11.1.4 Configure journald

Click one of the buttons to take you to that part of the video.

Configure jounald 0:00-0:14

In this demonstration, we're going to look at logging using journald. Understand that newer Linux distributions that use the systemd daemon will use typically the journald daemon for logging. Older Linux system that used init used the syslog daemon.

Find journald in the File System 0:15-0:54

The journald daemon maintains a system log file called the journal that's located in /var/log/journal. Within this, we should see a subdirectory here that contains the actual journal file.

In the old days when we were dealing with init and syslog, the log file was named, messages, and it was just a text file and you could view it with any text manipulation utility you wanted. You could view it with cat, less, head, tail, whatever it is you wanted to use.

That's not the case with the journal. The journal is not a straight text file. You have to use a special command to view it.

Use the journalctl Command to View Journal 0:55-1:46

That command is journalctl. If you enter the journalctl command with no parameters, then just the entire journal is displayed, as you can see. It starts with the earliest entries and starts working its way through.

As you can see, there are a lot of entries. These entries are from about a month ago. I'm going to press Q to get out.

Just using the journalctl command by itself is not very useful because of the fact that it displays the oldest entries first and you're trying to work down, most likely, to entries that occurred just recently.

If you've had a Linux system that's been up a long time, that can take quite a while to accomplish. Let's look at the man page for journalctl. If we page down here, two of the most useful options with journalctl in my opinion are the -r options and the -f options.

Use -r Option 1:47-2:20

Let's look at -r first. Notice here it tells us that it reverses the outputs so the newest entries are displayed first. That all of a sudden makes journalctl a lot more useful because, again, when I'm troubleshooting a system, most likely, I want to see the most recent entries in the log file, not those from a month ago.

I enter journalctl -r and it goes in reverse order, so my most current entries are listed first, and it starts working its way back as I page through the output. By the way, I'm just pressing the space bar to move down a page at a time. Press q to get out.

Use -f Option 2:21-3:23

Remember, there's also an option, -f, that you can use. I love the -f option. What this basically does is pull up the most recent entries in the journal and displays them onscreen, but it doesn't exit out.

And as new entries are added to the journal--as things happen on the system --those new entries are added to the bottom of the output. In the old days, we did this using the tail -f command to view the var log messages filed. It's an excellent troubleshooting tool.

We can accomplish the same thing using journalctl. We enter journalctl -f, and notice that the most recent entries from the journal are displayed, and then it continues listening for new entries to be added to the file.

Let's go ahead over here and open up a new terminal window, and let's create some journal entries. I'm first going to switch to my root user account. Notice that as I did, new entries were added down here to the journal.

Restart a Service 3:24-4:08

Let's restart a service, systemctl restart, and we have the mysql database service running on the system. Let's restart mysqld.

Notice down here that as the service is stopped and restarted, entries were continuously added to the journal and we could view them here.

If I'm having problems on the system and I'm trying to troubleshoot what's going on, I can open up one terminal window and run journalctl -f in one and then do my troubleshooting over here, and then I can view the log messages as they're being added to the journal.

Use -b Option 4:09-5:42

Another neat feature with journald is the fact that you can use it to view your system boot messages in addition to just your standard system log messages. This can be useful if you're trying to troubleshoot boot issues.

To do this, you enter journalctl but this time enter -b. When I do, the boot messages from the most recent system boot are displayed. You can see the most recent boot was at 12:33 in the afternoon today.

You can also use the journalctl command to view messages from previous system boots as well. You do this by specifying a number with the -b option. For example, if I wanted to display messages that were created during the first boot found at the very beginning of the journal, then I enter 1 here.

As you can see, the first boot events recorded in the journal occurred on August 19th; I believe that's when I installed this system. Here you can see those boot events. Enter q to get out. We can go the other direction as well. Instead of searching from the beginning of the journal, we can also start at the end of the journal and work our way backwards.

Specifying -b and then a negative number, such as -2 will look up the messages from the specified system boot starting from the end of the journal. For example, entering journalctl -b -2, we see the system messages that were created two boots ago on September 9th. Go ahead and get out of here.

Use -u Option 5:43-6:24

The journalctl command can also be used to display only the log entries in the journal that are related to a specific service running on the system. The syntax is to enter -journalctl -u followed by the name of the service.

As I said earlier, this system has the mysql database service running on it. We can use journalctl to view just those messages related to the mysqld daemon by entering -u mysqld. Here, we see just those entries that are somehow related to just the mysql database running on this system.

Configure the journald Daemon 6:25-8:08

The behavior of the journald daemon is configured using a configuration file in the /etc directory. It's in /etc/systemd. Do an ls command, we should see the journald.comp file.

If you want to customize the way that journald daemon works, this is the file that you edit. Let's go ahead and load it in the vi editor.

There are a lot of different parameters that you can configure in this file. Some of the more useful ones in my opinion are first of all, MaxFileSec. This specifies the maximum amount of time to store entries in the journal file before you rotate and start a new journal file.

Another one is MaxRetentionSec. This specifies the amount of time to store journal entries. Any entries older than the specified time will be automatically deleted from the journal file.

Another one is this one right here, MaxLevelStore. This controls the maximum log messages stored in the journal file. Basically, all messages that are equal to or less than the log file specified will be stored. Any messages created above the specified log level will be dropped. By default, this is set to debug.

That means that all log messages will be stored. debug is the highest level and it goes down through info, notice, warning, error, crit, alert, and emerge. If you wanted to reduce the number of log messages that are actually stored, you could set this to a value of say, error, then all warning, notice and info messages would not be stored. They would be dropped.

Summary 8:09-8:20

That's it for this demonstration. In this demo, we talked about working with journald. We first looked at the location where the journal is stored in the file system. We then looked at using the journalctl command to view the journal, and then we ended this demonstration by talking about how to configure the journald daemon.