**Module: 5-**

**Linux server - Deploy, configure, and maintain systems**

**Assignment**

38. · Schedule tasks using cron or at.

ANS:

Scheduling tasks in Linux can be done using two main utilities: cron and at. Here’s a brief overview of each:

**Using**cron

cron is used for scheduling repetitive tasks. It runs in the background and executes commands at specified intervals.

**Steps to Use**cron**:**

1. **Edit the crontab file**:
2. crontab -e
3. **Add a cron job**: The format is:
4. \* \* \* \* \* command

Each \* represents a time field (minute, hour, day of month, month, day of week). For example, to run a script every day at 2 AM:

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

1. **Save and exit**: The job will now run at the specified times.

**Using**at

at is used for scheduling one-time tasks.

**Steps to Use**at**:**

1. **Schedule a task**:
2. at 2pm
3. **Enter the command** you want to run at the specified time:
4. /path/to/script.sh
5. **Press**Ctrl+D to save the job.

**Examples**

* **Cron Example**: To run a backup script every Sunday at midnight:
* 0 0 \* \* 0 /path/to/backup.sh
* **At Example**: To run a maintenance script tomorrow at 3 PM:
* echo "/path/to/maintenance.sh" | at 3pm tomorrow

[These tools are powerful for automating tasks and can save you a lot of time](https://www.redhat.com/sysadmin/linux-cron-command)

39. · Use apt or yum (depending on your Linux distribution) to install, update, and remove

ANS:

1. **Install a package**:

**sudo yum install package name**

1. **Update package lists**:

**sudo yum check-update**

1. **Upgrade all installed packages**:

**sudo yum update**

1. **Remove a package**:

**sudo yum remove package name**

[These commands will help you manage software packages on your Linux system efficientl](https://phoenixnap.com/kb/yum-vs-apt)

**software packages.**

40. Install all httpd package

ANS:

1. Open the terminal.
2. Run the following command to install httpd:

sudo yum install httpd

1. Enable and start the httpd service:

sudo systemctl enable httpd

sudo systemctl start httpd

41. Open kickstart configuration graphically

1. ANS: **Install Kickstart Configurator**:
   * On **Red Hat Enterprise Linux**  you can install it using:

**sudo yum install system-config-kickstart**

1. **Launch Kickstart Configurator**:
   * Boot your system into a graphical environment.
   * Run the following command in the terminal:

**sudo system-config-kickstart**

* + Alternatively, you can navigate through the graphical interface:
    - On **GNOME desktop**: Go to Applications → System Tools → Kickstart.
    - On **KDE desktop**: Go to Kickoff Application Launcher → Applications → System → Kickstart.

42. Configure new kickstart file

ANS:

A kickstart configuration file contains all the information that kickstart requires to perform an automated installation. Every Linux installation creates a kickstart file, /root/anaconda-ks.cfg. can use this file to repeat an installation, or can customize the settings in this file for different system configurations. The file is also useful for troubleshooting a boot-time problem with an installed system. can use the Kickstart Configurator (**system-config-kickstart**) to create or modify a kickstart file.

43. Show full configuration of new kickstart file

ANS: # Kickstart file for Red Hat Enterprise Linux

# Generated by Kickstart Configuration Tool

# System language

lang en\_US.UTF-8

# Keyboard layout

keyboard us

# Network information

network --bootproto=dhcp

# Root password (encrypted)

rootpw --iscrypted $6$...

# System timezone

timezone America/New\_York

# Firewall configuration

firewall --enabled --service=ssh

# Disk partitioning

part / --fstype=xfs --size=10240

part swap --size=2048

# Package selection

%packages

@base

@core

@server

@workstation-product-environment

kexec-tools

# Installation logging

%addon com\_redhat\_kdump --enable --reserve-mb=auto

# System bootloader configuration

bootloader --location=mbr

# Reboot after installation

reboo # Kickstart file for Red Hat Enterprise Linux 8

# Generated by Kickstart Configuration Tool

# System language

lang en\_US.UTF-8

# Keyboard layout

keyboard us

# Network information

network --bootproto=dhcp

# Root password (encrypted)

rootpw --iscrypted $6$...

# System timezone

timezone America/New\_York

# Firewall configuration

firewall --enabled --service=ssh

# Disk partitioning

part / --fstype=xfs --size=10240

part swap --size=2048

# Package selection

%packages

@base

@core

@server

@workstation-product-environment

kexec-tools

# Installation logging

%addon com\_redhat\_kdump --enable --reserve-mb=auto

# System bootloader configuration

bootloader --location=mbr

# Reboot after installation

reboott

44. Validate new kickstart file

ANS:

Use the ksvalidator command line utility to verify that your Kickstart file is valid. This is useful when you make extensive changes to a Kickstart file.

Procedure

* Run ksvalidator on your Kickstart file:

$ ksvalidator */path/to/kickstart.ks*

Replace */path/to/kickstart.ks* with the path to the Kickstart file you want to verify.

45. All http on firewall

1. ANS: **Check FirewallD Status:**
   * First, ensure that **FirewallD** is installed and running on your system. You can check its status by executing the following command in your terminal:

**sudo systemctl status firewalld**

* + If it’s not running, start it with:

**sudo systemctl start firewalld**

1. **Allow HTTP and HTTPS Services:**
   * **HTTP (port 80)**: To allow HTTP traffic, add the HTTP service to your active zone. Typically, the default zone is set to ‘public’. You can check your active zone by typing:
   * **firewall-cmd --get-active-zones**

Then, add the HTTP service to your active zone:

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

* + **HTTPS (port 443)**: Similarly, to allow HTTPS traffic, use the following command:

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

1. **Make Changes Permanent:**
   * By default, changes are temporary and will be lost after a reboot. To make them permanent, add the --permanent flag to the commands:
   * **sudo firewall-cmd --permanent --zone=public --add-service=http**
   * **sudo firewall-cmd --permanent --zone=public --add-service=https**
2. **Reload FirewallD:**
   * After adding the services, reload FirewallD to apply the changes:
   * **sudo firewall-cmd --reload**
3. **Verify Configuration:**
   * Confirm that the services have been added correctly. Check the services enabled on your zone:
   * **sudo firewall-cmd --zone=public --list-services**
   * You should see http and https in the list of allowed services.

46. Reload firewall.

ANS:

To reload your firewall configuration in Linux using **firewalld**, you can execute the following command:

**sudo firewall-cmd --reload**

[This will reload the firewall rules while preserving state information, ensuring that your permanent configuration becomes the new runtime configuration](https://www.redhat.com/sysadmin/firewalld-linux-firewall)

If you’ve made any changes to your firewall settings, this command will apply them immediately

47. Start and restart http

ANS: To start, stop, or restart the Apache HTTP Server, you have a few options depending on your system. Here are the commands for different scenarios:

1. **Using systemd (systemctl):**
   * **Start: $ sudo systemctl start httpd**
   * Stop: $ sudo systemctl stop httpd
   * **Restart: $ sudo systemctl restart httpd**
   * Check status: $ sudo systemctl status httpd

48. Install new foundation using new kickstart file

ANS:

1. **Create a Kickstart File**:
   * The Kickstart file is a plain text file containing installation instructions. You can create it manually or use the GUI tool (system-config-kickstart) to generate one.
   * The recommended approach is to perform a manual installation on one system first. [After installation, find the **anaconda-ks.cfg** file in **/root/** and use it as a template for your Kickstart file](https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/installation_guide/sect-kickstart-howto)
2. **Make the Kickstart File Available**:
   * Copy your Kickstart file to a location accessible by the installation program. [This can be a local DVD, hard drive, or a network location (NFS, FTP, HTTP, or HTTPS)](https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/installation_guide/sect-kickstart-howto)
3. **Create Boot Media**:
   * Prepare bootable media (USB, CD/DVD, or PXE boot) that will initiate the installation process.
   * [Specify the path to your Kickstart file during boot to automate the installation](https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/installation_guide/sect-kickstart-howto)
4. **Make the Installation Source Available**:
   * Ensure the installation source (ISO image or repository) is accessible. [For example, if you’re using NFS, copy the DVD1 ISO image to a directory on the NFS server](https://docs.centos.org/en-US/8-docs/advanced-install/assembly_creating-installation-sources-for-kickstart-installations/)
5. **Start the Kickstart Installation**:
   * Boot the system using the boot media.
   * [The installer will read your Kickstart file and perform the installation based on the specified options](https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/installation_guide/sect-kickstart-howto)

**Module 6-**

**Linux server - Manage basic networking & Security**

49. Use ifconfig or ip to view and configure network interfaces.

ANS:

**ifconfig**:

* The ifconfig command displays information about network interfaces, including their IP addresses, MAC addresses, and other relevant details.
* The**ip** command is more powerful and versatile. It provides detailed information about network interfaces, routes, and more.
* To list all interfaces and their details, use:

ip addr show

50. Use ping to test network connectivity.

ANS:

**ping** helps troubleshoot network issues and determine if a device can reach another device on your local network or the internet.

ping 8.8.8.8

51. Understand basic firewall configuration using FIREWALL-CMD.

ANS:

**Firewalld** is a dynamic firewall management tool for Linux systems. It allows you to control network traffic by defining security zones, services, and custom rules. Let’s explore some basic **firewall-cmd** commands:

1. **Check Firewall Status**: To verify if **firewalld** is running, use:

**sudo firewall-cmd --state**

1. **View Zones**: Each network interface is associated with a specific zone (e.g., “public,” “internal”). List all zones with:

**sudo firewall-cmd --list-all --zone=public**

1. **Allow a Port**: To permit traffic through a specific port (e.g., SSH), use:

**sudo firewall-cmd --add-port=22/tcp --permanent**

1. **Reload Firewalld**: After making changes, reload the firewall:

**sudo firewall-cmd --reload**

1. **Block an IP Address**: To block an IP, use:

**sudo firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.168.1.100" reject'**

52. Add ssh services in firewall

ANS:

To open the SSH port (default is port 22) using **firewalld**, you have a couple of options:

1. **Using the**--add-service**Command**:
   * To allow SSH traffic, use:
   * **sudo firewall-cmd --zone=public --add-service=ssh --permanent**

This command adds the SSH service to the firewall rules permanently.

53. Graphicallymanagethefirewall

ANS:

1. **GUFW (Graphical Uncomplicated Firewall)**:
   * GUFW provides a user-friendly GUI for managing the **Uncomplicated Firewall (UFW)**.
   * Install GUFW using the following command:
   * **sudo apt install gufw**
   * [Once installed, you can access GUFW from your applications menu and configure your firewall settings through its intuitive interface](https://linuxconfig.org/ubuntu-24-04-firewall-a-quick-guide)
2. **Firewall-config (FirewallD)**:
   * If you’re using **FirewallD** (firewalld) on CentOS or Fedora, you can use the **firewall-config** tool.
   * It’s a graphical utility for managing firewall rules, zones, and services.
   * Launch it from the menu or by running:

**firewall-config**

55. What is selinux Security

**SELinux** (Security-Enhanced Linux) is a **security architecture** for Linux systems that provides **fine-grained access control**. Here are the key points about SELinux:

1. **Origins and Purpose**:
   * Developed by the United States National Security Agency (NSA).
   * Originally a series of patches to the Linux kernel using Linux Security Modules (LSM).
   * Released to the open-source community in 2000 and integrated into the upstream Linux kernel in 2003.
2. **How SELinux Works**:
   * Defines **access controls** for applications, processes, and files.
   * Uses **security policies** (rules) to enforce allowed access.
   * When an application requests access to an object (e.g., a file), SELinux checks permissions.
   * If necessary, it consults a security server to make access decisions based on security context.

56. How to Set Static IP in Linux?

ANS:

1. **Identify Your Network Interface**:
   * Open the terminal and use the command **ip addr show** to find your network interface name (e.g., **eth0, enp0s3**).
2. **Edit Network Configuration File**:
   * Create or edit the file **/etc/sysconfig/network-scripts/ifcfg-eth0 (**replace eth0 with your actual interface name).
   * Add the following configuration for a static IP address:
   * DEVICE=eth0
   * BOOTPROTO=none
   * ONBOOT=yes
   * PREFIX=24
   * IPADDR=192.168.2.203
   * GATEWAY=192.168.2.254
   * DNS1=192.168.2.254
   * DNS2=8.8.8.8
   * DEFROUTE=yes
   * IPV4\_FAILURE\_FATAL=no
   * IPV6INIT=no
   * NAME=eth0
   * Adjust the IP address, gateway, and DNS servers as needed.
3. **Restart Network Service**:
   * Apply the changes by running:
   * **systemctl restart network**
4. **Verify Configuration**:
   * Confirm the new settings using:
   * ip a s eth0
   * ip r
   * cat /etc/resolv.conf

ping -c 3 cyberciti.biz

**Module:7-**

**Linux server -deployment of network sevices**

57. What is Difference between LILO And GRUB?

ANS:

1. **GRUB (GNU GRand Unified Bootloader)**:
   * **Function**: GRUB is a modern boot loader that allows users to choose from a list of operating systems during startup.
   * **Supported OS**: GRUB supports multiple operating systems, including Linux, Windows, macOS, and Unix.
   * **Customization**: It offers dynamic configuration changes during boot, allowing users to modify settings.
   * **Features**: High portability, independence from geometry translation, support for various executable formats, and compatibility with multiple file systems.
   * **GUI Support**: Many Linux distributions provide a customized boot menu using GRUB’s GUI capabilities.
2. **LILO (Linux Loader)**:
   * **Function**: LILO is an older boot loader primarily used for Linux-based devices.
   * **Single OS**: It supports only Linux OS.
   * **Interface**: LILO lacks an interactive command interface and graphical menus.
   * **Network Boot**: LILO does not support booting from a network.
   * **Stability**: Despite being outdated, LILO remains reliable for Linux systems.

58. How to Recover Linux Password?

ANS: To recover the root password, you will first boot to the GRUB menu and perform a break at early stage of the boot process. Afterwards, you will remount the **sysroot** directory with read & write access and change the root password on RHEL 8 / CentOS 8 by using the **passwd** command to set the new root password.

59. Which command use for format partition in Linux OS?

ANS: To format a disk partition in Linux, you can use the**mkfs** command. The specific command depends on the desired file system type. Here are the commands for different file systems:

1. **ext4**: Format a partition with the ext4 file system using this command:
2. **sudo mkfs -t ext4 /dev/sdb1**

[Replace /dev/sdb1 with the actual partition you want to format](https://phoenixnap.com/kb/linux-format-disk)

[Again, replace /dev/sdb1 with the correct partition identifier](https://phoenixnap.com/kb/linux-format-disk)

1. **NTFS**: For NTFS file system, run:

**sudo mkfs -t ntfs /dev/sdb1**

60. How to enable “quota” in Linux ?

ANS:

**sudo dnf install quota**

1. **Edit /etc/fstab:**
   * **Open the /etc/fstab file with root privileges using your preferred text editor (e.g., nano):**
   * **sudo nano /etc/fstab**
   * **Add the usrquota option to the line corresponding to your storage partition. If you want to enable quotas for groups as well, include the grpquota option. Example:**
   * **/dev/sda1 /home ext4 defaults,usrquota,grpquota 0 2**
2. **Remount the File System:**
   * **Remount the file system to apply the changes:**

**sudo mount -o remount /home**

61. How to Mount Partition in Linux ?

ANS:

* + Create a directory (mount point) where you want to access the partition. For example:
  + sudo mkdir -p /mnt/my\_partition

1. **Mount the Partition**:
   * Use the mount command to mount the partition. Replace /dev/sdb1 with your actual partition name and /mnt/my\_partition with your desired mount point:
   * sudo mount -t auto /dev/sdb1 /mnt/my\_partition
2. **Check if It’s Mounted**:
   * Verify that the partition is mounted successfully:

df -h /mnt/my\_partition

62. What is use of “mdadm” Command ?

ANS:

The mdadm command is a powerful utility in Linux used for **building, managing, and monitoring software RAID devices** (also known as RAID arrays). Here’s what you can do with mdadm:

1. **Assemble**: Combine components of a pre-existing array into an active array.
2. **Build**: Create an array without per-device metadata (superblocks).
3. **Create**: Set up a new array with per-device metadata.
4. **Follow or Monitor**: Keep an eye on one or more md devices and respond to state changes.
5. **Grow**: Expand (or shrink) an array or reshape it.
6. **Incremental Assembly**: Add a single device to an existing array.
7. **Manage**: Perform actions on specific components (e.g., add spares, remove faulty devices).
8. **Misc**: Handle various tasks related to active arrays, component devices, and information gathering.
9. **Auto-detect**: Request the Linux Kernel to activate any auto-detected arrays.

To install mdadm, use the following commands based on your Linux distribution:

* On CentOS, Fedora, and RHEL:
* **sudo yum install mdadm**

63. How to configure secure Apache web server in Linux ?

ANS:

Configuring a secure Apache web server on a Linux system involves several steps. Below is a step-by-step guide to help you secure your Apache web server.

**1. Install Apache Web Server**

First, you need to ensure Apache is installed on your Linux system. Use the following commands based on your distribution:

**sudo yum install httpd**

**2. Keep Apache Updated**

**Always ensure your Apache server is up-to-date with security patches:**

**sudo yum update**

**3. Configure Basic Security Settings**

**Disable Server Signature and Server Tokens**

**By default, Apache exposes version information, which can be a security risk. Disable it by editing the configuration file.**

* **Open the Apache configuration file:**

**sudo nano /etc/httpd/conf/httpd.conf**

**Add or modify the following lines:**

**ServerSignature Off**

**ServerTokens Prod**

**Restart Apache to apply changes:**

**sudo systemctl restart httpd**

**Disable Directory Listing**

**To prevent attackers from seeing a list of files in directories that don't have an index.html file:**

* **Open the Apache configuration file or the virtual host configuration file:**

**sudo nano /etc/httpd/conf/httpd.conf**

**Edit your virtual host file or main Apache configuration file to include the following headers:**

**sudo systemctl restart httpd**

64. How to run Windows Software on Linux operating System?

ANS:

can run Windows software on Linux using a compatibility layer called **Wine**.

**Install Wine**:

* Most Linux distributions include Wine in their package repositories. You can install it using the following commands:

sudo apt update

sudo apt install wine

* + Wine stands for “Wine Is Not an Emulator.” It’s not a virtual machine; instead, it translates Windows commands to native UNIX/POSIX-compliant logic.
  + To run a Windows program, open a terminal and use:
  + wine /path/to/your/windows-program.exe
  + Replace /path/to/your/windows-program.exe with the actual path to your Windows executable.

65. what is difference between windows and Linux

ANS:

1. **Open Source vs. Proprietary**:
   * **Linux**: An open-source operating system where the source code is freely available for modification.
   * **Windows**: Proprietary, developed by Microsoft, with closed-source code.
2. **Cost**:
   * **Linux**: Free to use.
   * **Windows**: Requires a license and is not free.
3. **Kernel Type**:
   * **Linux**: Monolithic kernel.
   * **Windows**: Hybrid kernel.
4. **User Privileges**:
   * **Linux**: Root user has administration privileges.
   * **Windows**: Administrator account holds administration rights.
5. **Case Sensitivity**:
   * **Linux**: Case-insensitive file system (allows files with the same name but different cases).
   * **Windows**: Case-sensitive (no duplicate filenames with different cases).
6. **Security**:
   * **Linux**: Generally more secure due to package management, repositories, and features.
   * **Windows**: More susceptible to malware and viruses.
7. **Language Support**:
   * **Linux**: Multilingual, supporting around 138 languages.
   * **Windows**: Offers multiple language options.
8. **File System**:
   * **Linux**: Hierarchical file system where everything (including devices) is considered a file.
   * **Windows**: Uses drives (like C, D, E) and stores files within them.
9. **Stability**:
   * **Linux**: More stable; no need for frequent reboots during software installation.
   * **Windows**: May slow down over extended usage.

66.What is the advantage of Open Source?

ANS:

1. **Transparency and Trust**:
   * Open source projects allow users to access the source code. This transparency builds trust, as anyone can review and verify the code for security or quality.
2. **Customization and Flexibility**:
   * Users can modify open source software to suit their specific needs. This flexibility is especially valuable for businesses and developers.
3. **Cost-Effectiveness**:
   * Open source software is typically free to use, reducing licensing costs.
   * Organizations can allocate resources to other areas instead of software licenses.
4. **Community Collaboration**:
   * Open-source projects thrive on community contributions. Developers, users, and enthusiasts collaborate, share knowledge, and improve the software together.
5. **Rapid Innovation**:
   * Open-source projects evolve quickly due to the collective effort of contributors.
   * New features, bug fixes, and enhancements happen at a faster pace.
6. **Vendor Independence**:
   * Users are not tied to a specific vendor or company.
   * They can switch between different implementations or distributions without major disruptions.
7. **Global Accessibility**:
   * Open source software transcends geographical boundaries.
   * It’s accessible to users worldwide, promoting inclusivity.

68. · Host a simple website and configure virtual hosts

ANS:

1. **Configure Virtual Hosts**:
   * Create virtual host configuration files in /etc/apache2/sites-available/.
   * For each domain, create a file (e.g., your\_domain.conf) with the following content:
   * **<VirtualHost \*:80>**
   * ServerAdmin webmaster@your\_domain
   * ServerName your\_domain
   * DocumentRoot /var/www/your\_domain/public\_html
   * ErrorLog ${APACHE\_LOG\_DIR}/error.log
   * CustomLog ${APACHE\_LOG\_DIR}/access.log combined
   * **</VirtualHost>**
   * Enable the virtual host files using sudo a2ensite your\_domain.conf.
2. **Test and Reload Apache**:
   * Test the configuration using sudo apache2ctl configtest.
   * If successful, reload Apache with sudo systemctl reload apache2.
3. **DNS Configuration**:
   * Update DNS records for your domains to point to your server’s IP address.
4. **Test Locally** (optional):
   * Add entries to your local /etc/hosts file to test the configuration before DNS propagation.

69. · Install and manage databases like MySQL/mariaDB

ANS:

**MySQL and MariaDB: A Brief Overview**

Both MySQL and MariaDB are popular open-source relational database management systems (RDBMS). They share a similar structure but have some key differences:

1. **Ownership**:
   * **MySQL**: Owned by Oracle, MySQL has both open-source and closed-source components.
   * **MariaDB**: Fully open-source and developed as a fork of MySQL to ensure freedom and flexibility for users.
2. **Performance**:
   * **MariaDB**: Often considered faster and more efficient than MySQL.
   * **MySQL**: Still widely used but may lag behind MariaDB in terms of performance.
3. **Licensing**:
   * **MariaDB**: Fully GPL licensed.
   * **MySQL**: Dual-license approach (open-source and proprietary).

**Creating Databases**

To create a database in MySQL or MariaDB, follow these steps:

1. **Log in to MySQL/MariaDB**:
   * Open your terminal and enter:
   * **mysql -u root -p**
   * Provide the administrator password you set during installation.
2. **Create a Database**:
   * To create a new database (let’s call it **new\_database**), run:
   * CREATE DATABASE new\_database;
   * To avoid errors if the database already exists, use:
   * CREATE DATABASE IF NOT EXISTS new\_database;
3. **View Existing Databases**:
   * To see a list of databases, use:
   * SHOW DATABASES;
   * Default databases like information\_schema, performance\_schema, and mysql are usually present.
4. **Select a Database**:
   * To work with a specific database, use:
   * USE new\_database;
   * Confirm the selection with:
   * SELECT database();
5. **Deleting a Database**:
   * Be cautious! To delete a database, use:

DROP DATABASE new\_database;