Red Hat Enterprise Linux 10

Configuring time synchronization

Configuring time synchronization to maintain accurate timekeeping across network devices

Last Updated: 2026-01-15

Red Hat Enterprise Linux 10 Configuring time synchronization

Configuring time synchronization to maintain accurate timekeeping across network devices

Legal Notice

Abstract

You can configure time synchronization to maintain accurate time across a network, ensure reliable communication and system operations.

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

Submitting feedback through Jira (account required)

1. Log in to the Jira website.

2. Click **Create** in the top navigation bar.

3. Enter a descriptive title in the **Summary** field.

4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.

5. Click **Create** at the bottom of the dialogue.

CHAPTER 1. INTRODUCTION TO CHRONY SUITE

The implementation of the **Network Time Protocol (NTP)** is **chrony**. You can use **chrony**:

- To synchronize the system clock with **NTP** servers
- To synchronize the system clock with a reference clock, for example a GPS receiver
- To synchronize the system clock with a manual time input
- As an **NTPv4(RFC 5905)** server or peer to provide a time service to other computers in the network

chrony performs well in a wide range of conditions:

- Including network connections
- Heavily congested networks
- Changing temperatures (ordinary computer clocks are sensitive to temperature)
- Systems that do not run continuously, or run on a virtual machine.

chrony consists of **chronyd**, a daemon that runs in user space, and **chronyc**, a command line program which can be used to monitor the performance of **chronyd** and to change various operating parameters when it is running.

The command line utility **chronyc** can monitor and control the **chronyd** daemon. This utility provides a command prompt which lets you enter a number of commands to query the current state of **chronyd** and make changes to its configuration. By default, **chronyd** accepts only commands from a local instance of **chronyc**, but it can be configured to accept monitoring commands also from remote hosts. The remote access should be restricted.

CHAPTER 2. USING CHRONY

Learn how to start and stop **chronyd**, check it is synchronized, and manually adjust the system clock.

2.1. MANAGING CHRONY

The **chrony** suite is installed by default on Red Hat Enterprise Linux. You can start, stop, and check the status of **chronyd**.

Procedure

1. Optional: Check if the **chrony** suite is installed by running the following command as **root**:

   ```
   # dnf install chrony
   ```

   The default location for the **chrony** daemon is **/usr/sbin/chronyd**. The command line utility will be installed to **/usr/bin/chronyc**.

2. To check the status of **chronyd**, issue the following command:

   ```
   $ systemctl status chronyd
   chronyd.service - NTP client/server
      Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled)
      Active: active (running) since Wed 2013-06-12 22:23:16 CEST; 11h ago
   ```

3. To start **chronyd**, issue the following command as **root**:

   ```
   # systemctl start chronyd
   ```

   To ensure **chronyd** starts automatically at system start, issue the following command as **root**:

   ```
   # systemctl enable chronyd
   ```

4. To stop **chronyd**, issue the following command as **root**:

   ```
   # systemctl stop chronyd
   ```

   To prevent **chronyd** from starting automatically at system start, issue the following command as **root**:

   ```
   # systemctl disable chronyd
   ```

2.2. MANUALLY ADJUSTING THE SYSTEM CLOCK

You can manually adjust the system clock.

Procedure

- To step the system clock immediately, by passing any adjustments in progress by slewing, enter:

  ```
  # chronyc makestep
  ```

–

IMPORTANT

If the **rtcfile** directive is used, the real-time clock should not be manually adjusted. Random adjustments would interfere with **chrony**'s need to measure the rate at which the real-time clock drifts.

2.3. DISABLING A NETWORKMANAGER DISPATCHER SCRIPT

The **chrony** dispatcher script manages the online and offline state of the NTP servers. As a system administrator, you can disable the dispatcher script to keep **chronyd** polling the servers constantly.

The NetworkManager executes the **chrony** dispatcher script during interface reconfiguration, stop or start operations. However, if you configure certain interfaces or routes outside of NetworkManager, you can encounter the following situation:

- The dispatcher script might run when no route to the NTP servers exists, causing the NTP servers to switch to the offline state.
- If you establish the route later, the script does not run again by default, and the NTP servers remain in the offline state.

To ensure that **chronyd** can synchronize with your NTP servers, which have separately managed interfaces, disable the dispatcher script.

Procedure

- To disable the **chrony** dispatcher script, create a symlink to **/dev/null**:

```
# ln -f -s /dev/null /etc/NetworkManager/dispatcher.d/20-chrony-onoffline
```

NOTE

After this change, the NTP servers remain in the online state at all times.

2.4. SETTING UP CHRONY IN AN ISOLATED NETWORK

For a network that is never connected to the internet, one computer is selected to be the primary timeserver. The other computers are either direct clients of the server, or clients of clients. On the server, the drift file must be manually set with the average rate of drift of the system clock. If the server is rebooted, it will obtain the time from surrounding systems and calculate an average to set its system clock. Thereafter it resumes applying adjustments based on the drift file. The drift file will be updated automatically when the **settime** command is used.

To set up **chrony** for a system in an isolated network, follow the steps mentioned below:

Procedure

1. On the system selected to be the server, edit **/etc/chrony.conf** as follows:

```
driftfile /var/lib/chrony/drift
commandkey 1
keyfile /etc/chrony.keys
```

```
initstepslew 10 client1 client3 client6
local stratum 8
manual
allow <subnet>
```

Where **<subnet>** is the network from which the clients are allowed to connect. Use Classless Inter-Domain Routing (CIDR) notation to specify the subnet.

2. On the systems selected to be direct clients of the server, edit the **/etc/chrony.conf** as follows:

```
server <server_fqdn>
driftfile /var/lib/chrony/drift
logdir /var/log/chrony
log measurements statistics tracking
keyfile /etc/chrony.keys
commandkey 24
local stratum 10
initstepslew 20 ntp1.example.net
allow <server_ip_address>
```

Where **<server_fqdn>** is the host name of the server, and **<server_ip_address>** is the address of the server . Clients with this configuration will resynchronize with the server if it restarts.

On the client systems which are not to be direct clients of the server, the **/etc/chrony.conf** file should be the same except that the **local** and **allow** directives should be omitted.

In an isolated network, you can also use the **local** directive that enables a local reference mode, which allows **chronyd** operating as an **NTP** server to appear synchronized to real time, even when it was never synchronized or the last update of the clock happened a long time ago.

To allow multiple servers in the network to use the same local configuration and to be synchronized to one another without confusing clients that poll more than one server, use the **orphan** option of the **local** directive which enables the orphan mode. Each server needs to be configured to poll all other servers with **local**. This ensures that only the server with the smallest reference ID has the local reference active and other servers are synchronized to it. When the server fails, another one takes over.

2.5. CONFIGURING REMOTE MONITORING ACCESS

The **chronyc** utility can access **chronyd** by using the following methods:

- IPv4 or IPv6.
- A domain socket, which is accessible locally by the **root** and **chrony** user.

By default, **chronyc** connects to the Unix domain socket. The default path is **/var/run/chrony/chronyd.sock**. If this connection fails, **chronyc** tries to connect to 127.0.0.1 and then ::1.

Only the following monitoring commands, which do not affect the behavior of **chronyd**, are allowed from the network:

- activity
- manual list

- rtcdata
- smoothing
- sources
- sourcestats
- tracking
- waitsync

By default, the commands are accepted only from localhost (127.0.0.1 or ::1).

All other commands are allowed only through the Unix domain socket. When sent over the network, **chronyd** responds with a **Not authorised** error, even if it is from localhost.

The following procedure describes how to access chronyd remotely with **chronyc**.

Procedure

1. Configure **chrony** to listen on local interface by adding the following to the **/etc/chrony.conf** file:

   ```
   bindcmdaddress 0.0.0.0
   ```

   and

   ```
   bindcmdaddress ::
   ```

2. Add the following content to the **/etc/chrony.conf** file to allow commands from remote IP addresses, networks, and subnet:

   ```
   cmdallow 192.168.1.0/24
   ```

   ```
   cmdallow 2001:db8::/64
   ```

3. Open port 323 in the firewall to allow connections from remote systems:

   ```
   # firewall-cmd --permanent --add-port=323/udp
   ```

4. Reload the firewall configuration:

   ```
   # firewall-cmd --reload
   ```

   For more information, see the **chrony.conf(5)** man page on your system.

2.6. CHECKING IF CHRONY IS SYNCHRONIZED

You can check if **chrony** is synchronized with the use of the **tracking**, **sources**, and **sourcestats** commands.

Procedure

1. To check **chrony** tracking, enter:

   ```
   $  chronyc tracking
   Reference ID    : CB00710F (ntp-server.example.net)
   Stratum         : 3
   Ref time (UTC)  : Fri Jan 27 09:49:17 2017
   System time     :  0.000006523 seconds slow of NTP time
   Last offset     : -0.000006747 seconds
   RMS offset      : 0.000035822 seconds
   Frequency       : 3.225 ppm slow
   Residual freq   : 0.000 ppm
   Skew            : 0.129 ppm
   Root delay      : 0.013639022 seconds
   Root dispersion : 0.001100737 seconds
   Update interval : 64.2 seconds
   Leap status     : Normal
   ```

2. The **chronyc** sources command displays information about the current time sources that **chronyd** is accessing.

   ```
   $ chronyc sources
    210 Number of sources = 3
   MS Name/IP address         Stratum Poll Reach LastRx Last sample
   ===============================================================================
   ====
   #* GPS0                          0   4   377    11   -479ns[ -621ns] /-  134ns
   ^? a.b.c                         2   6   377    23   -923us[ -924us] +/-   43ms
   ^ d.e.f                          1   6   377    21  -2629us[-2619us] +/-   86ms
   ```

   You can specify the optional **-v** argument to print more verbose information. In this case, extra caption lines are shown as a reminder of the meanings of the columns.

3. The **sourcestats** command displays information about the drift rate and offset estimation process for each of the sources currently being examined by **chronyd**. To check **chrony** source statistics, enter the following command:

   ```
   $  chronyc sourcestats
   210 Number of sources = 1
   Name/IP Address            NP  NR  Span  Frequency  Freq Skew  Offset  Std Dev
   ===============================================================================
   ====
   abc.def.ghi                11   5   46m     -0.001      0.045      1us    25us
   ```

   The optional argument **-v** can be specified, meaning verbose. In this case, extra caption lines are shown as a reminder of the meanings of the columns. For more information, see the **chronyc(1)** man page on your system.

2.7. MONITORING NTP CLIENTS

Display the list of clients that accessed the **chronyd** Network Time Protocol (NTP) server. This information is useful for verifying which clients are actively synchronizing and checking their connection history.

Prerequisites

- The **chronyd** service is running.
- **chronyd** is configured to allow NTP clients. For example, add the **allow** directive to **/etc/chrony.conf**.

Procedure

1. Display a list of clients by using the **chronyc clients** command:

```
# chronyc clients
```

2. Review the command output:

```
Hostname                      NTP  Drop  Int  IntL  Last  Cmd  Drop  Int  Last
=========================================================================
=========
client1.example.net      10    0    6   -   10s  0    0    6   12s
client2.example.net       8    0    6   -   25s  0    0    6   28s
desktop-192-0-2-100.example.net  3        6   -    9s  0    0    6   11s
```

The output fields include:

- **Hostname**: The hostname or IP address of the client.
- **NTP**: The number of NTP packets received from the client.
- **Drop**: The number of packets dropped by **chronyd**, typically due to rate-limiting.
- **Int**: The client's polling interval in seconds (as a log base 2 value, e.g., 6 = $2^6$ = 64s).
- **IntL**: The average inter-poll interval (in seconds) for the client when rate limiting is active. This field shows **'-'** if rate limiting is not currently active for the client.
- **Last**: The elapsed time since the last NTP packet was received from this client. Use this value to determine which clients are synchronized or recently synchronized.
- **Cmd**: The number of **chronyc** (monitoring) command packets received from the client.

NOTE

If you have a number of clients and the command is slow to return, it might be due to performing DNS lookups. Use the **-n** option to disable hostname resolution and display IP addresses instead: **# chronyc -n clients**

IMPORTANT

By default, **chronyd** monitors up to 4,096 clients. If you have more clients than this, you can increase the memory limit with **clientloglimit** in **/etc/chrony.conf** file. For example: **clientloglimit 10000000**.

For more information, see the **chronyc(1)** and **chronyd(8)** man pages on your system.

2.8. ADDITIONAL RESOURCES

- Frequently Asked Questions

CHAPTER 3. CHRONY WITH HARDWARE TIMESTAMPING

Hardware (HW) timestamping in some Network Interface Controller (NICs) provides accurate timestamping of incoming and outgoing packets. **NTP** timestamps are usually created by the kernel and **chronyd** with the use of the system clock. However, when HW timestamping is enabled, the NIC uses its own clock to generate the timestamps when packets are entering or leaving the link layer or the physical layer. When used with **NTP**, hardware timestamping can significantly improve the accuracy of synchronization. For best accuracy, both **NTP** servers and **NTP** clients need to use hardware timestamping. Under ideal conditions, a sub-microsecond accuracy might be possible.

Another protocol for time synchronization that uses hardware timestamping is **PTP**.

Unlike **NTP**, **PTP** relies on assistance in network switches and routers. If you want to achieve the best accuracy of synchronization, use **PTP** on networks that have switches and routers with **PTP** support, and prefer **NTP** on networks that do not have such switches and routers.

3.1. VERIFYING SUPPORT FOR HARDWARE TIMESTAMPING

To verify that hardware timestamping with **NTP** is supported by an interface, use the **ethtool -T** command. An interface can be used for hardware timestamping with **NTP** if **ethtool** lists the **SOF_TIMESTAMPING_TX_HARDWARE** and **SOF_TIMESTAMPING_TX_SOFTWARE** capabilities and also the **HWTSTAMP_FILTER_ALL** filter mode.

Procedure

- Display a device’s time stamping capabilities and associated PTP hardware clock:

  ```
  # ethtool -T enp1s0
  ```

3.2. ENABLING HARDWARE TIMESTAMPING

You can enable the hardware timestamping on one or multiple interfaces by using the **hwtimestamp** directive in the **/etc/chrony.conf** file. The directive can either specify a single interface, or a wildcard character can be used to enable hardware timestamping on all interfaces that support it.

Procedure

1. Edit the **/etc/chrony.conf** file and make the following changes:

   a. Add the **hwtimestamp** setting for interfaces which support hardware timestamping. For example:

      ```
      hwtimestamp enp1s0
      hwtimestamp eno*
      ```

      You can use the * wildcard if no other application, such as **ptp4l** uses hardware timestamping.

   b. Configure a short client polling interval by appending the **minpoll** and **maxpoll** options to the server setting, for example:

      ```
      server ntp.example.comlocal minpoll 0 maxpoll 0
      ```

For hardware timestamping, you must configure a shorter polling interval than the default range (64-1024 seconds) to minimize the offset of the system clock.

c. Enable the NTP interleaved mode by appending the **xleave** option to the server setting:

```
server ntp.example.comlocal minpoll 0 maxpoll 0 xleave
```

With this setting, chrony gets the hardware transmit timestamp only after sending a packet. This behavior prevents the serever from saving the timestamp in packets to which it responds. With the **xleave** option, chrony can receive transmit timestamps that were generated after the transmission.

d. Optional: Increase the maximum size of memory allocated for logging of client's access on the server, for example:

```
clientloglimit 100000000
```

The default server configuration allows a few thousands of clients to use the interleaved mode concurrently. By increasing the value of the **clientloglimit** setting, you can configure the server for a large number of clients.

2. Restart the chronyd service:

```
# systemctl restart chronyd
```

Verification

1. Optional: Verify in the **/var/log/messages** log file that hardware timesamping is enabled:

```
chronyd[4081]: Enabled HW timestamping on enp1s0
chronyd[4081]: Enabled HW timestamping on eno1
```

2. If chronyd is configured as an NTP client or peer, display the transmit and receive timestamping modes and the interleaved mode:

```
# chronyc ntpdata

Remote address  : 203.0.113.15 (CB00710F)
Remote port     : 123
Local address   : 203.0.113.74 (CB00714A)
Leap status     : Normal
Version         : 4
Mode            : Server
Stratum         : 1
Poll interval   : 0 (1 seconds)
Precision       : -24 (0.000000060 seconds)
Root delay      : 0.000015 seconds
Root dispersion : 0.000015 seconds
Reference ID    : 47505300 (GPS)
Reference time  : Wed May 03 13:47:45 2017
Offset          : -0.000000134 seconds
Peer delay      : 0.000005396 seconds
Peer dispersion : 0.000002329 seconds
Response time   : 0.000152073 seconds
```

```
Jitter asymmetry: +0.00
NTP tests       : 111 111 1111
Interleaved     : Yes
Authenticated   : No
TX timestamping : Hardware
RX timestamping : Hardware
Total TX        : 27
Total RX        : 27
Total valid RX  : 27
```

3. Report the stability of NTP measurements:

```
# chronyc sourcestats

210 Number of sources = 1
Name/IP Address            NP  NR  Span  Frequency  Freq Skew  Offset  Std Dev
ntp.local                  12   7    11     +0.000      0.019     +0ns    49ns
```

   This stability is reported in the **Std Dev** column. With hardware timestamping enabled, stability of NTP measurements should be in tens or hundreds of nanoseconds, under normal load.

3.3. CONFIGURING PTP-NTP BRIDGE

If a highly accurate Precision Time Protocol (**PTP**) primary timeserver is available in a network that does not have switches or routers with **PTP** support, a computer may be dedicated to operate as a **PTP** client and a stratum-1 **NTP** server. Such a computer needs to have two or more network interfaces, and be close to the primary timeserver or have a direct connection to it. This will ensure highly accurate synchronization in the network.

Procedure

1. Configure the **ptp4l** and **phc2sys** programs from the **linuxptp** packages to use one interface to synchronize the system clock by using **PTP**.

2. Configure **chronyd** to provide the system time by using the other interface:

```
bindaddress 203.0.113.74
hwtimestamp enp1s0
local stratum 1
```

3. Restart the **chronyd** service:

```
# systemctl restart chronyd
```

CHAPTER 4. OVERVIEW OF NETWORK TIME SECURITY (NTS) IN CHRONY

Network Time Security (NTS) is an authentication mechanism for Network Time Protocol (NTP), designed to scale substantial clients. It verifies that the packets received from the server machines are unaltered while moving to the client machine. Network Time Security (NTS) includes a Key Establishment (NTS-KE) protocol that automatically creates the encryption keys used between the server and its clients.

> **WARNING**
>
> NTS is not compatible with the FIPS and OSPP profile. When you enable the FIPS and OSPP profile, **chronyd** that is configured with NTS can abort with a fatal message. You can disable the OSPP profile and FIPS mode for **chronyd** service by adding the **GNUTLS_FORCE_FIPS_MODE=0** setting to the **/etc/sysconfig/chronyd** file.

4.1. ENABLING NETWORK TIME SECURITY (NTS) ON A CLIENT

By default, Network Time Security (NTS) is not enabled. You can enable NTS in the **/etc/chrony.conf** file.

Prerequisites

- The time server supports NTS.

Procedure

1. Edit the **/etc/crony.conf** file to make the following changes:
2. Specify the server with the **nts** option in addition to the **iburst** option.

   ```
   For example:
   server time.example.com iburst nts
   server nts.netnod.se iburst nts
   server ptbtime1.ptb.de iburst nts
   ```

3. Add the following setting to avoid repeating the Network Time Security-Key Establishment (NTS-KE) session during system boot:

   ```
   ntsdumpdir /var/lib/chrony
   ```

4. If present, comment out or remove the following setting to disable synchronization with Network Time Protocol (NTP) servers provided by **DHCP**:

   ```
   sourcedir /run/chrony-dhcp
   ```

5. Restart the **chronyd** service:

```
systemctl restart chronyd
```

Verification

- Verify if the **NTS** keys were successfully established:

  ```
  # chronyc -N authdata

  Name/IP address  Mode KeyID Type KLen Last Atmp  NAK Cook CLen
  =========================================================================
  time.example.com  NTS     1   15  256  33m    0    0    8  100
  nts.netnod.se   NTS     1   15  256  33m    0    0    8  100
  ptbtime1.ptb.de   NTS     1   15  256  33m    0    0    8  100
  ```

  The **KeyID**, **Type**, and **KLen** should have non-zero values. If the value is zero, check the system log for error messages from **chronyd**.

- Verify the client is making NTP measurements:

  ```
  # chronyc -N sources

  MS Name/IP address Stratum Poll Reach LastRx Last sample
  =================================================================
  time.example.com   3        6   377    45   +355us[ +375us] +/-   11ms
  nts.netnod.se   1        6   377    44   +237us[ +237us] +/-   23ms
  ptbtime1.ptb.de    1        6   377    44   -170us[ -170us] +/-   22ms
  ```

  The **Reach** column should have a non-zero value; ideally 377. If the value rarely gets 377 or never gets to 377, it indicates that NTP requests or responses are getting lost in the network.

  For more details, see the **chrony.conf(5)** man page on your system.

4.2. ENABLING NETWORK TIME SECURITY (NTS) ON A TIME SERVER

If you run your own Network Time Protocol (NTP) server, you can enable the server Network Time Security (NTS) support to facilitate its clients to synchronize securely.

If the NTP server is a client of other servers, that is, it is not a Stratum 1 server, it should use NTS or symmetric key for its synchronization.

Prerequisites

- Server private key in **PEM** format
- Server certificate with required intermediate certificates in **PEM** format

Procedure

1. Edit the **/etc/chrony.conf** file, and make the following changes:

   ```
   ntsserverkey /etc/pki/tls/private/<ntp-server.example.net>.key
   ntsservercert /etc/pki/tls/certs/<ntp-server.example.net>.crt
   ```

2. Set permissions on both the private key and the certificate file that allow the chrony user to read the files, for example

```
# chown root:chrony /etc/pki/tls/private/<ntp-server.example.net>.key
/etc/pki/tls/certs/<ntp-server.example.net>.crt

# chmod 644 /etc/pki/tls/private/<ntp-server.example.net>.key /etc/pki/tls/certs/<ntp-
server.example.net>.crt
```

3. Ensure that the **ntsdumpdir /var/lib/chrony** setting is present.

4. Open the required ports in firewalld:

```
# firewall-cmd --permanent --add-port={323/udp,4460/tcp}
# firewall-cmd --reload
```

5. Restart the **chronyd** service:

```
# systemctl restart chronyd
```

Verification

1. Perform a test from a client machine:

```
$ chronyd -Q -t 3 'server

ntp-server.example.net iburst nts maxsamples 1'
2021-09-15T13:45:26Z chronyd version 4.1 starting (+CMDMON +NTP +REFCLOCK +RTC
+PRIVDROP +SCFILTER +SIGND +ASYNCDNS +NTS +SECHASH +IPV6 +DEBUG)
2021-09-15T13:45:26Z Disabled control of system clock
2021-09-15T13:45:28Z System clock wrong by 0.002205 seconds (ignored)
2021-09-15T13:45:28Z chronyd exiting
```

The **System clock wrong** message indicates the NTP server is accepting NTS-KE connections and responding with NTS-protected NTP messages.

2. Verify the NTS-KE connections and authenticated NTP packets observed on the server:

```
# chronyc serverstats

NTP packets received       : 7
NTP packets dropped        : 0
Command packets received   : 22
Command packets dropped    : 0
Client log records dropped : 0
NTS-KE connections accepted: 1
NTS-KE connections dropped : 0
Authenticated NTP packets: 7
```

If the value of the **NTS-KE connections accepted** and **Authenticated NTP packets** field is a non-zero value, it means that at least one client was able to connect to the NTS-KE port and send an authenticated NTP request.