Step 1: Update Repository Package Lists

sudo yum -y update

```
[root@svr1 ~]# yum update
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Resolving Dependencies
--> Running transaction check
---> Package NetworkManager.x86_64 1:1.18.8-1.el7 will be updated
---> Package NetworkManager.x86_64 1:1.18.8-2.el7_9 will be an update
---> Package NetworkManager-adsl.x86_64 1:1.18.8-1.el7 will be updated
---> Package NetworkManager-adsl.x86_64 1:1.18.8-2.el7_9 will be an update
---> Package NetworkManager-bluetooth.x86_64 1:1.18.8-1.el7 will be updated
---> Package NetworkManager-bluetooth.x86_64 1:1.18.8-2.el7_9 will be an update
```

Step 2: Install Extra Packages for Enterprise Linux (EPEL)

yum install -y epel-release

```
[root@svr1 ~]# yum install -y epel-release
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                        Arch                           Version
================================================================================
Installing:
 epel-release                   noarch                         7-11
```

Step 3: Install Nginx

yum –y install nginx

```
[root@svr1 ~]# yum install nginx -y
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * epel: epel.excellmedia.net
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Resolving Dependencies
--> Running transaction check
---> Package nginx.x86_64 1:1.20.1-10.el7 will be installed
--> Processing Dependency: nginx-filesystem = 1:1.20.1-10.el7 for package: 1:nginx-1.20.1-10.el7.x86_64
--> Processing Dependency: libcrypto.so.1.1(OPENSSL_1_1_0)(64bit) for package: 1:nginx-1.20.1-10.el7.x86_64
--> Processing Dependency: libssl.so.1.1(OPENSSL_1_1_0)(64bit) for package: 1:nginx-1.20.1-10.el7.x86_64
--> Processing Dependency: libssl.so.1.1(OPENSSL_1_1_1)(64bit) for package: 1:nginx-1.20.1-10.el7.x86_64
--> Processing Dependency: nginx-filesystem for package: 1:nginx-1.20.1-10.el7.x86_64
--> Processing Dependency: libcrypto.so.1.1()(64bit) for package: 1:nginx-1.20.1-10.el7.x86_64
--> Processing Dependency: libssl.so.1.1()(64bit) for package: 1:nginx-1.20.1-10.el7.x86_64
--> Running transaction check
---> Package nginx-filesystem.noarch 1:1.20.1-10.el7 will be installed
---> Package openssl11-libs.x86_64 1:1.1.1k-5.el7 will be installed
--> Finished Dependency Resolution
```

Step 4: Start Nginx Service

systemctl enable nginx

systemctl start nginx

systemctl status nginx

```
[root@svr1 ~]# systemctl enable nginx
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/s
[root@svr1 ~]# systemctl start nginx
[root@svr1 ~]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2023-07-16 14:58:32 IST; 10ms ago
  Process: 76113 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
  Process: 76110 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
  Process: 76108 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
 Main PID: 76115 (nginx)
    Tasks: 3
   CGroup: /system.slice/nginx.service
           ├─76115 nginx: master process /usr/sbin/nginx
           ├─76116 nginx: worker process
           └─76117 nginx: worker process
```

Step 6: Configure Firewall to Allow Traffic (for this, I have enabled firewall)

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

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

firewall-cmd --reload

```
[root@svr1 ~]# firewall-cmd --zone=public --permanent --add-service=http
success
[root@svr1 ~]# firewall-cmd --zone=public --permanent --add-service=https
success
[root@svr1 ~]# firewall-cmd --reload
success
[root@svr1 ~]#
```

Step 7: Verify Nginx Install

ip a

```
[root@svr1 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:66:43:97 brd ff:ff:ff:ff:ff:ff
    inet 192.168.88.132/24 brd 192.168.88.255 scope global noprefixroute dynamic ens33
       valid_lft 996sec preferred_lft 996sec
    inet6 fe80::b018:2fcb:f1ec:e8f5/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:f1:8b:5e brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:f1:8b:5e brd ff:ff:ff:ff:ff:ff
[root@svr1 ~]#
```

Access your web browser with this IP address:

Or, you might also receive below page.