Configuring a **YUM server** and **YUM client** on **RHEL (Red Hat Enterprise Linux)** involves setting up a repository on the server to host RPM packages and then configuring the client machines to access and install packages from that repository.

Here’s a detailed guide for setting up both the **YUM server** and **YUM client** in RHEL.

1. YUM Server Configuration (Creating a YUM Repository on RHEL Server)

A YUM server is used to host a local repository of RPM packages. This repository can be accessed by multiple client machines to install or update packages.

Prerequisites:

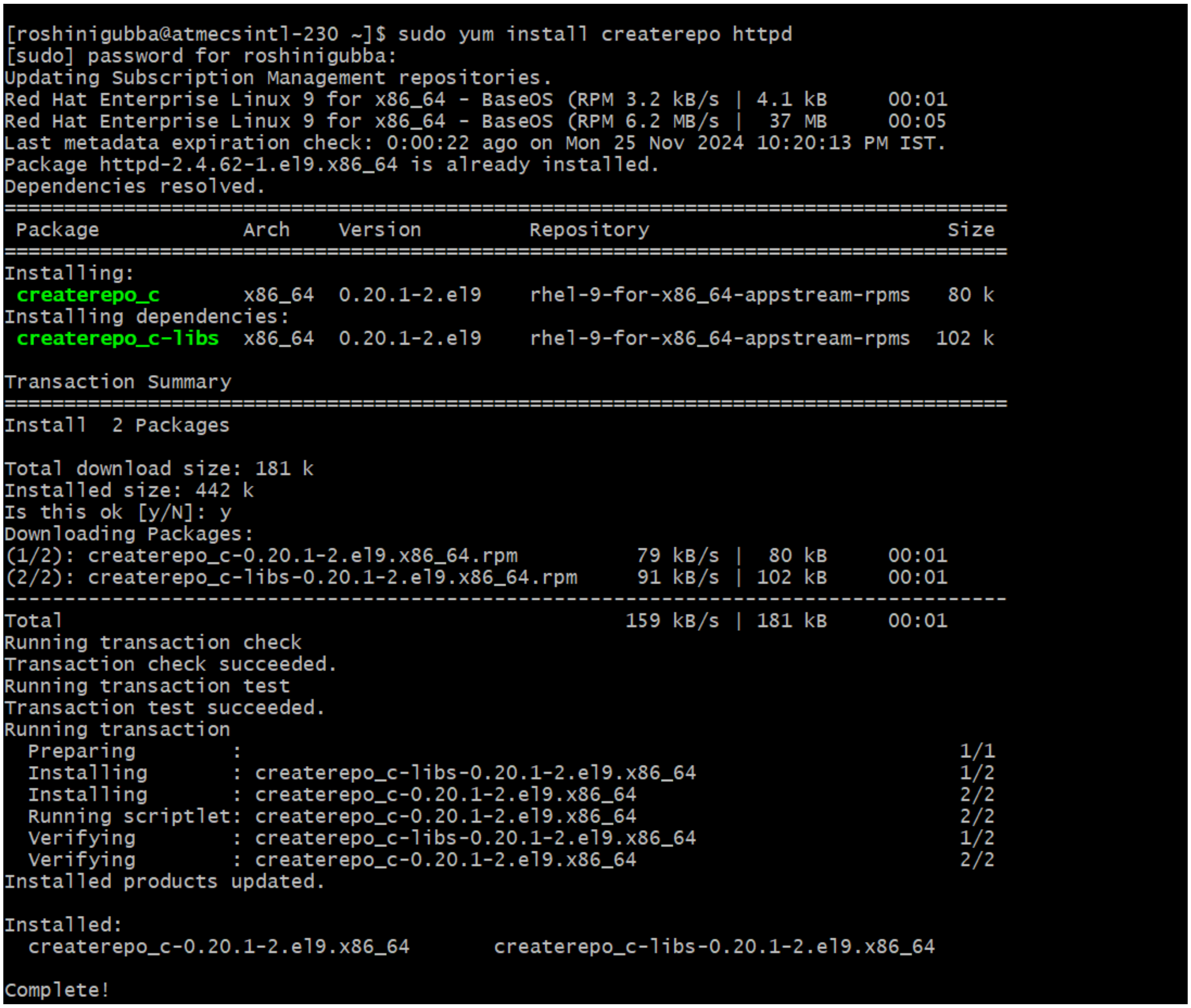
- A RHEL server (can be a virtual or physical machine).
- A web server like Apache to serve the repository.

Steps to Set Up YUM Server on RHEL:

Step 1: Install Required Packages

You will need the `createrepo` package to create metadata for the YUM repository, and the `httpd` package to serve the repository over HTTP.

```
sudo yum install createrepo httpd
```

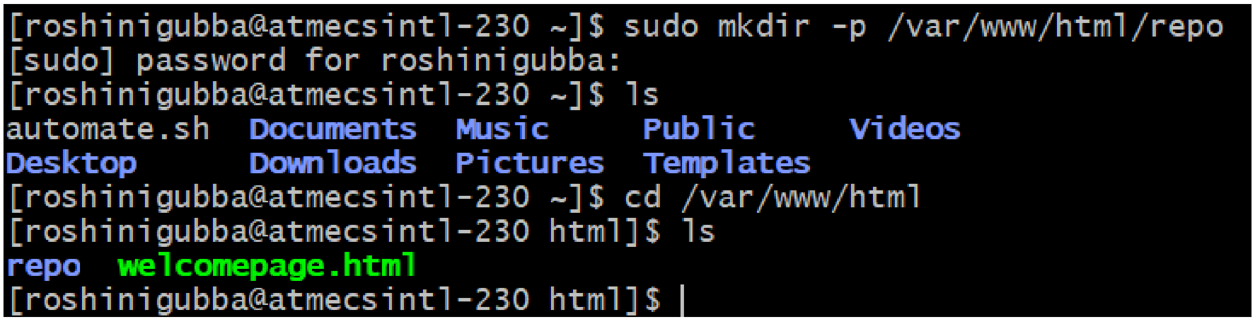


```
[roshinigubba@atmecsintl-230 ~]$ sudo yum install createrepo httpd
[sudo] password for roshinigubba:
Updating Subscription Management repositories.
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPM 3.2 kB/s | 4.1 kB     00:01
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPM 6.2 MB/s |  37 MB     00:05
Last metadata expiration check: 0:00:22 ago on Mon 25 Nov 2024 10:20:13 PM IST.
Package httpd-2.4.62-1.el9.x86_64 is already installed.
Dependencies resolved.
==================================================================================
 Package             Arch    Version        Repository                       Size
==================================================================================
Installing:
 createrepo_c        x86_64  0.20.1-2.el9   rhel-9-for-x86_64-appstream-rpms  80 k
Installing dependencies:
 createrepo_c-libs   x86_64  0.20.1-2.el9   rhel-9-for-x86_64-appstream-rpms 102 k

Transaction Summary
==================================================================================
Install  2 Packages

Total download size: 181 k
Installed size: 442 k
Is this ok [y/N]: y
Downloading Packages:
(1/2): createrepo_c-0.20.1-2.el9.x86_64.rpm          79 kB/s |  80 kB     00:01
(2/2): createrepo_c-libs-0.20.1-2.el9.x86_64.rpm     91 kB/s | 102 kB     00:01
----------------------------------------------------------------------------------
Total                                               159 kB/s | 181 kB     00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                          1/1
  Installing       : createrepo_c-libs-0.20.1-2.el9.x86_64                    1/2
  Installing       : createrepo_c-0.20.1-2.el9.x86_64                         2/2
  Running scriptlet: createrepo_c-0.20.1-2.el9.x86_64                         2/2
  Verifying        : createrepo_c-libs-0.20.1-2.el9.x86_64                    1/2
  Verifying        : createrepo_c-0.20.1-2.el9.x86_64                         2/2
Installed products updated.

Installed:
  createrepo_c-0.20.1-2.el9.x86_64       createrepo_c-libs-0.20.1-2.el9.x86_64

Complete!
```

Step 2: Create a Directory for the Repository

Create a directory where RPM packages will be stored.

sudo mkdir -p /var/www/html/repo

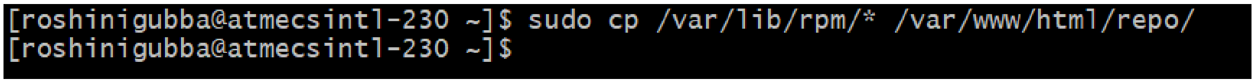


```
[roshinigubba@atmecsintl-230 ~]$ sudo mkdir -p /var/www/html/repo
[sudo] password for roshinigubba:
[roshinigubba@atmecsintl-230 ~]$ ls
automate.sh  Documents  Music     Public     Videos
Desktop      Downloads  Pictures  Templates
[roshinigubba@atmecsintl-230 ~]$ cd /var/www/html
[roshinigubba@atmecsintl-230 html]$ ls
repo  welcomepage.html
[roshinigubba@atmecsintl-230 html]$ |
```

Step 3: Add RPM Packages

You can copy RPM packages from another source (e.g., downloaded packages or CD/DVD). For this example, we assume you have the RPMs ready to be served.

sudo cp /path/to/rpms/*.rpm /var/www/html/repo/

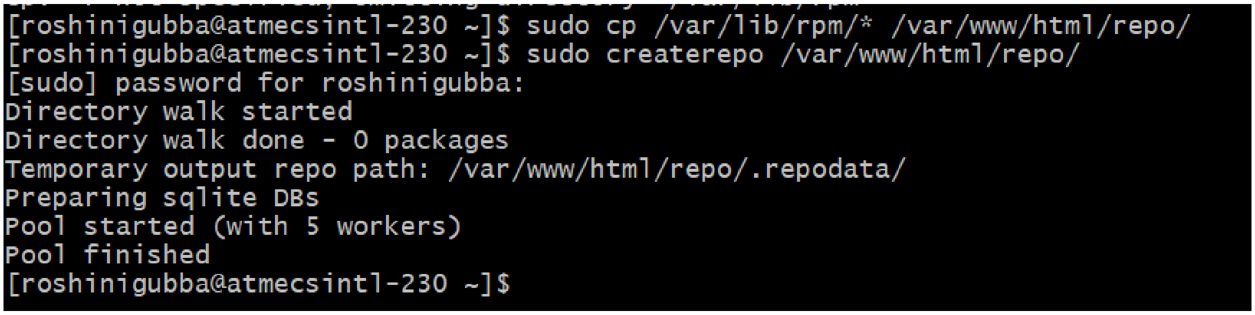


```
[roshinigubba@atmecsintl-230 ~]$ sudo cp /var/lib/rpm/* /var/www/html/repo/
[roshinigubba@atmecsintl-230 ~]$
```

Step 4: Create Repository Metadata

Use the `createrepo` command to generate repository metadata (which YUM clients will use to browse and install packages).

sudo createrepo /var/www/html/repo/



```
[roshinigubba@atmecsintl-230 ~]$ sudo cp /var/lib/rpm/* /var/www/html/repo/
[roshinigubba@atmecsintl-230 ~]$ sudo createrepo /var/www/html/repo/
[sudo] password for roshinigubba:
Directory walk started
Directory walk done - 0 packages
Temporary output repo path: /var/www/html/repo/.repodata/
Preparing sqlite DBs
Pool started (with 5 workers)
Pool finished
[roshinigubba@atmecsintl-230 ~]$
```

Step 5: Configure Apache to Serve the Repository

To serve the repository over HTTP, you need to configure Apache and ensure it's running.

Start and enable the Apache service:

sudo systemctl start httpd

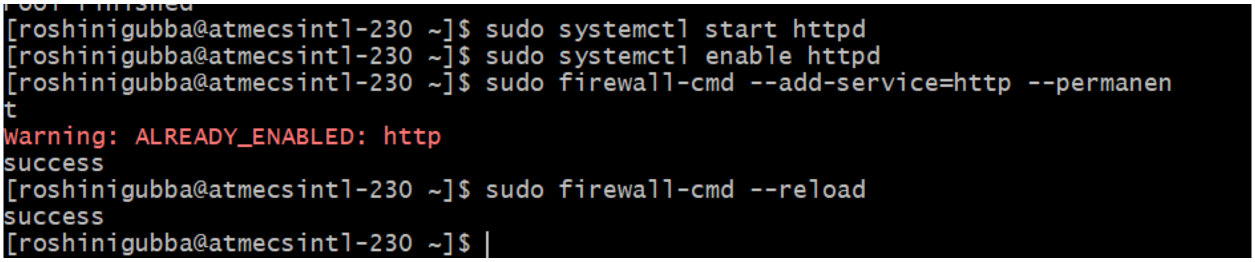
sudo systemctl enable httpd

1.

Allow HTTP traffic through the firewall:

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

sudo firewall-cmd --reload



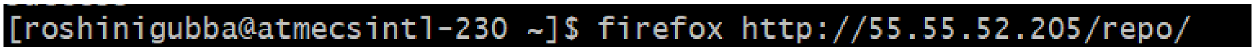
```
[roshinigubba@atmecsintl-230 ~]$ sudo systemctl start httpd
[roshinigubba@atmecsintl-230 ~]$ sudo systemctl enable httpd
[roshinigubba@atmecsintl-230 ~]$ sudo firewall-cmd --add-service=http --permanen
t
Warning: ALREADY_ENABLED: http
success
[roshinigubba@atmecsintl-230 ~]$ sudo firewall-cmd --reload
success
[roshinigubba@atmecsintl-230 ~]$ |
```

2.

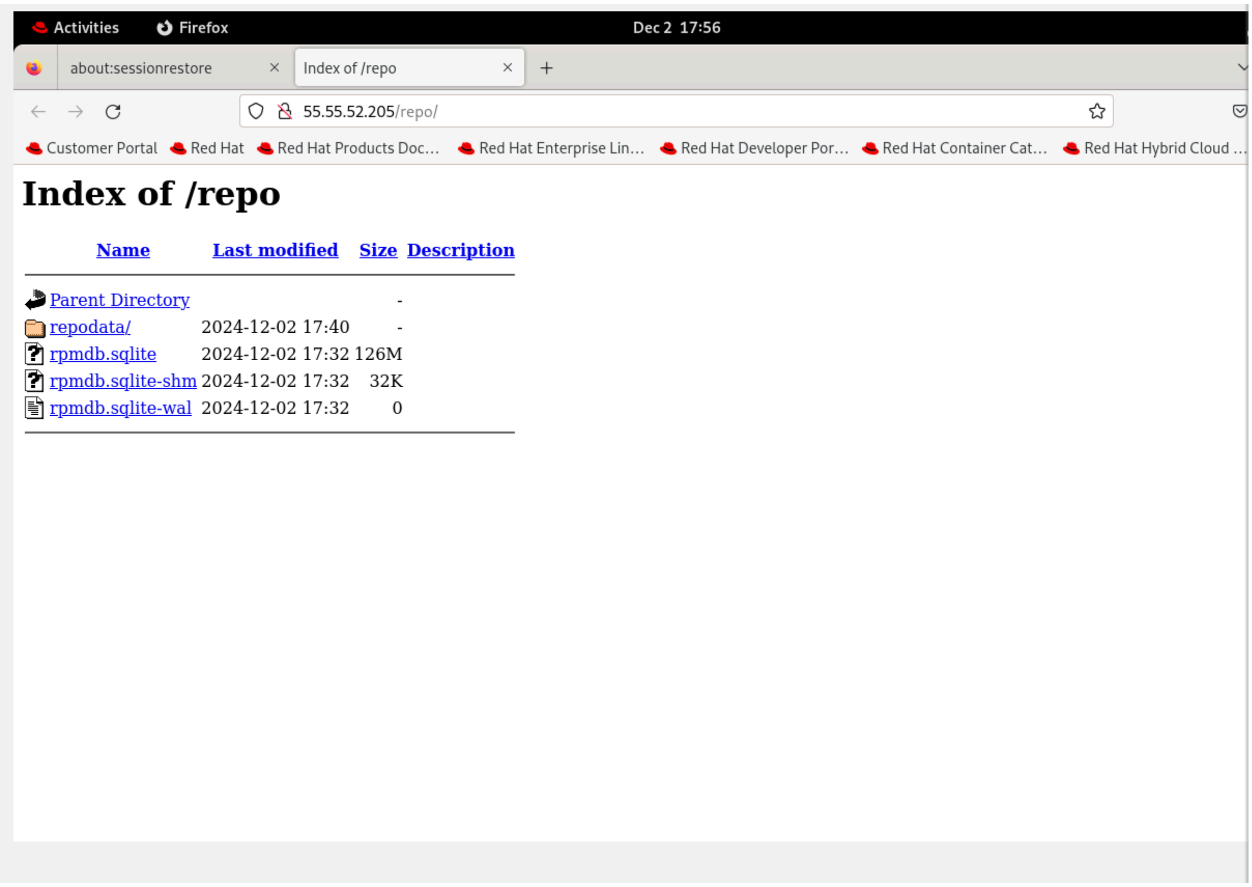
Confirm that Apache is working by opening a browser and navigating to:

http://<server-ip>/repo/

3. Replace `<server-ip>` with your server's IP address. You should see the list of RPM packages hosted in `/var/www/html/repo/`.



```
[roshinigubba@atmecsintl-230 ~]$ firefox http://55.55.52.205/repo/
```



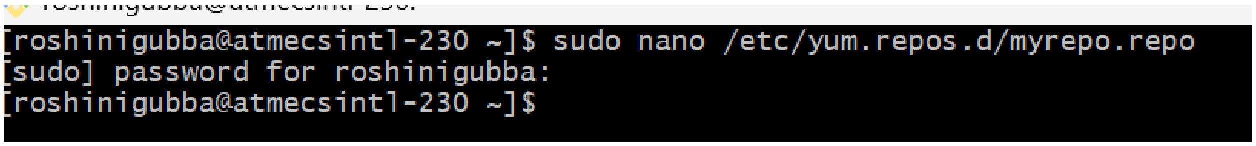
Index of /repo

```
      Name            Last modified     Size  Description
Parent Directory                          -
repodata/          2024-12-02 17:40       -
rpmdb.sqlite       2024-12-02 17:32    126M
rpmdb.sqlite-shm   2024-12-02 17:32     32K
rpmdb.sqlite-wal   2024-12-02 17:32       0
```

Step 6: Configure YUM Repository on the Server

On the YUM server, you may want to create a `.repo` file for easier management. This file will point to the local repository and can be used to configure the server as a YUM source.

```
sudo nano /etc/yum.repos.d/myrepo.repo
```

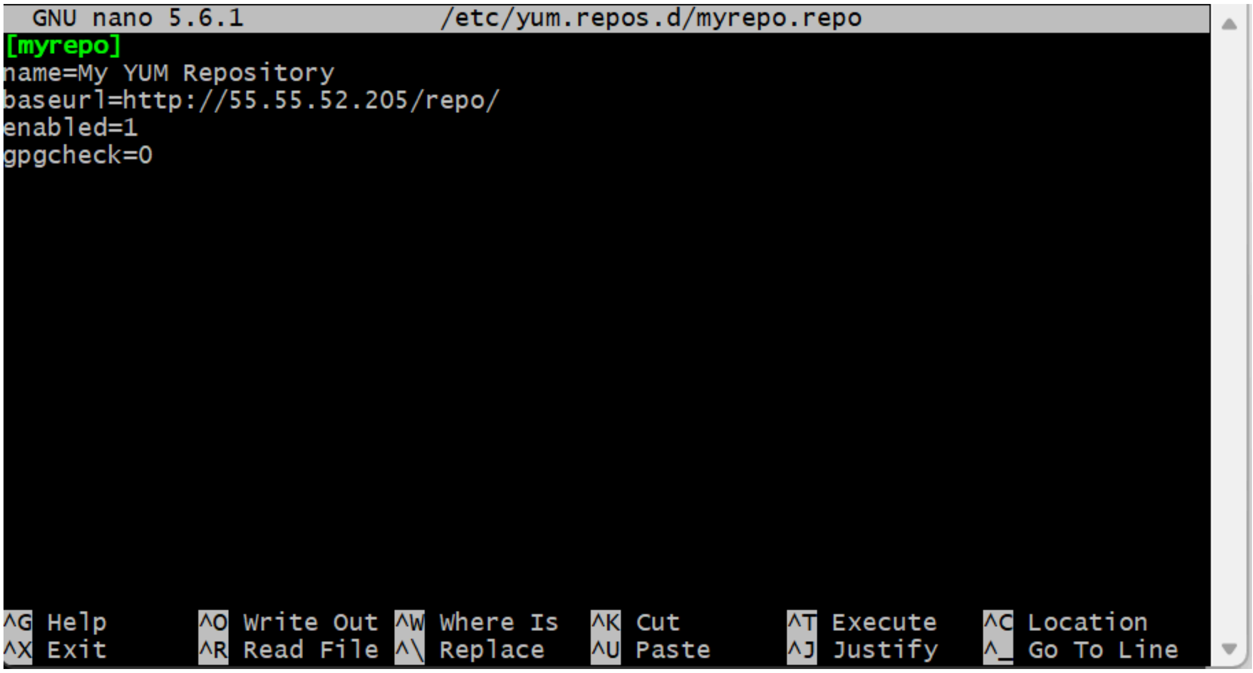


```
[roshinigubba@atmecsintl-230 ~]$ sudo nano /etc/yum.repos.d/myrepo.repo
[sudo] password for roshinigubba:
[roshinigubba@atmecsintl-230 ~]$
```

Add the following content:

```
[myrepo]

name=My YUM Repository

baseurl=http://<server-ip>/repo/

enabled=1

gpgcheck=0
```

- **`baseurl`**: The URL where the YUM repository is located (in this case, your Apache server).
- **`enabled`**: Set to `1` to enable the repository.
- **`gpgcheck`**: Set to `0` if you don’t use GPG keys for package signing (set to `1` if you do).



```
[myrepo]
name=My YUM Repository
baseurl=http://55.55.52.205/repo/
enabled=1
gpgcheck=0
```

2. YUM Client Configuration (Accessing YUM Repository on RHEL Client)

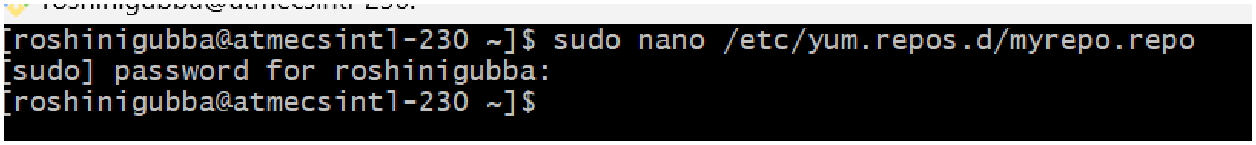
Once the YUM server is set up, client machines need to be configured to use the server's repository.

Steps to Set Up YUM Client on RHEL:

Step 1: Create YUM Repository Configuration File

On the client machine, create a `.repo` file in the `/etc/yum.repos.d/` directory to define the repository.

sudo nano /etc/yum.repos.d/myrepo.repo



```
[roshinigubba@atmecsintl-230 ~]$ sudo nano /etc/yum.repos.d/myrepo.repo
[sudo] password for roshinigubba:
[roshinigubba@atmecsintl-230 ~]$
```

Add the following configuration:

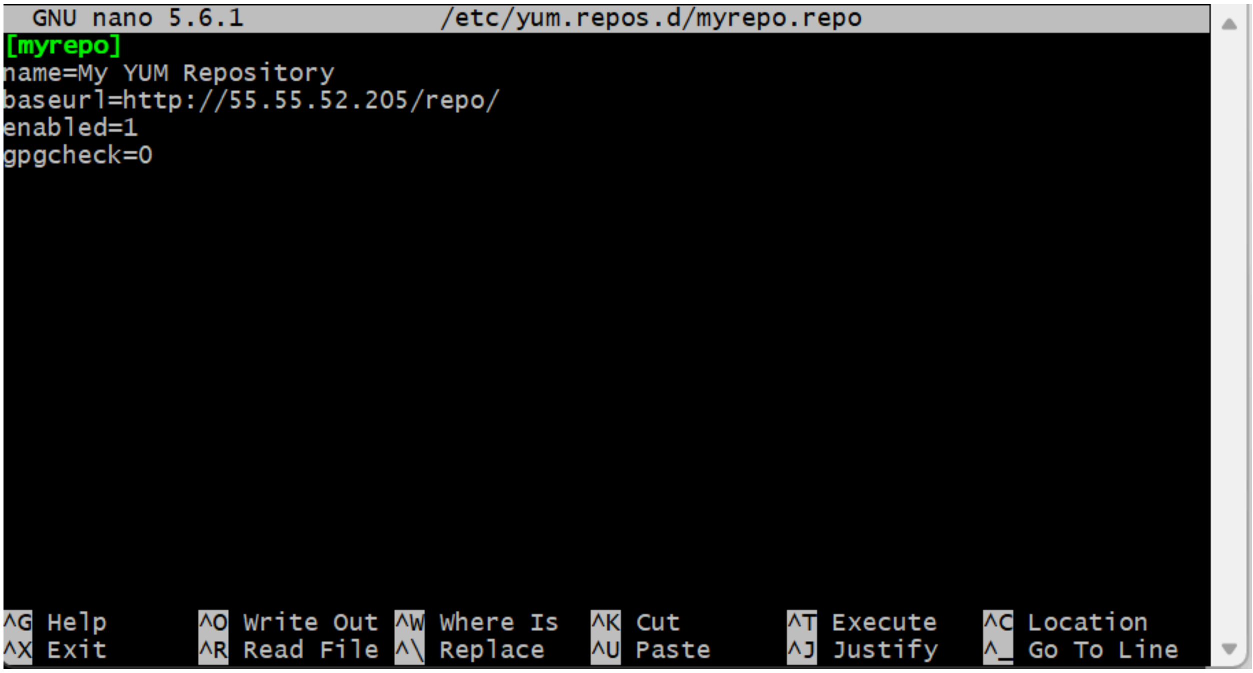
[myrepo]

name=My YUM Repository

```
baseurl=http://<server-ip>/repo/

enabled=1

gpgcheck=0
```

Replace `<server-ip>` with the IP address of the YUM server.

- **`baseurl`**: The URL of the YUM repository hosted on the server.
- **`enabled`**: Set to `1` to enable the repository.
- **`gpgcheck`**: Set to `0` if you are not using GPG keys for package signing.

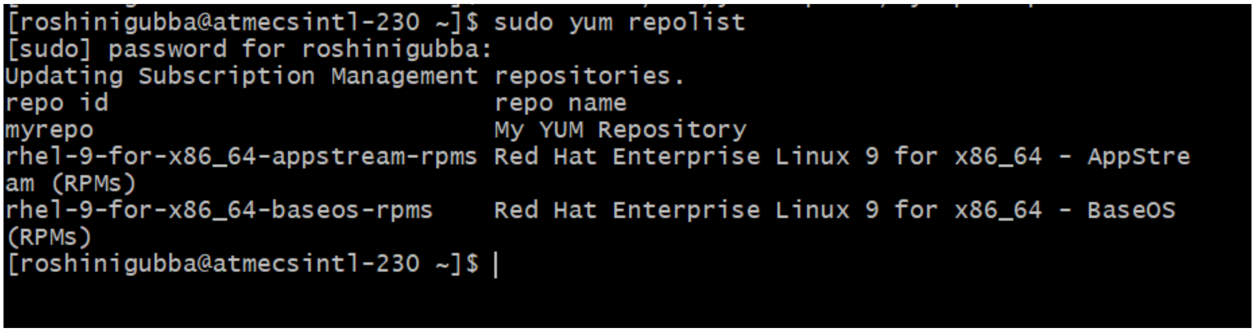


```
[myrepo]
name=My YUM Repository
baseurl=http://55.55.52.205/repo/
enabled=1
gpgcheck=0
```

Step 2: Verify Repository Configuration

After creating the `.repo` file, you can verify that the client machine recognizes the repository by running the following command:

```
sudo yum repolist
```



```
[roshinigubba@atmecsintl-230 ~]$ sudo yum repolist
[sudo] password for roshinigubba:
Updating Subscription Management repositories.
repo id                         repo name
myrepo                          My YUM Repository
rhel-9-for-x86_64-appstream-rpms Red Hat Enterprise Linux 9 for x86_64 - AppStre
am (RPMs)
rhel-9-for-x86_64-baseos-rpms    Red Hat Enterprise Linux 9 for x86_64 - BaseOS
(RPMs)
[roshinigubba@atmecsintl-230 ~]$ |
```

This command will display the list of repositories configured on the client, including the one you just added.

Step 3: Install Packages from the Repository

Once the repository is configured, you can install packages from the server’s repository. For example, to install `httpd`:

sudo yum install httpd

Step 4: Update Packages from the Repository

To update packages on the client machine from the repository, use:

sudo yum update

Step 5: Clear YUM Cache (Optional)

If you need to clear YUM’s cache (e.g., to ensure fresh metadata), use:

sudo yum clean all

---

3. Additional Configuration Options

Using GPG for Package Signing

If you are signing your RPM packages with GPG keys for integrity and authenticity, you can enable `gpgcheck` in both the YUM server and client configuration:

1. **On the YUM Server**:

Sign your RPM packages with a GPG key.

```
 rpm --import /path/to/RPM-GPG-KEY
```

    -
    - Enable GPG signature checking by setting `gpgcheck=1` in your `.repo` file on the server.

2. **On the YUM Client**:

Import the GPG key:

```
 rpm --import /path/to/RPM-GPG-KEY
```

    -
    - Enable GPG check by setting `gpgcheck=1` in the `.repo` file on the client machine.

Disabling Specific Repositories

You can disable a repository temporarily using the `--disablerepo` flag:

```
sudo yum install <package-name> --disablerepo=myrepo
```

Using Multiple Repositories

If you want to configure multiple repositories on the client or server, you can create multiple `.repo` files in the `/etc/yum.repos.d/` directory. Each repository will have its own configuration.

Summary

1. **YUM Server (RHEL)**:
    - Install `createrepo` and `httpd`.

    - Create a directory to store RPM packages and generate metadata using `createrepo`.
    - Configure Apache to serve the repository over HTTP.
    - Create a `.repo` file to point to the local repository.
2. **YUM Client (RHEL)**:
    - Create a `.repo` file to point to the YUM server's repository.
    - Run `yum repolist` to verify the repository configuration.
    - Install packages with `yum install` and update packages with `yum update`.

By following these steps, you can effectively configure and use a local YUM repository in a RHEL environment, making package management more efficient across multiple systems.