Oracle Linux 8

Collecting and Analyzing Metrics With Performance Co-Pilot

G23060-03
October 2025

Oracle Linux 8 Collecting and Analyzing Metrics With Performance Co-Pilot,

G23060-03

Contents

Preface

Oracle Linux 8: Collecting and Analyzing Metrics With Performance Co-Pilot describes how to install and use Performance Co-Pilot (PCP) to collect performance metrics that can be used to monitor and diagnose system and network performance issues on Oracle Linux servers.

1

About Performance Co-Pilot

Performance Co-Pilot (PCP) collects OS and network metrics that you can use to diagnose performance issues. PCP provides a monitor host that you can use to send requests for metrics and logs to a pair of collector host services that are installed on each Oracle Linux system that you monitor.

2

Installing and Starting PCP

Install the `pcp` package, and supplementary tools, on Oracle Linux 8.

Before installing the `pcp-oracle-conf` package, enable the `ol8_addons` yum repository. For more information, see Oracle Linux: Managing Software on Oracle Linux.

1. Install the `pcp-oracle-conf`, `pcp`, `pcp-system-tools`, and `pcp-gui` packages by using the `dnf` command:

   ```
   sudo dnf install pcp-oracle-conf pcp-system-tools pcp-gui
   ```

2. Enable and start the Performance Metrics Collector Daemon (`pmcd`) and Performance Metrics Logger (`pmlogger`) collector host services:

   ```
   sudo systemctl enable --now pmcd pmlogger
   ```

The `pcp` package and supplementary tools are installed. The `pmcd` and `pmlogger` are also running.

3 Stopping PCP

Stop the `pmcd` and `pmlogger` services to temporarily or permanently halt data collection.

Before halting or disabling the Performance Metrics Collector Daemon (`pmcd`) and Performance Metrics Logger (`pmlogger`) collector host services, verify that they're running:

```
sudo systemctl status pmcd pmlogger
```

1. To temporarily halt data collection, stop the `pmcd` and `pmlogger` services:

   ```
   sudo systemctl stop pmcd pmlogger
   ```

2. To halt data collection for an indefinite period and ensure that they don't start again automatically when the system boots, fully disable them:

   ```
   sudo systemctl disable --now pmcd pmlogger
   ```

The `pmcd` and `pmlogger` services have halted data collection either temporarily or permanently depending on the commands used.

> **Note**
>
> For more information about masking and unmasking services to prevent scripts from restarting disabled system services, see Oracle Linux 8: Managing the System With systemd.

4

Getting Started With PCP

Review information gathered by the PCP host services and configuring the frequency with which metrics are collected.

If the `pcp-oracle-conf` package is installed then the only metrics collected by the `pmlogger` service are those listed in the `/var/lib/pcp/config/pmlogger/config.ora` configuration file.

If PCP has been installed without the `pcp-oracle-conf` package, review the `/var/lib/pcp/config/pmlogger/config.default` configuration file instead.

1. You can change the frequency with which metrics are collected in the same configuration file. For example, to increase the frequency from each minute to every 5 seconds, revise the file as follows:

   ```
   ...
   # It is safe to make additions from here on ...
   #

   log mandatory on every 5 seconds {
     filesys.free
     filesys.used
     ...
   }
   ```

   All the archives that the `pmlogger` service generates are stored in the `/var/log/pcp/pmlogger/`*`hostname`* directory. For more information, see the `pmlogconf(1)` manual page.

2. To verify the PCP configuration at the time that `pmlogger` collected specific performance metrics, use the `pcp` command:

   ```
   sudo pcp -a 20250113.0.xz
   ```

The frequency with which metrics are gathered was adjusted and verified.

5

PCP Command Reference

This table provides information about PCP commands.

Action	Command	Description
Performance metrics reporting tool.	`pmrep`	Generates reports from data collected by the Performance Metrics Collector Daemon (`pmcd`) service.
Arbitrary performance metrics valude dumper.	`pmval`	Outputs the current or archived values for a specific performance metric from the `pmcd` service.
Review detailed information about a specific performance metric.	`pminfo`	Outputs the current or archived values for a specific performance metric from the `pmcd` service.
High level system performance overview.	`pmstat`	Outputs a one line system performance overview on a timed interval. By default, a new line is output every five seconds.
Compared archived logs and reports differences.	`pmdiff`	Outputs the differences from two archived logs created by the Performance Metrics Logger (`pmlogger`) service.
Output collected or live performance data as plain text.	`pmdumptext`	Generates an ASCII format plain text file containing information collected by the `pmcd` service.
Output internal information for an existing log archive.	`pmdumplog`	Outputs metadata for archived data collected by the `pmlogger` service.
Run `dstat` commands within PCP.	`pcp dstat`	Provides compatibility for legacy scripts and troubleshooting procedures that rely on the deprecated `dstat` utility.

Reviewing Live Performance Metrics in Real Time

Use the `pmrep` and `pmval` commands to review live performance metrics.

To monitor all the outgoing metrics from the `eth0` network interface in real time, use the `pmrep` command:

```
sudo pmrep -i eth0 -v network.interface.out
```

To monitor live hard drive operations for each partition with a two second interval, use the `pmval` command:

```
sudo pmval -t 2sec -f 3 disk.partitions.write
```

Reviewing Recorded Performance Metrics

Use the `pmdumptext`, `pmstat`, and `pmdiff` commands to review metrics collected by the `pmlogger` service.

All the archives that the `pmlogger` service generates are stored in the `/var/log/pcp/pmlogger/`*`hostname`* directory. Navigate to this directory to review the archived performance metrics.

To review the data for specific performance metrics within a specified time span, use the `pmdumptext` command. For example, to review resource usage metrics for CPU load, memory usage, and disk write operations between 13:00 and 14:00 on a specific date:

```
sudo pmdumptext -Xlimu -t 10m -S @13:00 -T @14:00 'kernel.all.load[1]'
'mem.util.used' 'disk.partitions.write' -a 20250113.0.xz
```

The `pmstat` command can provide system performance metrics in a format similar to that produced by the `sar` command. For example, to review performance metrics averaged over a 10 minute interval between 09:00 and 10:00 on a specific date:

```
sudo pmstat -t 10m -S @09:00 -T @10:00 -a 20250113.0.xz
```

To compare the metrics between two time periods, use the `pmdiff` command. For example, to compare the metrics between 02:00 and 03:00 on one day to the metrics between 09:00 and 10:00 on a different day:

```
sudo pmdiff -S @02:00 -T @03:00 -B @09:00 -E @10:00 20250114.0.xz
20250113.0.xz
```

Reviewing Details About Recorded Performance Metrics

Use the `pminfo` command to review detailed information about specific performance metrics.

To review detailed information about a specific metric, use the `pminfo` command. For example, to review details about free memory:

```
sudo pminfo -df mem.freemem -a 20250113.0.xz
```

Validating System Status When Performance Metrics Were Captured

Use the `pmdumplog` and `pminfo` commands to validate system status when performance metrics were captured.

To verify the host, timezone, and time period that an archive containing performance metrics contains, use the `pmdumplog` command:

```
sudo pmdumplog -L 20250113.0.xz
```

To review a list of every enabled performance metric, use the `pminfo` command:

```
sudo pminfo -a 20250113.0.xz
```

Running `dstat` With PCP

Use the `pcp dstat` command to review performance metrics collected by PCP.

The `dstat` utility that was provided in previous Oracle Linux releases is no longer being actively developed. Instead, Performance Co-Pilot (PCP) provides many of the same functions for diagnosing system performance problems.

To review the command options provided for the `pcp dstat` command, run the following command:

```
pcp dstat -h
```

```
Usage: pcp-dstat [-afv] [options...] [delay [count]]
Versatile tool for generating system resource statistics

Dstat options:
-c, --cpu             enable cpu stats
-C 0,3,total          include cpu0, cpu3 and total
-d, --disk            enable disk stats
-D total,sda          include sda and total
...
```

By default, running the command without any other options displays statistics about CPU, disk, network, page, and system:

```
pcp dstat
```

```
You did not select any stats, using -cdngy by default.
----total-usage---- -dsk/total- -net/total- ---paging-- ---system--
usr sys idl wai stl| read  writ| recv  send|  in   out | int   csw
0   0 100   0   0|   0     0 | 198B  719B|   0     0 | 156   254
0   0 100   0   0|   0    12k|  66B  302B|   0     0 | 160   264
0   0  99   0   0|   0     0 | 132B  384B|   0     0 | 136   219
...
```

As with the previous iteration of the tool, `pcp dstat` generates a running list of metrics and statistics in real time. To stop the process, press the `Ctrl` + `C` keys.

You can use different options to narrow the information output from the `pcp dstat` command command. For example, to display the metrics for CPU 1 only, run the following command:

```
pcp dstat -c -C 1,total
```

```
-----cpu1-usage---------total-usage----
usr sys idl wai stl:usr sys idl wai stl
0   0 100   0   0:  0   0 100   0   0
1   0 100   0   0:  0   0  99   0   0
0   0 100   0   0:  0   0 100   0   0
...
```

Similarly, to display only network statistics of a specific interface, such as `ens3`, and including totals, you would run the following command:

```
pcp dstat -n -N ens3,total
```

```
--net/ens3---net/total-
recv  send: recv  send
66B  350B:  66B  350B
66B  190B:  66B  190B
66B  198B:  66B  198B
66B  198B:  66B  198B
...
```

To store any statistics that are being gathered into a file for later review, include the `-o` *outputfile* option in the command.

For example, to collect network statistics and save the information in the `/tmp/netstat-log` file, and use the `-f` option to output full information, run the following command:

```
pcp dstat -n -f -o /tmp/netstat-log
```

```
--net/ens3-----net/lo--
recv  send: recv  send
66B  358B:   0     0
66B  174B:   0     0
66B  190B:   0     0
341B  419B:   0     0
66B  190B:   0     0
66B  190B:   0     0
66B  190B:   0     0
```

The output is saved in plain-text format. To review the contents of the `/tmp/netstat-log` file, run the following command:

```
cat /tmp/netstat-log
```

```
...
"Host:","hostname",,,,"User:","user"
```

```
"Cmdline:","pcp-dstat -n -f -o /tmp/netstat-log",,,,"Date:","date"
"net/ens3",,"net/lo",
"net/ens3:recv","net/ens3:send","net/lo:recv","net/lo:send"
65.934,357.641,0,0
66.000,173.999,0,0
66.000,190.001,0,0
340.992,418.991,0,0
66.001,190.004,0,0
66,190,0,0
66.000,189.999,0,0
```