Open. Together.

OCP REGIONAL SUMMIT

Networking

Devlink Linux

Provisioning. Health. Recovery.

Bill Holland
Facebook, Release To Production
Eran Ben-Elisha
Mellanox, Linux Development

OPEN
PLATINUM™

Motivation: Health Report and Recovery

We manage a diverse set of OCP compute and storage servers. Inevitably, errors appear in production environment

- How to identify the malfunctioning component? (Hardware, Firmware, Software)
- What info to collect? Are kernel logs enough?
- Need access to a live system to resolve?
- How to recover?

Linux Kernel Standard For Health and Debug

Reduce disruptive run-time debug time

- Real time monitoring (HW/FW/SW)
- Alerts when something goes south
- Self-healing
- Detailed snapshot for offline troubleshooting

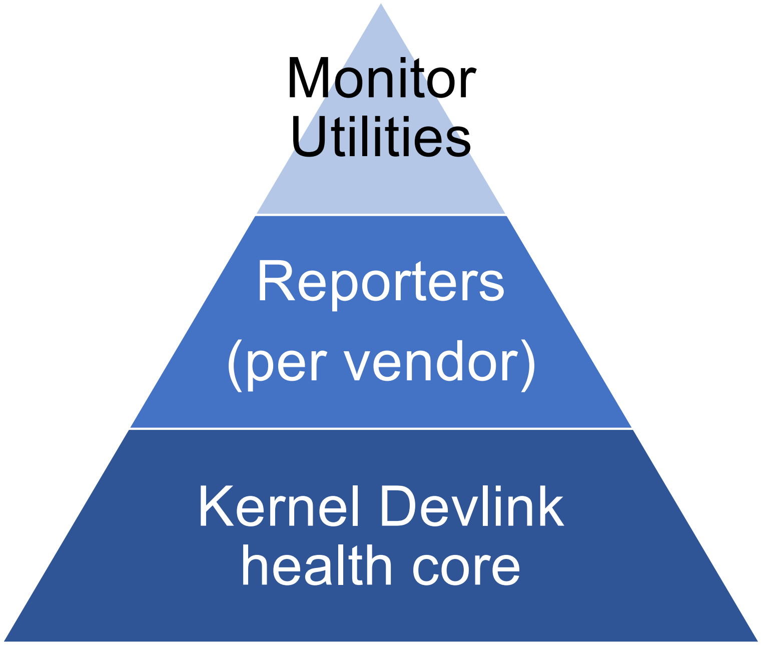
Replacing

- Sporadic, mostly vendor-specific tools
- Driver's vendor-specific log error format
- Non-error oriented tools

Devlink Health

Generic health handle in net/core/devlink

- Centralizing PCIe drivers' health
- Real time error detection
- Configurable reporters and callback ops
- Error recovery
- Real time diagnose/status
- Error related dumps
- Per reporter statistics
- Logging to Kernel's debug buffer
- Accepted into Kernel 5.1



Monitor Utilities

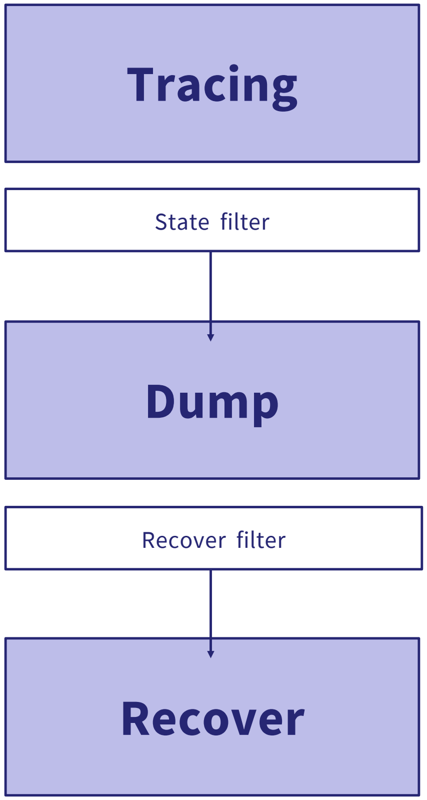
Reporters (per vendor)

Kernel Devlink health core

Error Handling Flow

Initiated by the driver upon an error detection

- Generic kernel handle for all errors
- Dump and recover per reporter



mlx5e Reporters

mlx5e driver is supporting the following reporters:
(including dumps and recover flows)

- TX : TX timeout, TX error completion
- RX : Init failure, RX error completion
- FW : Assert/Syndrome, PCI errors, FW stuck

User Commands

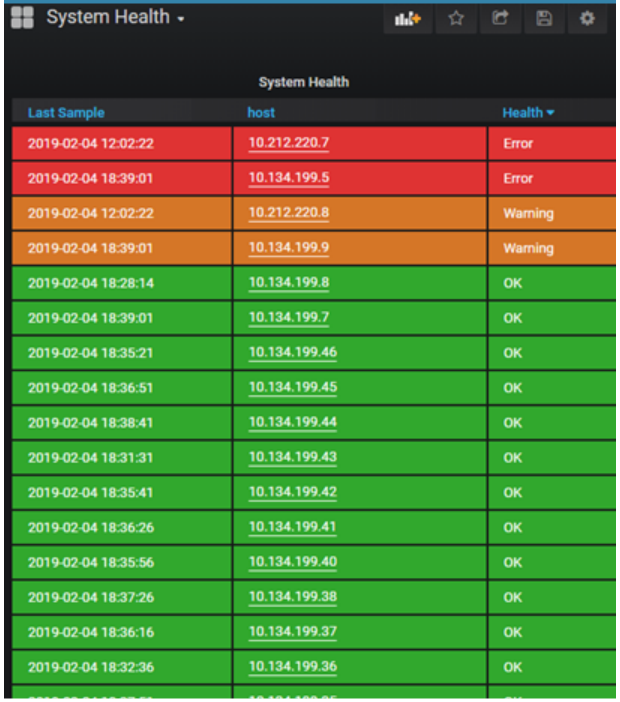
```
$ devlink health help
Usage:
    devlink health show [dev DEV reporter REPORTER]
    devlink health diagnose DEV reporter REPORTER
    devlink health dump show DEV reporter REPORTER
    devlink health dump clear DEV reporter REPORTER
    devlink health recover DEV reporter REPORTER
    devlink health set DEV reporter REPORTER {
   grace_period | auto_recover } { msec | boolean }
```

Based on iproute2 devlink tool

- **Show** status and config of supported reporters, for all devlink devices
- Reporter’s **diagnostic data** (JSON)
- **Show** and **clear** last saved **dump**
- Initiate the reporter’s **recovery** op
- **Set** attributes like period between auto recoveries, auto-recover toggle

Cluster Health Monitoring

General System Status (per host)

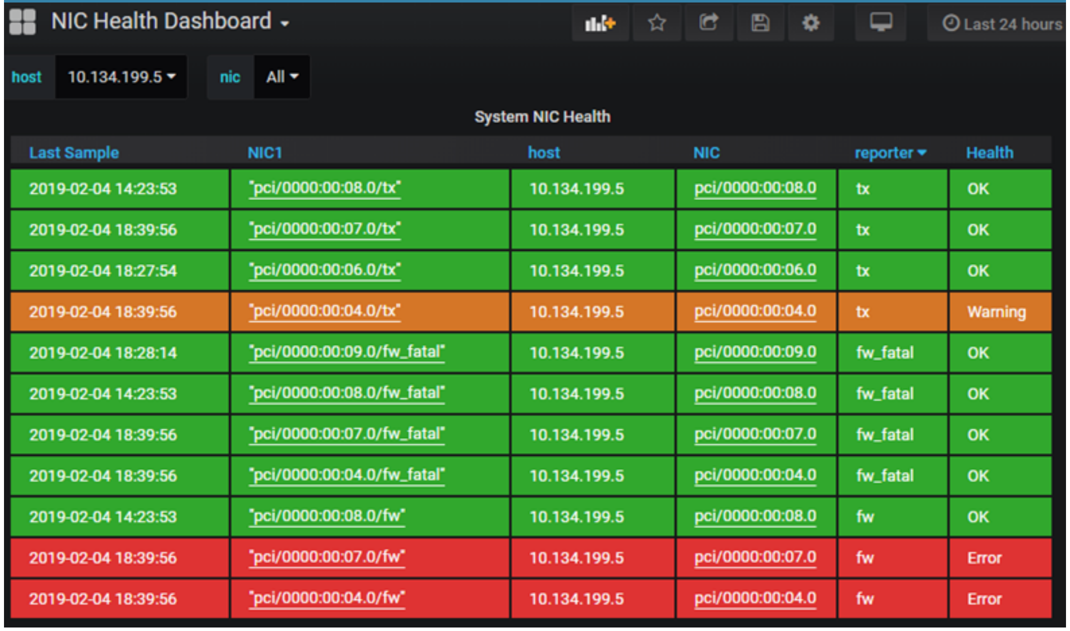


System Health ▾

System Health

Last Sample	host	Health ▾
2019-02-04 12:02:22	10.212.220.7	Error
2019-02-04 18:39:01	10.134.199.5	Error
2019-02-04 12:02:22	10.212.220.8	Warning
2019-02-04 18:39:01	10.134.199.9	Warning
2019-02-04 18:28:14	10.134.199.8	OK
2019-02-04 18:39:01	10.134.199.7	OK
2019-02-04 18:35:21	10.134.199.46	OK
2019-02-04 18:36:51	10.134.199.45	OK
2019-02-04 18:38:41	10.134.199.44	OK
2019-02-04 18:31:31	10.134.199.43	OK
2019-02-04 18:35:41	10.134.199.42	OK
2019-02-04 18:36:26	10.134.199.41	OK
2019-02-04 18:35:56	10.134.199.40	OK
2019-02-04 18:37:26	10.134.199.38	OK
2019-02-04 18:36:16	10.134.199.37	OK
2019-02-04 18:32:36	10.134.199.36	OK

System Status Per Host (raw per {reporter x NIC})



NIC Health Dashboard ▾

Last 24 hours

host: 10.134.199.5 ▾ nic: All ▾

System NIC Health

Last Sample	NIC1	host	NIC	reporter ▾	Health
2019-02-04 14:23:53	"pci/0000:00:08.0/tx"	10.134.199.5	pci/0000:00:08.0	tx	OK
2019-02-04 18:39:56	"pci/0000:00:07.0/tx"	10.134.199.5	pci/0000:00:07.0	tx	OK
2019-02-04 18:27:54	"pci/0000:00:06.0/tx"	10.134.199.5	pci/0000:00:06.0	tx	OK
2019-02-04 18:39:56	"pci/0000:00:04.0/tx"	10.134.199.5	pci/0000:00:04.0	tx	Warning
2019-02-04 18:28:14	"pci/0000:00:09.0/fw_fatal"	10.134.199.5	pci/0000:00:09.0	fw_fatal	OK
2019-02-04 14:23:53	"pci/0000:00:08.0/fw_fatal"	10.134.199.5	pci/0000:00:08.0	fw_fatal	OK
2019-02-04 18:39:56	"pci/0000:00:07.0/fw_fatal"	10.134.199.5	pci/0000:00:07.0	fw_fatal	OK
2019-02-04 18:39:56	"pci/0000:00:04.0/fw_fatal"	10.134.199.5	pci/0000:00:04.0	fw_fatal	OK
2019-02-04 14:23:53	"pci/0000:00:08.0/fw"	10.134.199.5	pci/0000:00:08.0	fw	OK
2019-02-04 18:39:56	"pci/0000:00:07.0/fw"	10.134.199.5	pci/0000:00:07.0	fw	Error
2019-02-04 18:39:56	"pci/0000:00:04.0/fw"	10.134.199.5	pci/0000:00:04.0	fw	Error

Call to Action

- Get familiar with **devlink**
  kernel.org/doc/Documentation/networking/devlink-health.txt

  git.kernel.org/pub/scm/linux/kernel/git/davem/net-next.git/commit/?id=51a5365c2b21da5b19b4905a1214cd20c32fbf6c
- Work with the OCP Networking Group on possible implementation intersections:
  - Mailing List: ocp-all.groups.io/g/OCP-Networking
  - OCP Networking Wiki Page: opencompute.org/wiki/Networking
- Work with your OCP NIC provider(s) to implement **devlink**
- Exploit **devlink** for a more scalable OCP solution

At Scale Operations with Devlink

- ➢Operation simplicity
- ➢Building blocks
- ➢Security
- ➢Real time data
- ➢Open source

OCP
REGIONAL
SUMMIT

Open. Together.

OCP Regional Summit

26–27, September, 2019