NAME

runchk.sh − run the HCP Cloud Scale health check suite

SYNOPSIS

runchk.sh [**−f** *healthcheck.conf*] [**−−full−detail**] [**−−no−metrics**]

DESCRIPTION

runchk.sh is the top-level orchestrator for the HCP Cloud Scale health check workflow (Step 5). It loads a *healthcheck.conf* configuration file, then runs every check and reporting script in a fixed sequence.

Three data-intensive checks are disabled by default and must be explicitly enabled with **−−full−detail**: **chk_disk_perf.sh**, **chk_filesystem.sh**, and **chk_messages.sh**.

The Prometheus query suite (**chk_metrics.sh**) can be skipped with **−−no−metrics** when no Prometheus container is running or the port is unknown.

At the end of the run a summary of all **WARNING**, **ERROR**, and **CRITICAL** lines found across every *health_report_*.log* file is printed together with a total issue count and elapsed wall-clock time.

If *healthcheck.conf* is absent or cannot be read, Prometheus-dependent checks (metrics, services) are skipped with a warning; all other checks continue normally.

OPTIONS

−f *FILE*
: Path to the Prometheus connection parameter file produced by **expand_hcpcs_support.sh**. Defaults to *healthcheck.conf* in the current directory. A bare positional argument is also accepted for backward compatibility.

−−full−detail
: Enable the three checks that are skipped by default:

  chk_disk_perf.sh
  : iostat device utilisation, latency, and queue depth per node.

  chk_filesystem.sh
  : df filesystem usage, lsblk block device layout, LVM PV/VG health.

  chk_messages.sh
  : journald error/warning analysis (last 30 days, deduplicated per day).

−−no−metrics
: Skip **chk_metrics.sh**. Use when no Prometheus container is available.

−h, **−−help**
: Print a usage summary and exit.

HEALTHCHECK.CONF KEYS

The config file is sourced as a shell script. The following variables are consumed by **runchk.sh** and the scripts it calls:

_prom_server
: Prometheus server hostname or IP address.

_prom_port
: Prometheus server port.

_cs_version
: HCP CS product version string (e.g. *2.6.0*).

_version_num
: Version string consumed by **chk_services_sh.sh**.

_prom_cmd_param_daily
: Extra arguments forwarded verbatim to **chk_metrics.sh**.

SCRIPTS EXECUTED (IN ORDER)

Scripts marked **[full−detail]** run only when **−−full−detail** is given. The script marked **[metrics]** is skipped when **−−no−metrics** is given.

selfcheck.sh
Verifies the integrity of the installed toolkit against *SHA256SUMS*.

print_cluster_identity_summary.sh
Prints cluster identity information extracted from the support bundle.

print_node_memory_summary.sh
Reports per-node memory layout (lsmem) and memory pressure (free).

print_node_os_summary.sh
Summarises the operating system release across all nodes.

chk_cluster.sh
Checks cluster-level health indicators.

prep_partitions_json.sh
Builds the partition event JSON consumed by later checks.

get_partition_tool_info.sh
Extracts information about the partition management tool version.

chk_partInfo.sh −d .
Analyses partition events and growth trends.

parse_instances_info.sh
Parses service instance placement data.

prep_services_instances.sh
Prepares per-service instance configuration for subsequent checks.

chk_service_placement.sh
Validates service placement rules and counts.

chk_lshw.sh −d .
Checks hardware configuration uniformity across nodes.

chk_chrony.sh −d .
Checks NTP synchronisation status on each node.

chk_top.sh −d .
Analyses CPU and load-average data from *top* batch output.

chk_docker.sh −d .
Checks Docker engine version, containers, inotify limits, and ulimits.

chk_disk_perf.sh −d . [full−detail]
Analyses iostat extended output; reports high utilisation, latency, and queue depth per device and node.

chk_filesystem.sh −d . [full−detail]
Checks filesystem usage (df), block device layout (lsblk), and LVM PV/VG health.

chk_messages.sh −d . [full−detail]
Parses journald output (last 30 days); writes error summaries to *health_report_messages.log* and warnings to *messages_warn.log*.

chk_alerts.sh
Checks pre-collected alert data from the support bundle.

chk_services_sh.sh −r *VERSION*
Runs per-service health scripts for the detected HCP CS version.

chk_snodes.sh
Checks storage node status.

chk_services_memory.sh
Compares per-service RSS memory against bounds in *memcheck.conf* .

chk_metrics.sh [metrics]
Queries Prometheus for each metric in *hcpcs_daily_alerts.json* and reports threshold violations.

OUTPUT

Each check script writes findings to a dedicated *health_report_*.log* file. After all scripts complete, **runchk.sh** aggregates every **WARNING/ERROR/CRITICAL** line from those files and prints them with a total count.

messages_warn.log (written by **chk_messages.sh**) contains raw journal WARNING lines for manual review and is *not* included in the aggregation grep.

EXIT STATUS

runchk.sh exits with the status of the last command executed. To detect issues examine the aggregated output or the individual *health_report_*.log* files.

ENVIRONMENT

GSC_LIB_PATH
Override the path to *gsc_core.sh*. Default: same directory as the script.

FILES

healthcheck.conf
Prometheus connection parameters written by **expand_hcpcs_support.sh** and updated in Step 4.

health_report_*.log
Per-check output files scanned for **WARNING/ERROR/CRITICAL** at the end of the run.

messages_warn.log
Journal WARNING lines; manual review only.

memcheck.conf
Per-service memory bounds for **chk_services_memory.sh**.

hcpcs_daily_alerts.json
Prometheus metric definitions for **chk_metrics.sh**.

SHA256SUMS
Integrity manifest verified by **selfcheck.sh**.

EXAMPLES

Standard run â core checks only

```
runchk.sh -f ./healthcheck.conf
```

Include disk, filesystem, and journal analysis

```
runchk.sh -f ./healthcheck.conf --full-detail
```

Core checks, skip Prometheus metrics

```
runchk.sh --no-metrics
```

Full detail, no metrics (Prometheus not yet started)

```
runchk.sh -f ./healthcheck.conf --full-detail --no-metrics
```

Full five-step workflow

```
expand_hcpcs_support.sh -r /ci/05304447
cd /ci/05304447/2025-11-26_20-48-48
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.
```

```
expand_hcpcs_support.sh --healthcheck-only -u -p 9092
runchk.sh -f ./healthcheck.conf --full-detail
```

SEE ALSO

expand_hcpcs_support(1), **gsc_prometheus**(1), **hcpcs−health−check**(7)

AUTHORS

Hitachi Vantara GSC