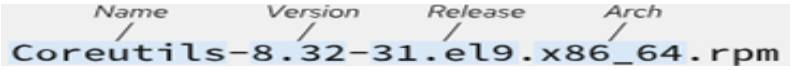
RH134 Summary



Chapter 12: Install and Update Software Packages

- In Red Hat, software is packaged in a compressed file/folder with the extension .rpm , also known as rpm packages.
- The local RPM database on your system stores the information about installed packages.
- You require only the package name to install RPM packages from repositories.
  - If multiple versions exist, then the RPM Package Manager installs the package with the later version number.
  - If multiple releases of a single version exist, then the RPM Package Manager installs the package with the later release number.
- The rpm utility is a low-level tool that retrieves information about the contents of package files and installed packages. By default, the tool gets information from a local database.

Command or option	**Description**
rpm -p package_name	to get information about a downloaded but uninstalled package file.
rpm -q package_name	to get information If its installed
rpm -qa	List all installed packages
rpm -ivh pak_name	-i installs the pak, -v add verbose , -h print # marks for nice display

Option	Purpose
-a	To list all installed packages.
-i	To get detailed package information.
-f	To determine which package provides *FILENAME*
-l	To list the files that the package installs.
-c	To list only the configuration files that the package installs.
-d	To list only the documentation files that the package installs.
-e	To erase a package
-scripts	To list the shell scripts that run before or after you install or remove the package.
-changelog	To list the change log information for the package.

Command or option	**Description**
Rpm2cpio package_name	to extract files from an RPM package file without installing the package.
Cpio -id **Cpio -t**	-i option to extract files from standard input. -d option to create subdirectories as needed. pattern can also be specified in “” (ex. “*txt”), -t list files

- DNF (Dandified YUM) replaced YUM as the package manager.
- The rpm command is used to install packages, not to work with package repositories or to resolve dependencies from multiple sources automatically.
- dnf command can download, install, update, remove, and get information about packages, their dependencies and transactions.

Command	Purpose
dnf list	To list all installed and available packages.
dnf info	To return detailed information about a package, including the needed disk space for installation
dnf search	To list packages by keywords that are in the name and summary fields only
dnf provides	To list packages that match the specified path name (for example /var/www/html).
dnf install	To obtain and install a software package, including any dependencies.

Command or option	Description
uname -r	show the kernel version and release.
dnf remove httpd	removes an installed httpd package, including any supported packages
dnf group list	shows the names of installed and available groups.
dnf group list hidden	Shows the hidden groups
dnf group install name	To install package group

Packages Groups :

- dnf command also has the concept of *groups*, which are collections of related software that are installed together.
- There are two types of groups:
  - Regular(collection of packages)
  - environment groups(collection of regular groups).
- The packages or groups that these collections provide might be mandatory, default or optional installed if the group is installed.

- All installation and removal transactions are logged in the /var/log/dnf.rpm.log file.

- **Command : [root@localhost ~]$** dnf history
  - this displays a summary of installation and removal transactions.
- **Command :**
  [root@localhost ~]$ dnf history undo 6
  This will reverses a transaction

```
[root@host ~]# dnf history
ID     | Command line             | Date and time    | Action(s)      | Altered
-------------------------------------------------------------------------------
     7 | group install RPM Develop | 2022-03-23 16:46 | Install        |   20
     6 | install httpd             | 2022-03-23 16:21 | Install        |   10 EE
     5 | history undo 4            | 2022-03-23 15:04 | Removed        |   20
     4 | group install RPM Develop | 2022-03-23 15:03 | Install        |   20
     3 |                           | 2022-03-04 03:36 | Install        |    5
     2 |                           | 2022-03-04 03:33 | Install        |  767 EE
     1 | -y install patch ansible- | 2022-03-04 03:31 | Install        |   80
```

RHEL9 distributes the content through two main software repositories: *BaseOS* and (AppStream)

Command	Purpose
dnf module list [module-name]	To list the available modules with module name, stream, profile, and summary.
dnf module info <module-name>	To return detailed information about a module with its streams and packages.
dnf module provides <package-name>	To display which module provides a specific package.
dnf module install	To obtain and install a software package, including any dependencies.

- Systems often have access to many Red Hat repositories.
- Non-Red Hat sources provide software through third-party repositories. For example, Adobe provides some of its software for Linux through DNF repositories.

Command	Purpose
dnf repolist all	To list the available repositories and their statuses.
dnf config-manager --enable|--disable <repository-name>	To enable and disable repositories.
dnf config-manager --add-repo=""	To add repositories to the machine.

- The file will be added in the **/etc/yum.repos.d/** directory.
- The .repo files often list multiple repository references in a single file. Each repository reference begins with a single-word name in square brackets.

Chapter 4: Archive And Transfer Files

- Common options with tar command:

Option	Purpose
-c or --create	To create an archive file.
-t or --list	To list the content of the archive file.
-x or --extract	To extract an archive file.
-f or --file	Follow this option with the file name to create or extract the file.
-p or --preserve-permissions	To preserve the original file permissions when extracting.
--xattrs	To enable extended attribute support, and store extended file attributes(metadata).
--selinux	To enable SELinux context support, and store SELinux file contexts.
-v or --verpose	To Show the archived or extracted files during the tar operation.

Compression algorithms

Option	Purpose
-a or --auto-compress	Follow this option with archive's suffix to determine the algorithm to use.
-z or --gzip	To use the gzip compression algorithm, resulting in a .tar.gz suffix
-j or --bzip2	To use the bzip2 compression algorithm, resulting in a .tar.bz2 suffix
-J or --xz	To use the xz compression algorithm, resulting in a .tar.xz suffix

- Extract a tar archive into an empty directory to avoid overwriting existing files.
- If the root extracts an archive, the original user and group ownership is preserved.
- If a regular user extracts files, the user becomes the owner of the extracted files.
- – C extract into specific dir rather than the present directory.

```
tar -xvf output_filename.tar -C /home/deploy/
```

- **Gzip Compression**- The Gzip format is the most widely used compression format for tar, it is fast for creating and extracting files. Files with gz compression have normally the file ending .tar.gz or .tgz.
- **Bzip2 Compression**- The Bzip2 format offers better compression than the Gzip format. Creating files is slower, the file ending is usually .tar.bz2. One problem with this it is not widely available on all distributions.
- **xz compression**- is newer, and offers the best compression ratio of the available methods.

  -l option to view the uncompressed size of a compressed single or archive file. Use this option to verify sufficient space is available before uncompressing or extracting a file.

Transferring Files Safely using SFTP

- To list files in the local machine, prefix the command with 'l' so that would be lls .
- Put allows to copy files from local to remote.
- Get allows to copy files from remote to local.
- **put** -r directory -> recursevly upload the whole dir.

```
sftp> mkdir hostbackup
sftp> cd hostbackup
sftp> put /etc/hosts
Uploading /etc/hosts to /home/remoteuser/hostbackup/hosts
/etc/hosts                                    100%  227     0.2KB/s
```

- **[user@host ~]$** sftp remoteuser@remotehost:/home/remoteuser/remotefile

  Get a remote file with the sftp command on a single command line, without opening an interactive session.

Options with rsync command

Command or option	Description
-r or --recursive	**To synchronize the whole directory tree recursively.**
-l or --links	**To synchronize symbolic links.**
-t or --times	**To preserve timestamps.**
-p or --perms	**To preserve permissions.**
-g or --group	**To preserve group ownership.**
-o or --owner	**To preserve user ownership.**
-a	Preserves all of the above
-n	**To dry run or simulate what changes that the rsync command would perform when executing the command.**
-v or --verpose	**To Show the verbose and provide a more detailed output.**
--max-size	limit the size of files to be copied and avoid copying very large files.

- **[root@host ~]$** rsync -av /var/log /temp
  synchronizes the contents of the /var/log directory to the /tmp directory.
- **Note that in the example above the directory itself was copied and inside is the contents of the directory. If you wish to copy the files separately add '/' at the end.**

```
[user@host ~]$ su -
Password: password
[root@host ~]# rsync -av /var/log /tmp
receiving incremental file list
log/
log/README
log/boot.log
...output omitted...
log/tuned/tuned.log

sent 11,592,423 bytes  received 779 bytes  23,186,404.00 bytes/sec
total size is 11,586,755  speedup is 1.00
[user@host ~]$ ls /tmp
log  ssh-RLjDdarkKiW1
```

```
[root@host ~]# rsync -av /var/log/ /tmp
sending incremental file list
./
README
boot.log
...output omitted...
tuned/tuned.log

sent 11,592,389 bytes  received 778 bytes  23,186,334.00 bytes/sec
total size is 11,586,755  speedup is 1.00
[root@host ~]# ls /tmp
anaconda                dnf.rpm.log-20190318  private
audit                   dnf.rpm.log-20190324  qemu-ga
boot.log                dnf.rpm.log-20190331  README
...output omitted...
```

- **[root@host ~]$** rsync A/ Backup-A-dir/ --include=*.py --exclude=*.tmp.py
  include and exclude options used to only copy python scripts and do not copy .tmp.py files.

- To display Progress of Transfer.
  Use -progress option

```
$ rsync -r A/ Backup-A-dir/ -progress
sending incremental file list
created directory Backup-A-dir
./
file1.txt 0 100% 0.00kB/s 0:00:00 (xfr#1, to-chk=5/7)
file2.txt 0 100% 0.00kB/s 0:00:00 (xfr#2, to-chk=4/7)
file3.txt 0 100% 0.00kB/s 0:00:00 (xfr#3, to-chk=3/7)
file4.txt 0 100% 0.00kB/s 0:00:00 (xfr#4, to-chk=2/7)
```

- Like the sftp command, the rsync command specifies remote locations in the [user@]host:/path format. The remote location can be either the source or the destination system, but one of the two machines must be local.
- You must be the root user on the destination system to preserve file ownership.
- If the destination is remote, then authenticate as the root user.
- If the destination is local, then you must run the rsync command as the root user.

```
[root@host ~]# rsync -av hosta:/var/log /tmp
root@hosta's password: password
receiving incremental file list
log/boot.log
log/dnf.librepo.log
log/dnf.log
...output omitted...

sent 9,783 bytes  received 290,576 bytes  85,816.86 bytes/sec
total size is 11,585,690  speedup is 38.57
```

```
[root@host ~]# rsync -av /var/log hosta:/tmp
root@hosta's password: password
receiving incremental file list
log/
log/README
log/boot.log
...output omitted...
sent 9,783 bytes  received 290,576 bytes  85,816.86 bytes/sec
total size is 11,585,690  speedup is 38.57
```

Chapter 2: Scheduling Tasks

- `at` command is **available and enabled by default** on RHEL systems.
- Used to **schedule one-time jobs** to run at a **specific time**.
- Jobs are **managed by the `atd` daemon**.
- **Any user** can queue jobs using the `at` command.
- `atd` uses **26 job queues** (`a` to `z`):
  - **Later letters (e.g., 'z')** = **lower priority** (i.e., higher `nice` values).
  - **Earlier letters (e.g., 'a')** = **higher priority**.

```
at [time] [date/day] command_to_run
```

- When you execute the **at** command, If no command is provided, you will enter interactive mode that you will be prompted into a terminal to write the commands to execute at the time specified .
- You can exit by typing CTRL+d.
- **Command : [user@host ~]$** at now + 1 min — Schedule to run after 1 minute.
- **Command : [user@host ~]$** at 03:55 — Schedule to run at a specific time.
- **Command : [user@host ~]$** atq — View queued jobs.
- **Command : [Student@localhost ~]$** atrm 28 — To remove job with ID =28 .
- **Command : [Student@localhost ~]$** at -c 28 — To view the content of the scheduled job.
- **Command : [Student@localhost ~]$** at now + 2 days -f task.sh — To schedule a file to execute.

- cron command is used to schedule recurring job
- When you edit the crontab with crontab -e , the system creates a temp copy of your crontab file in the /tmp
- **Unique Filename for Each Session**
- actual crontab files located in /var/spool/cron/crontabs

Syntax

*	*	*	*	*
Minute(0-59)	Hour(0-23)	Date(1-31)	Month(1-12)	Weekday(0-6)

Examples

* * * * *	Run Cron Job Every Minute
0 * * * *	Run Cron Job Every Hour
0 16 * 10 0	Run Cron Job Every 4PM on all Sundays in October
0 2 * * *	Run Cron Job at 2 am Every Day
0 0 1 * *	Run Cron Job Every 1st of the Month
0 0 15 * *	Run Cron Job Every 15th of the Month
0 0 * * 6	Run Cron Job on Saturday at Midnight

Files

/etc/cron.d	Location for system cron entries
/etc/crontab	Legacy system crontab
/var/spool/cron	Location for user crontabs
/lib/systemd/system/crontab.service	systemd cron service

- **Recurring jobs** are best run from **system accounts**, not user accounts.
- Use **system-wide crontab files** instead of the `crontab` command for system jobs.
- System-wide crontab entries are **similar to user crontab entries**, but include an **extra field** to specify the **user** who will run the job.
- **User jobs**:
  - Affect only the user.
  - Automate personal/manual tasks.
- **System jobs**:
  - Affect the **entire system**.
  - Usually require **privileged access** (e.g., `root`).
- System jobs are scheduled using files in:
  - `/etc/crontab`
  - `/etc/cron.d/` (recommended location for custom jobs).
- Use `/etc/cron.d/` for custom system jobs to **avoid overwriting** by package updates.
- **Packages** with recurring jobs also place their crontab files in `/etc/cron.d/`.
- Related jobs can be grouped in a **single file** in `/etc/cron.d/`.
- There are also **special directories** for periodic scripts:
  - `/etc/cron.hourly/`
  - `/etc/cron.daily/`
  - `/etc/cron.weekly/`
  - `/etc/cron.monthly/`
- These directories contain **executable shell scripts**, not crontab files.

anacron

- **/etc/anacrontab** ensures scheduled jobs **run even if the system was off or hibernating** at the scheduled time.
- **Major difference between cron and anacron**:
  - **Cron**: For systems that run **continuously**.
  - **Anacron**: For systems that are **powered off regularly** (e.g., laptops).
- When a job runs, crond updates its **timestamp**, so you can track **when it last ran**.
- Timestamps for job execution are stored in:
  ➤ /var/spool/anacron/
- This directory controls **daily**, **weekly**, and **monthly** job tracking.

🔁 **/etc/anacrontab Syntax:** Each line contains **four fields**:

1. **Period in days** – How often the job should run (e.g., 1, 7, or macros like @daily, @weekly).
2. **Delay in minutes** – How long to wait after boot before running the job.
3. **Job identifier** – Unique name used in logs.
4. **Command** – The actual command/script to run.

📜 **Example:** @daily 10 example.daily /bin/bash /home/aaronkilik/bin/backup.sh

- Runs **daily**
- **10 minutes** after boot (if missed)
- Logs as example.daily
- Runs the **backup.sh** script

Sample Timer Unit

- The sysstat package provides the systemd timer unit, called the sysstat-collect.timer service, to collect system statistics every 10 minutes.
- The following output shows the contents of the /usr/lib/systemd/system/sysstat-collect.timer configuration file.

A (2025-04-* 12:35,37,39:16) value against the OnCalendar option causes the timer unit to activate the corresponding service unit at the 12:35:16, 12:37:16, and 12:39:16 times, every day during April 2025.

```
...output omitted...
[Unit]
Description=Run system activity accounting tool every 10 minutes

[Timer]
OnCalendar=*:00/10

[Install]
WantedBy=sysstat.service
```

- After any change in the timer unit configuration file, use the systemctl daemon-reload command to ensure that the systemd timer unit loads the changes.
- After reloading the systemd daemon configuration, use the systemctl command to activate the timer unit.

- **Important note:** If you want to modify any file under the /usr/lib/systemd/system directory, it is best to first copy the file you wish to change to /etc/systemd/system directory and then modify the copied file to prevent any update to the provider package from overwriting the changes.

- If two files exist with the same name in the /usr/lib/systemd/system and /etc/systemd/system directories, then the systemd timer unit parses the file in the /etc/systemd/system

- To prevent long-running systems from filling up their disks with stale data, a systemd timer unit called systemd-tmpfiles-clean.timer at a regular interval triggers systemd-tmpfiles-clean.service, which executes the systemd-tmpfiles --clean command.
- A systemd timer unit configuration has a [Timer] section for indicating when to start the service.
- The systemd-tmpfiles-clean service configuration files can exist in three places:

/etc/tmpfiles.d/*.conf – /run/tmpfiles.d/*.conf — /usr/lib/tmpfiles.d/*.conf

Managing temporary files manually

- Use the files in the /etc/tmpfiles.d/ directory to configure temporary locations, and to overwrite vendor-provided defaults.
- The files in the /run/tmpfiles.d/ directory are volatile files, which daemons use to manage their own runtime temporary files.
- Relevant RPM packages provide the files in the /usr/lib/tmpfiles.d/ directory; therefore do not edit these files.
- If a file in the /run/tmpfiles.d/ has the same name as a file in the /usr/lib/tmpfiles.d/ directory, then the service uses the file in the /run/tmpfiles.d/ directory.
- If a file in the /etc/tmpfiles.d/ directory has the same file name as a file in either the /run/tmpfiles.d/ or the /usr/lib/tmpfiles.d/ directories, then the service uses the file in the /etc/tmpfiles.d/ directory.

Chapter 6: SELinux

- SELinux ensures access control for processes. What permissions are allowed for a process, what files and directories a process can access and manipulate.
- This type of access control is what is known as Mandatory access control (MAC) contrary to discretionary access control (DAC), which provides a simple and flexible way to manage file permissions based on user and group ownership.
- By default, an SELinux policy does not allow any access unless an explicit rule grants access. When no allow rule is defined, all access is disallowed.

```
SELinux User : Role : Type : Level  File
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

- You can use the -Z option to list the file contexts of a file, dir, or process. The following example shows the file contexts of some process and the /var/www directory.

```
[root@host ~]# ps axZ
LABEL                               PID TTY      STAT   TIME COMMAND
system_u:system_r:kernel_t:s0         2 ?        S      0:00 [kthreadd]
system_u:system_r:kernel_t:s0         3 ?        I<     0:00 [rcu_gp]
system_u:system_r:kernel_t:s0         4 ?        I<     0:00 [rcu_par_gp]
...output omitted...
[root@host ~]# systemctl start httpd
[root@host ~]# ps -ZC httpd
LABEL                               PID TTY          TIME CMD
system_u:system_r:httpd_t:s0       1550 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0       1551 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0       1552 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0       1553 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0       1554 ?        00:00:00 httpd
[root@host ~]# ls -Z /var/www
system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
system_u:object_r:httpd_sys_content_t:s0 html
```

Mode	Purpose
Enforced (SELinux security policy is enforced)	The default and recommended mode for SELinux. It means that SELinux will check and disallow processes from accessing a resource if the file context types do not match or a policy is not set.
Permissive (SELinux prints warnings instead of enforcing)	Useful for debugging purposes. SELinux will check for file type context and policies, but will not disallow access. It will still keep logs of such incident and print out a warning if types do not match. When a process is not working as intended, you can switch the SELinux mode to permissive and see if SELinux was causing the process not to work as expected.
Disabled (No SELinux policy is loaded)	This is not recommended as it could have serious security implications, disabling MAC.

- You can use the **setenforce** command to change the mode temporarily. SELinux can take one of these two values: enforcing or 1 - permissive or 0 .
- To change the default value of SELinux mode persistently, you will need to modify the configuration file which can be found at /etc/selinux/config .

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
...output omitted...
SELINUX=enforcing
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

- Each file, directory, process has a file context. These lists of contexts are stored in a database in the /etc/seLinux/targeted/contexts/files directory. In the file_contexts file.

- /.* means that any file or directory matches a forward slash followed by zero or more characters of any type after the "/var/www" directory is assigned a file context type of httpd_sys_content_t .
- ? makes the expression optional. This means that the regular expression will match both paths that end with a forward slash and paths that do not.
- If there are multiple policy rules for the /var/www directory. SELinux will look to match the most specific rule.
- For Copying, the cp -p command preserves all file attributes where possible, and the cp -c command preserves only SELinux contexts, during a copy.
- On the other hand, moving files will preserve the SELinux contexts.
- To manage SELinux contexts, install the policycoreutils and policycoreutils-python-utils packages, which contain the **restorecon** and **semanage** commands.
- **Note:** If you add a context it will be added in the 'file_contexts.local' file.
- You need to have admin privileges to run semanage .

Command or option	**Description**
cp	-**p** preserves all file attributes where possible / -**c** preserves only SELinux contexts
mv	preserve the SELinux contexts by defaults when moving
ls	-Z list the context of a file / -Zd list the contex of the dir
semanage fcontext	**-a , --add** Add a record of the specified object type. **-l , --list** List all records of the specified object type. **-d, --delete** Delete a record of the specified object type. **-t** option means that we are adding a file context type
restorecon	-C with the list option to only list the customised file contexts -R: To operate recursively on all files and directories under /virtual. -F: To force the relabelling of all files and directories, even if they already have the correct SELinux context. -v: To be verbose and provide detailed output as it processes each file and directory. Double v is used for a more verbose output
getsebool -a	to list the available booleans
semanage boolean -l -C	To list only Booleans with a current setting that is different from the default setting at boot
setsebool	Set a boolean on or off / -p option makes it persistant

```
[root@host ~]# semanage fcontext -a -t httpd_sys_content_t '/virtual(/.*)?'
```

- Booleans are typically used to enable or disable specific SELinux features, while file contexts are used to define the access control policies for specific files or directories.
- In general, booleans are used to control system-wide settings, while file contexts are used to control the access control policies for specific files and directories.

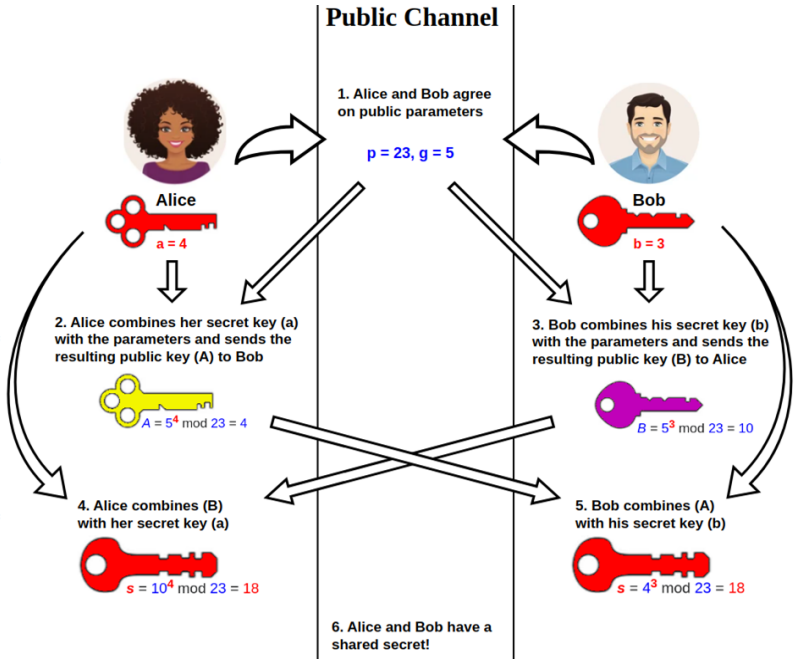
SELinux Logs

- The SELinux troubleshoot service, from the **setroubleshoot-server** package, provides tools to diagnose SELinux issues. When SELinux denies an action, an Access Vector Cache (AVC) message is logged to the /var/log/audit/audit.log security log file.
- The SELinux troubleshoot service monitors for AVC events and sends an event summary to the /var/log/messages file.
- The AVC summary includes an event unique identifier (UUID).
- Use the sealert -l UUID command to view comprehensive report details for the specific event.
- Use the sealert -a /var/log/audit/audit.log command to view all existing events.

- Use the ausearch command to search for AVC events in the /var/log/audit.log log file.
- Use the -m option to specify the AVC message type and the -ts option to provide a time hint, such as recent.

Chapter 10: SSH

- SSH uses a **client-server model** for connections.
- The **remote machine must run an SSH daemon/service**:
  - Listens on **port 22**.
  - **Authenticates** connection requests.
  - **Starts a session** if credentials are correct.
- The **user's computer needs an SSH client**:
  - Communicates using the **SSH protocol**.
  - Supplies **remote host info** and **credentials**.
  - Can specify **connection type details** (e.g., interactive session, port forwarding).



- The following figure perfectly explains how the session key is created using the Diffie Hellman.

- When a user runs the `ssh` command, it **checks for the server's public key** in the **known hosts files**.
- **Search order for known host keys**:
  - First: `/etc/ssh/ssh_known_hosts` (system-wide file).
  - Then: `~/.ssh/known_hosts` (user-specific file).
- **/etc/ssh/ssh_known_hosts**:
  - System-wide list of public keys for remote machines.
  - Must be **manually maintained** or updated using tools like `ssh-keyscan`.
- **~/.ssh/known_hosts**:
  - Contains public keys of remote machines trusted by the **specific user**.
  - Maintained automatically after first successful SSH connection (unless strict key checking is configured).
- Each entry in the **known hosts file** is a **single line** with **three fields**:
  - **Hostnames/IP addresses** associated with the public key.
  - **Encryption algorithm** used (e.g., ssh-rsa, ecdsa-sha2-nistp256).
  - The **public key** itself.

W -f Shows the logged on users (-f shows FROM field)
-f option specifies the files to save the keys in

- To authenticate a user using key-based approach:
  1. Generate public and private keys on the client/host machine. (SSH Key Generation)
  2. Copy the host's public key to the remote machine to a file within the user's home directory at ~/.ssh/authorized_keys.(SSH Key Sharing)
- ssh-keygen command is used to create a key pair. By default, the ssh-keygen command saves your private and public keys in the ~/.ssh/id_rsa and ~/.ssh/id_rsa.pub files, but you can specify a different name.

- A passphrase may be used to encrypt- later decrypt- the private key. The passphrase must be entered each time the private key is used.

- The **private key** must have **strict permissions**:
  - Should be **readable and writable only by the owner**.
  - **Permission:** `600` (owner can read/write, no access for others).
- The **public key** is **not sensitive**:
  - Can be **readable by anyone** on the system.
  - **Permission:** `644` (owner can read/write, others can read).

ssh-copy-id command which copies the public key of the SSH key pair to the remote server

flag -i is used to identify the file location

SSH Troubleshooting

- The ssh command provides three verbosity levels with the -v , -vv , and - vvv options, which respectively provide increasing debugging information .
- When using the lowest verbosity option :

1. OpenSSH and OpenSSL versions.
2. OpenSSH configuration files.
3. Connection to the remote host.
4. Trying to authenticate on the remote host.
5. Allowed authentication methods on the remote host.
6. Trying to authenticate by using the SSH key.
7. Using the /home/user/.ssh/id_rsa key file to authenticate.
8. The remote hosts accepts the SSH key.

```
[user@host ~]$ ssh -v user@remotehost
OpenSSH_8.7p1, OpenSSL 3.0.1 14 Dec 2021 1
debug1: Reading configuration data /etc/ssh/ssh_config 2
debug1: Reading configuration data /etc/ssh/ssh_config.d/01-training.conf
debug1: /etc/ssh/ssh_config.d/01-training.conf line 1: Applying options for *
debug1: Reading configuration data /etc/ssh/ssh_config.d/50-redhat.conf
...output omitted...
debug1: Connecting to remotehost [192.168.1.10] port 22. 3
debug1: Connection established.
...output omitted...
debug1: Authenticating to remotehost:22 as 'user' 4
...output omitted...
debug1: Authentications that can continue: publickey,gssapi-keyex,gssapi-with-mic,password 5
...output omitted...
debug1: Next authentication method: publickey 6
debug1: Offering public key: /home/user/.ssh/id_rsa RSA SHA256:hDVJjD7xrUjXGZVRJQixxFV6NF/ssMjS6AuQ1+VqUc4 7
debug1: Server accepts key: /home/user/.ssh/id_rsa RSA SHA256:hDVJjD7xrUjXGZVRJQixxFV6NF/ssMjS6AuQ1+VqUc4 8
Authenticated to remotehost ([192.168.1.10]:22) using "publickey".
...output omitted...
```

The configurations of OpenSSH is stored in /etc/ssh/sshd_config

Chapter 11: Manage Network Security (Firewall)

- **firewalld** inspects the **source address** of each incoming packet.
- **If the source address is assigned to a zone**:
  - The **rules for that zone** are applied.
- **If not assigned to any zone**:
  - firewalld uses the **zone of the incoming network interface**.
- **If the network interface has no zone**:
  - The packet is handled using the **default zone's rules**.

Zone	Default Configuration
Trusted	Allow all incoming traffic
Home / Internal	Reject incoming traffic unless related to outgoing traffic or matching the ssh, mdns, ipp-client, samba-client, or dhcpv6-client predefined services.
Work	Reject incoming traffic unless related to outgoing traffic or matching the ssh, ipp-client, or dhcpv6-client predefined services.
Public (default)	Reject incoming traffic unless related to outgoing traffic or matching the ssh or dhcpv6-client predefined services.
External / dmz	Reject incoming traffic unless related to outgoing traffic or matching the ssh predefined service. Outgoing IPv4 traffic that is forwarded through 'External' zone is *masqueraded* to appear that it originated from the IPv4 address of the outgoing network interface.
Block	Reject all incoming traffic unless related to outgoing traffic.
Drop	Drop all incoming traffic unless related to outgoing traffic (do not even respond with ICMP errors).

Service	Default Configuration
SSh	Local SSH server. Traffic to 22/tcp.
dhcpv6-client	Local DHCPv6 client. Traffic to 546/udp on the fe80::/64 IPv6 network.
ipp-client	Local IPP printing. Traffic to 631/udp.
samba-client	Local Windows file and print sharing client. Traffic to 137/udp and 138/udp.
mdns	Multicast DNS (mDNS) local-link name resolution. Traffic to 5353/udp to the 224.0.0.251 (IPv4) or ff02::fb (IPv6) multicast addresses.

Command or option	**Description**
firewall-cmd --get-services	list the services

firewall-cmd commands	Explanation
`--get-default-zone`	Query the current default zone.
`--set-default-zone=`*`ZONE`*	Set the default zone. This default zone changes both the runtime and the permanent configuration.
`--get-zones`	List all available zones.
`--get-active-zones`	List all zones that are currently in use (with an interface or source that is tied to them), along with their interface and source information.
`--add-source=`*`CIDR`* `[--zone=`*`ZONE`*`]`	Route all traffic from the IP address or network/netmask to the specified zone. If no `--zone=` option is provided, then the default zone is used.
`--remove-source=`*`CIDR`* `[--zone=`*`ZONE`*`]`	Remove the rule that routes all traffic from the zone that comes from the IP address or network. If no `--zone=` option is provided, then the default zone is used.
`--add-interface=`*`INTERFACE`* `[--zone=`*`ZONE`*`]`	Route all traffic from *`INTERFACE`* to the specified zone. If no `--zone=` option is provided, then the default zone is used.
`--change-interface=`*`INTERFACE`* `[--zone=`*`ZONE`*`]`	Associate the interface with ZONE instead of its current zone. If no `--zone=` option is provided, then the default zone is used.
`--list-all [--zone=`*`ZONE`*`]`	List all configured interfaces, sources, services, and ports for *`ZONE`*. If no `--zone=` option is provided, then the default zone is used.
`--list-all-zones`	Retrieve all information for all zones (interfaces, sources, ports, and services).
`--add-service=`*`SERVICE`* `[--zone=`*`ZONE`*`]`	Allow traffic to *`SERVICE`*. If no `--zone=` option is provided, then the default zone is used.
`--add-port=`*`PORT/PROTOCOL`* `[--zone=`*`ZONE`*`]`	Allow traffic to the *`PORT/PROTOCOL`* ports. If no `--zone=` option is provided, then the default zone is used.
`--remove-service=`*`SERVICE`* `[--zone=`*`ZONE`*`]`	Remove *`SERVICE`* from the allowed list for the zone. If no `--zone=` option is provided, then the default zone is used.
`--remove-port=`*`PORT/PROTOCOL`* `[--zone=`*`ZONE`*`]`	Remove the *`PORT/PROTOCOL`* ports from the allowed list for the zone. If no `--zone=` option is provided, then the default zone is used.
`--reload`	Drop the runtime configuration and apply the persistent configuration.

- **semanage port -l option used to list the current port label assignments.**
- **use the grep -w option to filter the SELinux port label using the port number.**
- **semanage port -a option, the -t option denotes the type, and the -p option denotes the protocol.**

```
[root@host ~]# semanage port –a –t port_label –p tcp|udp PORTNUMBER
```

Chapter 5: Tune System Performance

Tuned is installed by default on Red hat and it is responsible for setting system settings - low level settings, kernel,CPU, memory, etc- to maximise performance based on a specific workload.

- Tuned Configuration file is located in /etc/tuned/tuned-main.conf

By default, dynamic tuning is turned off (dynamic_tuning value is 0). Change this value to 1 to enable it.

Static Tuning	Dynamic Tuning
tuned sets the kernel parameters and other settings only once no change.	tuned can alter these settings as the workload changes.
Performance is not optimized.	Performance is optimized based on usage patterns and workloads
Consumes less system resources	Consumes more system resources to constantly monitor the system (cpu,memory,etc) to accommodate for the workload.

Profile	Purpose
balanced	Ideal for systems that require a compromise between power saving and performance.
throughput-performance	Tunes the system for maximum throughput.
accelerator-performance	Same as throughput-performance, and also reduces the latency to less than 100 µs.
latency-performance	For server systems that require low latency at the expense of power consumption.
powersave	Tunes the system for maximum power saving.
network-throughput	Derived from the throughput-performance. Additional network tuning parameters are applied for maximum network throughput.
network-latency	Derived from the latency-performance profile. Enables additional network tuning parameters to provide low network latency.
desktop	Derived from the balanced profile. Focus on less latency.
hpc-compute	Derived from the latency-performance. Ideal for high-performance computing.
virtual-guest	Tunes the system for maximum performance if it runs on a VM.
virtual-host	Tunes the system for maximum performance if it is a host for VMs.
intel-sst	Optimized for systems with Intel Speed Select Technology configurations. Use it as an overlay on other profiles.
optimize-serial-console	Increases responsiveness of the serial console. Use it as an overlay on other profiles.

The tuning profiles are stored under the /usr/lib/tuned and /etc/tuned directories.

- **Tuned** uses performance **profiles** stored in:
  - `/usr/lib/tuned/` → **Default profiles** provided by the system or packages.
  - `/etc/tuned/` → **Custom or user-defined profiles**.
- If there is a **profile with the same name** in both directories (e.g., `/usr/lib/tuned/myprofile` and `/etc/tuned/myprofile`), then:

✅ **The version in `/etc/tuned/` overrides the one in `/usr/lib/tuned/`**

tuned-adm command options

Command	Purpose
active	To view current profile.
Profile-info [profile-name]	To display profile information. If no specific profile is specified, it will display current profile info .
recommend	To recommend a suitable profile for the current system requirements.
list	To list available profiles.
profile <profile-name>	To switch profiles.
off	to disable any profile.

Linux Process scheduling

- On Linux, processes have a value called 'nice' that determine their priority.
- Nice values range from -20 to 19, with -20 having the highest priority and 19 the lowest priority.
- Processes have a default, nice value of 0. Processes inherit their starting nice value from their parent process.
- top command can be used to view Nice Values of running processes
- Use the nice command to start commands with a default or higher nice value with -n option.
- Use renice command to change the Nice Value of an existing process.
-

Chapter 7: Manage Basic Storage

Command or option	**Description**
Lsblk --fs	List block devices (fs option show filesystem and mounting)

- To be able to access the content of a block device, you need to create a file system for it.
- After creating a file system. You need to create a directory -mount point- which will contain the content of the storage device.
- The process of the aforementioned step, to store the contents of a block device on a directory, is known as mounting.
- When mounting a file system to a mount point. The mount point must be empty, otherwise mounting will produce

MBR	**GPT**
standard on systems that run BIOS firmware	systems that run *Unified Extensible Firmware Interface (UEFI)* firmware
Maximum of 4 primary partitions, 15 logical partitions	Maximum of 128 partitions.
With a 32-bit partition size, disks that are partitioned with MBR can have a size of up to 2 TiB.	64 bits for logical block addresses, to support partitions and disks of up to 8 billion tebibytes (TiB).

- There are many different partition editors such as:

1. gdisk and fdisk were initially created to support GPT.
2. parted and the libparted library have been the RHEL standard for years.

- The standard partition editor on the command line in RHEL is parted. parted command requires admin privileges.
- The following example uses the print subcommand to display the partition table on the /dev/vda disk.

```
[root@host ~]# parted /dev/vda print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 53.7GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  10.7GB  10.7GB  primary  xfs          boot
 2      10.7GB  53.7GB  42.9GB  primary  xfs
```

```
[root@host ~]# parted /dev/vda unit s print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start      End         Size       Type     File system  Flags
 1      2048s      20971486s   20969439s  primary  xfs          boot
 2      20971520s  104857535s  83886016s  primary  xfs
```

- you can also specify multiple subcommands (like unit and print) on the same line.
  - s for sector
  - B for byte
  - MiB , GiB , or TiB (powers of 2)
  - MB , GB , or TB (powers of 10)

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.4
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

- Use the mkpart subcommand to create a primary or extended partition.

```
(parted) mkpart
Partition type?  primary/extended? primary
```

- If you need more than four partitions on an MBR-partitioned disk, then create three primary partitions and one extended partition. The extended partition serves as a container within which you can create multiple logical partitions.

- Indicate the file-system type that you want to create on the partition, such as xfs or ext4. However, this value does not create the file system, but it is only a useful partition type label.

```
File system type?  [ext2]? xfs
```

- Specify the disk sector to start the new partition on.

```
Start? 2048s
```

- Specify the disk sector where the new partition should end, and exit parted. You can specify the end as a size or as an ending location(sector).

```
End? 1000MB
(parted) quit
Information: You may need to update /etc/fstab.
```

- As an alternative to interactive mode, you can create a partition in a single command:

```
[root@host ~]# parted /dev/vdb mkpart primary xfs 2048s 1000MB
```

With most disks, a start sector that is a multiple of 2048 is safe.

- Creating GPT partitions essentially follows the same steps, however with GPT, partitions are given names, unlike MBR where partitions are allocated names automatically based on the disk (i.e if the disk is sda, and you created 3 partitions, they will be name sda1,2,3).

Command or option	**Description**
Udevadm settle	waits for the system to detect the new partition and to create the associated device file in the /dev directory.
mkfs.xfs dev/vda1	apply an XFS file system to a block device
mkfs.ext4 dev/vda1	apply an ext4 file system to a block device
Mount dev/vda1 /mnt	manually attach a device to a mount point directory location
Mount | grep vda1	view currently mounted file systems, the mount points, and their options.

To configure the system to automatically mount the file system during system boot, add an entry to the /etc/fstab file. This configuration file lists the file systems to mount at system boot.

Deleting Partitions

- The following instructions apply for both the MBR and GPT partitioning schemes.

1. Specify the disk that contains the partition to remove.
2. Run the parted command with the disk device as the only argument.
3. Identify the partition number of the partition to delete.
4. Delete the partition, and exit parted.

The rm subcommand immediately deletes the partition from the partition table on the disk.

```
(parted) rm 1
(parted) quit
Information: You may need to update /etc/fstab.
```

```
[root@host ~]# parted /dev/vdb rm 1
```

RAM	Swap space	Swap space if allowing for hibernation
2 GB or less	Twice the RAM	Three times the RAM
Between 2 GB and 8 GB	Same as RAM	Twice the RAM
Between 8 GB and 64 GB	At least 4 GB	1.5 times the RAM
More than 64 GB	At least 4 GB	Hibernation is not recommended

The laptop and desktop hibernation function uses the swap space to save the RAM contents before powering off the system. When you turn the system back on, the kernel restores the RAM contents from the swap space and does not need a complete boot. For those systems, the swap space must be greater than the amount of RAM.

Create Swap Space

1. Create a partition with a file-system type of linux-swap.
2. Place a swap signature on the device.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.4
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
 1      1049kB  1001MB  1000MB               data

(parted) mkpart
Partition name?  []? swap1
```

```
Partition name?  []? swap1
File system type?  [ext2]? linux-swap
Start? 1001MB
End? 1257MB
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system     Name   Flags
 1      1049kB  1001MB  1000MB                  data
 2      1001MB  1257MB  256MB   linux-swap(v1)  swap1

(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

- The mkswap command applies a swap signature to the device. Unlike other formatting utilities, the mkswap command writes a single block of data at the beginning of the device, leaving the rest of the device unformatted so that the kernel can use it for storing memory pages.

```
[root@host ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 244 MiB (255848448 bytes)
no label, UUID=39e2667a-9458-42fe-9665-c5c854605881
```

Activate Swap Space

- You can use the swapon command to activate a formatted swap space.
- Use swapon with the device as a parameter, or use swapon -a to activate all the listed swap spaces in the /etc/fstab file. Use the swapon --show and free commands to inspect the available swap spaces.

```
[root@host ~]# free
               total        used        free      shared  buff/cache   available
Mem:         1873036      134688     1536436       16748      201912     1576044
Swap:              0           0           0
[root@host ~]# swapon /dev/vdb2
[root@host ~]# free
               total        used        free      shared  buff/cache   available
Mem:         1873036      135044     1536040       16748      201952     1575680
Swap:         249852           0      249852
```

- A swap space can be deactivated with the swapoff command. If pages are written to the swap space, then the swapoff command tries to move those pages to other active swap spaces or back into memory.
- If the swapoff command cannot write data to other places, then it fails with an error, and the swap space stays active.
  - The following example shows a typical line in the /etc/fstab file based on the previously created swap space.

```
UUID=39e2667a-9458-42fe-9665-c5c854605881   swap   swap   defaults   0 0
```

  - By default, the system uses swap spaces in series, meaning that the kernel uses the first activated swap space until it is full, and then it starts using the second swap space.
  - You can instead define a priority for each swap space to force a particular order.
  - To set the priority, use the pri option in the /etc/fstab file. The kernel uses the swap space with the highest priority first. The default priority is -2.

```
UUID=af30cbb0-3866-466a-825a-58889a49ef33   swap   swap   defaults  0 0
UUID=39e2667a-9458-42fe-9665-c5c854605881   swap   swap   pri=4     0 0
UUID=fbd7fa60-b781-44a8-961b-37ac3ef572bf   swap   swap   pri=10    0 0
```

  - The kernel uses the last entry first, because its priority is set to 10. When that space is full, it uses the second entry, because its priority is set to 4. Finally, it uses the first entry, which has a default priority of -2.
  - Use the **swapon** --show command to display the swap space priorities.

Chapter 8: Manage Storage Stack

LVM hides the hardware storage configuration from the software and enables you to resize volumes without stopping applications or unmounting file systems. LVM is used to create logical storage volumes as a layer on the physical storage. This storage system provides greater flexibility than using physical storage directly.

Step 1 - Prepare physical devices (creating the block devices)

- Partitioning is optional when already present. Use the parted command to create a new partition on the physical device. Set the physical device to the Linux LVM partition type. Use the udevadm settle command to register the new partition with the kernel.

```
[root@host ~]# parted /dev/vdb mklabel gpt mkpart primary 1MiB 769MiB
...output omitted...
[root@host ~]# parted /dev/vdb mkpart primary 770MiB 1026MiB
[root@host ~]# parted /dev/vdb set 1 lvm on
[root@host ~]# parted /dev/vdb set 2 lvm on
[root@host ~]# udevadm settle
```

Step 2 - Create Physical Volumes

- Use the pvcreate command to label the physical partition as an LVM physical volume.
- Label multiple devices simultaneously by using space-delimited device names as arguments to the pvcreate command. This example labels the /dev/vdb1 and /dev/vdb2 devices as PVs that are ready for creating volume groups.

```
[root@host ~]# pvcreate /dev/vdb1 /dev/vdb2
  Physical volume "/dev/vdb1" successfully created.
  Physical volume "/dev/vdb2" successfully created.
  Creating devices file /etc/lvm/devices/system.devices
```

Step 3 - Create a Volume Group

- The vgcreate command builds one or more physical volumes into a volume group. The first argument is a volume group name, followed by one or more physical volumes to allocate to this VG.

```
[root@host ~]# vgcreate vg01 /dev/vdb1 /dev/vdb2
  Volume group "vg01" successfully created
```

- This example creates the vg01 VG using the /dev/vdb1 and /dev/vdb2 PVs.

Step 4 - Create a Logical Volume

- The lvcreate command creates a new logical volume from the available PEs in a volume group.
- Use the lvcreate command to set the LV name by -n option and size by -L option and the VG name that contains this LV.
- This example creates lv01 LV with 300 MiB in size in the vg01 VG.

```
[root@host ~]# lvcreate -n lv01 -L 300M vg01
  Logical volume "lv01" created.
```

- This command might fail if the volume group does not have sufficient free physical extents.
- The LV size rounds up to the next value of PE size when the sizeb does not exactly match.
- The lvcreate command -L option requires sizes in bytes, mebibytes (binary megabytes, 1048576 bytes), and gibibytes (binary gigabytes), or similar.
- The lower case -l requires sizes specified as a number of physical extents.
- The following commands are two choices for creating the same LV with the same size:
  - lvcreate -n lv01 -L 128M vg01 : create an LV of size 128 MiB, rounded to the next PE.
  - lvcreate -n lv01 -l 32 vg01 : create an LV of size 32 PEs at 4 MiB each is 128 MiB.
- After creating the LV, you have to create a filesystem for it.
- Specify the LV by using either the /dev/vgname/lvname traditional name, or the /dev/mapper/vgname-lvname kernel device mapper name.
- Use the mkfs command to create a file system on the new LV.

```
[root@host ~]# mkfs -t xfs /dev/vg01/vdo-lv01
...output omitted...
```

- Mount the LV by using the mount command.

```
[root@host ~]# mkdir /mnt/data
```

- To make the file system available persistently, add an entry to the /etc/fstab file.

```
/dev/vg01/vdo-lv01 /mnt/data xfs defaults 0 0
```

- Do not forget to run the command findmnt --verify or manually mount before rebooting the system

- Mount the LV by using the mount command.

```
[root@host ~]# mount /mnt/data/
```

Use the pvdisplay, vgdisplay, and lvdisplay commands to show the status information of the LVM components.

- The associated pvs, vgs, and lvs commands are commonly used and show a subset of the status information, with one line for each entity.

Extend a Volume Group Size

- The vgextend command adds the new PV to the VG. Provide the VG and PV names as arguments to the vgextend command.

```
[root@host ~]# vgextend vg01 /dev/vdb3
  Volume group "vg01" successfully extended
```

- Use the vgdisplay command to confirm that the volume group has sufficient free space for the LV extension. Use the lvextend command to extend the LV.

```
[root@host ~]# lvextend -L +500M /dev/vg01/lv01
  Size of logical volume vg01/lv01 changed from 300.00 MiB (75 extents) to 800.00 MiB (200 extents)
  Logical volume vg01/lv01 successfully resized.
```

- The (+) means addition to the existing size, without the plus sign, the value defines the final size of the LV.
- The -l option expects the number of PE as the argument. While the -L option expects sizes in bytes, mebibytes, gibibytes, and similar.

Extend an XFS File System to the LV Size

- The xfs_growfs command helps expand the file system to occupy the extended LV. The target file system must be mounted before you use the xfs_growfs command.
- You can continue to use the file system while resizing.

```
[root@host ~]# xfs_growfs /mnt/data/
...output omitted...
data blocks changed from 76800 to 204800
```

Extend Swap Space Logical Volumes

- You can extend the LVs used as swap space; however, the process differs from expanding the ext4 or XFS file system.
- LVs used as swap space must be offline to extend them.
- Use the swapoff command to deactivate the swap space on the LV.

```
[root@host ~]# swapoff -v /dev/vg01/swap
swapoff /dev/vg01/swap
```

- Use the lvextend command to extend the LV.

```
[root@host ~]# lvextend -L +300M /dev/vg01/swap
  Size of logical volume vg01/swap changed from 500.00 MiB (125 extents) to 800.00 MiB (200 extents).
  Logical volume vg01/swap successfully resized.
```

- Use the mkswap command to format the LV as swap space.

```
[root@host ~]# mkswap /dev/vg01/swap
mkswap: /dev/vg01/swap: warning: wiping old swap signature.
Setting up swapspace version 1, size = 800 MiB (838856704 bytes)
no label, UUID=25b4d602-6180-4b1c-974e-7f40634ad660
```

- Use the swapon command to activate the swap space on the LV.

```
[root@host ~]# swapon /dev/vg01/swap
```

- Reducing a VG involves removing unused PV from the VG.
- The pvmove command moves data from extents on one PV to extents on another PV with enough free extents in the same VG.
- You may continue to use the LV from the same VG while reducing. Use the pvmove command -A option to automatically backup the metadata of the VG after a change. This option uses the vgcfgbackup command to backup metadata automatically.

```
[root@host ~]# pvmove -A y /dev/vdb3
```

- Use the vgreduce command to remove a PV from a VG.

```
[root@host ~]# vgreduce vg01 /dev/vdb3
  Removed "/dev/vdb3" from volume group "vg01"
```

- Use the lvremove, vgremove, and pvremove commands to remove an LVM component that is no longer required.
- Use the umount command to unmount the file system and then remove any /etc/fstab entries associated with this file system.

Chapter 9: Network Attached storage

- The *Network File System* (NFS) is an internet standard protocol that Linux, UNIX, and similar operating systems use as their native network file system. NFS is an open standard that supports native Linux permissions and file-system attributes.

- You must install the nfs-utils package to obtain the client tools for manually mounting, or for automounting, to obtain exported NFS directories.

NFSv3	NFSv4
Supports TCP or UDP as a transport protocol.	Supports TCP transport protocol only.
Uses **RPC** protocol to require a file from NFSv3 server. An NFSv3 client connects to the **rpcbind** service at port **111** on the server to request NFS service. The server responds with the current port for the NFS service.	Simpler than NFSv3 server. It introduces an ***export tree*** that contains all of the paths for the server's exported directories.

- Use the showmount command to query the available exports on an RPC-based NFSv3 server. while running this command on an NFSv4 server times out without receiving a response, because the rpcbind service is not running on the server.
- To view all of the exported directories for NFSv4, mount the root (/) of the server's export tree.

```
[root@host ~]# mkdir /mountpoint
[root@host ~]# mount server:/ /mountpoint
[root@host ~]# ls /mountpoint
```

```
[root@host ~]# mount -t nfs -o rw,sync server:/export /mountpoint
```

- The -t nfs specifies the NFS file-system type. However, when the mount command detects the *server:/export* syntax, the command defaults to the NFS type.
- The -o is to add a list of comma-separated options to the mount command such as rw option to mount with read/write access. The sync specifies that transactions must be completed or else return as failed.

you can mount the NFS export by using only the mount point. The mount command obtains the NFS server and mount options from the matching entry in the /etc/fstab file.

```
server:/export  /mountpoint  nfs  rw  0 0
```

- The lsof command returns a list of open file names and the process which is keeping the file open.
- umount -f option is used to force the unmount, which can cause loss of unwritten data for all open files.

Chapter 13: Run Containers

- A Container is a lightweight and portable method of packaging and running applications

- An image is a read-only template or snapshot of a containerized application and its dependencies. It contains the necessary files, libraries, and configurations required to run the application within a container. Images are created by a Dockerfile or similar instructions, stored in registries or locally.
- Container files refer to the files and directories that constitute a container. These files are typically derived from the image and include the application executable, libraries, configurations, and any additional files required for the application to run within the container.
- Container registries are repositories that store and distribute container images. They serve as centralized locations where images can be uploaded, downloaded, and shared. **Docker Hub** is a popular public container registry, while private registries can be set up for internal use. Container registries enable easy distribution and versioning of container images across different environments.
- Rootless container runtimes, like Podman, allow non-root users to create, manage, and run containers securely. This enhances security by reducing the attack surface and minimizing the risks associated with running containers as the root user.
- By default, Docker operates with root privileges, but it is also possible to configure Docker to run in rootless mode. Rootful containers offer more control and access to system resources, but they come with increased security considerations.

Feature	VMs	Containers
Architecture	Run on **Hypervisor** emulating an entire computer system, including OS, on a physical host machine maintaining separate resources, such as CPU, memory, and disk space.	Run on **Container engine** sharing the host machine's OS kernel and resources allowing multiple containers to run concurrently on the same host with less system resources consumption.
Size	Measured in Gigabytes.	Measured in Megabytes.
Startup Time	VMs take longer to start because they require booting a full OS.	Start up quickly as they only need to launch the specific application and its dependencies, without booting an entire OS.
Isolation	Provide strong isolation between multiple instances, as each VM has its own OS and runs independently.	Offer process-level isolation sharing the host machine's kernel. Containers may have limitations for certain highly sensitive or security-critical applications.
Portability	More portable since they encapsulate the entire OS and application stack. They can be migrated between different hypervisors or cloud platforms with minimal compatibility issues.	Highly portable due to being lightweight and reliance on shared host resources. They can be easily moved between different container environments that support containerization, regardless of the underlying infrastructure.
Ecosystem and Maturity	Widely adopted and has a mature ecosystem with established management tools, such as VMware and Hyper-V.	Gained significant popularity in recent years. The container ecosystem offers a wide range of tools and platforms for management, orchestration, and deployment.

- Podman is a container runtime and management tool that provides a command-line interface for running, building and managing containers and containers images.
- It is an open-source and supports the core containerization technologies provided by Red Hat, such as cgroups, SELinux, and Seccomp.
- Podman is daemonless which improves security by not requiring elevated privileges and simplifies the management of containers directly.
- Podman offers features like pod management, allowing users to group multiple containers together for easier orchestration.

Feature	Podman	Docker
Architecture	Daemonless, eliminating the need for a central daemon process. It operates as a command-line tool, directly interacting with the container, leveraging individual containers as lightweight processes.	A client-server architecture, with a daemon (dockerd) runs as a background service, managing container lifecycle operations. The client (docker) interacts with the daemon to execute commands and manage containers.
Security	More secure without a central daemon. It operates as an unprivileged user reducing the attack. Podman integrates with SELinux and supports rootless containers.	Requires privileged access, which can introduce potential security risks. Running Docker commands often requires elevated privileges.
Image Storage	Uses various image registries, including Docker Hub, also supporting local image repositories or image archives, offering more flexibility and control over image storage.	Uses a central image registry, with Docker Hub being the default public registry.
Compatibility	Strives for compatibility with Docker, supporting Dockerfiles, Docker images, and Docker Compose files.	Industry-standard containerization tool for a long time.
Orchestration	Focused on running individual containers, but it can work well with external orchestration tools like Kubernetes.	Has a built-in orchestration tool called Docker Swarm, which enables container clustering and service orchestration.

Command or option	**Description**
Podman -v	Check podman version
podman pull	used to fetch container images from image registries
podman images	List available images in the system
podman run	create a new container that uses the image |--name assigns a name to a container. -p option is used to map a port in the local machine to a port inside the container. -d option detached mode |-it (interactive terminal)
podman ps	List all running containers | add -all to view running and stopped
podman exec <name> <command>	**Runs a command inside a running container.**
podman logs <container name>	Displays the logs of a container.
podman inspect <container name>	Provides detailed information about a container
podman stop <name>	Stops a running container.
podman rm <name>	Removes a stopped container.
podman build	used to build the image | -t option to provide the name and tag for the new image.

- You must set the container_file_t SELinux context type before you can mount the directory as persistent storage to a container. If the directory does not have the container_file_t SELinux context, then the container cannot access the directory.
- You can append the Z option to the argument of the podman run command -v option to automatically set the SELinux context on the directory.

For example, **podman run -v /home/user/dbfiles:/var/lib/mysql:Z** command to set the SELinux context for the /home/user/dbfiles directory when you mount it as persistent storage for the /var/lib/mysql directory.