Image Deployment and System Monitoring with HPCM

Sue Miller, Engineering Resolution Team/Product Issue Resolution
Pete Guyan, Distinguished Technologist

May, 2024

Agenda

- What's a recipe?
- What's an image?
- Image creation and version control
- cm image cli
- Concepts associated with an image – rpmlists/deblists, repos and repo groups
- Deploying an image and miniroot
- The configuration framework

Monitoring is split in to 3 sections each starting with "the short story" summary of 1-2 slides on basic configuration for most Cray systems followed by a deeper dive. The focus is on EX and we cover preventing disk space issues.

- Kafka and the consumers
- Producers
- Alerting, SIM and rackmap

Images

- What's a recipe?
- What's an image?
- Image creation and version control
- cm image cli
- Concepts associated with an image – rpmlists/deblists, repos and repo groups
- Deploying an image and miniroot
- The configuration framework

What's a recipe?

- A HPCM recipe contains the critical HPC software stack tested as a complete entity on supported hardware
  - documented on HPE web pages
- As opposed to the HPCM quality engineering testing done specifically for supported OS and hardware
  - documented in /opt/clmgr/docs/HPCM-Release-Notes.txt
- Recipes release after HPCM versions which are ~6 monthly

What's a recipe?

RHEL or SLES based. For example:

- HPCM
- COS/COS-Base/USS
- Slingshot Host Software
- SLES base
- SLES updates
- AMD GPU
- ROCM
- Nvidia driver GPU
- Nvidia SDK
- CPE
- SLURM
- Firmware – HPC firmware pack

What's an image?

An image is:

- An operating system directory stored under `/opt/clmgr/image/images`
- Kernel and initrd plus PXE configuration stored under `/opt/clmgr/tftpboot`
- Miniroot – OS initrd with vendor scripts which are used to pull a stripped down environment used for network booting and imaging nodes which is termed as miniroot.
- Aria2c encrypted tar ball (default transport method)

Default **generic** images created via configure-cluster or `/opt/clmgr/bin/create-default-images`

- ICE rack leader images e.g. lead-sles15spX.
- SU leader images e.g. su-sles15spX.
- ICE compute images e.g. ice-sles15spX.
- x86_64 compute images e.g. sles15spX.
- Arm (aarch64) compute images e.g. sles15spX-aarch64.

Image creation and version control

Other than using the default images, images can be created thus:

- Using a rpmlist or deblist to build the image on the admin node (generic)
- Importing an existing image from another cluster with appropriate repos (generic)
- Using autoinstall (autoyast or kickstart) to build the image on the admin node (autoinstall)
- Capture an image from a running node with appropriate repos (generic)

Best practice is to use generic images

Version control system (VCS)

Switch versions easily

Multiple versions of each image.

First version is the same size as the original image

Subsequent versions are smaller as they only track the changes

cm image CLI

Edited for brevity and continued on next slide:

```
admin:~ # cm image --help
    activate                Activate an image for use by NFS clients.
    apt                     Perform apt operations on an image
    capture                 Capture an image from an existing node using rsync.
Any additional arguments are passed to rsync.
    copy                    Makes a copy of an image, committing the new image
into the image version control system at rev 1.
    create                  Each usage above represents a distinct command. The
first creates a new systemimager image, the second registers a previously
existing systemimager image with the cluster database, and the third sets
up an autoinstall configuration (kickstart or autoyast) for image
deployment.
    delete                  Delete a specified image or bittorrent tarball
    dnf                     Perform dnf operations on an image
    recreate                Deletes an image or associated miniroot, then
recreates it. Node associations are preserved.
```

cm image CLI continued

```
    refresh                 Refresh a specified image
    revision                Operations on stored image revisions
    rpmlist                 Used to generate an rpmlist from an image or node
or to show differences between image packages.
    set                     Set image properties
    show                    Show properties of an OS disk image
    sync                    sync global overrides and pre/post-scripts to su-
leaders
    unset                   Unset image properties
    update                  Update an image or its related elements
    yum                     NOTE: This command is deprecated. Consider using
'cm image dnf' instead Perform yum operations on an image
    zypper                  Perform zypper operations on an image
```

Concepts associated with an image

rpmlists/deblists: `/opt/clmgr/image/{deblists,rpmlists}`

repos:

```
admin# cm repo show
* Cluster-Manager-1.11-sles15sp5-x86_64 : /opt/clmgr/repos/cm/Cluster-Manager-1.11-sles15sp5-x86_64
* Cluster-Manager-AIOps-1.11-sles15sp5-x86_64 : /opt/clmgr/repos/cm/Cluster-Manager-AIOps-1.11-sles15sp5-x86_64
* HPE-MPI-1.9.5-sles15sp5-x86_64 : /opt/clmgr/repos/cm/HPE-MPI-1.9.5-sles15sp5-x86_64
* SLE-15-SP5-Full-x86_64 : /opt/clmgr/repos/distro/sles15sp5-x86_64
* sles15sp5-updates-x86_64 : /opt/clmgr/repos/sles15sp5-updates-x86_64
  slingshot-fmn-packages-2.1.1-1215-sles15sp5 : /opt/clmgr/repos/other/fmn211/slingshot-fmn-packages-2.1.1-1215-sles15sp5
  slingshot-host-software-2.1.1-64-sle15-sp5_x86_64 : /opt/clmgr/repos/other/slingshot-host-software-2.1.1-64-sle15-sp5_x86_64/rpms/cassini/sle15-sp5
  slurm-2.0.9-23.2.7-sle-15.5-sles15sp5-x86_64 : /opt/clmgr/repos/other/slurm-2.0.9-23.2.7-sle-15.5-sles15sp5-x86_64
```

* denotes selected i.e. in use by default with commands. Admins often de-select and use repo groups

Concepts associated with an image – repo groups

Environments have multiple versions of different operating systems running and differing software needed in images. Systems often have multiple people working on different image simultaneously so it saves time selecting and de-selecting repos plus conflicts with selection.

```
admin:~ # cm repo group show
Repo Groups:
amd-ubuntu-repo-group
antero-castle-sles15sp5
bard-parry-6.0.2-sles15sp5
blanca-cos3.0-ss220dkms
grizzly-cuda12.3-sles15sp5
nvidia-ubuntu-repo-group
sles15sp5-base
sles15sp5-fmn
ubuntu2204.4-signed
windom-sles15sp5
admin:~ # cm repo group show sles15sp5-fmn
Group: sles15sp5-fmn
       Cluster-Manager-1.11-sles15sp5-x86_64
       SLE-15-SP5-Full-x86_64
       sles15sp5-updates-x86_64
       slingshot-fmn-packages-2.2.0-311-sles15sp5
```

OS Provisioning

- Images based on diverse operating systems and versions
- Repos managed with "`cm repo`" using `--repo` or `--repo-group` to commands (rpms and debs)
- Diskful or diskless (tmpfs or nfs with read-only or tmpfs writable, iscsi (>=1.11))
- 2 "flavours" of image: Generic (best practice use case) or autoinstall images (kickstart/autoyast)
- Different transport mechanisms (rsync, udpcast or bittorrent (default and really aria))
- HPCM uses the operating system initrd (OS) but with added vendor scripts which are used to pull a stripped down environment used for network booting and imaging nodes - miniroot

admin and leaders: RHEL or Rocky, SLES

compute/service : RHEL or Rocky, SLES (with or without COS), TOSS, Ubuntu

More on miniroot

To troubleshoot: `/var/log/miniroot` and `/var/log/consoles/<node>`

That relies on having the correct console parameter set. Once initrd/kernel are transferred, if there are no console messages after "`Booting...`" or similar then the console parameter is not set correctly for the hardware. `cm node show --console-device -n <node>`

The files associated with miniroot creation are `/opt/clmgr/lib/miniroot*`

The configuration file is either `/opt/clmgr/etc/miniroot-aarch64.conf` or `/opt/clmgr/etc/miniroot-x86_64.conf`

There is an OS section which has the rpms to be included, commands, kmods and specific files. These are taken from the rpms for the repos selected directly or via repo-groups. e.g. `[rhel8]`

The initrd (-k) and miniroot (-m) can be recreated with: `cm image update -k -m -i <image>`

It extracts to `/opt/clmgr/image/miniroot/pre_squeeze/<image>` as an intermediate steps

It then works on that content to reduce it down: `/opt/clmgr/image/miniroot/squeezed/<image>`

Some aspects may be taken from the image e.g. Firmware is covered in `/opt/clmgr/lib/miniroot` and is copied from the image not pre_squeeze miniroot like kernel modules etc.

Deploying images

Transport/ RootFS	rsync	bt (default)	udpcast (legacy)
nfs or iscsi	`cm image activate --image <image>` then reboot node	`cm image activate --image <image>` then reboot node	Unsupported

disk	No leaders: `cm node provision`
	Leaders: `su-sync-image <image>`
	`cm node provision`
		Leaders or no SU leaders:
		`cm image refresh --bittorrent`
		`cm node provision`
		Miniroot via rsync still
			Leaders or no SU leaders: `cm node provision`

tmpfs	No leaders: node reboot Leaders: `su-sync-image <image>` then reboot node	Leaders or no SU leaders: `cm image refresh --bittorrent` then reboot node	Leaders or no SU leaders: reboot node

Not all options are listed for brevity

Configuration framework

cmdb systemd service formerly cmu and not to be confused with the new cmu service for native monitoring!

- conf.d scripts also known as boot configuration framework
- Pre- and post-installation scripts
- What HPCM controls
- Executing (vendor install) scripts in a chroot in the image
- Have the changes in the image

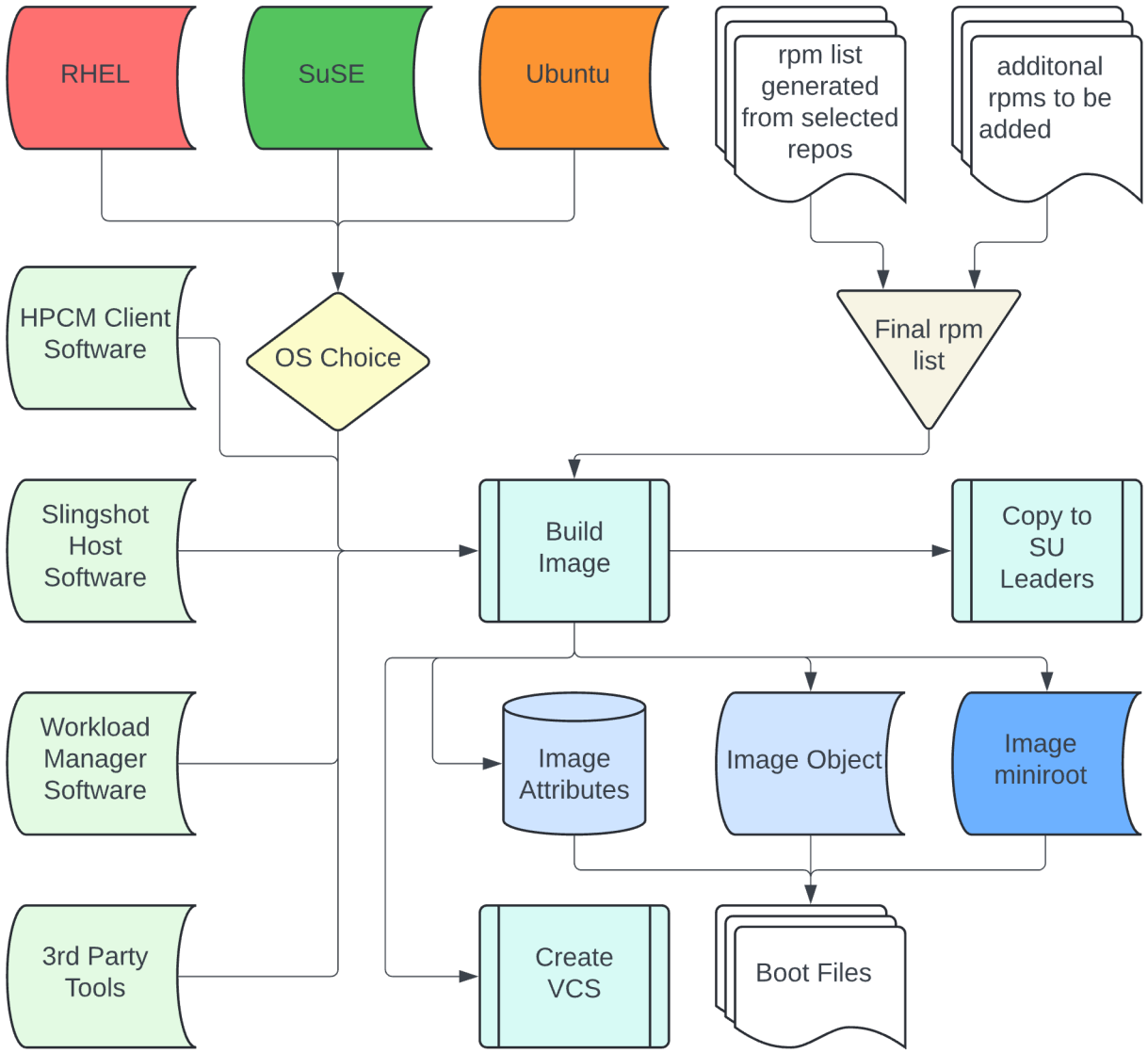
Database service log `/opt/clmgr/log/cmuserver-?.log` (where 0 is the current log)

You can have the changes in the image (however scripting preferred to make config re-playable)

You can chroot to an image under `/opt/clmgr/image/images` or you can setup a node and then use "cm image capture"

HPCM does change some systemd presets but places these in the standard location under the image:
`/opt/clmgr/image/images/image/usr/lib/systemd/system-preset`

Image Creation Process



A summary of the Image Creation Process

- Repositories (or Repository Groups) are used to create an image, there must be an OS and HPCM Client selected as a minimum.
- An rpmlist is used, which will have at least the generated rpmlist from the distribution and HPCM software. Additional packages can be added to the list.
- The Image build process creates an image object, image miniroot and populates the Image attributes in the database. Additional attributes can be set, such as custom kernel parameters
- A VCS commit is performed once the build is complete.
- If required, the image can be copied to the su-leaders
- Boot files are generated when any nodes are set to provision with the image.

Configuration framework

```
admin:~ # systemctl --no-pager -l status cmdb
● cmdb.service - Cluster Manager Backend Database
     Loaded: loaded (/usr/lib/systemd/system/cmdb.service; enabled; vendor preset: disabled)
     Active: active (running) since Sat 2024-02-10 05:41:00 CST; 4 days ago
   Main PID: 7035 (java)
      Tasks: 353
     CGroup: /system.slice/cmdb.service
             ├─  3885 sleep 30
             ├─  7035 /usr/bin/java -Xmx2048m -Xms2048m -server -Xmn1024m -XX:+UseConcMarkSweepGC -
XX:+AggressiveOpts -Djava.util.logging.config.file=/opt/clmgr/log/logging.properties -
Dlog4j.configuration=file:/opt/clmgr/log/logging.properties -Djdk.tls.acknowledgeCloseNotify=true -
Dcmu.monitoring.kafka=false -Dcmu.monitoring.runArchiver=true -
Dcmu.http.https.keystorePath=/opt/sgi/secrets/CA/private/server.p12 -cp
"bin/cmuserver.jar:bin/cmu_plugins/*:plugins/*" com.hpe.cmu.server.Main
             └─ 12019 /bin/bash /opt/clmgr/tools/pcm_cluster_ping

Feb 10 05:40:42 snowball-admin systemd[1]: Starting Cluster Manager Backend Database...
Feb 10 05:40:49 snowball-admin cmserver[2373]: cmu:core              configured
Feb 10 05:41:00 snowball-admin cmserver[2373]: cmu:backend           running GUI (RMI *:1099 and *:49150),
REST API (https://0.0.0.0:8080/cmu)
Feb 10 05:41:00 snowball-admin cmserver[2373]: cmu:web service       running (HTTP *:81 redirected to HTTPS
*:8443)
Feb 10 05:41:00 snowball-admin cmserver[2373]: cmu:cmustatus         running
Feb 10 05:41:00 snowball-admin cmserver[2373]: cmu:monitoring        running
Feb 10 05:41:00 snowball-admin systemd[1]: Started Cluster Manager Backend Database.
```

Configuration framework - conf.d scripts

Configuration on boot using `/etc/opt/sgi/conf.d` scripts

Scripts are run on boot as part of the cm-configuration service. (Formerly sgi-tempo-configuration service.)

e.g.

```
service0:/etc/opt/sgi/conf.d # grep -v ^# 80-insserv-adjustment-for-readonly-fixups
if [ -x /etc/init.d/boot.rootfsck ]; then
        rm -f /etc/init.d/boot.d/*boot.rootfsck
fi
```

There is an exclude file to which the name of a script can be added if you don't want HPCM to control it.

If for any reason, you modify a conf.d script and don't want it to be overwritten by an upgrade, you can use a ".local" file which would override the normal script. e.g. you could copy `15-network-setup` to `15-network-setup.local` and make changes to that .local file

Configuration framework - Pre- and post-installation scripts

The main port of call for customisations are post-installation scripts in `/opt/clmgr/image/scripts/post-install`.
There are also pre-installation scripts which are usually used for initial storage setup.
There are READMEs in the directories. An excerpt: - Scripts should be named in this way:

```
Two digit number to indicate order within a class.
 | Class name.
 | |  Period, followed by your description of the script.
 | |  |
 | |  |
 v v  v
 99all.harmless_example_script
 Classes include:
 - $IMAGENAME       (Ie: my_compute_image)
 - $BASE_HOSTNAME   (Ie: compute)
 - $HOSTNAME        (Ie: compute07)
 - all
 - $OVERRIDE
```

We generally say that it is best to have additional network configuration here.

Configuration framework - What HPCM controls

config_manager systemd service on the admin node

config_distrib systemd service on the SU leaders

config_client called as needed (used to be systemd service)

HPCM does control some configuration files which people sometimes do not expect e.g. `/etc/resolv.conf`

Those files are not individually documented as that is deemed a moving target but you can see which services HPCM configures and may trample on customisation for:

```
# cm node update config -h | grep -A 6 "^  --sync"
  --sync [SERVICE_OPTS]
              SERVICE_OPTS is a comma-separated list
              of the items to be updated, without white space.
              Available services are:  c3, clustershell, cminfo,
              crm, conserver, dhcp, dns, ganglia, hosts, ntp
              pcim, pdsh, flamethrower, nagios. Defaults to all
              services if no services are specified
```

Configuration framework - Overrides

The overrides mechanism useful as it does not require new image and activation/downtime for many nodes:

`/opt/clmgr/image/overrides/<image>`

A directory and file structure which contains differences to what is in the image and this does not need the usual image activation.

Create the required files and then use `su-sync-image` with no options. Newer versions have `cm image sync --scripts`

The override files will retain metadata (permissions, ownership, timestamps, etc.)

Configuration framework - Executing (vendor install) scripts in a chroot in the image

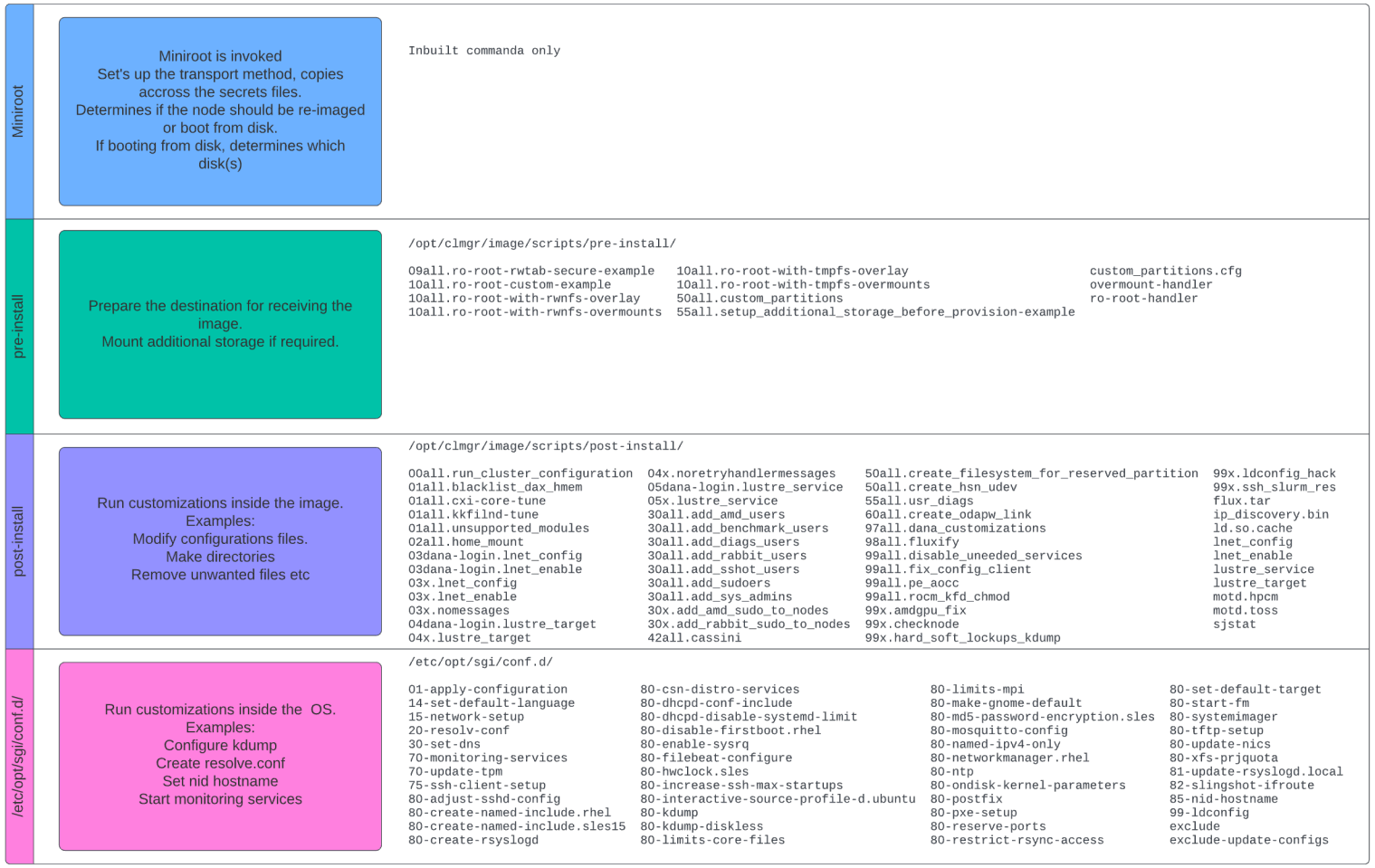
Executing vendor scripts within an image chroot, pre and post-creation options via "`cm image create`" or "`cm image update`":

```
 -p SCRIPT, --post-script SCRIPT
        Run SCRIPT after image creation. This is done in the
        package install environment (chrooted) with /proc, /sys,
        and other things mounted.
        If the SCRIPT name starts with '/' it is copied into /tmp in
        the image and run from there. If the SCRIPT name does not
        start with '/' it is assumed the SCRIPT already exists in
        the image and it is run chrooted relative to '/' in the chroot.
      NOTE: Cannot be used with -a|--autoinstall-file
 -P SCRIPT, --pre-script SCRIPT
        Run SCRIPT after image bootstrap. This is done in the
        package install environment (chrooted) with /proc, /sys,
        and other things mounted.
        This is run after the image bootstrap, but before packages
        in the list are added to the image. SCRIPT should be the
        complete path. It will be copied into the image and run
        chrooted.
      NOTE: Cannot be used with -a|--autoinstall-file
```

Image Deployment Process

Copied from Admin node

Installed into image

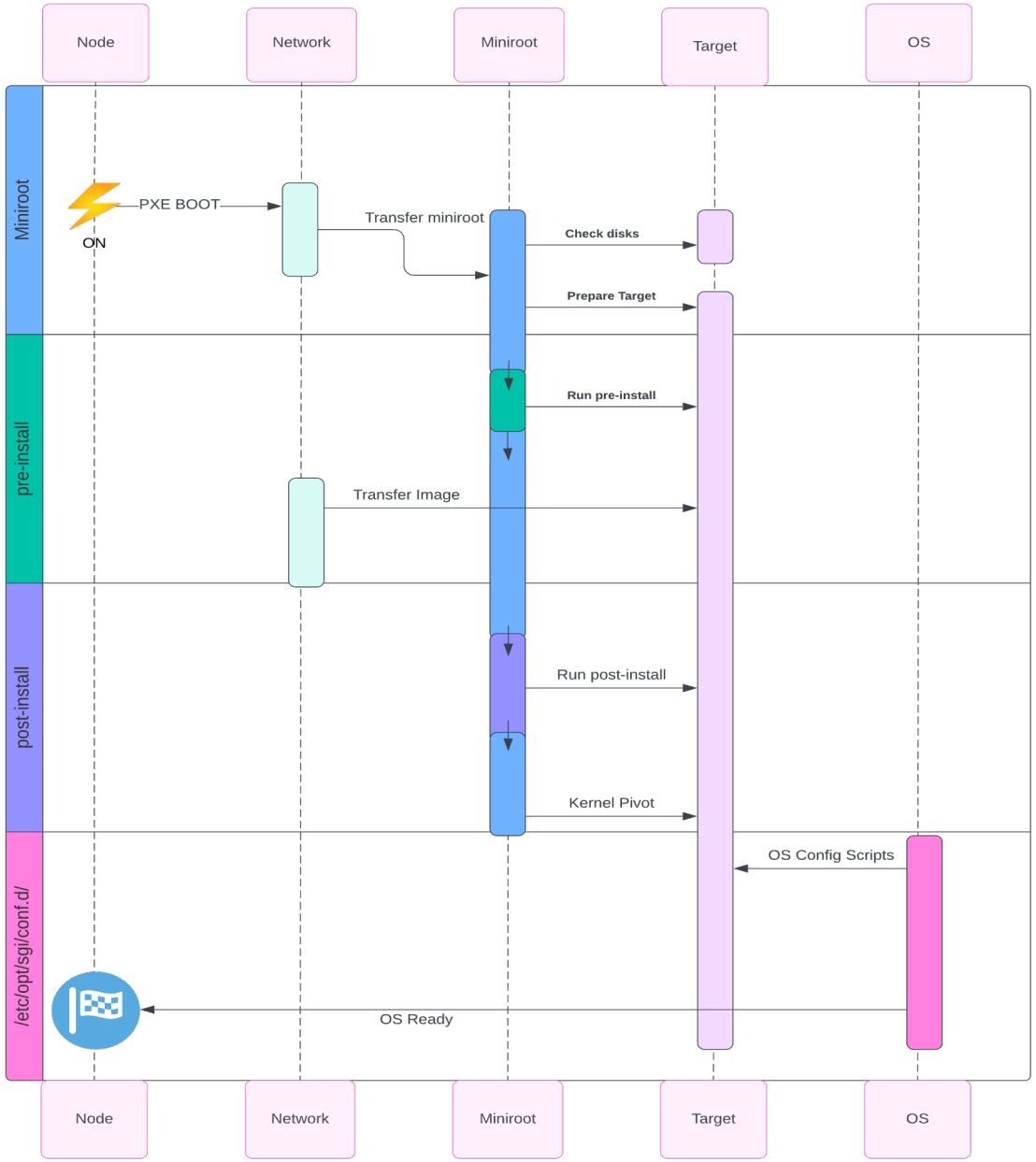


These are the default scripts

This a mix of default and custom scripts

This is example data

Image Deployment - Boot Process



This is a simplified view of how the boot process works.

If the target is a physical device, it is first probed, checked for HPCM labels, partitions created and formatted where needed, run the pre-install scripts then start the image transfer.

Once the image is transferred, the post-install scripts are run, if they all exit correctly, the OS is then started from the target.

As the OS boots, the conf.d scripts are executed before the full multi user level is reached.

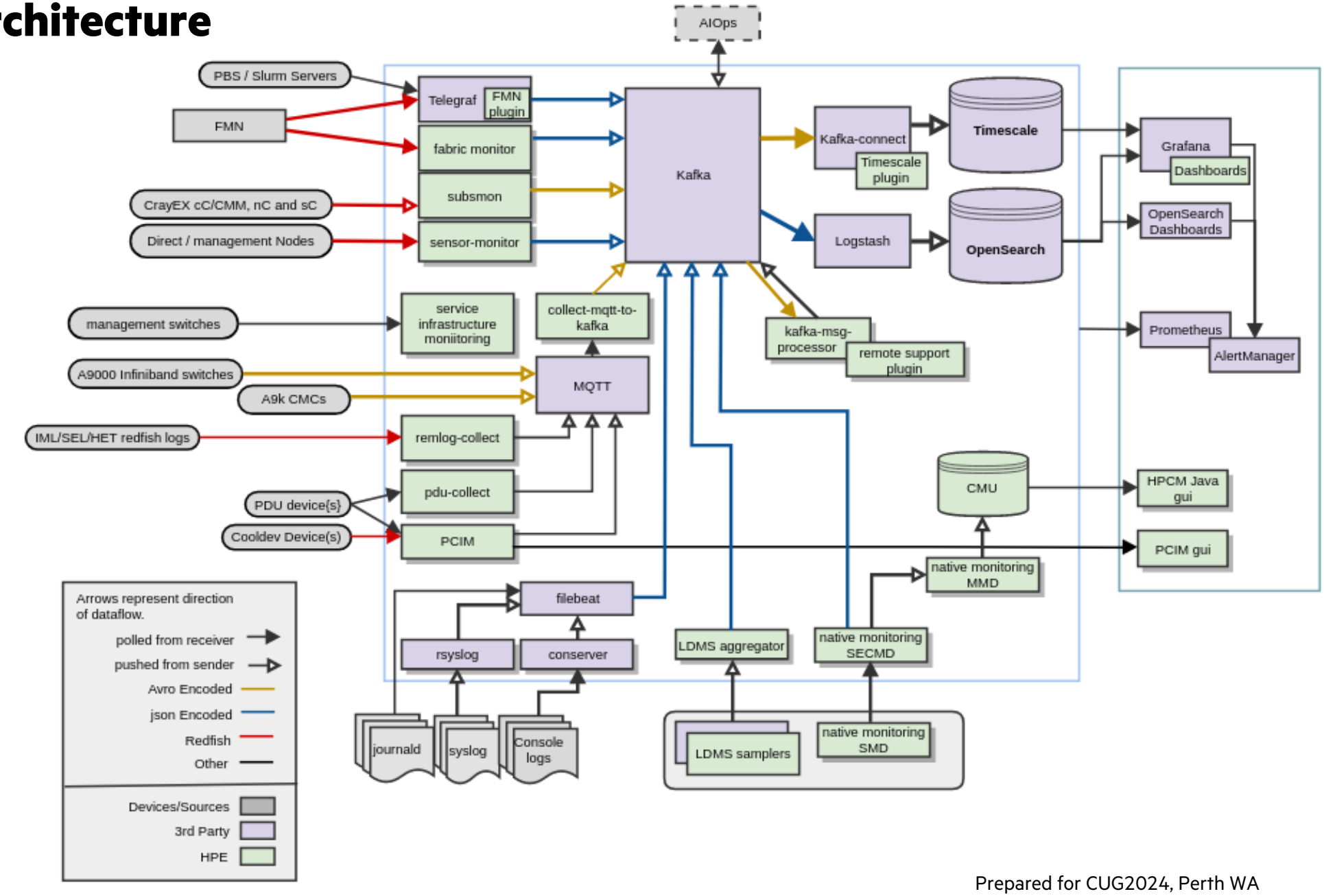
Monitoring

Monitoring is split in to 3 sections after the overview each starting with “the short story” summary of 1-2 slides on basic configuration for most Cray systems followed by a deeper dive.

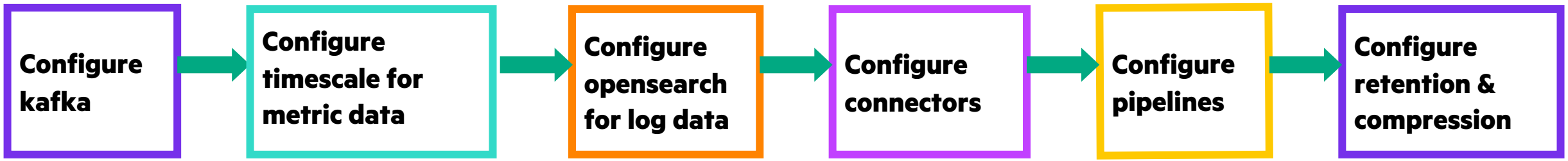
The focus is on EX and we cover preventing disk space issues.

- Kafka and the consumers
- Producers
- Alerting, SIM and rackmap

Monitoring architecture



Monitoring architecture configuration flow



Once complete, star the dashboards relevant to the system

Kafka and consumers: The short story

- **`cm monitoring kafka enable and start`**
- **`cm monitoring timescaledb enable and start`**
- **`cm monitoring elk enable and start`**

If there are SU leaders:

- **`kfka-dist-setup`**
- **`cm monitoring timescaledb node add <options>`**
- **`elk-dist-setup`**

In 1.10 and higher connectors are not enabled by default so enable the ones relevant to the system:

- **`cm monitoring connect enable --name <name>`**

Most pipelines (the producers) are not generally configured at this point.

Kafka and timescaledb use zookeeper to maintain a concept of cluster when using SU leaders but this is usually transparent to the user

Producers: The short story

Probably need to add a gpu type

- `cm monitoring native enable` **and** `start`
- `cm monitoring native metrics add -g slingshot -N <Max # NICs>` and `restart`
- `systemctl enable` **and** `start pcim`
- `cm monitoring dashboard grafana set --cdu|--cdu_ex2500 enable`
- Add cooling device other than Cray EX CDUs which are detected by default
- `systemctl enable` **and** `start sensor-monitor`

SU leaders: sensor-processor

- `cm monitoring slingshot enable` **and** `set <options>` **and** `start`
- `cm monitoring dashboard grafana set --slingshot enable`
- Double check FMN configuration

If changes are made on the FMN, new metrics with inappropriate compression can be created and consume disk. See last bullet.

- `cm node zypper|dnf -n <fmn> install slingshot-fabric-check`

Producers: The short story

- Set number of switches and switch groups in config file
- `systemctl enable` and `start` 3 services and timers on the fmn after installing rpm
- Curl commands to enable dashboards
- Install hpe-telegraf and telegraf on the slurm controller

Unnecessary in 1.11

- `cm monitoring slurm enable <options>` **and** `start`

Pre 1.11 see details

- For slurm power dependent on hardware, configure the plugin config in slurm and HPCM

  Configure `/opt/clmgr/wlm-mon/conf/wlm-mon.yml`

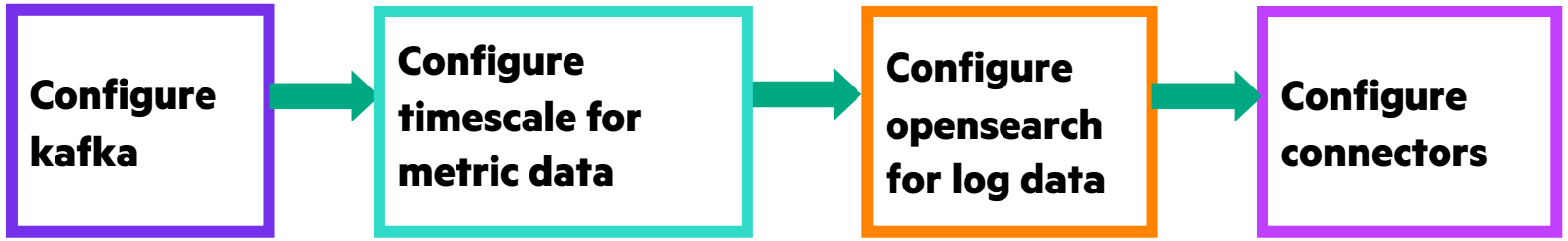
- Configure tsdb retention and **compression** after each stage **IMPORTANT:** tsdb compressions save >90% disk space

```
for i in slingshot cooldev pcm cray pdu disk; do cm monitoring
timescaledb retention --category  $i --interval 7d ;cm monitoring
timescaledb compression --category  $i --interval 1d ; done
```

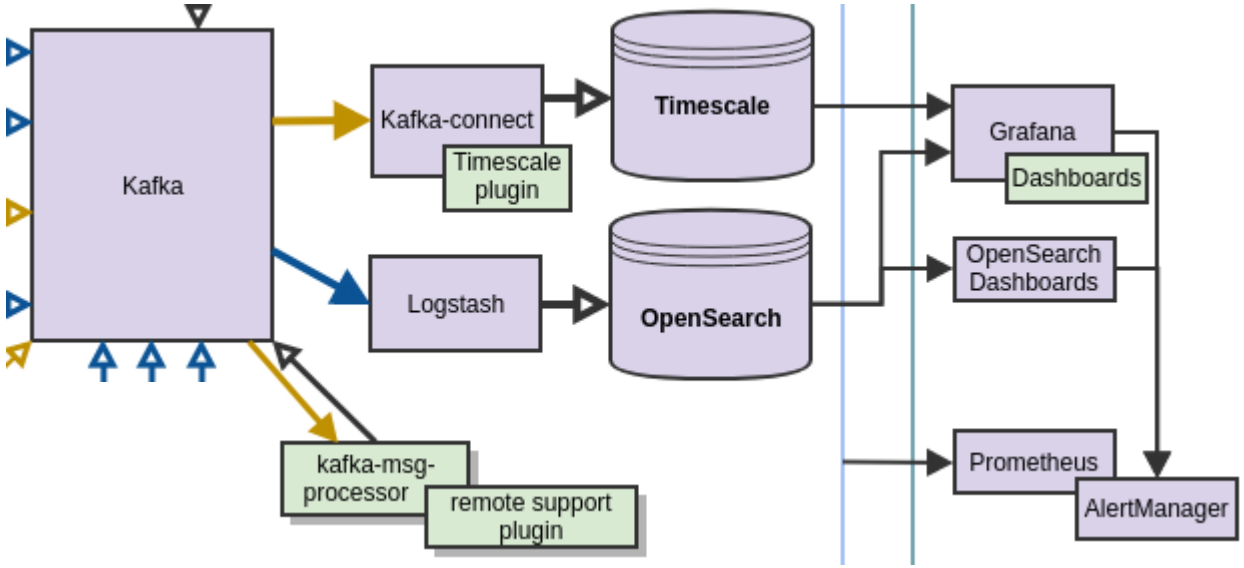
Alerting and SIM: The short story

1.10 plus 11796 or higher

- `cm monitoring alerting enable`
- `cm monitoring alerting opensearch` **or** `grafana --enable-rule <appropriate rules>`
- `cm monitoring alerting route email --from <email> --to <email> --smtp <smtp.server:25> --alert-group <group>`
- `cm sim enable` **and** `start` **and** `add {--service-group monitoring-services|suleader-services}`
- `cm monitoring rackmap map component-drift` **or** `power` **or** `cpu-temperature` **or** `slingshot-switch-status -l`



Kafka and the consumers



Kafka and consumers: The short story

- **`cm monitoring kafka enable and start`**
- **`cm monitoring timescaledb enable and start`**
- **`cm monitoring opensearch enable and start`**

Kafka and timescaledb use zookeeper to maintain a concept of cluster when using SU leaders but this is usually transparent to the user

If there are SU leaders:

- **`kfka-dist-setup`**
- **`cm monitoring timescaledb node add <options>`**
- **`elk-dist-setup`**

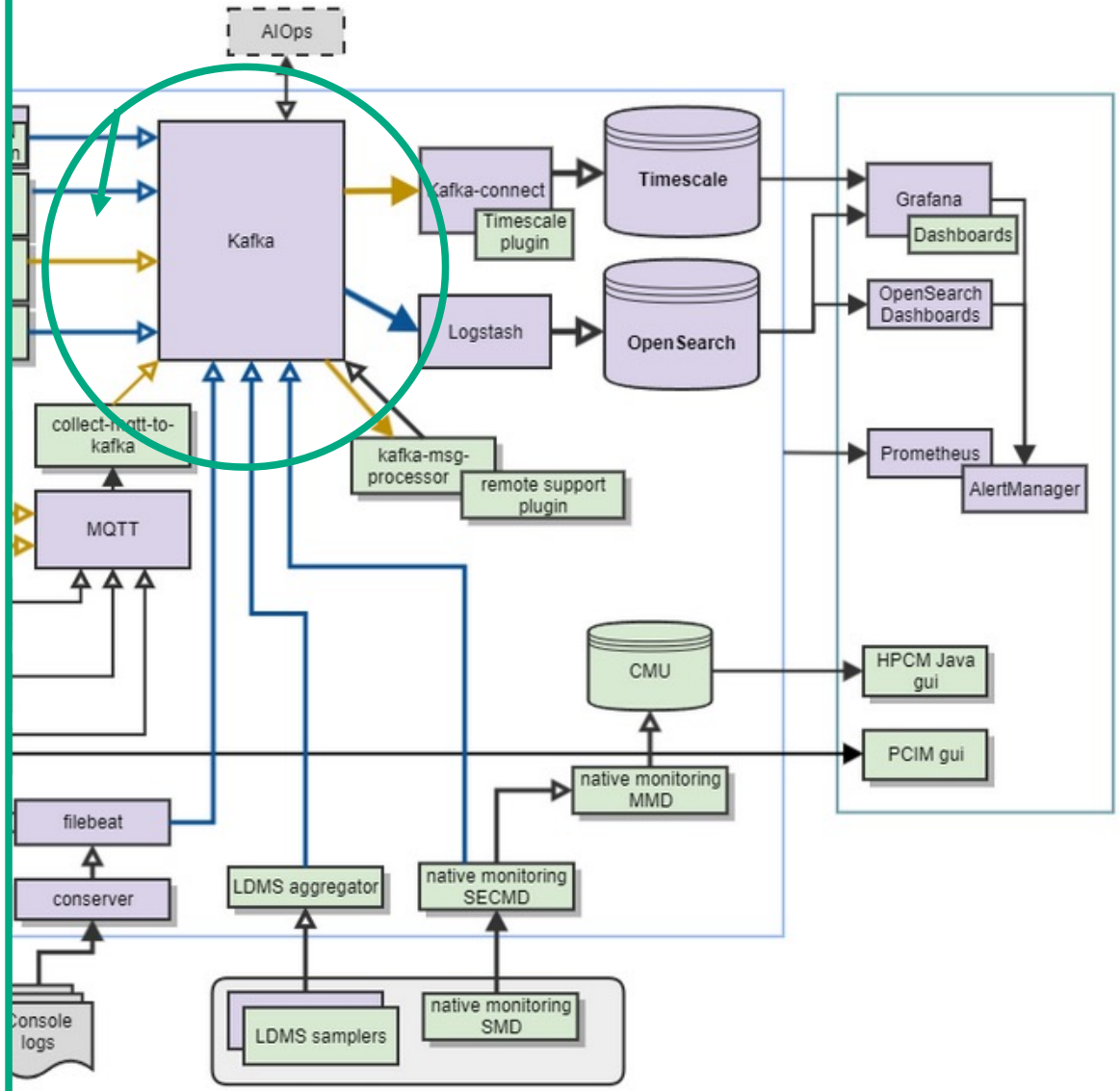
In 1.10 and higher connectors are not enabled by default so enable the ones relevant to the system:

- **`cm monitoring connect enable --name <name>`**

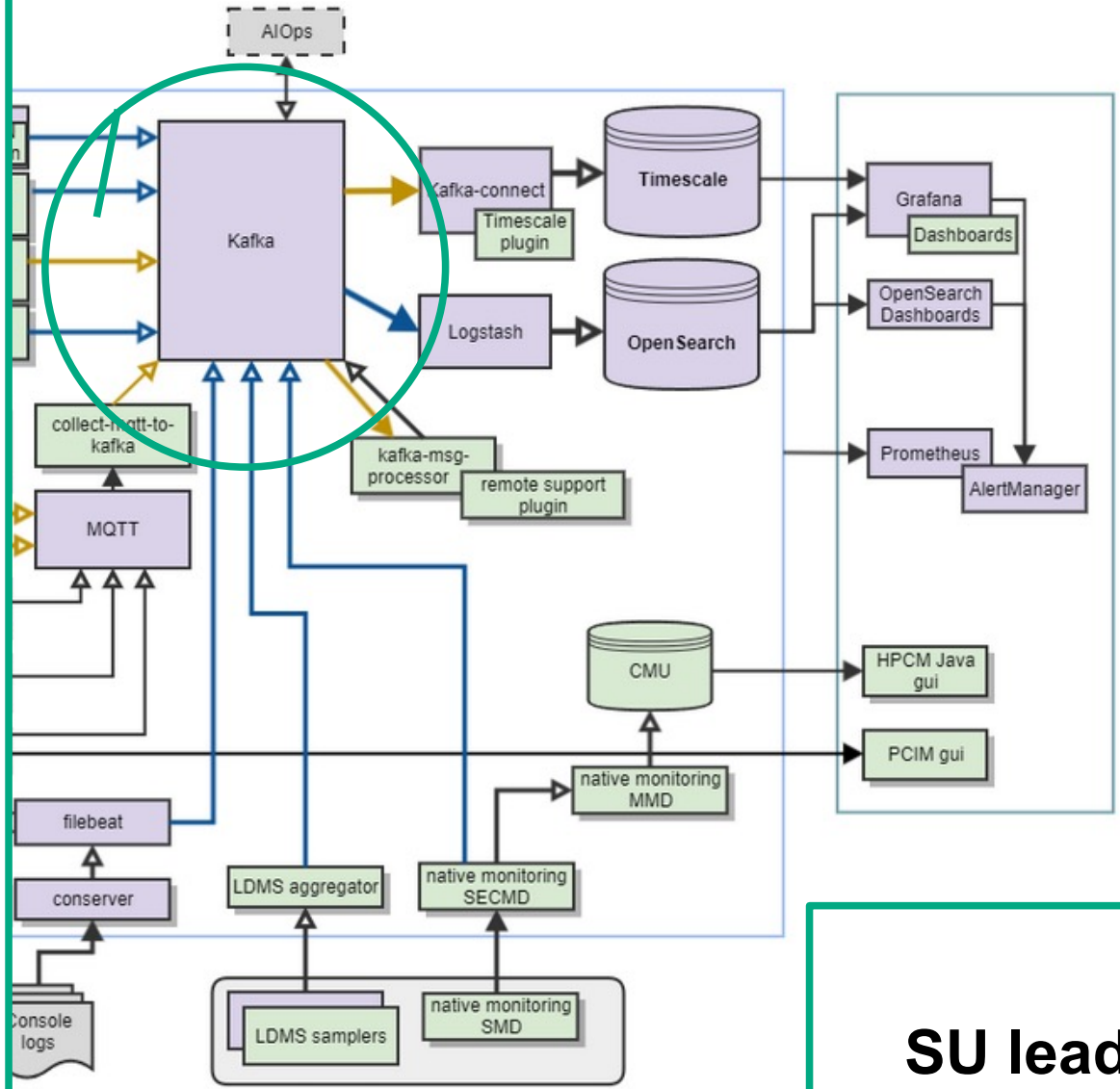
Most pipelines (the producers) are not generally configured at this point.

The material will now detail the above for understanding before covering producers.

```
admin:~ # cm monitoring kafka enable
Running enable command for kafka services
Configuration manager submitting node
configuration.
Populating Dataset...
Populating Dataset complete: 0.390s
0 of 17 nodes completed in 2.9 seconds, averaging
0.0s per node
17 of 17 nodes completed in 5.4 seconds, averaging
0.0s per node
17 of 17 nodes completed in 7.9 seconds, averaging
0.0s per node
17 of 17 nodes completed in 7.9 seconds, averaging
0.0s per node
admin:~ # cm monitoring kafka start
Running start command for kafka services
Running start command for confluent-zookeeper
services
Running start command for confluent-kafka services
Running start command for confluent-kafka-rest
services
Running start command for confluent-schema-
registry services
Running post-start cluster configuration
scripts...
Running start command for confluent-kafka-connect
services
Running start command for mosquitto services
Running start command for collect-mqtt-to-kafka
services
Running start command for kafka-msg-processor
services
Running start command for subsmon services
Running start command for kafka-connect-monitor
services
```



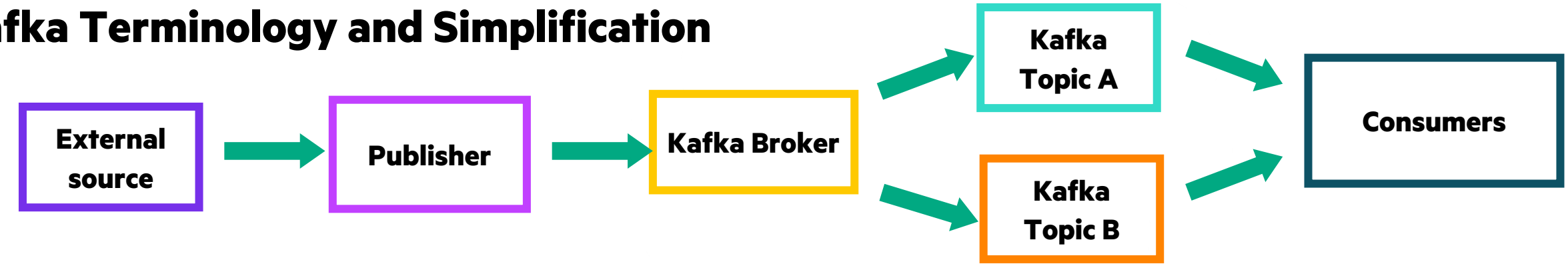
```
admin:~ # kfka-dist-setup
Checking and Distributing the zookeeper and Kafka
services among monitoring nodes
distributing zookeeper
zookeeper now running on 'admin' and ['leader1',
'leader2']
Zookeeper Status:
        .-- confluent-zookeeper.service status
        |    .-- zookeeper_node.lst contents
        |    |    .-- zookeeper.properties
contents
        |    |    |    .-- cmdb distributed
attribute
        |    |    |    |    .-- ping test
        |    |    |    |    |    .-- id (myid)
admin   OK   NA   OK   NA   OK   1
leader1 OK   OK   OK   OK   OK   2
leader2 OK   OK   OK   OK   OK   3

All nodes are already part of Kafka.
Would you like to reconfigure Kafka for all nodes
in /opt/clmgr/etc/kafka_node.lst [y|N]? y
Successfully configured
leader1.head.cm.white.hpcrb.rdlbas.ext.hpe.com
node to be part of the distributed kafka
Successfully configured
leader2.head.cm.white.hpcrb.rdlbas.ext.hpe.com
node to be part of the distributed kafka
Successfully configured
leader3.head.cm.white.hpcrb.rdlbas.ext.hpe.com
node to be part of the distributed kafka
Setting distributed node attributes...
```



SU leaders?

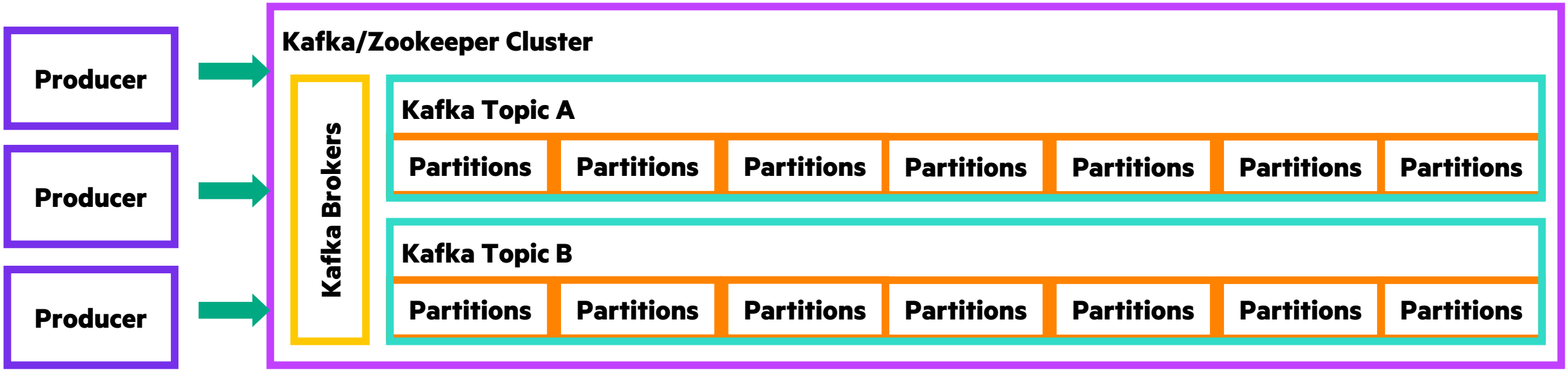
Prepared for CUG2024, Perth WA

Kafka Terminology and Simplification



Note: The colours on this diagram do not match those used later in the presentation

Kafka Terminology and Simplification



Multiple partitions for systems with SU leaders; If a broker fails the consumer can use partition replicas on the other brokers

Note: The colours on this diagram do not match those used later in the presentation

Kafka

- The broker service is confluent-kafka.service running on admin & leaders.
- Kafka is about large volume event streaming in a scaleable manner in a fault tolerant cluster providing storage for a shorter retention period crayex_telemetry is 24hrs and all others 168hrs)
- confluent-schema-registry.service: manages access to avro schema. Runs on the admin.
- confluent-kafka-rest.service: provides REST API for kafka. Provides an API to query, delete topics etc.
- ksqldb has been removed in 1.9

Kafka

- confluent-zookeeper.service on admin and a subset of leaders is a distributed configuration store. Aside from the admin the other instances are assigned to 2 other leaders. This used to be random but is now the first 2 leaders. (odd number needed and we use 3).
  - There is a cluster ID
  - A broker ID for the admin and leaders
  - Zookeeper elects a “leader” (not synonymous with SU leader) for each topic
  - The other brokers are replicas.
- kafka-msg-processor runs on admin, leader for filtering crayex telemetry data to just the information relevant to the power service.
- OpenSearch and TimeScaleDB provide persistent storage
  - confluent-kafka-connect. cm monitoring connect status and /var/log/kafka/connect.log (timescale)
  - Logstash: started with cm monitoring elk start and /var/log/logstash/ (opensearch)

Kafka

- List the topics: kafka-topics --bootstrap-server admin:9092 --list
- Logs: /var/log/kafka and /var/log/confluent
- cm monitoring kafka status
- cm monitoring kafka status -v
- Two command line ways to consume data, depending on avro on not kafka-console-consumer or kafka-avro-console-consumer
- Example with avro

```
# kafka-avro-console-consumer --bootstrap-server admin:9092 --topic metric_cooldev_craycdu12 --max-messages=1
{"name":"shinercdu","timestamp":1678452287000,"device_type":"CCDU","CDU_Current_Phase_1":{"float":0.0},"CDU_Current_
Phase_2":{"float":0.0},"CDU_Current_Phase_3":{"float":0.0},"VFD1_Current":{"float":6.9},"VFD2_Current":{"float":6.2}
,"VFD1_RunTime_Energy_Counter":{"int":8913},"VFD2_RunTime_Energy_Counter":{"int":8875},"Relative_Humidity":{"float":
<snip />
```

Kafka disk space usage – retention period

Retention periods are set in templates to make topic configuration persistent in case it needs to be re-created for whatever reason:

/etc/kafka/topics/templates/reduced.template:`retention.ms=43200000`

Specific topics will use the template e.g.

/etc/kafka/topics/crayex_telemetry.topic:`template=reduced.template`

If you have something different to that defined in /etc/kafka/server.properties or template it can be seen with:

```
kafka-topics --bootstrap-server admin:9092 --describe
```

Kafka disk space usage – retention period

In 1.9 or higher, `/opt/clmgr/tools/mon_kafka_topics_config.py` was introduced to do configuration at the top level.

Ensure it is defined in `/opt/clmgr/etc/monitoring_services.yml`:

```
confluent-schema-registry:
   pre_confd: []
   post_confd:
   # place this here as a workaround to make sure kafka is up and
going
   #- 81-kafka-topic-configure
   - /opt/clmgr/tools/mon_kafka_topics_config.py
```

Kafka disk space usage – retention period

Defaults may be too long. crayex_telemetry is 24hrs and all others 168hrs (7 days).

`log.retention.hours=168` in `/etc/kafka/server.properties`

Log retention can also be configured based on size e.g., log.retention.bytes. /etc/kafka/server.properties can be altered for the global settings

e.g., `log.retention.hours=48`

```
admin:~ # cm monitoring kafka restart
```

The broker has to be re-started to pick up changes to server.properties but topic level retention can be changed while running.

```
admin:~ # kafka-configs --bootstrap-server admin:9092 --entity-type
topics --alter --entity-name crayex_telemetry --add-config
retention.ms=43200000
```

This example uses 43200000ms (12hrs). `kafka-topics --bootstrap-server admin:9092 --describe --topic` **`crayex_telemetry`** could verify this

Kafka disk space usage – retention period

```
admin:/etc/kafka # grep ^log.retention.hours server.properties
log.retention.hours=168
admin:/etc/kafka # vi server.properties
admin:/etc/kafka # grep ^log.retention.hours server.properties
log.retention.hours=60
admin:/etc/kafka # grep "retention.ms" topics/templates/reduced.template
retention.ms=86400000
admin:/etc/kafka # grep ^template topics/crayex_telemetry.topic
template=reduced.template
admin:/etc/kafka # cm monitoring kafka restart
Running restart command for kafka services
Running restart command for confluent-zookeeper services
<truncated for brevity>
```

What data flows by default?

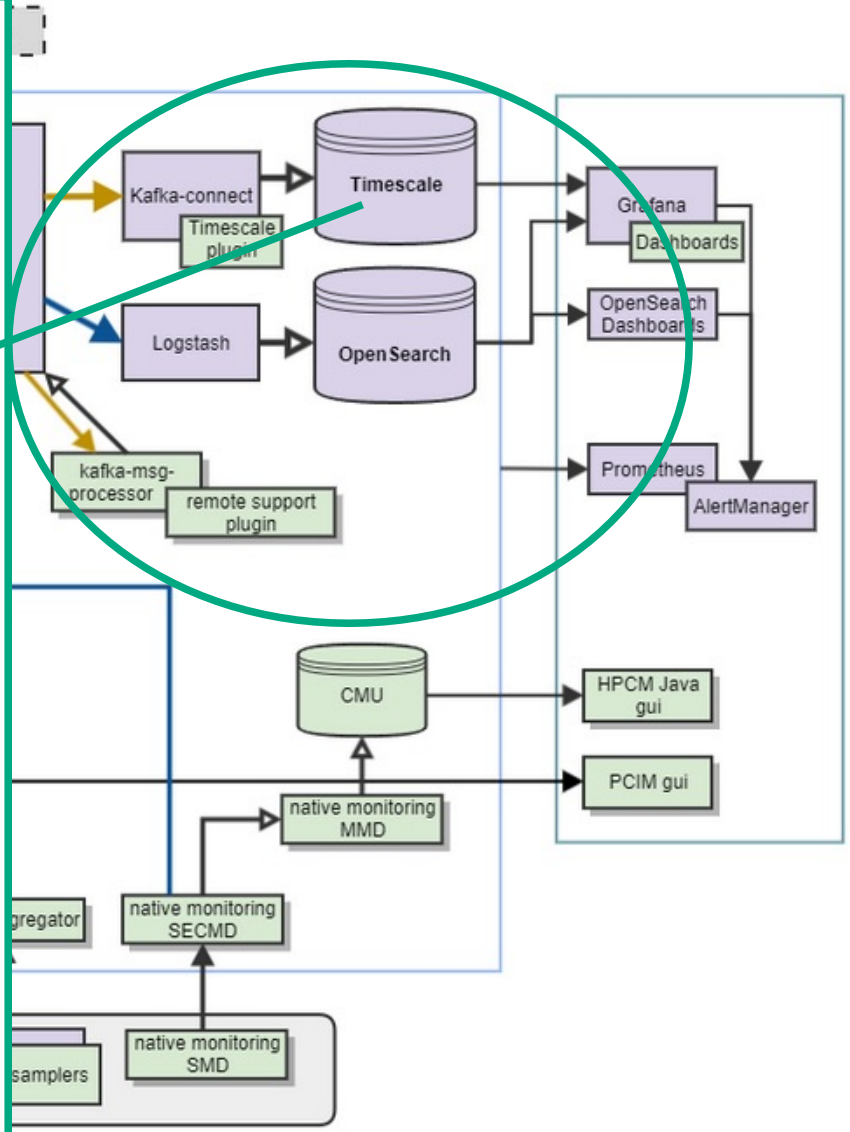
If remlog-collect is enabled, then **ilo/BMC redfish logs** will start to flow to kafka with its default configuration. Monitoring, generally, does not work out of the box. The only other pipelines enabled by default are those using subsmon when kafka is enabled and that will configure redfish **subscriptions to nC, cC, sC** and **logs** via logstash when elk is enabled.

```
admin:~ # systemctl status remlog-collect | cat
● remlog-collect.service - HPCM Remote log collector
     Loaded: loaded (/usr/lib/systemd/system/remlog-collect.service; enabled; vendor
preset: disabled)
     Active: active (running) since Tue 2023-12-05 09:45:09 CST; 21h ago
   Main PID: 36759 (RemLogCollect /)
      Tasks: 15
     CGroup: /system.slice/remlog-collect.service
             └─ 36759 "RemLogCollect /opt/clmgr/remlog-collect/tlib/twistd -o -n --
pidfile= -y /opt/clmgr/remlog-collect/sacmain.tac" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" ""

Dec 05 09:45:09 snowball-admin systemd[1]: Started HPCM Remote log collector.
```

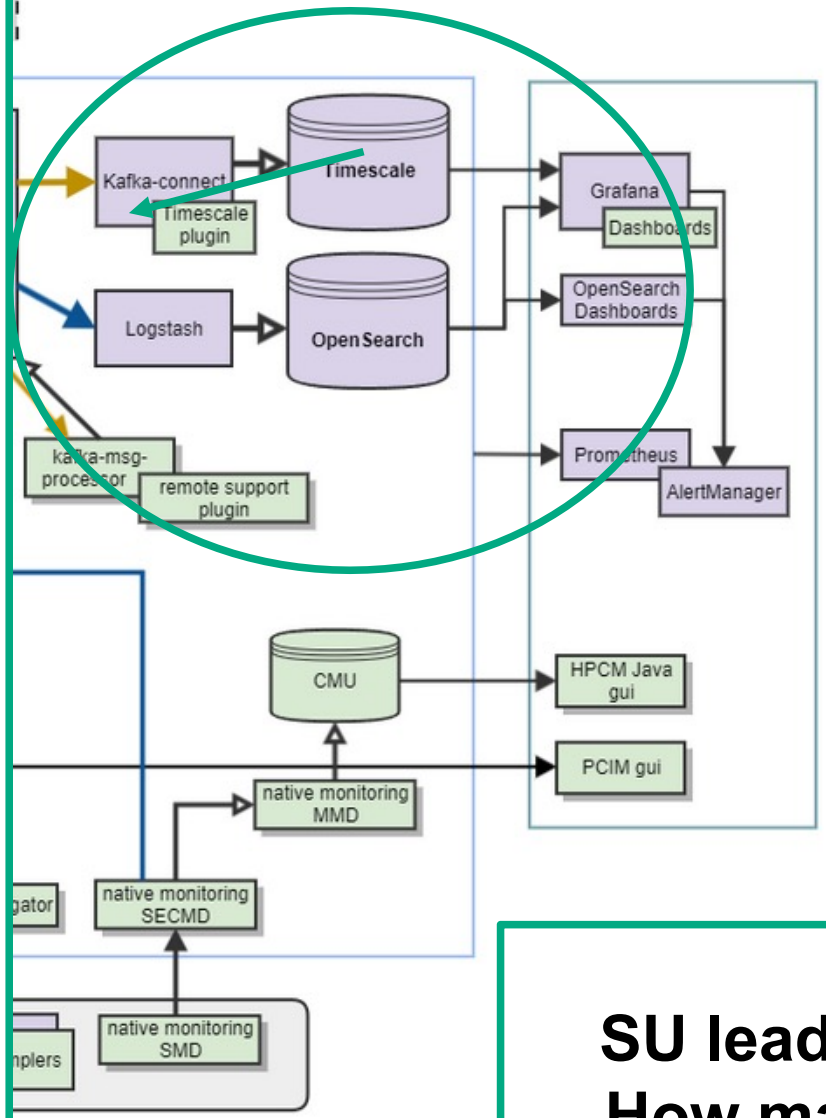
TimescaleDB

- Timescale is for time-series data and runs on admin and leaders
- It partitions tables on a time range; these partitions are called chunks
- Its core is based on postgres
- One node is the “access” node and acts a gateway to all read and write queries
- Others are “data” nodes which store the data and service queries
- Data replication occurs between data nodes
- Patroni, zookeeper and postgres streaming replication maintain access replicas running on two other nodes to handle a failure of the access node
- HA Proxy is used so that queries always go to the access primary or, if it fails, one of the access replicas
- Timescale has compression and retention built-in – default retention = 30 days

```
admin:~ # cm monitoring timescaledb enable
admin:~ # cm monitoring timescaledb start
admin:~ # cm monitoring timescaledb status
Data Node Status
--------------------

leader1 - postgres: active connection: success
pingable: True
leader2 - postgres: active connection: success
pingable: True
leader3 - postgres: active connection: success
pingable: True

Access Node Status
--------------------

admin - patroni: active role: leader postgres: running
lag: none connect: success

Zookeeper Status
--------------------

zookeeper: active

HAProxy Status
--------------------

haproxy: active
connect: success

monitoringdb Version
--------------------

1.15
```



```
admin:~ # cm monitoring timescaledb node add --help
<usage removed for brevity>

Adds a node, either data or access replica, to the
Timescaledb cluster. Timescaledb will be enabled and
started as part of the add.
<other options removed for brevity>

  --data-node             Add the specified nodes as
data nodes
  --access-replica        Add the specified nodes as
access replicas


<truncated for brevity>
```



SU leaders?
How many?

TimescaleDB

buroak-adm:~ # cm monitoring timescaledb status

Data Node Status

ld01 - postgres: active connection: success pingable: True

ld02 - postgres: active connection: success pingable: True

ld03 - postgres: active connection: success pingable: True

Access Node Status

admin - patroni: active role: leader postgres: running lag: none connect: success

Zookeeper Status

zookeeper: active

HAProxy Status

haproxy: active

connect: success

monitoringdb Version

1.5

TimescaleDB

```
buroak-adm:~ #  cm monitoring timescaledb show --metrics

 name                                    | category  | type    | timestamp scale | compression interval (sec) | retention interval
(sec)

-----------------------------------------------------------------------------------------------------------------------------------
-

 Actuator_2_Feedback_Position            | cooldev   | FLOAT8 | 1000            | 604800                     | 2592000

 CDU_Current_Phase_1                     | cooldev   | FLOAT8 | 1000            | 604800                     | 2592000

 CDU_Current_Phase_2                     | cooldev   | FLOAT8 | 1000            | 604800                     | 2592000

 CDU_Current_Phase_3                     | cooldev   | FLOAT8 | 1000            | 604800                     | 2592000

 CDU_Power                               | cooldev   | FLOAT8 | 1000            | 604800                     | 2592000

<snip />
```

```
buroak-adm:~ # ls /opt/clmgr/postgresql/var/lib/pgsql/14/data/log/

postgresql-Fri.log  postgresql-Mon.log  postgresql-Sat.log  postgresql-Sun.log  postgresql-Thu.log
postgresql-Tue.log  postgresql-Wed.log
```

Check /var/log/messages for haproxy and patroni

psql -h admin -p 5434 -U postgres -d monitoringdb

Timescale disk space usage – retention and compression

Timescale has compression and retention built-in – default retention = 30 days. The compression interval and retention interval can be changed using cm monitoring timescaledb with the following 2 options:

compression: Adjust compression policy for metric(s) stored in Timescaledb

retention: Adjust retention policy for metric(s) stored in Timescaledb

View the current settings with (will error at 1.9 or 1.10 if no metrics yet):

```
admin:~ # cm monitoring timescaledb show --metrics
 name                  | category | type   | timestamp scale| compression interval (sec)| retention (sec)
-------------------------------------------------------------------------------------------------------------
 CrayTelemetry.Current | cray     | FLOAT8 | 1000           | 604800                    | 2592000
 CrayTelemetry.Energy  | cray     | FLOAT8 | 1000           | 604800                    | 2592000
```

Timescale disk space usage – retention and compression

```
admin:~ # for i in slingshot cooldev pcm cray pdu disk; do cm
monitoring timescaledb retention --category  $i --interval 7d ;cm
monitoring timescaledb compression --category  $i --interval 1d ;
done
```

Valid units are d (day), w (week), m (month)

Compression can make a very large difference as its developers say it can "achieve 90%+ storage efficiencies".

Check the categories to list in the above with: `cm monitoring timescaledb show --categories`

Important: Metrics are only created as they come in once pipelines are configured. If you manually change you slingshot FMN configuration for example you will have to configure the retention/compression for those. As monitoring is configured you will need to repeat the above. Slingshot metrics are the big hitter for disk space.

Connectors

Unlike logstash for ELK, configuration is needed for the connectors.

There are many which are not needed – make your choices:

tsdb-aiops-* are not used by most sites

tsdb-disk-stats – Are you going to use SIM?

tsdb-pcm-monitoring or **tsdb-ldms-monitoring** – native has more like Slingshot NIC and GPUs

tsdb-metric_cooldev – Do you have supported CDUs, RDHX…? (see the release notes /opt/clmgr/doc)

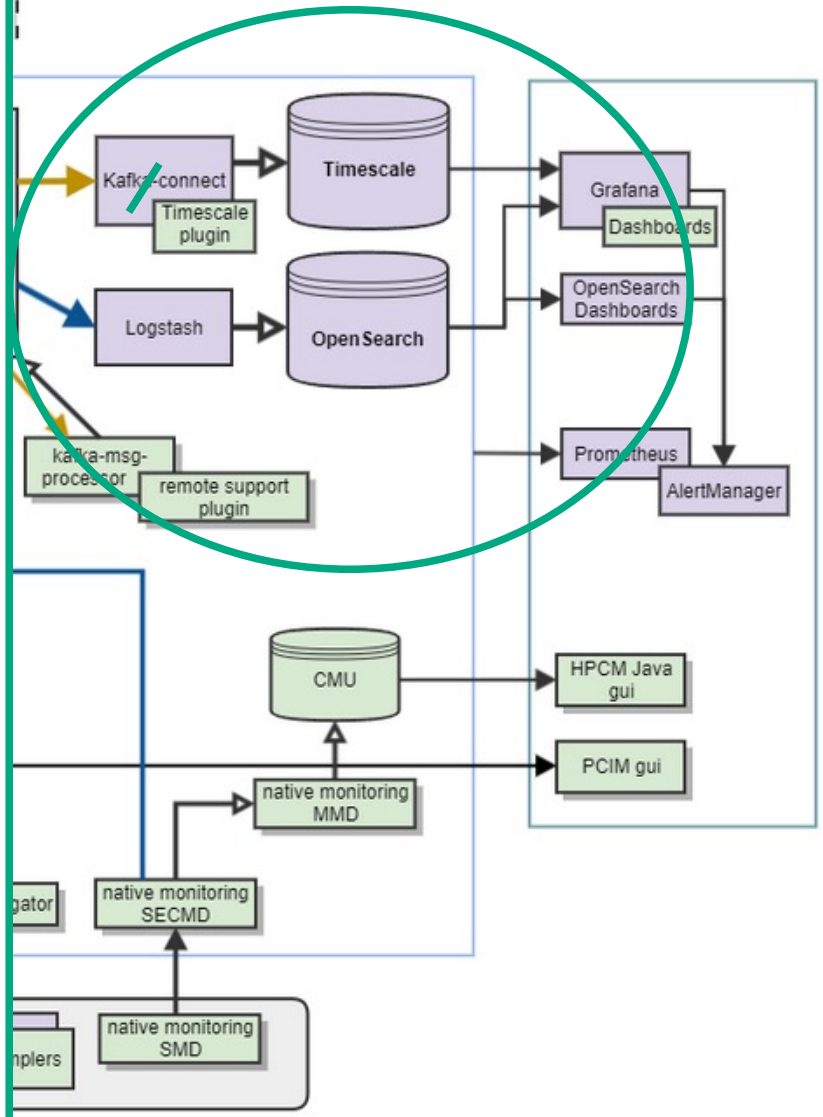
tsdb-slurm or tsdb-pbs – depending on scheduler

tsdb-pdu – Do you have supported PDUs (see the release notes /opt/clmgr/doc)

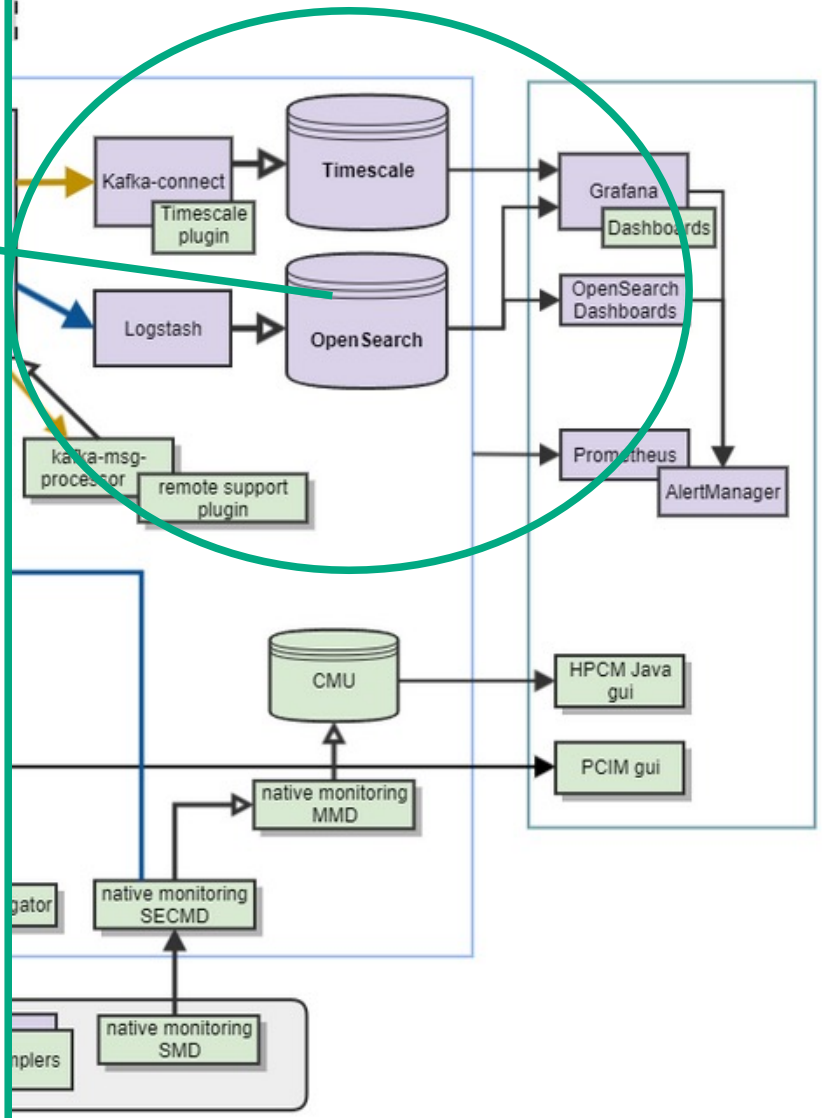
tsdb-slingshot, tsdb-slingshot-diag-perf, tsdb-slingshot-fabric-check, tsdb-slingshot-hardware

tsdb-cray-crayex_telemetry

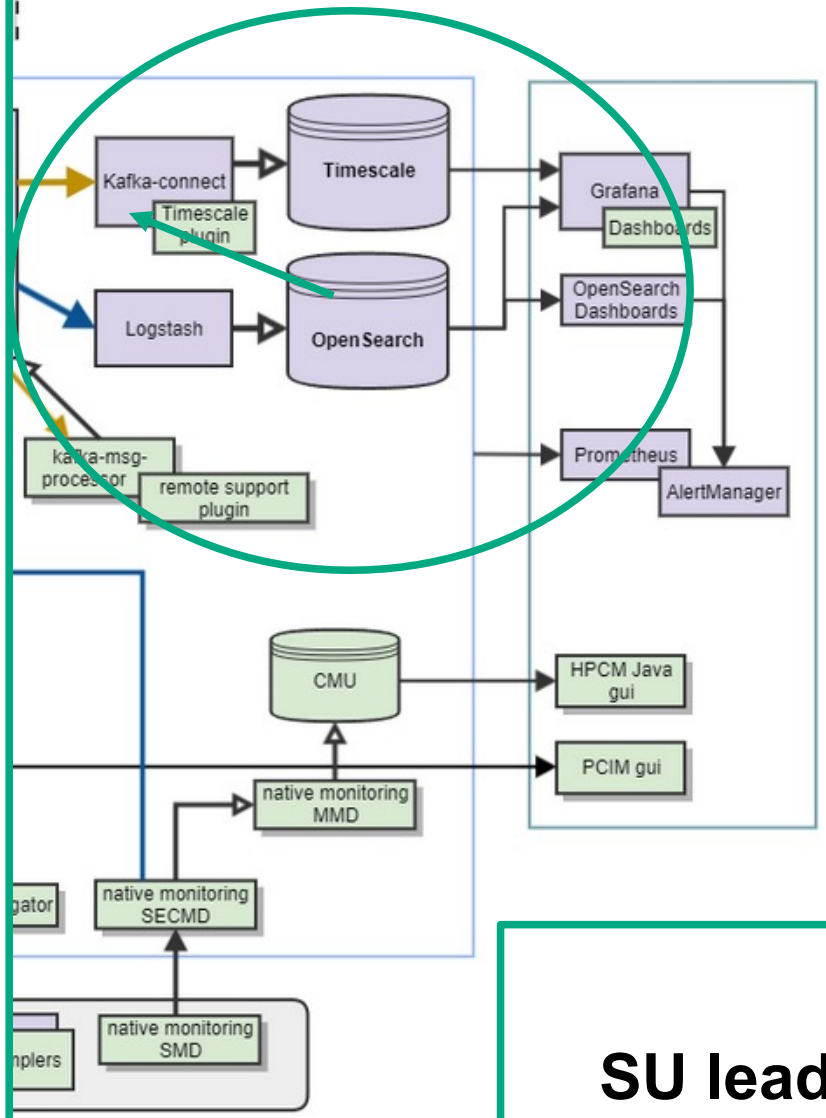
```
admin:~ # for i in tsdb-disk-stats tsdb-metric_cooldev
tsdb-pcm-monitoring tsdb-pdu tsdb-slingshot tsdb-
slingshot-diag-perf tsdb-slingshot-fabric-check tsdb-
slingshot-hardware tsdb-slurm tsdb-cray-
crayex_telemetry; do cm monitoring connect enable --
name $i ; done

admin:~ # clush -bw 'admin,leader*' 'systemctl restart
confluent-kafka-connect'
```



```
admin:/etc/kafka # cm monitoring elk enable
Running enable command for elk services
Configuration manager submitting node configuration.
Populating Dataset...
Populating Dataset complete: 0.427s
0 of 17 nodes completed in 2.9 seconds, averaging 0.0s per node
15 of 17 nodes completed in 5.4 seconds, averaging 0.0s per node
17 of 17 nodes completed in 7.9 seconds, averaging 0.0s per node
17 of 17 nodes completed in 7.9 seconds, averaging 0.0s per node
Node configuration complete.

admin:/etc/kafka # cm monitoring elk start
Running start command for elk services
Confirmed connection to opensearch DB
Running post-start cluster configuration scripts...
Running script: /opt/clmgr/tools/mon_elk_template_config.py
```



```
admin:~ # elk-dist-setup
Below nodes are going to be part of opensearch cluster :: -
['leader1', 'leader2', 'leader3']
If you want to remove some nodes, Remove nodes from
/opt/clmgr/etc/opensearch_node.lst file and press 'y' to continue
the setup
Want to continue [y|n] (default y):
Reconfigure Opensearch conf file and restart the service...
Successfully configured leader1 node to be part of the distributed
opensearch
Successfully configured leader2 node to be part of the distributed
opensearch
Successfully configured leader3 node to be part of the distributed
opensearch
Waiting for other data nodes to add in ES cluster
leader2: Checking logstash opensearch plugin...
leader1: Checking logstash opensearch plugin...
leader3: Checking logstash opensearch plugin...
admin: Checking logstash opensearch plugin...
Restart logstash service
SUCCESSFULLY added leader1 node into opensearch cluster
SUCCESSFULLY added leader2 node into opensearch cluster
SUCCESSFULLY added leader3 node into opensearch cluster
Setting distributed node attributes...
Configuration manager submitting node configuration.
Populating Dataset...
Populating Dataset complete: 0.549s
3 of 3 nodes completed in 3.1 seconds, averaging 0.0s per node
3 of 3 nodes completed in 3.1 seconds, averaging 0.0s per node
Node configuration complete.
```



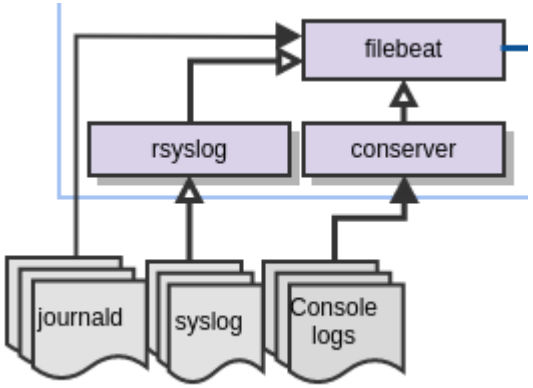
SU leaders?

ELK or OpenSearch?

OpenSearch is still called ELK under the tooling despite the move from ElasticSearch

"`cm monitoring elk enable|start`" will

- enable/start filebeat which capture syslog, console and journald
- enable/start logstash to get data from kafka to opensearch



OpenSearch disk space usage – ISM policy

Index State Management Policy

Set the initial default policy (if 1.9 – automatically done on =>1.10 so no --initial):

```
admin:~ # cm monitoring elk set
policy --initial

Successfully set the retention
policy for index event_cmc

Successfully set the retention
policy for index syslog

Successfully set the retention
policy for index rasdae...

<truncated for brevity>
```

OpenSearch disk space usage – ISM policy

Change the policy:

```
admin:~ # cm monitoring elk set policy --help
usage: cm monitoring elk set policy [-h] [-r RETENTION] [-n INDEX] [-i]

set retention for opensearch indices

options:
  -h, --help            show this help message and exit
  -r RETENTION, --retention RETENTION
                        Specify the number of retention days.
  -n INDEX, --index INDEX
                        Specify the opensearch index name
  -i, --initial         Set default 7d retention period to all indices
```

OpenSearch disk space usage – ISM policy

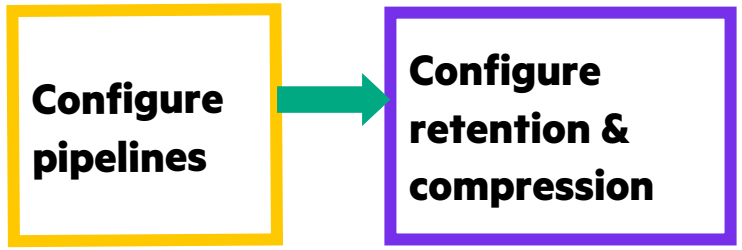
Important: The retention policy will only apply to indices created after the policy has been set.

You will need to do some manual clean up from today and before if reducing from 7 days default.

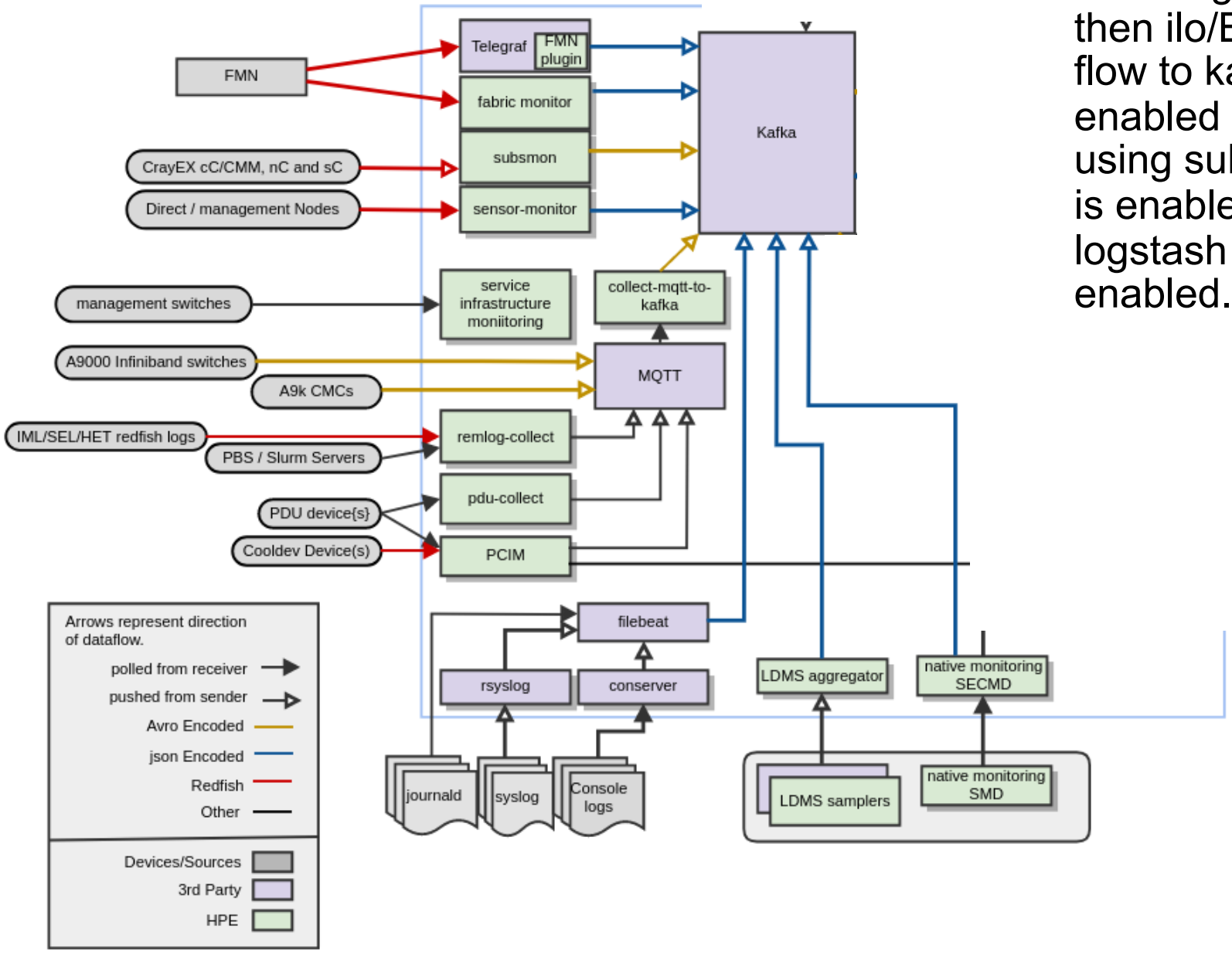
e.g.

```
for IND in $(curl -s http://admin:9200/_cat/indices?v | grep
2023.11 | awk '{print $3}'); do echo $IND; curl -X DELETE
admin:9200/$IND ; done
```

Producers



Producers



If remlog-collect is enabled, then ilo/BMC redfish logs will flow to kafka. Other pipelines enabled by default are those using subsmon when kafka is enabled and logs via logstash when elk is enabled.

Producers: The short story

- `cm monitoring native enable` **and** `start`

Probably need to add a gpu type

- `cm monitoring native metrics add -g slingshot -N <Max # NICs>` and `restart`
- `systemctl enable` **and** `start pcim`
- `cm monitoring dashboard grafana set --cdu|--cdu_ex2500 enable`
- Add cooling device other than Cray EX CDUs which are detected by default
- `systemctl enable` **and** `start sensor-monitor`

SU leaders: sensor-processor

- `cm monitoring slingshot enable` **and** `set <options>` **and** `start`
- `cm monitoring dashboard grafana set --slingshot enable`
- Double check FMN configuration

If changes are made on the FMN, new metrics with inappropriate compression can be created and consume disk. See last but one bullet.

- `cm node zypper|dnf -n <fmn> install slingshot-fabric-check`

Producers: The short story

- Set number of switches and switch groups in config file
- `systemctl enable` and `start` 3 services and timers on the fmn after installing an rpm
- Curl commands to enable dashboards

Unnecessary in 1.11

- Install hpe-telegraf and telegraf on the slurm controller

Pre 1.11 see details

- `cm monitoring slurm enable <options>` **and** `start`
- For slurm power dependent on hardware, configure the plugin config in slurm and HPCM

  Configure `/opt/clmgr/wlm-mon/conf/wlm-mon.yml`

IMPORTANT: tsdb compressions save >90% disk space

- Configure tsdb retention and **compression** after each stage **particularly slingshot pipelines**:

```
for i in slingshot cooldev pcm cray pdu disk; do cm monitoring
timescaledb retention --category  $i --interval 7d ;cm
monitoring timescaledb compression --category  $i --interval 1d
; done
```

Node level monitoring

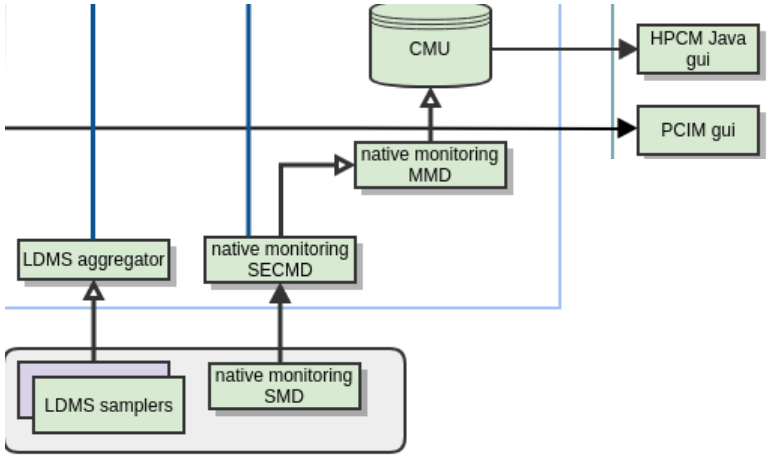
Do you want to use LDMS or native monitoring?

Most sites go with native monitoring because of the following dashboards:

"System Monitoring (CN)"

"* GPU Monitoring Dashboard"

"HPE Slingshot 200Gbps NIC"



Plus native monitoring populates the java GUI.

Native monitoring has default metrics for those (some require groups setups but more shortly)

Native monitoring

"System Monitoring (CN)", "System Monitoring (NCN)", "AMD GPU Monitoring Dashboard", "NVIDIA GPU Monitoring Dashboard", "HPE Slingshot 200Gbps NIC", "System Monitoring"

- `MainMonitoringDaemon` (Main or MMD) runs on the admin (as well as Sec and SMD for its NE (network entity/group)
- ssh to some nodes (elected from ones booted as can be seen in logs under /opt/clmgr/log and starts the `SecondaryServerMonitoringDaemon` (Sec)
- Sec connects to other nodes in its network group to start the `SmallMonitoringDaemon` (SMD).
- For hierarchical clusters these network groups a based on the rack leader; On ICE, the Sec runs on the rack leader and helps the admin with data for its rack.
- Customisable to run any command on a node to collect metrics: /opt/clmgr/etc/ActionAndAlertsFile.txt (AAA file)

Native monitoring - config considerations

Decisions: user, any ssh restrictions, frequency

Before starting anything consider your configuration! /opt/clmgr/etc/cmuserver.conf (there are more in the same section of the file e.g. `CMU_NONROOT_USER_ACCOUNT_KEY_TYPE`):

```
admin# grep ^CMU_MONIT /opt/clmgr/etc/cmuserver.conf
CMU_MONITORING_SYNCHRO=true
CMU_MONITORING=on
CMU_MONITORING_USER=root
CMU_MONITORING_USER_UID=default
CMU_MONITORING_USER_GID=default
CMU_MONITORING_INTERVAL=5
CMU_MONITORING_MEMLOCK=off
CMU_MONITORING_PRIORITY=0
CMU_MONITORING_HISTORY_FILES=300
CMU_MONITORING_STATUS_CHK=0
```

Native monitoring – dedicated user

```
# local user account on compute nodes to run CMU monitoring agents
# if 'root' then legacy mode: monitoring agents run as root user
# otherwise, the Administrator needs to make sure that the relevant
# CMU_MONITORING_USER settings are correct here below, save and exit,
# and then run the following command:
#
#  /opt/clmgr/tools/cm_config_nonroot_mon_user -c
#
# This command will create this user account in /opt/clmgr/users/hpemon/
# and create and synchronize user ssh keys between the admin node and all
# of the existing HPCM images (except for autoinstall images).
# The last step to enable a non-root monitoring user is to restart
# monitoring and redeploy the updated image to the compute nodes.
# NOTE #1: Do not create this user account beforehand, HPCM will create it.
# NOTE #2: Make sure to rerun the 'cm_config_nonroot_mon_user -c' command
#          whenever a new image is created and before it is deployed.
```

Native monitoring – dedicated user

Known bug in `/opt/clmgr/tools/cmu_mon_ssh_wrapper` fixed in 1.11

```
keypath=/opt/clmgr/etc/$user/.ssh/id_$key
```

Needs to be:

```
keypath=/opt/clmgr/users/hpemon/$user/.ssh/id_$key
```

Native monitoring

1.10 separated out a systemctl service cmu from cmdb – just on the admin.

Enable native HPCM monitoring either globally or per-node using -n

Before starting make sure compute nodes have Slingshot and GPU software installed.

```
admin:~ # cm monitoring native enable
admin:~ # cm monitoring native start
Adjusting nodes in network group admin
Adjusting nodes in network group rack8000
monitoring daemon started
```

Remember the connector from earlier:

```
admin:~ # cm monitoring connect enable --name tsdb-pcm-monitoring
```

Status really needs to be verified on nodes: **`ps -elf| grep Monit`**

```
admin:~ # cm monitoring native status
Running
```

Native monitoring - The Sec (sometimes termed aggregator)

Where should the Sec run?

```
cm monitoring native set -p
<priority> -n <node>
```

Meaning	Priority
The node can never become the aggregator node. i.e. Does not run the Sec	-1
The node can run the Sec if higher priority nodes are unavailable.	0
Higher pri nodes chosen first e.g. 10 chosen over 5	1 to n

Native monitoring - Additional metrics

"`cm monitoring native metrics add -g <group> [-N <Max # NICs>]`" can be used to add groups of metrics

This is in addition to anything added to the AAA file manually.

Once you have added all metrics and data is flowing review your timescale retention/compression!

Meaning	Group
Power and temperature metrics for HPE servers gathered out-of-band via iLO using HPE's Agentless Management Service	ams
AMD GPU metrics	gpu-amd
INTEL GPU metrics	gpu-intel
NVIDIA GPU metrics	gpu-nvidia
Power and temperature metrics for HPE Moonshot servers gathered out-of-band via ILOCM	moonshot
Metrics for each Slingshot NIC	slingshot

Native monitoring - Additional metrics

```
admin:~ # cm monitoring native metrics add -g slingshot -N 4
You are about to update the HPCM ActionsAndAlerts.txt file with metrics for
monitoring slingshot devices.

Continue? [y/N] y

Slingshot monitoring successfully configured.

Copy of original /opt/clmgr/etc/ActionAndAlertsFile.txt can be found in
/opt/clmgr/etc/ActionAndAlertsFile.txt_before_cm_config_slingshot

Please restart HPCM monitoring to enable these changes.

admin:~ # cm monitoring native restart
initiating monitoring shutdown...
Running: SendUdpMessage -client -host 127.0.0.1 -port 48559 -haltAll
checking every 5 seconds if monitoring is stopped...
starting monitoring...
Adjusting nodes in network group rack8000
Adjusting nodes in network group admin
monitoring daemon started
```

Native monitoring - tsdb retention/compression

Check if the SMD is running on the nodes an if so:

```
admin:~ # for i in slingshot cooldev pcm cray pdu disk; do cm
monitoring timescaledb retention --category  $i --interval 7d ;cm
monitoring timescaledb compression --category  $i --interval 1d ;
done
```

Valid units are d (day), w (week), m (month)

This could be done for just the pcm category but, I like to regularly make sure all the catgories are covered see “`cm monitoring timescaledb show --categories`”

Native Monitoring Troubleshooting

Use ps -elf| grep Monit to look for the daemons on nodes as the "cm monitoring native status" only checks for the MMD on the admin node.

```
1 S root     2266671       1  0  80   0 - 287499 -     Mar20 ?        00:04:02
/opt/clmgr/bin/MainMonitoringDaemon -a /opt/clmgr/etc/ActionAndAlertsFile.txt -m
/opt/clmgr/etc/MetaActionFile.txt -h 172.23.0.1 -s 1 -L 0 -r 0 -k 1 -b CMDB -t 5000000 -d 1 -e 1 -f
1 -R 0

1 S root     2267051       1  0  80   0 - 146205 -     Mar20 ?        00:00:06
/opt/clmgr/bin/SecondaryServerMonitoringDaemon -h 172.23.0.1 -S 172.23.0.1 -o 49141 -O 50303 -i
48558 -n /opt/clmgr/etc/NodesList.txt -a /opt/clmgr/etc/ActionAndAlertsFile.txt -t 5000000 -e 1 -f
1 -s 1 -L 0 -r 0 -p 172.23.0.1 -k 1 -b admin:9092

1 S root     2267389       1  0  80   0 - 84681 -      Mar20 ?        00:00:11
/opt/clmgr/bin/SmallMonitoringDaemon -h 172.23.0.1 -o 48560 -O 49722 -i 48557 -a
/opt/clmgr/etc/ActionAndAlertsFile.txt -t 5000000 -M 172.23.0.1 -f 1 -s 1 -L 0 -r 0 -p 172.23.0.1

0 S root     2669844       1  3  80   0 - 11586171 -   Mar07 ?        11:26:28 /usr/bin/java -
Xmx2048m -Xms2048m -server -Xmn1024m -XX:+UseConcMarkSweepGC -XX:+AggressiveOpts -
Djava.util.logging.config.file=/opt/clmgr/log/logging.properties -
Dlog4j.configuration=file:/opt/clmgr/log/logging.properties -Djdk.tls.acknowledgeCloseNotify=true -
Dcmu.monitoring.kafka=false -Dcmu.monitoring.runArchiver=true -
Dcmu.http.https.keystorePath=/opt/sgi/secrets/CA/private/server.p12 -cp
bin/cmuserver.jar:bin/cmu_plugins/*:plugins/* com.hpe.cmu.server.Main
```

Native Monitoring Troubleshooting

If daemons are not starting see if you can start them manually with the correct IPs and check the errors both in the terminal and logs

e.g.

```
root@leader2 ~]# LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/clmgr/lib/
/opt/clmgr/bin/SecondaryServerMonitoringDaemon -h 172.20.0.1 -S 10.64.0.68
-o 49142 -O 50304 -i 48558 -n /opt/clmgr/etc/NodesList.txt -a
/opt/clmgr/etc/ActionAndAlertsFile.txt -t 5000000 -e 1 -f 1 -s 1 -L 0 -r 0
-p 10.64.0.68 -k 2 -b admin:9092
```

Native Monitoring Troubleshooting

The debug level can be increased to 6 to get maximum verbosity:

```
# grep RING_DEBUG /opt/clmgr/etc/cmuserver.conf
CMU_MAIN_MONITORING_DEBUG_LEVEL=1
CMU_SEC_MONITORING_DEBUG_LEVEL=1
CMU_SMD_MONITORING_DEBUG_LEVEL=1
```

Change the debug level in cmuserver.conf and reduce the debug level again after troubleshooting. Restart native monitoring (cm monitoring native restart) to generate debug logs.

The admin node has logs for the 3 daemons as it runs all 3. There should be one node per network group (or NE - network entity) running the Sec. It runs through the nodes in the NE sequentially. Logs are moved to *.bak every time the daemons are re-started so you have logs from this instance and the previous instance.

MainMonitoringDaemon_<admin>.log, MainMonitoringDaemon_<admin.log.bak, SecondaryServerMonitoring_<node>.log, SecondaryServerMonitoring_<node>.log.bak, SmallMonitoringDaemon_<node>.log, SmallMonitoringDaemon_<node>.log.bak

There are also some logs for start and stop with smd or sec in the names for issues stopping or starting the process.

Native Monitoring

```
# cm monitoring native metrics show -n service0
service0 : time = 1676012585
service0 : __cm_monitoring_state__ = 5
service0 : kernel_version = 5.14.21-150400.24.21-default
service0 : cpuload = 0
service0 : memory_used = 1.641557
service0 : process_memory = 1.270441
service0 : page_cache = 0.374057
service0 : buffer_cache = 0.002942
service0 : uptime = 7.881000
<snip />
```

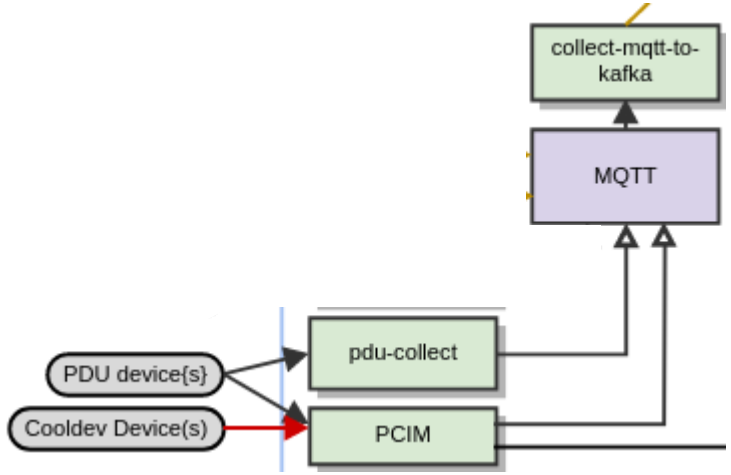
Power and cooling

Supported devices listed in the release notes (/opt/clmgr/doc). Not normally on the network when shipped.

6.2 Supported Power Distribution Units (PDUs)

A power distribution unit (PDU) reads AC power and energy measurements on cluster rack-level power domains. For the AC power measurement feature to function, the cluster must have one or more of the following PDUs:

- Server Technology Sentry3
- Server Technology Sentry4
- 880459-B21 (Raritan) HPE Mtrd 3P 39.9kVA/60A 48A/277V FIO PDU
- PX-5946V-F5V2 (Raritan) HPE Mtrd 3P 17.3kVA/48A 9brkr PDU
- P9R82A HPE G2 Metered 3Ph 17.3kVA/60309 4-wire 48A/208V
- P9R84A HPE G2 Metered 3Ph 22kVA/60309 5-wire 32A/230V



For more details on power management, see the HPE Performance Cluster Manager Power Consumption Management Guide.

Power and cooling

6.3 HPE Power and Cooling Infrastructure Monitor (PCIM) Supported Devices

---

The HPE Power and Cooling Infrastructure Monitor provides insight into the state of the hardware related to the power and water-cooling components of an HPE water-cooled solution. Supported devices include the following:

- HPE Apollo 9000 CDU (Cooling Distribution Unit)
- HPE Apollo 9000 Chassis (Power Supplies and Switches)
- HPE Cray EX CDU (1.2 MW and 1.6 MW)
- Apollo DLC Passive CDU (for A2k and A6500 clusters)
- HPE SGI 8600 CDU
- ARCS (Adaptive Rack Cooling System)
- SGI 8600 CRC (Cooling Rack Controller)
- Motivair RDHX (Rear Door Heat Exchanger)
- Raritan PDUs (Power Distribution Unit)
- HPE Cray EX VCDU (Virtual Cooling Distribution Unit)
- HPE PDUs
- ServerTech Cray ClusterStor Switch 63A 400V PDU (R4M34A)
- ServerTech Cray ClusterStor Switch 60A 415V PDU (R4M35A)

PCIM – Power and cooling infrastructure manager

```
admin:~ # systemctl enable pcim
admin:~ # systemctl start pcim
```

Remember the connector from earlier:

```
# cm monitoring connect enable tsdb-metric_cooldev
```

Cray EX: configures the CDUs and VCDUs automatically during the installation process when '`cm node update config --sync pcim -n admin`' is run.

Cray XD, Apollo 9000, HPE Apollo DLC CDUs, Adaptive rack cooling systems (ARCS)

Components, Rear-door heat exchangers (RXHX): '`cm cooldev`' used to add them

SGI 8600/ICE CDUs and CRCs, supported PDUs: `/opt/cmu/pcim/configure_snmp_device`

Once metrics are flowing, set your tsdb retention/compression!

cm monitoring dashboard grafana set --cdu enable

cm monitoring dashboard grafana set --cdu_ex2500 enable

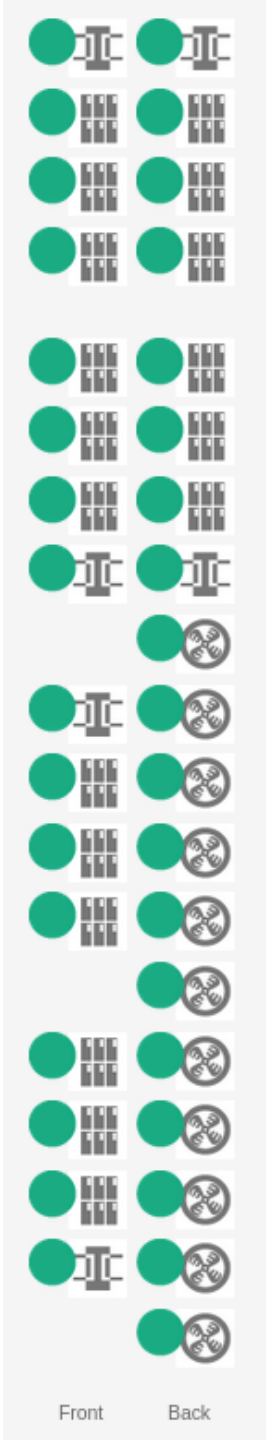
PCIM – Power and cooling infrastructure manager

```
admin:~ # /opt/cmu/pcim/configure_snmp_device -n x3000-pdu0 -i
172.24.253.200 -v 1 -c public

admin:~ # cm cooldev cdu show

x8000cdu1 10.176.0.1

x8000cdu0 10.176.0.1

admin:~ # cm cooldev cdu|rdhx|arcs add --name <NAME> --type
<2000|9000> --ip <IP> --mac <MAC>
```

"`/opt/cmu/pcim/tools/find_chassis.pl -r`" to create `/opt/cmu/pcim/config/.pcimchassis.conf` with IP to chassis mappings pulling information from the DB.

PCIM – Web GUI

```
# cat /opt/cmu/pcim/layout.txt
x1005cdu x1105cdu
x1005 x1105
x1006 x1106
x1007 x1107

x1008 x1108
x1009 x1109
x1010 x1110
x1010cdu x1110cdu
         x3100rdhx
x1011cdu x3101rdhx
x1011 x3102rdhx
x1012 x3103rdhx
x1013 x3104rdhx
 x3105rdhx
x1014 x3105rdhx
x1015 x3106rdhx
x1016 x3107rdhx
x1016cdu x3108rdhx
        x3109rdhx
```



PCIM - tsdb retention/compression

Data is now flowing so:

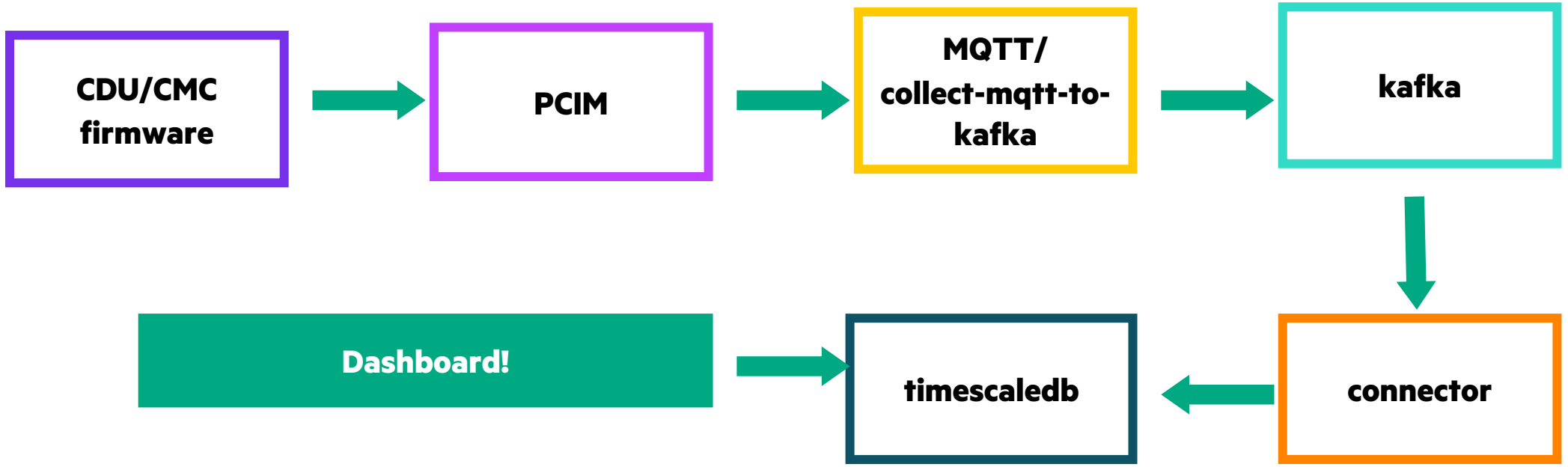
```
admin:~ # for i in slingshot cooldev pcm cray pdu disk; do cm
monitoring timescaledb retention --category  $i --interval 7d ;cm
monitoring timescaledb compression --category  $i --interval 1d ;
done
```

Valid units are d (day), w (week), m (month)

This could be done for just the cooldev category but, I like to regularly make sure all the catgories are covered see "`cm monitoring timescaledb show --categories`"

The CDU monitoring pipeline - Troubleshooting

Start troubleshooting at the start of the pipeline



The CDU monitoring pipeline - Troubleshooting

```
x9000c1:> grep . /var/volatile/cec/cdu/plc/*
/var/volatile/cec/cdu/plc/actuator1_fb:4294967272
/var/volatile/cec/cdu/plc/actuator2_fb:35
/var/volatile/cec/cdu/plc/belimo_valve_output_volts:4.8
<snip />
```

If it is CDU power monitoring check :

```
grep . /var/volatile/cec/cdu*/power_mon/*
```

Note: EX2500 uses 4U-F version CDU (Model Code: 04205). The 4U-F does not have an internal power meter.

The CDU monitoring pipeline - Troubleshooting

- **cm cooldev cdu|rdhx|arcs show**
- **systemctl status pcim**
- /opt/**cmu**/pcim/log
- /opt/**cmu**/pcim/tools/get_metric_data
- **systemctl status mosquito**
- **mosquitto_sub -t \# -v | grep cdu**
- **systemctl status collect-mqtt-to-kafka**
- /opt/clmgr/log/collect-mqtt-to-kafka/collect-mqtt-to-kafka.log
  /var/log/messages
- **kafka-avro-console-consumer --bootstrap-server admin:9092 --topic metric_cooldev_craycdu12 –max-messages=1|jq**
- **cm monitoring kafka status -v**
- /var/log/kafka/ and /var/log/confluent/

The CDU monitoring pipeline - Troubleshooting

- **`cm monitoring connect status`**
- `/var/log/kafka/connect.log`

- **`cm monitoring timescaledb status`**
- `/opt/clmgr/postgresql/var/lib/pgsql/14/data/log/`
- `/var/log/messages`
- **`psql -h admin -p 5434 -U postgres -d monitoringdb`**
  or see next slide

The CDU monitoring pipeline - Troubleshooting

```
# cm monitoring timescaledb show --metrics | grep cooldev
Actuator_2_Feedback_Position                | cooldev  | FLOAT8 | 1000           | 604800                    | 2592000
CDU_Current_Phase_1                         | cooldev  | FLOAT8 | 1000           | 604800                    | 2592000
CDU_Current_Phase_2                         | cooldev  | FLOAT8 | 1000           | 604800                    | 2592000
CDU_Current_Phase_3                         | cooldev  | FLOAT8 | 1000           | 604800                    | 2592000
CDU_Power                                   | cooldev  | FLOAT8 | 1000           | 604800                    | 2592000
CDU_Voltage_Phase_1_2                       | cooldev  | FLOAT8 | 1000           | 604800                    | 2592000
CDU_Voltage_Phase_2_3                       | cooldev  | FLOAT8 | 1000           | 604800                    | 2592000
CDU_Voltage_Phase_3_1                       | cooldev  | FLOAT8 | 1000           | 604800                    | 2592000
CWV_Valve_Actuator_Voltage                  | cooldev  | FLOAT8 | 1000           | 604800                    | 2592000
PLC_Temperature                             | cooldev  | FLOAT8 | 1000           | 604800                    | 2592000
PLC_to_VFD_Voltage                          | cooldev  | FLOAT8 | 1000           | 604800                    | 2592000
Primary_Facility_Flow                       | cooldev  | FLOAT8 | 1000           | 604800                    | 2592000
<truncated for brevity>
# cm monitoring timescaledb query --metric Primary_Facility_Flow
timestamp     | location | value
-------------------------------
1714480620000 | x9000cdu | 5.4   <truncated for brevity>
```

Node sensor information

Consider you hardware: sgi 8600/ICE, Cray EX, other ilo/BMC

Nearly every site will require sensor-monitor and its helper sensor-processor whereas many will not require HET for sgi hardware or may not require subsmon for Cray EX.

```
admin:~ # systemctl start sensor-monitor.service
```

```
admin:~ # systemctl enable sensor-monitor.service
```

SU leaders?

```
admin:~ # pdsh -w 'admin,leader*' systemctl enable sensor-processor.service
```

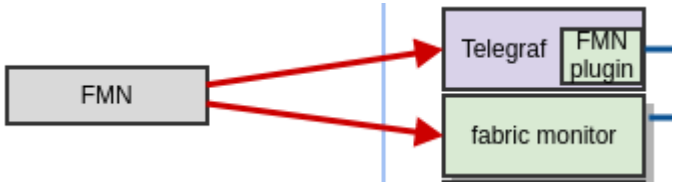
```
admin:~ # pdsh -w 'admin,leader*' systemctl start sensor-processor.service
```

EX? subsmon will establish a subscription on the controller to the admin or SU leader alias automatically as it was enabled with kafka.

```
admin:~ # systemctl restart subsmon
```

Slingshot

The architecture diagram is a simplification and there are multiple pipelines which need configuring to capture Slingshot monitoring for all Slingshot dashboards.



Health check dashboards become relevant one these multiple pipelines are configured and alerting is setup. i.e. graphs are an end point and everything involved in the pipeline must be configured first.

With 1.10: CHC and alerting are split out so new dashboard is “Slingshot Alerts”. See later.

1 pipeline: Node level (200Gbps NIC dashboard) data comes from native monitoring.

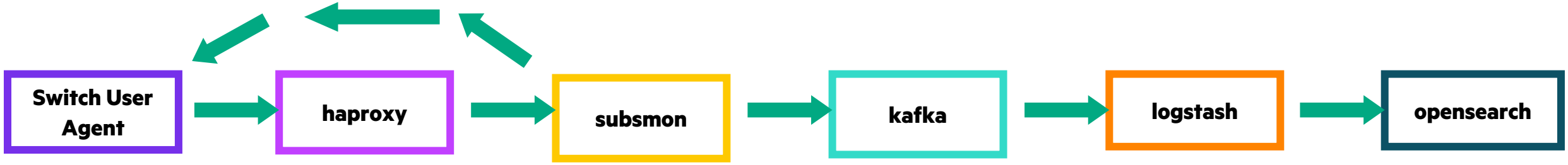
1pipeline: When we enabled kafka it will have enabled subsmon which sets redfish subscriptions on the switches providing switch hardware telemetry. Subscriptions are setup every time subsmon restarts. slingshot_CraySwitchHardwareTelemetry

Slingshot Monitoring

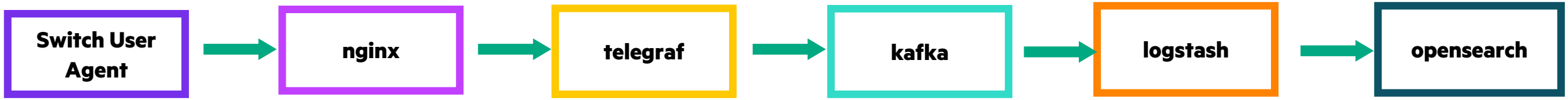
Individual pipelines for which the source needs to be working before even contemplating grafana and alerting.

6 different pipelines involved in slingshot (note some data is in both timescale and opensearch):

slingshot_CraySwitchHardwareTelemetry (redfish)



Congestion,PortState,LinkErrors,RFC,PortErrors,RoutingErrors,HardErrors (dashboard uses opensearch currently)

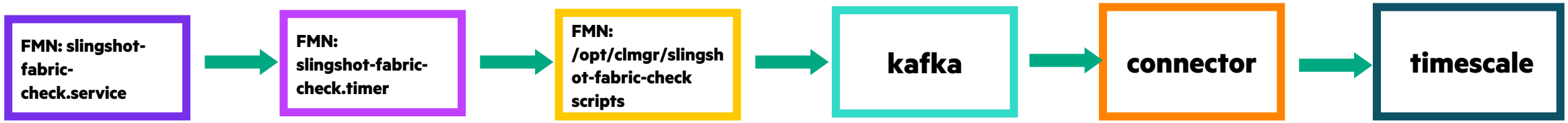


Slingshot Monitoring

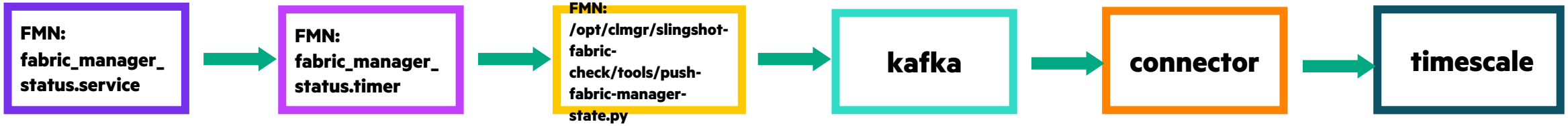
Individual pipelines for which the source to be working before even contemplating grafana and the steps in between.

6 different pipelines involved in slingshot (note some data is in both timescale and opensearch):

Slingshot - Switch performance



Slingshot – Switches online|offline, fabric online|offline, edge online|offline

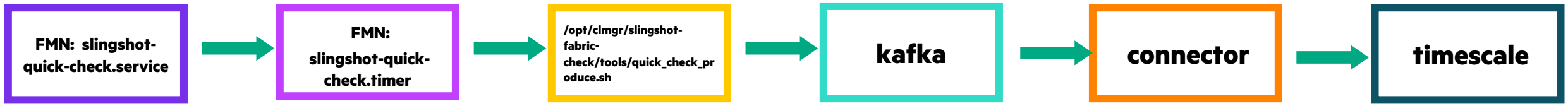


Slingshot Monitoring

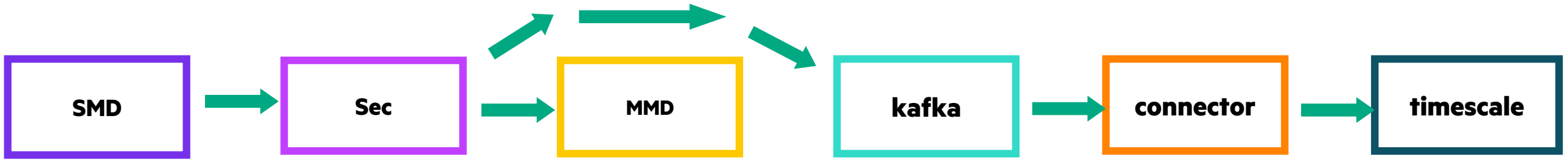
Individual pipelines for which the source to be working before even contemplating grafana and the steps in between.

6 different pipelines involved in slingshot (note some data is in both timescale and opensearch):

Slingshot – slingshot-switch-state-live (a more frequent check on a subset of metrics)



Slingshot – 200Gbps NIC (Native monitoring)



slingshot_CraySwitchHardwareTelemetry - Troubleshooting

- Ping the switch controller?
- List subscriptions: **`rest_agent_tool -b x3000c0r37b0 -u EventService/Subscriptions`**
- Check destination of those listed: **`rest_agent_tool -b x3000c0r37b0 -u EventService/Subscriptions/X`**
- **`systemctl status hmpad`** on the switch
- **`systemctl status haproxy`**
- `/var/log/messages`
- **`pdsh -g su-leader systemctl status subsmon-worker@* | grep Active -B 2`**
- `/opt/clmgr/log/subsmon-*.log`
- **`journalctl -u subsmon-worker@*`**
- **`curl -s admin:11890/metrics | grep subscriptions`** (worker ports 11890-5)

slingshot_CraySwitchHardwareTelemetry - Troubleshooting

- **kafka-avro-console-consumer --bootstrap-server admin:9092 --topic slingshot_CraySwitchHardwareTelemetry -max-messages=1|jq**
- **cm monitoring kafka status -v**
- /var/log/kafka/ and /var/log/confluent/
- Kafka topics from Redfish subscription:
  slingshot_CraySwitchHardwareTelemetry: CrayTelemetry.Power, CrayTelemetry.Voltage, CrayTelemetry.Current,CrayTelemetry.Temperature

  crayex_alerts: CrayAlerts.1.0.HsnLinkDownDetected, CrayAlerts.1.0.HsnLinkUpDetected, CrayAlerts.1.0.HsnLinkFlapDetected, CrayAlerts.1.0.HsnLinkErrorDetected, CrayAlerts.1.0.HsnTransceiverInstalled, CrayAlerts.1.0.ResourcePowerStateChanged

confluent-kafka-connect runs on the admin and leaders and is needed to get data in to timescaledb

- /var/log/kafka/connect.log
- **cm monitoring connect status | jq**

Slingshot - Switch telemetry and health (1 pipeline)

```
admin:~ # cm monitoring slingshot enable
Enabling Slingshot Telemetry Agent on all leader nodes.

Enabling nginx on all leader nodes.
```

Creating a config uses the first active FMN the command encounters or use --fmn for larger clusters and to be more precise and possibly use the IP to save lookups:

```
admin:~ # cm monitoring slingshot set -c config_1 --listener su-
aliases.head.cm.white.hpcrb.rdlbas.ext.hpe.com
Connecting to the FMN on the port chosen in the -c/--config.

Adding config_1 as a new configuration with fmn as fmn1 and
listener as su-aliases.head.cm.white.hpcrb.rdlbas.ext.hpe.com to
Slingshot Telemetry Configuration.
```

Switch telemetry and health (1 pipeline)

```
admin:~ #  cm monitoring slingshot start
Starting Slingshot Telemetry Agent on all leader nodes.

Starting nginx on all leader nodes.
```

We already did:

```
admin:~ # cm monitoring connect enable --name tsdb-slingshot
admin:~ # cm monitoring connect enable --name tsdb-slingshot-diag-perf
admin:~ # cm monitoring connect enable --name tsdb-slingshot-hardware
```

So metrics will start flowing and you will need to check retention/compression:

```
admin:~ # for i in slingshot cooldev pcm cray pdu disk; do cm monitoring
timescaledb retention --category  $i --interval 7d ;cm monitoring
timescaledb compression --category  $i --interval 1d ; done
```

```
admin:~ # cm monitoring dashboard grafana set --slingshot enable
Note: From 1.11, the above command is not needed since # cm monitoring
slingshot enable takes care of enabling the dashboards also.
```

Slingshot – FMN configuration comments

"`cm monitoring slingshot set`" sets up some basics on the first active or specified FMN but people often change things manually so check that side.

You can enable all sorts and the metrics will flow impacting storage but the following is what we graph:

```
fmn1:~ # fmctl get /switch-telemetry/settings
DOCUMENT/KEY              SUBKEY          VALUE
agentTelemetryRateLimit                   50000
collector                 value           http://172.23.255.230:9400/
counters                  values[0]       HardwareTelemetry.Temperature
counters                  values[1]       HardwareTelemetry.Power
counters                  values[2]       HardwareTelemetry.Voltage
counters                  values[3]       HardwareTelemetry.Current
counters                  values[4]       HardwareTelemetry.Rotational
documentSelfLink                          /switch-telemetry/settings
locality                  enable          true
periodicity               value           120
statistics                values[0]       PortErrors
statistics                values[1]       3635
statistics                values[2]       HardErrors
statistics                values[3]       RoutingErrors
```

Slingshot – FMN configuration comments

To manually set those:

```
fmn # fmctl update /switch-telemetry/settings \
statistics.values=["PortErrors","3635","HardErrors","RoutingErrors"]
fmn # fmctl update /switch-telemetry/settings \
counters.values=["HardwareTelemetry.Temperature","HardwareTelemetry.Power
","HardwareTelemetry.Voltage","HardwareTelemetry.Current","HardwareTeleme
try.Rotational"]
```

Slingshot produces a massive amount of data. Default periodicity is 60. Using say 120 that has a massive impact on reducing the amount of data.

```
fmn # fmctl update /switch-telemetry/settings periodicity.value=120
```

Note: It is recommended that the FMN uses UTC as the switch controller timezone cannot be changed from UTC

Congestion,PortState,LinkErrors,RFC,PortErrors,RoutingErrors,HardErrors

HTTP streaming from the switches

- telegraf setting on FMN as configured by "`cm monitoring slingshot config`": SU leader alias or admin port 9400 for nginx
- FM: `fmctl get /switch-telemetry/settings`
- FM: `systemctl status fabric-manager.service`
- `systemctl status sst-nginx` and `telegraf` (if leader nodes are in use those will run there)

Congestion,PortState,LinkErrors,RFC,PortErrors,RoutingErrors,HardErrors

- **`kafka-console-consumer --bootstrap-server admin:9092 --topic slingshot_CrayFabricTelemetry –max-messages=1|jq`**
- **`cm monitoring kafka status -v`**
- `/var/log/kafka/` and `/var/log/confluent/`

confluent-kafka-connect runs on the admin and leaders and is needed to get data in to timescaledb

- `/var/log/kafka/connect.log`
- **`cm monitoring connect status | jq`**

Slingshot - Switch fabric information (3 related pipelines)

I had the cluster manager repo selected and am using sles15sp5:

```
admin:~ # cm node zypper -n fmn1 install slingshot-fabric-check
Running cinstallman for node(s): fmn1
Using pdsh fanout (-f) 10
Running command: pdsh -f 10 -w ^/opt/clmgr/tmp/cinst <snip for brevity>
admin:~ # ssh fmn1
Last login: Tue Dec  5 08:23:39 2023 from 172.23.0.1
fmn1:~ # /opt/clmgr/slingshot-fabric-check/get_groups_switches.sh
Number of groups = 5

Group 0 has 1 switches
Group 1 has 2 switches
Group 2 has 4 switches
Group 3 has 4 switches
Group 4 has 4 switches
fmn1:~ # vi /opt/clmgr/etc/slingshot-fabric-check.conf
fmn1:~ # egrep '^GROU|^SWI' /opt/clmgr/etc/slingshot-fabric-check.conf
GROUPS="5"
SWITCHES="4"
```

Slingshot - Switch fabric information (different services/pipelines)

```
fmn1:~ # systemctl enable slingshot-fabric-check.service
Created symlink /etc/systemd/system/default.target.wants/slingshot-fabric-check.service →
/usr/lib/systemd/system/slingshot-fabric-check.service.
fmn1:~ # systemctl enable fabric_manager_status.service
Created symlink /etc/systemd/system/default.target.wants/fabric_manager_status.service →
/usr/lib/systemd/system/fabric_manager_status.service.
fmn1:~ # systemctl enable slingshot-quick-check.service
Created symlink /etc/systemd/system/default.target.wants/slingshot-quick-check.service →
/usr/lib/systemd/system/slingshot-quick-check.service.
fmn1:~ # systemctl enable slingshot-fabric-check.timer
Created symlink /etc/systemd/system/timers.target.wants/slingshot-fabric-check.timer →
/usr/lib/systemd/system/slingshot-fabric-check.timer.
fmn1:~ # systemctl start slingshot-fabric-check.timer
fmn1:~ # systemctl enable slingshot-quick-check.timer
Created symlink /etc/systemd/system/timers.target.wants/slingshot-quick-check.timer →
/usr/lib/systemd/system/slingshot-quick-check.timer.
fmn1:~ # systemctl start slingshot-quick-check.timer
fmn1:~ # systemctl enable fabric_manager_status.timer
Created symlink /etc/systemd/system/timers.target.wants/fabric_manager_status.timer →
/usr/lib/systemd/system/fabric_manager_status.timer.
fmn1:~ # systemctl start fabric_manager_status.timer
```

Slingshot - Switch fabric information

We enabled the connector earlier:
`admin:~ # cm monitoring connect enable --name tsdb-slingshot-fabric-check`

This creates a lot of data every time the systemd timers fires for large systems so:

```
admin:~ # for i in slingshot cooldev pcm cray pdu disk; do cm monitoring
timescaledb retention --category  $i --interval 7d ;cm monitoring
timescaledb compression --category  $i --interval 1d ; done
```

(It only really needs slingshot but I like to do the lot to make sure nothing has been missed!)

Slingshot - Switch fabric information

Enabling the dahsboards:

```
curl -sk -X POST -u admin:admin -H 'Content-Type: application/json' --data-binary
'@./fabric-summary.json' https://admin/grafana/api/dashboards/db | jq .

curl -sk -X POST -u admin:admin -H 'Content-Type: application/json' --data-binary
'@./group-health.json' https://admin/grafana/api/dashboards/db | jq .

curl -sk -X POST -u admin:admin -H 'Content-Type: application/json' --data-binary
'@./switch-overview.json' https://admin/grafana/api/dashboards/db | jq .

curl -sk -X POST -u admin:admin -H 'Content-Type: application/json' --data-binary
'@./switch-performance.json' https://admin/grafana/api/dashboards/db | jq .
```

Slingshot – Switch performance - Troubleshooting

- **systemctl status slingshot-fabric-check.service**
- **systemctl status slingshot-fabric-check.timer**
- /opt/clmgr/slingshot-fabric-check/logdir/
- **kafka-console-consumer --bootstrap-server admin:9092 --topic slingshot-perf --max-messages=1**

  IfInOctets, IfOutOctets, cfrx_rx_pause_pfc_cycles_00, cfrx_rx_pause_pfc_cycles_01, cftx_tx_pause_pfc_cycles_00, cftx_tx_pause_pfc_cycles_01, ifct_not_blocked_a, ifct_blocked_for_egress_fe_a, ifct_blocked_for_congestion_a, ifct_blocked_for_bandwidth_a, ifct_blocked_for_upstream_fe_a, ifct_blocked_for_incast_a, ifct_blocked_until_empty_a, ifct_blocked_other_reason, frf_empty_route_cntr, pcs_corrected_cw, pcs_uncorrected_cw, ifct_discard_acks_a, ifct_error_acks_a, ifct_flow_timeouts, ofct_flow_timeouts, llr_tx_replay_event, llr_rx_replay_event, llr_tx_poisoned_lossless, llr_rx_poisoned_lossless, ifct_over_injection_limit_a, ifct_blocking_over_il_a, ifct_blocking_for_redirect_a, ifct_redirect_acks_below_ecat_a, ifct_redirect_acks_above_ecat_a, ibuf_ibuf_full, frf_empty_route_uf_cntr, frf_empty_route_edge_cntr, cfrx_rx_pause_pfc_cycles_06, cfrx_rx_pause_pfc_cycles_07, cftx_tx_pause_pfc_cycles_06, cftx_tx_pause_pfc_cycles_07, ofct_cycles_n_flows_allocated_0, ibuf_ifct_disc, ifct_hdr_always_abort, ifct_intr

- **cm monitoring kafka status -v**
- /var/log/kafka/ and /var/log/confluent/

Slingshot - Kafka topics through http streaming from switches

confluent-kafka-connect runs on the admin and leaders and is needed to get data in to timescaledb

- `/var/log/kafka/connect.log`
- **`cm monitoring connect status | jq`**

Slingshot – Switches online|offline, fabric online|offline, edge online|offline

- **systemctl status fabric_manager_status.service**
- **systemctl status fabric_manager_status.timer**
- /opt/clmgr/slingshot-fabric-check/logdir/
- **kafka-console-consumer --bootstrap-server admin:9092 --topic slingshot-fabric-manager-state --max-messages=1**
- **cm monitoring kafka status -v**
- /var/log/kafka/ and /var/log/confluent/

confluent-kafka-connect runs on the admin and leaders and is needed to get data in to timescaledb

- /var/log/kafka/connect.log
- **cm monitoring connect status | jq**

Slingshot – slingshot-switch-state-live (a more frequent check on a subset of metrics)

- **systemctl status slingshot-quick-check.service**
- **systemctl status slingshot-quick-check.timer**
- /opt/clmgr/slingshot-fabric-check/logdir/
- **kafka-console-consumer --bootstrap-server admin:9092 --topic slingshot-switch-state-live –max-messages=1**
- **cm monitoring kafka status -v**
- /var/log/kafka/ and /var/log/confluent/

confluent-kafka-connect service runs on the admin and leaders and is needed to get data in to timescaledb

- /var/log/kafka/connect.log
- cm monitoring connect status | jq

Slingshot – 2.2

Things are constantly evolving and come changes 2.2 are:

- Ability to run multiple configurations which can be targeted at 1 or more switches so you have different configs running simultaneously
- Granular control per metric/event sub-category e.g. periodicity changed just for RFC3635 (where category is, for example, CrayFabricPerfTelemetry, CrayFabricCriticalTelemetry, CrayFabricHealth
- `fmctl get /telemetry/configurations/` and `fmctl get /telemetry/configurations/settings1 --raw | jq` and `switch-telemetry` will coexist initially in 2.2.
- Updated `fmn-show-telemetry-config` with new options such as --list and --detail – Recommended as it shows you if these config are active
- >=1.11 has more options to support the above in `cm monitoring slingshot set`

Slingshot – 2.2

```
[root@fmn1 ~]# fmn-show-telemetry-config  -l
+-------------------------------------+--------+
|            Configurations           | State  |
+-------------------------------------+--------+
| /telemetry/configurations/settings1 | ACTIVE |
+-------------------------------------+--------+
[root@fmn1 ~]# fmn-show-telemetry-config  -n settings1 --detail
{
    "categories": {
        "CrayFabricPerfTelemetry": {
            "Congestion": {
                "periodicity": 1.0
            },
            "CongestionDetails": {
                "periodicity": 15.0
            },
            "PauseDetails": {
                "periodicity": 15.0
            },
            "RFC1213": {
                "periodicity": 10.0
            },
            "RFC2819": {
                "periodicity": 10.0
            },
            "RFC2863": {
                "periodicity": 10.0
            },
            "RFC3635": {
                "periodicity": 10.0
            },
            "RFC4188": {
                "periodicity": 10.0
            }
        }
    },
    "collector": "https://172.23.0.200/telemetry-collector",
    "enable": true,
    "name": "/telemetry/configurations/settings1"
}
```

HPCM changes for SS 2.2

-t <telemetry level>
-t basic : critical health events, heartbeat, congestion metrics
-t vital: basic + pause details, hard errors, routing errors
-t max: vital + congestion details + RFC3635 metrics + port error

Also introduced a new service to monitor heartbeat from switches which runs when slingshot monitoring is started (cm monitoring slingshot start)

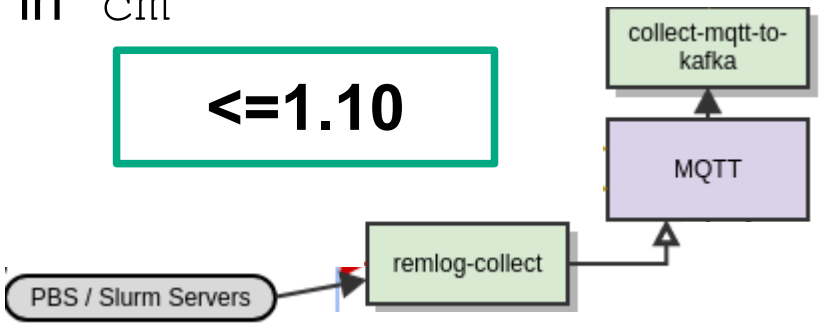
```
cm monitoring slingshot set -c c_name -L listener [-co collector]
[--fmn fmn_name] [-P fmn_port] [-S] [-u username] [-p password]
[-pr periodicity] [-t telemetry] [-h]
```

Workload manager

Consider: Slurm or PBS

We will cover Slurm as that is the most common.
We will not cover installing and setting up Slurm.
Basic Slurm installation and configuration can be done by following the step by step guide in "`cm wlm help slurm`"

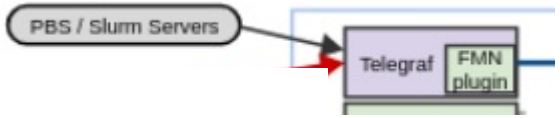
<=1.10



Both WLMs cover 2 aspects:

- Operations
- Power

>=1.10 with patch 11796



Slurm operations configuration

remlog-collect. Remote log collection tool.

1.10 with patch 11796 changes to use telegraf. (Enabled via the previous slingshot step)

Edit `/opt/clmgr/etc/remlog-collect.conf`

`slurm_server_name`	Resolvable Slurm controller hostname. Requires passwordless ssh.
`slurm_service_enable`	True|False
`slurm_install_path`	/usr/bin default. Change if Slurm was installed to a different path.
`slurm_time_limit`	Number of seconds to wait before the Slurm monitoring script responds with results. (20s default)
`slurm_poll_interval`	Polling interval, in seconds, to poll the Slurm metrics. (60s default). Always bigger than time_limit

Slurm operations configuration

After changes:

```
admin:~ # systemctl restart remlog-collect.service
```

1.10 with patch 11796 and in 1.11 use telegraf rather than remlog-collect. The Slurm server must have the `hpe-telegraf` RPM and the `telegraf` RPM installed.

```
slurm# cp /etc/telegraf/telegraf.d/slurm.disable /etc/telegraf/telegraf.d/slurm.conf
```

In the telegraf slurm.conf file verify the path for commands such as sinfo and scontrol is correct. You need to add the path with sinfo etc. With 1.11 but not 1.10 with 11796, the wlm-monitoring rpm must be installed on the slurm server to provide the script which may need updating in the telegraf config file to /opt/clmgr/wlm-mon/bin/chc_slurm_mon.py rather than /opt/clustertest/bin/chc_slurm_mon.py

```
commands = ["/opt/clmgr/bin/cm-python3 /opt/clustertest/bin/chc_slurm_mon.py
/opt/bin"]
```

```
slurm# systemctl restart telegraf.service
```

Remember the connector from earlier: **`cm monitoring connect enable --name tsdb-slurm`**

Slurm operations configuration

<= 1.10: Ensure the interval and poll period in /opt/clmgr/etc/remlog-collect.conf are sufficient. Poll period should always be greater than the interval.

>=1.10 and 11796: Ensure the interval and timeout settings are sufficient in/etc/telegraf/telegraf.d/slurm.conf or pbs.conf. Interval period should always be greater than the timeout.

The script name for PBS is `chc_pbs_mon.py`

The path on 1.11 is /`opt/clmgr/wlm-mon/bin`

The path (`/usr/bin`) at the end should be replaced with your slurm_install_path or pbs_install_path

```
slurm# time /opt/clustertest/bin/chc_slurm_mon.py /usr/bin
```

Do this during load and multiple times at different times to chose an appropriate value.

Slurm operations configuration

Enable monitoring of Slurm dashboards within Grafana:

1) remlog method: `admin:~ # cm monitoring dashboard grafana set --slurm enable`

2) telegraf method:

```
admin:~ # grep path /etc/grafana/provisioning/dashboards/dashboard.yaml
path: /var/lib/grafana/dashboards/flat
```

The above path maybe be :

```
path: /var/lib/grafana/dashboards/cray-EX
```

The following may be cray-EX or flat:

```
admin:~ # cp /var/lib/grafana/dashboards/generic/slurm_job_metrics.json.disable \
/var/lib/grafana/dashboards/flat/slurm_job_metrics.json
```

systemctl restart grafana-server.service

SLURM monitoring setup (from 1.11)

```
cm monitoring slurm enable -h
usage: cm monitoring slurm enable [-h] [--install-path INSTALL_PATH] [--timeout TIMEOUT] [--interval INTERVAL] [--server-name SERVER_NAME]

Enable Slurm Monitoring and allows users to make slurm configuration changes.

options:
  -h, --help            show this help message and exit

optional arguments:
  --install-path INSTALL_PATH
                        Specify the installation path to Slurm. The default path is /usr/bin.
  --timeout TIMEOUT     Specify the number of seconds to wait before the slurm monitoring script responds with results.The default is 20 seconds.
  --interval INTERVAL   Specify the polling interval in seconds to poll the slurm metrics.The default is 60 seconds.
  --server-name SERVER_NAME
                        Specify the hostname of the node upon which slurm is installed. If it is admin, specify localhost.Default is localhost.

This command only enables Slurm Monitoring.
Slurmctld service should be running and Telegraf related RPMS must be installed on the slurm server node before executing this command.
```

```
cm monitoring slurm disable -h
usage: cm monitoring slurm disable [-h] [--server-name SERVER_NAME]

Disable Slurm Monitoring

options:
  -h, --help            show this help message and exit

optional arguments:
  --server-name SERVER_NAME
                        Specify the hostname of the node upon which slurm is installed. If it is admin, specify localhost.Default is localhost.

This command only disables Slurm Monitoring.
```

Slurm power monitoring configuration

```
slurm:~ # cat /opt/clmgr/etc/wlm-mon.yml
#
# ========== Slurm ==========
#
slurm:
  job_monitor_module: True
  pm_counters: False # possibly to be
introduced upstream
  power_api_timeout: 20
#
# ========== PBS ==========
#
pbs:
  job_monitor_module: False
  power_api_timeout: 20
```

1.11 path is /opt/clmgr/wlm-mon/conf

Slurm power monitoring configuration – jobmonitor (1.10 and patch)

On EX and clusters that do not have SU leaders (but not HPE Apollo 9000) for job-level energy and power consumption monitoring: use the jobmonitor service

```
slurm# grep ^PlugStackConfig /etc/slurm/slurm.conf
PlugStackConfig = /opt/clmgr/etc/jobmonitor_plugstack.conf
```

Create /opt/clmgr/etc/jobmonitor_plugstack.conf with the correct version un-commented:

```
#optional /opt/clustertest/lib/jobmonitor_slurm-20.02_power_plugin.so 172.23.0.1 4442
#optional /opt/clustertest/lib/jobmonitor_slurm-20.11_power_plugin.so 172.23.0.1 4442
#optional /opt/clustertest/lib/jobmonitor_slurm-21.08_power_plugin.so 172.23.0.1 4442
#optional /opt/clustertest/lib/jobmonitor_slurm-22.05_power_plugin.so 172.23.0.1 4442
optional /opt/clustertest/lib/jobmonitor_slurm-23.02_power_plugin.so 172.23.0.1 4442
slurm# systemctl restart slurmctld.service
```

1.11 path is /opt/clmgr/wlm-mon/{conf,lib}

Slurm power monitoring configuration – power API

```
admin:~ # cat /opt/clmgr/etc/wlm-mon.yml
#
# ========== Slurm ==========
#
slurm:
  job_monitor_module: False
  power_api_timeout: 20
#
# ========== PBS ==========
#
pbs:
  job_monitor_module: False
  power_api_timeout: 20
```

1.11 path is /opt/clmgr/wlm-mon/conf

Older versions may not have this file as the only possibility was to use the power service

```
admin:~ # vi /opt/clmgr/etc/clmgr-power.conf

admin:~ # grep ^node_power_monitoring_active /opt/clmgr/etc/clmgr-power.conf

node_power_monitoring_active = True

admin:~ # systemctl restart clmgr-power
```

Slurm power monitoring configuration – power API

```
slurm # grep ^PlugStackConfig /etc/slurm/slurm.conf
PlugStackConfig = /opt/clmgr/etc/plugstack.conf
```

Create /opt/clmgr/etc/plugstack.conf with the correct version uncommented:

```
#optional /opt/clustertest/lib/chc_slurm-20.02_power_plugin.so 172.23.0.1 8888
#optional /opt/clustertest/lib/chc_slurm-20.11_power_plugin.so 172.23.0.1 8888
#optional /opt/clustertest/lib/chc_slurm-21.08_power_plugin.so 172.23.0.1 8888
#optional /opt/clustertest/lib/chc_slurm-22.05_power_plugin.so 172.23.0.1 8888
optional /opt/clustertest/lib/chc_slurm-23.02_power_plugin.so 172.23.0.1 8888


slurm # systemctl restart slurmctld.service
```

1.11 path is /opt/clmgr/wlm-mon/{conf,lib}

The slurm monitoring pipeline

<=1.10

```
admin# systemctl status mosquitto
admin# mosquitto_sub -t \# -v | grep slurm
admin# systemctl status collect-mqtt-to-kafka
/opt/clmgr/log/collect-mqtt-to-kafka/collect-mqtt-to-kafka.log
admin# systemctl status remlog-collect
/opt/clmgr/log/remlog-collect
```

ssh to slurm server works?

>=1.10 plus patch 11796

```
slurm# systemctl restart telegraf.service

/var/log/telegraf/
```

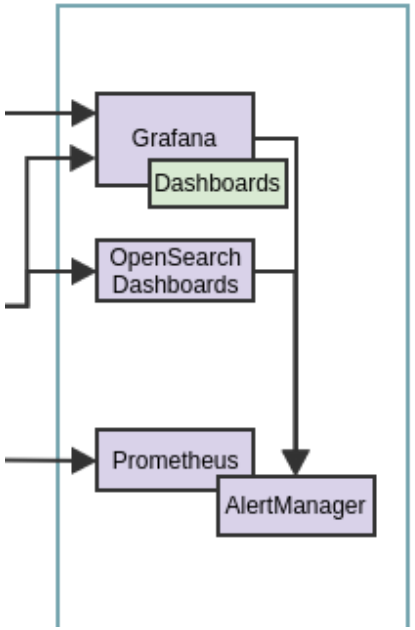
The slurm monitoring pipeline

- **`kafka-console-consumer --bootstrap-server admin:9092 --topic slurm_jobs --max-messages=1 | jq`**

  (with telegraf use `kafka-console-consumer`)

- **`cm monitoring kafka status -v`**
- `/var/log/kafka/` and `/var/log/confluent/`
- **`systemctl status logstash`**
- `/var/log/logstash/`
- **`cm monitoring elk status -v`**
- **`curl -s http://admin:9200/_cat/indices`**

Alerting, SIM and rackmap



Service Infrastructure Monitoring is monitoring of the monitoring and best practice

Alerting and SIM: The short story

- `cm monitoring alerting enable`

1.10 plus 11796 or higher

- `cm monitoring alerting opensearch` **or** `grafana --enable-rule <appropriate rules>`
- `cm monitoring alerting route email --from <email> --to <email> --smtp <smtp.server:25> --alert-group <group>`
- `cm sim enable` **and** `start` **and** `add {--service-group monitoring-services|suleader-services}`
- `cm monitoring rackmap map component-drift` **or** `power` **or** `cpu-temperature` **or** `slingshot-switch-status -l`

1.11

Unified Alerting

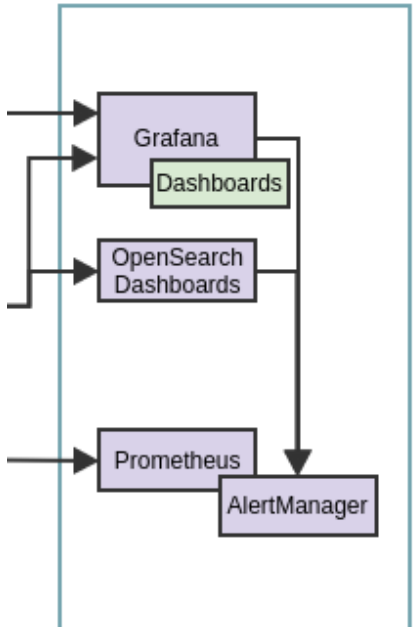
Unified alerting is discussed in a separate CUG presention. (May 9 – Tech Session 6A).

The material on the next 7 slides is just very basic in case of questions.

Cluster Health Check- CHC

<=1.9 If you want a dashboard with a name containing health check to work then you need alerting setup!

- Obviously dependent on everything which has gone before
- 1.10 has both alerta/elastalert (old style – retired in 1.11) and alertman (new style)
- Unified notifications/alerting using alertman in >=1.10
- **1.10 segregates unified alerting from CHC (new dashboards)**
- There has been a TKT specifically on unified alerting
- Not covering migrating any custom alerts
- "`cm heath alertman`" is a single interface to
  - OpenSearch data
  - Timescale data



Unified Alerting

Related service checking and initial configuration controlled by “`cm monitoring alerting`”

```
admin:~ # cm monitoring alerting enable
Enabling Alerting..
Configuration manager submitting node configuration.
Populating Dataset...
Populating Dataset complete: 0.583s
0 of 1 nodes completed in 3.1 seconds, averaging 0.0s per node
0 of 1 nodes completed in 5.6 seconds, averaging 0.0s per node
0 of 1 nodes completed in 8.1 seconds, averaging 0.0s per node
1 of 1 nodes completed in 10.6 seconds, averaging 5.5s per node
1 of 1 nodes completed in 10.6 seconds, averaging 5.5s per node
Node configuration complete.
Checking and starting the dependent services...

Creating channels for alerting tools...
Successfully enabled the alertmanager contact point in grafana
Successfully enabled the notification policy for the contact points in grafana
Creating the opensearch notification channels
Successfully enabled 'Alertmanager' notification channel.
Successfully enabled 'Kafka' notification channel.
Waiting for channels to ready for alerts..

Enabling Alert rules in opensearch...
Validating and Enabling opensearch rule(s)..
Successfully enabled the monitor for alert rule 'CDU Alert'
Successfully enabled the monitor for alert rule 'IML Alert'
Successfully enabled the monitor for alert rule 'Rasdaemon Alert'
Successfully enabled the monitor for alert rule 'Syslog Alert'
Successfully enabled the monitor for alert rule 'Hardware Alert'
Successfully enabled the monitor for alert rule 'Native Alert'
Successfully enabled the monitor for alert rule 'Node Down'
Successfully enabled the monitor for alert rule 'Node UP'
Successfully enabled the monitor for alert rule 'Slingshot Health Alert'
Successfully enabled the monitor for alert rule 'Slingshot Rosetta Alert'

Enabling Alert rules in grafana...
Successfully enabled the slingshot_congestion_rxPausePercent alert rule in grafana
Successfully enabled the slingshot_congestion_txPausePercent alert rule in grafana

Adding respective alerting dashboard in grafana...
Successfully added the respective dashboards.
```

Unified Alerting

```
admin:~ # cm monitoring alerting status
Alerting is enabled.

Dependent services status:
                .- service status
                |               .- api status
opensearch      OK              OK (200)
grafana-server  OK              OK (200)
alertmanager    OK              OK (200)

Alert Rules Status:
                                        .- datasource
                                        |           .- state
                                        |           |           .- notification
CDU Alert                               OS          Enabled     Email
IML Alert                               OS          Enabled     Email
Rasdaemon Alert                         OS          Enabled     Email
Syslog Alert                            OS          Enabled     Email
Hardware Alert                          OS          Enabled     Email
Native Alert                            OS          Enabled     Email
Node Down                               OS          Enabled     Email
Node UP                                 OS          Enabled     Email
PBS jobsteps Alert                      OS          Disabled    Email
PBS Alert                               OS          Disabled    Email
Slingshot Health Alert                  OS          Enabled     Email
Slingshot Rosetta Alert                 OS          Enabled     Email
SLURM Alert                             OS          Disabled    Email
slingshot_congestion_rxPausePercent     TS          Enabled     Email
slingshot_congestion_txPausePercent     TS          Enabled     Email

Note: Datasource - OS-Opensearch and TS-Timescale.
Notification: Email - Routed the alerts for the alert group of the rule to Email. OS-Routed the
alerts of the rule to OS
```

```
cm monitoring alerting opensearch --enable-rule
/opt/clmgr/alerting/opensearch-
alerting/alert_rules/slurm/slurm_node_down.yml
```

Unified Alerting

```
admin:~ # cm health alertman -s
```

Alert Status	Count
Critical	2
Warnings	52
Active	54

Group	Severity	Alerts
cooldev	ok	critical : 0, warning : 0
iml	ok	critical : 0, warning : 0
native	ok	critical : 0, warning : 0
node_status	ok	critical : 0, warning : 0
rasdaemon	ok	critical : 0, warning : 0
redfish	critical	critical : 1, warning : 3
sim	ok	critical : 0, warning : 0
syslog	warning	critical : 0, warning : 49
switch	critical	critical : 1, warning : 0
wlm	ok	critical : 0, warning : 0
others	ok	critical : 0, warning : 0

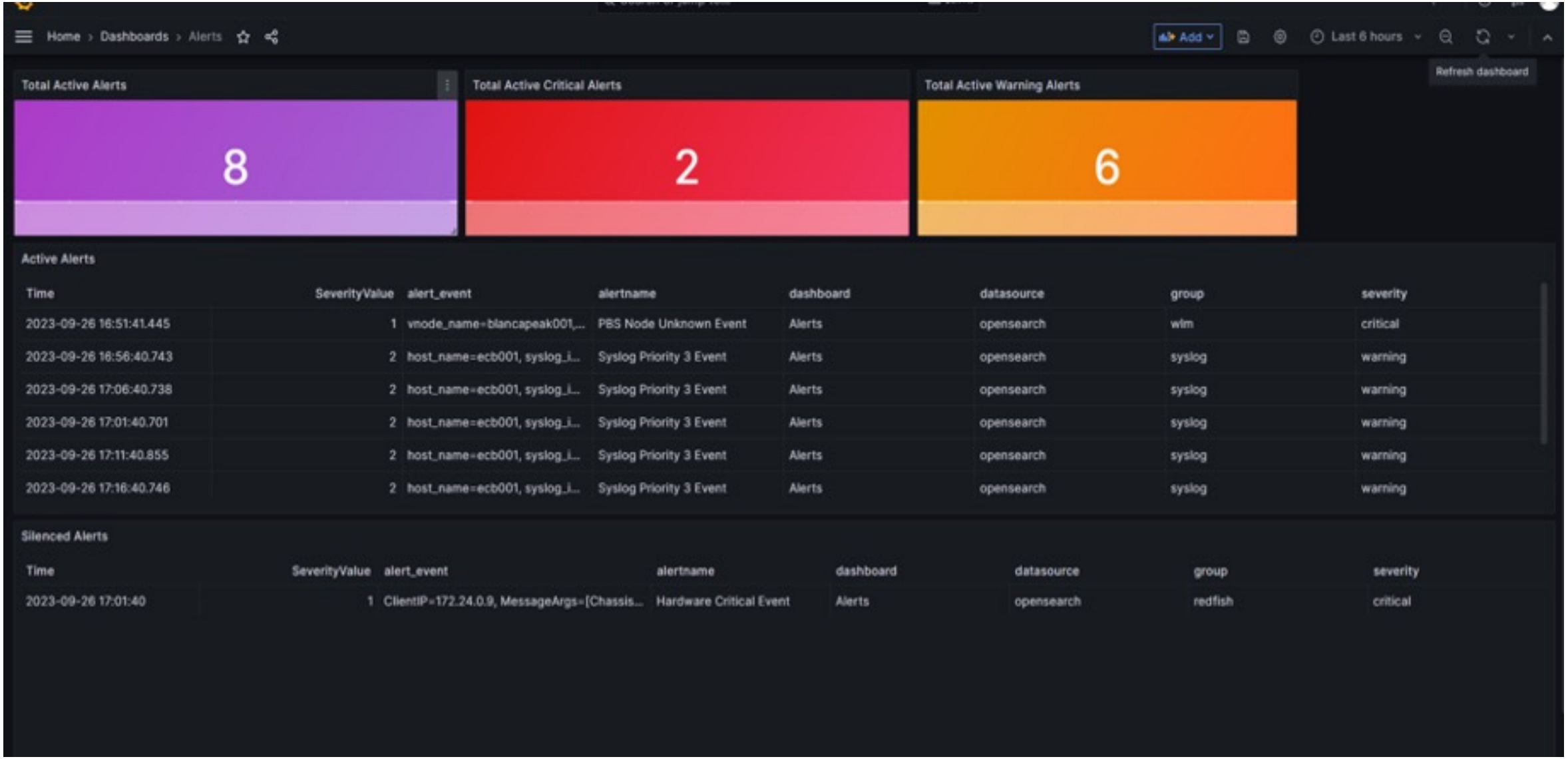
Unified Alerting

```
admin:~ # cm health alertman -g
     cooldev
     iml
     native
     node_status
     rasdaemon
     redfish
     sim
     syslog
     switch
     wlm
     others
admin:~ # cm monitoring alerting route email --from no-one@org.com --to myemail@org.com --smtp
smtp.org.com:25 --alert-group iml
```

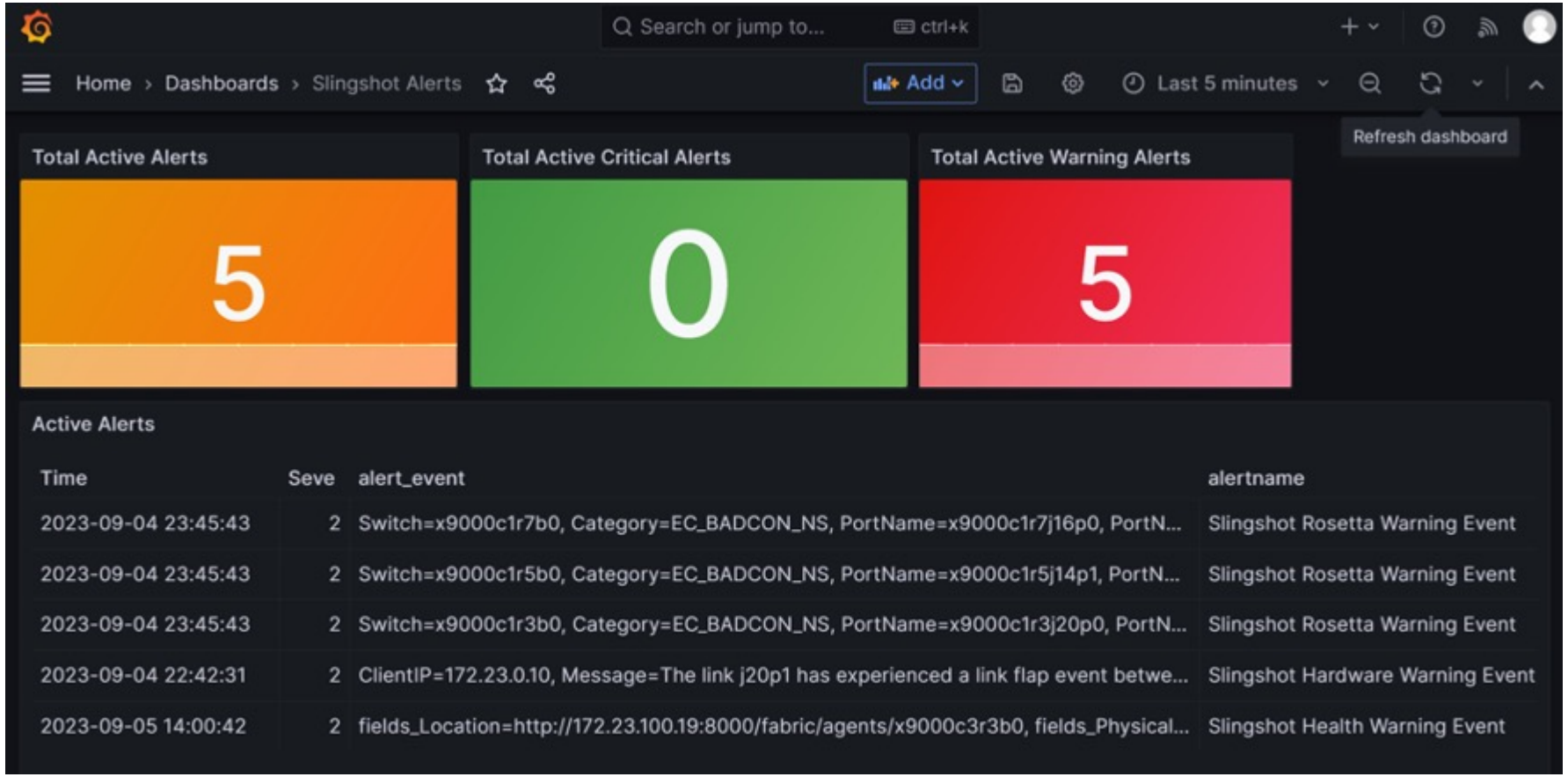
This material does not cover custom rules. See the manual and external documentation.

```
admin:~ # cd /opt/clmgr/alerting/opensearch-alerting/alert_rules
admin:/opt/clmgr/alerting/opensearch-alerting/alert_rules # ls
CDU  hw_events  IML  native_monitoring  node_status  pbs  Rasdaemon  SAMPLE.yml.reference
slingshot  slurm  Syslog
admin:/opt/clmgr/alerting/opensearch-alerting/alert_rules # cd /opt/clmgr/alerting/grafana-
alerting/alert_rules
admin:/opt/clmgr/alerting/grafana-alerting/alert_rules # ls
slingshot
```

Unified Alerting



Unified Alerting



Service infrastructure monitor - SIM

The service infrastructure monitor (SIM) shows information about the health of your infrastructure services.

SIM includes metrics, dashboards, and alerts that help you understand how HPCM core services, monitoring services and su-leader services and resources are operating.

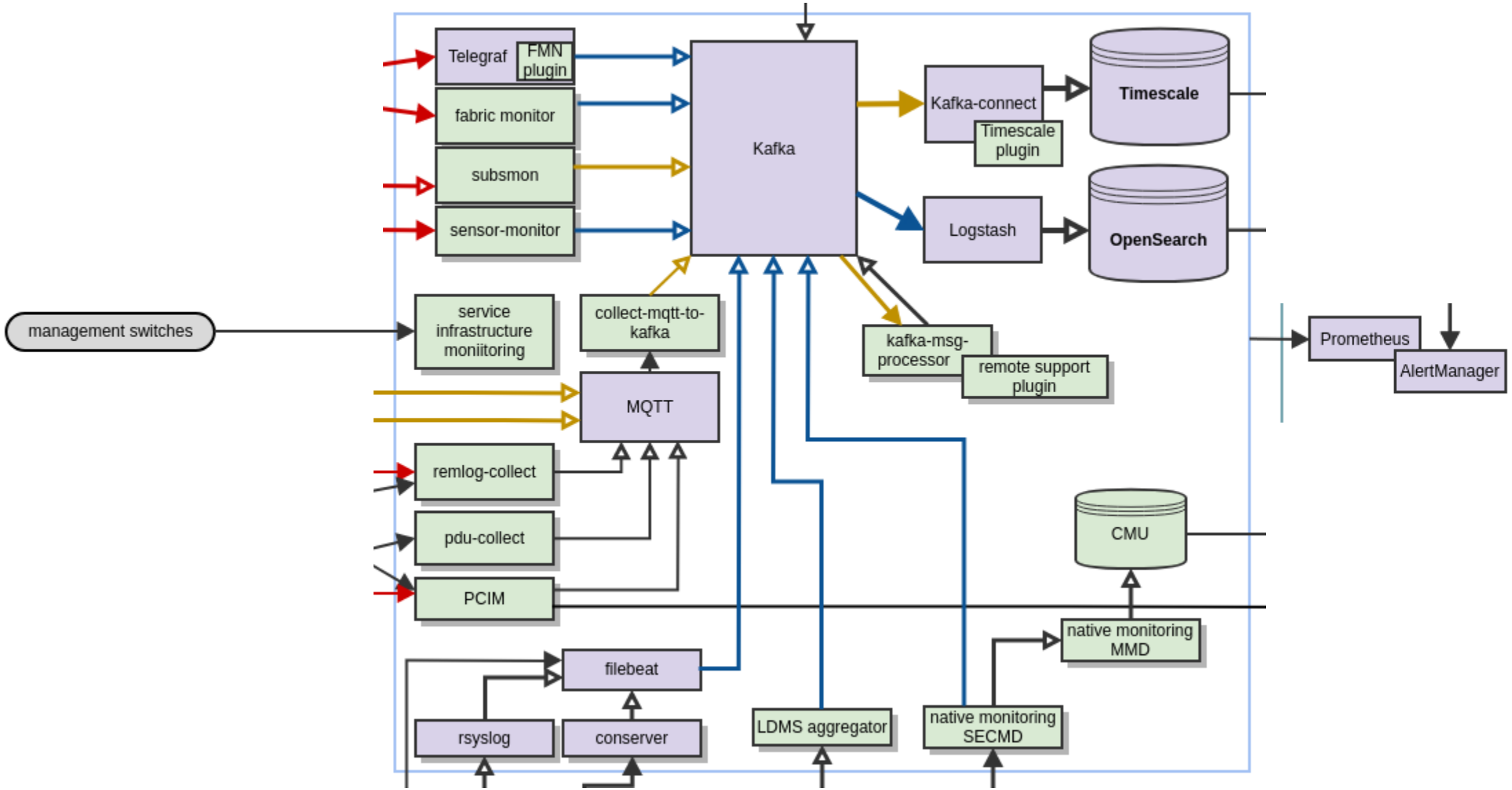
It also supports AIOPs.

SIM displays any alerts that arise including disk space issues if retention/ccompression has not been configure appropriately. If you do have full disks please refer to the article on the portal.

The cluster manager includes tools and commands to help you monitor the services in Grafana.

It is best practice to enable SIM.

Service infrastructure monitor - SIM



Service infrastructure monitor - SIM

```
admin:~ # cm sim enable
Enabling SIM features...
Installing opensearch plugin for expose opensearch metrics..
Configuration manager submitting node configuration.
Populating Dataset...
Populating Dataset complete: 0.471s
4 of 4 nodes completed in 3.0 seconds, averaging 0.0s per node
4 of 4 nodes completed in 3.0 seconds, averaging 0.0s per node
Node configuration complete.
Running enable for prometheus service : prometheus.service

Running enable for alertmanager service : alertmanager.service

Running enable for core-services service: node_exporter.service

Running enable for core-services service: disk_exporter.service

Running enable for core-services service: disk_exporter.timer

Running enable for core-services service: process_exporter.service

Running enable for core-services service: snmp_exporter@1.service

Running restart for elastalert service : elastalert.service

Adding respective dashboards in to grafana
admin:~ # cm sim start
Running start for prometheus service : prometheus.service

Running start for alertmanager service : alertmanager.service

Running start for core-services service: node_exporter.service

Running start for core-services service: disk_exporter.service

Running start for core-services service: disk_exporter.timer

Running start for core-services service: process_exporter.service

Running start for core-services service: snmp_exporter@1.service
```

Service infrastructure monitor - SIM

```
admin:~ # cm sim add --service-group monitoring-services
Configuration manager submitting node configuration.
Populating Dataset...
Populating Dataset complete: 0.439s
0 of 4 nodes completed in 2.9 seconds, averaging 0.0s per node
4 of 4 nodes completed in 5.4 seconds, averaging 0.0s per node
4 of 4 nodes completed in 5.4 seconds, averaging 0.0s per node
Node configuration complete.
Running enable for monitoring-services service: mosquitto_exporter.service

Running enable for monitoring-services service: postgres_exporter.service

Running start for monitoring-services service: mosquitto_exporter.service

Running start for monitoring-services service: postgres_exporter.service

Adding respective dashboard in to grafana
admin:~ # cm sim add --service-group suleader-services
Configuration manager submitting node configuration.
Populating Dataset...
Populating Dataset complete: 0.406s
0 of 4 nodes completed in 2.9 seconds, averaging 0.0s per node
4 of 4 nodes completed in 5.4 seconds, averaging 0.0s per node
4 of 4 nodes completed in 5.4 seconds, averaging 0.0s per node
Node configuration complete.
Running enable for suleader-services service: ctdb_exporter.service

Running enable for suleader-services service: gluster_exporter.service

Running start for suleader-services service: ctdb_exporter.service

Running start for suleader-services service: gluster_exporter.service

Adding respective dashboard in to grafana
```

Service infrastructure monitor - SIM

For Aruba switches with <1.11:

```
admin:~ # grep scrape
/etc/prometheus/prometheus.yml
  scrape_interval: 60s
  scrape_timeout: 30s
scrape_configs:
```

cm monitoring rackmap

Terminal mapping of **component-drift, power, slingshot-switch-status and cpu-temperature**

- -l will prepend a legend
- -s summary
- --no-color
- --interactive
- -b to select a blade type

New in 1.11

```
# cm monitoring rackmap map cpu-temperature -l

Legend

======

0:.o

   Node temperature is within acceptable range of the median temperature of other nodes in its rack.

1:./

   Node temperature deviates from the median temperature of other nodes in its rack.

2:.X

   Node temperature significantly deviates from the median temperature of other nodes in its rack.
```

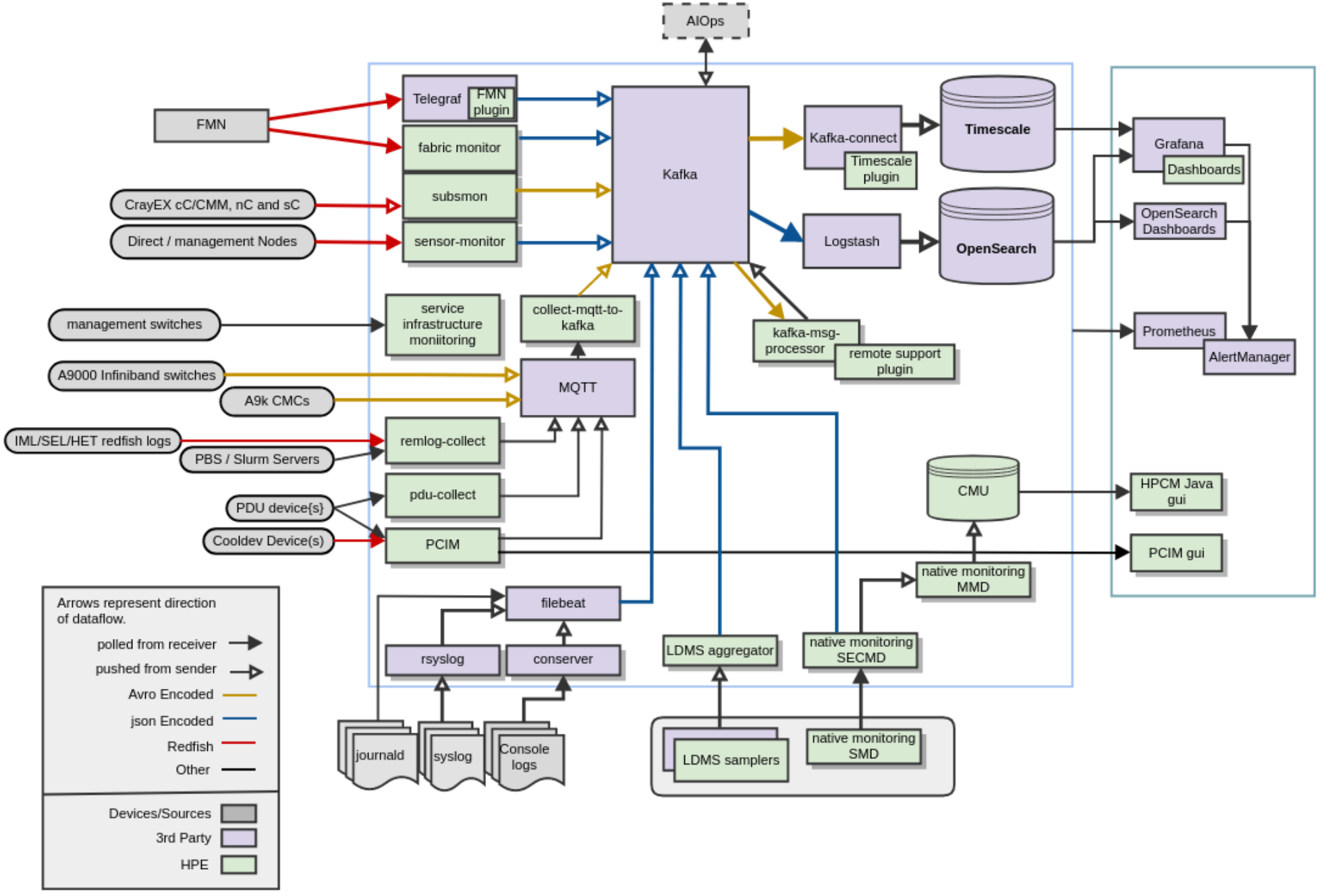
cm monitoring rackmap

```
harding-adm:~ # cm monitoring rackmap map component-drift -l --no-color
Legend
======
OK:...o
    All components match
BIOS:.B
    BIOS versions does not match the most common version in the rack
CPU:..C
    Not all CPU's match
DIMM:.D
    Not all DIMM's match

          x1000
6:7 --------|--------
    --------|--------
    --------|---B----
    B-------|BB-B----
    --------|--------
    --------|--------
    --------|--CBCC--
    B-------|BBCBCC--
4:5 --------|--------
    --------|--------
    --------|---B--B-
    B-------|B--B--B-
    -ooD----|--o--D-B
    -Doo----|--D--D-B
    -DBo----|--oBCDBB
    BooD----|B-oBCDBB
2:3 --------|--------
    --------|--------
    --------|--------
    B-BBBBB-|B-------
    --------|-o------
    --------|-D------
    -C------|-o------
    BCBBBBB-|BD------
0:1 --------|--------
    --------|--------
    DDBB----|--------
    DDBB----|--------
    --------|--------
    --------|--------
    DDBB----|--------
    DDBB----|--------
   s01234567  s01234567
```

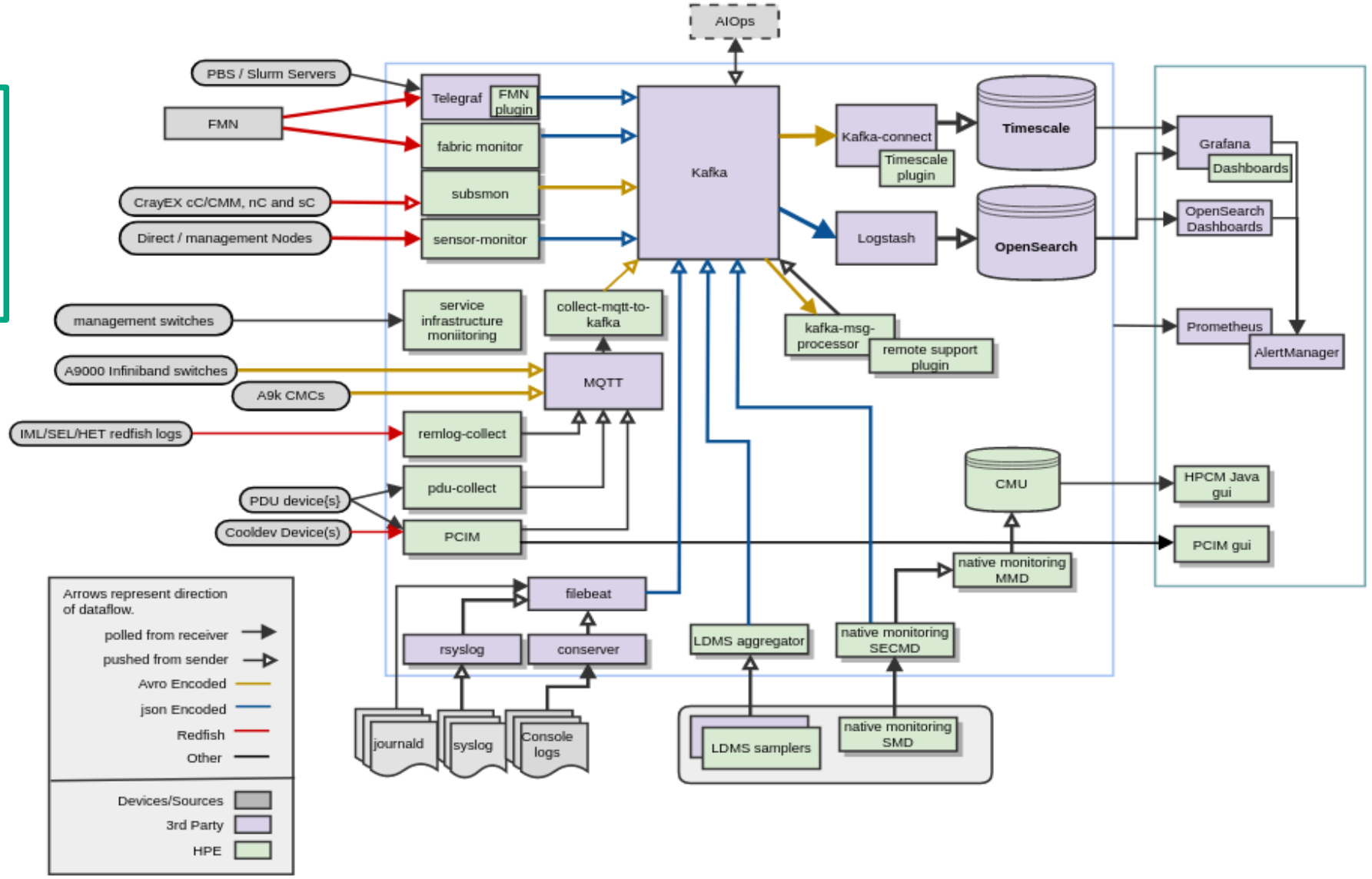
This rack purposely has varied hardware and firmware plus blades not available so it makes a good example

Wrapping up

Monitoring


AIOps
FMN
Telegraf
FMN plugin
fabric monitor
subsmon
sensor-monitor
CrayEX cC/CMM, nC and sC
Direct / management Nodes
Kafka
Kafka-connect
Timescale plugin
Timescale
Logstash
OpenSearch
Grafana
Dashboards
OpenSearch Dashboards
Prometheus
AlertManager
management switches
service infrastructure moniitoring
collect-mqtt-to-kafka
kafka-msg-processor
remote support plugin
A9000 Infiniband switches
A9k CMCs
MQTT
IML/SEL/HET redfish logs
PBS / Slurm Servers
remlog-collect
PDU device{s}
pdu-collect
Cooldev Device(s)
PCIM
CMU
HPCM Java gui
PCIM gui
native monitoring MMD
filebeat
rsyslog
conserver
LDMS aggregator
native monitoring SECMD
journald
syslog
Console logs
LDMS samplers
native monitoring SMD
Arrows represent direction of dataflow.
polled from receiver
pushed from sender
Avro Encoded
json Encoded
Redfish
Other
Devices/Sources
3rd Party
HPE

1.10 with patch 11796



cm support moncollect

1.11 introduces a command for log, status and configuration information pertaining to monitoring to aid support in troubleshooting issues.

```
admin:~ # cm support moncollect
This command takes a long time to run as it observes data flow across different parts of the
monitoring architecture to aid support in troubelshooting.
Create early kafka topic numbers plus opensearch to assess if data is flowing.
Collecting generic linux background information...
Starting with several command outputs including commands which will take a while to run such as
rpm -Va
Collecting monitoring log files...
Collecting monitoring configuration...
Collecting monitoring status...this may take a while.
Querying metrics in tsb. This will take a while...
Checking for network traffic to nginx/telegraf.
Checking if cooldev data is flowing through mosquitto. This will take at least 30s.
Checking if this is an ICE system...
No ICE rack leader nodes detected.
Checking if this system has SU leaders...
Scalable Unit leader nodes detected so collecting data!
Creating tar ball and compressing data. This may take some time.
Monitoring tar ball is available here: /var/tmp/cm-support-monitoring-2023-12-05T0655CST.tar.xz
```

Questions?

Thank you

susan.miller@hpe.com

jeff.hanson@hpe.com

pguyan@hpe.com

andy.warner@hpe.com

raghul-vasudevan@hpe.com (myself)