

**NAME**

hcpcs-health-check – end-to-end health check workflow for HCP Cloud Scale support bundles

**SYNOPSIS**

```
expand_hcpcs_support.sh
cd
<expanded-directory>
sudo gsc_prometheus.sh
expand_hcpcs_support.sh --healthcheck-only -u -p
<PORT>
runchk.sh
[-f
healthcheck.conf]
[--full-detail]
[--no-metrics]
```

**DESCRIPTION**

This manual page describes the complete five-step workflow for analysing an HCP Cloud Scale (HCP CS) support bundle. Each step is performed by a dedicated script; this page explains how they connect and what each produces.

The workflow assumes a support bundle has already been delivered, typically as one or more *supportLogs\_\*.tar.xz* archives, with one or more Prometheus snapshot archives (*psnap\_\*.tar.xz* or *\*Prometheus\*.tar.xz*) alongside them.

**WORKFLOW****Step 1 — Expand the support bundle**

`expand_hcpcs_support.sh` unpacks all *supportLogs\_\*.tar.xz* files found under the root directory, normalises any *\*Prometheus\*.tar.xz* filenames to the *psnap\_YYYY-Mon-DD\_HH-MM-SS.tar.xz* convention, and writes an initial *healthcheck.conf* for each psnap discovered.

```
expand_hcpcs_support.sh -r /ci/05304447
```

After this step the directory tree looks similar to:

```
/ci/05304447/
  05304447.supportLogs_2025-11-26_20-48-48_<node>.tar.xz
  05304447.cluster_triage_2025-11-26_20-48-48.tar.20251126.1320.xz
  2025-11-26_20-48-48/
    cluster_triage/
    collect_healthcheck_data/
    psnap_2025-Nov-26_20-48-48.tar.xz
    healthcheck.conf
```

**Step 2 — Change into the expanded directory**

Move into the timestamped directory that was just created so that `gsc_prometheus.sh` and `runchk.sh` can locate the `psnap` and `healthcheck.conf` files by relative path.

```
cd /ci/05304447/2025-11-26_20-48-48
```

**Step 3 — Start the Prometheus container**

`gsc_prometheus.sh` extracts the Prometheus snapshot archive into a working directory and starts a container (Docker or Podman, auto-detected) that serves that data. It scans ports **9090–9200** and selects the lowest free port, skipping ports already used by running containers and reserved exporter ports (9093, 9100, 8080, 9115, 9116, 9104).

The selected port is printed at the end of the run:

```
sudo gsc_prometheus.sh \
-s 05304447 \
-c AcmeCorp \
-f psnap_2025-Nov-26_20-48-48.tar.xz \
```

```
-b /opt/prom_instances
```

Sample output (final line):

```
[ OK ] Prometheus for AcmeCorp/05304447 started on port 9092.
```

Note the port number; it is required for Step 4.

#### Step 4 — Update healthcheck.conf with the Prometheus port

`expand_hpcs_support.sh --healthcheck-only -u` patches the existing `healthcheck.conf` in place, writing only the fields supplied on the command line. The `-u` (`--update`) flag preserves all other fields (time-stamp, CS version, install directory) unchanged.

Replace `PORT` with the port printed in Step 3:

```
expand_hpcs_support.sh --healthcheck-only -u -p 9092
```

To also set the Prometheus server address when it is not localhost:

```
expand_hpcs_support.sh --healthcheck-only -u -p 9092 -s 192.0.2.10
```

#### Step 5 — Run the health check suite

`runchk.sh` reads `healthcheck.conf` (supplied via `-f` or as a positional argument), runs every check script in sequence, and aggregates all **WARNING**, **ERROR**, and **CRITICAL** lines from the resulting `health_report_*.log` files. A total issue count is printed at the end.

By default the three data-intensive checks (disk performance, filesystem, journal messages) are skipped. Pass `--full-detail` to include them. Pass `--no-metrics` to skip the Prometheus query suite when no container is running.

```
# Core checks only
runchk.sh -f ./healthcheck.conf

# Include disk, filesystem, and journal analysis
runchk.sh -f ./healthcheck.conf --full-detail

# Core checks, Prometheus not yet started
runchk.sh --no-metrics
```

### COMPLETE WORKED EXAMPLE

```
# Step 1 — expand the bundle
expand_hpcs_support.sh -r /ci/05304447

# Step 2 — enter the expanded directory
cd /ci/05304447/2025-11-26_20-48-48

# Step 3 — start Prometheus (note port in final OK line)
sudo gsc_prometheus.sh \
-s 05304447 \
-c AcmeCorp \
-f psnap_2025-Nov-26_20-48-48.tar.xz \
-b /opt/prom_instances
#[ OK ] Prometheus for AcmeCorp/05304447 started on port 9092.

# Step 4 — patch healthcheck.conf with the port
expand_hpcs_support.sh --healthcheck-only -u -p 9092

# Step 5 — run health checks (full detail)
runchk.sh -f ./healthcheck.conf --full-detail
```

## FILES

### *healthcheck.conf*

Prometheus connection parameters consumed by **runchk.sh**. Written by **expand\_hcpes\_support.sh** (Step 1) and updated in Step 4. Contains *\_prom\_server*, *\_prom\_port*, *\_prom\_time\_stamp*, *\_cs\_version*, and the **PROM\_CMD\_PARAM\_HOURLY** / **PROM\_CMD\_PARAM\_DAILY** query strings.

### *health\_report\_\*.log*

Per-check output files written by each *chk\_\*.sh* script and *print\_node\_memory\_summary.sh*. Scanned by **runchk.sh** for **WARNING/ERROR/CRITICAL** lines at the end of Step 5.

### *messages\_warn.log*

Journal WARNING lines written by **chk\_messages.sh**. Not included in the aggregation grep; provided for manual review.

### *psnap\_YYYY-Mon-DD\_HH-MM-SS.tar.xz*

Prometheus snapshot archive extracted in Step 3.

## SEE ALSO

**expand\_hcpes\_support(1)**, **gsc\_prometheus(1)**, **runchk(1)**

## NOTES

- **gsc\_prometheus.sh** requires root (or equivalent container runtime privileges). Steps 1, 4, and 5 do not require elevated privileges.
- When multiple psnap files exist in the same SupportLog directory each gets its own timestamped config (*healthcheck.conf-YYYY-MM-DDTHH:MM:SS*). Repeat Steps 3–5 for each config file.
- The Prometheus container is started with **--rm** by default and is removed when stopped. Pass **--keep-container** in Step 3 to retain it across restarts.

## AUTHORS

Hitachi Vantara GSC