



Fault Management

Table of contents

Logging

nvlog message format

nvlog Components

Event codes

List of Figures

Figure 0. Image11

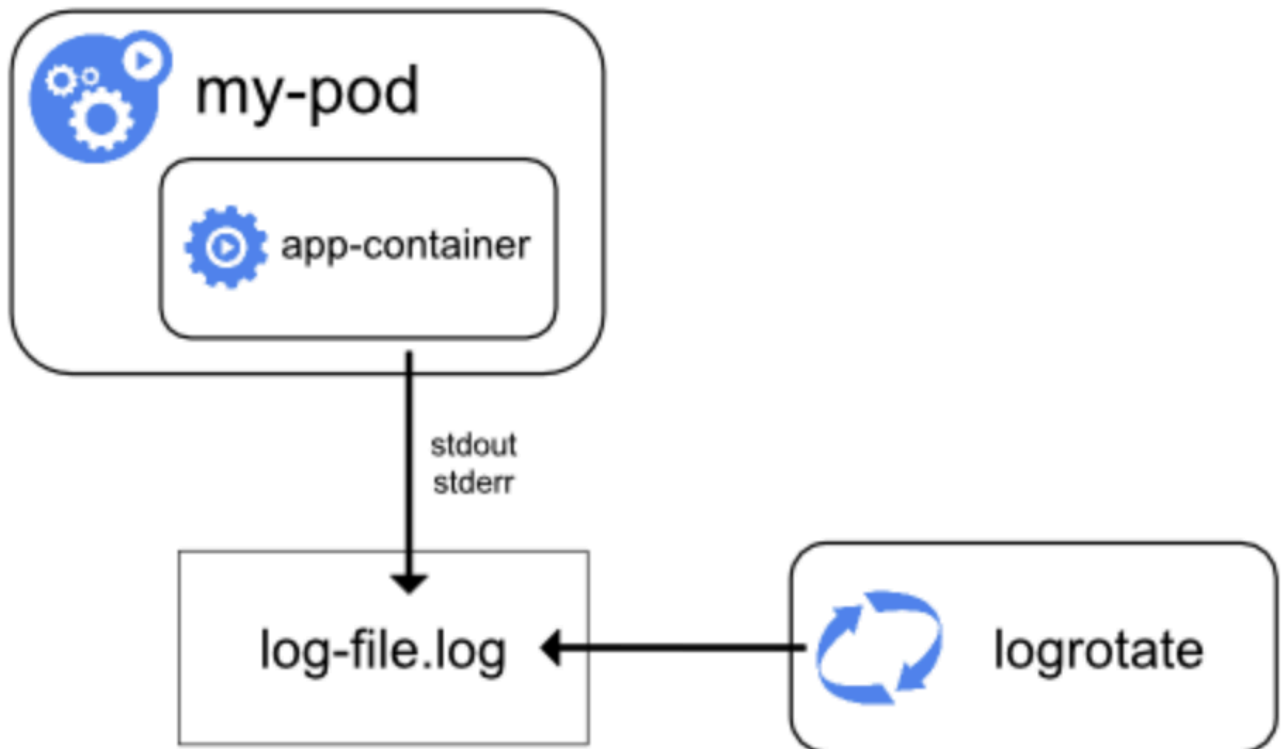
Figure 1. Image12

Figure 2. Image13

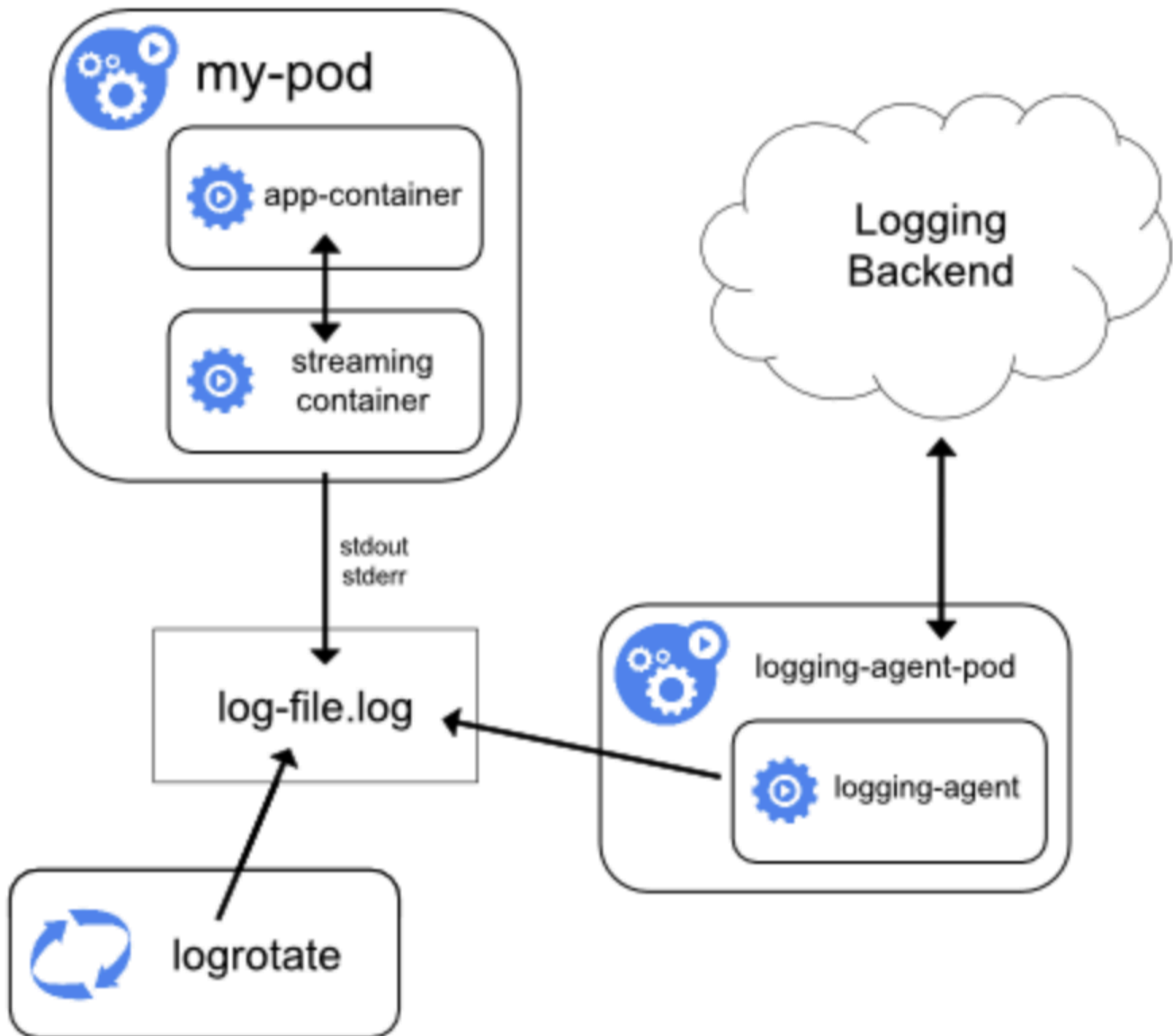
Logging

Aerial follows the best practices of Kubernetes (<https://kubernetes.io/docs/concepts/cluster-administration/logging/>) for implementing logging.

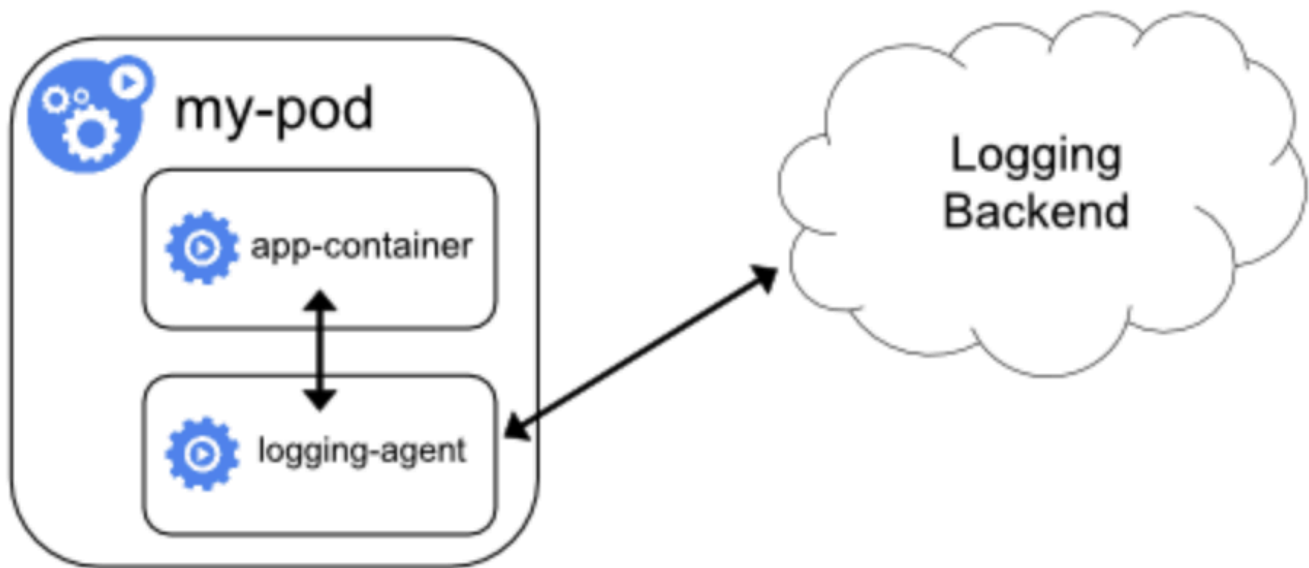
The cuphycontroller application outputs log messages, where the log level is less than or equal to the *nvlog.console_log_level* cuphycontroller YAML configuration parameter, directly to stdout using the **logging at the node level** pattern:



For high performance logs, Aerial uses a shared memory logger to offload the I/O bottleneck from the real-time threads. Log messages, where level is less than or equal to the *nvlog.shm_log_level* cuphycontroller YAML configuration parameter, are output to the shared memory logger. The shared memory logger outputs can be retrieved using either the **streaming sidecar** pattern with logs written directly to the local disk:



Or the **sidecar with logging agent** pattern to stream directly to an external logging backend:



nvlog message format

Each nvlog message is a string of the form “[Software Component Name] Msg” prefixed with the following space-separated optional fields:

- Date
- Timestamp
- Primary or Secondary nvlog process
- Log level
- Log event code id
- Log event code string
- CPU core number the calling thread is running on
- 64-bit sequence number
- Thread ID
- Thread Name

These fields are enabled in the `nvlog_config.yaml`.

An example nvlog message is:

```
20:58:09.036299 C [NVLOG.CPP] nvlog_create: name=phy shm_level=1
console_level=1 max_file_size=0x10000000 shm_cache_size=0x200000
log_buf_size=1024 prefix_opts=0x09
```

The message above had the following prefaces enabled:

- Timestamp
- Log level

Here are three more example nvlog messages, where all prefixed fields are enabled, taken at the start of the cuphycontroller process execution:

```
2021-09-15 21:29:22.926521 P C 0 SUCCESS 1 0 140699056300032 cuphycontroller
[NVLOG.CPP] nvlog_create: name=phy shm_level=1 console_level=1
max_file_size=0x10000000 shm_cache_size=0x200000 log_buf_size=1024
prefix_opts=0xFF 2021-09-15 21:29:22.926560 P C 0 SUCCESS 1 1 140699056300032
cuphycontroller [CTL.SCF] Config file: /cuBB_21-3/cuPHY-
CP/cuphycontroller/config/cuphycontroller_V08.yaml 2021-09-15 21:29:23.130882 P
C 0 SUCCESS 22 2 140699056300032 cuphycontroller [CTL.YAML] Standalone mode:
No
```

Here is an example of an nvlog message at Fault level with Event Code AERIAL_MEMORY_EVENT:

```
20:58:09.036299 F MEMORY_EVENT Unable to allocate memory for FH buffers
```

The message above had the following prefaces enabled:

- Timestamp
- Log level
- Log event code string

The fields are further described herein:

Date is YYYY-MM-DD format, for example, 1970-01-01

Timestamp is HH:MM:SS.us, for example, 20:58:09.036299

Primary process is P, secondary process is S.

Log level is:

- F - Fatal
- E - Error
- C - Console
- W - Warning
- I - Info
- D - Debug
- V - Verbose

Log event code string or log event code id is a string (or a numerical id) that indicates the category of event that has occurred.

nvlog Components

Aerial implements the following default logging component tags:

nvlog component:

- 10: "NVLOG"
- 11: "NVLOG.TEST"
- 12: "NVLOG.ITAG"

nvipc component:

- 30: "NVIPC"

cuPHY-CP Controller component:

- 100: "CTL"
- 101: "CTL.SCF"
- 102: "CTL.ALTRAN"
- 103: "CTL.DRV"
- 104: "CTL.YAML"

cuPHY-CP driver component:

- 200: "DRV"
- 201: "DRV.SA"
- 202: "DRV.TIME"
- 203: "DRV.CTX"
- 204: "DRV.API"
- 205: "DRV.FH"
- 206: "DRV.GEN_CUDA"
- 207: "DRV.GPUDEV"
- 208: "DRV.PHYCH"
- 209: "DRV.TASK"
- 210: "DRV.WORKER"
- 211: "DRV.DLBUF"
- 212: "DRV.CSIRS"
- 213: "DRV.PBCH"
- 214: "DRV.PDCCH_DL"

- 215: "DRV.PDSCH"
- 216: "DRV.MAP_DL"
- 217: "DRV.FUNC_DL"
- 218: "DRV.HARQ_POOL"
- 219: "DRV.ORDER_CUDA"
- 220: "DRV.ORDER_ENTITY"
- 221: "DRV.PRACH"
- 222: "DRV.PUCCH"
- 223: "DRV.PUSCH"
- 224: "DRV.MAP_UL"
- 225: "DRV.FUNC_UL"
- 226: "DRV.ULBUF"
- 227: "DRV.MPS"
- 228: "DRV.METRICS"
- 229: "DRV.MEMFOOT"
- 230: "DRV.CELL"

cuPHY-CP cuphyl2adapter component:

- 300: "L2A"
- 301: "L2A.MAC"
- 302: "L2A.MACFACT"
- 303: "L2A.PROXY"
- 304: "L2A.EPOLL"

- 305: "L2A.TRANSPORT"
- 306: "L2A.MODULE"
- 307: "L2A.TICK"
- 308: "L2A.UEMD"

cuPHY-CP scfl2adapter component:

- 330: "SCF"
- 331: "SCF.MAC"
- 332: "SCF.DISPATCH"
- 333: "SCF.PHY"
- 334: "SCF.SLOT CMD"
- 335: "SCF.L2SA"
- 336: "SCF.DUMMYMAC"

cuPHY-CP testMAC component:

- 400: "MAC"
- 401: "MAC.LP"
- 402: "MAC.FAPI"
- 403: "MAC.UTILS"
- 404: "MAC.SCF"
- 405: "MAC.ALTRAN"
- 406: "MAC.CFG"
- 407: "MAC.PROC"

cuPHY-CP ru-emulator component:

- 500: "RU"
- 501: "RU.EMULATOR"
- 502: "RU.PARSER"

cuPHY-CP aerial-fh-driver component:

- 600: "FH"
- 601: "FH.FLOW"
- 602: "FH.FH"
- 603: "FH.GPU_MP"
- 604: "FH.LIB"
- 605: "FH.MEMREG"
- 606: "FH.METRICS"
- 607: "FH.NIC"
- 608: "FH.PDUMP"
- 609: "FH.PEER"
- 610: "FH.QUEUE"
- 611: "FH.RING"
- 612: "FH.TIME"

cuPHY-CP compression_decompression component:

- 700: "COMP"

cuPHY-CP cuphyoam component:

- 800: "OAM"

cuPHY component:

- 900: "CUPHY"

Note

These strings can be changed using the `nvlog_config.yaml`.

Event codes

The following is the list of event codes (see `aerial_event_code.h`). The event strings match the event code names, minus the `AERIAL_`.

```
| AERIAL_SUCCESS = 0, | AERIAL_INVALID_PARAM_EVENT = 1, |  
AERIAL_INTERNAL_EVENT = 2, | AERIAL_CUDA_API_EVENT = 3, |  
AERIAL_DPDK_API_EVENT = 4, | AERIAL_THREAD_API_EVENT = 5, |  
AERIAL_CLOCK_API_EVENT = 6, | AERIAL_NVIPC_API_EVENT = 7, |  
AERIAL_ORAN_FH_EVENT = 8, | AERIAL_CUPHYDRV_API_EVENT = 9, |  
AERIAL_INPUT_OUTPUT_EVENT = 10, | AERIAL_MEMORY_EVENT = 11, |  
AERIAL_YAML_PARSER_EVENT = 12, | AERIAL_NVLOG_EVENT = 13, |  
AERIAL_CONFIG_EVENT = 14, | AERIAL_FAPI_EVENT = 15, |  
AERIAL_NO_SUPPORT_EVENT = 16, | AERIAL_SYSTEM_API_EVENT = 17, |  
AERIAL_L2ADAPTER_EVENT = 18, | AERIAL_RU_EMULATOR_EVENT = 19,
```

© Copyright 2024, NVIDIA.. PDF Generated on 06/06/2024