Oracle Linux 10

Configuring a Network Time Service With Chrony

G24144-01
June 2025

Oracle Linux 10 Configuring a Network Time Service With Chrony,

G24144-01

Contents

Preface

1 About the chrony Suite

2 About PTP

Preface

Oracle Linux 10: Configuring a Network Time Service With Chrony provides information about configuring a network time service with `chrony` on Oracle Linux 10 systems.

Documentation License

The content in this document is licensed under the Creative Commons Attribution–Share Alike 4.0 (CC-BY-SA) license. In accordance with CC-BY-SA, if you distribute this content or an adaptation of it, you must provide attribution to Oracle and retain the original copyright notices.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at https://www.oracle.com/corporate/accessibility/.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

About the chrony Suite

chrony is a feature that implements NTP to maintain timekeeping accurately on the network. In Oracle Linux 8 and later, the chrony daemon service replaces ntpd for the management of NTP.

chrony has two components, which are provided in the chrony package:

- chronyd service daemon
- chronyc service utility

For practical exercises in using chrony, see Configure Chrony on Oracle Linux.

About the chronyd Service Daemon

The chronyd service daemon updates the system clock of mobile systems and virtual machines after a period of suspension or disconnection from a network. The service can also be used to implement a basic NTP client or NTP server. As an NTP server, chronyd can synchronize with upper level stratum NTP servers or act as a stratum 1 server using time signals that are received from the Global Positioning System (GPS) or radio broadcasts such as DCF77, MSF, or WWVB.

In an Oracle Linux 10 system, this service daemon is enabled by default.

> **Note:**
>
> chronyd uses NTP version 3 (RFC 1305), with features that are compatible with NTP version 4 (RFC 5905). However, chronyd does not support several important features of NTP version 4, nor does it support the use of PTP.

For more information, see the chronyd(8) manual page and files in the /usr/share/doc/chrony/ directory.

About the chronyc Service Utility

The chronyc utility is a tool for managing the chronyd service, displaying information about the service's operation, or changing the service's configuration.

The command operates in two modes:

- Non interactive mode: In this mode, you use the following syntax:

  ```
  sudo chronyc subcommand
  ```

- Interactive mode: Typing the command by itself activates the interactive mode and displays the `chronyc>` prompt. From this prompt you can issue `chronyc` subcommands.

  ```
  sudo chronyc
  ```

  ```
  chronyc>
  ```

  From the prompt, you can issue the different `chronyc` subcommands as needed. The following examples show the information that's generated by the `sources`, `sourcestats`, and `tracking` subcommands:

  ```
  chronyc> sources
  ```

  ```
  210 Number of sources = 4
  MS Name/IP address         Stratum Poll Reach LastRx Last sample
  ===========================================================================
  ====
  ^+ service1-eth3.debrecen.hp     2   6    37    21  -2117us[-2302us] +/-
  50ms
  ^* ns2.telecom.lt                2   6    37    21   -811us[ -997us] +/-
  40ms
  ^+ strato-ssd.vpn0.de            2   6    37    21   +408us[ +223us] +/-
  78ms
  ^+ kvm1.websters-computers.c     2   6    37    22  +2139us[+1956us] +/-
  54ms
  ```

  ```
  chronyc> sourcestats
  ```

  ```
  210 Number of sources = 4
  Name/IP Address            NP  NR  Span  Frequency  Freq Skew  Offset  Std
  Dev
  ===========================================================================
  ===
  service1-eth3.debrecen.hp   5   4   259     -0.394     41.803  -2706us
  502us
  ns2.telecom.lt              5   4   260     -3.948     61.422   +822us
  813us
  strato-ssd.vpn0.de          5   3   259      1.609     68.932   -581us
  801us
  kvm1.websters-computers.c   5   5   258     -0.263      9.586  +2008us
  118us
  ```

  ```
  chronyc> tracking
  ```

  ```
  Reference ID    : 212.59.0.2 (ns2.telecom.lt)
  Stratum         : 3
  Ref time (UTC)  : Tue Sep 30 12:33:16 2014
  System time     : 0.000354079 seconds slow of NTP time
  ```

```
Last offset     : -0.000186183 seconds
RMS offset      : 0.000186183 seconds
Frequency       : 28.734 ppm slow
Residual freq   : -0.489 ppm
Skew            : 11.013 ppm
Root delay      : 0.065965 seconds
Root dispersion : 0.007010 seconds
Update interval : 64.4 seconds
Leap status     : Normal
```

To quit using the interactive mode, type `exit`.

> **Note:**
>
> Any changes you implement with the `chronyc` command are effective only until the next restart of the `chronyd` daemon. To make the changes permanent, you must enter these in the `/etc/chrony.conf` file. See chronyd Configuration File Examples.

For more information, see the `chronyc(1)` manual page and files in the `/usr/share/doc/chrony/` directory.

Configuring the chronyd Service

To configure the `chronyd` service on a system:

1. Optional: If the `chrony` package isn't installed, run the following command.

   ```
   sudo dnf install chrony
   ```

2. If remote access to the local NTP service is required, configure the system firewall to allow access to the NTP service in the appropriate zones.

   For example:

   ```
   sudo firewall-cmd --zone=zone --add-service=ntp
   ```

   ```
   sudo firewall-cmd --zone=zone --permanent --add-service=ntp
   ```

3. Start the `chronyd` service and configure it to start following a system reboot.

   > **Note:**
   >
   > By default, `chrony` is enabled after installation.

   ```
   sudo systemctl start chronyd
   ```

   ```
   sudo systemctl enable chronyd
   ```

chronyd Configuration File Examples

The `/etc/chrony.conf` file contains configuration settings for `chronyd`. The default configuration assumes that the system has network access to public NTP servers with which it can synchronize. The following examples show modifications you can make in different networking scenarios.

For more information about the configuration file and its directives, see the `chrony.conf(5)` manual page.

The following example configures a system to access three NTP servers:

```
pool NTP_server_1
pool NTP_server_2
pool NTP_server_3
driftfile /var/lib/chrony/drift
keyfile /etc/chrony.keys
...
```

To configure `chronyd` to act as an NTP server for a specified client or subnet, use the `allow` directive, as shown in bold in the following example:

```
pool NTP_server_1
pool NTP_server_2
pool NTP_server_3
allow 192.168.2/24
driftfile /var/lib/chrony/drift
keyfile /etc/chrony.keys
...
```

To create keys for an authentication mechanism based on public key cryptography, use the `chronyc keygen` command.

> **Note:**
>
> `Autokey` in `ntp` no longer works in `chrony`.

If a system has only intermittent access to NTP servers, the following configuration might be appropriate:

```
pool NTP_server_1 offline
pool NTP_server_2 offline
pool NTP_server_3 offline
driftfile /var/lib/chrony/drift
keyfile /etc/chrony.keys
...
```

If you specify the `offline` keyword, `chronyd` doesn't poll the NTP servers until it receives communication that network access is available. You can use the `chronyc online` and `chronyc offline` commands to inform `chronyd` of the state of network access.

Replacement of ntp with chrony

In Oracle Linux 8 and later, `chrony` replaces `ntp` for the management of NTP. The following table shows file, command, and terminology equivalents between `ntp` and `chrony`.

ntp	chrony
`/etc/ntp.conf`	`/etc/chrony.conf`
`/etc/ntp/keys`	`/etc/chrony.keys`
`ntpd`	`chronyd`
`ntpq` command	`chronyc` command
`ntpd.service`	`chronyd.service`
`ntp-wait.service`	`chrony-wait.service`
`ntpdate` and `sntp` utilities	`chronyd -q` and `chronyd -t` commands

The `ntpstat` utility which is available in the `ntpstat` package, now supports `chronyd`. Thus, you can still use the utility in Oracle Linux 8 and later. The command generates output that's similar to when it's used with `ntp`.

2
About PTP

Use PTP to synchronize system clocks on a LAN more accurately than NTP. If network drivers support either hardware or software time stamping, a PTP clock can use the time stamps in PTP messages to resolve propagation delays across a network. With software time stamping, PTP synchronizes systems to within a few tens of microseconds. With hardware time stamping, PTP can synchronize systems to within a few tenths of a microsecond. If you require high-precision time synchronization of systems, use hardware time stamping.

A typical PTP configuration on an enterprise local area network consists of:

- One or more **grandmaster clock** systems.

  A grandmaster clock is typically implemented as specialized hardware that can use high-accuracy GPS signals or lower-accuracy code division several access (CDMA) signals, radio clock signals, or NTP as a time reference source. If several grandmaster clocks are available, the best master clock (BMC) algorithm selects the grandmaster clock based on the settings of their `priority1`, `clockClass`, `clockAccuracy`, `offsetScaledLogVariance`, and `priority2` parameters and their unique identifier, in that order.

- Several **boundary clock** systems.

  Each boundary clock is backed up to a grandmaster clock on one subnetwork and relays PTP messages to one or more added subnetworks. A boundary clock is usually implemented as a function of a network switch.

- Several **secondary clock** systems.

  Each secondary clock on a subnetwork is backed up to a boundary clock, which acts as the **master clock** for that secondary clock.

For a basic configuration, set up a single grandmaster clock and several secondary clocks on the same network segment and thus eliminates any need for an intermediate layer of boundary clocks.

Grandmaster and secondary clock systems that use only one network interface for PTP are termed **ordinary clocks**.

Boundary clocks require at least two network interfaces for PTP: one interface acts a secondary to a grandmaster clock or a higher-level boundary clock; the other interfaces act as masters to secondary clocks or lower-level boundary clocks.

Synchronization of boundary and secondary clock systems is achieved by sending time stamps in PTP messages. By default, PTP messages are sent in UDPv4 datagrams. You can also configure PTP to use UDPv6 datagrams or Ethernet frames as its transport mechanism.

To use PTP on a system, the driver for at least one of the system's network interfaces must support either software or hardware time stamping. To find out whether the driver for a network interface supports time stamping, use the `ethtool` command:

```
sudo ethtool -T eno1
```

```
Time stamping parameters for eno1:
Capabilities:
    hardware-transmit     (SOF_TIMESTAMPING_TX_HARDWARE)
    software-transmit     (SOF_TIMESTAMPING_TX_SOFTWARE)
    hardware-receive      (SOF_TIMESTAMPING_RX_HARDWARE)
    software-receive      (SOF_TIMESTAMPING_RX_SOFTWARE)
    software-system-clock (SOF_TIMESTAMPING_SOFTWARE)
    hardware-raw-clock    (SOF_TIMESTAMPING_RAW_HARDWARE)
...
```

The output in the example shows that the `eno1` interface supports both hardware and software time stamping capabilities.

With software time stamping, `ptp4l` synchronizes the system clock to an external grandmaster clock.

If hardware time stamping is available, `ptp4l` can synchronize the PTP hardware clock to an external grandmaster clock. In this case, you use the `phc2sys` daemon to synchronize the system clock with the PTP hardware clock.

Configuring the PTP Service

To configure the PTP service on a system:

1. Install the `linuxptp` package.

   ```
   sudo dnf install linuxptp
   ```

2. Edit `/etc/sysconfig/ptp4l` and define the start-up options for the `ptp4l` daemon.

   Grandmaster clocks and secondary clocks require that you define only one interface.

   For example, to use hardware time stamping with interface `eno1` on a secondary clock:

   ```
   OPTIONS="-f /etc/ptp4l.conf -i eno1 -s"
   ```

   To use software time stamping instead of hardware time stamping, specify the `-S` option:

   ```
   OPTIONS="-f /etc/ptp4l.conf -i eno1 -S -s"
   ```

   > **Note:**
   >
   > The `-s` option specifies that the clock operates only as a secondary (`clientOnly` mode). Don't specify this option for a grandmaster clock or a boundary clock.

   For a grandmaster clock, omit the `-s` option, for example:

   ```
   OPTIONS="-f /etc/ptp4l.conf -i eno1"
   ```

   A boundary clock requires that you define at least two interfaces, for example:

   ```
   OPTIONS="-f /etc/ptp4l.conf -i eno1 -i eno2"
   ```

You might need to edit the /etc/ptp4l.conf file to customize ptp4l further, for example:

- For a grandmaster clock, set the value of the priority1 parameter to a value between 0 and 127, where lesser values have greater priority when the BMC algorithm selects the grandmaster clock. For a configuration that has a single grandmaster clock, a value of 127 is suggested.
- If you set the value of summary_interval to an integer value *N* instead of 0, ptp4l writes summary clock statistics to /var/log/messages every 2^N seconds instead of every second ($2^0 = 1$). For example, a value of 10 would correspond to an interval of 2^{10} or 1024 seconds.
- The logging_level parameter controls the amount of logging information that ptp4l records. The default value of logging_level is 6, which corresponds to LOG_INFO. To turn off logging, set the value of logging_level to 0. Alternatively, specify the -q option to ptp4l.

See the ptp4l(8) manual page.

3. Configure the system firewall to accept access by PTP event and general messages to UDP ports 319 and 320 in the appropriate zone.

   For example:

   ```
   sudo firewall-cmd --zone=zone --add-port=319/udp --add-port=320/udp
   ```

   ```
   sudo firewall-cmd --permanent --zone=zone --add-port=319/udp --add-port=320/udp
   ```

4. Start the ptp4l service and configure it to start following a system reboot.

   ```
   sudo systemctl start ptp4l
   ```

   ```
   sudo systemctl enable ptp4l
   ```

5. To configure phc2sys on a clock system that uses hardware time stamping:

   a. Edit the /etc/sysconfig/phc2sys file and define the start-up options for the phc2sys daemon.

      On a boundary clock or secondary clock, synchronize the system clock with the PTP hardware clock that's associated with the secondary network interface, for example:

      ```
      OPTIONS="-c CLOCK_REALTIME -s eno1 -w"
      ```

      > **Note:**
      >
      > The secondary network interface on a boundary clock is the one that it uses to communicate with the grandmaster clock.

      The -w option specifies that phc2sys waits until ptp4l has synchronized the PTP hardware clock before synchronizing the system clock.

On a grandmaster clock, which derives its system time from a reference time source such as GPS, CDMA, NTP, or a radio time signal, synchronize the network interface's PTP hardware clock from the system clock, for example:

```
OPTIONS="-c eno1 -s CLOCK_REALTIME -w"
```

See the `phc2sys(8)` manual page.

b. Start the `phc2sys` service and configure it to start following a system reboot.

```
sudo systemctl start phc2sys
```

```
sudo systemctl enable phc2sys
```

You can use the `pmc` command to query the status of `ptp4l` operation. The following example shows the results of running `pmc` on a slave clock system that's directly connected to the grandmaster clock system without any intermediate boundary clocks:

```
sudo pmc -u -b 0 'GET TIME_STATUS_NP'
```

```
sending: GET TIME_STATUS_NP
    080027.fffe.7f327b-0 seq 0 RESPONSE MANAGEMENT TIME_STATUS_NP
        master_offset              -98434
        ingress_time               1412169090025854874
        cumulativeScaledRateOffset +1.000000000
        scaledLastGmPhaseChange    0
        gmTimeBaseIndicator        0
        lastGmPhaseChange          0x0000'0000000000000000.0000
        gmPresent                  true
        gmIdentity                 080027.fffe.d9e453
```

```
sudo pmc -u -b 0 'GET CURRENT_DATA_SET'
```

```
sending: GET CURRENT_DATA_SET
    080027.fffe.7f327b-0 seq 0 RESPONSE MANAGEMENT CURRENT_DATA_SET
        stepsRemoved     1
        offsetFromMaster  42787.0
        meanPathDelay    289207.0
```

This output examples include the following useful information:

`gmIdentity`
The unique identifier of the grandmaster clock, which is based on the MAC address of its network interface.

`gmPresent`
Whether an external grandmaster clock is available. This value is displayed as `false` on the grandmaster clock itself.

`meanPathDelay`
An estimate of how many nanoseconds by which synchronization messages are delayed.

`offsetFromMaster`
The most recent measurement of the time difference in nanoseconds relative to the grandmaster clock.

`stepsRemoved`
The number of network steps between this system and the grandmaster clock.

For more information, see the `phc2sys(8)`, `pmc(8)`, and `ptp4l(8)` manual pages, and IEEE 1588.

Using PTP as a Time Source for NTP

You can make the PTP-adjusted system time on an NTP server available to NTP clients.

- Include the following entries in the `/etc/chrony.conf` file on the NTP server:

  ```
  server    127.127.1.0
  fudge     127.127.1.0 stratum 0
  ```

  These entries define the local system clock as the time reference.

  > **Note:**
  >
  > Don't configure any added `server` lines in the file.