# RedHat\_ch3

## Chapter 3. Analyze and Store Logs

```

Describe System Log Architecture

Quiz: Describe System Log Architecture

Review Syslog Files

Guided Exercise: Review Syslog Files

Review System Journal Entries

Guided Exercise: Review System Journal Entries

Preserve the System Journal

Guided Exercise: Preserve the System Journal

Maintain Accurate Time

Guided Exercise: Maintain Accurate Time

Lab: Analyze and Store Logs

Summary

Abstract

Goal Locate and accurately interpret system event logs for troubleshooting purposes.

Objectives

Describe the basic Red Hat Enterprise Linux logging architecture to record events.

Interpret events in the relevant syslog files to troubleshoot problems or to review system status.

Find and interpret entries in the system journal to troubleshoot problems or review system status.

Configure the system journal to preserve the record of events when a server is rebooted.

Maintain accurate time synchronization with Network Time Protocol (NTP) and configure the time zone to ensure correct time stamps for events that are recorded by the system journal and logs.

Sections

Describe System Log Architecture (and Quiz)

Review Syslog Files (and Guided Exercise)

Review System Journal Entries (and Guided Exercise)

Preserve the System Journal (and Guided Exercise)

Maintain Accurate Time (and Guided Exercise)

Lab

Analyze and Store Logs

Describe System Log Architecture

Objectives

Describe the basic Red Hat Enterprise Linux logging architecture to record events.

System Logging

The operating system kernel and other processes record a log of events that happen when the system is running. These logs are used to audit the system and to troubleshoot problems. You can use text utilities such as the less and tail commands to inspect these logs.

Red Hat Enterprise Linux uses a standard logging system that is based on the syslog protocol to log the system messages. Many programs use the logging system to record events and to organize them into log files. The systemd-journald and rsyslog services handle the syslog messages in Red Hat Enterprise Linux 9.

The systemd-journald service is at the heart of the operating system event logging architecture. The systemd-journald service collects event messages from many sources:

System kernel

Output from the early stages of the boot process

Standard output and standard error from daemons

Syslog events

The systemd-journald service restructures the logs into a standard format and writes them into a structured, indexed system journal. By default, this journal is stored on a file system that does not persist across reboots.

The rsyslog service reads syslog messages that the systemd-journald service receives from the journal when they arrive. The rsyslog service then processes the syslog events, and records them to its log files or forwards them to other services according to its own configuration.

The rsyslog service sorts and writes syslog messages to the log files that do persist across reboots in the /var/log directory. The service also sorts the log messages to specific log files according to the type of program that sent each message and the priority of each syslog message.

In addition to syslog message files, the /var/log directory contains log files from other services on the system. The following table lists some useful files in the /var/log directory.

Table 3.1. Selected System Log Files

Log file Type of stored messages

/var/log/messages Most syslog messages are logged here. Exceptions include messages about authentication and email processing, scheduled job execution, and purely debugging-related messages.

/var/log/secure Syslog messages about security and authentication events.

/var/log/maillog Syslog messages about the mail server.

/var/log/cron Syslog messages about scheduled job execution.

/var/log/boot.log Non-syslog console messages about system startup.

Some applications do not use the syslog service to manage their log messages. For example, the Apache Web Server saves log messages to files in a subdirectory of the /var/log directory.

References

systemd-journald.service(8), rsyslogd(8), and rsyslog.conf(5) man pages

For more information, refer to the Troubleshooting Problems Using Log Files section in the Red Hat Enterprise Linux 9 Configuring Basic System Settings guide at https://access.redhat.com/documentation/en-us/red\_hat\_enterprise\_linux/9/html-single/configuring\_basic\_system\_settings/index

```

### Quiz: Describe System Log Architecture

```

Choose the correct answer to the following questions:

1.

Which log file stores most syslog messages, except for the ones about authentication, mail, scheduled jobs, and debugging?

A

/var/log/maillog

B

/var/log/boot.log

C

/var/log/messages

D

/var/log/secure

2.

Which log file stores syslog messages about security and authentication operations in the system?

A

/var/log/maillog

B

/var/log/boot.log

C

/var/log/messages

D

/var/log/secure

3.

Which service sorts and organizes syslog messages into files in the /var/log directory?

A

rsyslog

B

systemd-journald

C

auditd

D

tuned

4.

Which directory accommodates the human-readable syslog files?

A

/sys/kernel/debug

B

/var/log/journal

C

/run/log/journal

D

/var/log

5.

Which file stores syslog messages about the mail server?

A

/var/log/lastlog

B

/var/log/maillog

C

/var/log/tallylog

D

/var/log/boot.log

6.

Which file stores syslog messages about scheduled jobs?

A

/var/log/cron

B

/var/log/tallylog

C

/var/log/spooler

D

/var/log/secure

7.

Which file stores console messages about system startup?

A

/var/log/messages

B

/var/log/cron

C

/var/log/boot.log

D

/var/log/secure

```

### Review Syslog Files

```

Objectives

Interpret events in relevant syslog files to troubleshoot problems or review system status.

Log Events to the System

Many programs use the syslog protocol to log events to the system. Each log message is categorized by facility (the subsystem that produces the message) and priority (the message's severity).

The following table lists the standard syslog facilities:

Table 3.2. Overview of Syslog Facilities

Code Facility Facility description

0 kern Kernel messages

1 user User-level messages

2 mail Mail system messages

3 daemon System daemon messages

4 auth Authentication and security messages

5 syslog Internal syslog messages

6 lpr Printer messages

7 news Network news messages

8 uucp UUCP protocol messages

9 cron Clock daemon messages

10 authpriv Non-system authorization messages

11 ftp FTP protocol messages

16-23 local0 to local7 Custom local messages

The following table lists the standard syslog priorities in descending order:

Table 3.3. Overview of Syslog Priorities

Code Priority Priority description

0 emerg System is unusable

1 alert Action must be taken immediately

2 crit Critical condition

3 err Non-critical error condition

4 warning Warning condition

5 notice Normal but significant event

6 info Informational event

7 debug Debugging-level message

The rsyslog service uses the facility and priority of log messages to determine how to handle them. Rules configure this facility and priority in the /etc/rsyslog.conf file and in any file in the /etc/rsyslog.d directory with the .conf extension. Software packages can add rules by installing an appropriate file in the /etc/rsyslog.d directory.

Each rule that controls how to sort syslog messages has a line in one of the configuration files. The left side of each line indicates the facility and priority of the syslog messages that the rule matches. The right side of each line indicates which file to save the log message in (or where else to deliver the message). An asterisk (\*) is a wildcard that matches all values.

For example, the following line in the /etc/rsyslog.d file would record messages that are sent to the authpriv facility at any priority to the /var/log/secure file:

authpriv.\* /var/log/secure

Sometimes, log messages match more than one rule in the rsyslog.conf file. In such cases, one message is stored in more than one log file. The none keyword in the priority field indicates that no messages for the indicated facility are stored in the given file, to limit stored messages.

Instead of being logged to a file, syslog messages can also be printed to the terminals of all logged-in users. The rsyslog.conf file has a setting to print all the syslog messages with the emerg priority to the terminals of all logged-in users.

Sample Rules of the rsyslog Service

#### RULES ####

# Log all kernel messages to the console.

# Logging much else clutters up the screen.

#kern.\* /dev/console

# Log anything (except mail) of level info or higher.

# Don't log private authentication messages!

\*.info;mail.none;authpriv.none;cron.none /var/log/messages

# The authpriv file has restricted access.

authpriv.\* /var/log/secure

# Log all the mail messages in one place.

mail.\* -/var/log/maillog

# Log cron stuff

cron.\* /var/log/cron

# Everybody gets emergency messages

.emerg :omusrmsg:

# Save news errors of level crit and higher in a special file.

uucp,news.crit /var/log/spooler

# Save boot messages also to boot.log

local7.\* /var/log/boot.log

Note

The syslog subsystem has many more features beyond the scope of this course. To explore further, refer to the rsyslog.conf(5) man page and the extensive HTML documentation at /usr/share/doc/rsyslog/html/index.html that the rsyslog-doc package provides.

Log File Rotation

The logrotate command rotates log files to prevent them from taking too much space in the /﻿var/log directory. When a log file is rotated, it is renamed with an extension that indicates the rotation date. For example, the previous /var/log/messages file is renamed to the /var/log/messages-20220320 file when it is rotated on 2022-03-20. After the previous log file rotates, it creates a log file and notifies the service that wrote the log file.

After rotations during typically four weeks, the earliest log file is discarded to free disk space. A scheduled job runs the logrotate command daily to see the rotation requirement of any log files. Most log files rotate weekly; the logrotate command rotates some log files faster, or more slowly, or when they reach a specific size.

Analyze a Syslog Entry

Log messages start with the earliest message at the start and the latest message at the end of the log file. The rsyslog service uses a standard format for recording entries in log files. The following example explains the anatomy of a log message in the /var/log/secure log file.

Mar 20 20:11:48 localhost sshd[1433]: Failed password for student from 172.25.0.10 port 59344 ssh2

Mar 20 20:11:48 : Records the time stamp of the log entry.

localhost : The host that sends the log message.

sshd[1433] : The program or process name and PID number that sent the log message.

Failed password for …​: The message that was sent.

Monitor Log Events

Monitoring log files for events is helpful to reproduce issues. The tail -f /path/to/file command outputs the last ten lines of the specified file and continues to output newly written lines in the file.

For example, to monitor for failed login attempts, run the tail command in one terminal, and then run in another terminal the ssh command as the root user while a user tries to log in to the system.

In the first terminal, run the tail command:

[root@host ~]# tail -f /var/log/secure

In the second terminal, run the ssh command:

[root@host ~]# ssh root@hosta

root@hosta's password: redhat

...output omitted...

[root@hostA ~]#

The log messages are visible in the first terminal.

...output omitted...

Mar 20 09:01:13 host sshd[2712]: Accepted password for root from 172.25.254.254 port 56801 ssh2

Mar 20 09:01:13 host sshd[2712]: pam\_unix(sshd:session): session opened for user root by (uid=0)

Send Syslog Messages Manually

The logger command sends messages to the rsyslog service. By default, the logger command sends the message to the user type with the notice priority (user.notice) unless specified otherwise with the -p option. It is helpful to test any change to the rsyslog service configuration.

To send a message to the rsyslog service to be recorded in the /var/log/boot.log log file, execute the following logger command:

[root@host ~]# logger -p local7.notice "Log entry created on host"

References

logger(1), tail(1), rsyslog.conf(5), and logrotate(8) man pages

rsyslog Manual

/usr/share/doc/rsyslog/html/index.html provided by the rsyslog-doc package

For further information, refer to Troubleshooting Problems Using Log Files at https://access.redhat.com/documentation/en-us/red\_hat\_enterprise\_linux/9/html-single/configuring\_basic\_system\_settings/assembly\_troubleshooting-problems-using-log-files\_configuring-basic-system-settings

```

### Guided Exercise: Review Syslog Files

```

In this exercise, you reconfigure the rsyslog service to write specific log messages to a new file.

Outcomes

Configure the rsyslog service to write all log messages with the debug priority to the /﻿var/log/messages-debug log file.

As the student user on the workstation machine, use the lab command to prepare your system for this exercise.

This command prepares your environment and ensures that all required resources are available.

[student@workstation ~]$ lab start logs-syslog

Instructions

Log in to the servera machine as the student user and switch to the root user.

[student@workstation ~]$ ssh student@servera

...output omitted...

[student@servera ~]$ sudo -i

[sudo] password for student: student

[root@servera ~]#

Configure the rsyslog service on the servera machine to log all messages with the debug or higher priority, for any service to the new /var/log/messages-debug log file by changing the /etc/rsyslog.d/debug.conf configuration file.

Create the /etc/rsyslog.d/debug.conf file with the necessary entries to redirect all log messages with the debug or higher priority to the /var/log/messages-debug log file.

\*.debug /var/log/messages-debug

This configuration line logs syslog messages with any facility and with the debug or higher priority level:

The wildcard (\*) in the facility field of the configuration line indicates any facility of log messages.

The rsyslog service writes the matching messages to the /var/log/messages-debug log file.

Restart the rsyslog service.

[root@servera ~]# systemctl restart rsyslog

Verify that all the log messages with the debug priority appear in the /var/log/messages-debug log file.

Generate a log message with the user type and the debug priority.

[root@servera ~]# logger -p user.debug "Debug Message Test"

View the last ten log messages from the /var/log/messages-debug log file, and verify that you see the Debug Message Test message among the other log messages.

[root@servera ~]# tail /var/log/messages-debug

Feb 13 18:22:38 servera systemd[1]: Stopping System Logging Service...

Feb 13 18:22:38 servera rsyslogd[25176]: [origin software="rsyslogd" swVersion="8.37.0-9.el8" x-pid="25176" x-info="http://www.rsyslog.com"] exiting on signal 15.

Feb 13 18:22:38 servera systemd[1]: Stopped System Logging Service.

Feb 13 18:22:38 servera systemd[1]: Starting System Logging Service...

Feb 13 18:22:38 servera rsyslogd[25410]: environment variable TZ is not set, auto correcting this to TZ=/etc/localtime [v8.37.0-9.el8 try http://www.rsyslog.com/e/2442 ]

Feb 13 18:22:38 servera systemd[1]: Started System Logging Service.

Feb 13 18:22:38 servera rsyslogd[25410]: [origin software="rsyslogd" swVersion="8.37.0-9.el8" x-pid="25410" x-info="http://www.rsyslog.com"] start

Feb 13 18:27:58 servera root[25416]: Debug Message Test

Return to the workstation system as the student user.

[root@servera ~]# exit

logout

[student@servera ~]$ exit

logout

Connection to servera closed.

[student@workstation ~]$

Finish

On the workstation machine, change to the student user home directory and use the lab command to complete this exercise. This step is important to ensure that resources from previous exercises do not impact upcoming exercises.

[student@workstation ~]$ lab finish logs-syslog

This concludes the section.

```

### Review System Journal Entries

```

Objectives

Find and interpret entries in the system journal to troubleshoot problems or review system status.

Find Events on the System Journal

The systemd-journald service stores logging data in a structured, indexed binary file called a journal. This data includes extra information about the log event. For example, for syslog events, this information includes the priority of the original message and the facility, which is a value that the syslog service assigns to track the process that originated a message.

Important

In Red Hat Enterprise Linux, the memory-based /run/log directory holds the system journal by default. The contents of the /run/log directory are lost when the system is shut down. You can change the journald directory to a persistent location, which is discussed later in this chapter.

To retrieve log messages from the journal, use the journalctl command. You can use the journalctl command to view all messages in the journal, or to search for specific events based on options and criteria. If you run the command as root, then you have full access to the journal. Although regular users can also use the journalctl command, the system restricts them from seeing certain messages.

[root@host ~]# journalctl

...output omitted...

Mar 15 04:42:16 host.lab.example.com systemd[2127]: Listening on PipeWire Multimedia System Socket.

Mar 15 04:42:16 host.lab.example.com systemd[2127]: Starting Create User's Volatile Files and Directories...

Mar 15 04:42:16 host.lab.example.com systemd[2127]: Listening on D-Bus User Message Bus Socket.

Mar 15 04:42:16 host.lab.example.com systemd[2127]: Reached target Sockets.

Mar 15 04:42:16 host.lab.example.com systemd[2127]: Finished Create User's Volatile Files and Directories.

Mar 15 04:42:16 host.lab.example.com systemd[2127]: Reached target Basic System.

Mar 15 04:42:16 host.lab.example.com systemd[1]: Started User Manager for UID 0.

Mar 15 04:42:16 host.lab.example.com systemd[2127]: Reached target Main User Target.

Mar 15 04:42:16 host.lab.example.com systemd[2127]: Startup finished in 90ms.

Mar 15 04:42:16 host.lab.example.com systemd[1]: Started Session 6 of User root.

Mar 15 04:42:16 host.lab.example.com sshd[2110]: pam\_unix(sshd:session): session opened for user root(uid=0) by (uid=0)

Mar 15 04:42:17 host.lab.example.com systemd[1]: Starting Hostname Service...

Mar 15 04:42:17 host.lab.example.com systemd[1]: Started Hostname Service.

lines 1951-2000/2000 (END) q

The journalctl command highlights important log messages; messages with the notice or warning priority are in bold text, whereas messages with the error priority or higher are in red text.

The key to successful use of the journal for troubleshooting and auditing is to limit journal searches to show only relevant output.

By default, the journalctl command -n option shows the last 10 log entries. You can adjust the number of log entries with an optional argument that specifies how many log entries to display. For example, to review the last five log entries, you can run the following journalctl command:

[root@host ~]# journalctl -n 5

Mar 15 04:42:17 host.lab.example.com systemd[1]: Started Hostname Service.

Mar 15 04:42:47 host.lab.example.com systemd[1]: systemd-hostnamed.service: Deactivated successfully.

Mar 15 04:47:33 host.lab.example.com systemd[2127]: Created slice User Background Tasks Slice.

Mar 15 04:47:33 host.lab.example.com systemd[2127]: Starting Cleanup of User's Temporary Files and Directories...

Mar 15 04:47:33 host.lab.example.com systemd[2127]: Finished Cleanup of User's Temporary Files and Directories.

Similar to the tail command, the journalctl command -f option outputs the last 10 lines of the system journal and continues to output new journal entries when the journal appends them. To exit the journalctl command -f option, use the Ctrl+C key combination.

[root@host ~]# journalctl -f

Mar 15 04:47:33 host.lab.example.com systemd[2127]: Finished Cleanup of User's Temporary Files and Directories.

Mar 15 05:01:01 host.lab.example.com CROND[2197]: (root) CMD (run-parts /etc/cron.hourly)

Mar 15 05:01:01 host.lab.example.com run-parts[2200]: (/etc/cron.hourly) starting 0anacron

Mar 15 05:01:01 host.lab.example.com anacron[2208]: Anacron started on 2022-03-15

Mar 15 05:01:01 host.lab.example.com anacron[2208]: Will run job `cron.daily' in 29 min.

Mar 15 05:01:01 host.lab.example.com anacron[2208]: Will run job `cron.weekly' in 49 min.

Mar 15 05:01:01 host.lab.example.com anacron[2208]: Will run job `cron.monthly' in 69 min.

Mar 15 05:01:01 host.lab.example.com anacron[2208]: Jobs will be executed sequentially

Mar 15 05:01:01 host.lab.example.com run-parts[2210]: (/etc/cron.hourly) finished 0anacron

Mar 15 05:01:01 host.lab.example.com CROND[2196]: (root) CMDEND (run-parts /etc/cron.hourly)

^C

[root@host ~]#

To help to troubleshoot problems, you can filter the output of the journal by the priority of the journal entries. The journalctl command -p option shows the journal entries with a specified priority level (by name or by number) or higher. The journalctl command processes the debug, info, notice, warning, err, crit, alert, and emerg priority levels, in ascending priority order.

As an example, run the following journalctl command to list journal entries with the err priority or higher:

[root@host ~]# journalctl -p err

Mar 15 04:22:00 host.lab.example.com pipewire-pulse[1640]: pw.conf: execvp error 'pactl': No such file or direct

Mar 15 04:22:17 host.lab.example.com kernel: Detected CPU family 6 model 13 stepping 3

Mar 15 04:22:17 host.lab.example.com kernel: Warning: Intel Processor - this hardware has not undergone testing by Red Hat and might not be certif>

Mar 15 04:22:20 host.lab.example.com smartd[669]: DEVICESCAN failed: glob(3) aborted matching pattern /dev/discs/disc\*

Mar 15 04:22:20 host.lab.example.com smartd[669]: In the system's table of devices NO devices found to scan

You can show messages for a specified systemd unit by using the journalctl command -u option and the unit name.

[root@host ~]# journalctl -u sshd.service

May 15 04:30:18 host.lab.example.com systemd[1]: Starting OpenSSH server daemon...

May 15 04:30:18 host.lab.example.com sshd[1142]: Server listening on 0.0.0.0 port 22.

May 15 04:30:18 host.lab.example.com sshd[1142]: Server listening on :: port 22.

May 15 04:30:18 host.lab.example.com systemd[1]: Started OpenSSH server daemon.

May 15 04:32:03 host.lab.example.com sshd[1796]: Accepted publickey for user1 from 172.25.250.254 port 43876 ssh2: RSA SHA256:1UGy...>

May 15 04:32:03 host.lab.example.com sshd[1796]: pam\_unix(sshd:session): session opened for user user1(uid=1000) by (uid=0)

May 15 04:32:26 host.lab.example.com sshd[1866]: Accepted publickey for user2 from ::1 port 36088 ssh2: RSA SHA256:M8ik...

May 15 04:32:26 host.lab.example.com sshd[1866]: pam\_unix(sshd:session): session opened for user user2(uid=1001) by (uid=0)

lines 1-8/8 (END) q

When looking for specific events, you can limit the output to a specific time frame. To limit the output to a specific time range, the journalctl command has the --since option and the --until option. Both options take a time argument in the "YYYY-MM-DD hh:mm:ss" format (the double quotation marks are required to preserve the space in the option).

The journalctl command assumes that the day starts at 00:00:00 when you omit the time argument. The command assumes the current day when you omit the day argument. Both options take yesterday, today, and tomorrow as valid arguments in addition to the date and time field.

As an example, run the following journalctl command to list all journal entries from today's records:

[root@host ~]# journalctl --since today

...output omitted...

Mar 15 05:04:20 host.lab.example.com systemd[1]: Started Session 8 of User student.

Mar 15 05:04:20 host.lab.example.com sshd[2255]: pam\_unix(sshd:session): session opened for user student(uid=1000) by (uid=0)

Mar 15 05:04:20 host.lab.example.com systemd[1]: Starting Hostname Service...

Mar 15 05:04:20 host.lab.example.com systemd[1]: Started Hostname Service.

Mar 15 05:04:50 host.lab.example.com systemd[1]: systemd-hostnamed.service: Deactivated successfully.

Mar 15 05:06:33 host.lab.example.com systemd[2261]: Starting Mark boot as successful...

Mar 15 05:06:33 host.lab.example.com systemd[2261]: Finished Mark boot as successful.

lines 1996-2043/2043 (END) q

Run the following journalctl command to list all journal entries from 2022-03-11 20:30:00 to 2022-03-14 10:00:00:

[root@host ~]# journalctl --since "2022-03-11 20:30" --until "2022-03-14 10:00"

...output omitted...

You can also specify all entries since a relative time to the present. For example, to specify all entries in the last hour, you can use the following command:

[root@host ~]# journalctl --since "-1 hour"

...output omitted...

Note

You can use other, more sophisticated time specifications with the --since and --until options. For some examples, see the systemd.time(7) man page.

In addition to the visible content of the journal, you can view additional log entries if you turn on the verbose output. You can use any displayed extra field to filter the output of a journal query. The verbose output is useful to reduce the output of complex searches for certain events in the journal.

[root@host ~]# journalctl -o verbose

Tue 2022-03-15 05:10:32.625470 EDT [s=e7623387430b4c14b2c71917db58e0ee;i...]

\_BOOT\_ID=beaadd6e5c5448e393ce716cd76229d4

\_MACHINE\_ID=4ec03abd2f7b40118b1b357f479b3112

PRIORITY=6

SYSLOG\_FACILITY=3

SYSLOG\_IDENTIFIER=systemd

\_UID=0

\_GID=0

\_TRANSPORT=journal

\_CAP\_EFFECTIVE=1ffffffffff

TID=1

CODE\_FILE=src/core/job.c

CODE\_LINE=744

CODE\_FUNC=job\_emit\_done\_message

JOB\_RESULT=done

\_PID=1

\_COMM=systemd

\_EXE=/usr/lib/systemd/systemd

\_SYSTEMD\_CGROUP=/init.scope

\_SYSTEMD\_UNIT=init.scope

\_SYSTEMD\_SLICE=-.slice

JOB\_TYPE=stop

MESSAGE\_ID=9d1aaa27d60140bd96365438aad20286

\_HOSTNAME=host.lab.example.com

\_CMDLINE=/usr/lib/systemd/systemd --switched-root --system --deserialize 31

\_SELINUX\_CONTEXT=system\_u:system\_r:init\_t:s0

UNIT=user-1000.slice

MESSAGE=Removed slice User Slice of UID 1000.

INVOCATION\_ID=0e5efc1b4a6d41198f0cf02116ca8aa8

JOB\_ID=3220

\_SOURCE\_REALTIME\_TIMESTAMP=1647335432625470

lines 46560-46607/46607 (END) q

The following list shows some fields of the system journal that you can use to search for relevant lines to a particular process or event:

\_COMM is the command name.

\_EXE is the path to the executable file for the process.

\_PID is the PID of the process.

\_UID is the UID of the user that runs the process.

\_SYSTEMD\_UNIT is the systemd unit that started the process.

You can combine multiple system journal fields to form a granular search query with the journalctl command. For example, the following journalctl command shows all related journal entries to the sshd.service systemd unit from a process with PID 2110.

[root@host ~]# journalctl \_SYSTEMD\_UNIT=sshd.service \_PID=2110

Mar 15 04:42:16 host.lab.example.com sshd[2110]: Accepted publickey for root from 172.25.250.254 port 46224 ssh2: RSA SHA256:1UGybTe52L2jzEJa1HLVKn9QUCKrTv3ZzxnMJol1Fro

Mar 15 04:42:16 host.lab.example.com sshd[2110]: pam\_unix(sshd:session): session opened for user root(uid=0) by (uid=0)

Note

For a list of journal fields, consult the systemd.journal-fields(7) man page.

References

journalctl(1), systemd.journal-fields(7), and systemd.time(7) man pages

For more information refer to the Troubleshooting Problems Using Log Files section in the Red Hat Enterprise Linux 9 Configuring Basic System Settings guide at https://access.redhat.com/documentation/en-us/red\_hat\_enterprise\_linux/9/html-single/configuring\_basic\_system\_settings/index#troubleshooting-problems-using-log-files\_getting-started-with-system-administration

```

### Guided Exercise: Review System Journal Entries

```

In this exercise, you search the system journal for entries to record events that match specific criteria.

Outcomes

Search the system journal for entries to record events based on different criteria.

As the student user on the workstation machine, use the lab command to prepare your system for this exercise.

This command prepares your environment and ensures that all required resources are available.

[student@workstation ~]$ lab start logs-systemd

Instructions

From the workstation machine, open an SSH session to the servera machine as the student user.

[student@workstation ~]$ ssh student@servera

...output omitted...

[student@servera ~]$

Use the journalctl command \_PID=1 option to display only log events that originate from the systemd PID 1 process on the servera machine. To quit from the journalctl command, press q. The following output is an example and might differ on your system:

[student@servera ~]$ journalctl \_PID=1

Mar 15 04:21:14 localhost systemd[1]: Finished Load Kernel Modules.

Mar 15 04:21:14 localhost systemd[1]: Finished Setup Virtual Console.

Mar 15 04:21:14 localhost systemd[1]: dracut ask for additional cmdline parameters was skipped because all trigger condition checks failed.

Mar 15 04:21:14 localhost systemd[1]: Starting dracut cmdline hook...

Mar 15 04:21:14 localhost systemd[1]: Starting Apply Kernel Variables...

lines 1-5 q

[student@servera ~]$

Use the journalctl command \_UID=81 option to display all log events that originated from a system service with a UID of 81 on the servera machine.

[student@servera ~]$ journalctl \_UID=81

Mar 15 04:21:17 servera.lab.example.com dbus-broker-lau[727]: Ready

Use the journalctl command -p warning option to display log events with a warning or higher priority on the servera machine.

[student@servera ~]$ journalctl -p warning

Mar 15 04:21:14 localhost kernel: wait\_for\_initramfs() called before rootfs\_initcalls

Mar 15 04:21:14 localhost kernel: ACPI: PRMT not present

Mar 15 04:21:14 localhost kernel: acpi PNP0A03:00: fail to add MMCONFIG information, can't access extended PCI configuration space under this bridge.

Mar 15 04:21:14 localhost kernel: device-mapper: core: CONFIG\_IMA\_DISABLE\_HTABLE is disabled. Duplicate IMA measurements will not be recorded in the IMA log.

...output omitted...

Mar 15 04:21:18 servera.lab.example.com NetworkManager[769]: <warn> [1647332478.5504] device (eth0): mtu: failure to set IPv6 MTU

Mar 15 04:21:27 servera.lab.example.com chronyd[751]: System clock wrong by -0.919695 seconds

Mar 15 04:22:34 servera.lab.example.com chronyd[751]: System clock wrong by 0.772805 seconds

Mar 15 05:41:11 servera.lab.example.com sshd[1104]: error: kex\_exchange\_identification: Connection closed by remote host

lines 1-19/19 (END) q

[student@servera ~]$

Display all recorded log events in the past 10 minutes from the current time on the servera machine.

[student@servera ~]$ journalctl --since "-10min"

Mar 15 05:40:01 servera.lab.example.com anacron[1092]: Job `cron.weekly' started

Mar 15 05:40:01 servera.lab.example.com anacron[1092]: Job `cron.weekly' terminated

Mar 15 05:41:11 servera.lab.example.com sshd[1104]: error: kex\_exchange\_identification: Connection closed by remote host

Mar 15 05:41:11 servera.lab.example.com sshd[1104]: Connection closed by 172.25.250.9 port 45370

Mar 15 05:41:14 servera.lab.example.com sshd[1105]: Accepted publickey for student from 172.25.250.9 port 45372 ssh2: RSA SHA256:M8ikhcEDm2tQ95Z0o7ZvufqEixCFCt+wowZLNzNlBT0

Mar 15 05:41:14 servera.lab.example.com systemd[1]: Created slice User Slice of UID 1000.

Mar 15 05:41:14 servera.lab.example.com systemd[1]: Starting User Runtime Directory /run/user/1000...

Mar 15 05:41:14 servera.lab.example.com systemd-logind[739]: New session 1 of user student.

Mar 15 05:41:14 servera.lab.example.com systemd[1]: Finished User Runtime Directory /run/user/1000.

Mar 15 05:41:14 servera.lab.example.com systemd[1]: Starting User Manager for UID 1000...

...output omitted...

Mar 15 05:44:56 servera.lab.example.com systemd[1109]: Stopped target Sockets.

Mar 15 05:44:56 servera.lab.example.com systemd[1109]: Stopped target Timers.

Mar 15 05:44:56 servera.lab.example.com systemd[1109]: Stopped Mark boot as successful after the user session has run 2 minutes.

Mar 15 05:44:56 servera.lab.example.com systemd[1109]: Stopped Daily Cleanup of User's Temporary Directories.

lines 1-48 q

[student@servera ~]$

Use the journalctl command --since and \_SYSTEMD\_UNIT="sshd.service" options to display all the recorded log events that originated from the sshd service since 09:00:00 this morning on the servera machine.

Note

Online classrooms typically run on the UTC time zone. To obtain results that start at 9:00 AM in your local time zone, adjust your --since value by the amount of your offset from UTC. Alternatively, ignore the local time and use a value of 9:00 to locate journal entries that occurred since 9:00 for the servera time zone.

[student@servera ~]$ journalctl --since 9:00:00 \_SYSTEMD\_UNIT="sshd.service"

Mar 15 09:41:14 servera.lab.example.com sshd[1105]: Accepted publickey for student from 172.25.250.9 port 45372 ssh2: RSA SHA256:M8ikhcEDm2tQ95Z0o7ZvufqEixCFCt+wowZLNzNlBT0

Mar 15 09:41:15 servera.lab.example.com sshd[1105]: pam\_unix(sshd:session): session opened for user student(uid=1000) by (uid=0)

Mar 15 09:44:56 servera.lab.example.com sshd[1156]: Accepted publickey for student from 172.25.250.9 port 45374 ssh2: RSA SHA256:M8ikhcEDm2tQ95Z0o7ZvufqEixCFCt+wowZLNzNlBT0

Mar 15 09:44:56 servera.lab.example.com sshd[1156]: pam\_unix(sshd:session): session opened for user student(uid=1000) by (uid=0)

Return to the workstation system as the student user.

[student@servera ~]$ exit

logout

Connection to servera closed.

[student@workstation ~]$

Finish

On the workstation machine, change to the student user home directory and use the lab command to complete this exercise. This step is important to ensure that resources from previous exercises do not impact upcoming exercises.

[student@workstation ~]$ lab finish logs-systemd

This concludes the section.

```

### Preserve the System Journal

```

Objectives

Configure the system journal to preserve the record of events when a server is rebooted.

System Journal Storage

By default, Red Hat Enterprise Linux 9 stores the system journal in the /run/log directory, and the system clears the system journal after a reboot. You can change the configuration settings of the systemd-journald service in the /etc/systemd/journald.conf file so that the journals persist across a reboot.

The Storage parameter in the /etc/systemd/journald.conf file defines whether to store system journals in a volatile manner or persistently across a reboot. Set this parameter to persistent, volatile, auto, or none, as follows:

persistent: Stores journals in the /var/log/journal directory, which persists across reboots. If the /var/log/journal directory does not exist, then the systemd-journald service creates it.

volatile: Stores journals in the volatile /run/log/journal directory. Because the /run file system is temporary and exists only in the runtime memory, the data in it, including system journals, does not persist across a reboot.

auto: If the /var/log/journal directory exists, then the systemd-journald service uses persistent storage; otherwise it uses volatile storage. This action is the default if you do not set the Storage parameter.

none: Do not use any storage. The system drops all logs, but you can still forward the logs.

The advantage of persistent system journals is that the historical data is available immediately at boot. However, even with a persistent journal, the system does not keep all data forever. The journal has a built-in log rotation mechanism that triggers monthly. In addition, the system does not allow the journals to get larger than 10% of the file system that they are on, or leaving less than 15% of the file system free. You can modify these values for both the runtime and persistent journals in the /etc/systemd/journald.conf configuration file.

The systemd-journald process logs the current limits on the size of the journal when it starts. The following command output shows the journal entries that reflect the current size limits:

[user@host ~]$ journalctl | grep -E 'Runtime Journal|System Journal'

Mar 15 04:21:14 localhost systemd-journald[226]: Runtime Journal (/run/log/journal/4ec03abd2f7b40118b1b357f479b3112) is 8.0M, max 113.3M, 105.3M free.

Mar 15 04:21:19 host.lab.example.com systemd-journald[719]: Runtime Journal (/run/log/journal/4ec03abd2f7b40118b1b357f479b3112) is 8.0M, max 113.3M, 105.3M free.

Mar 15 04:21:19 host.lab.example.com systemd-journald[719]: System Journal (/run/log/journal/4ec03abd2f7b40118b1b357f479b3112) is 8.0M, max 4.0G, 4.0G free.

Note

In the previous grep command, the vertical bar (|) symbol acts as an or operator. That is, the grep command matches any line with either the Runtime Journal string or the System Journal string from the journalctl command output. This command fetches the current size limits on the volatile (Runtime) journal store and on the persistent (System) journal store.

Configure Persistent System Journals

Configure the systemd-journald service as follows to preserve system journals persistently across a reboot:

Create the /var/log/journal directory.

[root@host ~]# mkdir /var/log/journal

Set the Storage parameter to the persistent value in the /etc/systemd/journald.conf file. Run your chosen text editor as the superuser to edit the /etc/systemd/journald.conf file.

[Journal]

Storage=persistent

...output omitted...

Restart the systemd-journald service to apply the configuration changes.

[root@host ~]# systemctl restart systemd-journald

If the systemd-journald service successfully restarts, then the service creates subdirectories in the /var/log/journal directory. The subdirectory in the /var/log/journal directory has hexadecimal characters in its long name and contain files with the .journal extension. The .journal binary files store the structured and indexed journal entries.

[root@host ~]# ls /var/log/journal

4ec03abd2f7b40118b1b357f479b3112

[root@host ~]# ls /var/log/journal/4ec03abd2f7b40118b1b357f479b3112

system.journal user-1000.journal

Although the system journals persist after a reboot, the journalctl command output includes entries from the current system boot as well as from the previous system boots. To limit the output to a specific system boot, use the journalctl command -b option. The following journalctl command retrieves the entries from the first system boot only:

[root@host ~]# journalctl -b 1

...output omitted...

The following journalctl command retrieves the entries from the second system boot only. The argument is meaningful only if the system was rebooted at least twice:

[root@host ~]# journalctl -b 2

...output omitted...

You can list the system boot events that the journalctl command recognizes, by using the --list-boots option.

[root@host ~]# journalctl --list-boots

-6 27de... Wed 2022-04-13 20:04:32 EDT—Wed 2022-04-13 21:09:36 EDT

-5 6a18... Tue 2022-04-26 08:32:22 EDT—Thu 2022-04-28 16:02:33 EDT

-4 e2d7... Thu 2022-04-28 16:02:46 EDT—Fri 2022-05-06 20:59:29 EDT

-3 45c3... Sat 2022-05-07 11:19:47 EDT—Sat 2022-05-07 11:53:32 EDT

-2 dfae... Sat 2022-05-07 13:11:13 EDT—Sat 2022-05-07 13:27:26 EDT

-1 e754... Sat 2022-05-07 13:58:08 EDT—Sat 2022-05-07 14:10:53 EDT

0 ee2c... Mon 2022-05-09 09:56:45 EDT—Mon 2022-05-09 12:57:21 EDT

The following journalctl command retrieves the entries from the current system boot only:

[root@host ~]# journalctl -b

...output omitted...

Note

When debugging a system crash with a persistent journal, usually you must limit the journal query to the reboot before the crash happened. You can use the journalctl command -b option with a negative number to indicate how many earlier system boots to include in the output. For example, the journalctl -b -1 command limits the output to only the previous boot.

References

systemd-journald.conf(5), systemd-journald(8) man pages

For more information, refer to the Troubleshooting Problems Using Log Files section in the Red Hat Enterprise Linux 9 Configuring Basic System Settings guide at https://access.redhat.com/documentation/en-us/red\_hat\_enterprise\_linux/9/html-single/configuring\_basic\_system\_settings/index#troubleshooting-problems-using-log-files\_getting-started-with-system-administration

```

### Guided Exercise: Preserve the System Journal

```

In this exercise, you configure the system journal to preserve its data after a reboot.

Outcomes

Configure the system journal to preserve its data after a reboot.

As the student user on the workstation machine, use the lab command to prepare your system for this exercise.

This command prepares your environment and ensures that all required resources are available.

[student@workstation ~]$ lab start logs-preserve

Instructions

From the workstation machine, log in to the servera machine as the student user.

[student@workstation ~]$ ssh student@servera

...output omitted...

[student@servera ~]$

As the superuser, confirm that the /var/log/journal directory does not exist. Use the ls command to list the /var/log/journal directory contents. Use the sudo command to elevate the student user privileges. If prompted, use the student password.

[student@servera ~]$ sudo ls /var/log/journal

[sudo] password for student: student

ls: cannot access '/var/log/journal': No such file or directory

Because the /var/log/journal directory does not exist, the systemd-journald service does not preserve the log data after a reboot.

Configure the systemd-journald service on the servera machine to preserve journals after a reboot.

Create the /var/log/journal directory.

[student@servera ~]$ sudo mkdir /var/log/journal

Uncomment the Storage=auto line in the /etc/systemd/journald.conf file, and set the Storage parameter to the persistent value. You might use the sudo vim /etc/systemd/journald.conf command to edit the configuration file. You can type /Storage=auto in the vim editor command mode to search for the Storage=auto line.

...output omitted...

[Journal]

Storage=persistent

...output omitted...

Restart the systemd-journald service to apply the configuration changes.

[student@servera ~]$ sudo systemctl restart systemd-journald.service

Verify that the systemd-journald service on the servera machine preserves its journals so that they persist after a reboot.

Restart the servera machine.

[student@servera ~]$ sudo systemctl reboot

Connection to servera closed by remote host.

Connection to servera closed.

[student@workstation ~]$

The SSH connection terminates as soon as you restart the servera machine.

Log in to the servera machine.

[student@workstation ~]$ ssh student@servera

...output omitted...

[student@servera ~]$

Verify that a subdirectory with a long hexadecimal name exists in the /var/log/journal directory. You can find the journal files in that directory. The subdirectory name on your system might be different.

[student@servera ~]$ sudo ls /var/log/journal

[sudo] password for student: student

63b272eae8d5443ca7aaa5593479b25f

[student@servera ~]$ sudo ls /var/log/journal/63b272eae8d5443ca7aaa5593479b25f

system.journal user-1000.journal

Return to the workstation system as the student user.

[student@servera ~]$ exit

logout

Connection to servera closed.

Finish

On the workstation machine, change to the student user home directory and use the lab command to complete this exercise. This step is important to ensure that resources from previous exercises do not impact upcoming exercises.

[student@workstation ~]$ lab finish logs-preserve

This concludes the section.

```

### Maintain Accurate Time

```

Objectives

Maintain accurate time synchronization with Network Time Protocol (NTP) and configure the time zone to ensure correct time stamps for events recorded by the system journal and logs.

Administer Local Clocks and Time Zones

System time synchronization is critical for log file analysis across multiple systems. Also, some services might require time synchronization to work correctly. Machines use the Network Time Protocol to provide and obtain correct time information over the internet. A machine might get accurate time information from public NTP services, such as the NTP Pool Project. Another option is to sync with a high-quality hardware clock to serve accurate time to local clients.

The timedatectl command shows an overview of the current time-related system settings, including the current time, time zone, and NTP synchronization settings of the system.

[user@host ~]$ timedatectl

Local time: Wed 2022-03-16 05:53:05 EDT

Universal time: Wed 2022-03-16 09:53:05 UTC

RTC time: Wed 2022-03-16 09:53:05

Time zone: America/New\_York (EDT, -0400)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

You can list a database of time zones with the timedatectl command list-timezones option:

[user@host ~]$ timedatectl list-timezones

Africa/Abidjan

Africa/Accra

Africa/Addis\_Ababa

Africa/Algiers

Africa/Asmara

Africa/Bamako

...output omitted...

The Internet Assigned Numbers Authority (IANA) provides a public time zone database, and the timedatectl command bases the time zone names on that database. IANA names time zones based on the continent or ocean, and then typically (not always) the largest city within the time zone region. For example, most of the US Mountain time zone is America/Denver.

Some localities inside the time zone have different daylight saving time rules. For example, in the US, much of the state of Arizona (US Mountain time) does not change to daylight saving time, and is in the America/Phoenix time zone.

Use the tzselect command to identify the correct time zone name. This command interactively prompts the user with questions about the system's location, and outputs the name of the correct time zone. It does not change the system's time zone setting.

The root user can change the system setting to update the current time zone with the timedatectl command set-timezone option. For example, the following timedatectl command updates the current time zone to America/Phoenix.

[root@host ~]# timedatectl set-timezone America/Phoenix

[root@host ~]# timedatectl

Local time: Wed 2022-03-16 03:05:55 MST

Universal time: Wed 2022-03-16 10:05:55 UTC

RTC time: Wed 2022-03-16 10:05:55

Time zone: America/Phoenix (MST, -0700)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

Note

You can set a server's time zone to Coordinated Universal Time (UTC). The tzselect command does not include the name of the UTC time zone. Use the timedatectl set-timezone UTC command to set the system's current time zone to UTC.

Use the timedatectl command set-time option to change the system's current time. You might specify the time in the "YYYY-MM-DD hh:mm:ss" format, where you can omit either the date or the time. For example, the following timedatectl command changes the time to 09:00:00.

[root@host ~]# timedatectl set-time 9:00:00

[root@host ~]# timedatectl

Local time: Fri 2019-04-05 09:00:27 MST

Universal time: Fri 2019-04-05 16:00:27 UTC

RTC time: Fri 2019-04-05 16:00:27

Time zone: America/Phoenix (MST, -0700)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

Note

The previous example might fail with the "Failed to set time: Automatic time synchronization is enabled" error message. In that case, first disable the automatic time synchronization before manually setting the date or time, as explained after this note.

The timedatectl command set-ntp option enables or disables NTP synchronization for automatic time adjustment. The option requires either a true or a false argument to turn it on or off. For example, the following timedatectl command turns off NTP synchronization.

[root@host ~]# timedatectl set-ntp false

Note

In Red Hat Enterprise Linux 9, the timedatectl set-ntp command adjusts whether the chronyd NTP service is enabled. Other Linux distributions might use this setting to adjust a different NTP or a Simple Network Time Protocol (SNTP) service.

Enabling or disabling NTP with other utilities in Red Hat Enterprise Linux, such as in the graphical GNOME Settings application, also updates this setting.

Configure and Monitor the chronyd Service

The chronyd service tracks the usually inaccurate local Real-Time Clock (RTC) by synchronizing it to the configured NTP servers. If no network connectivity is available, then the chronyd service calculates the RTC clock drift, and records it in the file that the driftfile value specifies in the /etc/chrony.conf configuration file.

By default, the chronyd service uses servers from the NTP Pool Project to synchronize time and requires no additional configuration. You might need to change the NTP servers for a machine that runs on an isolated network.

The stratum of the NTP time source determines its quality. The stratum determines the number of hops that the machine is away from a high-performance reference clock. The reference clock is a stratum 0 time source. An NTP server that is directly attached to the reference clock is a stratum 1 time source. A machine that synchronizes time from the NTP server is a stratum 2 time source.

The server and the peer are the two categories of time sources that you can declare in the /etc/chrony.conf configuration file. The server is one stratum above the local NTP server, and the peer is at the same stratum level. You can define multiple servers and peers in the chronyd configuration file, one per line.

The first argument of the server line is the IP address or DNS name of the NTP server. Following the server IP address or name, you can list a series of options for the server. Red Hat recommends using the iburst option, because then the chronyd service takes four measurements in a short time period for a more accurate initial clock synchronization after the service starts. For more information about the chronyd configuration file options, use the man 5 chrony.conf command.

As an example, with the following server classroom.example.com iburst line in the /﻿etc/chrony.conf configuration file, the chronyd service uses the classroom.example.com server as the NTP time source.

# Use public servers from the pool.ntp.org project.

...output omitted...

server classroom.example.com iburst

...output omitted...

Restart the service after pointing the chronyd service to the classroom.example.com local time source.

[root@host ~]# systemctl restart chronyd

The chronyc command acts as a client to the chronyd service. After setting up NTP synchronization, verify that the local system is seamlessly using the NTP server to synchronize the system clock, by using the chronyc sources command. For more verbose output with additional explanations about the output, use the chronyc sources -v command.

[root@host ~]# 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

===============================================================================

^\* 172.25.254.254 3 6 17 26 +2957ns[+2244ns] +/- 25ms

The asterisk character (\*) in the S (Source state) field indicates that the chronyd service uses the classroom.example.com server as a time source and is the NTP server that the machine is currently synchronized to.

References

timedatectl(1), tzselect(8), chronyd(8), chrony.conf(5), and chronyc(1) man pages

```

### Guided Exercise: Maintain Accurate Time

```

In this exercise, you adjust the time zone on a server and ensure that its system clock is synchronized with an NTP time source.

Outcomes

Change the time zone on a server.

Configure the server to synchronize its time with an NTP time source.

As the student user on the workstation machine, use the lab command to prepare your system for this exercise.

This command prepares your environment and ensures that all required resources are available.

[student@workstation ~]$ lab start logs-maintain

Instructions

Log in to the servera machine as the student user.

[student@workstation ~]$ ssh student@servera

...output omitted...

[student@servera ~]$

For this exercise, pretend that the servera machine is relocated to Haiti and that you need to update the time zone. Elevate the privileges of the student user to run the timedatectl command to update the time zone.

Select the appropriate time zone for Haiti.

[student@servera ~]$ tzselect

Please identify a location so that time zone rules can be set correctly.

Please select a continent, ocean, "coord", or "TZ".

1) Africa

2) Americas

3) Antarctica

4) Asia

5) Atlantic Ocean

6) Australia

7) Europe

8) Indian Ocean

9) Pacific Ocean

10) coord - I want to use geographical coordinates.

11) TZ - I want to specify the timezone using the Posix TZ format.

#? 2

Please select a country whose clocks agree with yours.

1) Anguilla 19) Dominican Republic 37) Peru

2) Antigua & Barbuda 20) Ecuador 38) Puerto Rico

3) Argentina 21) El Salvador 39) St Barthelemy

4) Aruba 22) French Guiana 40) St Kitts & Nevis

5) Bahamas 23) Greenland 41) St Lucia

6) Barbados 24) Grenada 42) St Maarten (Dutch)

7) Belize 25) Guadeloupe 43) St Martin (French)

8) Bolivia 26) Guatemala 44) St Pierre & Miquelon

9) Brazil 27) Guyana 45) St Vincent

10) Canada 28) Haiti 46) Suriname

11) Caribbean NL 29) Honduras 47) Trinidad & Tobago

12) Cayman Islands 30) Jamaica 48) Turks & Caicos Is

13) Chile 31) Martinique 49) United States

14) Colombia 32) Mexico 50) Uruguay

15) Costa Rica 33) Montserrat 51) Venezuela

16) Cuba 34) Nicaragua 52) Virgin Islands (UK)

17) Curaçao 35) Panama 53) Virgin Islands (US)

18) Dominica 36) Paraguay

#? 28

The following information has been given:

Haiti

Therefore TZ='America/Port-au-Prince' will be used.

Selected time is now: Wed Mar 16 07:10:35 EDT 2022.

Universal Time is now: Wed Mar 16 11:10:35 UTC 2022.

Is the above information OK?

1) Yes

2) No

#? 1

You can make this change permanent for yourself by appending the line

TZ='America/Port-au-Prince'; export TZ

to the file '.profile' in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you

can use the /usr/bin/tzselect command in shell scripts:

America/Port-au-Prince

Update the time zone on the servera machine to America/Port-au-Prince.

[student@servera ~]$ sudo timedatectl set-timezone \

America/Port-au-Prince

[sudo] password for student: student

Verify that you correctly set the time zone to America/Port-au-Prince.

[student@servera ~]$ timedatectl

Local time: Wed 2022-03-16 07:13:25 EDT

Universal time: Wed 2022-03-16 11:13:25 UTC

RTC time: Wed 2022-03-16 11:13:24

Time zone: America/Port-au-Prince (EDT, -0400)

System clock synchronized: no

NTP service: inactive

RTC in local TZ: no

Configure the chronyd service on the servera machine to synchronize the system time with the classroom.example.com server as the NTP time source.

Edit the /etc/chrony.conf configuration file to specify the classroom.example.com server as the NTP time source. The following output shows the configuration line to add to the configuration file, which includes the iburst option to speed up initial time synchronization:

...output omitted...

server classroom.example.com iburst

...output omitted...

Enable time synchronization on the servera machine. The command activates the NTP server with the settings from the /etc/chrony.conf configuration file. That command might activate either the chronyd or the ntpd service, depending on which service is currently installed on the system.

[student@servera ~]$ sudo timedatectl set-ntp true

Verify that the servera machine configuration synchronizes with the classroom.example.com time source in the classroom environment.

Verify that time synchronization is enabled on the servera machine.

Note

If the output shows that the clock is not synchronized, then wait for a few seconds and rerun the timedatectl command. It takes a few seconds to successfully synchronize the time settings with the time source.

[student@servera ~]$ timedatectl

Local time: Wed 2022-03-16 07:24:13 EDT

Universal time: Wed 2022-03-16 11:24:13 UTC

RTC time: Wed 2022-03-16 11:24:13

Time zone: America/Port-au-Prince (EDT, -0400)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

Verify that the servera machine currently synchronizes its time settings with the classroom.example.com time source.

The output shows an asterisk character (\*) in the source state (S) field for the classroom.example.com NTP time source. The asterisk indicates that the local system time successfully synchronizes with the NTP time source.

[student@servera ~]$ 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

===============================================================================

^\* 172.25.254.254 2 6 377 33 +84us[ +248us] +/- 21ms

Return to the workstation system as the student user.

[student@servera ~]$ exit

logout

Connection to servera closed.

[student@workstation ~]$

Finish

On the workstation machine, change to the student user home directory and use the lab command to complete this exercise. This step is important to ensure that resources from previous exercises do not impact upcoming exercises.

[student@workstation ~]$ lab finish logs-maintain

This concludes the section.

```

### Lab: Analyze and Store Logs

```

In this lab, you change the time zone on an existing server and configure a new log file for all events for authentication failures.

Outcomes

Update the time zone on an existing server.

Configure a new log file to store all messages for authentication failures.

As the student user on the workstation machine, use the lab command to prepare your system for this exercise.

This command prepares your environment and ensures that all required resources are available.

[student@workstation ~]$ lab start logs-review

Instructions

Log in to the serverb machine as the student user.

[student@workstation ~]$ ssh student@serverb

...output omitted...

[student@serverb ~]$

Pretend that the serverb machine is relocated to Jamaica and that you must update the time zone to America/Jamaica. Verify that you correctly set the appropriate time zone.

Select the appropriate time zone for Jamaica.

[student@serverb ~]$ tzselect

Please identify a location so that time zone rules can be set correctly.

Please select a continent, ocean, "coord", or "TZ".

1) Africa

2) Americas

3) Antarctica

4) Asia

5) Atlantic Ocean

6) Australia

7) Europe

8) Indian Ocean

9) Pacific Ocean

10) coord - I want to use geographical coordinates.

11) TZ - I want to specify the timezone using the Posix TZ format.

#? 2

Please select a country whose clocks agree with yours.

1) Anguilla 19) Dominican Republic 37) Peru

2) Antigua & Barbuda 20) Ecuador 38) Puerto Rico

3) Argentina 21) El Salvador 39) St Barthelemy

4) Aruba 22) French Guiana 40) St Kitts & Nevis

5) Bahamas 23) Greenland 41) St Lucia

6) Barbados 24) Grenada 42) St Maarten (Dutch)

7) Belize 25) Guadeloupe 43) St Martin (French)

8) Bolivia 26) Guatemala 44) St Pierre & Miquelon

9) Brazil 27) Guyana 45) St Vincent

10) Canada 28) Haiti 46) Suriname

11) Caribbean NL 29) Honduras 47) Trinidad & Tobago

12) Cayman Islands 30) Jamaica 48) Turks & Caicos Is

13) Chile 31) Martinique 49) United States

14) Colombia 32) Mexico 50) Uruguay

15) Costa Rica 33) Montserrat 51) Venezuela

16) Cuba 34) Nicaragua 52) Virgin Islands (UK)

17) Curaçao 35) Panama 53) Virgin Islands (US)

18) Dominica 36) Paraguay

#? 30

The following information has been given:

Jamaica

Therefore TZ='America/Jamaica' will be used.

Selected time is now: Wed Mar 16 07:17:15 EST 2022.

Universal Time is now: Wed Mar 16 12:17:15 UTC 2022.

Is the above information OK?

1) Yes

2) No

#? 1

You can make this change permanent for yourself by appending the line

TZ='America/Jamaica'; export TZ

to the file '.profile' in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you

can use the /usr/bin/tzselect command in shell scripts:

America/Jamaica

Elevate the student user privileges to update the time zone of the serverb server to America/Jamaica.

[student@serverb ~]$ sudo timedatectl set-timezone America/Jamaica

[sudo] password for student: student

Verify that you successfully set the time zone to America/Jamaica.

[student@serverb ~]$ timedatectl

Local time: Wed 2022-03-16 07:18:40 EST

Universal time: Wed 2022-03-16 12:18:40 UTC

RTC time: Wed 2022-03-16 12:18:40

Time zone: America/Jamaica (EST, -0500)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

View the recorded log events in the previous 30 minutes on the serverb machine.

Determine the time frame to view the journal entries.

[student@serverb ~]$ date

Wed Mar 16 07:19:29 AM EST 2022

[student@serverb ~]$ date -d "-30 minutes"

Wed Mar 16 06:49:38 AM EST 2022

View the recorded log events in the previous 30 minutes on the serverb machine.

[student@serverb ~]$ journalctl --since 06:49:00 --until 07:19:00

...output omitted...

Mar 16 07:10:58 localhost kernel: x86/PAT: Configuration [0-7]: WB WC UC- UC WB WP UC- WT

Mar 16 07:10:58 localhost kernel: found SMP MP-table at [mem 0x000f5bd0-0x000f5bdf]

Mar 16 07:10:58 localhost kernel: Using GB pages for direct mapping

Mar 16 07:10:58 localhost kernel: RAMDISK: [mem 0x2e0d9000-0x33064fff]

Mar 16 07:10:58 localhost kernel: ACPI: Early table checksum verification disabled

Mar 16 07:10:58 localhost kernel: ACPI: RSDP 0x00000000000F5B90 000014 (v00 BOCHS )

Mar 16 07:10:58 localhost kernel: ACPI: RSDT 0x000000007FFE12C4 00002C (v01 BOCHS BXPCRSDT 00000001 BXPC 00000001)

Mar 16 07:10:58 localhost kernel: ACPI: FACP 0x000000007FFE11D0 000074 (v01 BOCHS BXPCFACP 00000001 BXPC 00000001)

Mar 16 07:10:58 localhost kernel: ACPI: DSDT 0x000000007FFDFDC0 001410 (v01 BOCHS BXPCDSDT 00000001 BXPC 00000001)

lines 1-50/50 q

[student@serverb ~]$

Create the /etc/rsyslog.d/auth-errors.conf file. Configure the rsyslog service to write the Logging test authpriv.alert message to the /var/log/auth-errors file. Use the authpriv facility and the alert priority.

Create the /etc/rsyslog.d/auth-errors.conf file and specify the new /var/log/auth-errors file as the destination for authentication and security messages.

authpriv.alert /var/log/auth-errors

Restart the rsyslog service to apply the configuration file changes.

[student@serverb ~]$ sudo systemctl restart rsyslog

Use the logger -p command to write the Logging test authpriv.alert message to the /var/log/auth-errors file. Use the authpriv facility and the alert priority.

[student@serverb ~]$ logger -p authpriv.alert "Logging test authpriv.alert"

Verify that the /var/log/auth-errors file contains the log entry with the Logging test authpriv.alert message.

[student@serverb ~]$ sudo tail /var/log/auth-errors

Mar 16 07:25:12 serverb student[1339]: Logging test authpriv.alert

Return to the workstation system as the student user.

[student@serverb ~]$ exit

logout

Connection to serverb closed.

[student@workstation ~]$

Evaluation

As the student user on the workstation machine, use the lab command to grade your work. Correct any reported failures and rerun the command until successful.

[student@workstation ~]$ lab grade logs-review

Finish

On the workstation machine, change to the student user home directory and use the lab command to complete this exercise. This step is important to ensure that resources from previous exercises do not impact upcoming exercises.

[student@workstation ~]$ lab finish logs-review

This concludes the section.

```

### Summary

The systemd-journald and rsyslog services capture and write log messages to the appropriate files.

The /var/log directory contains log files.

Periodic rotation of log files prevents them from filling up the file-system space.

The systemd journals are temporary and do not persist across a reboot.

The chronyd service helps to synchronize time settings with a time source.

You can update the time zone of the server based on its location.

[RadHad\_ch4](/44gek5blQiG6jQWVr3hS8g)