

Logging with TRACE from Containers pocket and podman status

Ron Rechenmacher

CCM Meeting

6 October 2021

Initial Setup

- Be mindful of container storage on NFS – adjust `~/.config/containers/storage.conf` to local filesystem
- Check `max_user_namespaces` – most distros will have reasonable value (15K)
- Admin needed to adjust `/etc/sub{u,g}id`
- Versions – podman v1.4 seems too old

TRACE_FILE – accessible from container, persistent on host.

Red = for podman.

Docker needs `su` or `via docker group`.

Container storage seems to be an issue for both podman and docker (but this may be a version issue).

Podman example – 3 containers

```
/home/ron
ron@mu2edaql1:~$ rm -f $TRACE_FILE; tmodeS 0; for cc in `seq 3`;do
> podman run --rm -d --privileged --pid=host -v /tmp:/tmp -v ${PRODUCTS/*:}/:p scientificlinux/sl bash -c "\
> export UPS_OVERRIDE='-H `ups flavor`' TRACE_FILE=$TRACE_FILE;\
> . /p/setup; setup TRACE $TRACE_VERSION;\
> example_main -nex `hostname`; \
> done; sleep 3; tshow | egrep '^([idx|---])[example_sub4.*again' | PAGER= tdelta -ct 1 -d 1
9bab8dc271c5896d7588db4548f1178023a0c97e4c6cd9669ed1862b70744c75
ab056ff6c8711b3e9be78ff3e714f6e43ba6077d574dc64c22d421aa8bc69e17
415bb611a51649e3eabd7ca89516f7e35c51487f15c5ccacd32f5f07894890ca
idx      us_tod      delta      pid      tid cpu      trcname lvl msg
-----
  9 10-06 00:25:38.922224      0 3434934 3434936 30 ex_415bb611a516 NFO example_sub4: hello again from example_sub4 ()
 14 10-06 00:25:38.922221      3 3434934 3434937 44 ex_415bb611a516 NFO example_sub4: hello again from example_sub4 ()
 16 10-06 00:25:38.922220      1 3434934 3434935 48 ex_415bb611a516 NFO example_sub4: hello again from example_sub4 ()
 40 10-06 00:25:38.922178      42 3434934 3434937 44 ex_415bb611a516 NFO example_sub4: hello again from example_sub4 ()
 41 10-06 00:25:38.922178      0 3434934 3434936 30 ex_415bb611a516 NFO example_sub4: hello again from example_sub4 ()
 43 10-06 00:25:38.922174      4 3434934 3434935 43 ex_415bb611a516 NFO example_sub4: hello again from example_sub4 ()
 66 10-06 00:25:38.922140      34 3434934 3434936 30 ex_415bb611a516 NFO example_sub4: hello again from example_sub4 ()
 68 10-06 00:25:38.922132      8 3434934 3434935 43 ex_415bb611a516 NFO example_sub4: hello again from example_sub4 ()
 70 10-06 00:25:38.922132      0 3434934 3434937 44 ex_415bb611a516 NFO example_sub4: hello again from example_sub4 ()
 94 10-06 00:25:38.922026     106 3434934 3434937 44 ex_415bb611a516 NFO example_sub4: hello again from example_sub4 ()
 96 10-06 00:25:38.922020      6 3434934 3434935 43 ex_415bb611a516 NFO example_sub4: hello again from example_sub4 ()
 97 10-06 00:25:38.922019      1 3434934 3434936 30 ex_415bb611a516 NFO example_sub4: hello again from example_sub4 ()
118 10-06 00:25:38.142462    779557 3434605 3434608 44 ex_ab056ff6c871 NFO example_sub4: hello again from example_sub4 ()
133 10-06 00:25:38.142436     26 3434605 3434607 31 ex_ab056ff6c871 NFO example_sub4: hello again from example_sub4 ()
136 10-06 00:25:38.142432      4 3434605 3434606 48 ex_ab056ff6c871 NFO example_sub4: hello again from example_sub4 ()
138 10-06 00:25:38.142431      1 3434605 3434608 44 ex_ab056ff6c871 NFO example_sub4: hello again from example_sub4 ()
148 10-06 00:25:38.142418     13 3434605 3434607 31 ex_ab056ff6c871 NFO example_sub4: hello again from example_sub4 ()
167 10-06 00:25:38.142396     22 3434605 3434606 43 ex_ab056ff6c871 NFO example_sub4: hello again from example_sub4 ()
169 10-06 00:25:38.142394      2 3434605 3434607 31 ex_ab056ff6c871 NFO example_sub4: hello again from example_sub4 ()
186 10-06 00:25:38.142370     24 3434605 3434606 43 ex_ab056ff6c871 NFO example_sub4: hello again from example_sub4 ()
187 10-06 00:25:38.142370      0 3434605 3434608 44 ex_ab056ff6c871 NFO example_sub4: hello again from example_sub4 ()
207 10-06 00:25:38.142271     99 3434605 3434607 31 ex_ab056ff6c871 NFO example_sub4: hello again from example_sub4 ()
208 10-06 00:25:38.142267      4 3434605 3434606 43 ex_ab056ff6c871 NFO example_sub4: hello again from example_sub4 ()
209 10-06 00:25:38.142265      2 3434605 3434608 44 ex_ab056ff6c871 NFO example_sub4: hello again from example_sub4 ()
231 10-06 00:25:37.453097    689168 3434223 3434224 29 ex_9bab8dc271c5 NFO example_sub4: hello again from example_sub4 ()
244 10-06 00:25:37.453064     33 3434223 3434225 44 ex_9bab8dc271c5 NFO example_sub4: hello again from example_sub4 ()
245 10-06 00:25:37.453060      4 3434223 3434224 29 ex_9bab8dc271c5 NFO example_sub4: hello again from example_sub4 ()
252 10-06 00:25:37.453050     10 3434223 3434226 80 ex_9bab8dc271c5 NFO example_sub4: hello again from example_sub4 ()
267 10-06 00:25:37.453029     21 3434223 3434226 30 ex_9bab8dc271c5 NFO example_sub4: hello again from example_sub4 ()
274 10-06 00:25:37.453012     17 3434223 3434225 44 ex_9bab8dc271c5 NFO example_sub4: hello again from example_sub4 ()
288 10-06 00:25:37.452997     15 3434223 3434226 30 ex_9bab8dc271c5 NFO example_sub4: hello again from example_sub4 ()
293 10-06 00:25:37.452996      1 3434223 3434225 44 ex_9bab8dc271c5 NFO example_sub4: hello again from example_sub4 ()
298 10-06 00:25:37.452982     14 3434223 3434224 29 ex_9bab8dc271c5 NFO example_sub4: hello again from example_sub4 ()
320 10-06 00:25:37.452905     77 3434223 3434226 30 ex_9bab8dc271c5 NFO example_sub4: hello again from example_sub4 ()
322 10-06 00:25:37.452902      3 3434223 3434224 29 ex_9bab8dc271c5 NFO example_sub4: hello again from example_sub4 ()
324 10-06 00:25:37.452896      6 3434223 3434225 44 ex_9bab8dc271c5 NFO example_sub4: hello again from example_sub4 ()
--2021-10-06_00:25:41_CDT--
```

- Podman options:

- privileged
allow write

- pid=host
no pid collisions

- v
get access to TRACE_FILE

- Use TRACE_NAME to distinguish containers/instances

Red = same as docker

Podman example – 3 containers, NFS (/home)

```
/home/ron
ron@mu2edaq11 (^) TRACE_FILE=$HOME/trace_${HOSTNAME}
--2021-10-06_01:15:49_CDT--
/home/ron
ron@mu2edaq11 (^) rm -f $TRACE_FILE; tmodes 0; for cc in `seq 3`;do podman run --rm -d --privileged --pid=host -v /tmp:/tmp
-v /home:/home -v ${PRODUCTS/*}/:/p scientificlinux/sl bash -c "\
  export UPS_OVERRIDE='-H `ups flavor`' TRACE_FILE=$TRACE_FILE;\
  . /p/setup; setup TRACE $TRACE_VERSION;\
  example_main -nex \${HOSTNAME}";done; sleep 3; tshow | egrep '^(idx|---)|example_sub4.*again' | PAGER= tdelta -ct 1 -d 1
75e70f00887ae14a660974d2d73738efaf615e26b289584d7b8bd66a58f6b13e
7dd42fd941c8c6c2d4cee14765ef9ba0955ce91fa2b032d60004327b71cdd5b6
fb5cf7b4ef3d0437e8624fbbfa0e1f4e2c278ecd28efa689973fcb2b777ade59
idx      us_tod      delta      pid      tid cpu      trcname  lvl msg
-----
10 10-06 01:15:57.359652      0 3487915 3487922 29 ex_fb5cf7b4ef3d NFO example_sub4: hello again from example_sub4 ()
12 10-06 01:15:57.359651      1 3487915 3487923 30 ex_fb5cf7b4ef3d NFO example_sub4: hello again from example_sub4 ()
15 10-06 01:15:57.359650      1 3487915 3487924 44 ex_fb5cf7b4ef3d NFO example_sub4: hello again from example_sub4 ()
36 10-06 01:15:57.359631     19 3487915 3487923 30 ex_fb5cf7b4ef3d NFO example_sub4: hello again from example_sub4 ()
40 10-06 01:15:57.359628      3 3487915 3487924 44 ex_fb5cf7b4ef3d NFO example_sub4: hello again from example_sub4 ()
42 10-06 01:15:57.359627      1 3487915 3487922 29 ex_fb5cf7b4ef3d NFO example_sub4: hello again from example_sub4 ()
64 10-06 01:15:57.359609     18 3487915 3487923 30 ex_fb5cf7b4ef3d NFO example_sub4: hello again from example_sub4 ()
69 10-06 01:15:57.359600      9 3487915 3487924 44 ex_fb5cf7b4ef3d NFO example_sub4: hello again from example_sub4 ()
70 10-06 01:15:57.359600      0 3487915 3487922 29 ex_fb5cf7b4ef3d NFO example_sub4: hello again from example_sub4 ()
94 10-06 01:15:57.359509     91 3487915 3487922 29 ex_fb5cf7b4ef3d NFO example_sub4: hello again from example_sub4 ()
96 10-06 01:15:57.359498     11 3487915 3487924 44 ex_fb5cf7b4ef3d NFO example_sub4: hello again from example_sub4 ()
98 10-06 01:15:57.359493     5 3487915 3487923 30 ex_fb5cf7b4ef3d NFO example_sub4: hello again from example_sub4 ()
118 10-06 01:15:56.823681    535812 3487584 3487585   3 ex_7dd42fd941c8 NFO example_sub4: hello again from example_sub4 ()
131 10-06 01:15:56.823668     13 3487584 3487585   3 ex_7dd42fd941c8 NFO example_sub4: hello again from example_sub4 ()
132 10-06 01:15:56.823667      1 3487584 3487586   30 ex_7dd42fd941c8 NFO example_sub4: hello again from example_sub4 ()
151 10-06 01:15:56.823648     19 3487584 3487586   30 ex_7dd42fd941c8 NFO example_sub4: hello again from example_sub4 ()
155 10-06 01:15:56.823647      1 3487584 3487585   3 ex_7dd42fd941c8 NFO example_sub4: hello again from example_sub4 ()
158 10-06 01:15:56.823640      7 3487584 3487587 44 ex_7dd42fd941c8 NFO example_sub4: hello again from example_sub4 ()
174 10-06 01:15:56.823626     14 3487584 3487587 44 ex_7dd42fd941c8 NFO example_sub4: hello again from example_sub4 ()
175 10-06 01:15:56.823625      1 3487584 3487586   30 ex_7dd42fd941c8 NFO example_sub4: hello again from example_sub4 ()
191 10-06 01:15:56.823611     14 3487584 3487587 44 ex_7dd42fd941c8 NFO example_sub4: hello again from example_sub4 ()
207 10-06 01:15:56.823547     64 3487584 3487585   3 ex_7dd42fd941c8 NFO example_sub4: hello again from example_sub4 ()
209 10-06 01:15:56.823546      1 3487584 3487586   30 ex_7dd42fd941c8 NFO example_sub4: hello again from example_sub4 ()
211 10-06 01:15:56.823534     12 3487584 3487587 44 ex_7dd42fd941c8 NFO example_sub4: hello again from example_sub4 ()
235 10-06 01:15:56.014633    808901 3487215 3487217 30 ex_75e70f00887a NFO example_sub4: hello again from example_sub4 ()
236 10-06 01:15:56.014633      0 3487215 3487216   3 ex_75e70f00887a NFO example_sub4: hello again from example_sub4 ()
252 10-06 01:15:56.014603     30 3487215 3487217 30 ex_75e70f00887a NFO example_sub4: hello again from example_sub4 ()
255 10-06 01:15:56.014602      1 3487215 3487218 44 ex_75e70f00887a NFO example_sub4: hello again from example_sub4 ()
275 10-06 01:15:56.014566     36 3487215 3487218 44 ex_75e70f00887a NFO example_sub4: hello again from example_sub4 ()
276 10-06 01:15:56.014566      0 3487215 3487216   3 ex_75e70f00887a NFO example_sub4: hello again from example_sub4 ()
277 10-06 01:15:56.014565      1 3487215 3487217 30 ex_75e70f00887a NFO example_sub4: hello again from example_sub4 ()
293 10-06 01:15:56.014544     21 3487215 3487216   3 ex_75e70f00887a NFO example_sub4: hello again from example_sub4 ()
297 10-06 01:15:56.014543      1 3487215 3487218 44 ex_75e70f00887a NFO example_sub4: hello again from example_sub4 ()
320 10-06 01:15:56.014434    109 3487215 3487216   3 ex_75e70f00887a NFO example_sub4: hello again from example_sub4 ()
322 10-06 01:15:56.014430      4 3487215 3487218 44 ex_75e70f00887a NFO example_sub4: hello again from example_sub4 ()
323 10-06 01:15:56.014429      1 3487215 3487217 30 ex_75e70f00887a NFO example_sub4: hello again from example_sub4 ()
--2021-10-06_01:16:00_CDT--
```

- TRACE_FILE on NFS is OK
- Add -v /home:/home

Problems with pocket (build.sh)

- sed command in ./build.sh acts on "*" – errors on directory
- Remove buildx in Docker buildx build in ./build.sh
- TRACE_FILE currently different on host and in container
 - Edit rebuild_work_area.sh – set TRACE_FILE=/dunedaq/pocket/trace_buffer_xx
 - "_xx" can be partition and/or hostname, etc.
 - Maybe nice to have /tmp:/tmp or /home:/home

Versions

/nfs/home/ron

```
ron@np04-srv-030 :^) lsb_release -a
```

```
LSB Version:    :core-4.1-amd64:core-4.1-noarch:cxx-4.1-amd64:cxx-4.1-noarch:desktop-4.1-amd64:desktop-4.1-noarch:languages-4.1-amd64:languages-4.1-noarch:printing-4.1-amd64:printing-4.1-noarch
```

```
Distributor ID: CentOS
```

```
Description:   CentOS Linux release 7.6.1810 (Core)
```

```
Release:       7.6.1810
```

```
Codename:      Core
```

```
--2021-10-06_06:49:21_CEST--
```

/nfs/home/ron

```
ron@np04-srv-030 :^) podman --version
```

```
podman version 1.4.4
```

```
--2021-10-06_06:49:48_CEST--
```

Versions

/home/ron

```
ron@mu2edaq11 :^) lsb_release -a
```

```
LSB Version:    :core-4.1-amd64:core-4.1-noarch:cxx-4.1-amd64:cxx-4.1-noarch:desktop-4.1-amd64:desktop-4.1-noarch:languages-4.1-amd64:languages-4.1-noarch:printing-4.1-amd64:printing-4.1-noarch
```

```
Distributor ID: Scientific
```

```
Description:   Scientific Linux release 7.9 (Nitrogen)
```

```
Release: 7.9
```

```
Codename: Nitrogen
```

```
--2021-10-05_23:54:48_CDT--
```

/home/ron

```
ron@mu2edaq11 :^) podman --version
```

```
podman version 1.6.4
```

```
--2021-10-05_23:54:55_CDT--
```

Pocket example

```
-----  
idx      us_tod      pid      tid      cpu      trcname  lvl  msg  
-----  
0 1633510064603972      7      9      26      DAQModuleManager  DBG  init_modules: construct: FakeHSIEventGenerator : fhsig  
1 1633510064603964      7      9      26      DAQModuleManager  D02  init_queues: Adding queue: time_sync_from_netq  
2 1633510064603962      7      9      26      DAQModuleManager  D02  init_queues: Adding queue: hsievent_q_to_net  
3 1633510064603901      7      9      26      DAQModuleManager  D01  execute: Command id:init  
4 1633510064588517      7      8      3      RestEndpoint      LOG  handleResponseCommand: Response code = OK  
5 1633510064586368      7      9      0      RestEndpoint      LOG  handleResponseCommand: Sending POST request to nanorc.my-  
partition:56789/response  
6 1633510064586282      7      9      0      DAQModuleManager  DBG  init_modules: construct: QueueToNetwork : qton_datareq_1  
7 1633510064586274      7      9      0      DAQModuleManager  DBG  init_modules: construct: QueueToNetwork : qton_datareq_0  
8 1633510064586262      7      9      0      DAQModuleManager  DBG  init_modules: construct: NetworkToQueue : ntoq_fragments_7  
9 1633510064586259      7      9      0      DataWriter        D05  init: datawriter: Exiting init() method  
10 1633510064586257      7      9      0      DAQSink           D01  DAQSink: Queue token_q is at 0x7f3c2007cf00  
11 1633510064586203      7      9      0      DAQSource         D01  DAQSource: Queue trigger_record_q is at 0x7f3c2006e700  
12 1633510064586164      7      9      0      DataWriter        D05  init: datawriter: Entering init() method  
13 1633510064583412      7      9      0      DAQModuleManager  DBG  init_modules: construct: DataWriter : datawriter  
14 1633510064583407      7      9      0      TriggerRecordBuilder D05  init: trb: Exiting init() method  
15 1633510064583406      7      9      0      DAQSink           D01  DAQSink: Queue data_requests_1 is at 0x7f3c20075780  
16 1633510064583394      7      9      0      DAQSink           D01  DAQSink: Queue data_requests_0 is at 0x7f3c20071200  
17 1633510064583356      7      9      0      DAQSource         D01  DAQSource: Queue data_fragments_q is at 0x7f3c2006fc80  
18 1633510064583313      7      9      0      DAQSink           D01  DAQSink: Queue trigger_record_q is at 0x7f3c2006e700  
19 1633510064583285      7      9      0      DAQSource         D01  DAQSource: Queue trigger_decision_from_netq is at  
0x7f3c2006a180  
20 1633510064583167      7      9      0      TriggerRecordBuilder D05  init: trb: Entering init() method  
21 1633510062708662      7      9      7      DAQModuleManager  DBG  init_modules: construct: TriggerRecordBuilder : trb  
22 1633510061357057      7      9      3      DAQModuleManager  DBG  init_modules: construct: QueueToNetwork : qton_token  
23 1633510059553195      7      9      3      DAQModuleManager  DBG  init_modules: construct: NetworkToQueue : ntoq_trigdec  
24 1633510059553182      7      9      3      DAQModuleManager  D02  init_queues: Adding queue: trigger_record_q  
25 1633510059553022      7      9      3      DAQModuleManager  D02  init_queues: Adding queue: trigger_decision_q  
26 1633510059553022      7      9      3      DAQModuleManager  D02  init_queues: Adding queue: trigger_decision_from_netq  
27 1633510059553021      7      9      3      DAQModuleManager  D02  init_queues: Adding queue: token_q  
28 1633510059553020      7      9      3      DAQModuleManager  D02  init_queues: Adding queue: data_requests_1  
29 1633510059553020      7      9      3      DAQModuleManager  D02  init_queues: Adding queue: data_requests_0  
30 1633510059553019      7      9      3      DAQModuleManager  D02  init_queues: Adding queue: data_fragments_q  
31 1633510059552930      7      9      3      DAQModuleManager  D01  execute: Command id:init  
32 1633510039123903      7      8      3      RestEndpoint      LOG  handleResponseCommand: Response code = OK  
33 1633510039121696      7      9      2      RestEndpoint      LOG  handleResponseCommand: Sending POST request to nanorc.my-  
partition:56789/response
```

- Need to add `-pid=host`