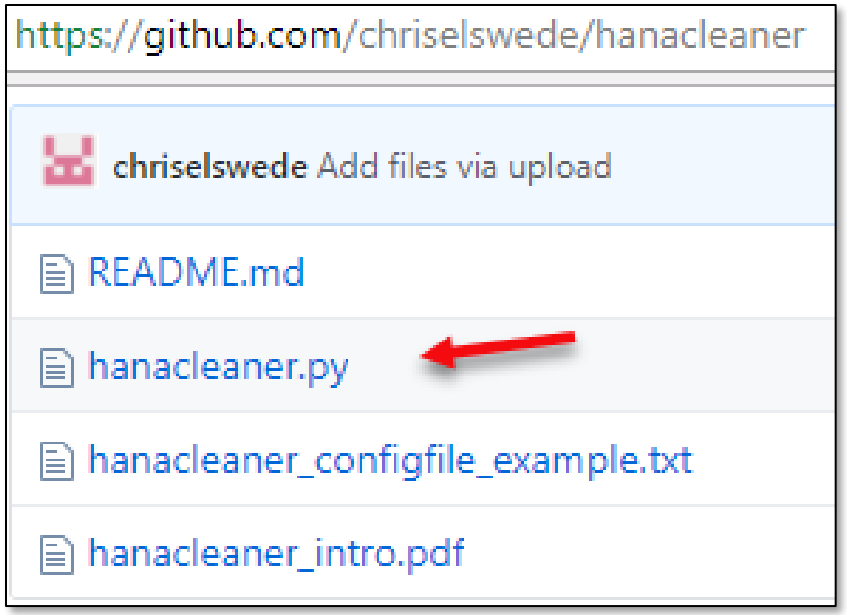
House Keeping

HANACleaner – SAP Note 2399996

SAP Note 2399996 presents a tool that can help with housekeeping tasks

2399996 - How-To: Configuring automatic SAP HANA Cleanup with SAP HANACleaner

- It is a python script to be downloaded from https://github.com/chriselswede/hanacleaner
- It is intended to be executed as <sid>adm on your SAP HANA Server
  (since then the proper python version is already in your path,
  installed together with hana)
- It connects via host, port and DB user, provided in hdbuserstore
- That DB user needs proper privileges



For more about the SAP HANACleaner see SAP Note 2399996
SAP Note 2400024 provides administration suggestions, e.g. recommendations about the hanacleaner

House Keeping

HANACleaner – using hdbuserstore

Host, port and DB user needs to be provided in the hdbuserstore:

```
mo-fc8d991e0:~> hdbuserstore SET HANACLEANER1KEY mo-fc8d991e0:30015 HANACLEANER1 PassWord1
mo-fc8d991e0:~> hdbuserstore LIST
DATA FILE       : /usr/sap/CH0/home/.hdb/mo-fc8d991e0/SSFS_HDB.DAT
KEY FILE        : /usr/sap/CH0/home/.hdb/mo-fc8d991e0/SSFS_HDB.KEY

KEY HANACLEANER1KEY
  ENV : mo-fc8d991e0:30015
  USER: HANACLEANER1
```

Then the hanacleaner can connect using the info stored in hdbuserstore:

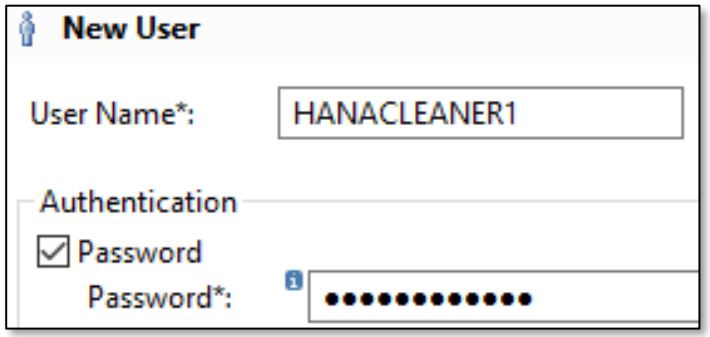
```
mo-fc8d991e0:/tmp/HANACleaner> whoami
ch0adm
mo-fc8d991e0:/tmp/HANACleaner> python hanacleaner.py -k HANACLEANER1KEY -be 20
The most used filesystem is using
21 %
In total 0 data backup entries were removed from the backup catalog
```

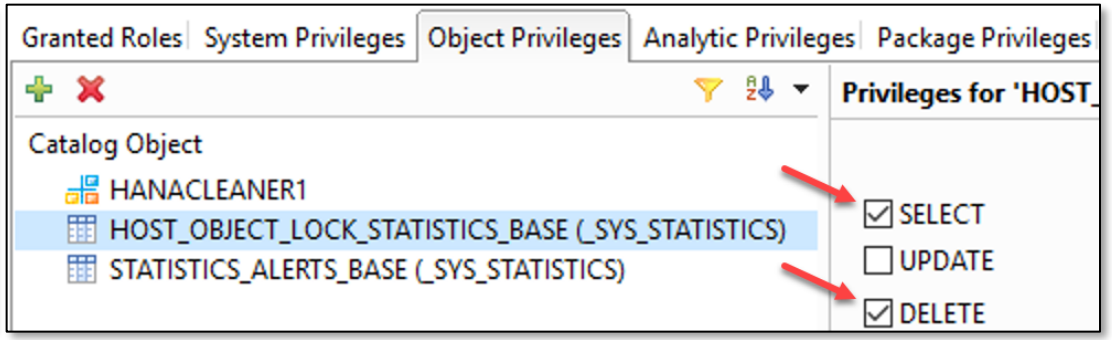
House Keeping

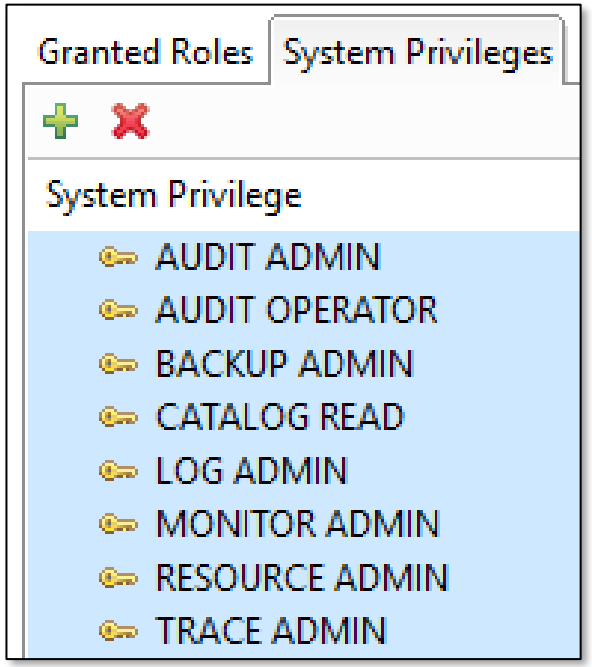
HANACleaner – needs privileges

The DB user that hanacleaner uses to connect needs proper privileges

Depending on what housekeeping tasks the specific hanacleaner user will do, he needs specific sets of privileges, for example (note: sometimes also ALTER on specific schemas are needed and/or DATA ADMIN):







As an example, the HANACLEANER user could be created like this:

```
CREATE USER HANACLEANER PASSWORD Pass1234 NO FORCE_FIRST_PASSWORD_CHANGE;
ALTER USER HANACLEANER DISABLE PASSWORD LIFETIME;
GRANT SELECT, DELETE ON _SYS_STATISTICS.HOST_OBJECT_LOCK_STATISTICS_BASE TO HANACLEANER;
GRANT SELECT, DELETE ON _SYS_STATISTICS.STATISTICS_ALERTS_BASE TO HANACLEANER;
GRANT SELECT, DELETE ON _SYS_REPO.OBJECT_HISTORY TO HANACLEANER;
GRANT SELECT ON _SYS_STATISTICS.STATISTICS_EMAIL_PROCESSING TO HANACLEANER;
GRANT audit admin , audit operator, backup admin, catalog read, log admin, monitor admin, resource admin, trace admin to HANACLEANER;
```

House Keeping

HANACleaner – tells missing privileges

If the DB user is missing privileges, hanacleaner will indicate that

E.g. here the user A2 is missing the system privilege CATALOG READ:

```
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -ct 300 -dt 300 -or true -k A2KEY
The most used filesystem is using
96 %
Cleaning of the backup catalog was not done since -rb and -rd were both negative (or not specified)

INSUFICIENT PRIVILEGE WARNING: It appears that there are no traces.
One possible reason for this is that the user represented by the key A2KEY has unsuficient privilege,
e.g. lacking the system privilege CATALOG READ.

0 trace files were removed
```

E.g. here the user A2 is missing the system privilege TRACE ADMIN:

```
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -ct 225 -or true -k A2KEY
The most used filesystem is using
96 %
Cleaning of the backup catalog was not done since -rb and -rd were both negative (or not
hdbsql -U A2KEY "ALTER SYSTEM CLEAR TRACES ('ALERT','CLIENT','CRASHDUMP','EMERGENCYDUMP',
 UNTIL '2016-07-15 00:00:00'"
* 258: insufficient privilege: Not authorized SQLSTATE: HY000

ERROR: The user represented by the key A2KEY could not clear traces.
One possible reason for this is unsuficient privilege,
e.g. lack of the system privilege TRACE ADMIN.
```

House Keeping

HANACleaner – backupcatalog cleanup (1/2)

For cleaning up the backup catalog (and possibly also backups) hanacleaner has the following input flags

Flag	Unit	Details	Explanation	Default
-be		minimum number of retained backup entries in the catalog	this number of entries of successful data backups will remain in the backup catalog	-1 (not used)
-bd	days	minimum retained days of backup entries in the catalog	the youngest successful data backup entry in the backup catalog that is older than this number of days is the oldest successful data backup entry not removed from the backup catalog	-1 (not used)
-bb	true/ false	switch to delete backups also	if set to true the backup files corresponding to the backup entries are also deteleted	false
-bo	true/ false	output the backup catalog	if set to true the backup catalog is printed before and after cleanup	false
-br	true/ false	output the deleted entries	if set to true the deleted backup entries are printed after the cleanup	false

Example:
Here backup catalog entries (i.e. not the backups themselves) older than 42 days are deleted, but at least 5 backup entries are kept, and the deleted backup entries are printed out

```
python hanacleaner.py -bd 42 -be 5 -br true
```

House Keeping

HANACleaner – backupcatalog cleanup (2/2)

Cleaning up the backup catalog can be done with the hanacleaner

Example:
Here backup catalog entries (i.e. not the backups themselves) older than 30 days are deleted, but at least 5 backup entries are kept, and the deleted backup entries are printed out:

```
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -bd 30 -be 5 -br true
The most used filesystem is using
96 %
********************
2017-02-28 19:38:13
********************
hdbsql -U SYSTEMKEY "BACKUP CATALOG DELETE ALL BEFORE BACKUP_ID 1485547216621"

REMOVED:
| ENTRY_ID             | ENTRY_TYPE_NAME      | BACKUP_ID            | SYS_START_TIME                | STATE_NAME |
|       1484942410880 | complete data backup |       1484942410880 | 2017-01-20 21:00:10.880000000 | successful |


In total 1 data backup entries were removed from the backup catalog
```

House Keeping

HANACleaner – trace cleanup (1/3)

For cleaning up the traces hanacleaner has the following input flags

Flag	Unit	Details	Explanation	Default
-tc	days	retention days for trace files	trace files with their latest modification time older than these number of days are removed from all hosts, conceptually -tc is the same as -tf, but -tc is using ALTER SYSTEM CLEAR TRACES (see SQL. Ref.) Note: see also –tcd for trace archiving	-1 (not used)
-tb	days	retention days for backup.log and backint.log	same as for -tc, but only for backup.log and backint.log	-1 (not used)
-te	days	retention days for expensive statement files	same as for -tc, but only for expensive statement files, only use if you read and understood SAP Note 2819941 (should probably only be used if use_in_memory_tracking = false)	-1 (not used)
-tf	days	retention days for trace files	trace files, in all hosts, that are older than this number of days are removed (except for the currently opened trace files), only files with these extensions are taken into account: .trc, .log, .stat, .py, .tpt, .gz, .zip, .old, .xml, .txt, .docs, .cfg, .dmp, .cockpit, .xs and there is a list of files that are ignored (to add your own files to this ignore list, see -ti): kill.sap, backup.log, backint.log, hdbdaemon.status, sapstart.sem, sapstart.log, .sap<SID>_HDB<dbinstance>, http_fe.log, lss/, nameserver_suschksrv.trc Conceptually -tf is the same as -tc, but -tf is using ALTER SYSTEM REMOVE TRACES (see SQL. Ref.)	-1 (not used)
-ti	files	trace file removal ignore list	List of files that should not be removed with –tf (wildcard(s) with * is possible)	
-to	true/false	output trace files	displays trace files before and after the cleanup	false
-td	true/false	output the deleted trace files	displays the trace files that were deleted	false

Example:
Here trace file older than 42 days are removed in two different ways:

```
python hanacleaner.py -tc 42 -tf 42
```

House Keeping

HANACleaner – trace cleanup (2/3)

Cleaning of traces can be done with hanacleaner as in this example

Example:
Here trace files older than 200 days are deleted and the removed trace files are displayed:

```
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -tc 200 -tf 200 -td true
The most used filesystem is using
96 %
**********
2017-02-28 19:52:42
**********
    (Cleaning of the backup catalog was not done since -be and -bd were both negative
hdbsql -U SYSTEMKEY "ALTER SYSTEM CLEAR TRACES ('ALERT','CLIENT','CRASHDUMP','EMERGENC

REMOVED (1):
ls80010 | indexserver_ls80010.30003.executed_statements.000.trc


1 trace files were removed
```

House Keeping

HANACleaner – trace cleanup (3/3)

Cleaning of traces can be done with hanacleaner as in these examples

Example:
Here trace files older than 5 days are deleted, but some indexserver and scriptserver trace files are kept, and the removed trace files are displayed:

```
xscadm@atgvmls866:/tmp/HANACleaner> python hanacleaner.py -k TlKEY -tf 5 -ti indexserver*.30003.*.trc,scriptserver*.trc -td true
Will now check most used memory in the file systems. If it hangs, there is an issue with df -h, then see if the -fs flag helps
```

```
REMOVED (9):
atgvmls866 | indexserver_atgvmls866.30003.hdbcons.trc_backup
atgvmls866 | xsengine_atgvmls866.30007.271.trc
atgvmls866 | xsengine_atgvmls866.30007.stat
atgvmls866 | xsengine_alert_atgvmls866.trc
atgvmls866 | indexserver_atgvmls866.30003.stat
atgvmls866 | scriptserver_atgvmls866.30004.stat
atgvmls866 | DEVELOPMENT_ADMINISTRATOR_sqltrace_atgvmls866_30003_000.py
atgvmls866 | indexserver_atgvmls866.30003_TESTAUSER.000.trc
atgvmls866 | rsutil_atgvmls866.30003.000.trc


9 trace files were removed (-tc and -tf)
```

Example:
Here backup.log and backint.log files older than 1 day are deleted, and the removed trace files are displayed:

```
xscadm@atgvmls866:/tmp/HANACleaner> python hanacleaner.py -k TlKEY -tb 1 -td true
```

```
REMOVED (1):
atgvmls866 | backup.log


1 trace files were removed (-tc, -tb, -te and -tf)
```

House Keeping

HANACleaner – trace archiving (1/2)

For removing trace files "with backup" and then moving these backups to a different folder, hanacleaner has the following input flags

Flag	Unit	Details	Explanation	Default
-tcb	true/false	clear traces with backup	the trace files that are removed with the –tc, -tb and -te flags are backed up (as .gz) with ALTER SYSTEM CLEAR TRACES … WITH BACKUP (not applicable for -tf, since "WITH BACKUP" does not exist for ALTER SYSTEM REMOVE TRACES) Note: Some small, unnecessary files, like .stat (SAP Note 2370780) might be deleted, so number removed and archived file might seem incorrect.	false
-tbd		directory for the trace backups	full path of the directory to where the back up (see -tcb) of the trace files are moved	' ' (they stay)
-tmo	seconds	time out for the move	the move, requested by the -tbd flag, must wait until the compression, requested by the -tcb flag, is finished. This can take some seconds. If it is not finished before the time out, specified by the -tmo flag, the move will not happen (a warning will be logged).	30 seconds

Example:

Here all trace files older than 3 days are backed up and moved to /tmp/tracebackup → 1 .trc file is moved and one .stat file is deleted

```
hsiadm@atgvmls7071:/tmp/HANACleaner> python hanacleaner.py -k T1KEY -tc 3 -te 3 -tcb true -tbd /tmp/tracebackup -td true -to true -os true
```

```
ARCHIVED (1):
diserver_atgvmls7071.30025.159_20210726164903.gz
```

```
REMOVED (2):
atgvmls7071 | diserver_atgvmls7071.30025.159.trc
atgvmls7071 | diserver_atgvmls7071.30025.stat
```

```
2 trace files were removed
```

```
hsiadm@atgvmls7071:/tmp/HANACleaner> ls /tmp/tracebackup/diserver_atgvmls7071.30025.159_20210726164903.gz
/tmp/tracebackup/diserver_atgvmls7071.30025.159_20210726164903.gz
```

House Keeping

HANACleaner – trace archiving (2/2)

More Examples:

```
xscadm@atgvmls866:/tmp/HANACleaner> python hanacleaner.py -k T1KEY -tc 35 -te 35 -tcb true -tbd /tmp/tracebackup -td true -os true

ARCHIVED (1):
indexserver_atgvmls866.30003.903_20210726173040.gz

REMOVED (1):
atgvmls866 | indexserver_atgvmls866.30003.903.trc

xscadm@atgvmls866:/tmp/HANACleaner> ls /tmp/tracebackup/
indexserver_atgvmls866.30003.903_20210726173040.gz
```

Note: Archived trace files can be cleaned up with General File Cleanup, see -gr, -gw, -gd

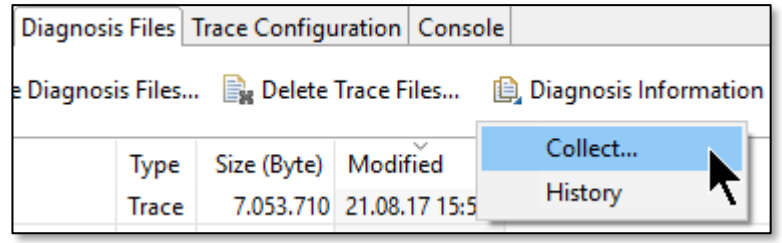
```
xscadm@atgvmls866:/tmp/HANACleaner> python hanacleaner.py -k T1KEY -tc 10 -te 10 -tcb true -tbd /tmp/tracebackup -td true

ARCHIVED (1):
diserver_alert_atgvmls866_20210726173803.gz

REMOVED (1):
atgvmls866 | diserver_alert_atgvmls866.trc


1 trace files were removed

xscadm@atgvmls866:/tmp/HANACleaner> ls /tmp/tracebackup/diserver_alert_atgvmls866_20210726173803.gz
/tmp/tracebackup/diserver_alert_atgvmls866_20210726173803.gz
```

House Keeping

HANACleaner – dump cleanup

Manually created dump files (a.k.a. rte or fullsystem dumps) can be deleted with the following flag

Flag	Unit	Details	Explanation	Default
-dr	days	retention days for dump files	manually created dump files (a.k.a. fullsytem dumps and runtime dumps) that are older than this number of days are removed	-1 (not used)



Example:
Here dump files older than 1 day are deleted

```
ch0adm@mo-fc8d991e0:/tmp/HANACleaner> cdglo
ch0adm@mo-fc8d991e0:/usr/sap/CH0/SYS/global> ll sapcontrol/snapshots/
total 28824
-rw-r--r-- 1 ch0adm sapsys  3173927 Aug 21 15:50 fullsysteminfodump_mo-fc8d991e0_CH0_2017_08_21_15_50_33.zip
-rw-r--r-- 1 ch0adm sapsys 26300975 Aug 23 17:32 fullsysteminfodump_mo-fc8d991e0_CH0_2017_08_23_17_32_02.zip
ch0adm@mo-fc8d991e0:/usr/sap/CH0/SYS/global> cd /tmp/HANACleaner/
ch0adm@mo-fc8d991e0:/tmp/HANACleaner> python hanacleaner.py -dr 1
1 fullsysteminfodump zip files (that can contain both fullsystem dumps and runtime dumps) were removed
ch0adm@mo-fc8d991e0:/tmp/HANACleaner> cdglo
ch0adm@mo-fc8d991e0:/usr/sap/CH0/SYS/global> ll sapcontrol/snapshots/
total 25720
-rw-r--r-- 1 ch0adm sapsys 26300975 Aug 23 17:32 fullsysteminfodump_mo-fc8d991e0_CH0_2017_08_23_17_32_02.zip
ch0adm@mo-fc8d991e0:/usr/sap/CH0/SYS/global>
```

House Keeping

HANACleaner – hdbcons log cleanup

Since .hdbcons.trc files are still (as of Jan 2025) not being file-rotated (see SAP Note 3051846 for latest update) hanacleaner must solve this by removing lines inside the file

Flag	Unit	Details	Explanation	Default
-hr	days	retention days for lines in the hdbcons trace file	unfortunately there is no file rotation for *.hdbcons.trc files (see SAP Note 3051846), therefore -hr can be used to define retention days for lines in these files. WARNING: this might temporarily require up to twice the file's memory usage! If the file is larger than 10 GB hanacleaner will print a warning and then ignore that file.	-1 (not used)

Example: Here lines older than 200 days are removed from the .hdbcons.trc files

```
xscadm@atgvmls866:/tmp/HANACleaner> more //usr/sap/XSC/HDB00//atgvmls866//trace//DB_XSC/indexserver_atgvmls866.30003.hdbcons.trc
COMMAND VIA_SQL IS_AUTHORIZED   START_TIME      EXECUTION_SUCCESS       END_TIME        DURATION
"context list -s"       FALSE   TRUE    2022-02-15 12:38:30.348 TRUE    2022-02-15 12:38:30.365 16.715 msec
"context list -s"       FALSE   TRUE    2022-02-15 12:40:32.569 TRUE    2022-02-15 12:40:32.586 16.541 msec
"connection c 261637"   FALSE   FALSE   2022-06-20 17:20:08.885 TRUE    2022-06-20 17:20:08.900 2.758 msec
"connection d 261637"   FALSE   FALSE   2022-06-20 17:32:17.570 TRUE    2022-06-20 17:32:17.577 7.579 msec
"tablepreload status -i -c"     FALSE   FALSE   2022-11-17 14:18:11.192 TRUE    2022-11-17 14:18:11.204 14.000 usec     FALSE   2022-11-17 14:18:11.20
"tablepreload status -i -c"     FALSE   FALSE   2022-11-17 14:22:04.619 TRUE    2022-11-17 14:22:04.619 11.000 usec     FALSE   2022-11-17 14:22:04.61
"tablepreload status -i -c"     FALSE   FALSE   2022-11-17 14:22:09.085 TRUE    2022-11-17 14:22:09.085 9.000 usec      FALSE   2022-11-17 14:22:09.08
"context list -s"       FALSE   FALSE   2023-01-21 19:17:49.328 TRUE    2023-01-21 19:17:49.369 40.886 msec     FALSE   2023-01-21 19:17:49.369 0
xscadm@atgvmls866:/tmp/HANACleaner> python hanacleaner.py -k TlKEY -hr 200
In total 4 rows where cleaned from hdbcons.trc files (-hr)
xscadm@atgvmls866:/tmp/HANACleaner> more //usr/sap/XSC/HDB00//atgvmls866//trace//DB_XSC/indexserver_atgvmls866.30003.hdbcons.trc
COMMAND VIA_SQL IS_AUTHORIZED   START_TIME      EXECUTION_SUCCESS       END_TIME        DURATION
"tablepreload status -i -c"     FALSE   FALSE   2022-11-17 14:18:11.192 TRUE    2022-11-17 14:18:11.204 14.000 usec     FALSE   2022-11-17 14:18:11.20
"tablepreload status -i -c"     FALSE   FALSE   2022-11-17 14:22:04.619 TRUE    2022-11-17 14:22:04.619 11.000 usec     FALSE   2022-11-17 14:22:04.61
"tablepreload status -i -c"     FALSE   FALSE   2022-11-17 14:22:09.085 TRUE    2022-11-17 14:22:09.085 9.000 usec      FALSE   2022-11-17 14:22:09.08
"context list -s"       FALSE   FALSE   2023-01-21 19:17:49.328 TRUE    2023-01-21 19:17:49.369 40.886 msec     FALSE   2023-01-21 19:17:49.369 0
xscadm@atgvmls866:/tmp/HANACleaner>
```

House Keeping

HANACleaner – General File Clean Up

Any folder with files including any word in their file names can be cleaned:

Flag	Unit	Details	Explanation	Default
-gr	list of days	retention days for any general file	files in the directory specified with -gd and with the file names including the word specified with -gw are only saved for this number of days Note: -gr must be a list with the same length as -gd and -gw	“” (not used)
-gd		directories	a comma separated list with full paths of directories with files to be deleted according to -gr (entries pairs with entries in -gw)	default "" (not used)
-gw		filename parts	a comma separated list with words that files should have in their names to be deleted according to -gr (entries pairs with entries in -gd)	default "" (not used)

Example:
Here files hana in their file names, in the folders /tmp/hanachecker_output/ & /tmp/hanasitter_output older than 10 resp. 20 days are deleted

```
xscadm@atgvmls7040:/tmp/HANACleaner> python hanacleaner.py -k TlKEY -gr 10,20 -gd /tmp/hanasitter_output/,/tmp/hanachecker_output -gw hana,hana -os true
Will now check most used memory in the file systems. If it hangs, there is an issue with df -h, then see if the -fs flag helps
find /tmp/hanasitter_output/ -maxdepth 1 -name '*hana*' -type f -mtime +10 -delete
find /tmp/hanachecker_output -maxdepth 1 -name '*hana*' -type f -mtime +20 -delete
12 general files were removed (-gr)
```

House Keeping

HANACleaner – alerts

For deleting old alerts from the alert table (filled by the statistics service) hanacleaner has the following input flags

Flag	Unit	Details	Explanation	Default
-ar	days	minimum number retained days of the alerts	minimum retained age of statistics server alerts	-1 (not used)
-ao	true/ false	output alerts	if true, then all alerts will be displayed before and after the cleanup (if number of alerts are more than 10 thousand, hanacleaner will not do this output)	false
-ad	true/ false	output deleted alerts	if true, then deleted alerts will be displaye after the cleanup (if number of alerts are more than 10 thousand, hanacleaner will not do this output)	false

Example:
Here alerts older than 5 days are removed from the statistics server alert table:

```
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -ar 5
The most used filesystem is using
96 %
**********
2017-02-28 21:24:18
**********
1701680 alerts were removed
```

House Keeping

HANACleaner – log segments

For reclaiming free log segments hanacleaner has the following input flag

Flag	Unit	Details	Explanation	Default
-lr		maximum number of free log segments per service	if there are more free log segments for a service that this number then ALTER SYSTEM RECLAIM LOG will be executed Note: A too small value could cause overhead and contention due to RECLAIM LOG executions. The setting -lr 250 has been found to be a good compromise.	-1 (not used)

Example:
Here the ALTER SYSTEM RECLAIM LOG command is executed since there was a hana process that had more than one free log segment:

```
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -lr 1
The most used filesystem is using
96 %
**********
2017-02-28 21:32:13
**********
hdbsql -j -A -U SYSTEMKEY "ALTER SYSTEM RECLAIM LOG"
In total 1 log segments were reclaimed
```

House Keeping

HANACleaner – Audit Log Table

To clear the audit log database table hanacleaner has the following input flag

Flag	Unit	Details	Explanation	Default
-ur		retention time [days] of the audit log table	if the audit log database table has audit log older than these number days ALTER SYSTEM CLEAR AUDIT LOG UNTIL will be executed	-1 (not used)

Example:
Here the ALTER SYSTEM CLEAR AUDIT LOG UNTIL is executed and 29 entries in the audit log table were removed:

```
mo-fc8d991e0:/tmp/HANACleaner> python hanacleaner.py -ur 100
Will now check most used memory in the file systems.
The most used filesystem is using
36 %
************************
2017-07-31 14:22:48
hanacleaner by SYSTEMKEY
************************
29 entries in the audit log table were removed
```

House Keeping

HANACleaner – Pending Emails

To clear pending emails hanacleaner has the following input flag

Flag	Unit	Details	Explanation	Default
-pe		retention days for pending e-mails [days]	pending statistics server e-mail notifications older than these number of days are removed (requires SELECT and DELETE on the _SYS_STATISTICS schema)	-1 (not used)

Example:
Here the DELETE FROM _SYS_STATISTICS.STATISTICS_EMAIL_PROCESSING WHERE SECONDS_BETWEEN(SNAPSHOT_ID, CURRENT_TIMESTAMP) > 0 * 86400 is executed and 58 emails were removed:

```
pqladm@atgls90010:/tmp/HANACleaner> python hanacleaner.py -k HANACLEANERUSERKEY -pe 0
58 pending statistics server email notifications were removed
```

House Keeping

HANACleaner – Unknown Object Lock Entries

The transactional lock history in HOST_OBJECT_LOCK_STATISTICS may have unknown object entries that refer to dropped temporary tables (as per SAP Note 2147247)
These entries can be removed by the hanacleaner with following input flag

Flag	Unit	Details	Explanation	Default
-kr	days	min retained unknown object lock days	min age (today not included) of retained object lock entries with OBJECT_NAME = '(unknown)', see SAP Note 2147247	-1 (not used)

Example:
Here all transactional lock history entries with OBJECT_NAME = '(unknown)' are removed:

```
mo-fc8d991e0:/tmp/HANACleaner> python hanacleaner.py -kr 0
Will now check most used memory in the file systems.
The most used filesystem is using
35 %
**********************
2017-08-15 18:47:58
hanacleaner by SYSTEMKEY
**********************
    (Cleaning of the backup catalog was not done since -be and -bd
    (Cleaning traces was not done since -tc and -tf were both -1 (
    (Compression of the backup logs was not done since -zb was neg
    (Cleaning of the alerts was not done since -ar was negative (o
13345 object locks entries with unknown object names were removed
```

House Keeping

HANACleaner – Object History

Object history can be cleaned (as per SAP Note 2479702) using these flags:

Flag	Unit	Details	Explanation	Default
-om	mb	object history table max size	if the table _SYS_REPO.OBJECT_HISTORY is bigger than this threshold this table will be cleaned up according to SAP Note 2479702	-1 (not used)
-oo	true/ false	output cleaned memory from object table	displays how much memory was cleaned up from object history table	-1 (not used)

Example:
In this example there was nothing to clean up from the object history:

```
hsiadm@dewdfglp00836:/tmp/HANACleaner> python hanacleaner.py -om 1 -oo true
Will now check most used memory in the file systems. If it hangs there is an
    (Cleaning of unknown object locks entries was not done since -kr was nega
Object History was:0 mb and is now 0 mb.
0 mb were cleaned from object history
```

House Keeping

HANACleaner – Disk Fragmentation (1/2)

Unused space in the disk volumes can be fixed with the flag –fl

Flag	Unit	Details	Explanation	Default
-fl	%	fragmentation limit	maximum fragmentation of data volume files, of any service, before defragmentation of that service is started: ALTER SYSTEM RECLAIM DATAVOLUME '<host>:<port>' 120 DEFRAGMENT Note: If you use HSR see next slide	-1 (not used)
-fo	true/false	output fragmentation	displays data volume statistics before and after defragmentation	false

Example:

Here defragmentation will be done of all ports if fragmentation is more than 20% for any port:

```
ha1adm@dewdfglp00765:/tmp/HANACleaner> python hanacleaner.py -fl 20 -fo true

BEFORE FRAGMENTATION:
Host                Port          Used Space [B]          Total Space [B]   Fragmentation [%]
dewdfglp00765       30003         4337033216              4747952128        9.0
dewdfglp00765       30007         70078464                268566528         74.0



AFTER FRAGMENTATION:
Host                Port          Used Space [B]          Total Space [B]   Fragmentation [%]
dewdfglp00765       30003         4337033216              4747952128        9.0
dewdfglp00765       30007         93069312                268435456         65.0


For Host dewdfglp00765 and Port 30007 defragmentation changed by 9.0 %
```

House Keeping

HANACleaner – Disk Fragmentation (2/2)

If SAP HANA has snapshots preserved RECLAIM DATAVOLUME fails with

```
general error: Shrink canceled, probably because of snapshot pages
```

This situation is normal if you use SAP HANA System Replication (HSR) (see SAP Note 1999880 Q19)

SAP Note 2332284 explains that to make RECLAIM DATAVOLUME work if you have HSR you have to temporarily change some parameters

This is not, and will not be, implemented in SAP HANACleaner!

Why?

- **HANACleaner is an automatic house-keeper → dangerous if it starts to automatically change SAP HANA parameters**
- **Additionally, from security point of view, the technical user used to execute SAP HANACleaner should not have INIFILE ADMIN**

House Keeping

HANACleaner – LOB Fragmentation

High LOB fragmentation can be fixed using the –lob* flags

Flag	Unit	Details	Explanation	Default
-lobf	%	max allowed fragmentation for an LOB	an LOB column with higher percentage of fragmentation, i.e. (physical size - binary size)/physical size, will be reorganized	-1 (not used)
-lobp	true/false	fragmentation check also includes packed	decides if -lobf should also include LOBs that already are packed	false
-lobs	millions	too many small file LOBs	if an LOB column has too many small (< 4 KB) file LOBs, this column will be reorganized	-1 (not used)
-lobn	millions	too many LOBs	if an LOB column has too many LOB objects, it will be reorganized	-1 (not used)
-lobw	true/false	write LOBs	print the LOBs before and after the lob reorg	false
-lobl	list of schemas	LOB location	comma seperated list of schemas that are considered for -lobf, -lobs, and -lobn	

Example:
Here LOB reorg (ONLINE for COLUMN store tables only) will be done for all file LOBs with > 40% fragmentation:

It turned out that it worked for all COLUMN store tables, but not for the one and only ROW store table (this is currently under investigation)

```
chpadm@atgvmls7040:/tmp/HANACleaner> python hanacleaner.py -k HANACLEANERUSERKEYT1 -lobf 40 -lobw true -lobl SAPQH1
LOB Columns With Too High Fragmentation BEFORE:
Schema  Table            Column             Lob_Storage_Type  Fragmentation [%]
SAPQH1  CLS_ASSIGNMENT   REMARK             FILE              55.753
SAPQH1  /IWBEP/I_SBD_AT  VALUE              FILE              67.119
SAPQH1  CTS_HOT_OBJECT   HANA_CONTENT_BDATA  FILE              74.145
SAPQH1  ECCUST_TIPS      TEXT               FILE              70.129


LOB Columns With Too High Fragmentation AFTER:
Schema  Table           Column  Lob_Storage_Type  Fragmentation [%]
SAPQH1  CLS_ASSIGNMENT  REMARK  FILE              55.753

3 is the difference of total number lob columns with too high fragmentation after LOB reorg (-lobf)
```

House Keeping

HANACleaner – Table Compression (1/2)

Compression re-optimization of column store tables can be automated

Flag	Unit	Details	Explanation	Default
1. Both following two flags, -cc, and -ce, must be > 0 to control the force compression optimization on tables that never was compression re-optimized (i.e. last_compressed_record_count = 0):				
-cc		Max allowed raw main records	If number raw main rows are larger this could be compression optimized if compressed rows = 0 and -ce indicates it also	-1 (not used) *e.g. 10000000*
-ce	[GB]	Max allowed estimated size	If estimated size is larger this could be compression optimized if compressed rows = 0 and -cc indicates it also	-1 (not used) *e.g. 1*
2. All following three flags, -cr, -cs, and -cd, must be > 0 to control the force compression optimization on tables with columns with compression type 'DEFAULT' (i.e. no additional compression algorithm in main)				
-cr		Max allowed rows	If a column has more rows and compression = 'DEFAULT' this table could be re-compressed if -cs and -cd indicate it also	-1 (not used) *e.g. 10000000*
-cs	[MB]	Max allowed size	If a column is larger and compression = 'DEFAULT' this table could be re-compressed if -cr and -cd indicate it also	-1 (not used) *e.g. 500*
-cd	[%]	Min allowed distinct count	If a column has smaller distinct row quota this table could be re-compressed if -cr and -cs indicate it also	-1 (not used) *e.g. 1*
3. Both following two flags, -cq and -cu, must be > 0 to control the force compression optimization on tables whose UDIV quota is too large, i.e. #UDIVs/(#raw main + #raw delta)				
-cq	[%]	Max allowed UDIV quota	If a column's UDIV quota is larger this table could be re-compressed if -cu indicates it also	-1 (not used) *e.g. 150*
-cu		Max allowed UDIVs	If a column has more UDIVs → compress if –cq indicates it also	-1 (not used) *e.g. 10000000*

House Keeping

HANACleaner – Table Compression (2/2)

Some column store tables might have to have its compression re-optimized

This can be atomized with the following flags:

Flag	Unit	Details	Explanation	Default
4. Flag -cb must be > 0 to control the force compression optimization on tables with columns with SPARSE (<122.02) or PREFIXED and a BLOCK index				
-cb		Max allowed rows	If more rows → compress if BLOCK and PREFIXED	-1 (not used) *e.g. 100000*
Following three flags are general; they control all three, 1., 2., 3., and 4. compression optimization possibilities above				
-cp	[true/false]	Per partition	Switch to consider above flags per partition	false
-cm	[true/false]	Merge before	Switch to perform a delta merge before compression	false
-co	[true/false]	Output	Switch to print out tables selected for compression optimization	false

Example: Here (1.) tables that were never compressed with more than 10 million raw records and more than 1 GB of estimated size or (2.) tables with columns only default compressed with more than 10 million rows, distinct row quota less than 1% and size more than 500 MB or (3.) tables with UDIV quota larger than 150% and more than 10 million UDIVs, will be compression re-optimized:

```
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -cc 10000000 -ce 1
 -cr 10000000 -cs 500 -cd 1 -cq 150 -cu 10000000 -cp true -cm true
    (Reclaim of row store containers was not done since -rc was negative
2 column store tables were compression re-optimized
```

House Keeping

HANACleaner – events (handled/unhandled)

Events can be acknowledged and handled (in case of unhandled events) with the following input flags

Flag	Unit	Details	Explanation	Default
-eh	day	minimum retained days for handled events	handled events that are older that this number of days will be acknowledged and then deleted	-1 (not used)
-eu	day	minimum retained days for unhandled events	unhandled events that are older that this number of days will be handled, acknowledged and then deleted	-1 (not used)

Example:
Here handled events older than 5 days and unhandled events older than 34 days were deleted.
It turned out the 113 unhandled events were deleted:

```
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -eh 5 -eu 34
In total 113 events were cleaned, 0 of those were handled. There are 61 events left, 0 of those are handled.
```

House Keeping

HANACleaner – Virtual Tables‘ Statistics

Smart Data Access Virtual Tables can get their statistics created with the -vs flag (see SAP Note 1872652)

Flag	Unit	Details	Explanation	Default
-vs	true / false	create statistics for virtual tables	Switch to create optimization statistics for those virtual tables that are missing statistics (Note: could cause expensive operations!)	false
-vl		schema list of virtual tables	if you only want tables in some schemas to be considered for the creation of statistics provide here a comma separated list of those schemas	default '' (all schemas are considered)
-vr	true/ false	ignore secondary monitoring tables	normaly statistics for the the virtual tables in the _SYS_SR_SITE* schemas are not needed to be created nor updated, so they are by default ignored	true
-vm	int	max number columns for statistic creations	Creating VTs' statistics can be expensive if there are many columns (SAP Note 1872652), so if -vm is less number of columns of the VT, statistics will be created with multiple executions. Example: if -vm = 100 and # columns of the VT is 230, 3 executions will be done for this VT, the last execution with 30 columns	1000

Examples (Note: Here default Table Statistics (i.e. HISTOGRAM, see SAP Note 1872652) is used! For more options see next slide.):

Here statistics optimization was created for 3 out of 4 virtual tables (the 4th already had statistics):

```
ha1adm@dewdfglp00766:/tmp/HANACleaner> python hanacleaner.py -vs true
Will now check most used memory in the file systems. If it hangs there is an issue with  df -h,
Optimization statistics was created for 3 virtual tables (in total there are 4 virtual tables)
    (Cleaning of the hanacleaner logs was not done since -or was negative (or not specified))
```

Here number of columns per executions was limited to 2, so 3 executions were done to create statistics on this table:

```
python hanacleaner.py -vs true -vm 2 -os true -k HA1SYSTEMKEY
CREATE STATISTICS ON \"SYSTEM\".\"OQL_INVOICE_IX_CS\" (\"INVOICE_ID\", \"INVOICE_AMT\") TYPE HISTOGRAM
CREATE STATISTICS ON \"SYSTEM\".\"OQL_INVOICE_IX_CS\" (\"ITEM_ID\", \"CUSTOMER_ID\") TYPE HISTOGRAM
CREATE STATISTICS ON \"SYSTEM\".\"OQL_INVOICE_IX_CS\" (\"STORE_ID\") TYPE HISTOGRAM
```

House Keeping

HANACleaner – Virtual Tables‘ Statistics Creation

Smart Data Access Virtual Tables can get their statistics created, defined by these flags

Flag	Unit	Details	Explanation	Default
-vt	HISTOGRAM/SIMPLE/TOPK/SKETCH/SAMPLE/RECORD_COUNT	default statistics type	type of data statistics object	HISTOGRAM
-vn	number rows	max number of rows for defult type	if the VT has less or equal number of rows specified by -vn the default statistics type, defined by -vt, is used, else the type defined by -vtt is used	-1 (not considered)
-vtt	HISTOGRAM/SIMPLE/TOPK/SKETCH/SAMPLE/RECORD_COUNT	statistics type for large tables	type of data statistics object used if the VT has more rows than specified by -vn and the database is HANA	SIMPLE
-vto	HISTOGRAM/SIMPLE/TOPK/SKETCH/SAMPLE/RECORD_COUNT	statistics type for other DBs	type of data statistics object if the remote database is not HANA	

Example:
Here statistics optimization HISTOGRAM was created for 1 out of 1 virtual tables (since number of rows were less than 40 and the DB was HANA):

```
python hanacleaner.py -k K1 -vs true -vt HISTOGRAM -vn 40 -vtt RECORD_COUNT -vto SIMPLE -os true
CREATE STATISTICS ON \"DMM_MOD_1\".\"QUALITY_VT\" (\"CUSTOMER\", \"MATERIAL_ID\", \"PRICE\") TYPE HISTOGRAM
Optimization statistics was created for 1 virtual tables (in total there are 1 virtual tables)
```

Note: This only creates statistics! For statistics refresh, see next slide

House Keeping

HANACleaner – Virtual Tables‘ Statistics Refresh

Smart Data Access Virtual Tables can get their statistics refreshed, defined by this flag

Flag	Unit	Details	Explanation	Default
-vnr	number of days > 0	refresh age of VT statistics	if the VT statistics of a table is older than this number of days it will be refreshed (Note: -vl and -vr holds also for refresh)	-1 (no refresh)

Example:
Here statistics optimization was refreshed for 3 out of 3 data statistics (i.e. the 3 columns of the 1 existing VT):

```
xscadm@atgvmls866:/tmp/HANACleaner> python hanacleaner.py -k K1 -vnr 1 -os true
```

```
REFRESH STATISTICS \"DMM_MOD_1\".\"_STAT_SYS_637500647428310001_354234\"
REFRESH STATISTICS \"DMM_MOD_1\".\"_STAT_SYS_637500647426940001_354233\"
REFRESH STATISTICS \"DMM_MOD_1\".\"_STAT_SYS_637500647428620001_354235\"
Refresh of statistics was done for 3 data statistics (in total there are 3 data statistics)
```

Note: This is not the same thing as refresh of a virtual table! See -vtr below.

House Keeping

HANACleaner – Data Statistics Refresh

Data Statistics (see SAP Note 2800028) can be refreshed, defined by this flag

Flag	Unit	Details	Explanation	Default
-dsr	number of days > 0	refresh age of data statistics	if the data statistics is older than this number of days it will be refreshed (Note: this is the same as –vnr, except that -vl and -vr are ignored)	-1 (no refresh)

Example:

Here a table and statistics were created as per example in SAP Note 2800028 #8, so we have two statistics reasons CREATE STATISTICS:

```
select DATA_STATISTICS_NAME, DATA_STATISTICS_SCHEMA_NAME, DATA_SOURCE_SCHEMA_NAME, DATA_SOURCE_OBJECT_NAME, DATA_SOURCE_OBJECT_TYPE, DATA_SOURCE_COLUMN_NAMES, LAST_REFRESH_TIME,
LAST_REFRESH_REASON from "SYS"."DATA_STATISTICS"
```

DATA_STATISTICS_NAME	DATA_STATISTICS_SCHEMA_NAME	DATA_SOURCE_SCHEMA_NAME	DATA_SOURCE_OBJECT_NAME	DATA_SOURCE_OBJECT_TYPE	DATA_SOURCE_COLUMN_NAMES	LAST_REFRESH_TIME	LAST_REFRESH_REASON
_STAT_SYS_638024397154130001_689074	ASCHEMA	ASCHEMA	AAA	TABLE	X, Y	25 Oct 2022, 03:55:15.531	CREATE STATISTICS
_STAT_SYS_638024397158020001_689075	ASCHEMA	ASCHEMA	AAA	TABLE	X, Z	25 Oct 2022, 03:55:15.913	CREATE STATISTICS

```
ha2adm@atgvmls7045:/tmp/HANACleaner> python hanacleaner.py -dsr 1 -os true -k T1KEY
REFRESH STATISTICS \"ASCHEMA\".\"_STAT_SYS_638024397154130001_689074\"
REFRESH STATISTICS \"ASCHEMA\".\"_STAT_SYS_638024397158020001_689075\"
```

```
select DATA_STATISTICS_NAME, DATA_STATISTICS_SCHEMA_NAME, DATA_SOURCE_SCHEMA_NAME, DATA_SOURCE_OBJECT_NAME, DATA_SOURCE_OBJECT_TYPE, DATA_SOURCE_COLUMN_NAMES, LAST_REFRESH_TIME,
LAST_REFRESH_REASON from "SYS"."DATA_STATISTICS"
```

DATA_STATISTICS_NAME	DATA_STATISTICS_SCHEMA_NAME	DATA_SOURCE_SCHEMA_NAME	DATA_SOURCE_OBJECT_NAME	DATA_SOURCE_OBJECT_TYPE	DATA_SOURCE_COLUMN_NAMES	LAST_REFRESH_TIME	LAST_REFRESH_REASON
_STAT_SYS_638024397154130001_689074	ASCHEMA	ASCHEMA	AAA	TABLE	X, Y	25 Oct 2022, 04:54:48.605	REFRESH STATISTICS
_STAT_SYS_638024397158020001_689075	ASCHEMA	ASCHEMA	AAA	TABLE	X, Z	25 Oct 2022, 04:54:48.769	REFRESH STATISTICS

House Keeping

HANACleaner – Virtual Tables Refresh (1/2)

Virtual Tables might need to get refreshed if e.g. the definition of the source table has changed. The procedure CHECK_VIRTUAL_TABLES shows all virtual tables that need to be refreshed. HANACleaner can automatically run this check and execute the refresh of virtual tables, based on following flags:

Flag	Unit	Details	Explanation	Default
-vtr	'I', 'W', and/or, 'E' or any combination	refresh virtual tables	a string that defines if refresh should be done on virtual tables shown as mismatch from the procedure CHECK_VIRTUAL_TABLES with a severity of INFO, WARNING, and/or ERROR. The string can be 1, 2, or 3 characters long with the characters I (for INFO), W (for WARNING), and E (for ERROR).	'' (not used)
-vts	name of a schema	schema with the virtual tables	the -vtr will only be done for tables in this schema	'' (all schemas)
-vta	name of a VT	the virtual tables	the -vtr will only be done for this virtual table	'' (all tables)
-vtp	true/false	print VT checks	print VTs found by CHECK_VIRTUAL_TABLES before and after the refreshes	false

Example:
Here all virtual tables in schema VT_SCHEMA will be checked if any of those needs to be refreshed, but -es false, the refreshes will not actually be executed (only shown, since -os true):

```
xscadm@atgvmls7040:/tmp/HANACleaner> python hanacleaner.py -k T1KEY -vtr WI -vts VT_SCHEMA -es false -os true
ALTER VIRTUAL TABLE \"VT_SCHEMA\".\"QUALITY2_XS2\" REFRESH DEFINITION
ALTER VIRTUAL TABLE \"VT_SCHEMA\".\"QUALITY3_XS2\" REFRESH DEFINITION
Refresh of 0 virtual tables was done (-vtr)
```

House Keeping

HANACleaner – Virtual Tables Refresh (2/2)

Example:
Here all virtual tables in schema VT_SCHEMA will be checked if they need to be refreshed. Only on those tables where the mismatch found by CHECK_VIRTUAL_TABLES had severity WARNING and INFO, a refresh will be executed. All the mismatches found before and after the refreshes are printed (here 2 before, and 0 after):

```
xscadm@atgvmls7040:/tmp/HANACleaner> python hanacleaner.py -k T1KEY -vtr WI -vts VT_SCHEMA -vtp true
*** Virtual Table Checks (by CHECK_VIRTUAL_TABLES) BEFORE refresh:
------------------------------------------------------------
Virtual Table VT_SCHEMA.QUALITY2_XS2 has an issue with column PRICE with severity WARNING.
The action that caused the issue is COLUMN_TYPE_CHANGED.
Explanation: column PRICE in remote table changed type from INTEGER to DOUBLE.
------------------------------------------------------------
------------------------------------------------------------
Virtual Table VT_SCHEMA.QUALITY3_XS2 has an issue with column NEW_COLUMN with severity INFO.
The action that caused the issue is COLUMN_ADDED.
Explanation: column NEW_COLUMN added to remote table.
------------------------------------------------------------
*** Virtual Table Checks (by CHECK_VIRTUAL_TABLES) AFTER refresh:
Refresh of 2 virtual tables was done (-vtr)
```

House Keeping

HANACleaner – INI File History (≥H2SPS03) (1/2)

To remove old inifile content history hanacleaner has the following input flag

Flag	Unit	Details	Explanation	Default
-ir	days	inifile content history retention	deletes older inifile content history (should be more than 1 year)	-1 (not used)

Example:

```
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -ir 300

INPUT ERROR: -ir must be larger than 365. Please see --help for more inf
ormation. (If you disagree please remove this check on your own risk.)
```

Example:

```
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -ir 400

ERROR: the -ir flag is only supported starting with SAP HANA 2.0 SPS03.
You run on SAP HANA 1 revision 122 maintenance revision 15
```

House Keeping

HANACleaner – INI File History (≥H2SPS03) (2/2)

To remove old inifile content history hanacleaner has the following input flag

Flag	Unit	Details	Explanation	Default
-ir	days	inifile content history retention	deletes older inifile content history (should be more than 1 year)	-1 (not used)

Example:

```
pqladm@atgls90012:/tmp/HANACleaner> python hanacleaner.py -ir 400 -k HANACLEANERUSERKEY_PQL90012
Will now check most used memory in the file systems. If it hangs there is an issue with  df -h, t
    (Creation of optimization statistics for virtual tables was not done since -vs was false (or
5 inifile history contents were removed
    (Cleaning of the hanacleaner logs was not done since -or was negative (or not specified))
pqladm@atgls90012:/tmp/HANACleaner>
```

House Keeping

HANACleaner – No Execute

HANACleaner questions are normally HANA questions! With these flags it is possible to let HANACleaner print out the crucial SQLs without actually executing them → useful for debugging

Flag	Unit	Details	Explanation	Default
-es	true/false	execute sql	Execute all crucial housekeeping tasks (useful to turn off for investigations with –os=true)	True
-os	true/false	output sql	Prints all crucial housekeeping tasks (useful for debugging with –es=false)	False

```
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -es false -os true -be 12 -bd 12 -tc 42 -ar 12 -lr 0
Will now check most used memory in the file systems. (If it takes too long, investigate why   df -h  hangs.)
The most used filesystem is using
94%
**********************************************
2017-09-24 11:38:47
hanacleaner by SYSTEMKEY
Cleanup Statements will NOT be executed
**********************************************
SELECT * from DUMMY
BACKUP CATALOG DELETE ALL BEFORE BACKUP_ID 1501268432361
0 data backup entries and 0 log backup entries were removed from the backup catalog
ALTER SYSTEM CLEAR TRACES ('ALERT','CLIENT','CRASHDUMP','EMERGENCYDUMP','EXPENSIVESTATEMENT','RTEDUMP','UNLOAD'
0 trace files were removed
    (Cleaning dumps was not done since -dr was -1 (or not specified))
    (Compression of the backup logs was not done since -zb was negative (or not specified))
DELETE FROM _SYS_STATISTICS.STATISTICS_ALERTS_BASE WHERE ALERT_TIMESTAMP < ADD_DAYS(CURRENT_TIMESTAMP, -12)
0 alerts were removed
    (Cleaning of unknown object locks entries was not done since -kr was negative (or not specified))
    (Cleaning of the object history was not done since -om was negative (or not specified))
ALTER SYSTEM RECLAIM LOG
0 log segments were reclaimed
```

House Keeping

HANACleaner – Configuration File

HANACleaner can be controlled with a configuration file (additional flags will overwrite the config file)

Flag	Unit	Details	Explanation	Default
-ff		flag file	full path to the configuration file	

Example:

```
xshadm@atgvmls666:/tmp/HANACleaner> more hanacleaner_configfile.txt
My HANACleaner Configuration:
-zb 50
-tf 42
-td true
-ar 42
-eh 7
-eu 42
-fs /dev/sdb1
-op /tmp/hanacleaneroutput/
-or 42
-fs "|grep sdc3"

xshadm@atgvmls666:/tmp/HANACleaner> python hanacleaner.py -ff hanacleaner_configfile.txt
Will now check most used memory in the file systems. (If it takes too long, investigate why   df -h  hangs.)
The most used filesystem is using
18%
**********************
2017-09-05 09:42:57
hanacleaner by SYSTEMKEY
**********************
    (Cleaning of the backup catalog was not done since -be and -bd were both negative (or not specified))
0 trace files were removed
    (Cleaning dumps was not done since -dr was -1 (or not specified))
```

<u>Note:</u> For multiple configuration files, see some slides below

House Keeping

HANACleaner – Configuration File Example

Note: HANACleaner will never give any recommendations! Here is an EXAMPLE of a config file:

start in "chicken mode" (comment out if we want to execute)
-es false
-os true
backup catalog retention days
-bd 42
backup.log backint.log size limit, use -zk to remove zip file (to be added later maybe)
-zb 50
trace files retention days see sql ref for details
-tf 42
-tc 42
alert retention days
-ar 42
event log cleanup
-eh 42
-eu 42
retention day for dump files
-dr 42
redo log reclaim
-lr 10
disk fragmentation reorg
-fl 40
ini file history cleanup older than 1 year
-i 365
hanacleaner log destination and cleanup
-op <full path>/xsc_output
-or 42
key and database
-k HANACLEANERKEY
-dbs SYSTEMDB,XSC,XS1
-df false

House Keeping

HANACleaner – Configuration Files (1/2)

HANACleaner can be controlled with a list of configuration files (files listed later in the -ff list will overwrite flags from files listed earlier in the list, flags on the command line will overwrite the configuration files)

Flag	Unit	Details	Explanation	Default
-ff		flag files	full paths to the configuration files	
-oc	true/false	output configuration	logs all set parameters and where the flags were set	false

Example:

```
xscadm@atgvmls866:/tmp/HANACleaner> more hanacleanerconfig.txt
backupcatalog
-be 5
-bd 10
traces
-tc 42
general cleanup
-gd /tmp,/tmp/HANACleaner
-gw asdf,asdf
key
-k T1KEY
xscadm@atgvmls866:/tmp/HANACleaner>
xscadm@atgvmls866:/tmp/HANACleaner> more hanacleanerconfig2.txt
backupcatalog
-be 50
-bd 100
traces
-tc 420
```

(the example continues on the next slide)

House Keeping

HANACleaner – Configuration Files (2/2)

Example (continued):

```
xscadm@atgvmls866:/tmp/HANACleaner> python hanacleaner.py -ff hanacleanerconfig.txt,hanacleanerconfig2.txt -tc 4200 -oc true
Will now check most used memory in the file systems. If it hangs there is an issue with  df -h, then see if the -fs flag helps.
The most used filesystem is using
51%
************************************************************
2021-09-02 10:30:01
hanacleaner as xscadm by T1KEY on XSC(00) on DB XSC with
hanacleaner.py -ff hanacleanerconfig.txt,hanacleanerconfig2.txt -tc 4200 -oc true
Cleanup Statements will be executed (-es is default true)
-ff      = hanacleanerconfig.txt,hanacleanerconfig2.txt from command line
-gd      = /tmp,/tmp/HANACleaner from hanacleanerconfig.txt
-bd      = 100 from hanacleanerconfig2.txt
-be      = 50 from hanacleanerconfig2.txt
-k       = T1KEY from hanacleanerconfig.txt
-gw      = asdf,asdf from hanacleanerconfig.txt
-oc      = true from command line
-tc      = 4200 from command line
Before using HANACleaner read the disclaimer!
python hanacleaner.py --disclaimer
************************************************************
0 data backup entries and 0 log backup entries were removed from the backup catalog
0 trace files were removed
```

House Keeping

HANACleaner – output

To control the output of the hanacleaner there are these flags

Flag	Unit	Details	Explanation	Default
-op		output path	full path of the folder (will be created if not there) where the hanacleaner logs are written Note: if you include %SID in the output path, it will automatically be replaced with the actually SID of your system	(not used)
-or	days	retention	logs in the path specified with -op are only saved for this number of days	-1 (not used)
-so	true/ false	standard out switch	1: write to std out, 0: do not write to std out	1

Example:
Here a output folder is deleted and then automatically created again by hanacleaner and the daily log file written into it:

```
oqladm@ls80010:/tmp/HANACleaner> rm -r /tmp/hanacleaneroutput/
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -be 100 -op /tmp/hanacleaneroutput
The most used filesystem is using
96 %
********************
2017-02-28 23:06:33
********************
In total 0 data backup entries were removed from the backup catalog
oqladm@ls80010:/tmp/HANACleaner> more /tmp/hanacleaneroutput/hanacleanerlog_2017-02-28.txt
********************
2017-02-28 23:06:33
********************
In total 0 data backup entries were removed from the backup catalog
```

House Keeping

HANACleaner – Emails

To warn if something goes wrong HANACleaner can send emails

Flag	Unit	Details	Explanation	Default
-en		emails for email notification	a comma separated list of email addresses that will receive emails in case of fatal errors	(not used)
-et	seconds	timeout time for email warning	emails specified with –en gets emails if HANACleaner ran too long	(not used)
-enc		email client	to explicitly specify the client (e.g mail, mailx, mutt,..), only useful if -en if used	mailx
-ens		sender's email	to explicitly specify sender's email address, only useful if -en if used	(configured used)
-enm		mail server	to explicitly specify the mail server, only useful if -en is used	(configured used)

Example:
Here HANACleaner took longer than 10 seconds, so it send an email

```
xscadm@atgvmls866:/tmp/HANACleaner> python hanacleaner.py -be 400 -tc 50 -en christian.hansen01@sap.com -et 10 -k T1KEY
Will now check most used memory in the file systems. If it hangs there is an issue with  df -h, then see if the -fs flag helps.
The most used filesystem is using
51%
Warning: HANACleaner has been running longer than 10 seconds.
**********************************************************
```

Message from HANACleaner on XSC

SH SAP HANA Database System Administrator <xscadm@atgvmls866.wdf.sap.corp>
To Hansen, Christian

Hi Team,

HANACleaner reports:

Warning: HANACleaner has been running longer than 10 seconds.

House Keeping

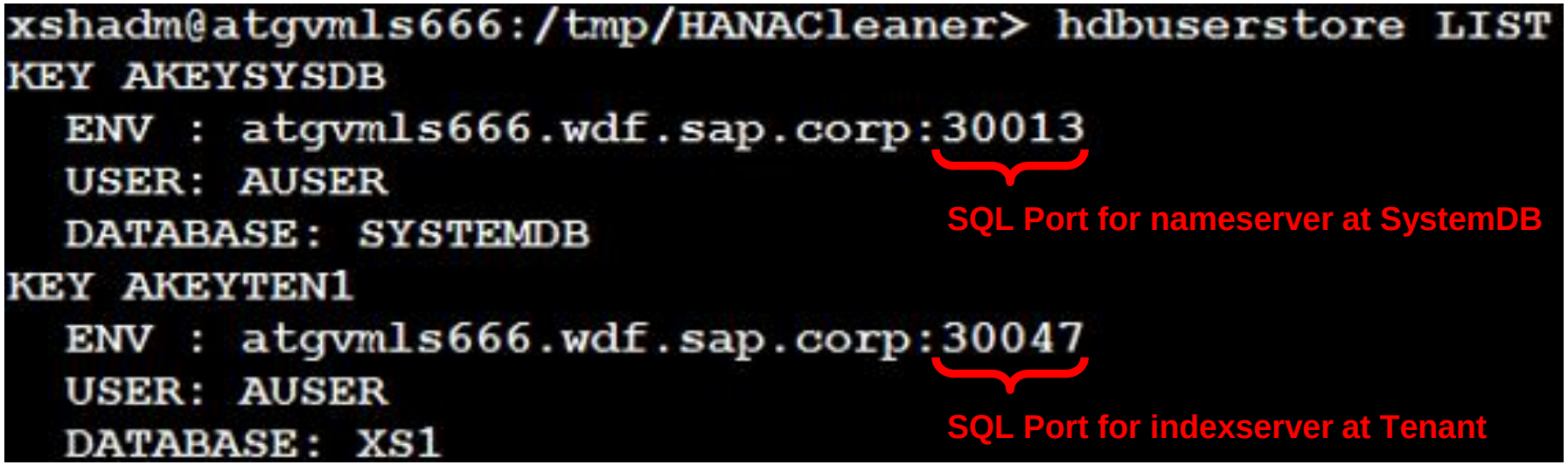
HANACleaner – MDC (1/4)

In a MDC system the hanacleaner can clean the SystemDB and multiple Tenants in one execution

List the DB users for the system and the tenants in hdbuserstore and list them with the –k flag

Flag	Unit	Details	Explanation	Default
-k		DB user key(s)	This is the DB user key saved in the hdbuserstore, it could also be a list of comma separated userkeys (useful in MDC environments)	SYSTEMKEY

Example:
Here two keys are stored; one for SystemDB and one for a Tenant:



House Keeping

HANACleaner – MDC (2/4)

Example:
Here trace files older than 42 days are deleted from the SystemDB and from a Tenant:

```
xshadm@atgvmls666:/tmp/HANACleaner> python hanacleaner.py -tf 42 -k AKEYSYSDB,AKEYTEN1
Will now check most used memory in the file systems. If it hangs there is an issue with
The most used filesystem is using
85%
*********************************************
2017-09-27 15:14:35
hanacleaner by AKEYSYSDB
Cleanup Statements will be executed
*********************************************
49 trace files were removed
*********************************************
2017-09-27 15:14:38
hanacleaner by AKEYTEN1
Cleanup Statements will be executed
*********************************************
21 trace files were removed
```

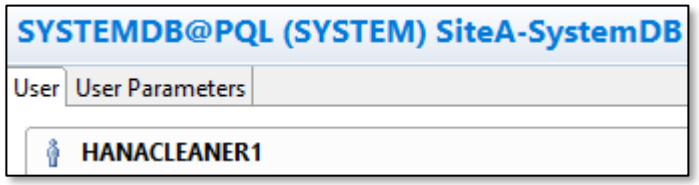
House Keeping

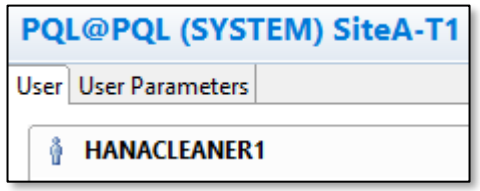
HANACleaner – MDC (3/4)

In a MDC system the hanacleaner can clean the SystemDB and multiple Tenants with one key

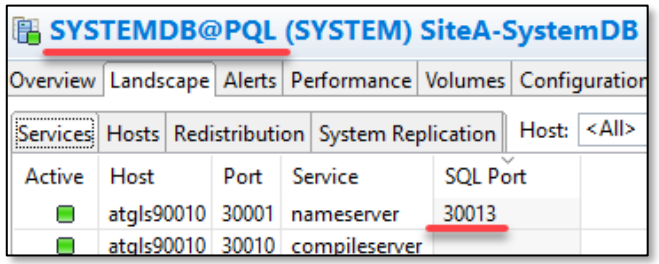
Maintain a user with same user name and same password in multiple DBs in one HANA System

Example:
Here the user HANACLEANER1 with same password was created in both SystemDB and in a Tenant





(for privileges,
see earlier slides)



Then only one key, for the SystemDB, was provided in hdbuserstore

```
pqladm@atgls90010:/tmp> hdbuserstore set SDBKEY atgls90010:30013 HANACLEANER1 PassWd1234
```

Test that this single key can be used to access both databases:

```
pqladm@atgls90010:/tmp> hdbsql -j -A -x -U SDBKEY -d SYSTEMDB "select * from m_database"
| SYS | DATABASE | HOST       | START_TIME                    | VERSION                  | USAG |
| --- | -------- | ---------- | ----------------------------- | ------------------------ | ---- |
| PQL | SYSTEMDB | atgls90010 | 2018-09-27 15:27:00.060000000 | 2.00.032.00.1533114046 | TEST |
pqladm@atgls90010:/tmp>
pqladm@atgls90010:/tmp> hdbsql -j -A -x -U SDBKEY -d PQL "select * from m_database"
| SYS | DAT | HOST       | START_TIME                    | VERSION                  | USAG |
| --- | --- | ---------- | ----------------------------- | ------------------------ | ---- |
| PQL | PQL | atgls90010 | 2018-09-27 15:27:10.593000000 | 2.00.032.00.1533114046 | TEST |
```

House Keeping

HANACleaner – MDC (4/4)

In a MDC system the hanacleaner can clean the SystemDB and multiple Tenants with one key

Flag	Unit	Details	Explanation	Default
-dbs		DB key(s)	this can be a list of databases accessed from the system defined by -k (-k can only be one key if -dbs is used)	''

Example:
Here the key SDBKEY is used to access the system, then it is specified with -dbs that two databases, SYSTEMDB and PQL, will be cleaned up on their old trace files

```
pqladm@atgls90010:/tmp/HANACleaner> python hanacleaner.py -k SDBKEY -dbs SYSTEMDB,PQL -tc 20
Will now check most used memory in the file systems. If it hangs there is an issue with  df -h
The most used filesystem is using
78%
************************************************************
2018-10-08 20:10:50
hanacleaner by SDBKEY on PQL(00) on DB SYSTEMDB with
hanacleaner.py -k SDBKEY -dbs SYSTEMDB,PQL -tc 20
Cleanup Statements will be executed (-es is default true)
Before using HANACleaner read the disclaimer!
python hanacleaner.py --disclaimer
************************************************************
    (Cleaning of the backup catalog was not done since -be and -bd were both negative (or not
0 trace files were removed
    (Cleaning dumps was not done since -dr was -1 (or not specified))
************************************************************
2018-10-08 20:10:51
hanacleaner by SDBKEY on PQL(00) on DB PQL with
hanacleaner.py -k SDBKEY -dbs SYSTEMDB,PQL -tc 20
Cleanup Statements will be executed (-es is default true)
Before using HANACleaner read the disclaimer!
python hanacleaner.py --disclaimer
************************************************************
    (Cleaning of the backup catalog was not done since -be and -bd were both negative (or not
6 trace files were removed
    (Cleaning dumps was not done since -dr was -1 (or not specified))
```

House Keeping

HANACleaner – HANA System Replication, HSR (1/2)

On a Secondary in a HSR setup one can use -oi to let HANACleaner wait for a takeover

Flag	Unit	Details	Explanation	Default
-oi	seconds	Online Check Interval (s)	< 0: HANACleaner does not check if online or secondary = 0: if not online or not primary HANACleaner will abort > 0: time it waits before it checks if DB is online and primary again Note: For > 0, you might have to use cron with a lock (see the HANASitter pdf, "HANASitter & CRON" slide)	-1 (not used)

Example:

Here the HANACleaner is started on a system that is a Secondary in a HSR setup, with online check interval 10 seconds:

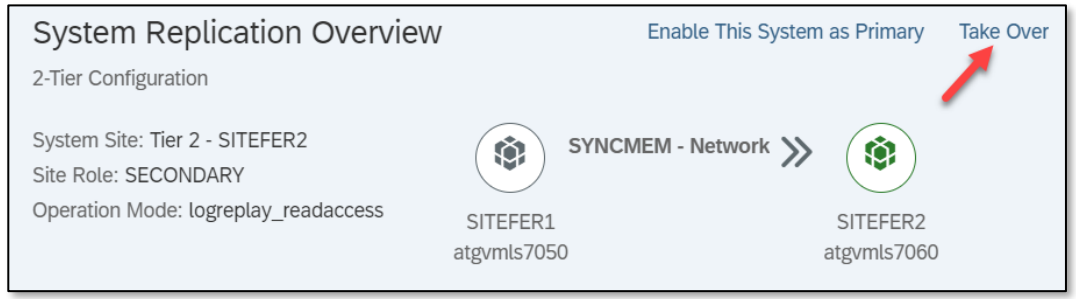
```
ha2adm@atgvmls7060:/tmp/HANACleaner> python hanacleaner.py -be 400 -bd 400 -oi 10 -k HANACLEANERUSERKEY
Will now check most used memory in the file systems. If it hangs there is an issue with  df -h, then see if the -fs flag helps.
The most used filesystem is using
77%
************************************************************
2019-11-26 23:48:57
hanacleaner by HANACLEANERUSERKEY on HA2(00)  with
hanacleaner.py -be 400 -bd 400 -oi 10 -k HANACLEANERUSERKEY
Cleanup Statements will be executed (-es is default true)
Before using HANACleaner read the disclaimer!
python hanacleaner.py --disclaimer
************************************************************
Online Check      , 2019-11-26 23:48:57    ,     -           , True         , True       , Number running services: 10 out of 10
Primary Check     , 2019-11-26 23:49:00    ,     -           , True         , False      ,

One of the online checks found out that this HANA instance, 00, is not online.
HANACleaner will now have a 10 seconds break and check again if this Instance is online after the break.
```

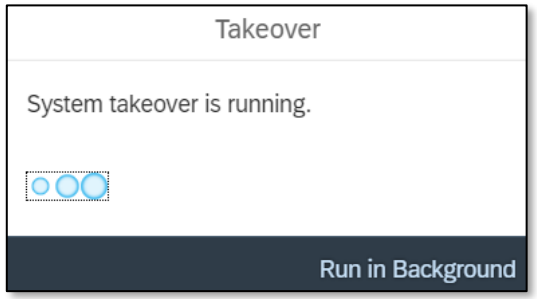
(example continues on next slide)

House Keeping

HANACleaner – HANA System Replication, HSR (2/2)

Example (continued from previous slide):
A take over is performed:





Then the HANACleaner that runs on the previous Secondary can now start with the cleanup tasks:

```
HANACleaner will now have a 10 seconds break and check again if this Instance is online after the break.

Online Check      , 2019-11-26 23:50:15    ,     -             , True         , True         , Number running services: 10 out of 10
Primary Check     , 2019-11-26 23:50:18    ,     -             , True         , True         ,
0 data backup entries and 0 log backup entries were removed from the backup catalog
    (Cleaning traces was not done since -tc and -tf were both -1 (or not specified))
```

House Keeping

HANACleaner – Run as ROOT

To restrict the access to the HANACLEANER user it is possible to run HANACleaner as root, i.e. using the hdbuserstore of root, for that one must set the key also in root's hdbuserstore

```
atgvmls866:~ # source /usr/sap/XSC/home/.sapenv.sh
root@atgvmls866:/usr/sap/XSC/HDB00> hdbuserstore set T1CLEANKEY atgvmls866:30015@XSC HANACLEANERUSER
root@atgvmls866:/usr/sap/XSC/HDB00> hdbuserstore LIST
DATA FILE       : /root/.hdb/atgvmls866/SSFS_HDB.DAT
KEY FILE        : /root/.hdb/atgvmls866/SSFS_HDB.KEY

KEY T1CLEANKEY
  ENV : atgvmls866:30015
  USER: HANACLEANERUSER
  DATABASE: XSC
```

Example: Here root's hdbuserstore is filled with the key T1CLEANKEY, only root can access it, so only root can run HANACleaner using this key, preventing <sid>adm users that don't have root access to get access to all privileges of the HANACLEANERUSER

```
atgvmls866:~ # source /usr/sap/XSC/home/.sapenv.sh
root@atgvmls866:/usr/sap/XSC/HDB00> cd /tmp/HANACleaner/
root@atgvmls866:/tmp/HANACleaner>
root@atgvmls866:/tmp/HANACleaner> python hanacleaner.py -k T1CLEANKEY -be 400 -os true
Will now check most used memory in the file systems. If it hangs there is an issue with
s flag helps.
The most used filesystem is using
48%
************************************************************
2019-11-29 14:13:52
hanacleaner as root by T1CLEANKEY on XSC(00)  with
hanacleaner.py -k T1CLEANKEY -be 400 -os true
Cleanup Statements will be executed (-es is default true)
Before using HANACleaner read the disclaimer!
python hanacleaner.py --disclaimer
************************************************************
SELECT * from DUMMY
0 data backup entries and 0 log backup entries were removed from the backup catalog
```

House Keeping

HANACleaner – Interval

Run hanacleaner “forever” with the –hci flag

Flag	Unit	Details	Explanation	Default
-hci	Days	hanacleaner interval	After these number days hanacleaner will restart	-1 (exits)

Example:
(tries to clean trace files older than 400 days again after 1 day):

```
oqladm@ls80010:/tmp/HANACleaner> python hanacleaner.py -tc 400 -hci 1
The most used filesystem is using
80 %
**********************
2017-07-02 20:18:09
hanacleaner by SYSTEMKEY
**********************
    (Cleaning of the backup catalog was not done since -be and -bd were both negative (or not specified))
23 trace files were removed
    (Compression of the backup logs was not done since -zb was negative (or not specified))
    (Cleaning of the alerts was not done since -ar was negative (or not specified))
    (Cleaning of the object history was not done since -om was negative (or not specified))
    (Reclaim of free logsements was not done since -lr was negative (or not specified))
    (Cleaning of events was not done since -eh and -eu were negative (or not specified))
    (Defragmentation was not done since -fl was negative (or not specified))
    (Reclaim of row store containers were not done since -rc was negative (or not specified))
    (Cleaning of the hanacleaner logs was not done since -or was negative (or not specified))
**********************
2017-07-03 20:19:49
hanacleaner by SYSTEMKEY
**********************
    (Cleaning of the backup catalog was not done since -be and -bd were both negative (or not specified))
0 trace files were removed
    (Compression of the backup logs was not done since -zb was negative (or not specified))
```

House Keeping

HANACleaner & CRON

Do not use together with -hci flag!

HANACleaner can be scheduled with CRON to do cleanup e.g once per day

Note: hanacleaner expects the environment of <sid>adm → same environment as <sid>adm has to be provided to use CRON

Example: In /etc/passwd it is specified what environment <sid>adm is using, here bash:

```
oqladm@ls80010:/tmp/HANACleaner> grep oqladm /etc/passwd
oqladm:x:1001:1002:SAP HANA Database System Administrator:/home/oqladm:/bin/bash
```

This shell script, hanacleaner.sh, provides the <sid>adm environment, with **`source $HOME/.bashrc`** and then executes the hanacleaner command:

```
oqladm@ls80010:/tmp/HANACleaner> vi hanacleaner.sh
#! /bin/bash
source $HOME/.bashrc
python /tmp/HANACleaner/hanacleaner.py -be 100 -bo true -op /tmp/hanacleaneroutput
```

Then a new crontab can be created, calling this shell script, e.g. once every night at 1 o'clock:

```
oqladm@ls80010:/tmp/HANACleaner> crontab -e
0 1 * * * /tmp/HANACleaner/hanacleaner.sh
```

Note: if you want to log the output to std_out set up the crontab like this:

```
oqladm@ls80010:/tmp/HANACleaner> crontab -e
0 1 * * * /tmp/HANACleaner/hanacleaner.sh >> /tmp/HANACleaner/hanacleaner.log 2>&1
```

Hint 1: **00 18 * * 0** → execute 18:00 Sundays Hint 2: 00 18 * * * 0 su - <sid>adm –c "python ..hanacleaner.py … " as root → no need to source

House Keeping

HANACleaner & CRON with Lock (e.g. for Scale-Out)

HANACleaner can be scheduled with CRON to run in background, so that the terminal can be closed

Note: hanacleaner expects the environment of <sid>adm → source /bin/bash (or equivalent)

Note: cron can try to start hanacleaner e.g. every week, but thanks to a lock, it will not start until the first process stopped

This shell script, hanacleaner.sh, provides the <sid>adm environment, with **`source $HOME/.bashrc`** and then executes hanacleaner:

```
xscadm@atgvmls866:/tmp/HANACleaner> more hanacleaner.sh
#! /bin/bash
source $HOME/.bashrc
python /tmp/HANACleaner/hanacleaner.py -ff /tmp/HANACleaner/hanacleaner_conf.txt
```

(Note: Check, by manually executing ./hanacleaner.sh, that it works. If not, there might be a hidden ^M originating from using VI in dos mode. This can be solved by :set ff=unix in VI as described in this forum.)

Then a new crontab can be created, calling this shell script regularly (in this example 0 0 * * 0 = once a week), but only if the previous lock is gone:

```
hsiadm@atgvmls7071:/tmp/HANACleaner> crontab -e
crontab: installing new crontab
hsiadm@atgvmls7071:/tmp/HANACleaner> crontab -l
0 0 * * 0 /usr/bin/flock -xn /tmp/HANACleaner/hanacleaner.lck -c "/tmp/HANACleaner/hanacleaner.sh"
```

It is possible with ps to find that hanacleaner is running in background (but maybe not as easy as with hanasitter):

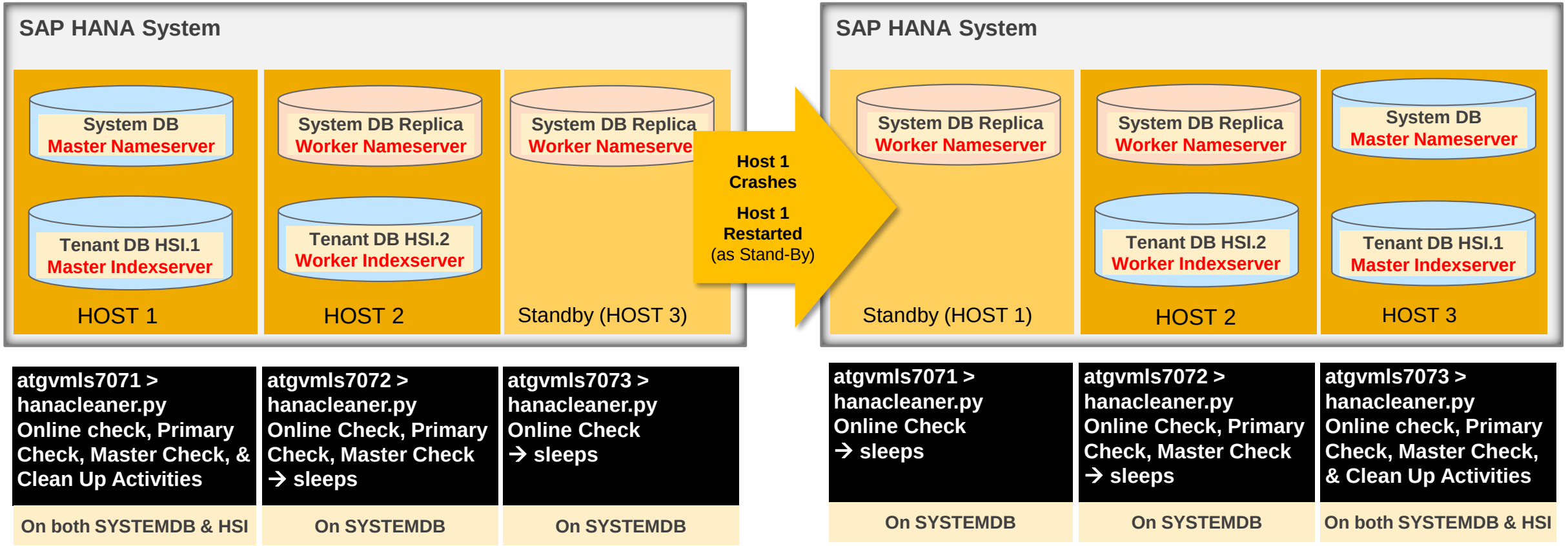
```
xscadm@atgvmls866:/tmp/HANACleaner> ps aux | grep cleaner
xscadm    8899  0.0  0.0   8752  1484 ?        Ss   14:54   0:00 /usr/bin/flock -xn /tmp/HANACleaner/hanacleaner.lck -c /tmp/HANACleaner/hanacleaner.sh
xscadm    8901  0.0  0.0  12136  2940 ?        S    14:54   0:00 /bin/bash /tmp/HANACleaner/hanacleaner.sh
xscadm    8931  3.7  0.0  23812 11092 ?        S    14:54   0:00 python /tmp/HANACleaner/hanacleaner.py -ff /tmp/HANACleaner/hanacleaner_conf.txt
```

House Keeping

HANACleaner – Scale Out Example (1/10)

Scenario for this example:

HANACleaner should only be used on the master node. However, in case stand-by node(s), the master node can change. Therefore a HANACleaner instance could run on each node, but should only be active on the current master:



House Keeping

HANACleaner – Scale Out Example (2/10)

On the first host, atgvmls7071, HANACleaner performs the house keeping activities after the online check, primary check and the master check, on both the System DB and the Tenant, HSI

Example on Master Node:
Here HANACleaner checks that it is online, primary and master and then cleans up trace files older than 100 days (-tc) for both SYSTEMDB and the tenant HSI (-dbs). It puts the logs in /tmp/hanacleaner_output (-op) and removes the logs that are older than 10 days (-or). Once all this is done, the hanacleaner instance stops. CRONTAB will start this at 00:00 every Sunday (0 0 * * 0).

```
hsiadm@atgvmls7071:/tmp/HANACleaner> more hanacleaner_conf.txt
-tc 100
-oi 60
-os true
-op /tmp/hanacleaner_output/
-or 10
-dbs SYSTEMDB,HSI
-k SDBKEY
hsiadm@atgvmls7071:/tmp/HANACleaner> more hanacleaner.sh
#! /bin/bash
source $HOME/.bashrc
python /tmp/HANACleaner/hanacleaner.py -ff /tmp/HANACleaner/hanacleaner_conf.txt
```

```
hsiadm@atgvmls7071:/tmp/HANACleaner> hdbuserstore LIST SDBKEY
KEY SDBKEY
  ENV : atgvmls7071:30013
  USER: SYSTEM   *
  DATABASE: SYSTEMDB
```

* Storing SYSTEM in hdbuserstore is ofcourse unacceptable from security point of view! Instead I should have created a HANACLEANERUSER, with proper privileges, in SystemDB and used that user in SDBKEY instead.

```
hsiadm@atgvmls7071:/tmp/HANACleaner> crontab -e
crontab: installing new crontab
hsiadm@atgvmls7071:/tmp/HANACleaner> crontab -l
0 0 * * 0 /usr/bin/flock -xn /tmp/HANACleaner/hanacleaner.lck -c "/tmp/HANACleaner/hanacleaner.sh"
```

(see next slide for the log)

House Keeping

HANACleaner – Scale Out Example (3/10)

Example on Master Node (continue):
Here HANACleaner checks that it is online, primary and master and then cleans up trace files older than 100 days (-tc) for both SYSTEMDB and the tenant HSI (-dbs). It puts the logs in /tmp/hanacleaner_output (-op) and removes the logs that are older than 10 days (-or). Once all this is done, the hanacleaner instance stops. CRONTAB will start this at 00:00 every Sunday (0 0 * * 0).

Here we are checking the log:

```
hsiadm@atgvmls7071:/tmp/HANACleaner> more /tmp/hanacleaner_output/hanacleanerlog_2021-08-03.txt
hanacleaner as hsiadm by SDBKEY on HSI(00) on DB SYSTEMDB with
Online Check      , 2021-08-03 21:18:02    ,     -             , True         , True       , # index services: 1, # running s
Primary Check     , 2021-08-03 21:18:03    ,     -             , True         , True       ,
Master Check      , 2021-08-03 21:18:04    ,     -             , True         , True       , Nameserver actual role = master
ALTER SYSTEM CLEAR TRACES ('ALERT','CLIENT','CRASHDUMP','EMERGENCYDUMP','RTEDUMP','UNLOAD','ROWSTOREREORG','SQLTRACE','*') UNTIL '2021-04-25 21:1
8 trace files were removed
hanacleaner as hsiadm by SDBKEY on HSI(00) on DB HSI with
Online Check      , 2021-08-03 21:18:05    ,     -             , True         , True       , # index services: 1, # running s
Primary Check     , 2021-08-03 21:18:06    ,     -             , True         , True       ,
Master Check      , 2021-08-03 21:18:07    ,     -             , True         , True       , Nameserver actual role = master
ALTER SYSTEM CLEAR TRACES ('ALERT','CLIENT','CRASHDUMP','EMERGENCYDUMP','RTEDUMP','UNLOAD','ROWSTOREREORG','SQLTRACE','*') UNTIL '2021-04-25 21:1
0 trace files were removed
```

House Keeping

HANACleaner – Scale Out Example (4/10)

On the second host, atgvmls7072, the master checks puts HANACleaner to sleep for a while

Example on Slave Node:
Here HANACleaner checks that it is online, primary and master. Since it is not the master, HANACleaner sleeps for one minute (-oi) and then checks again. It puts the logs under /tmp/hanacleaner_output (-op). CRONTAB tries to start another HANACleaner instance every Sunday, but will not succeed due to the lock "hanacleaner.lck".

```
hsiadm@atgvmls7072:/tmp/HANACleaner> more hanacleaner_conf.txt
-tc 100
-oi 60
-os true
-op /tmp/hanacleaner_output/
-or 10
-dbs SYSTEMDB,HSI
-k SDBKEY
hsiadm@atgvmls7072:/tmp/HANACleaner> more hanacleaner.sh
#! /bin/bash
source $HOME/.bashrc
python /tmp/HANACleaner/hanacleaner.py -ff /tmp/HANACleaner/hanacleaner_conf.txt
```

```
hsiadm@atgvmls7072:/tmp/HANACleaner> hdbuserstore LIST SDBKEY
KEY SDBKEY
  ENV : atgvmls7072:30013
  USER: SYSTEM   *
  DATABASE: SYSTEMDB
```

* Storing SYSTEM in hdbuserstore is ofcourse unacceptable from security point of view! Instead I should have created a HANACLEANERUSER, with proper privileges, in SystemDB and used that user in SDBKEY instead.

```
hsiadm@atgvmls7072:/tmp/HANACleaner> crontab -e
crontab: installing new crontab
hsiadm@atgvmls7072:/tmp/HANACleaner> crontab -l
0 0 * * 0 /usr/bin/flock -xn /tmp/HANACleaner/hanacleaner.lck -c "/tmp/HANACleaner/hanacleaner.sh"
```

(see next slide for the log)

House Keeping

HANACleaner – Scale Out Example (5/10)

Example on Slave Node (continue):
Here HANACleaner checks that it is online, primary and master. Since it is not the master, HANACleaner sleeps for one minute (-oi) and then checks again. It puts the logs under /tmp/hanacleaner_output (-op). CRONTAB tries to start another HANACleaner instance every Sunday, but will not succeed due to the lock "hanacleaner.lck".

Here we are checking the log:

```
hsiadm@atgvmls7072:/tmp/HANACleaner> more /tmp/hanacleaner_output/hanacleanerlog_2021-08-03.txt
hanacleaner as hsiadm by SDBKEY on HSI(00) on DB SYSTEMDB with
Online Check      , 2021-08-03 22:04:01    ,     -           , True         , True       , # index services: 1, # running services: 8 out
Primary Check     , 2021-08-03 22:04:02    ,     -           , True         , True       ,
Master Check      , 2021-08-03 22:04:03    ,     -           , True         , False       , Nameserver actual role = slave

One of the online checks found out that this HANA instance, 00, is not online or not master.
HANACleaner will now have a 60 seconds break and check again if this Instance is online, or master, after the break.

Online Check      , 2021-08-03 22:05:03    ,     -           , True         , True       , # index services: 1, # running services: 8 out
Primary Check     , 2021-08-03 22:05:04    ,     -           , True         , True       ,
Master Check      , 2021-08-03 22:05:05    ,     -           , True         , False       , Nameserver actual role = slave

One of the online checks found out that this HANA instance, 00, is not online or not master.
HANACleaner will now have a 60 seconds break and check again if this Instance is online, or master, after the break.
```

This hanacleaner instance is running continuously in the background:

```
hsiadm@atgvmls7072:/tmp/HANACleaner> ps aux | grep cleaner
hsiadm   19641  0.0  0.0   8752  1480 ?        Ss   22:04   0:00 /usr/bin/flock -xn /tmp/HANACleaner/hanacleaner.lck -c /tmp/HANACleaner/hanacleaner.sh
hsiadm   19642  0.0  0.0  12132  2844 ?        S    22:04   0:00 /bin/bash /tmp/HANACleaner/hanacleaner.sh
hsiadm   19666  0.1  0.0  24036 11404 ?        S    22:04   0:00 python /tmp/HANACleaner/hanacleaner.py -ff /tmp/HANACleaner/hanacleaner_conf.txt
```

House Keeping

HANACleaner – Scale Out Example (6/10)

On the third host, atgvmls7073, the online checks puts HANACleaner to sleep for a while

Example on Stand-By Node:
Here HANACleaner checks that it is online. Since it is not (there is no indexserver), HANACleaner sleeps for one minute (-oi) and then checks again. It puts the logs under /tmp/hanacleaner_output (-op). CRONTAB tries to start another HANACleaner instance every Sunday, but will not succeed due to the lock "hanacleaner.lck".

```
hsiadm@atgvmls7073:/tmp/HANACleaner> more hanacleaner_conf.txt
-tc 100
-oi 60
-os true
-op /tmp/hanacleaner_output/
-or 10
-dbs SYSTEMDB,HSI
-k SDBKEY
hsiadm@atgvmls7073:/tmp/HANACleaner> more hanacleaner.sh
#! /bin/bash
source $HOME/.bashrc
python /tmp/HANACleaner/hanacleaner.py -ff /tmp/HANACleaner/hanacleaner_conf.txt
```

```
hsiadm@atgvmls7073:/tmp/HANACleaner> hdbuserstore LIST SDBKEY
KEY SDBKEY
  ENV : atgvmls7073:30013
  USER: SYSTEM   *
  DATABASE: SYSTEMDB
```

* Storing SYSTEM in hdbuserstore is ofcourse unacceptable from security point of view! Instead I should have created a HANACLEANERUSER, with proper privileges, in SystemDB and used that user in SDBKEY instead.

```
hsiadm@atgvmls7073:/tmp/HANACleaner> crontab -e
crontab: installing new crontab
hsiadm@atgvmls7073:/tmp/HANACleaner> crontab -l
0 0 * * 0 /usr/bin/flock -xn /tmp/HANACleaner/hanacleaner.lck -c "/tmp/HANACleaner/hanacleaner.sh"
```

(see next slide for the log)

House Keeping

HANACleaner – Scale Out Example (7/10)

Example on Stand-By Node (continue):
Here HANACleaner checks that it is online. Since it is not (there is no indexserver), HANACleaner sleeps for one minute (-oi) and then checks again. It puts the logs under /tmp/hanacleaner_output (-op). CRONTAB tries to start another HANACleaner instance every Sunday, but will not succeed due to the lock "hanacleaner.lck".

Here we are checking the log:

```
hsiadm@atgvmls7073:/tmp/HANACleaner> more /tmp/hanacleaner_output/hanacleanerlog_2021-08-03.txt
hanacleaner as hsiadm by SDBKEY on HSI(00) on DB SYSTEMDB with
Online Check      , 2021-08-03 22:17:01    ,     -           , True         , False       , # index services: 0, # running services: 5 out

One of the online checks found out that this HANA instance, 00, is not online or not master.
HANACleaner will now have a 60 seconds break and check again if this Instance is online, or master, after the break.

Online Check      , 2021-08-03 22:18:01    ,     -           , True         , False       , # index services: 0, # running services: 5 out

One of the online checks found out that this HANA instance, 00, is not online or not master.
HANACleaner will now have a 60 seconds break and check again if this Instance is online, or master, after the break.
```

This hanacleaner instance is running continuously in the background:

```
hsiadm@atgvmls7073:/tmp/HANACleaner> ps aux | grep cleaner
hsiadm   15410  0.0  0.0   8752  1400 ?        Ss   22:17   0:00 /usr/bin/flock -xn /tmp/HANACleaner/hanacleaner.lck -c /tmp/HANACleaner/hanacleaner.sh
hsiadm   15411  0.0  0.0  12132  3020 ?        S    22:17   0:00 /bin/bash /tmp/HANACleaner/hanacleaner.sh
hsiadm   15435  0.0  0.0  24036 11404 ?        S    22:17   0:00 python /tmp/HANACleaner/hanacleaner.py -ff /tmp/HANACleaner/hanacleaner_conf.txt
```

House Keeping

HANACleaner – Scale Out Example (8/10)

If first host, atgvmls7071, crashes, a failover to third host, atgvmls7073, will happen

Example:
Here first host, atgvmls7071, is stopped (HDB stop) to simulate a crash. Third host, atgvmls7073, takes over as the master node

SYSTEMDB@HSI (SYSTEM) atgvmls7071.wdf.sap.corp 00 Last Update: 03.08.2021 22:2

Overview | Landscape | Alerts | Performance | Volumes | Configuration | System Information | Diagnosis Files | Trace Configuration

Services | Hosts | Redistribution | System Replication

Host	Active	Host Status	Name Server Role (Configured)	Name Server Role (Actual)	Index Server Role (Configured)	Index Server Role (Actual)
atgvmls7071	YES	OK	MASTER 1	MASTER	WORKER	MASTER
atgvmls7072	YES	OK	MASTER 3	SLAVE	WORKER	SLAVE
atgvmls7073	YES	IGNORE	MASTER 2	SLAVE	STANDBY	STANDBY

```
hsiadm@atgvmls7071:/tmp/HANACleaner> HDB stop
```

```
hsiadm@atgvmls7072:/usr/sap/HSI/HDB00/exe/python_support> python landscapeHostConfiguration.py
| Host        | Host   | Host   | Failover | Remove | Storage   | Storage   | Failover | Failover | NameServer | NameServer | IndexServer | IndexServer |
|             | Active | Status | Status   | Status | Config    | Actual    | Config   | Actual   | Config     | Actual     | Config      | Actual      |
|             |        |        |          |        | Partition | Partition | Group    | Group    | Role       | Role       | Role        | Role        |
| ----------- | ------ | ------ | -------- | ------ | --------- | --------- | -------- | -------- | ---------- | ---------- | ----------- | ----------- |
| atgvmls7071 | no     | info   |          |        |         1 |         0 | default  | default  | master 1   | slave      | worker      | standby     |
| atgvmls7072 | yes    | ok     |          |        |         2 |         2 | default  | default  | master 3   | slave      | worker      | slave       |
| atgvmls7073 | yes    | info   |          |        |         0 |         1 | default  | default  | master 2   | master     | standby     | master      |
```

When the first host, atgvmls7071, starts up again,
it is now the stand by node:

```
hsiadm@atgvmls7071:/tmp/HANACleaner> HDB start
```

```
hsiadm@atgvmls7071:/usr/sap/HSI/HI
| Host        | Host   | Host   |
|             | Active | Status |
| ----------- | ------ | ------ |
| atgvmls7071 | yes    | info   |
| atgvmls7072 | yes    | ok     |
| atgvmls7073 | yes    | info   |

| NameServer | IndexServer | IndexServer |
| Actual     | Config      | Actual      |
| Role       | Role        | Role        |
| ---------- | ----------- | ----------- |
| slave      | worker      | standby     |
| slave      | worker      | slave       |
| master     | standby     | master      |
```

House Keeping

HANACleaner – Scale Out Example (9/10)

The new master node, atgvmls7073, takes over the HANACleaner house keeping tasks

Example on the new Master Node:
Here HANACleaner slept since it was before not online. Once it became the Master Node HANACleaner checks that it is online, primary and master, then cleans up trace files older than 100 days (-tc) for both SYSTEMDB and the tenant HSI (-dbs). It puts the logs in /tmp/hanacleaner_output (-op) and removes the logs that are older than 10 days (-or). Once all this is done, the hanacleaner instance stops. CRONTAB will start this at 00:00 every Sunday (0 0 * * 0).

```
One of the online checks found out that this HANA instance, 00, is not online or not master.
HANACleaner will now have a 60 seconds break and check again if this Instance is online, or master, after the break.

Online Check      , 2021-08-03 22:28:02    ,     -           , True         , True       , # index services: 1, # running services: 8 out of 8
Primary Check     , 2021-08-03 22:28:04    ,     -           , True         , True       ,
Master Check      , 2021-08-03 22:28:05    ,     -           , True         , True       , Nameserver actual role = master
SELECT * from DUMMY
    (Cleaning of the backup catalog was not done since -be and -bd were both negative (or not specified))
ALTER SYSTEM CLEAR TRACES ('ALERT','CLIENT','CRASHDUMP','EMERGENCYDUMP','RTEDUMP','UNLOAD','ROWSTOREREORG','SQLTRACE','*') UNTIL '2021-04-25 22:2
0 trace files were removed
```

```
hanacleaner as hsiadm by SDBKEY on HSI(00) on DB HSI with
```

```
Online Check      , 2021-08-03 22:28:05    ,     -           , True         , True       , # index services: 1, # running services: 8 out of 8
Primary Check     , 2021-08-03 22:28:07    ,     -           , True         , True       ,
Master Check      , 2021-08-03 22:28:08    ,     -           , True         , True       , Nameserver actual role = master
SELECT * from DUMMY
    (Cleaning of the backup catalog was not done since -be and -bd were both negative (or not specified))
ALTER SYSTEM CLEAR TRACES ('ALERT','CLIENT','CRASHDUMP','EMERGENCYDUMP','RTEDUMP','UNLOAD','ROWSTOREREORG','SQLTRACE','*') UNTIL '2021-04-25 22:2
0 trace files were removed
```

House Keeping

HANACleaner – Scale Out Example (10/10)

The first node, atgvmls7071, is now the new stand-by, and will start online checking next Sunday

Example on the new Stand-By Node:
If the first node is started again it starts up as the new stand-by node. At 00:00 the next Sunday CRONTAB will start a new hanacleaner instance which will check each minute (-oi) if it is online: Here we are checking the log:

```
hsiadm@atgvmls7071:/tmp/HANACleaner> more /tmp/hanacleaner_output/hanacleanerlog_2021-08-03.txt

hanacleaner as hsiadm by SDBKEY on HSI(00) on DB SYSTEMDB with

Online Check      , 2021-08-03 23:01:01    ,     -           , True         , False       , # index services: 0, # running services: 5 o

One of the online checks found out that this HANA instance, 00, is not online or not master.
HANACleaner will now have a 60 seconds break and check again if this Instance is online, or master, after the break.

Online Check      , 2021-08-03 23:02:02    ,     -           , True         , False       , # index services: 0, # running services: 5 o

One of the online checks found out that this HANA instance, 00, is not online or not master.
HANACleaner will now have a 60 seconds break and check again if this Instance is online, or master, after the break.
```

Example on the Slave Node:
The second node is still the slave node, so the same hanacleaner instance continues sleeping one minute (-oi) after it finds out that it is not the master:

```
Online Check      , 2021-08-03 22:54:35    ,     -           , True         , True        , # index services: 1, # running services: 8 o
Primary Check     , 2021-08-03 22:54:37    ,     -           , True         , True        ,
Master Check      , 2021-08-03 22:54:37    ,     -           , True         , False        , Nameserver actual role = slave

One of the online checks found out that this HANA instance, 00, is not online or not master.
HANACleaner will now have a 60 seconds break and check again if this Instance is online, or master, after the break.
```

House Keeping

HANACleaner & CRON with Lock - Terminate

If HANACleaner was started with cron as in previous example, it will continue until the crontab is changed and, for the two non-master nodes, after the running hanacleaner instances are killed

Edit the crontab:

```
hsiadm@atgvmls7071:/tmp/HANACleaner> crontab -e
```

```
0 0 * * 0 /usr/bin/flock -xn /tmp/HANACleaner/hanacleaner.lck -c "/tmp/HANACleaner/hanacleaner.sh"
```

E.g. comment out the line that is restarting HANACleaner:

```
#0 0 * * 0 /usr/bin/flock -xn /tmp/HANACleaner/hanacleaner.lck -c "/tmp/HANACleaner/hanacleaner.sh"
```

Then exit vim by saving this change (esc : wq):

```
crontab: installing new crontab
```

Now we can kill the current running HANACleaner by first checking the PIDs:

```
hsiadm@atgvmls7071:/tmp/HANACleaner> ps aux | grep cleaner
hsiadm   17980  0.0  0.0   8752  1496 ?        Ss   23:01   0:00 /usr/bin/flock -xn /tmp/HANACleaner/hanacleaner.lck -c /tmp/HANACleaner/hanacleaner.sh
hsiadm   17981  0.0  0.0  12132  2988 ?        S    23:01   0:00 /bin/bash /tmp/HANACleaner/hanacleaner.sh
hsiadm   18008  0.0  0.0  24036 11320 ?        S    23:01   0:00 python /tmp/HANACleaner/hanacleaner.py -ff /tmp/HANACleaner/hanacleaner_conf.txt
```

Then stopping hanasitter by using kill -9:

```
hsiadm@atgvmls7071:/tmp/HANACleaner> kill -9 17980 17981 18008
```

Double check that no HANACleaner restarted:

```
hsiadm@atgvmls7071:/tmp/HANACleaner> ps aux | grep cleaner
hsiadm   20377  0.0  0.0  10548  1720 pts/1    S+   23:23   0:00 grep --color=auto cleaner
```

House Keeping

HANACleaner – list all flags

The --help flag lists all possible input flags:

```
xscadm@atgvmls866:/tmp/HANACleaner> python hanacleaner.py --help

DESCRIPTION:
 The HANA cleaner is a house keeping service for SAP HANA. It can be used to clean the backup catalog, diagnostic files,
 and alerts and to compress the backup logs. It should be executed by <sid>adm or, in case you use a CRON job, with the same
 environment as the <sid>adm. See SAP Note 2399996 and SAP Note 2400024.

INPUT ARGUMENTS:
         ----  BACKUP ENTRIES in BACKUP CATALOG (and possibly BACKUPS)  ----
 -be     minimum retained number of data backup (i.e. complete data backups and data snapshots) entries in the catalog, this
         number of entries of data backups will remain in the backup catalog, all older log backup entries will also be removed
         with BACKUP CATALOG DELETE BACKUP_ID <id> (see SQL reference for more info) default: -1 (not used)
 -bd     min retained days of data backup (i.e. complete data backups and data snapshots) entries in the catalog [days], the
```

Always get latest HANACleaner as the flags keep evolving!