**MODULE - 2**

**YUM**

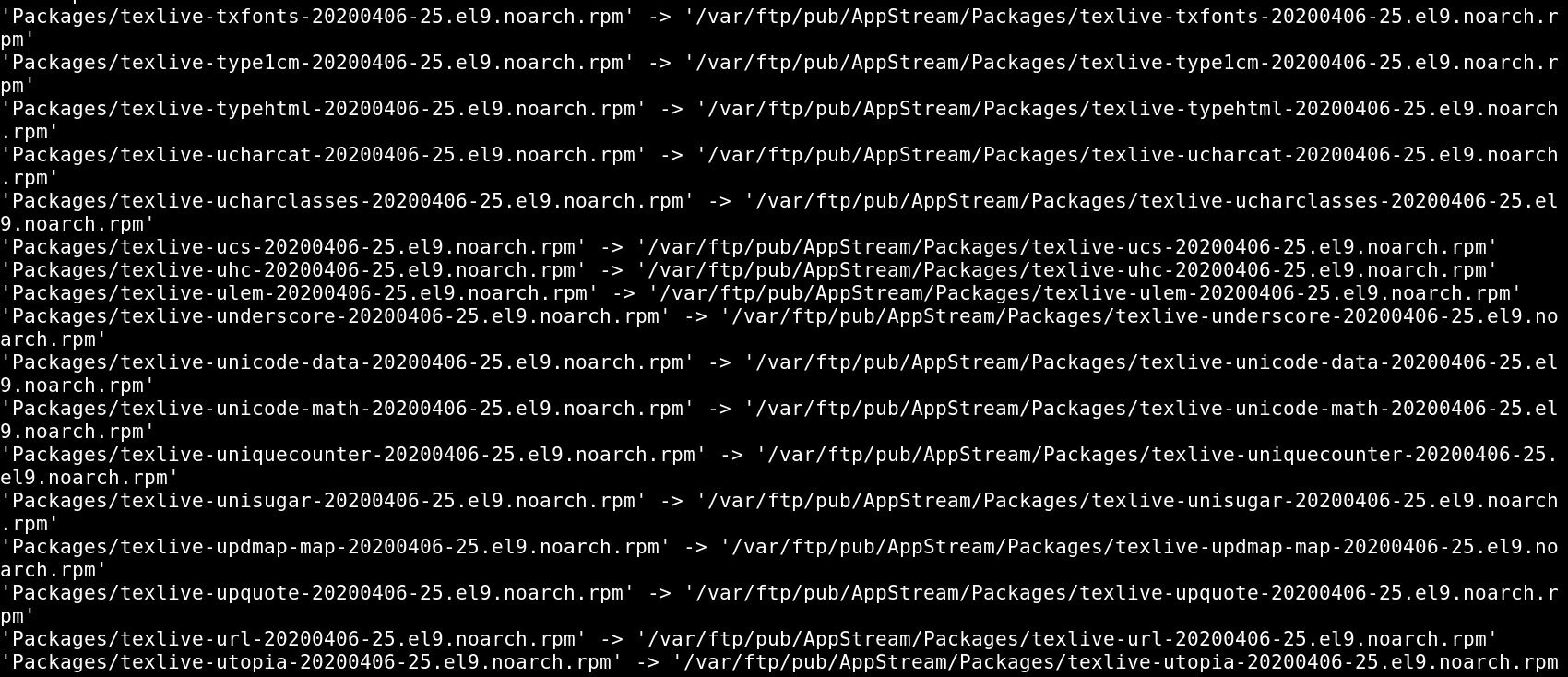
1. **configure local yum repository in red hat Linux? repository file name should be local-data. Repo? Also configure client repository using ftp?**

**CONFIGURING A LOCAL YUM REPOSITORY**

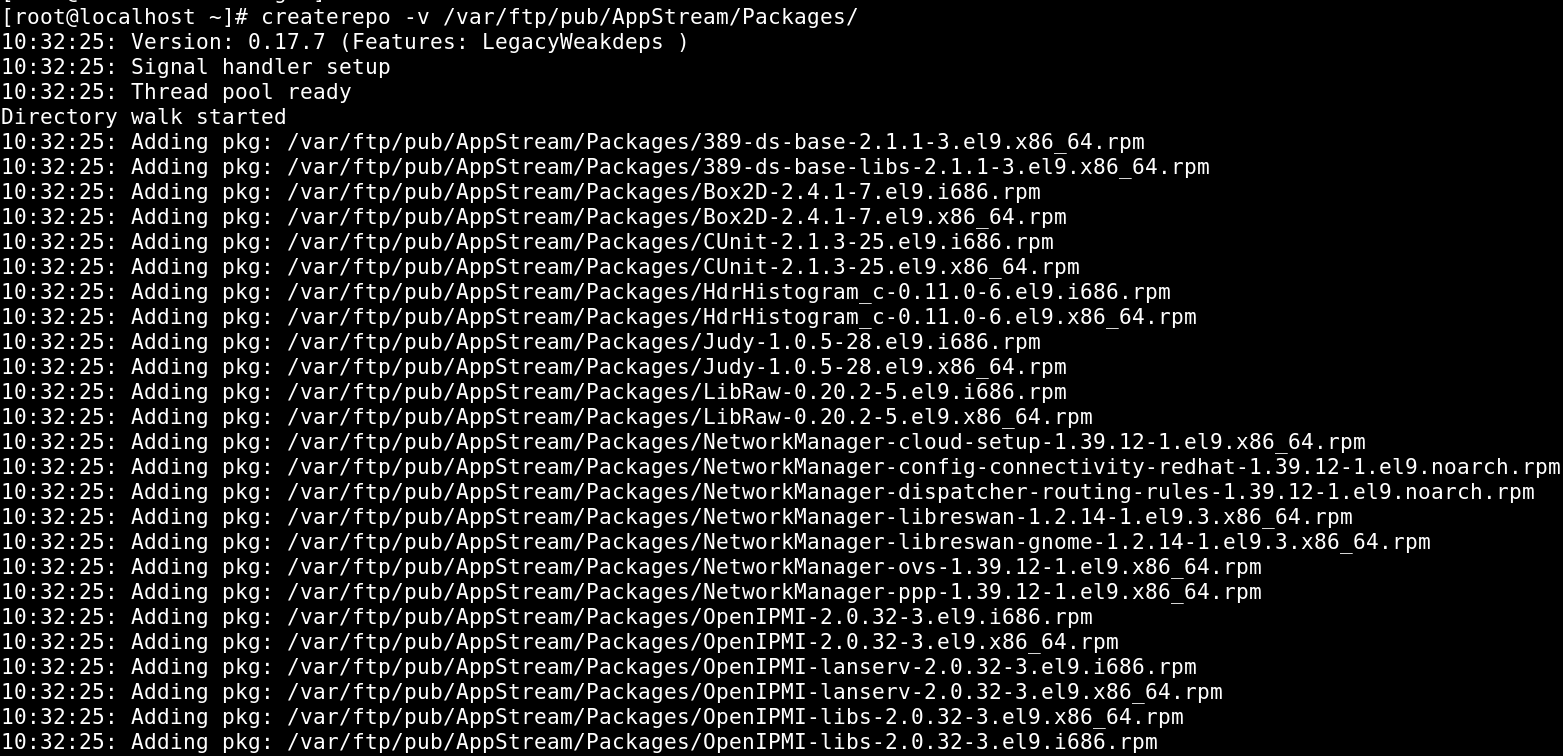
**Step 1: Create a Directory for the Repository**

****

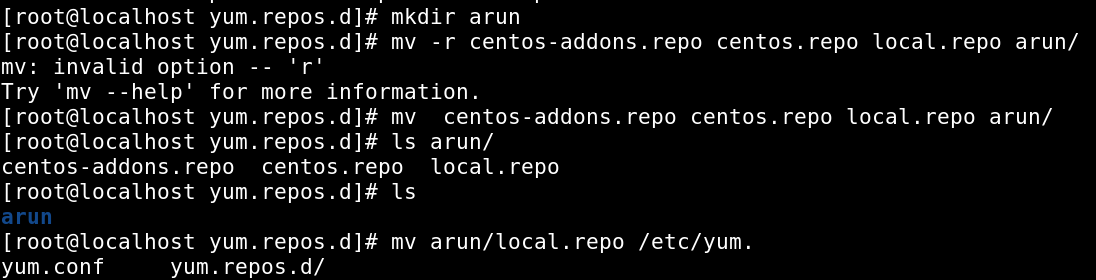
**Step 2: Copy RPM Packages to the Repository Directory**



**Step 3: Create the Repository Metadata**

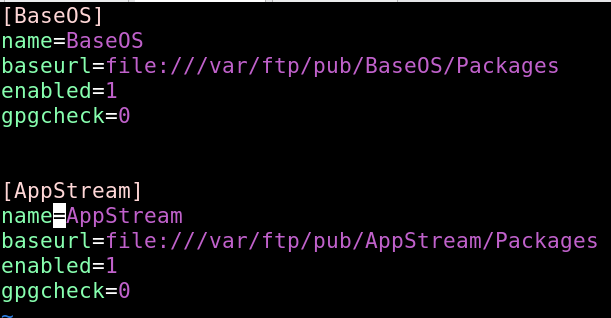
****

**Step 4: Create a YUM Repository Configuration File**



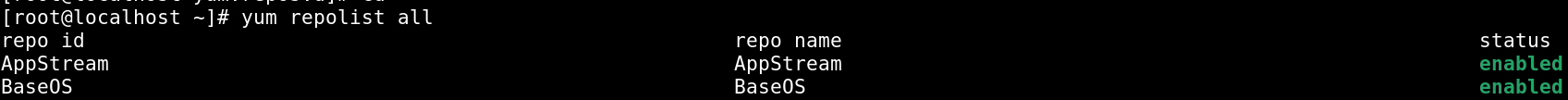
* **vi local.repo**

****



**Step 5: Clean YUM Cache and Test**

* **yum repolist list all**



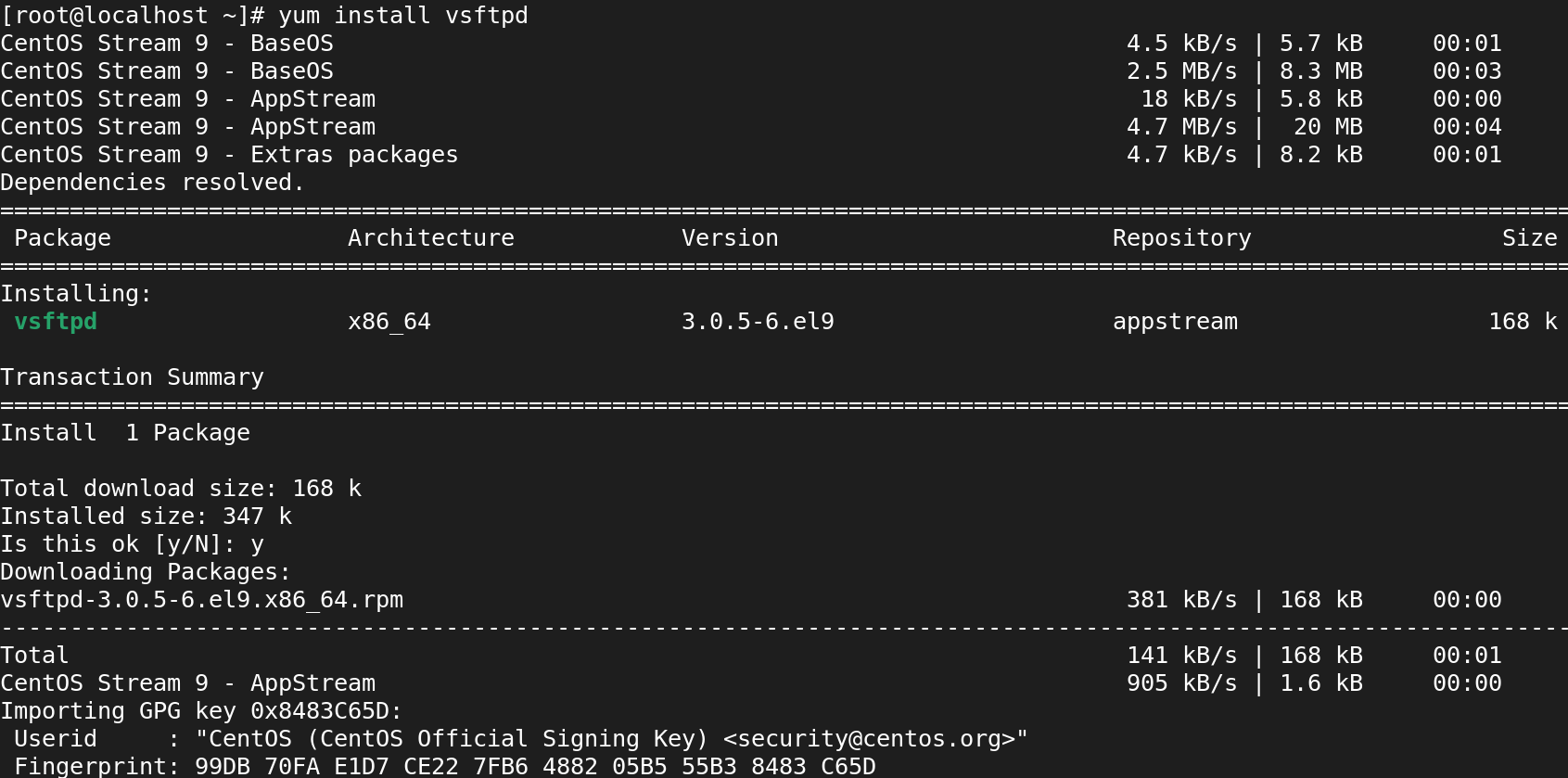
* **yum clean all**



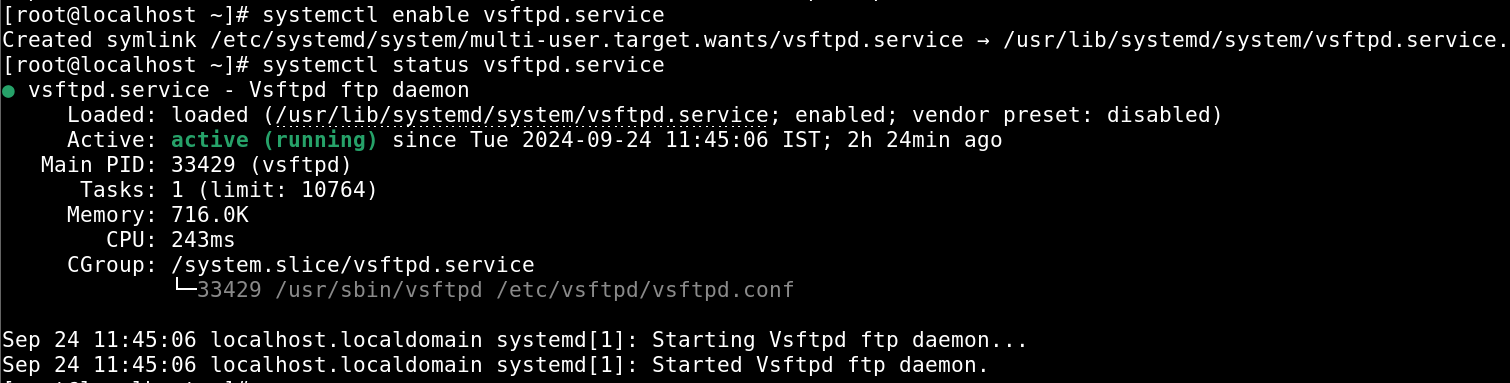
**CONFIGURE CLIENT REPOSITORY USING FTP**

**Step 1: Install and Configure FTP Server**

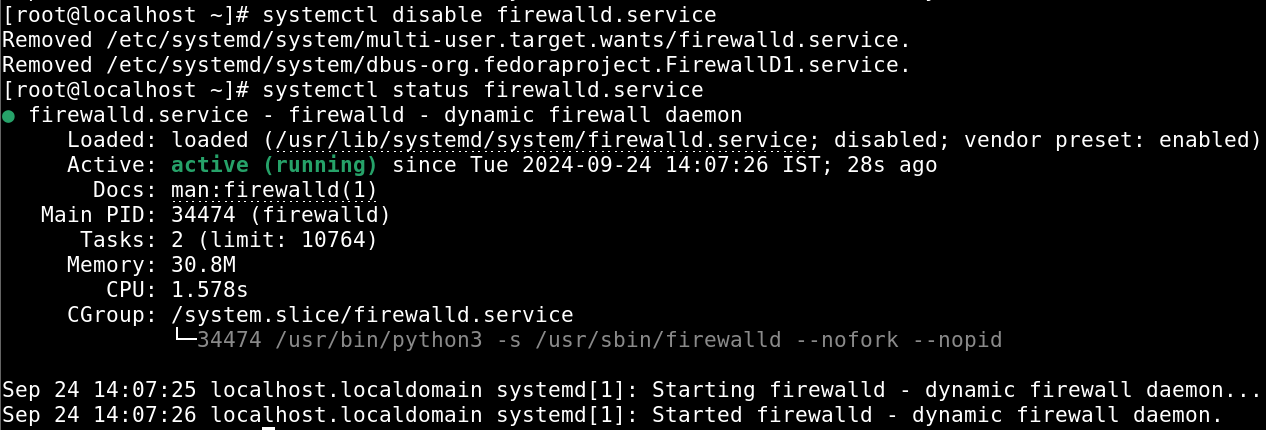
* **Install vsftpd**



* **Enabling VSFTPD services**

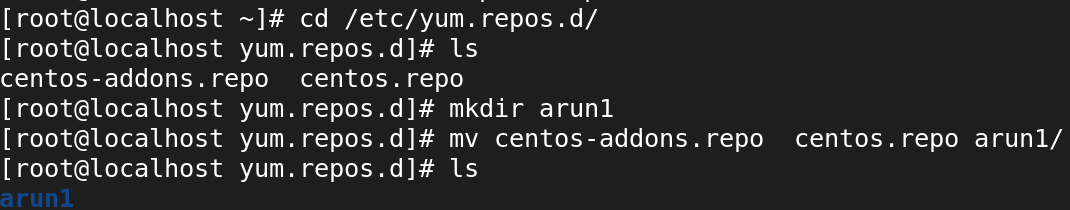
****

* **Disabling Firewall**

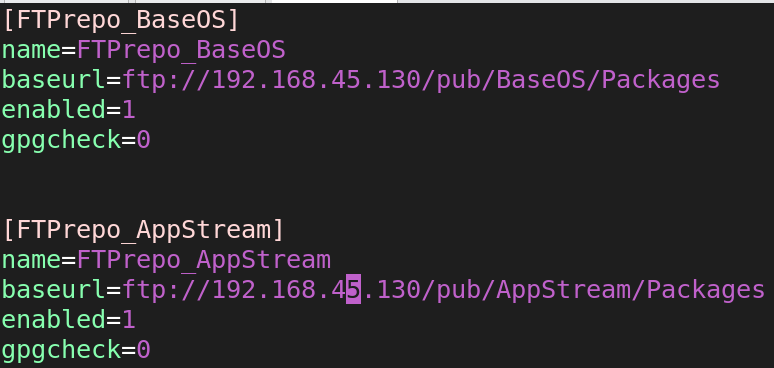
****

**Step 2: Configure the Client to Use the FTP Repository**

* **cd /etc/yum.repos.d**

****

* **vi ftplocal.repo**

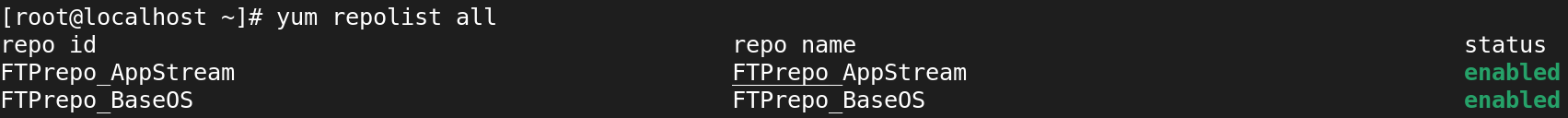


**Step 3: Clean YUM Cache and Test**

* yum clean all



* yum repolist all



1. **Difference between rpm and yum?**

**RPM:**

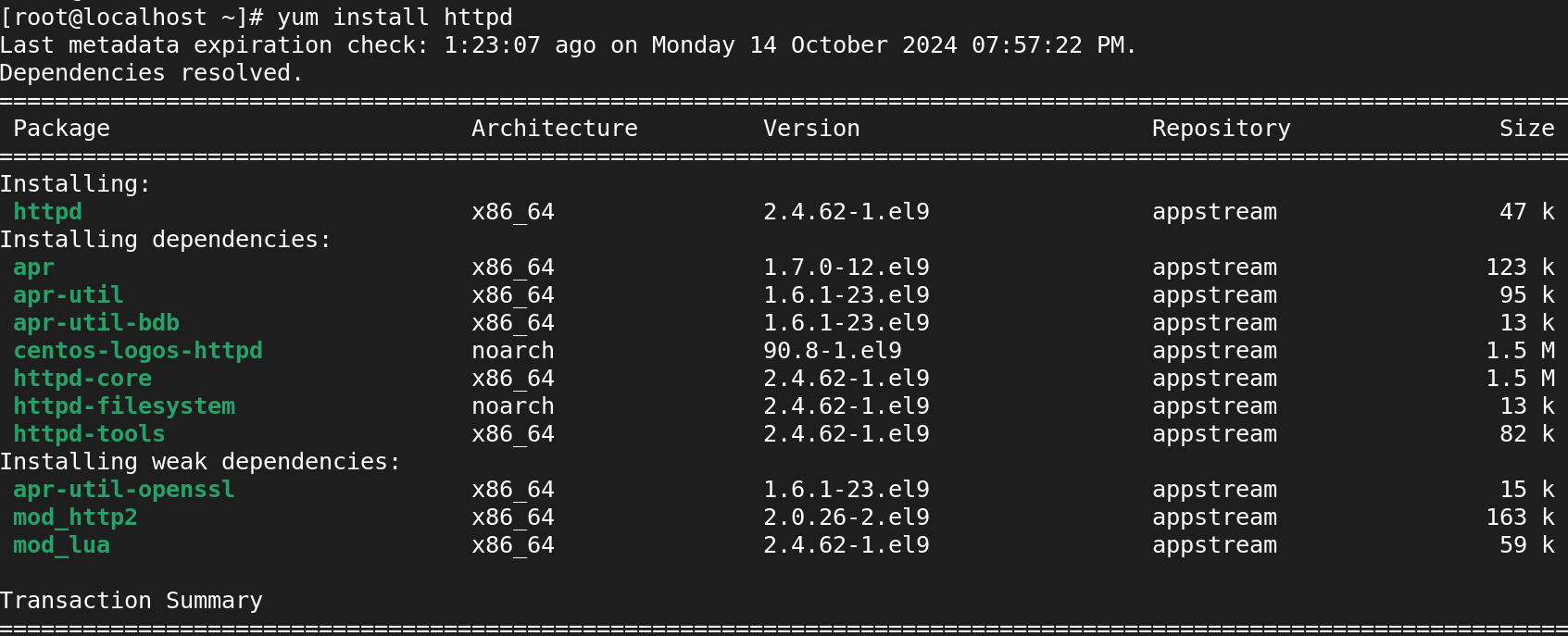
* Does not resolve dependencies automatically.
* Works with local .rpm files only.
* Installs packages using rpm -ivh package.rpm.
* Requires manual updates by downloading new packages.
* Manual package removal using rpm -e package-name.

**YUM:**

* Automatically resolves dependencies.
* Fetches packages from repositories.
* Installs packages with yum install package-name.
* Automatically updates packages using yum update.
* Removes packages along with dependencies using yum remove package-name.

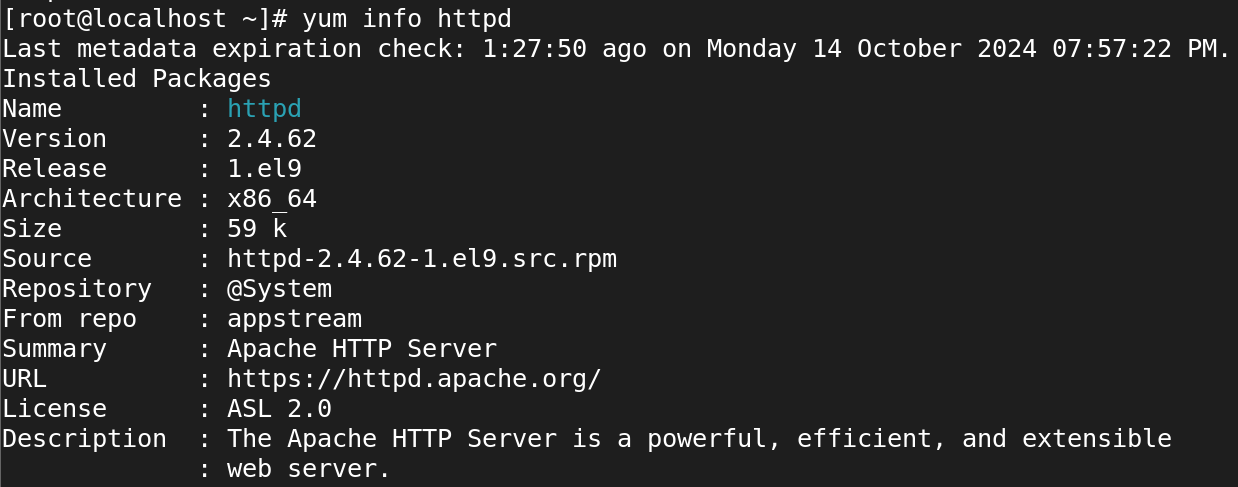
1. **write command to install package?**

* Yum install package-name



1. **How to get the package information using yum?**

* Yum info package-name



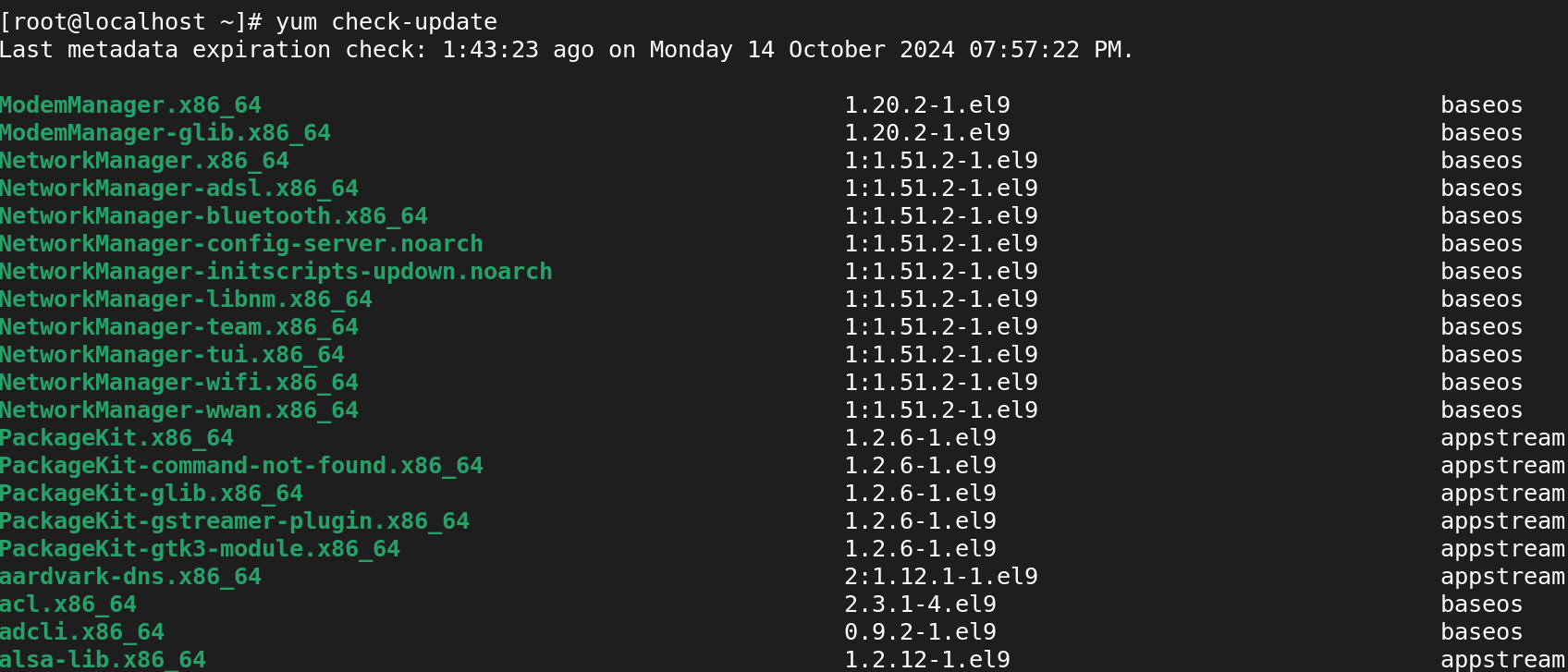
1. **How to find files belongs to which rpm package?**

* rpm -qf /usr /bin/mkdir



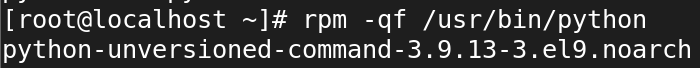
1. **How to check the updates for yum repository?**

* Yum check-update



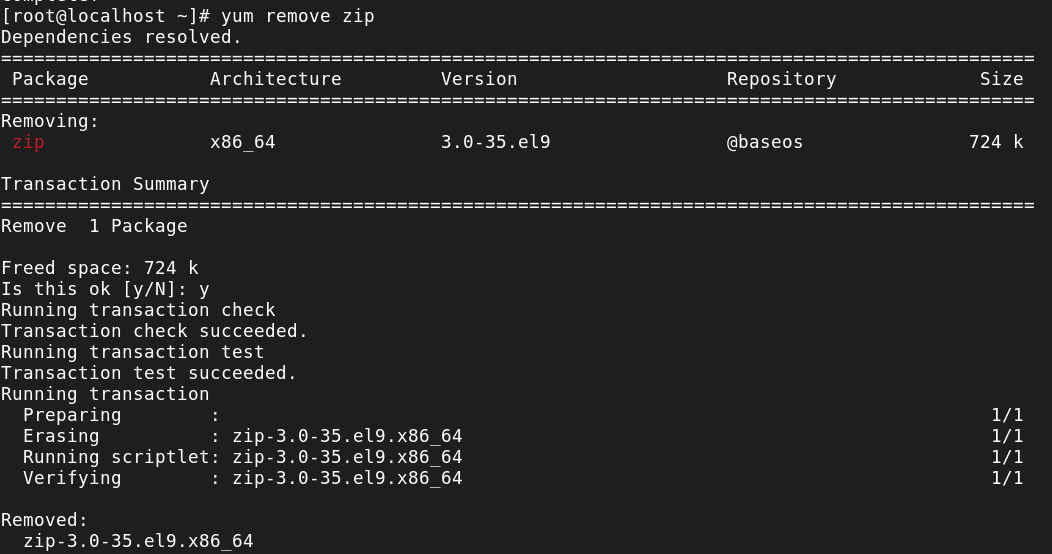
1. **How to know the particular files belongs to which package?**

* Rpm -qf /usr/bin/python



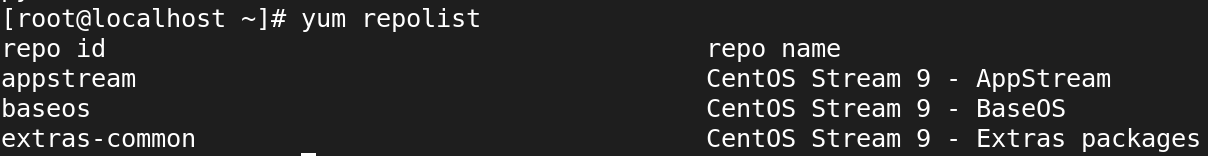
1. **How to remove the package using yum?**

* yum remove zip



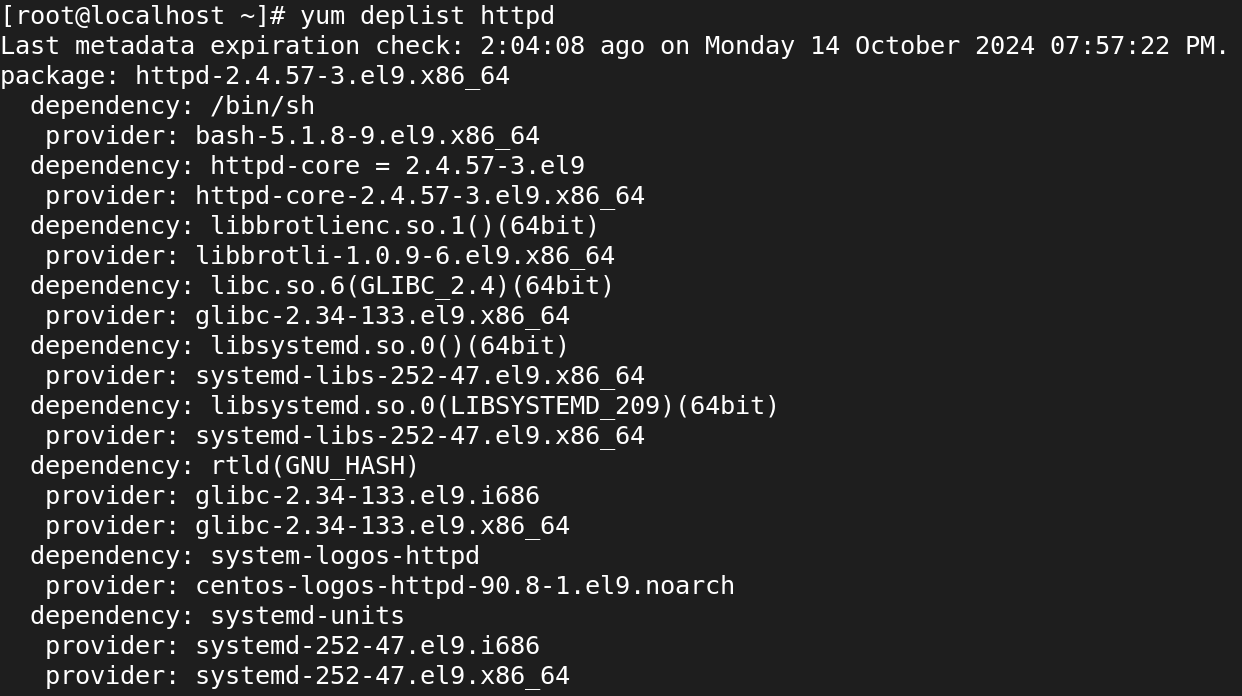
1. **How to list the enabled repositories?**

* Yum repolist



1. **How to get the list of dependencies for specific package?**

* Yum deplist httpd



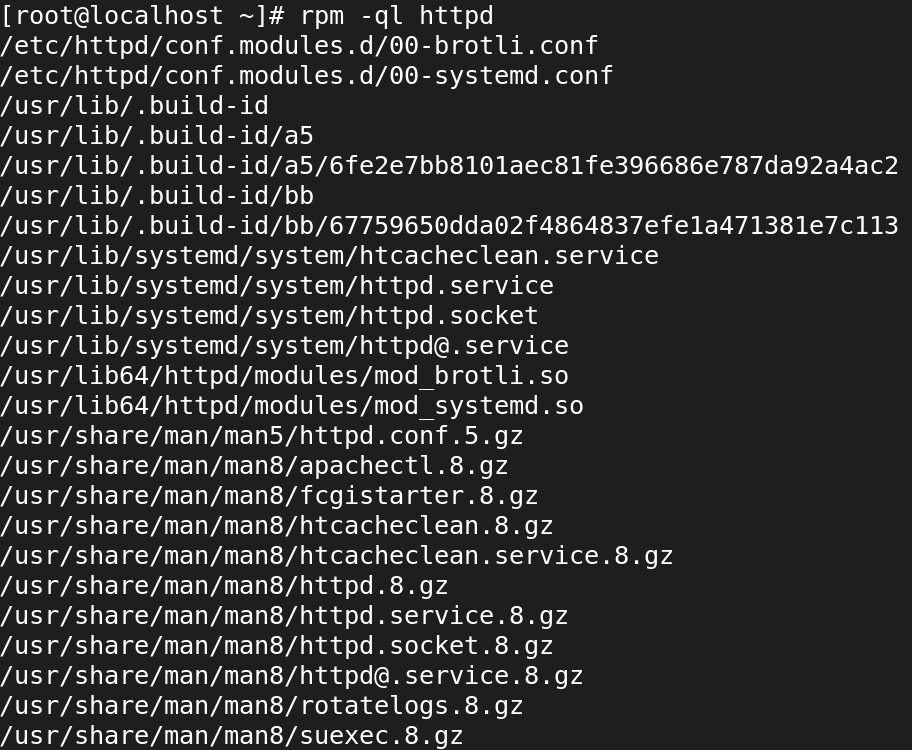
1. **How to display a list of recently installed software?**

* Yum history



1. **How to display the list of installed software package’s files?**

* Rpm -ql package-name



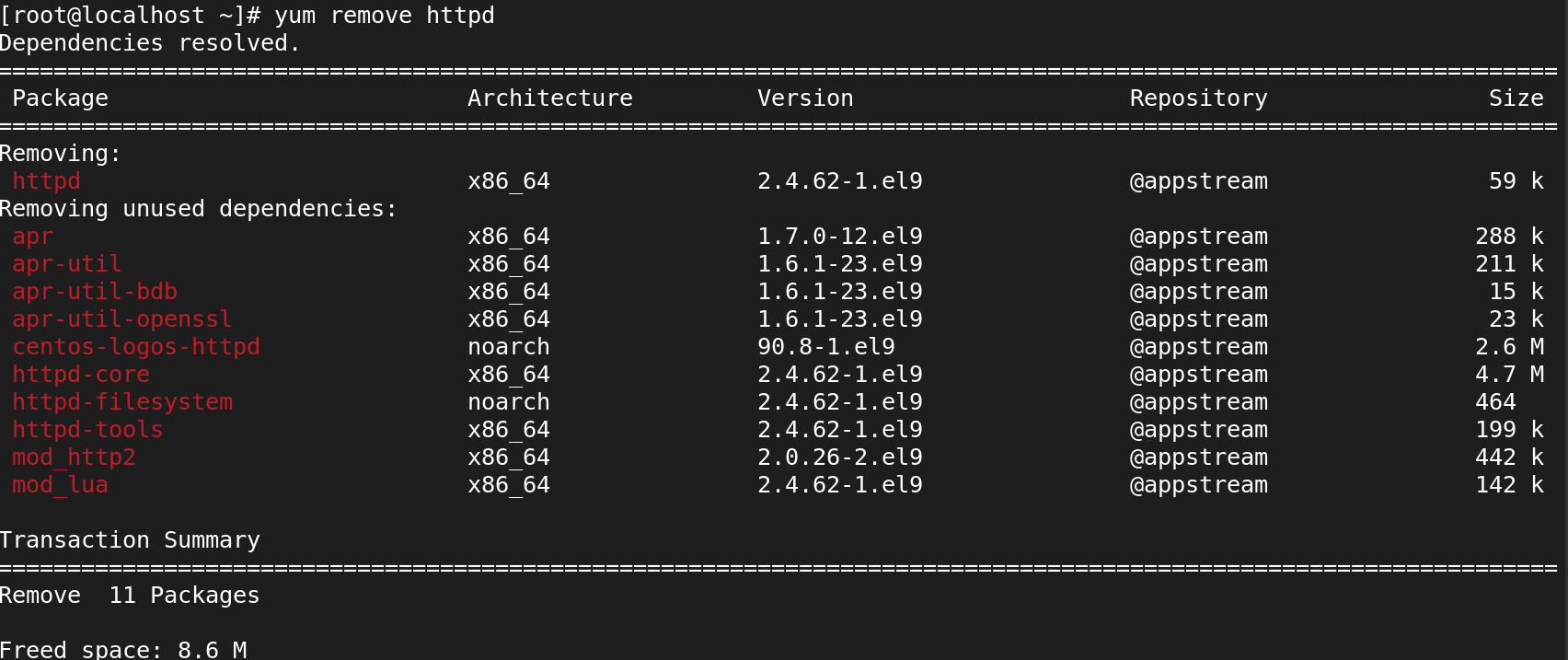
1. **How to install specific software package using rpm command on Redhat Linux?**

* rpm -ivh waypipe-0.8.2-1.el9.x86\_64.rpm
* i: install a package
* v: verbose
* h: print hash marks as the package archive is unpacked.



1. **How to remove the software package from on Redhat Linux?**

* Yum remove httpd



1. **what is the repository file location?**

* All repository configuration files are stored in this directory, typically with a .repo extension.
* Each .repo file contains the configuration for one or more repositories.
* **/etc/yum.repos.d/**

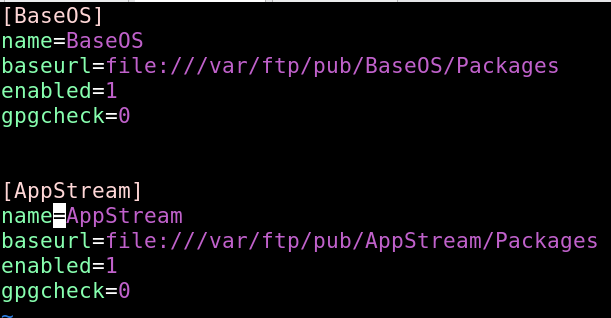
1. **write yum configuration file?**

**Step 1: Create a New .repo File**

* **vi local.repo**

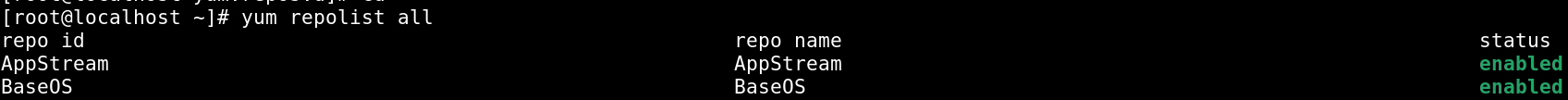
****

**Step 2: Add the Repository Configuration**



**Step 3: Clean YUM Cache and Test**

* **yum repolist list all**



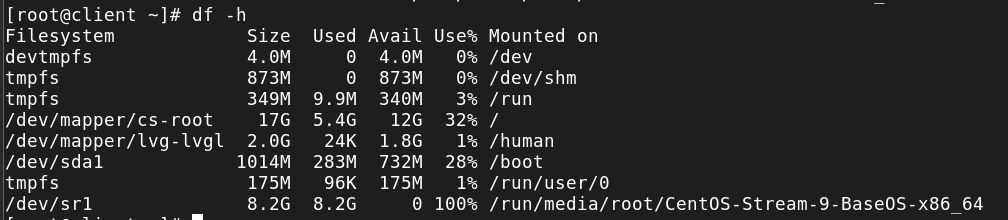
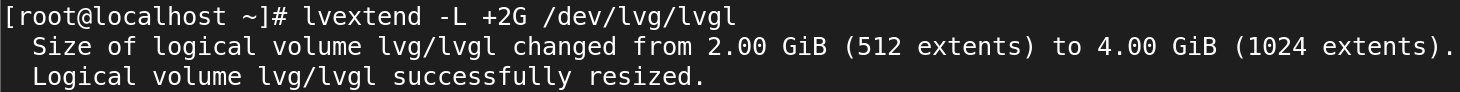
* **yum clean all**



**LVM**

1. **Is it possible to increase the logical volume online?**

* lvextend -L +2G /dev/lvg/lvgl



1. **how to remove LVM completely from the host?**

* **Unmount any mounted logical volumes**.
* **Remove logical volumes** using lvremove.
* **Remove volume groups** using vgremove.
* **Remove physical volumes** using pvremove.
* **Remove LVM packages** from the system.
* **Verify the removal** to ensure no LVM components remain.

1. **how to reduce lvm file system?**

* resize2fs /dev/vg\_name/lv\_name 10G
* lvreduce -L 10G /dev/vg\_name/lv\_name

1. **In MBR how many numbers of partition can create?**

* **Maximum Primary Partitions**: 4
* **Maximum Logical Partitions**: Up to 26 (if one primary partition is an extended partition).

1. **In GPT how many numbers of partition can create?**

* **Maximum Partitions in GPT**: Up to **128 partitions** by default, although this limit can potentially be increased based on the operating system or specific tools used.

1. **what are the commands to create partition?**

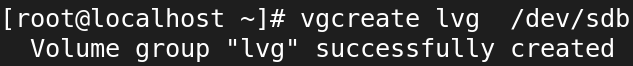
**To create a partition with fdisk:**

* Start: fdisk /dev/sda
* New partition: n
* Write changes: w

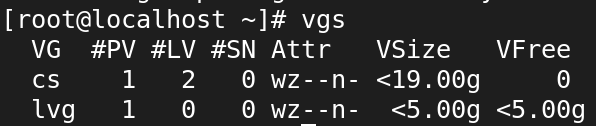
**To create a partition with parted:**

* Start: parted /dev/sda
* Create partition: mkpart primary ext4 1MiB 20GiB
* Exit: quit

1. **what is the command to create volume group?**

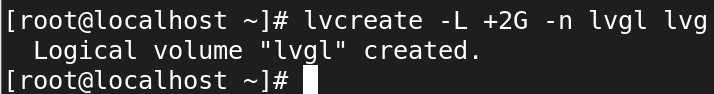
* vgcreate lvg /dev/sdb

**LIST VOLUME GROUP:**

* vgs 

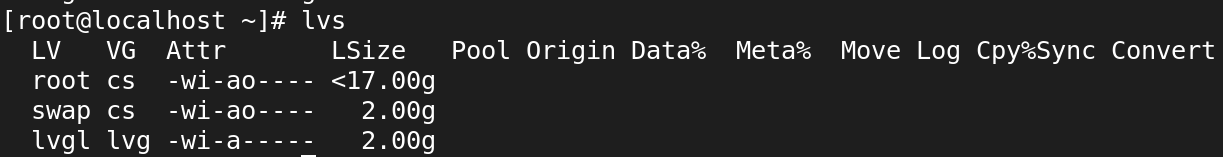
1. **what is the command to create logical volume?**

* lvcreate -L +5G -n lvgl lvg



**TO LOGICAL VOLUME:**

* lvs



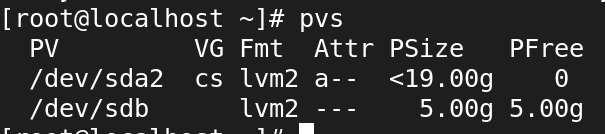
1. **how to create ext4 and xfs file system?**

* mkfs.ext4 /dev/lvg/lvgl
* mkfs.xfs /dev/lvg/lvgl



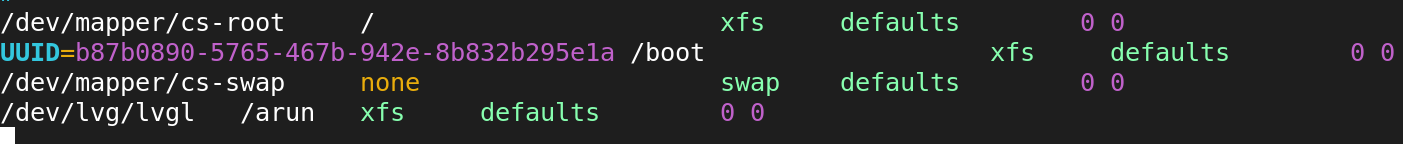
1. **how to scan physical volume?**

* pvs



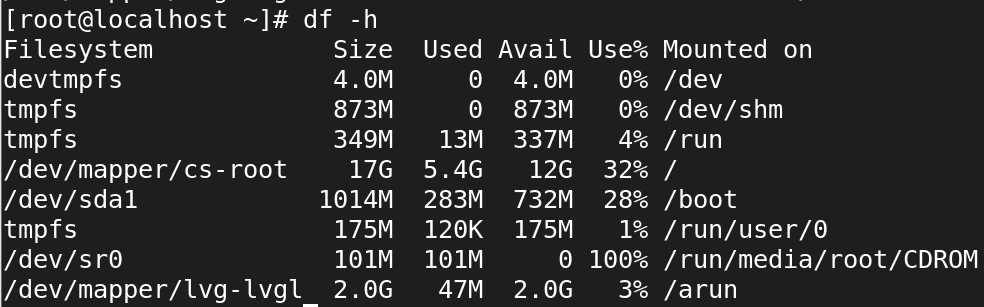
1. **what are the field in /etc/fstab?**

* vi /etc/fstab



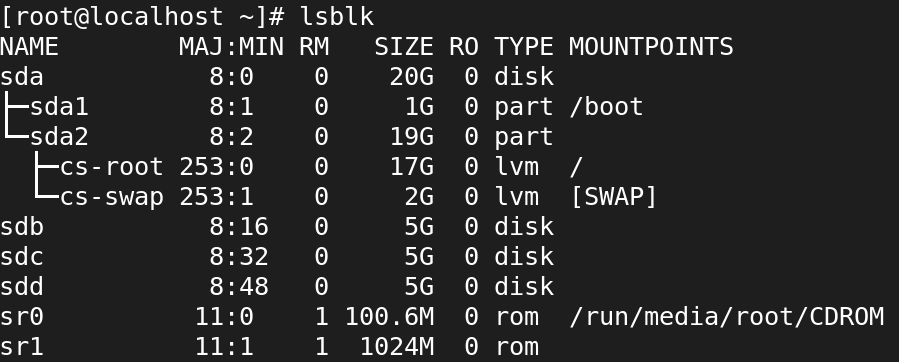
1. **what is the command see the mounted file system?**

* df -h



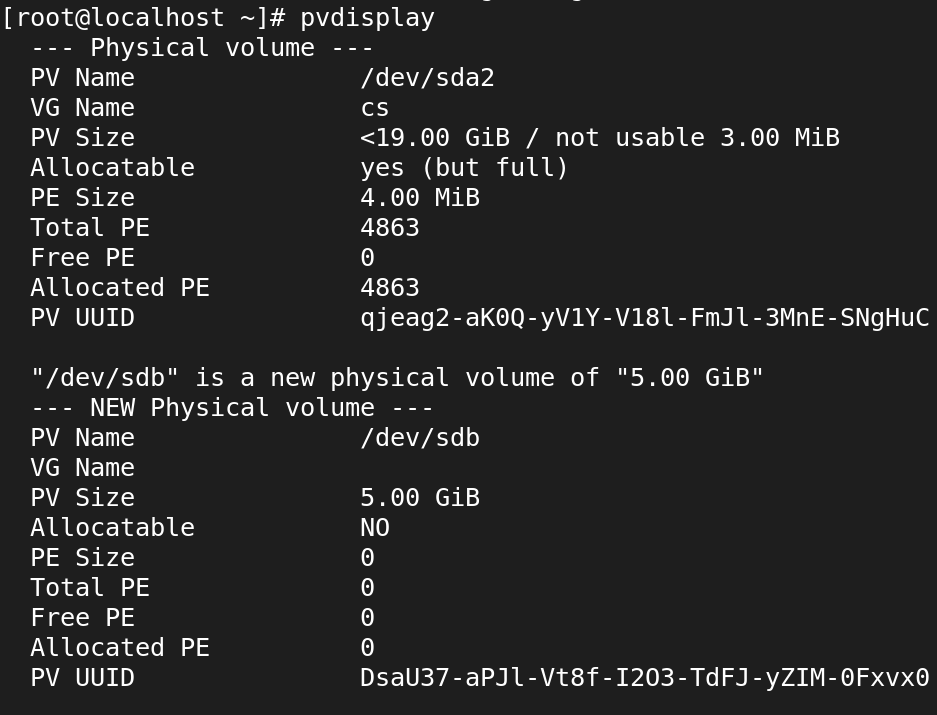
1. **what is the command get disk details?**

* lsblk

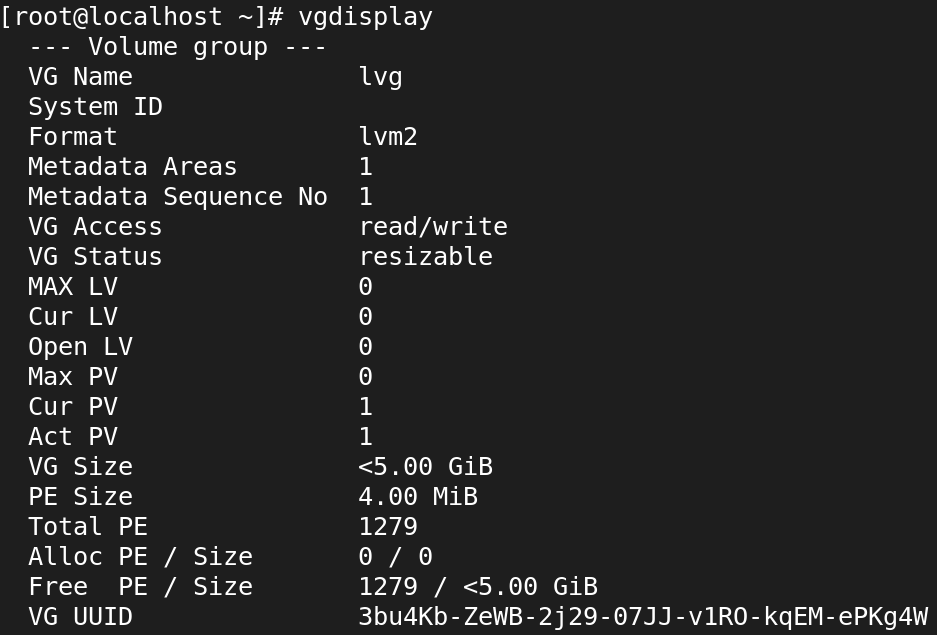


1. **How to see the detailed volume group, physical and logical volume information?**

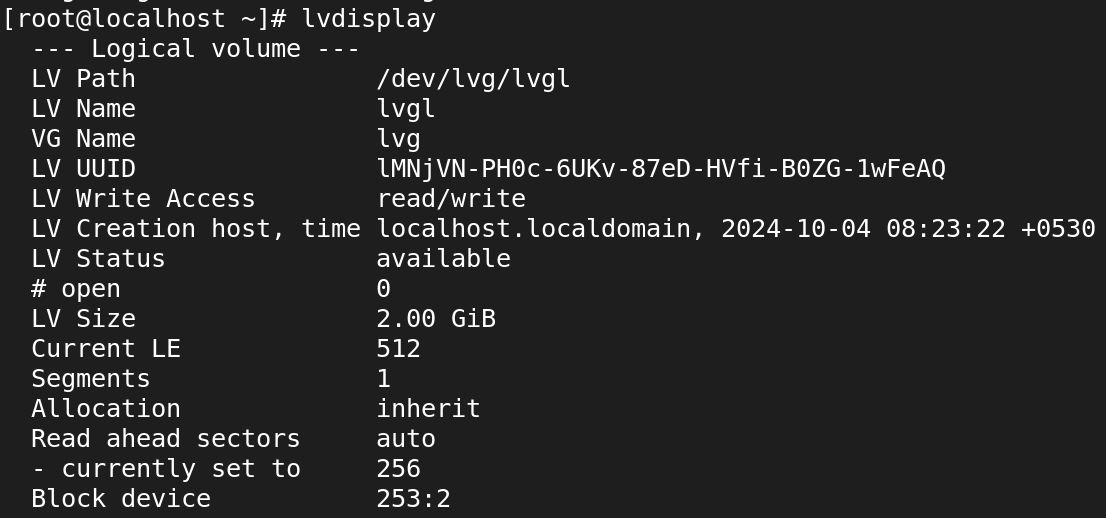
* pvdisplay



* vgdisplay



* lvdisplay



1. **Attach LUN/disk 4GB, create lvm based filesystem(xfs), mountpoint should be /dev\_data**
2. Extend the above /dev\_data filesystem 3GB final file system size should be 7GB?
3. reduce above /dev\_data filesystem 2GB?
4. Attach LUN/disk 6GB, create lvm based filesystem(ext4), mountpoint should be /prod\_data
5. reduce above lvm filesystem 2GB and final size of filesystem should be 4G?
6. is it possible to reduce the xfs filesystem?
7. Attach disk 4GB, create partition-based filesystem, mountpoint should be /drive
8. Attach disk 6GB, create partitions 3GB & 3GB respectively and create filesystem
9. Attach disk 3GB, create raw disk-based filesystem, mount point should be /device
10. What are the steps involved to create the logical volume from scratch?
11. what is swap?
12. create 3Gb swap filesystem and enable swap?
13. create 2GB swap file and enable swap

**SSH**

1. **what is ssh?**

* **SSH (Secure Shell)** is a network protocol used to securely connect to and communicate with remote systems over an unsecured network.
* It provides encrypted communication for tasks such as logging into remote servers, executing commands, transferring files, and managing network services securely.

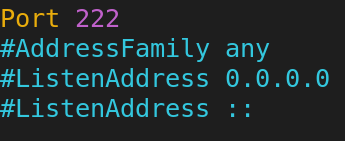
1. **port number of SSH?**

* Default port number of SSH is 22.

1. **Is it possible to change ssh port number?**

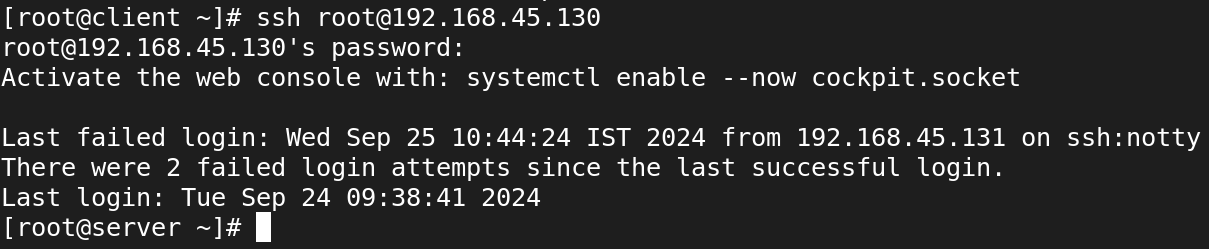
* Yes, it is possible to change the SSH port number. This can be done by editing the SSH configuration file on the server.





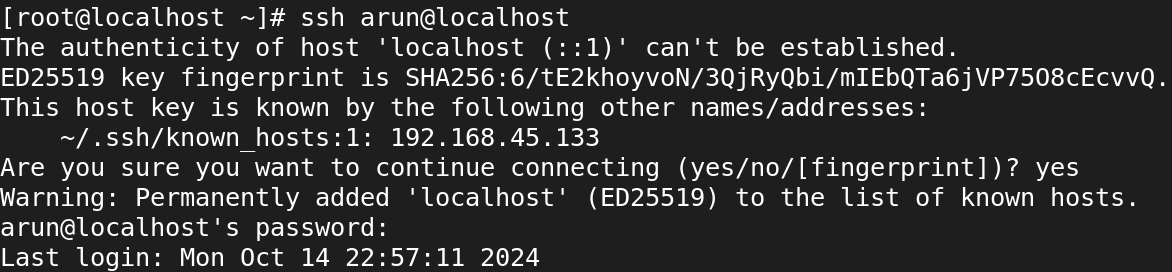
1. **Write command to login remote machine in terminal?**

* Ssh user@hostname



1. **Write command to login local machine in terminal?**

* Ssh root@localhost

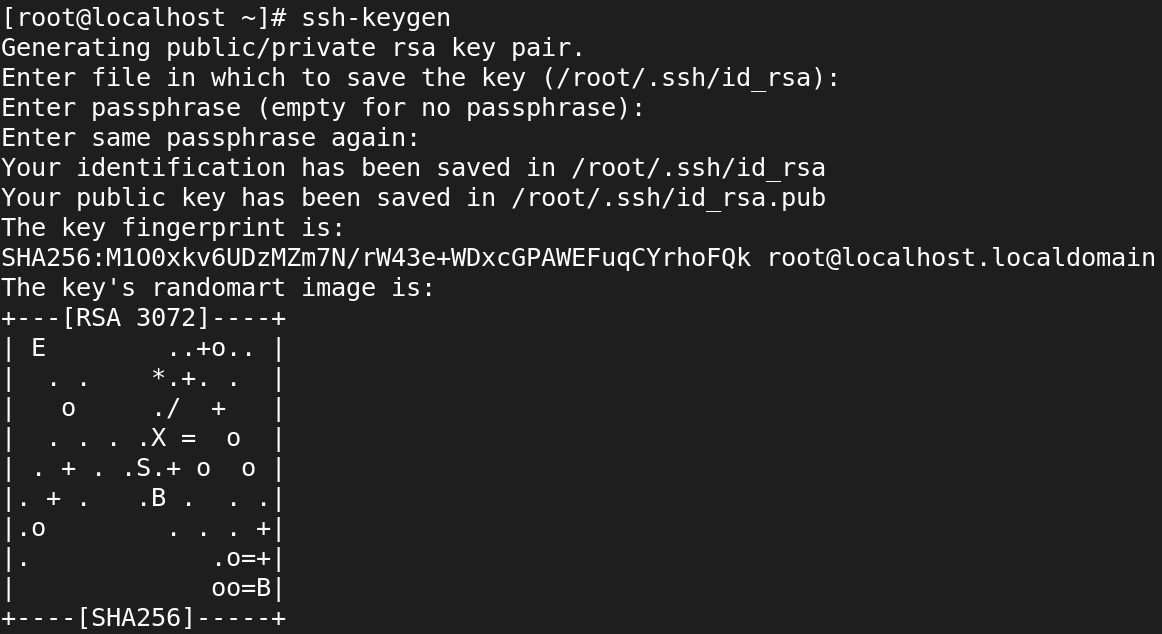
****

1. **ssh configuration file location?**

* /etc/ssh/sshd\_config

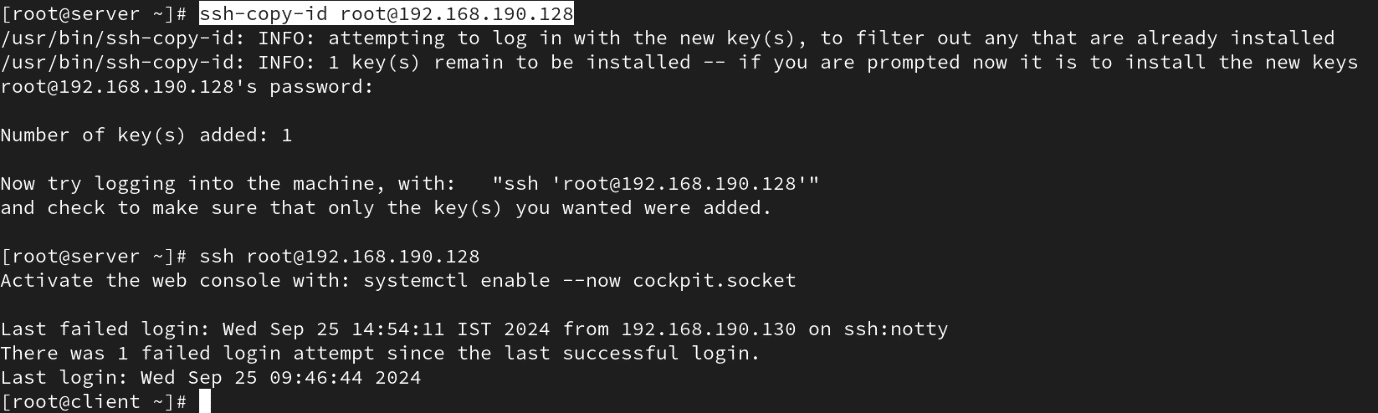
1. **configure SSH key based authentication? write path of key file location?**
2. **server1-user: admin1 to server2-user: admin1**
3. **what is the command to generate the ssh key?**

* Ssh-keygen



1. **what is the command to copy the keys to remote machine?**

* ssh-copy-id user@remote-host



1. **when you generate the key what are files will be create?**

* **Private Key**: ~/.ssh/id\_rsa
* **Public Key**: ~/.ssh/id\_rsa.pub

1. **what is the ssh daemon name?**

* **sshd** stands for **SSH Daemon**, which is the server-side program that listens for incoming SSH connections and manages the authentication and communication with clients.

1. **write ssh service-related commands which you know?**

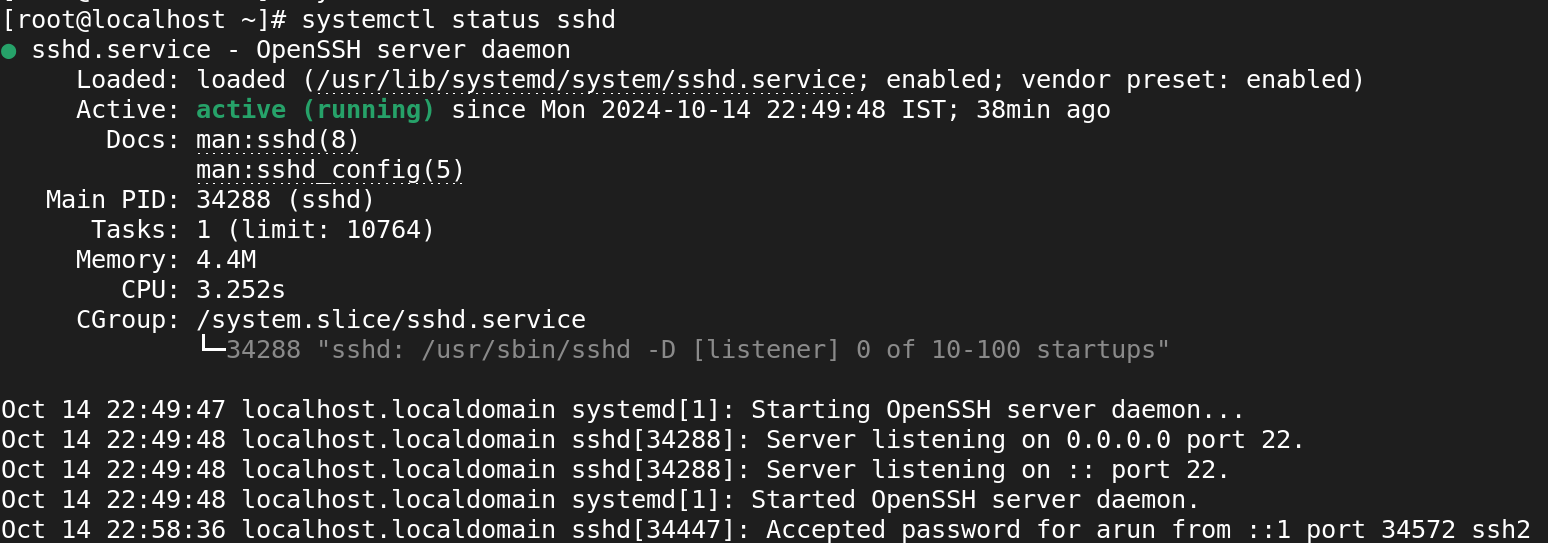
* **Start the SSH Service**

****

* **Stop the SSH Service**

****

* **Check the SSH Service**

****

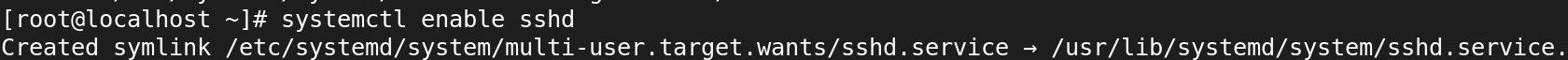
* **Reload the SSH Service**

****

* **Restart the SSH Service**

****

* **Enable SSH Service**

****

* **Disable SSH Service**

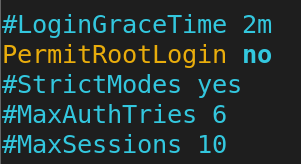
****

1. **Disable ssh root login?**

* **Open the SSH configuration file (/etc/ssh/sshd\_config)**



* **Change the value** from yes to no to disable root login



* **Restart the SSH service** to apply the changes

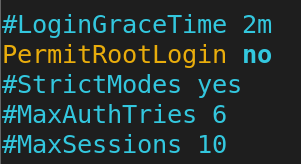
****

1. How to disable password-based authentication?

* **Open the SSH configuration file (/etc/ssh/sshd\_config)**



* **Change the value** from yes to no to disable root login



* **Restart the SSH service** to apply the changes

****

1. **Allow ssh access to particular user? (user1, user2, user3)**

* **Open the SSH configuration file (/etc/ssh/sshd\_config)**



* **Add the AllowUsers directive** at the end of the file, specifying the users you want to allow

1. Change ssh port number 4044 and try to ssh login.
2. How to start, stop, restart, check the status of ssh service?

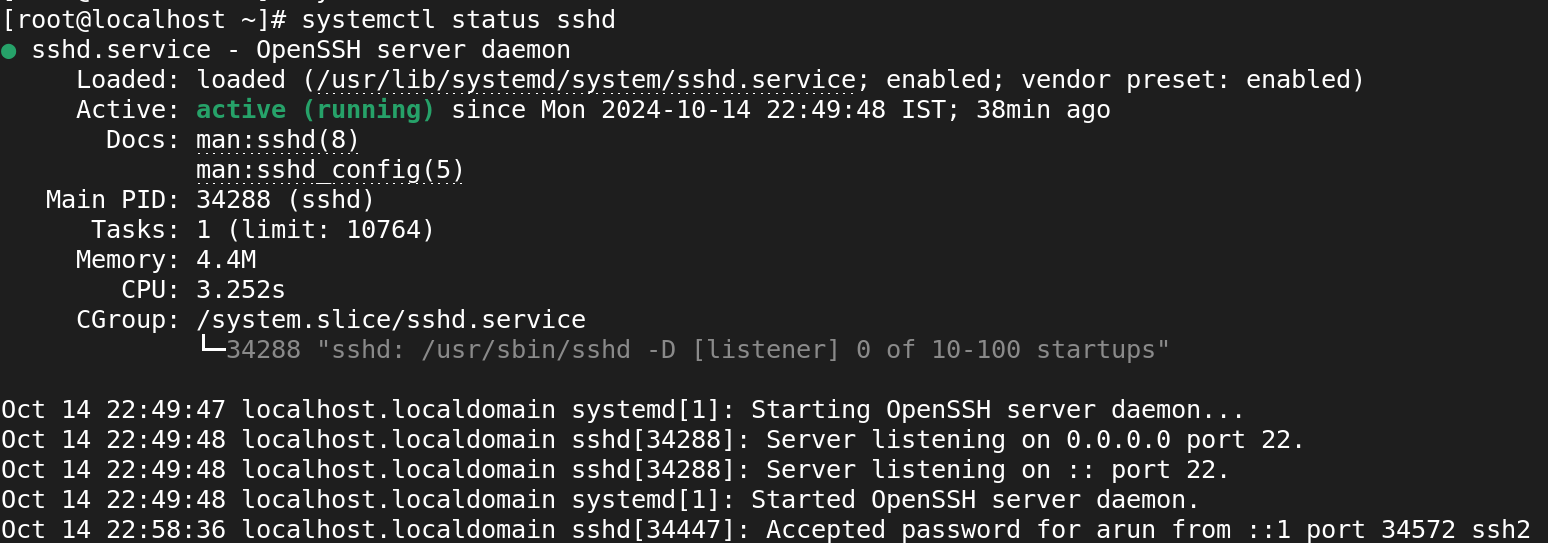
* **Start the SSH Service**

****

* **Stop the SSH Service**

****

* **Check the SSH Service**

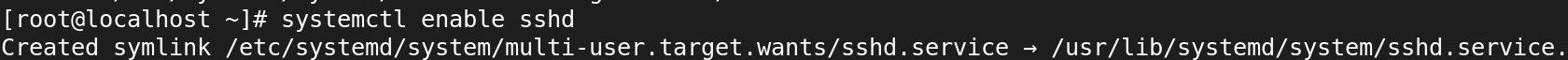
****

* **Restart the SSH Service**

****

1. **How to start the ssh service permanently?**

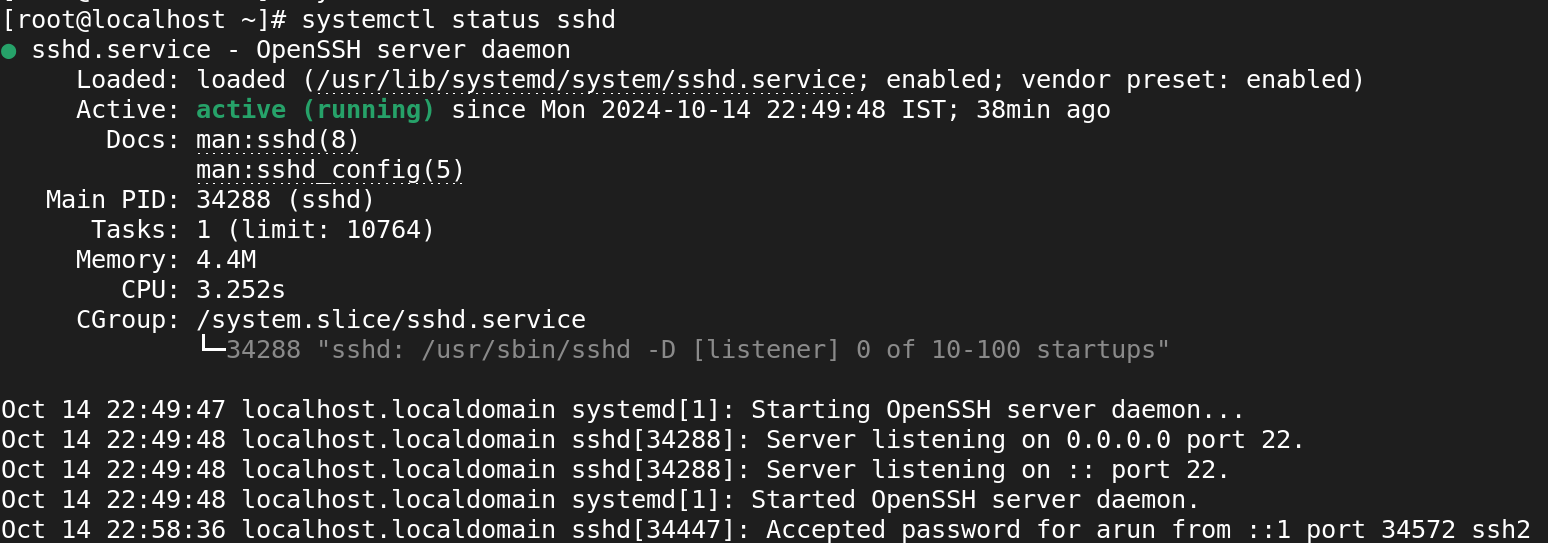
* **Enable SSH Service**

****

* **Start the SSH Service**

****

* **Check the SSH Service**

****

1. **How to stop the ssh service permanently?**

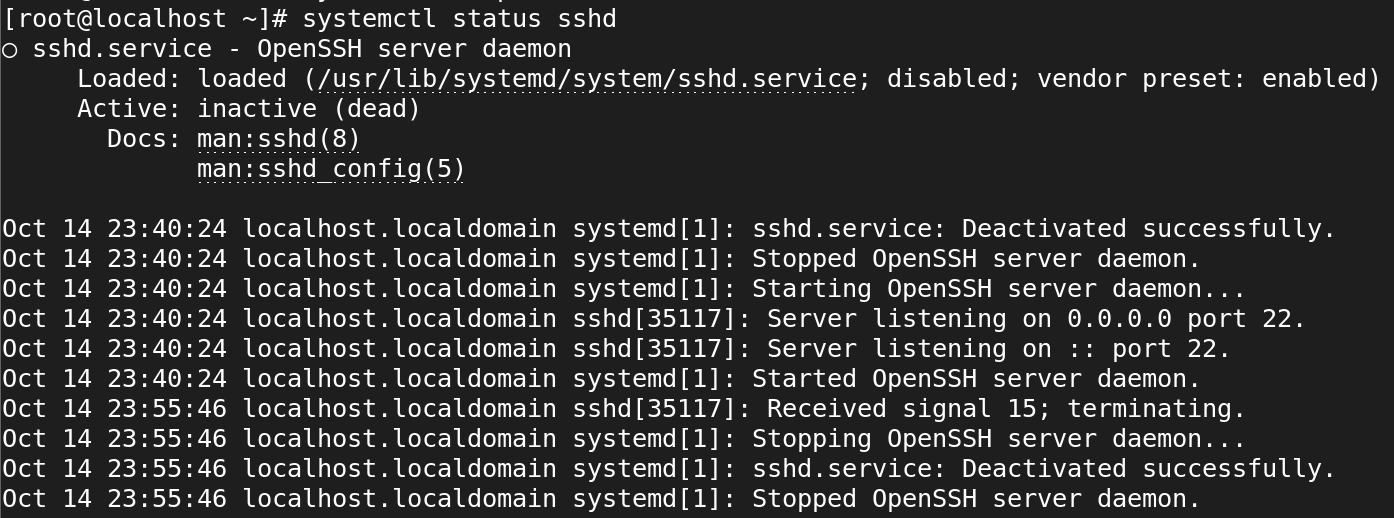
* **Disable SSH Service**

****

* **Stop the SSH Service**

****

* **Check the SSH Service**



**Cron**

1. **what is cronjob?**

* A **cron job** is a scheduled task in Unix-like operating systems (such as Linux) that runs automatically at specified intervals.
* It uses the **cron** daemon, a background service that executes scheduled commands or scripts at particular times or dates.
* cron jobs are defined in a **crontab** file (cron table), where each line represents a task and specifies when it should be executed.

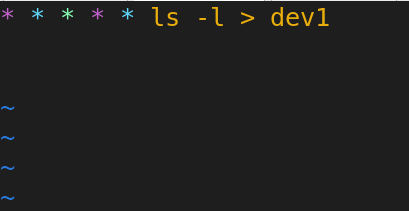
1. **what are the fields in cronjob?**

A cron job consists of five time and date fields, followed by the command to be executed. These fields determine when the cron job will run. The fields represent:

* Minute (0-59): Specifies the exact minute of the hour when the job should run.
* Hour (0-23): Specifies the hour of the day (in 24-hour format) when the job should run.
* Day of the Month (1-31): Specifies the day of the month when the job should run.
* Month (1-12): Specifies the month of the year when the job should run.
* Day of the Week (0-7): Specifies the day of the week when the job should run (0 and 7 both represent Sunday).

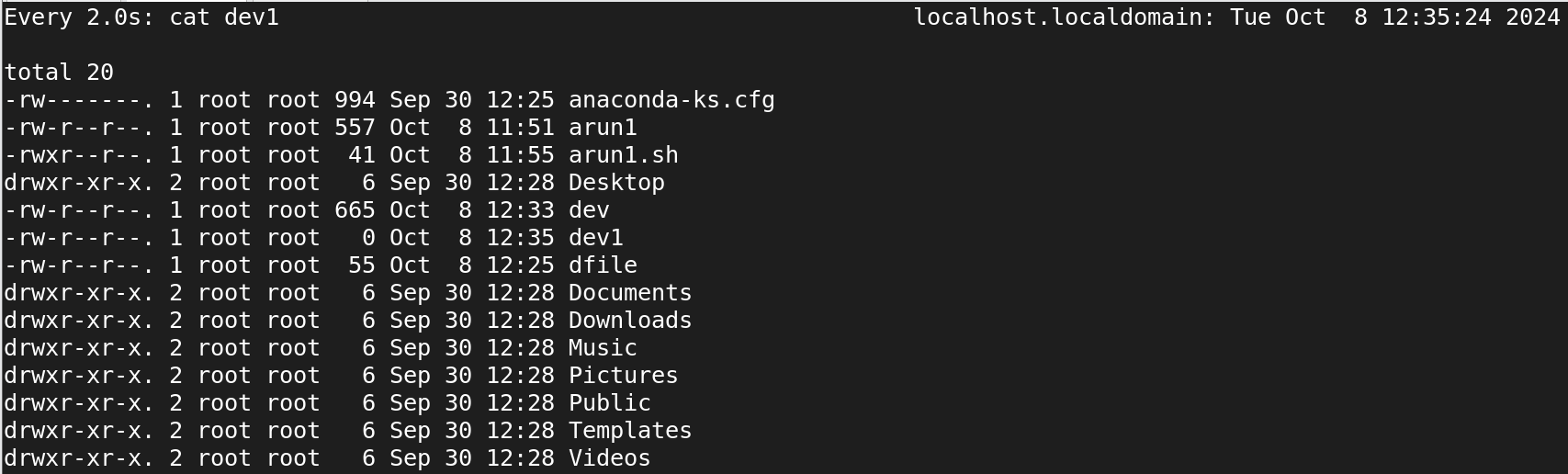
1. **how to edit cronjob?**

* crontab -e



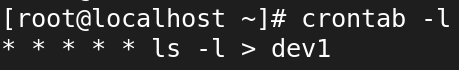
**To Watch live output:**

* watch ‘cat dev1’



1. **How to list cronjob?**

* Crontab -l



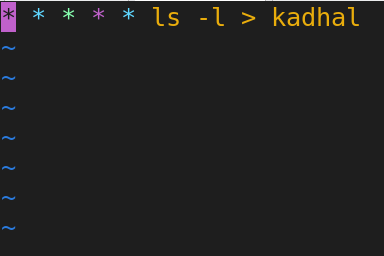
1. **how to remove cronjob?**

* Crontab -r



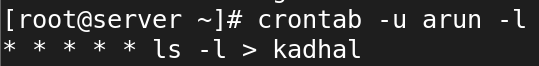
1. **how to edit, list, remove other users cronjob?**

* crontab -u username -e

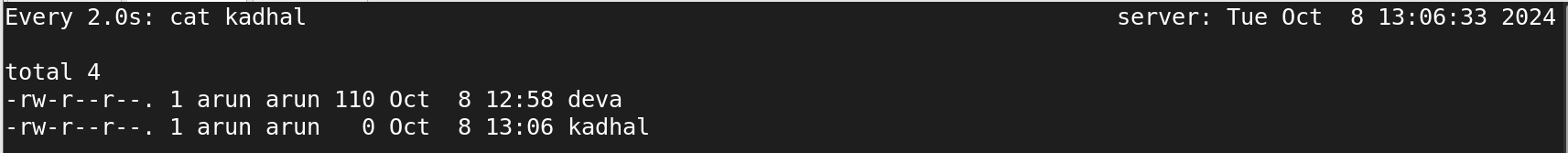


**List Existing Cron Jobs specific user:**

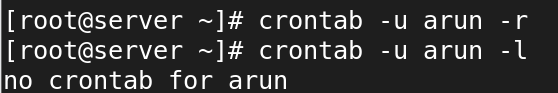
* crontab -u username -l



* watch ‘cat kadhal’



* crontab -u username -r



1. **what is the cron log file?**

* The **cron log file** is a log file that records the activities and events related to the execution of cron jobs on a system.
* It helps system administrators and users monitor whether scheduled tasks (cron jobs) have been executed successfully, failed, or experienced any issues.

1. **schedule cronjob at yearly once?**

* 0 0 1 1 \* command\_to\_execute
* **0**: This indicates the **minute** (0 minutes past the hour).
* **0**: This indicates the **hour** (midnight).
* **1**: This indicates the **day of the month** (1st).
* **1**: This indicates the **month** (January).
* **\***: This indicates the **day of the week** (any day).

1. **schedule cronjob at monthly once?**

* 0 0 1 \* \* command\_to\_execute

1. **schedule cronjob at daily once?**

* 0 0 \* \* \* command\_to\_execute

1. **schedule cronjob at hourly once?**

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

1. **schedule the cronjob at machine reboot?**

* To schedule a cron job to run at machine reboot, you can use the special string @reboot in your crontab.
* This allows you to specify a command that will execute once the machine starts up.
* **@reboot command\_to\_execute**

1. **schedule task every weekday (Monday to Friday) at 10pm**.

* 0 22 \* \* 1-5 command\_to\_execute
* **0**: This indicates the **minute** (0 minutes past the hour).
* **22**: This indicates the **hour** (10 PM in 24-hour format).
* **\***: This indicates every **day of the month**.
* **\***: This indicates every **month**.
* **1-5**: This indicates the **days of the week** (Monday to Friday, where 1 is Monday and 5 is Friday).

1. **Perform a backup at midnight every Tuesday.**

* 0 0 \* \* 2 command\_to\_execute
* 0: This indicates the minute (0 minutes past the hour).
* 0: This indicates the hour (midnight).
* \*: This indicates every day of the month.
* \*: This indicates every month.
* 2: This indicates the day of the week (Tuesday, where 0 is Sunday and 2 is Tuesday).

1. **Perform monitoring every minute during January, February, and May.**

* \* \* 1,2,5 \* command\_to\_execute

1. **Clear the cache every 10 minutes at 5am, starting from 5:10am.**

* 10-59/10 5 \* \* \* /path/to/your/clear\_cache.sh

1. **Make the task run quarterly on the first day of the month at 8am.**

* 0 8 1 1,4,7,10 /path/to/your/quarterly\_task.sh

1. **Create a backup every hour.**

* 0 \* \* \* \* /path/to/your/backup.sh

1. **Include multiple tasks on a single cron job. Useful for scheduling multiple tasks to run at the same time.**

You can include multiple tasks in a single cron job by chaining the commands together using &&, ;, or ||. Here’s how you can do it:

**Using &&**

* The && operator will execute the next command only if the previous command was successful (i.e., it exited with a status of 0).
* \* \* \* \* command1 && command2 && command3

**Using ;**

* The ; operator will execute the next command regardless of whether the previous command was successful or not.
* \* \* \* \* \* command1; command2; command3

**Using ||**

* The || operator will execute the next command only if the previous command failed (i.e., it exited with a status other than 0).
* \* \* \* \* \* command1 || command2 || command3

1. **Clear cache every time you turn on the system.**

* Use @reboot command\_to\_execute to schedule a task to run **every time the system starts**.
* Replace /path/to/your/clear\_cache.sh with the path to your actual cache-clearing script or command.

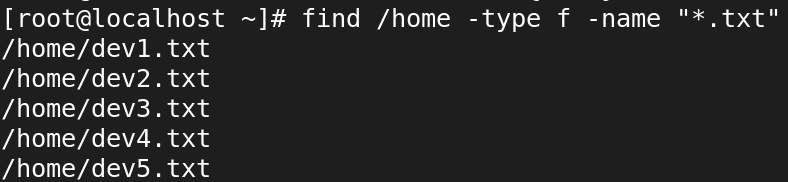
**Network**

1. **Add additional ethernet card and assign static ip**
2. Add additional ethernet card and assign dynamic ip
3. how to check the server IP also show the current IP details?
4. how to check the server routes also show current route details?
5. add two ethernet card ans assignee one ip?
6. assign two ip address for one Ethernet card?
7. how to check the server hostname
8. change server hostname as server.glotechcorp.com permanently

**Find**

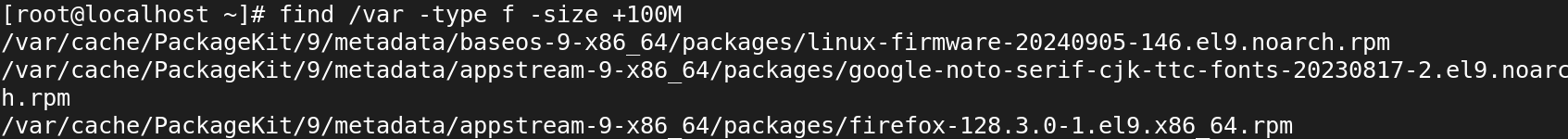
1. **How would you use the `find` command to search for all files with the `.txt` extension in the `/home` directory?**

* find /home -type f -name “\*.txt”

****

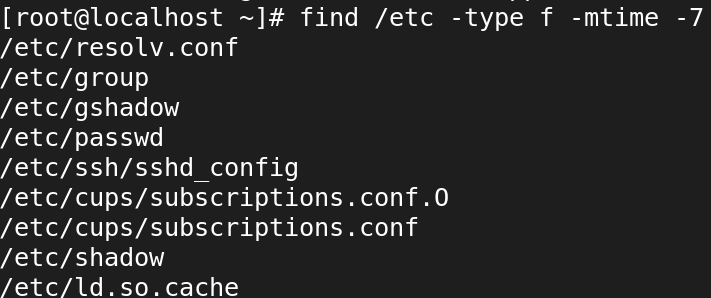
1. **Write a command using `find` to search for files larger than 100MB in the `/var` directory.**

* find /var -type f -size +100M



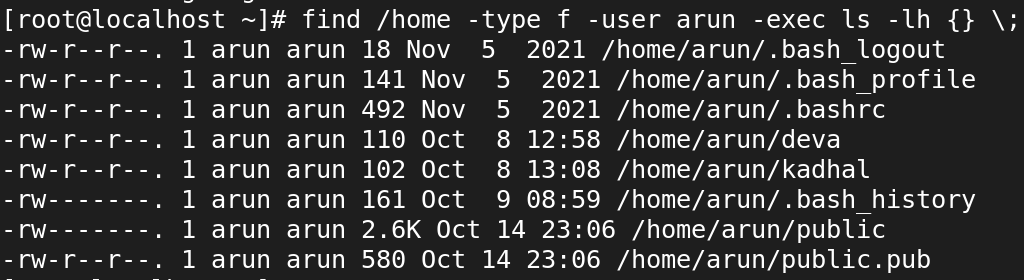
1. **Use `find` to locate all files in the `/etc` directory that were modified within the last 7 days.**

* find /etc -type f -mtime -7



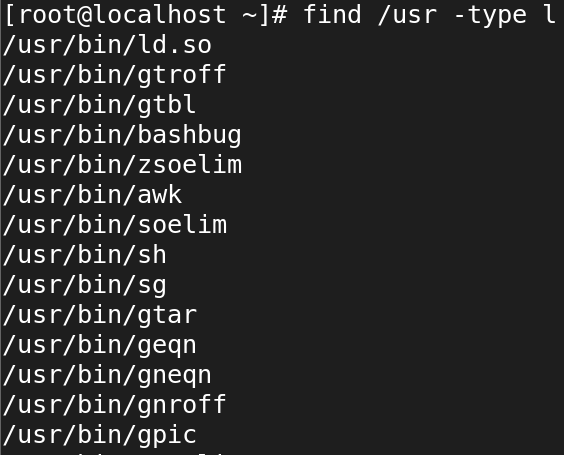
1. **How can you use `find` to search for files owned by a specific user, for example, `user1`, in the `/home` directory?**

* find /home -type f -user username -exec ls -lh {} \;



1. **Find all symbolic links within the `/usr` directory using the `find` command.**

* find /usr -type l

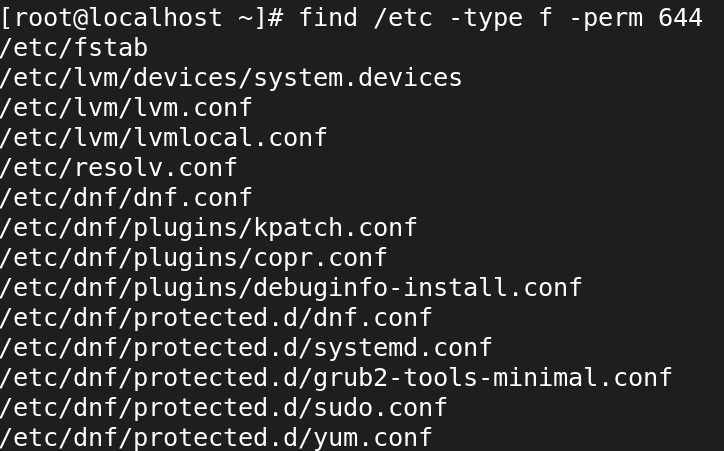


1. **Write a `find` command to delete all `.log` files in the `/tmp` directory.**

* find /tmp -type f -name "\*.log" -delete

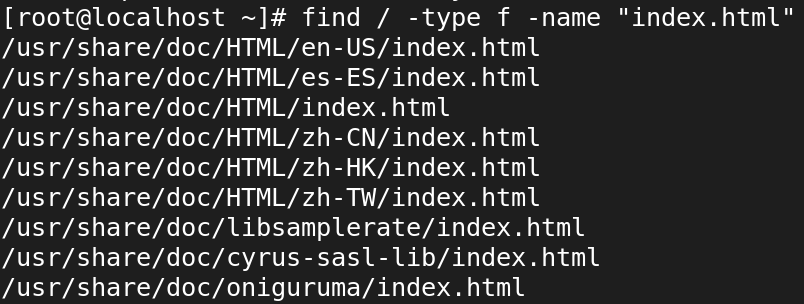
1. **How would you use the `find` command to locate files with the permissions `644` within `/etc`?**

* find /etc -type f -perm 644

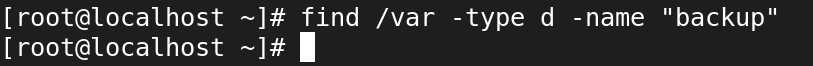
****

1. **Use the `find` command to locate all files named `index.html` starting from the root directory (`/`).**

* find / -type f -name “index.html”

****

1. **Write a command using `find` to search for directories (not files) named `backup` within the `/var` directory.**

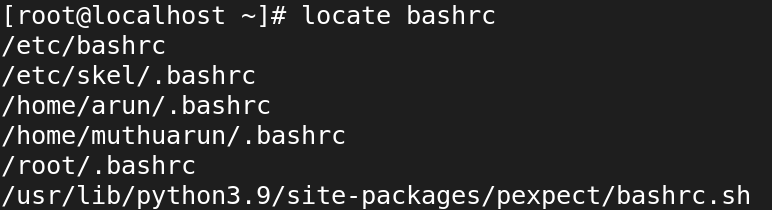
****

1. **How would you find all empty files within the `/home/user/Documents` directory using `find`?**

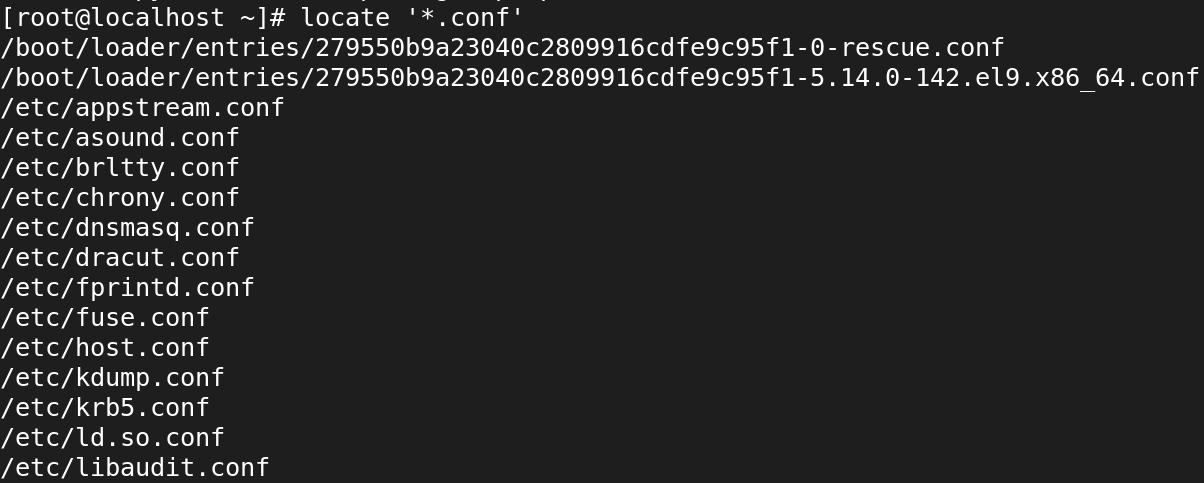
****

**Locate**

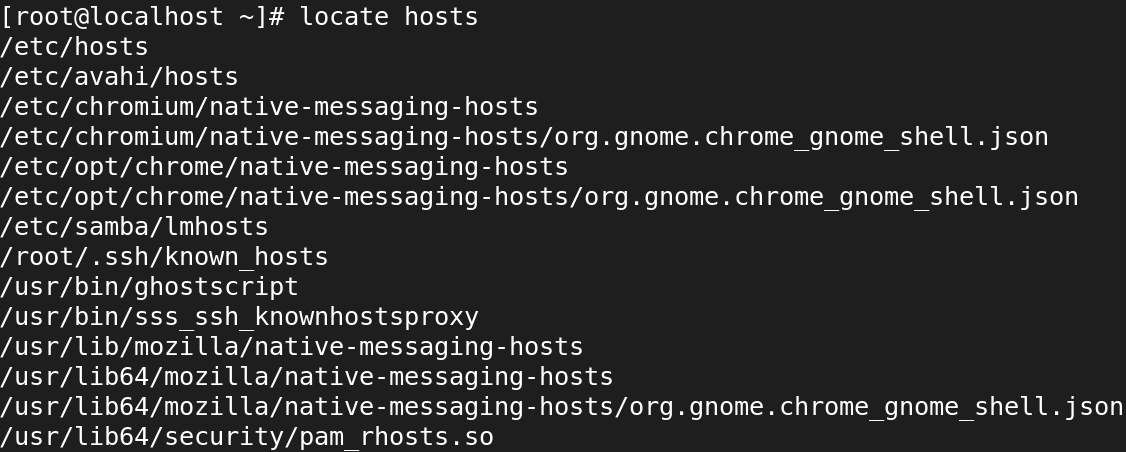
**1**. **How would you use the `locate` command to find all files containing the name `bashrc`?**



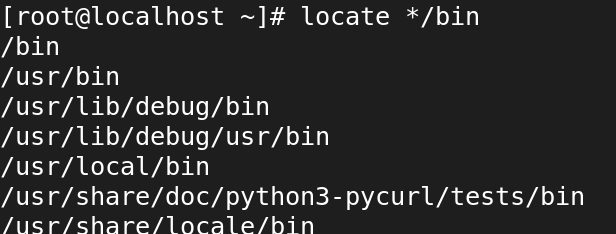
**2. Write a `locate` command to search for files with the `.conf` extension on your system.**

****

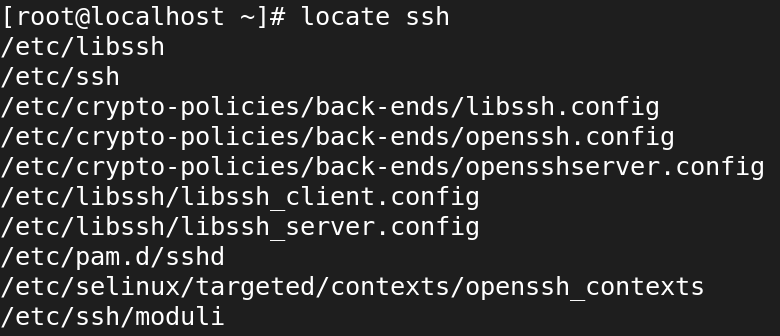
**3. How can you use `locate` to find all occurrences of a file named `hosts` in the filesystem?**



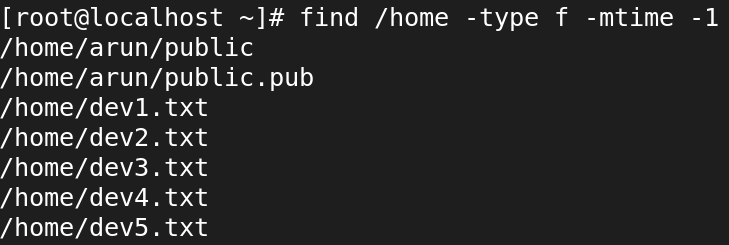
**4. Use the `locate` command to find directories containing the name `bin` in their path.**

****

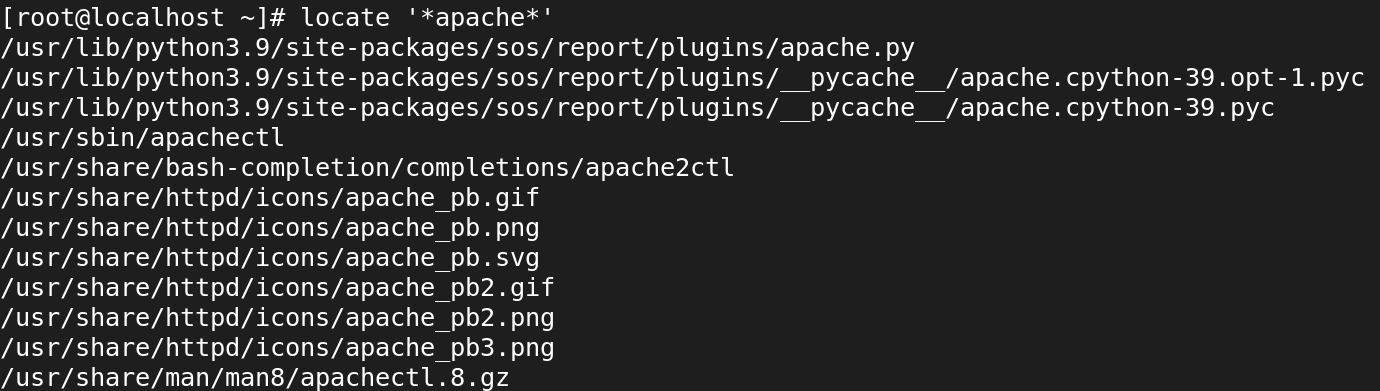
**5. Write a `locate` command to find files related to the program `ssh`.**

****

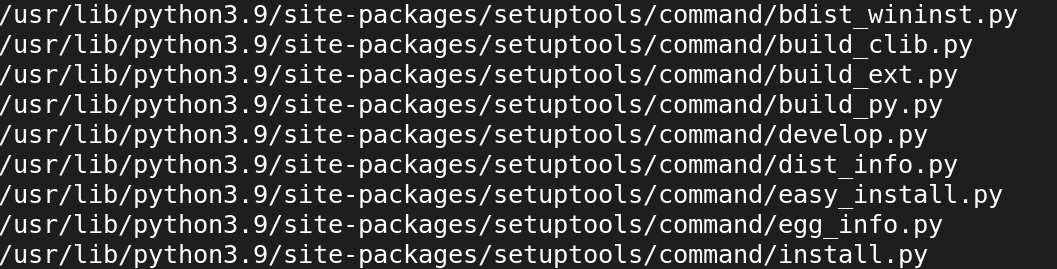
**6. How would you use `locate` to search for files that were updated in the last 24 hours?**

****

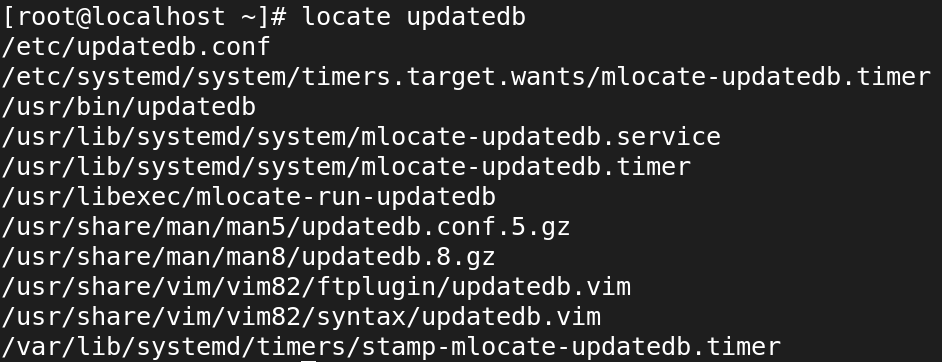
**7. Use the `locate` command to find all files and directories containing the word `apache`.**



**8. Write a `locate` command to search for all files in the `/usr` directory that contain the string `lib`.**

****

**9. How can you update the `locate` database manually before performing a search?**

****

**10. Write a `locate` command to find all files with the `.jpg` extension.**

