**Modue: 5- Linux server - Deploy, configure, and maintain systems Assignment**

**38. Schedule Tasks Using cron or at**

cron:

Purpose: cron is used for scheduling repetitive tasks.

Configuration: Tasks are defined in a file called crontab.

Syntax: Each line in a crontab file represents a task and follows this format:

\* \* \* \* \* command\_to\_execute

- - - - -

| | | | |

| | | | +---- Day of the week (0 - 7) (Sunday is both 0 and 7)

| | | +------ Month (1 - 12)

| | +-------- Day of the month (1 - 31)

| +---------- Hour (0 - 23)

+------------ Minute (0 - 59) 😂😂😂😂😂😂😂😂😂😂(just kiding)

Example: To run a script every day at 2 AM:

0 2 \* \* \* /path/to/script.sh

at:

Purpose: at is used for scheduling one-time tasks.

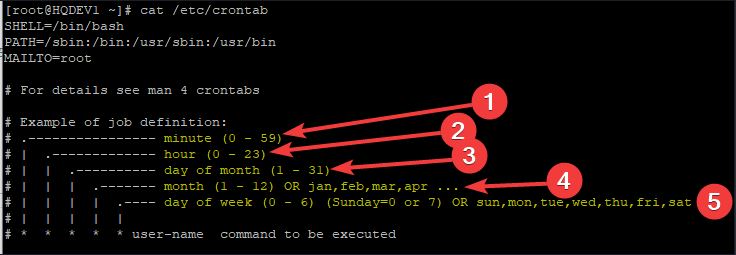
Usage: You can schedule a task to run at a specific time.

Syntax:

echo "command\_to\_execute" | at time

Example: To run a script at 2 PM tomorrow:

echo "/path/to/script.sh" | at 2pm tomorrow



**39. Use apt or yum to Install, Update, and Remove Software Packages**

apt (for Debian-based distributions like Ubuntu):

Update Package List:

sudo apt update

Upgrade All Packages:

sudo apt upgrade

Install a Package:

sudo apt install package\_name

Remove a Package:

sudo apt remove package\_name

yum (for Red Hat-based distributions like CentOS):

Update Package List:

sudo yum check-update

Upgrade All Packages:

sudo yum update

Install a Package:

sudo yum install package\_name

Remove a Package:

sudo yum remove package\_name

**40. Install the httpd Package**

For Debian-based distributions (using apt):

sudo apt update

sudo apt install apache2

For Red Hat-based distributions (using yum):

sudo yum install httpd

For offline:

We need to use existing image for copy httpd packages.

So First we need to connect Disk images that we have or connected and connect it to the linux that we are using.

Go to disk packages and open rpm’s and search for all httpd rpm’s.

Follow the commands:

rpm -ivh httpd\* (this will install all httpd packages)

if not work then try,

rpm -ivh httpd\* --force -nodeps (forcefully and nodependency)

**41. Open Kickstart Configuration Graphically**

Kickstart is used for automating the installation of Red Hat-based distributions. To open the Kickstart configuration tool graphically:

To open the Kickstart Configurator graphically on a Red Hat Enterprise Linux system, follow these steps:

Install Kickstart Configurator:

Open a terminal.

Run the following command to install the Kickstart Configurator:

sudo yum install system-config-kickstart

Launch Kickstart Configurator:

Boot your system into a graphical environment.

You can launch the Kickstart Configurator by running:

system-config-kickstart

Alternatively, you can navigate through the graphical interface:

On GNOME desktop: Click on Applications > System Tools > Kickstart.

On KDE desktop: Click on Kickoff Application Launcher > Applications > System > Kickstart.

Install the Kickstart Configurator:

sudo yum install system-config-kickstart

Run the Kickstart Configurator:

sudo system-config-kickstart

**42. Configure a New Kickstart File**

Open the Kickstart Configurator:

sudo system-config-kickstart

Configure Installation Settings:

Basic Configuration: Set the installation method, language, keyboard layout, etc.

Installation Method: Choose between CD-ROM, NFS, FTP, HTTP, or Hard Drive.

Partition Information: Define how the disk should be partitioned.

Network Configuration: Set up network settings.

Authentication: Configure authentication methods.

Package Selection: Choose which packages to install.

Save the Configuration:

Save the file with a .ks extension, e.g., my\_kickstart.ks.

**43. Show Full Configuration of New Kickstart File**

To view the full configuration of your Kickstart file:

Open the Kickstart File:

cat /path/to/your\_kickstart\_file.ks

Review the Contents: The file will contain all the settings you configured, including partitioning, package selection, and post-installation scripts.

# Kickstart file automatically generated by anaconda.

#version=RHEL8

install

cdrom

lang en\_US.UTF-8

keyboard us

network --bootproto=dhcp --device=eth0

rootpw --iscrypted $6$example$examplepassword

firewall --enabled --service=ssh

authconfig --enableshadow --passalgo=sha512

selinux --enforcing

timezone America/New\_York --isUtc

bootloader --location=mbr --boot-drive=sda

# Partitioning

clearpart --all --initlabel

part /boot --fstype="xfs" --size=1024

part pv.01 --grow --size=1

volgroup vg0 --pesize=4096 pv.01

logvol / --fstype="xfs" --name=root --vgname=vg0 --size=10240

logvol swap --fstype="swap" --name=swap --vgname=vg0 --size=2048

%packages

@core

%end

%post

echo "Kickstart installation complete"

%end

Explanation of Key Sections:

Installation Method: Specifies the installation source (e.g., cdrom).

Language and Keyboard: Sets the language and keyboard layout.

Network Configuration: Configures network settings.

Root Password: Sets the root password (encrypted).

Firewall and SELinux: Configures firewall and SELinux settings.

Timezone: Sets the system timezone.

Bootloader: Configures the bootloader.

Partitioning: Defines disk partitioning and logical volumes.

Package Selection: Specifies which packages to install.

Post-installation Script: Runs commands after the installation is complete.

**44. Validate New Kickstart File**

To validate a Kickstart file, you can use the ksvalidator tool, which is part of the pykickstart package. This tool checks the syntax of the Kickstart file to ensure it is correct.

Install pykickstart:

sudo yum install pykickstart

Validate the Kickstart File:

ksvalidator /path/to/kickstart.ks

Replace /path/to/kickstart.ks with the actual path to your Kickstart file. This command will check the file for syntax errors1.

**45. Allow HTTP on Firewall**

To allow HTTP traffic through the firewall using firewalld, follow these steps:

Check FirewallD Status:

sudo systemctl status firewalld

Start FirewallD if Not Running:

sudo systemctl start firewalld

Allow HTTP Service:

sudo firewall-cmd --zone=public --add-service=http --permanent

Allow HTTPS Service (Optional):

sudo firewall-cmd --zone=public --add-service=https --permanent

Reload FirewallD to Apply Changes:

sudo firewall-cmd --reload

These commands will ensure that HTTP (and optionally HTTPS) traffic is allowed through the firewall2.

**46. Reload Firewall**

Reloading the firewall applies any changes made to the configuration:

Reload FirewallD:

sudo firewall-cmd --reload

This command reloads the firewall rules and applies any changes made to the permanent configuration3.

**47. Start and Restart HTTP**

To manage the HTTP service (Apache) on your system, use the following commands:

Start HTTP Service:

sudo systemctl start httpd

Restart HTTP Service:

sudo systemctl restart httpd

Check HTTP Service Status:

sudo systemctl status httpd

These commands will start and restart the Apache HTTP server, ensuring it is running with the latest configuration4.

**48. Install New Foundation Using New Kickstart File**

To perform a Kickstart installation using your new Kickstart file, follow these steps:

Create Boot Media:

Prepare a bootable USB drive or DVD with the installation media.

Ensure the Kickstart file is accessible on the media or via a network location.

Boot the System:

Boot the target system using the prepared boot media.

At the boot prompt, specify the location of the Kickstart file using the inst.ks boot option. For example:

linux inst.ks=hd:LABEL=USB:/path/to/kickstart.ks

Replace LABEL=USB and /path/to/kickstart.ks with the appropriate values for your setup.

Start the Installation:

The system will automatically start the installation process using the settings specified in the Kickstart file.

Monitor the Installation:

Monitor the installation process to ensure it completes successfully. The system will reboot once the installation is finished.