Lab: Environment Setup

Introduction:

This Lab Exercise is to Perform Some of the Environment Specific Configuration Settings.

Objectives:

- **Hostname Resolutions**
- **Disabling SELinux**
- **Disabling Firewalld Service**
- **Installing and Configuring Chrony (NTP) Service**
- **Installing and Configuring EPEL Repository**

Note: (**Run It on all of the Servers**)

Environment Details:

Below table contains details of the servers that we will be using in this lab setup.

Host Name	IP Address	Role	OS	RAM
eoc-controller	192.168.100.150	Controller Node	CentOS-8	16GB
eoc-node1	192.168.100.151	Managed Node	CentOS-8	4GB
eoc-node2	192.168.100.152	Managed Node	CentOS-8	4GB
eoc-node3	192.168.100.153	Managed Node	CentOS-8	4GB

1. Hostname Resolution

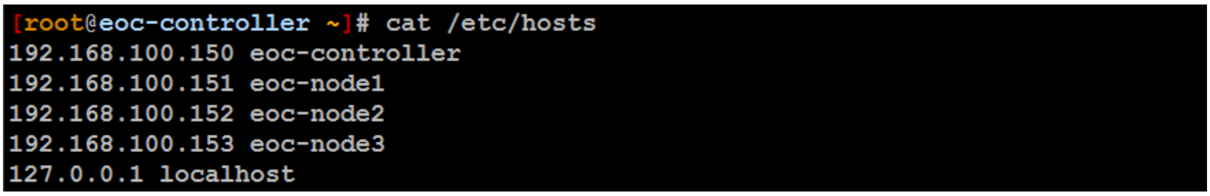
1.1 Add an entry to **/etc/hosts** file for **Local Name Resolution**.

```
# cat > /etc/hosts <<EOF
192.168.100.150 eoc-controller
192.168.100.151 eoc-node1
192.168.100.152 eoc-node2
192.168.100.153 eoc-node3
127.0.0.1 localhost
EOF
```

1.2 Let’s Verify the /etc/hosts file updated successfully, by executing the below command

```
# cat /etc/hosts
```

Output:

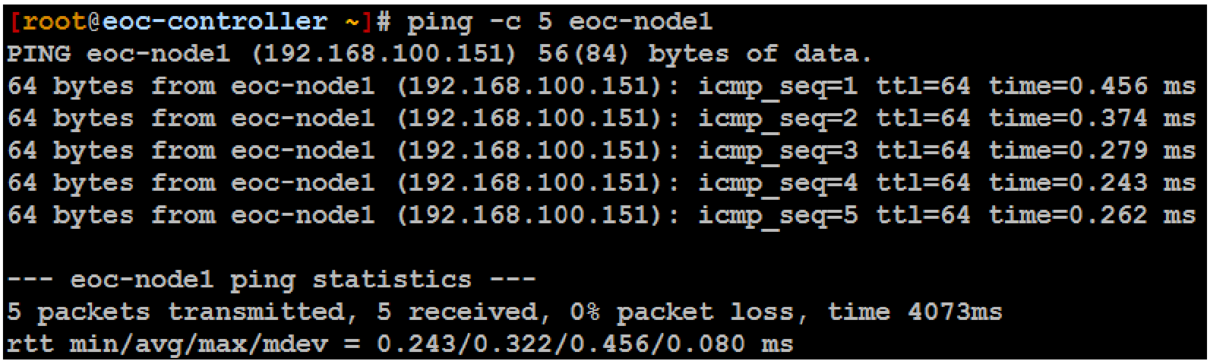


```
[root@eoc-controller ~]# cat /etc/hosts
192.168.100.150 eoc-controller
192.168.100.151 eoc-node1
192.168.100.152 eoc-node2
192.168.100.153 eoc-node3
127.0.0.1 localhost
```

1.3 Test network connectivity between servers to ensure name resolution is working.

```
# ping -c 5 eoc-node1
```

Output:



```
[root@eoc-controller ~]# ping -c 5 eoc-node1
PING eoc-node1 (192.168.100.151) 56(84) bytes of data.
64 bytes from eoc-node1 (192.168.100.151): icmp_seq=1 ttl=64 time=0.456 ms
64 bytes from eoc-node1 (192.168.100.151): icmp_seq=2 ttl=64 time=0.374 ms
64 bytes from eoc-node1 (192.168.100.151): icmp_seq=3 ttl=64 time=0.279 ms
64 bytes from eoc-node1 (192.168.100.151): icmp_seq=4 ttl=64 time=0.243 ms
64 bytes from eoc-node1 (192.168.100.151): icmp_seq=5 ttl=64 time=0.262 ms

--- eoc-node1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4073ms
rtt min/avg/max/mdev = 0.243/0.322/0.456/0.080 ms
```

2. Disabling SELinux

2.1 Make sure **SELinux** is **disabled**

Security-Enhanced Linux (SELinux) is a **mandatory access control** (MAC) security mechanism implemented in the kernel.

SELinux has **three basic** modes of operation, of which Enforcing is set as the installation default mode.

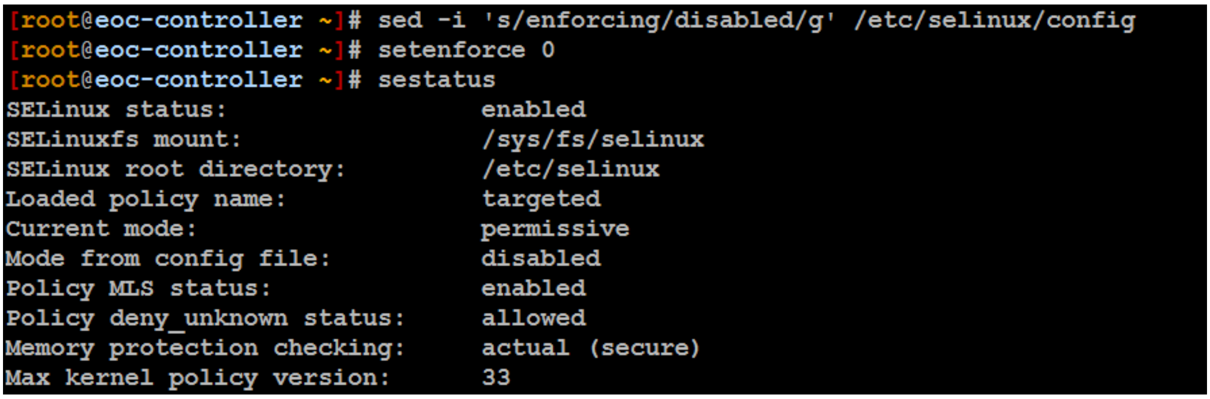
- **Enforcing:** The default mode which will enable and enforce the SELinux security policy on the system, denying access and logging actions
- **Permissive:** In Permissive mode, SELinux is enabled but will not enforce the security policy, only warn and log actions. Permissive mode is useful for troubleshooting SELinux issues.
- **Disabled:** SELinux is turned off

```
# sed -i 's/enforcing/disabled/g' /etc/selinux/config
```

```
# setenforce 0
```

```
# sestatus
```

Output:



```
[root@eoc-controller ~]# sed -i 's/enforcing/disabled/g' /etc/selinux/config
[root@eoc-controller ~]# setenforce 0
[root@eoc-controller ~]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          disabled
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

3. Disabling Firewalld Service

3.1 Let's disable **firewalld** Service.

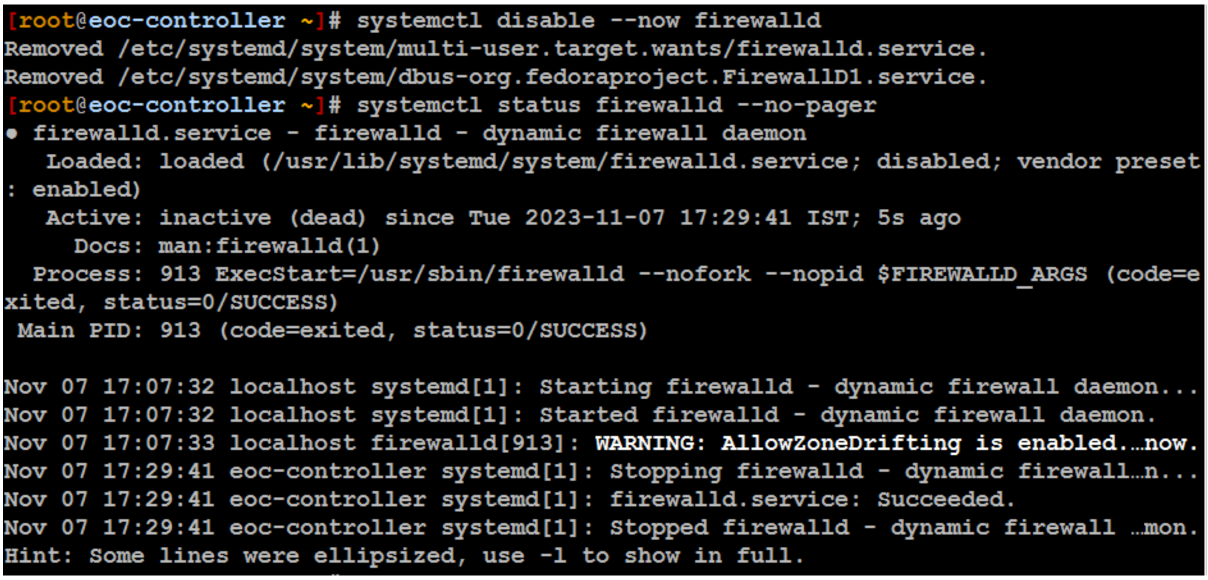
- A firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules.

- A firewall typically establishes a barrier between a trusted network and an untrusted network, such as the Internet.

```
# systemctl disable --now firewalld
```

```
# systemctl status firewalld --no-pager
```

Output:



```
[root@eoc-controller ~]# systemctl disable --now firewalld
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@eoc-controller ~]# systemctl status firewalld --no-pager
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset
: enabled)
   Active: inactive (dead) since Tue 2023-11-07 17:29:41 IST; 5s ago
     Docs: man:firewalld(1)
  Process: 913 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=e
xited, status=0/SUCCESS)
 Main PID: 913 (code=exited, status=0/SUCCESS)

Nov 07 17:07:32 localhost systemd[1]: Starting firewalld - dynamic firewall daemon...
Nov 07 17:07:32 localhost systemd[1]: Started firewalld - dynamic firewall daemon.
Nov 07 17:07:33 localhost firewalld[913]: WARNING: AllowZoneDrifting is enabled.…now.
Nov 07 17:29:41 eoc-controller systemd[1]: Stopping firewalld - dynamic firewall…n...
Nov 07 17:29:41 eoc-controller systemd[1]: firewalld.service: Succeeded.
Nov 07 17:29:41 eoc-controller systemd[1]: Stopped firewalld - dynamic firewall …mon.
Hint: Some lines were ellipsized, use -l to show in full.
```

4. Installing and Configuring Chrony (NTP) Service

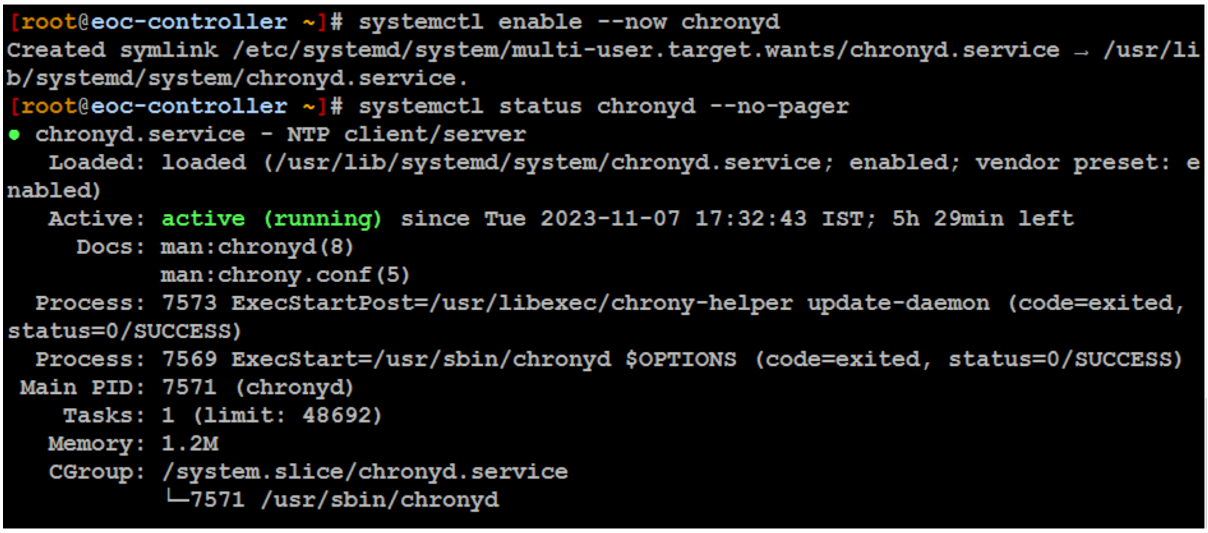
4.1 Let’s **Enable** and **Start** **Chrony** service (**NTP Server**).

```
# systemctl enable --now chronyd
```

```
# systemctl status chronyd --no-pager
```

- **Chrony** is a versatile implementation of the **Network Time Protocol** (NTP).
- The chrony suite installed by default and provides.
- **Chronyd** - Chrony daemon.
- **Chronyc** - Command-line utility.

Output:

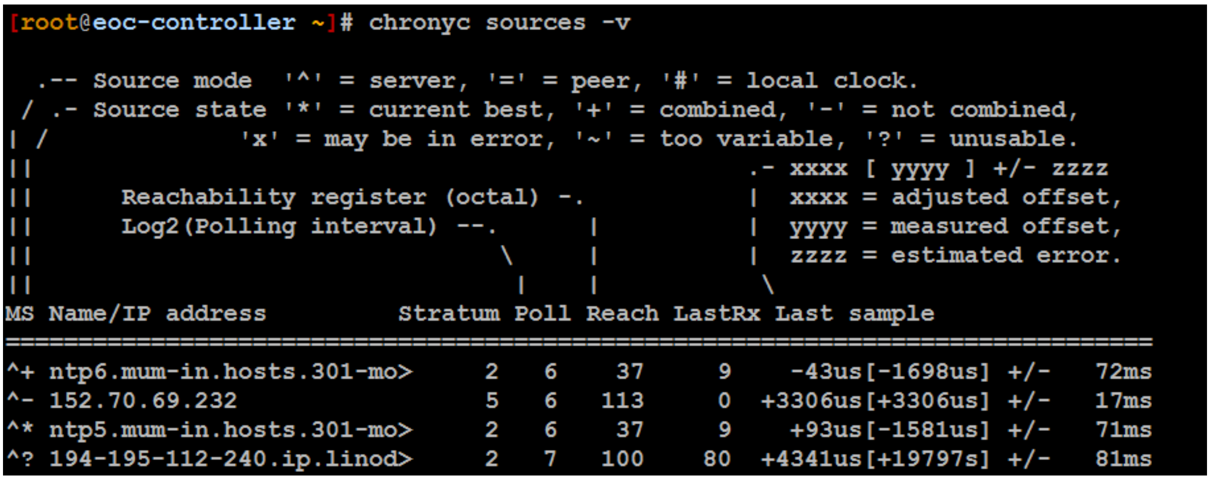


```
[root@eoc-controller ~]# systemctl enable --now chronyd
Created symlink /etc/systemd/system/multi-user.target.wants/chronyd.service → /usr/li
b/systemd/system/chronyd.service.
[root@eoc-controller ~]# systemctl status chronyd --no-pager
● chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; vendor preset: e
nabled)
   Active: active (running) since Tue 2023-11-07 17:32:43 IST; 5h 29min left
     Docs: man:chronyd(8)
           man:chrony.conf(5)
  Process: 7573 ExecStartPost=/usr/libexec/chrony-helper update-daemon (code=exited,
status=0/SUCCESS)
  Process: 7569 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 7571 (chronyd)
    Tasks: 1 (limit: 48692)
   Memory: 1.2M
   CGroup: /system.slice/chronyd.service
           └─7571 /usr/sbin/chronyd
```

4.2 Let's verify and update the chrony sources by using below command

```
# chronyc sources -v
```

Output:



```
[root@eoc-controller ~]# chronyc sources -v

  .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
 / .- Source state '*' = current best, '+' = combined, '-' = not combined,
| /             'x' = may be in error, '~' = too variable, '?' = unusable.
||                                                 .- xxxx [ yyyy ] +/- zzzz
||      Reachability register (octal) -.           |  xxxx = adjusted offset,
||      Log2(Polling interval) --.      |          |  yyyy = measured offset,
||                                \     |          |  zzzz = estimated error.
||                                 |    |           \
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^+ ntp6.mum-in.hosts.301-mo>     2   6    37     9    -43us[-1698us] +/-   72ms
^- 152.70.69.232                 5   6   113     0  +3306us[+3306us] +/-   17ms
^* ntp5.mum-in.hosts.301-mo>     2   6    37     9    +93us[-1581us] +/-   71ms
^? 194-195-112-240.ip.linod>     2   7   100    80  +4341us[+19797s] +/-   81ms
```

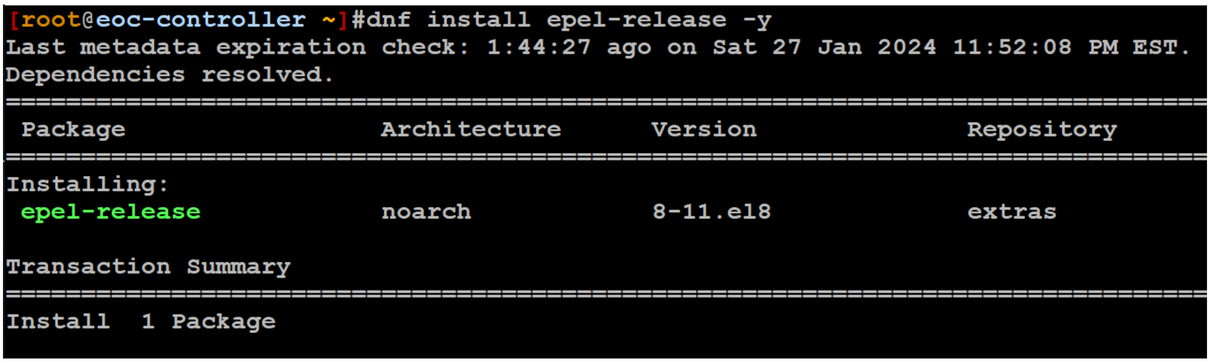
5. Installing and Configuring EPEL Repository

5.1 Install and Configure EPEL repository:

EPEL (**Extra Packages for Enterprise Linux**) is an open-source and free community-based repository project from Fedora team which provides 100% high-quality add-on software packages for Linux distribution including RHEL (Red Hat Enterprise Linux), CentOS.

```
# dnf install epel-release -y
```

Output:



```
[root@eoc-controller ~]#dnf install epel-release -y
Last metadata expiration check: 1:44:27 ago on Sat 27 Jan 2024 11:52:08 PM EST.
Dependencies resolved.
================================================================================
 Package                Architecture       Version              Repository
================================================================================
Installing:
 epel-release           noarch             8-11.el8             extras

Transaction Summary
================================================================================
Install  1 Package
```

6. **Create a user account "admin" which will be used to administer managed hosts via ansible controller.**

6.1 Run the below commands on all of the servers including all worker nodes.

- **eoc-controller**
- **eoc-node1**
- **eoc-node2**
- **eoc-node3**

Note: Directly execute the commands on all of the servers.

```
# useradd -m -G wheel admin
```

```
# echo "linux" | passwd --stdin admin
```

```
# sed -e '/%wheel/ s/^#*/#/' -i /etc/sudoers
```

```
# cat >> /etc/sudoers <<EOF
%wheel        ALL=(ALL)       NOPASSWD: ALL
EOF
```