verify mounted filesystem

```
# df –h
```

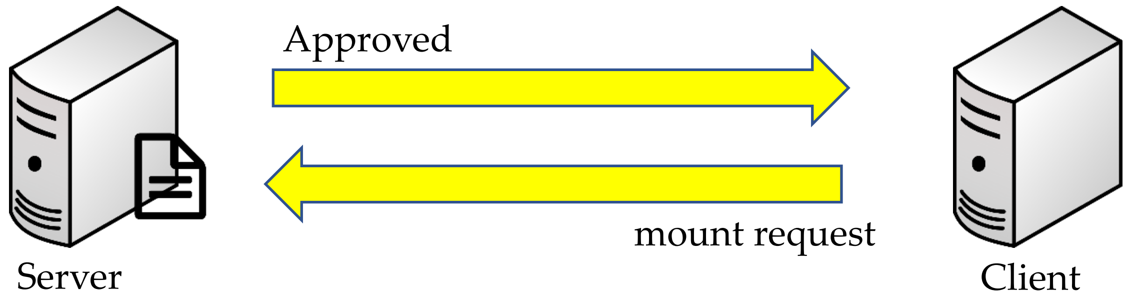
- To unmount

```
# umount /mnt/kramer
```

Access Network-Attached Storage

Samba

- Samba is a Linux tool or utility that allows sharing for Linux resources such as files and printers to with other operating systems
- It works exactly like NFS but the difference is NFS shares within Linux or Unix like system whereas Samba shares with other OS (e.g. Windows, MAC etc.)
- For example, computer "A" shares its filesystem with computer "B" using Samba then computer "B" will see that shared filesystem as if it is mounted as the local filesystem



Access Network-Attached Storage

Samba (smb vs. CIFS

- Samba shares its filesystem through a protocol called **SMB** (Server Message Block) which was invented by IBM
- Another protocol used to share Samba is through **CIFS** (Common Internet File System) invented by Microsoft and NMB (NetBios Named Server)
- **CIFS** became the extension of **SMB** and now Microsoft has introduced newer version of SMB v2 and v3 that are mostly used in the industry
- In simple term, most people, when they use either SMB or CIFS, are talking about the same exact thing

Access Network-Attached Storage

Samba Installation and Configuration

- Take snapshot of your VM
- Install samba packages
- Enable samba to be allowed through firewall (Only if you have firewall running)
- Disable firewall
- Create Samba share directory and assign permissions
- Also change the SELinux security context for the samba shared directory
- Or disable SELinux
- Modify **/etc/samba/smb.conf** file to add new shared filesystem
- Verify the setting
- Once the packages are installed, enable and start **Samba** services (smb and nmb)
- Mount Samba share on Windows client
- Mount Samba share on Linux client
- Additional instructions on creating secure Samba share.

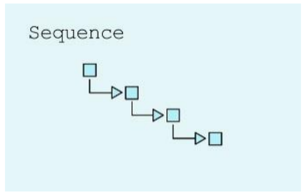
Control Boot Process

In this lesson we will learn…

- The Linux boot process, set the default target used when booting, and boot a system to a non-default target
- Recover root password
- Repair file system configuration or corruption issues

Control Boot Process

Linux Boot Process (Newer Versions)



- The boot sequence changes in CentOS/Redhat 7 and above
- **systemd** is the new service manager in CentOS/RHEL 7 that manages the boot sequence
- It is backward compatible with SysV init scripts used by previous versions of RedHat Linux including RHEL 6
- Every system administrator needs to understand the boot process of an OS in order to troubleshoot effectively

Control Boot Process

Linux Boot Process (Newer Versions)

BIOS = Basic Input and Output Setting (firmware interface)
POST = Power-On Self-Test started

MBR = Master Boot Record
Information saved in the first sector of a hard disk that indicates where the GRUB2 is located so it can be loaded in computer RAM

GRUB2 = Grand Unified Boot Loader v2
Loads Linux kernel
/boot/grub2/grub.cfg

Kernel = Core of Operating System
Loads required drivers from initrd.img
Starts the first OS process (systemd)

Systemd = System Daemon (PID # 1)
It then starts all the required processes
Reads = /etc/systemd/system/default.target to bring the system to the run-level
Total of 7 run-levels (0 thru 6)

By: Imran Afzal
www.utclisolutions.com

Control Boot Process

How to Reboot/Shutdown

- To power off or reboot a system from the command line, you can use the **`systemctl`** command

- **`systemctl poweroff =`** stops all running services, unmounts all file systems, and then powers down the system
- **`systemctl reboot =`** stops all running services, unmounts all file systems, and then reboots the system

- You can also use the shorter version of these commands, **`shutdown`**, **`poweroff`** and **`reboot`**, which are symbolic links to their **`systemctl`** equivalents

Control Boot Process

Selecting a `Systemd` Target

- Systemd is the first Linux process which decides at which run-level the operating system needs to be in
- These run-levels are now referred to as targets
- The following table lists the most important targets

Target	Purpose
graphical.target	System supports multiple users, graphical and text based logins
multi-user.target	System supports multiple user, text-based logins only
rescue.target	sulogin prompt, bashic system initialization completed
emergency.target	sulogin prompt, initramfs pivot complete and system root mounted on / read only

Control Boot Process

Selecting a **Systemd** Target

- To check the current target or run-level
  - **`systemctl get.default`**
  - **`who -r`**

- A target can be a part of another target. e.g., the **`graphical.target`** includes **`multi-user.target`** , which in turn depends on basic.target and others

- You can view these dependencies with the following command
  - **`systemctl list-dependencies graphical.target | grep target`**

- You can display the new runlevels/targets by issuing the following command:
  - **`ls -al /lib/systemd/system/runlevel*`**

- Setting default target
  - **`systemctl set-default graphical.target`**

Control Boot Process

Recover root Password

- Restart your computer
- Edit grub
- Change password
- reboot

By: Imran Afzal
www.utclisolutions.com

Control Boot Process

Repair Filesystem Corruption

- Filesystem corruption can occur when you either make mistakes in **`/etc`** configuration files or filesystem become corrupted at the disk level
- In either of the above cases the **`systemd`** will not be able to boot the system in the defined target and bring the system in emergency mode
- The following table lists some common errors and their results

Problem	**Result**
Corrupt file system	systemd attempts to repair the file system. If the problem is too severe for an automatic fix, the system drops the user to an emergency shell
Nonexistent device or UUID referenced in /etc/fstab	systemd waits for a set amount of time, waiting for the device to become available. If the device does not become available, the system drops the user to an emergency shell after the timeout
Nonexistent mount point in /etc/fstab	The system drops the user to an emergency shell
ncorrect mount option specified in /etc/fstab	The system drops the user to an emergency shell

Control Boot Process

Repair Filesystem Corruption

- In any case administrators can use the emergency target to diagnose and fix the issue, because no file systems are mounted before the emergency shell is displayed

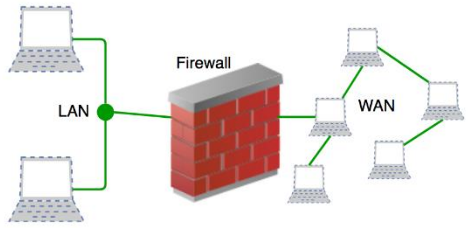
- When using the emergency shell to fix filesystem issues, do not forget to run **`systemctl daemon-reload`** after editing **`/etc/fstab`** . Without this reload, **`systemd`** may continue using the old version.

Manage Network Security (Firewall)

Introduction to Firewall

- What is Firewall
  - A wall that prevents the spread of fire
  - When data moves in and out of a server its packet information is tested against the firewall rules to see if it should be allowed or not
  - In simple words, a firewall is like a watchman, a bouncer, or a shield that has a set of rules given and based on that rule they decide who can enter and leave
  - There are 2 type of firewalls in IT

• Software	=	Runs on operating system
• Hardware	=	A dedicated appliance with firewall software



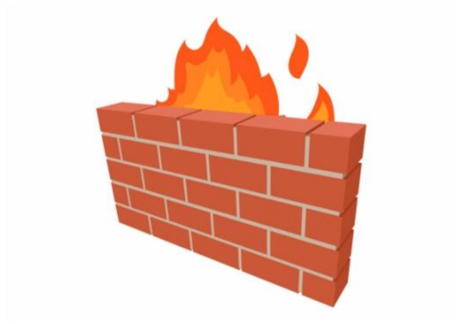
By: Imran Afzal
www.utclisolutions.com

Firewall *(firewalld)*

- Firewalld works the same way as iptables but of course it has it own commands
  - **`firewall-cmd`**

- It has a few pre-defined service rules that are very easy to turn on and off
  - **`Services such as: NFS, NTP, HTTPD etc.`**

- Firewalld also has the following:
  - **`Table`**
  - **`Chains`**
  - **`Rules`**
  - **`Targets`**



Firewall *(firewalld)*

- You can run one or the other
  - iptables or firewalld

- Make sure iptables is stopped, disabled and mask
  - **`systemctl stop iptables`**
  - **`systemctl disable iptables`**
  - **`systemctl mask iptables`**

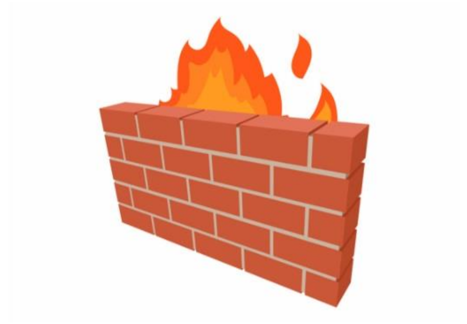
- Now check if filewalld package is installed
  - **`rpm –qa | grep firewalld`**

- Start firewalld
  - **`systemctl start/enable firewalld`**

- Check the rule of firewalld
  - **`firewall-cmd --list-all`**

- Get the listing of all services firewalld is aware of:
  - **`firewall-cmd --get-services`**

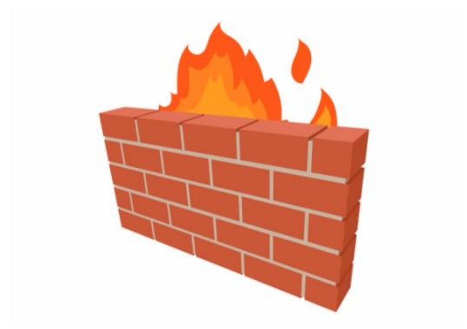
- To make firewalld re-read the configuration added
  - **`firewall-cmd --reload`**

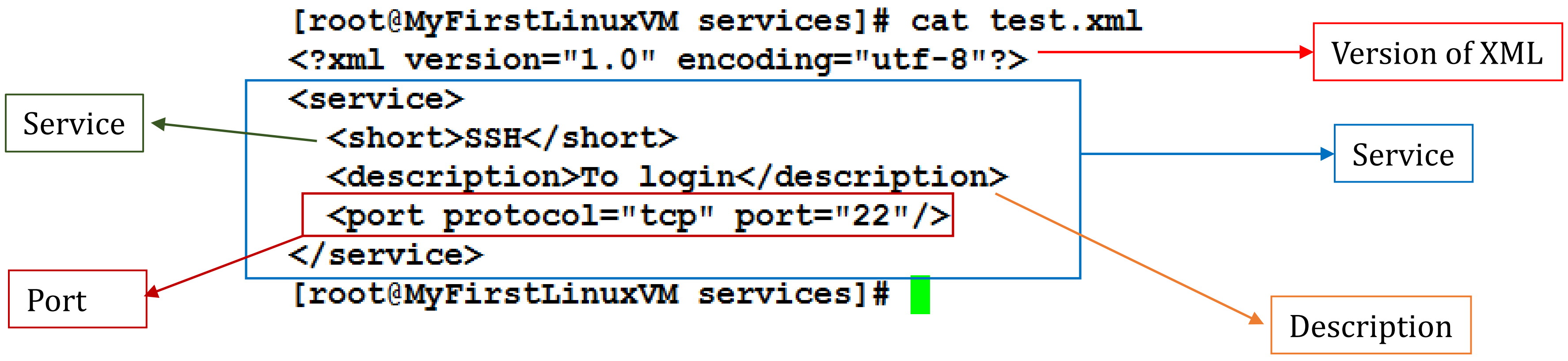


Firewall *(firewalld – Practical Examples)*

- The firewalld has multiple zone, to get a list of all zones
  - **`firewall-cmd --get-zones`**
- To get a list of active zones
  - **`firewall-cmd --get-active-zones`**
- To get firewall rules for public zone
  - **`firewall-cmd --zone=public --list-all`**

    OR
  - **`firewall-cmd --list-all`**
- All services are pre-defined by firewalld. What if you want to add a 3rd party service
  - **`/usr/lib/firewalld/services/allservices.xml`**
  - **`Simply cp any .xml file and change the service and port number`**

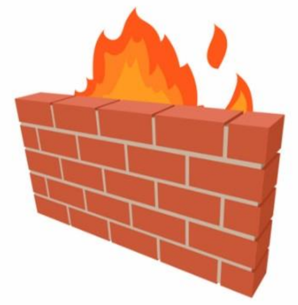




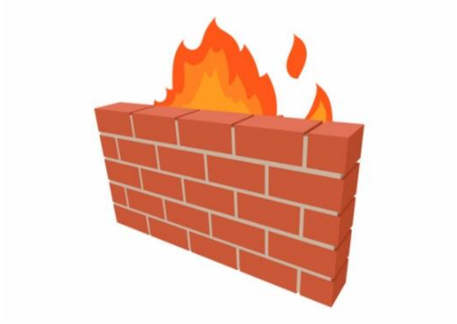
```
[root@MyFirstLinuxVM services]# cat test.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH</short>
  <description>To login</description>
  <port protocol="tcp" port="22"/>
</service>
[root@MyFirstLinuxVM services]#
```

Firewall *(firewalld – Practical Examples)*

- To add a service (http)
  - `firewall-cmd --add-service=http`
- To remove a service
  - `firewall-cmd --remove-service=http`
- To reload the firewalld configuration
  - `firewall-cmd --reload`
- To add or remove a service permanently
  - `firewall-cmd --add-service=http --permanent`
  - `firewall-cmd --remove-service=http --permanent`
- To add a service that is not pre-defined by firewalld
  - `/usr/lib/firewalld/services/allservices.xml`
  - `Simply cp any .xml file sap.xml and change the service and port number (32)`
  - `systemctl restart firewalld`
  - `firewall-cmd --get-services     (to verify new service)`
  - `Firewall-cmd --add-service=sap`



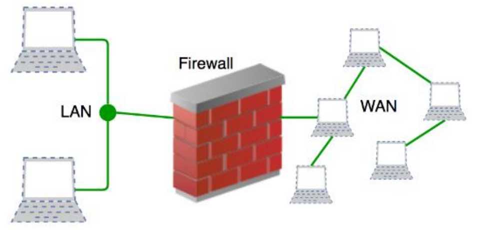
Firewall *(firewalld – Practical Examples)*

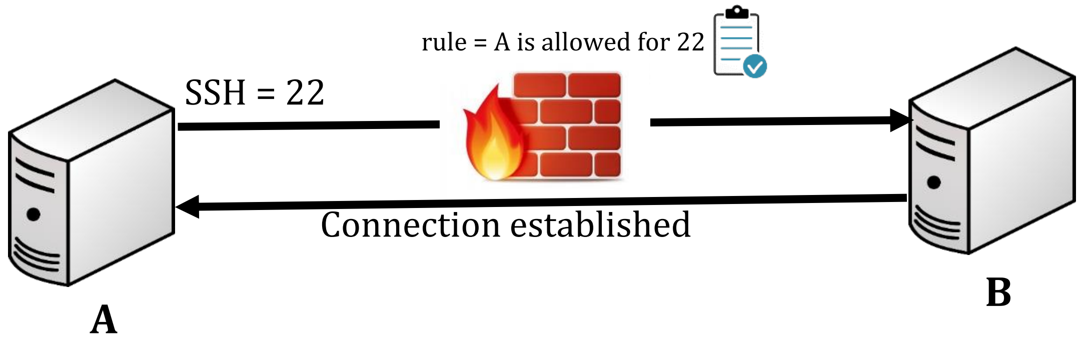


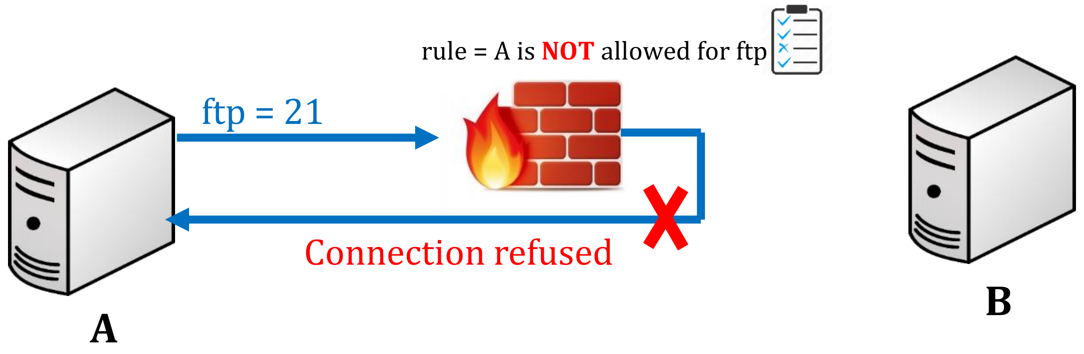
- To add a port
  - `firewall-cmd --add-port=1110/tcp`
- To remove a port
  - `firewall-cmd --remove-port=1110/tcp`
- To reject incoming traffic from an IP address
  - `firewall-cmd --add-rich-rule='rule family="ipv4" source address=“192.168.0.25" reject’`
- To block and unblock ICMP incoming traffic
  - `firewall-cmd --add-icmp-block-inversion`
  - `firewall-cmd --remove-icmp-block-inversion`
- To block outgoing traffic to a specific website/IP address
  - **host -t a** www.facebook.com **= find IP address**
  - `firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -d 31.13.71.36 -j DROP`

Manage Network Security (Firewall)

Introduction to Firewall



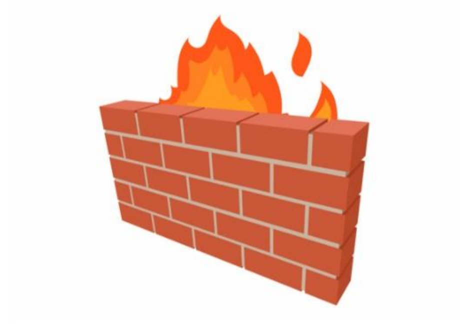




By: Imran Afzal
www.utclisolutions.com

Manage Network Security (Firewall)

Firewall (firewalld)



- Firewalld works the same way as iptables but of course it has it own commands
  - **`firewall-cmd`**
- It has a few pre-defined service rules that are very easy to turn on and off
  - **`Services such as: NFS, NTP, HTTPD etc.`**
- Firewalld also has the following:
  - **`Table`**
  - **`Chains`**
  - **`Rules`**
  - **`Targets`**

Manage Network Security (Firewall)

Firewall (firewalld)

- You can run one or the other
  - iptables or firewalld

- Make sure iptables is stopped, disabled and mask
  - `systemctl stop iptables`
  - `systemctl disable iptables`
  - `systemctl mask iptables`

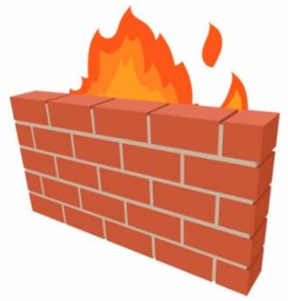
- Now check if filewalld package is installed
  - `rpm –qa | grep firewalld`

- Start firewalld
  - `systemctl start/enable firewalld`

- Check the rule of firewalld
  - `firewall-cmd --list-all`

- Get the listing of all services firewalld is aware of:
  - `firewall-cmd --get-services`

- To make firewalld re-read the configuration added
  - `firewall-cmd --reload`



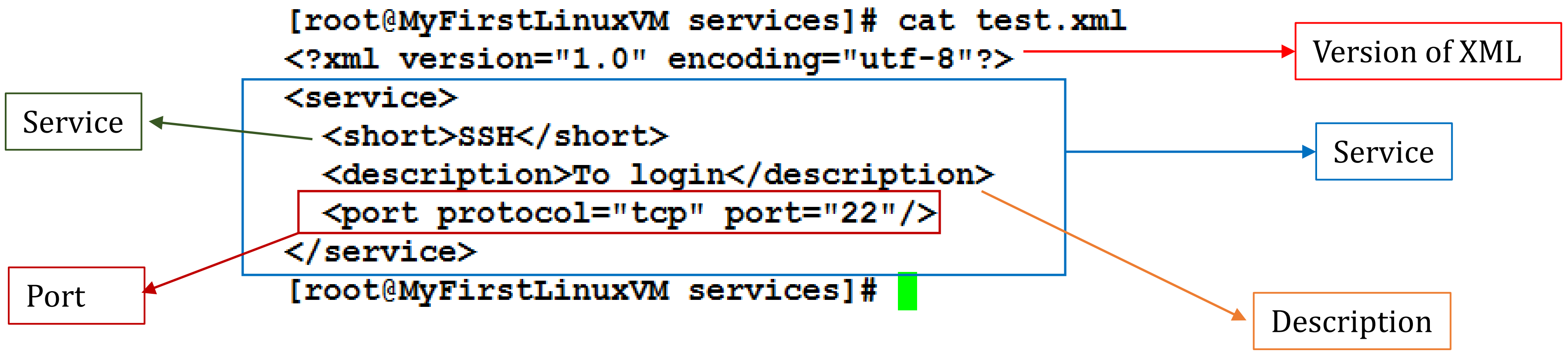
By: Imran Afzal
www.utclisolutions.com

Manage Network Security (Firewall)

Firewall (firewalld, practical examples)

- The firewalld has multiple zone, to get a list of all zones
  - `firewall-cmd --get-zones`
- To get a list of active zones
  - `firewall-cmd --get-active-zones`
- To get firewall rules for public zone
  - `firewall-cmd --zone=public --list-all`

    OR
  - `firewall-cmd --list-all`
- All services are pre-defined by firewalld. What if you want to add a 3rd party service
  - `/usr/lib/firewalld/services/allservices.xml`
  - `Simply cp any .xml file and change the service and port number`

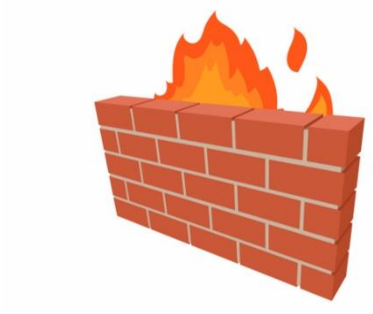


```
[root@MyFirstLinuxVM services]# cat test.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH</short>
  <description>To login</description>
  <port protocol="tcp" port="22"/>
</service>
[root@MyFirstLinuxVM services]#
```

By: Imran Afzal
www.utclisolutions.com

Manage Network Security (Firewall)

Firewall (firewalld, practical examples)

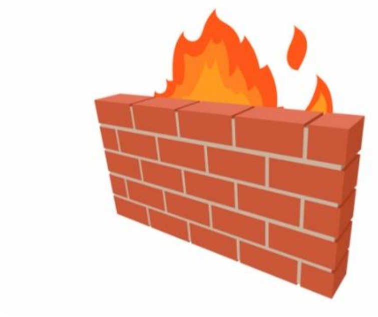


- To add a service (http)
  - `firewall-cmd --add-service=http`
- To remove a service
  - `firewall-cmd --remove-service=http`
- To reload the firewalld configuration
  - `firewall-cmd --reload`
- To add or remove a service permanently
  - `firewall-cmd --add-service=http --permanent`
  - `firewall-cmd --remove-service=http --permanent`
- To add a service that is not pre-defined by firewalld
  - `/usr/lib/firewalld/services/allservices.xml`
  - `Simply cp any .xml file sap.xml and change the service and port number (32)`
  - `systemctl restart firewalld`
  - `firewall-cmd --get-services     (to verify new service)`
  - `Firewall-cmd --add-service=sap`

Manage Network Security (Firewall)

Firewall (firewalld, practical examples)

- To add a port
  - `firewall-cmd --add-port=1110/tcp`
- To remove a port
  - `firewall-cmd --remove-port=1110/tcp`
- To reject incoming traffic from an IP address
  - `firewall-cmd --add-rich-rule='rule family="ipv4" source address=“192.16`
- To block and unblock ICMP incoming traffic
  - `firewall-cmd --add-icmp-block-inversion`
  - `firewall-cmd --remove-icmp-block-inversion`
- To block outgoing traffic to a specific website/IP address
  - `host -t a www.facebook.com = find IP address`
  - `firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -d 31.13.71.36 -j DROP`



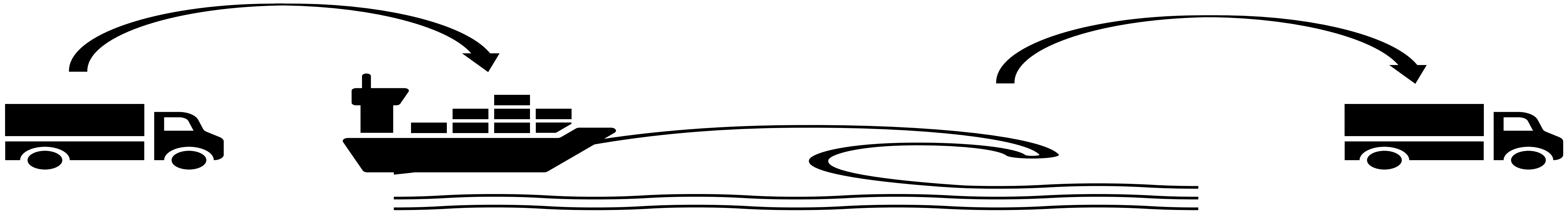
By: Imran Afzal
www.utclisolutions.com

Run Containers

What is a Container?

- The term container and the concept came from the shipping container

- These containers are shipped from city to city and country to country
- No matter which part of the world you go to, you will find these containers with the exact same measurements… <u>**YOU KNOW WHY???**</u>
- Because around the world all docks, trucks, ships and warehouses are built to easily transport and store them



By: Imran Afzal
www.utclisolutions.com

Run Containers

What is a Container?

Now when we are talking about containers in IT we are fulfilling somewhat similar purpose

Please Note:

Container technology is mostly used by developers or programmers who write codes to build applications

As a system administrator your job is to install, configure and manage them.

You can move the application anywhere without moving its OS just like moving the actual physical container anywhere that would fit on any dockyard, truck, ship or warehouse

- An OS can run single or multiple containers at the same time

By: Imran Afzal
www.utclisolutions.com

Run Containers

What are the Container Software?

Developed by:
Solomon Hykes

Released on:
March 20th 2013

Developed by:
Red Hat

Released on:
August 2018

- Docker is the software used to create and manage containers
- Just like any other package, docker can be installed on your Linux system and its service or daemon can be controlled through native Linux service management tool

- Podman is an alternative to docker
- Docker is not supported in RHEL 8
- It is daemon less, open source, Linux-native tool designed to develop, manage, and run containers.

By: Imran Afzal
www.utclisolutions.com

Run Containers

Getting Familiar with Redhat Container Technology

Red Hat provides a set of command-line tools that can operate without a container engine, these include:

- **podman** - for directly managing pods and container images (run, stop, start, ps, attach, etc.)
- **buildah** - for building, pushing, and signing container images
- **skopeo** - for copying, inspecting, deleting, and signing images
- **runc** - for providing container run and build features to podman and buildah
- **crun** - an optional runtime that can be configured and gives greater flexibility, control, and security for rootless containers.

Getting Familiar with podman Container Technology

When you hear about containers then you should know the following terms as well

- **images** – containers can be created through images and containers can be converted to images
- **pods** – Group of containers deployed together on the host. In the podman logo there are 3 seals grouped together as a pod.

podman

Run Containers

Building, Running and Managing Containers

To install podman
- **`yum/dnf install podman –y`**
- **`yum install docker –y`** *(For dockers)*

Creating alias to docker
- **`alias docker=podman`**

Check podman version
- **`podman –v`**

Getting help
- **`podman --help or man podman`**

Check podman environment and registry/repository information
- **`podman info`** *(If you are trying to load a container image, then it will look at the local machine and then go through each registry by the order listed)*

To search a specific image in repository.
- **`podman search httpd`**

Run Containers

Building, Running and Managing Containers

To list any previously downloaded podman images
- `podman images`

To download available images
- `podman pull docker.io/library/httpd`
- `podman images` *(Check downloaded image status)*

To list podman running containers
- `podman ps`

To run a downloaded httpd containers
- `podman run -dt -p 8080:80/tcp docker.io/library/httpd`
  (d=detach, t=get the tty shell, p=port)
- `podman ps` **or** ***Check httpd through web browser***

To view podman logs.
- `podman logs –l`

By: Imran Afzal
www.utclisolutions.com

Run Containers

Building, Running and Managing Containers

To stop a running container
- **`podman stop con-name`** *(con-name from podman ps command)*
- **`podman ps`** *(To list running containers)*

To run a multiple containers of httpd by changing the port #
- **`podman run -dt -p 8081:80/tcp docker.io/library/httpd`**
- **`podman run -dt -p 8082:80/tcp docker.io/library/httpd`**
- **`podman ps`**

To stop and start a previously running container
- **`podman stop|start con-name`**

To create a new container from the downloaded image
- **`podman create –-name httpd-con docker.io/library/httpd`**

To start the newly created container.
- **`podman start httpd-con`**

Run Containers

Building, Running and Managing Containers

Manage containers through systemd

- First you have to generate a unit file
  - **`podman generate systemd --new --files --name httpd-con`**

- Copy it systemd directory
  - **`cp /root/container-httpd.service /etc/systemd/system`**

- Enable the service
  - **`systemctl enable container-httpd-con.service`**

- Start the service.
  - **`systemctl start container-httpd-con.service`**