Hewlett Packard
Enterprise

CSM-based Software Stack Overview 2024

Harold Longley, HPE
Jason Sollom, HPE

May 8, 2024

Agenda

- CSM Software Stack Release 24.03
- Boot Orchestration Service Version 2 (BOS v2) in CSM-1.4
- Configuration Framework Service Improvements in CSM-1.4
- Configuration Framework Service Improvements in CSM-1.5
- Power Control Service (PCS) in CSM-1.4

CSM Software Stack Release 24.03

Key features for CSM Software Stack Release 24.03

- Installation and Upgrade Automation Framework (IUF)
- Installation and Upgrade improvements
- Management nodes
  - SLE 15 SP5 on management nodes
- Compute and application node environment
  - SLE 15 SP5
    - x86_64
    - aarch64
  - AMD ROCm 5.7
  - NVIDIA HPC SDK 23.9
  - Rootless run-time containers on compute nodes

What is different in the software recipe 24.03?

- Software installation workflow with IUF
- New CSM functionality
- New hardware support
- Compute nodes and UANs
  - New features
- SMA
  - New features
- Slingshot
  - New features

Installation and upgrade framework (IUF)

- Decreases the time it takes to install or upgrade products
- Provides a centralized and common installation/upgrade experience across all non-CSM products
  - Each product distribution includes an iuf-product-manifest.yaml file which IUF uses to determine what operations are needed to install, upgrade, and deploy the product
  - IUF groups the install, upgrade, and deploy operations into stages
  - The administrator can execute some or all of the stages with one or multiple products in a single activity
  - IUF arguments for all stages can be specified prior to execution in order to automate the operations and minimize user interaction
- IUF utilizes Argo workflows to execute and parallelize IUF operations
  - iuf CLI invokes Argo workflows based on the subcommand specified
  - Argo UI provides visibility into the status of the operations
  - IUF provides metric and annotation capabilities
    - Can view status and record historical information associated with an install or upgrade
- IUF enhancements in CSM 1.5
  - Argo-driven upgrade automation for utility storage nodes
  - Ceph upgrade added to automated storage upgrade
  - Added support for specifying IMS image and recipe architecture
  - Improved logging
  - IUF stage for management-nodes-rollout consumes logs from ncn-rebuild

IUF Documentation and stages

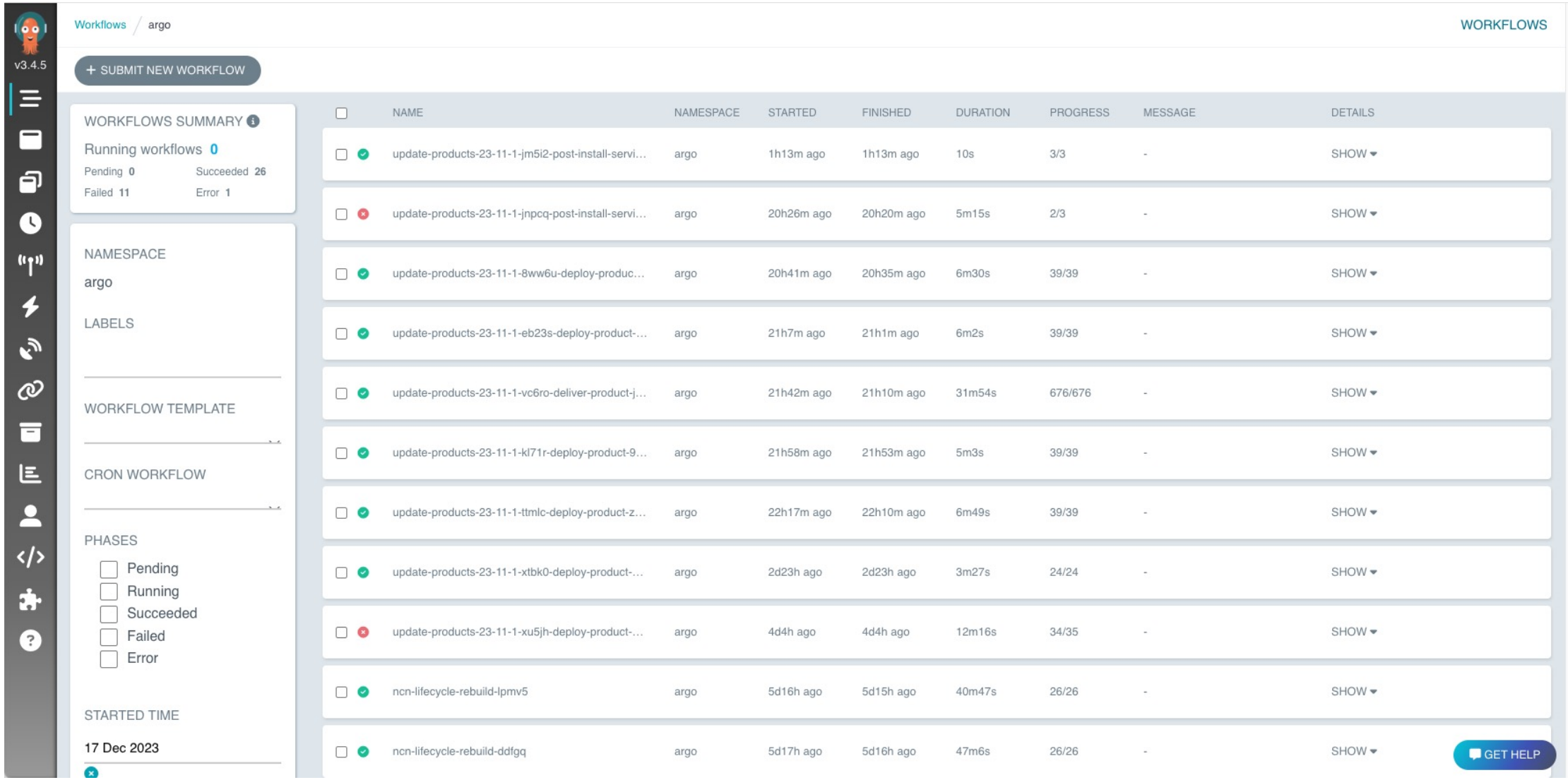
- Majority of install/upgrade procedures are in one IUF location in the CSM documentation
- Each non-CSM product guide has a similar IUF section
- The CSM guide's IUF instructions point the user to each product's IUF section if any manual operations are required

```
ncn-m# iuf list-stages
+---------------------------+-----------------------------------------------------------------------------+
| Stage                     | Description                                                                 |
+---------------------------+-----------------------------------------------------------------------------+
| process-media             | Inventory and extract products in the media directory for use in subsequent stages    |
| pre-install-check         | Perform pre-install readiness checks                                        |
| deliver-product           | Upload product content onto the system                                      |
| update-vcs-config         | Merge working branches and perform automated VCS configuration              |
| update-cfs-config         | Update CFS configuration utilizing sat bootprep                             |
| prepare-images            | Build and configure management node and/or managed node images utilizing sat bootprep |
| management-nodes-rollout  | Rolling rebuild of management nodes                                         |
| deploy-product            | Deploy services to system                                                   |
| post-install-service-check | Perform post-install checks of deployed product services                   |
| managed-nodes-rollout     | Rolling reboot of managed nodes                                             |
| post-install-check        | Perform post-install checks                                                 |
+---------------------------+-----------------------------------------------------------------------------+
```

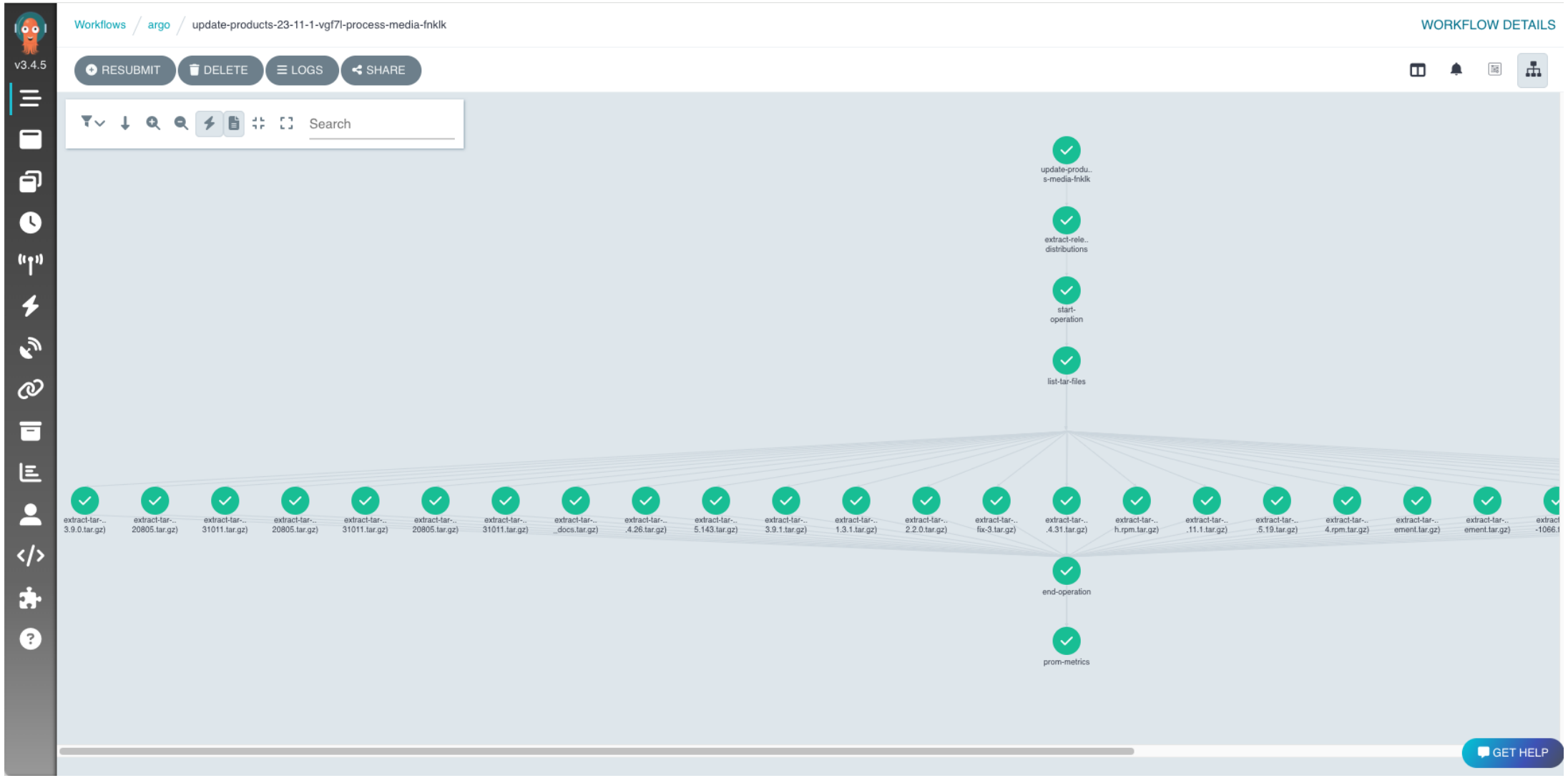
Node lifecycle service (cray-NLS)

- Argo Workflows is an open source container-native workflow engine for orchestrating parallel jobs on Kubernetes
  - https://argoproj.github.io/workflows/
  - Implemented as a Kubernetes CRD (Custom Resource Definition)
  - Easily orchestrate highly parallel jobs on Kubernetes
  - Define workflows where each step in the workflow is a container
- Model multi-step workflows as a sequence of tasks or capture the dependencies between tasks using a graph (Directed Acyclic Graph)
- Argo UI with CSM
  - Requires authentication with Keycloak
  - Useful for watching the progress of an install or upgrade and debugging
- NLS workflows
  - Once a workflow is started, it will proceed through multiple steps in a set order
  - Most steps depend on previous steps and will wait for its dependencies to finish before starting
  - If any step fails, by default, that step will be continuously retried until it succeeds
    - There are two ways to make Argo not continuously retry a failed step
  - Logs in the Argo UI show output from individual stages of a workflow and are useful for debugging

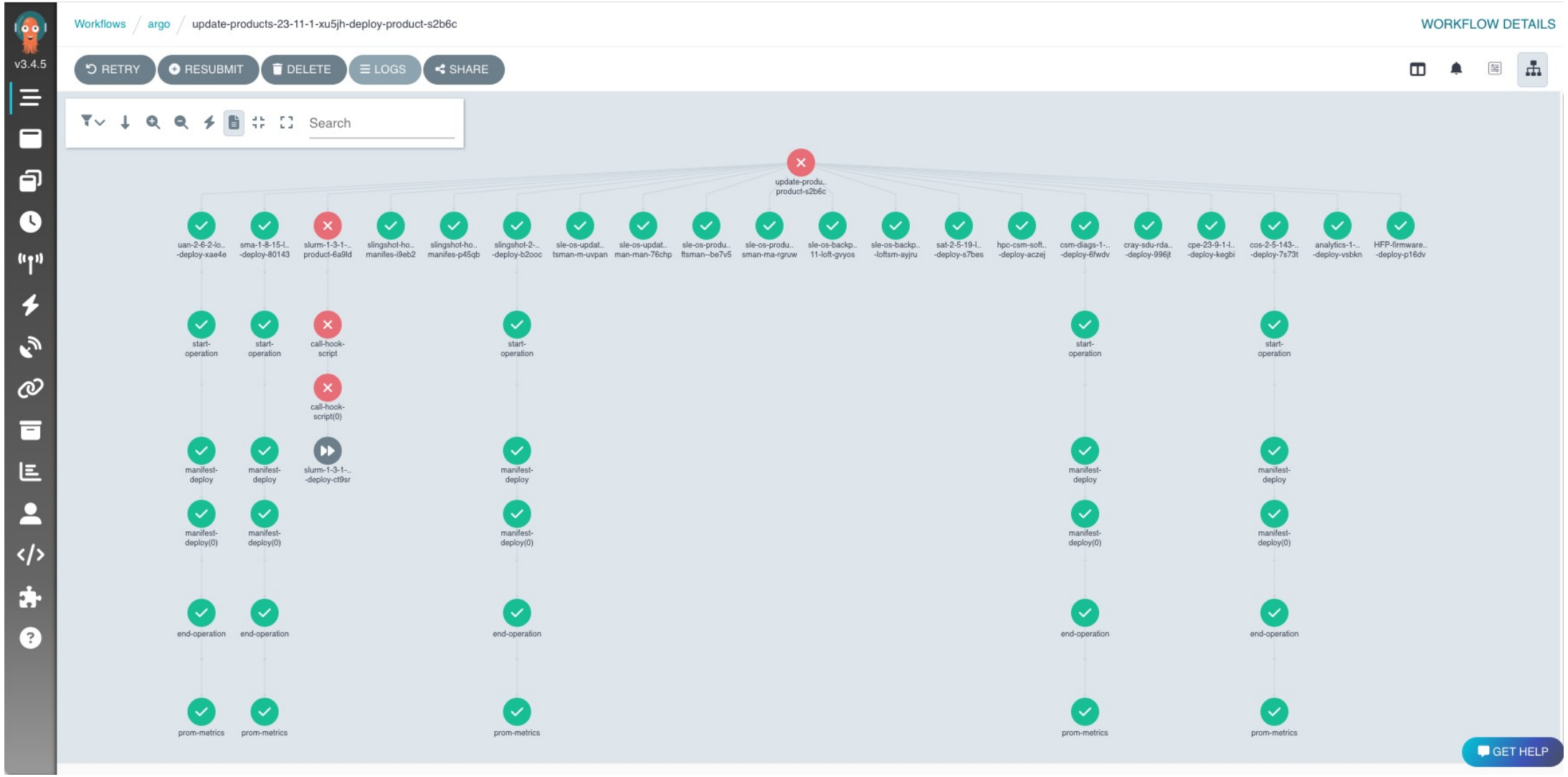
Argo UI – first screen



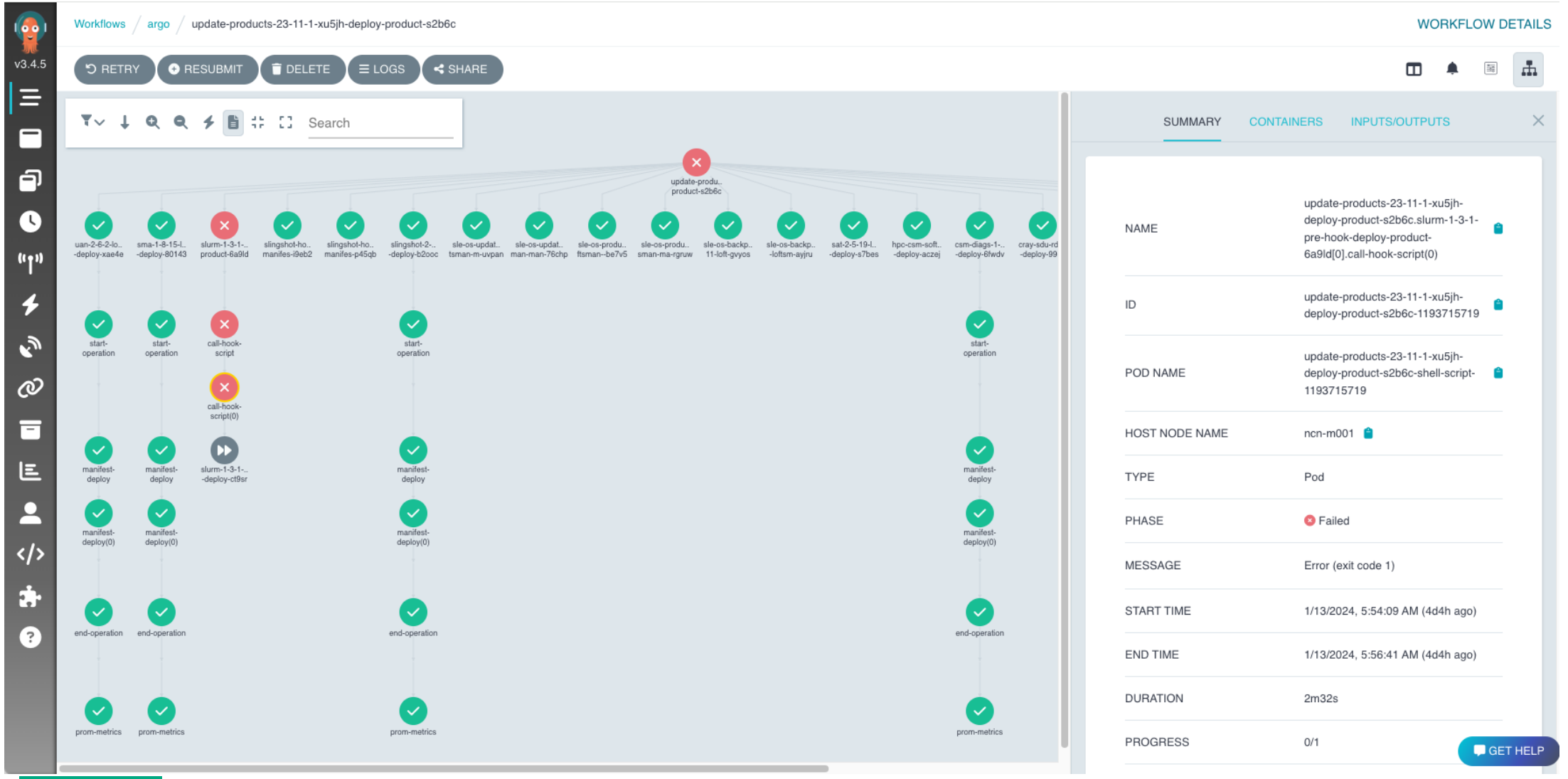
Argo – process media



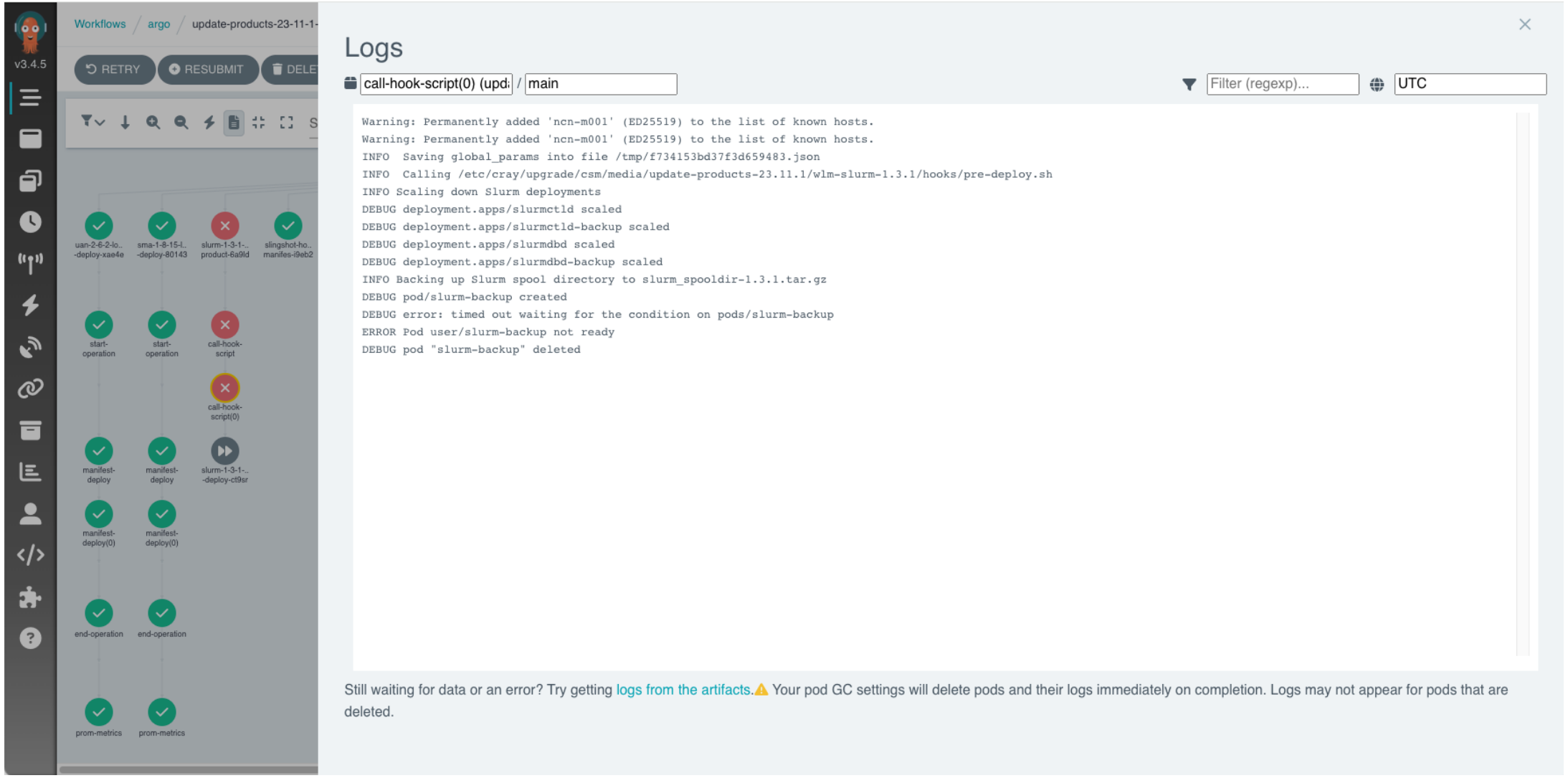
Argo – deploy products (with error)



Argo – deploy products look at red X for Slurm



Argo – deploy products red X for Slurm log

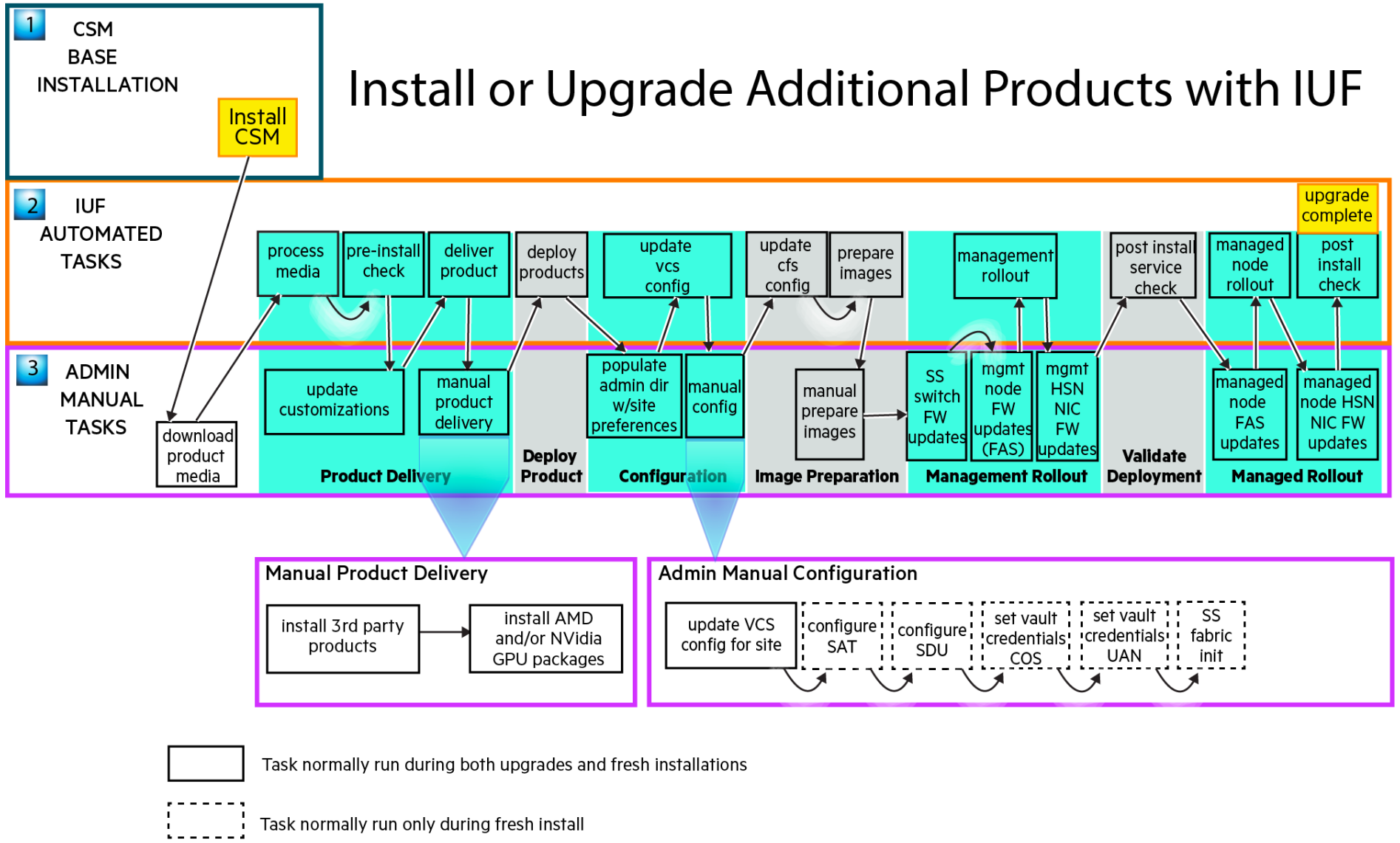


HPE Cray system management software recipe 24.03

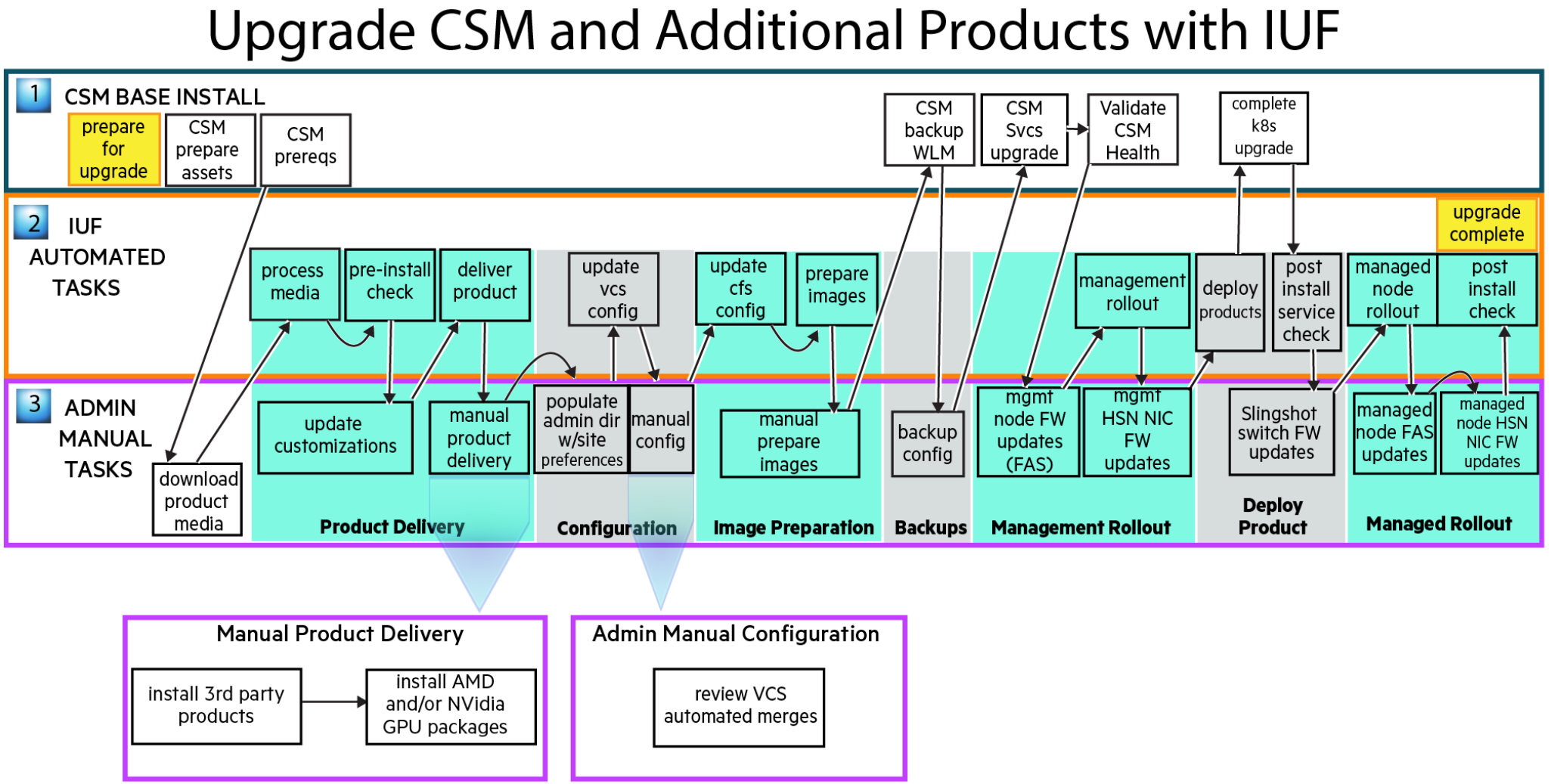
Software Product streams

- Software in recipe 24.03.0
  - HPE HPC CSM Software Recipe 24.03
  - HPE COS Base 3.0.0
  - HPE CPE 23.12.3
  - HPE CSM 1.5.0
  - HPE CSM Diags 1.5.25
  - HPE HFP 23.12.0
  - HPE SAT 2.6.14
  - HPE SDU RDA 2.3.1
  - HPE Slingshot 2.1.1
  - HPE SHS 2.1.2
  - HPE SMA 1.9.11
  - HPE UAN 2.7.1
  - HPE USS 1.0.0 cos-base 3.0
  - HPE USS 1.0.0 CSM 1.5
  - HPE WLM PBS 2.0.3
  - HPE WLM Slurm 2.0.3-23.02.6
  - SLE 15 SP5 23.10.30 (for x86_64 and aarch64)
- Software product streams
  - CPE – Cray Programming Environment
  - COS Base – Cray Operating System Base
  - CSM – Cray System Management
  - CSM Diags – CSM Diagnostics
  - HFP – HPC Firmware Pack
  - HPC CSM Software Recipe
  - SAT – System Admin Toolkit
  - SDU – System Diagnostic Utility
  - SHS – Slingshot Host Software
  - SLE OS – SUSE Linux Enterprise Operating System
  - Slingshot – High speed network fabric management
  - SMA – System Monitoring Application
  - UAN – User Access Nodes
  - USS - User Services Software
  - WLM – Workload Manager (Slurm or PBS Pro)

IUF Installation Workflow



IUF Upgrade Workflow



CSM 1.5 New functionality

- New CFS configuration name for management nodes includes version
  - Replaces generic “ncn-personalization” configuration name
- Networking
  - Created DNAME records in PowerDNS
  - cray-dns-unbound no longer forwards “.hsn” queries to site DNS
  - Created DNS records for all aliases on the NMN
  - IPv6 DNS PTR records for UAIs and UANs
  - Enabled bonded NMN connections for the UANs
  - Bonded HSN Interfaces to apply correct configurations to SHS
- Spire
  - Support for old and new Spire versions running simultaneously to reduce downtime during upgrades
  - Updated to work with TPM
- Support for large system ARP configuration for first boot and DHCP
  - CSM Ansible role csm.ncn.sysctl
- SLS: Added caching to improve performance and robustness
- Transitioned from COS-provided cray-heartbeat to CSM-provided csm-node-heartbeat

NID allocation defragmentation

- Problem:
  - NIDs are not correctly allocated when Cray Site Init (CSI) generates System Layout Service SLS input file for liquid-cooled Antero nodes
    - CSI assumes all blades in the cabinet are Windom compute blades
  - Both Antero and Windom blades both have 4 nodes but they have different physical layouts
    - Windom blades have 2 node BMCs with 2 nodes per node BMC with the following nodes: b0n0, b0n1, b1n0, b1n1
    - Antero blades have 1 Node BMC with 4 nodes per node BMC with the following nodes: b0n0, b0n1, b0n2, b0n3
  - SLS has NIDS only allocated for nodes b0n0, b0n1, b1n0, b1n1 on a compute node blade
    - On an Antero blade the nodes b0n2, b0n3 will have automatically assigned NIDs that are not contiguous with the NIDs on nodes b0n0 and b0n1
      - Nodes b0n2 and b0n3 on an Antero blade are functional, but do not have NIDs in contiguous range with its peers
- Added documentation for NID allocation defragmentation
  - Different blade types (which have different 'density' (number of BMCs or number of nodes) cause spurious or missing NIDS to manifest in the system
  - Do procedure only if absolutely required – must be performed while system is down
    - Compute nodes were added to SLS with incorrect NID numbering, missing node entries, and/or extra node entries
    - Compute nodes were permanently moved, removed, or re-provisioned and there is a desire to remove NID numbering gaps

Technology previews

- In CSM 1.5:
  - Support for Spire TPM-based remote node attestation
  - Disaster recovery support
    - Backup important configuration and use it during reinstall of the same software versions
- In CSM 1.4:
  - Support for parallel Kubernetes worker node upgrades

New Hardware support

- aarch64 architecture
  - Hardware discovery process (hms-discovery) populates a node's architecture in Hardware State Manager (HSM)
  - Image management (IMS) support for cross-architecture building of images through emulation
  - aarch64 version of the barebones IMS recipe
  - CFS added support for Ansible tasks to target specific architecture (`x86_64` or `aarch64`)
    - `IMS_ARCH` is defined when IMS is building images
    - Booted nodes have `ansible_architecture` defined as an Ansible fact

```
- name: set target_arch variable
  set_fact:
        target_arch: "{{ ansible_env.IMS_ARCH |default(ansible_architecture) }}"
```

- Compute nodes
  - Olympus Hardware
    - HPE Cray EX254n NVIDIA Grace Hopper
- Added management network switches to HSM so PCS can perform power reset actions
- Hardware validation of the EX2500 cabinet

CSM monitoring

CSM 1.5

- HA Prometheus setup with Thanos for long term storage
  - Thanos dashboards
- CANU tests red light/green light dashboard
- IUF timing dashboard
  - Report timing data for each stage in the install/upgrade of software products
- Networking statistics from incoming and outgoing TCP/IP traffic
  - SNMP Stats
  - SNMP Interface Detail
- SMARTMON (local storage) dashboard
  - Self-Monitoring, Analysis and Reporting Technology (S.M.A.R.T.)

CSM 1.4

- Implemented pod monitors to scrape SMA Kafka server and zookeeper Prometheus metrics
- Kyverno dashboards
- Monitor Kyverno policy metrics with Prometheus
- SMA Kafka and zookeeper dashboards to monitor their internals
- OpenSearch cluster monitoring dashboard using Prometheus metrics
- Created Prometheus Alerts for CPU and Memory usage for NCNs

CSM security

CSM 1.5

- Ongoing CVE remediation for NCN management OS and several container images
- Addition of kyverno and network policies to ensure some secure controls over MQTT namespace (used by DVS)
- Developed OPA policy to force Keycloak admin operations through CMN
- Moved istio-ingresgateway-cmn service to use the customer-admin-gateway
- Added default RBAC Role for Telemetry API
- TPM-based attestation in Spire

CSM 1.4

- IPXE binary name randomization for added security
- Created read-only `tapms` API for getting tenant status
- Added OPA Rules for TPM workloads
- Protected S3 NCN images
- Moved from OPA-gatekeeper to Kyverno for enforcement of security policies
- Kubernetes security policy is running in audit mode via Kyverno

CSM multi-tenancy

CSM 1.5

- Vault transit (KMS) support for encrypted secrets in VCS
- Enabled tenant ID and tenant admin AuthZ awareness for API ingress (OPA policy)
- Enabled tenant ID and tenant admin AuthZ awareness for API ingress
- Automated power-off-on-entry and power-off-on-exit for managed node resource groups
- Support for user access nodes in tenant resource groups
- Tenant administrator support for managed node resources
  - BOS support for boot, reboot, node power on and off in tenant
  - BOS implemented OPA policies for Multi-Tenancy

CSM 1.4

- Soft multi-tenancy – cooperative tenant
- Tenant and Partition Management Service (TAPMS) dual API (tenant status)
- Slurm operator tenant integration

CSM Deprecated features

Deprecated in CSM 1.6

- BOSv1 removed
- CAPMCv3 features removed

Deprecated in CSM 1.5

- CRUS (Compute Rolling Upgrade Service) removed
- BOS v1 v1 session template and boot set fields are no longer stored in BOS
  - When upgrading to CSM 1.5, these fields will automatically be removed from all BOS session templates that contain them
  - When creating BOS v1 session templates, these fields are automatically removed

Deprecated in CSM 1.4

- Cray CLI default to BOSv2 when no version explicitly specified in BOS commands
- CAPMC announcement of deprecation and being replaced by PCS
- Removed SLS support for downloading and uploading credentials in the dumpstate and loadstate REST APIs

Compute node New Features

COS 23.11

- First release with COS Base 3.0 and User Services Software (USS) 1.0
  - /opt/cray/etc/release/cos renamed to /opt/cray/etc/release/uss
- SLE 15 SP5 on compute nodes
- COS Base 3.0
  - COS kernel default configuration (defconfig) more similar to SLE 15 SP5 defconfig
    - COS kernel size has increased noticeably
  - `cray-crash` can analyze crash dumps from the COS Base kernel which supports larger huge page sizes
  - Kernel memory compaction support interfaces documented
    - Precompaction /proc filesystem
      - `/proc/sys/vm/precompaction/purge_vmap_area`
      - `/proc/sys/vm/precompaction/kmem_cache_shrink`
      - `/proc/sys/vm/precompaction/radix_tree_cleaning`
      - `/proc/sys/vm/precompaction/force_pagedrain`
    - Premap vmalloc allocations with kernel cmdline options
      - Problem: `vmalloc()` created pagetables are never freed causing fragmentation with CONFIG_VMAP_STACKS enabled
      - `vmap_premap_enable`: set to false to disable the whole thing
      - `vmap_premap_tasks_per_cpu`: account for this many tasks per cpu
      - `vmap_premap_adj_pages`: adjust calculated pages by this much
      - `vmap_premap_num_pages`: use this instead of calculated pages if non-zero

User Services Software (USS) New Features

- USS 1.0
  - AMD ROCm 5.7
  - NVIDIA SDK 23.9 and driver 535.129.03
  - Cray ClusterStor Lustre client 2.15.1.x
    - Lustre community 2.15.1 LTS release
  - Dynamic Kernel Module Support (DKMS) for GPU support during image creation
  - Low Noise Mode (LNM) default configuration excludes migrating the Lustre and Spectrum Scale related processes to the system CPUs
  - `/usr/sbin/lnmctl –validate` checks LNM config files for syntax errors
  - Containers on compute nodes (COCN)
    - Documentation to build and run containerized applications using Cray MPI
    - Podman, SingularityCE, or Kubernetes
  - Message Queuing Telemetry Transport (MQTT) for DVS node health monitoring uses Artemis MQTT broker
    - Alternate choice for DVS node health messaging
    - Lightweight and needs less amount of data transfer or storage
  - CPS broker uses `cos-agent` running on workers
    - Takes requests from CPS to maintain CPS contents data, DVS server list, and DVS exports
    - CPS will not deploy cray-cps-cm-pm and cray-cps-etcd pods
  - `gpu-nexus-tool`
    - Allows additional GPU content versions
    - Supporting multiple architectures for Nvidia content

COS 2.5 (previous release)

- DVS has experimental support for monitoring the networks that DVS is using and automatically doing fail-over when those networks have problems
- SLES 15 SP4 based kernels introduce support for idmapped mounts
  - DVS does not support idmapped mounts
- CPS now runs cray-cps-cm-pm pods on all worker nodes by default
  - CPS provides a static DVS server list for mounting DVS and runs CPS PM on all worker nodes to ensure DVS contents are exported consistently

UAN Changes

UAN 2.7.1

- K3s may be optionally deployed to UANs using the playbook k3s.yaml
  - UAIs on UANs
- A new Nexus raw repo provides 3rd party packages to deploy k3s and related services
- UAN rpms have been removed and replaced with Ansible roles

UAN 2.6.0

- UAN CFS configurations now require a CSM and two COS layers
  - Roles that were duplicated from COS CFS in the UAN CFS repo have been removed
    - Values for COS CFS roles that were previously set in the UAN CFS `group_vars` directory should now be set in COS CFS group_vars
- UAN CFS has been restructured to work for COS and Standard SLES images
- `uan_packages` variables are now `vars/uan_packages.yml` and `vars/uan_repos.yml` and have been renamed
  - Admins will need to migrate to the new settings
- The NMN connection now supports bonding (optional)
  - The default is a non-bonded single interface

SMA New features

SMA 1.9

- SMA service API endpoints support Customer High Speed Network (CHN)
- Enhanced Power monitoring in SMF using Power and Cooling Infrastructure Management (PCIM)
  - CDU Monitoring Web UI provides CDU alert history
- Slingshot fabric health reporting
- View alerts with “cm health alert” CLI command

SMA 1.8

- Application nodes, such as UAN and Gateway nodes, can now be configured to report LDMS metric data to SMA
- Alerts based on Cray EX hardware event Telemetry
- Alerts based on Slingshot fabric and hardware event Telemetry
- Telemetry-filter service
- Slingshot 2.0 support
- Moved from Elasticsearch to Opensearch (due to licensing change)

cm health alert

```
ncn# cm health alert -s
Alert Status        Count
------------        -----
Critical            2
Warnings            0
Information         0

Open                2
Acknowledged        0
Closed              0
Expired             0

Group                Severity        Alerts
------               --------        ------
compute              critical        critical : 2, warning : 0, info : 0
fabric               ok              critical : 0, warning : 0, info : 0
slingshothsn         ok              critical : 0, warning : 0, info : 0
slingshotswitch      ok              critical : 0, warning : 0, info : 0
prometheus           ok              critical : 0, warning : 0, info : 0
aiops                ok              critical : 0, warning : 0, info : 0
ncn# cm health alert query
ID        STATUS    SEVERITY    GROUP    ENV           SERVICE      RESOURCE             EVENT          VALUE  DESCRIPTION                                    DUPL  LAST RECEIVED
--------  --------  ----------  -------  ------------  -----------  -------------------  -----------  -------  ---------------------------------------------  ----  -------------------
2809d804  open      critical    compute  x3700c0r41b0  SensorEvent  dmtf.redfish_event   PSU1-Voltage         0  Sensor _PSU1 Voltage_ reading of 0 _V_ is  0     2024/03/05 19:13:21
                                                                                                                 below the 11.16 lower critical threshold.
85082bef  open      critical    compute  x3700c0r39b0  SensorEvent  dmtf.redfish_event   PSU1-Voltage         0  Sensor _PSU1 Voltage_ reading of 0 _V_ is  0.    2024/03/05 19:17:35
                                                                                                                 below the 11.16 lower critical threshold.
```

- Manage alerts from many sources: Prometheus Alertmanager, Monasca, Slingshot
  - Looks for events in the data
  - Constantly analyzes each event
  - Alerts the user regarding the event
  - Stores the event in the alert dashboard
- Manage the life cycle of alerts
  - Retrieve alerts
  - Process alerts
  - Close alerts
  - Disable during maintenance periods and re-enable after maintenance ends

SMA New Grafana dashboards

SMA 1.9

- See Monitoring Cooling Devices with Artificial Intelligence for IT operations
  - AIOps Anomaly Forecast
  - AIOps Slingshot Physical Context Congestion
  - AIOps Slingshot Physical Context Congestion Details
  - AIOps Slingshot Physical Context Temperature Details
  - AIOps Univariate Dashboard

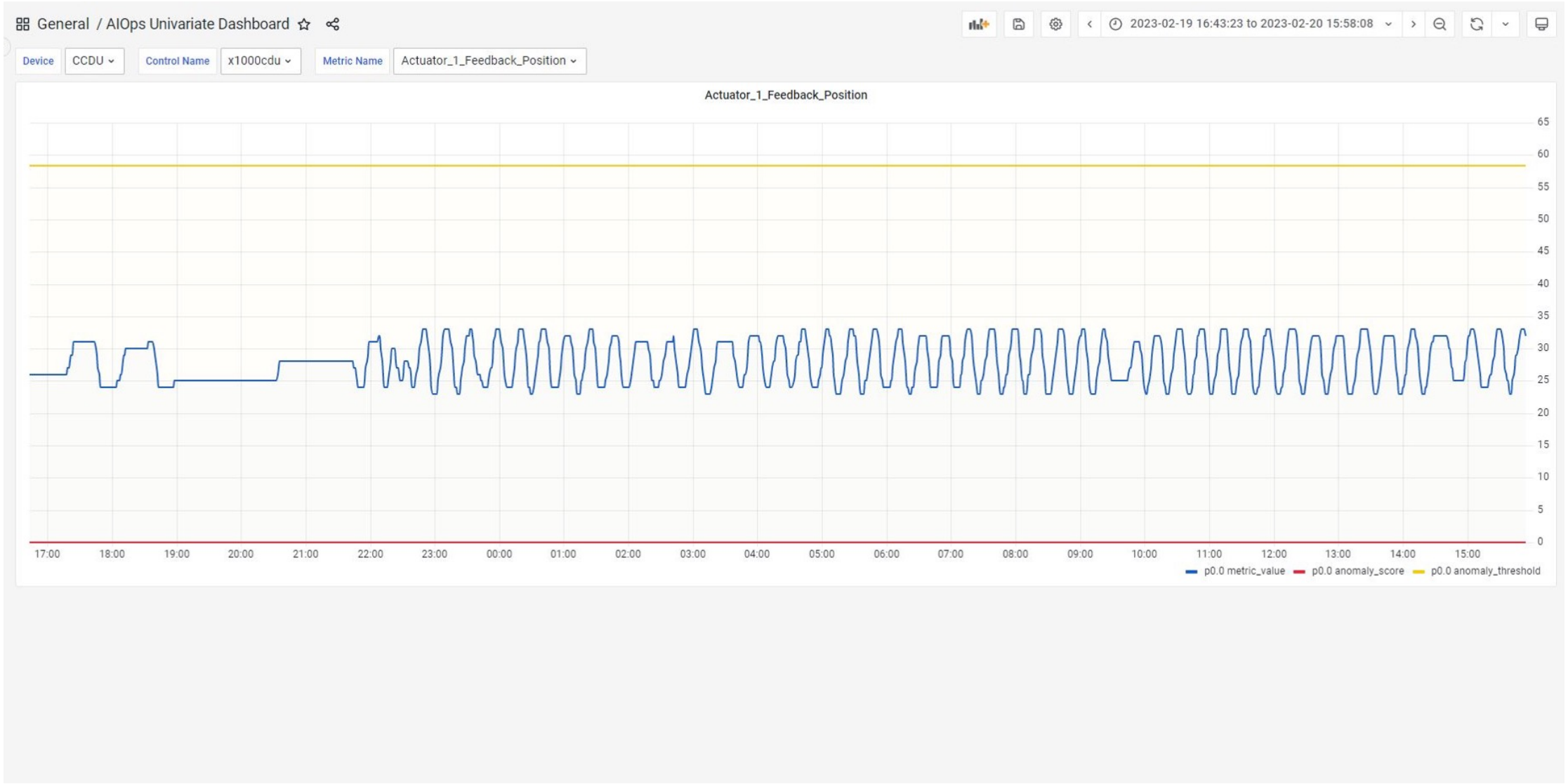
SMA 1.8

- CDU Monitoring
- CPU Power Monitoring
- CPU Temperature Monitoring
- GPU Temperature Monitoring
- Prometheus Alerts
- Slingshot Port Flap
- Slingshot Port State
- Slingshot Bit Error Rate (BER)
- Slingshot rxCongestion
- Slingshot rxBW/txBW (Receive/Transmit Bandwidth)
- Slingshot Routing Error
- Slingshot Hard Error

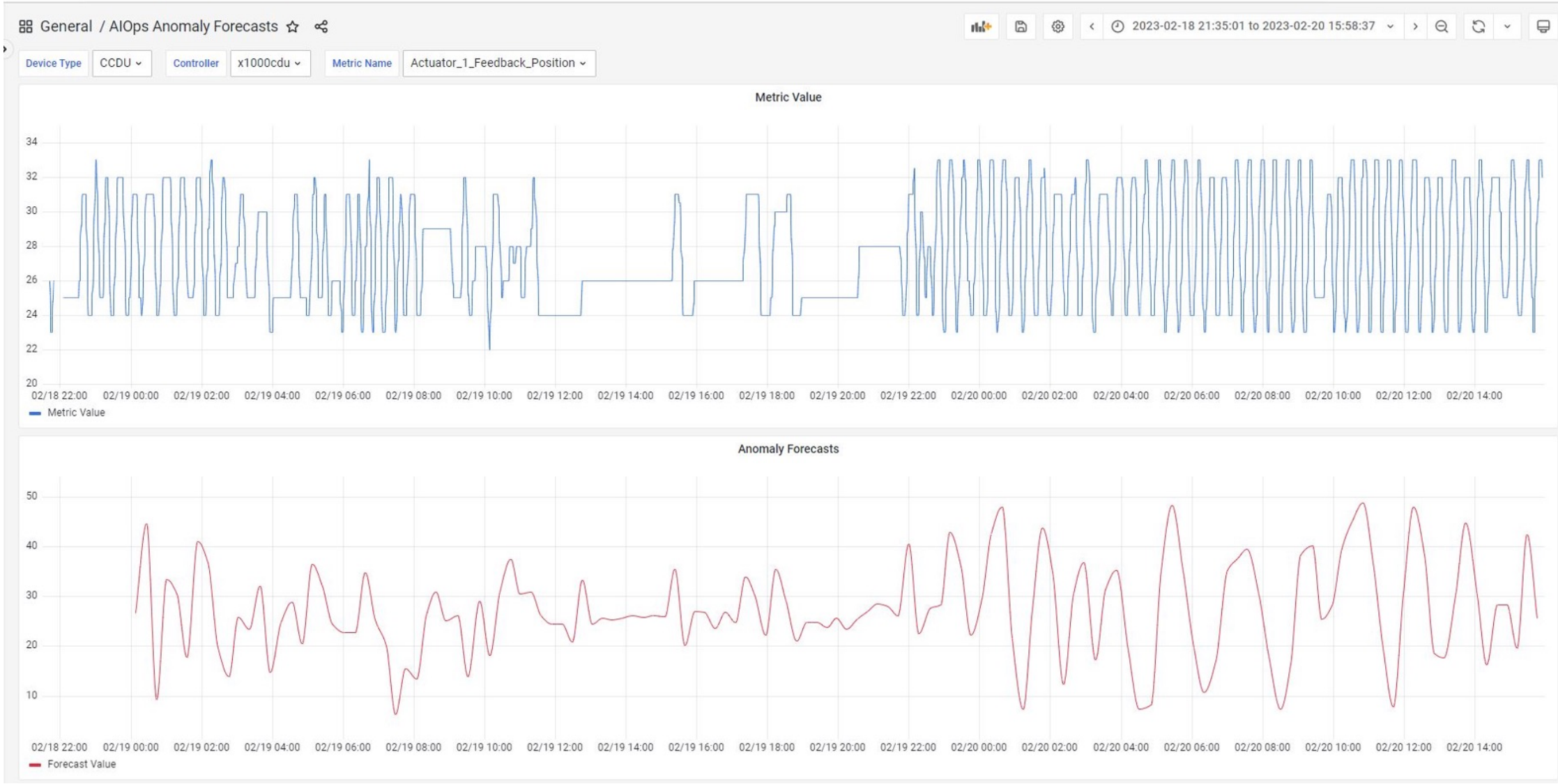
AIOps

- Typical monitoring systems are based on thresholds
  - IT operations require administrators to monitor dashboards
  - The dashboards consolidate data from multiple monitoring systems based on established thresholds
- AIOps offers the following features:
  - Anomaly detection and processing
    - AIOps issues notifications for critical anomalies detected in the metrics derived from the cooling distribution units (CDUs)
    - AIOps simplifies data center management by reducing the number of false alarms, surfacing only anomalous results, limiting the number of dashboards needed, and providing other features
  - Default cooling device monitoring
    - Rather than rely on established thresholds, the default AIOps cooling device monitor uses dynamic thresholds for monitoring cooling devices
    - These dynamic thresholds are calculated automatically and are based on the latest data used to train the AI models
    - The data from the cooling systems can change over time for a number of reasons, and this approach makes alerting relevant to the latest data
  - Alert processing
    - You can display AIOps data in Grafana
    - Within Grafana, AIOps provides several dashboards in JSON format

AIOps univariate dashboard



AIOps anomaly forecast dashboard

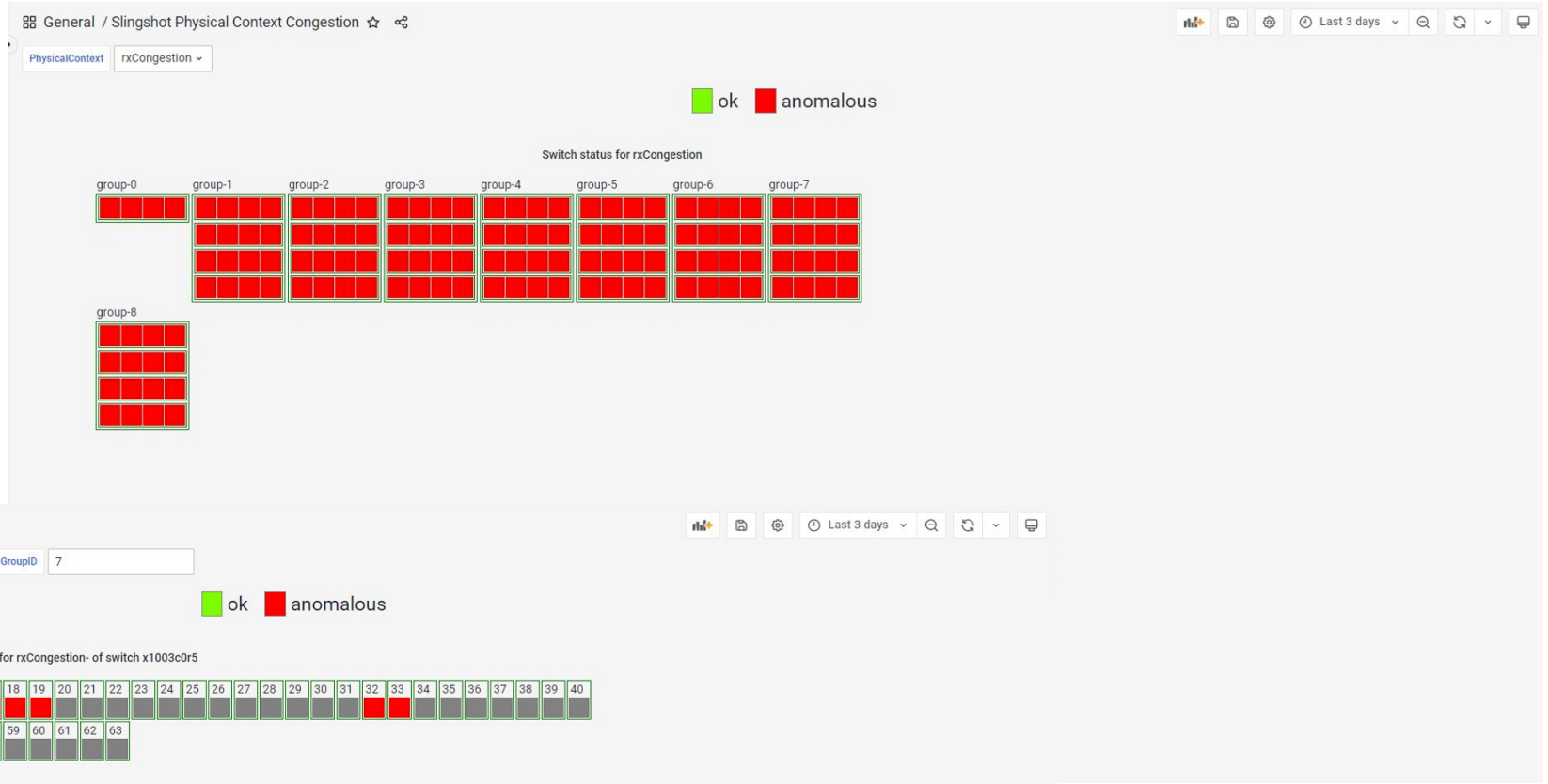


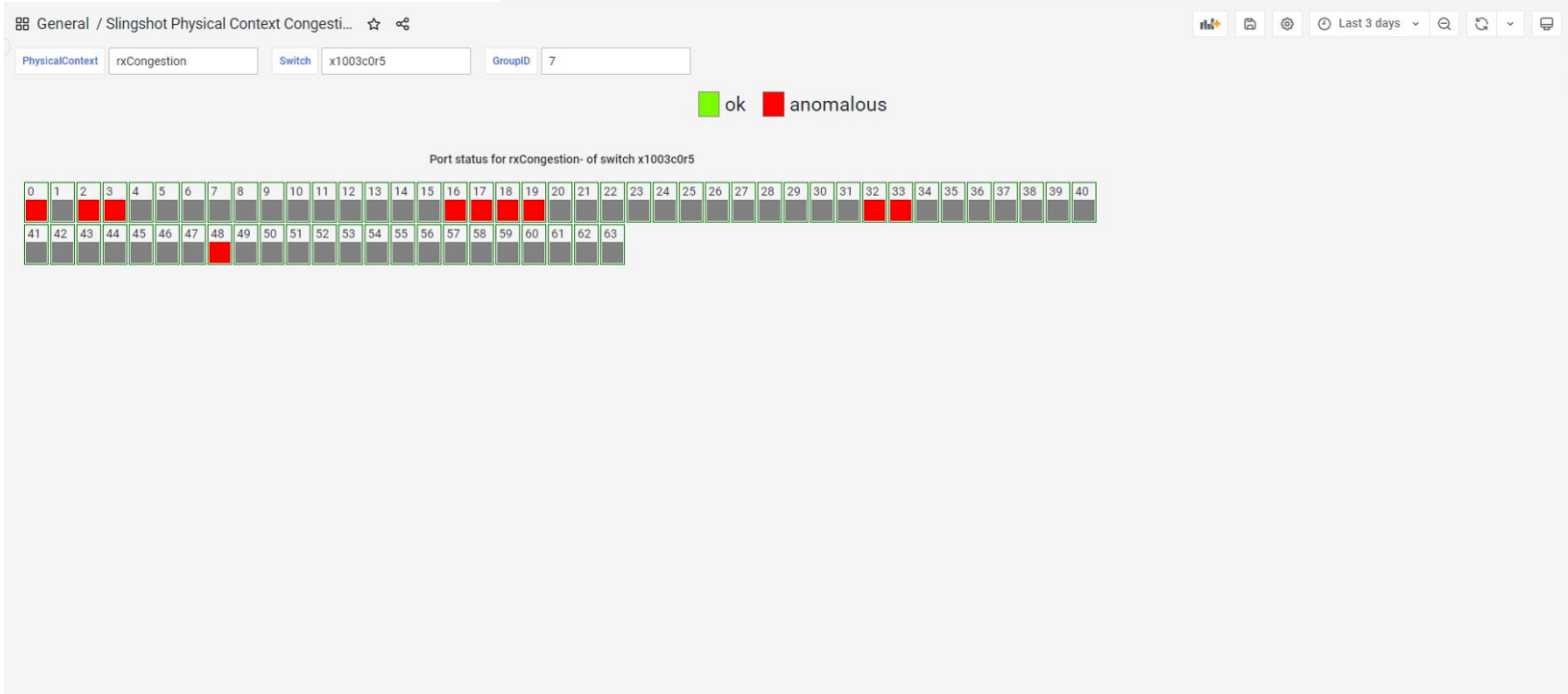
AIOps anomaly detection slingshot temperature

- Metrics used for cray-telemetry-temperature
  - ASIC
  - NetworkingDevice
  - SystemBoard
  - VoltageRegulator
  - Chassis
  - PowerSupply

AIOps anomaly detection slingshot congestion

- Metrics for cray-fabric-perf-telemetry
  - rxPausePercent
  - txPausePercent
  - rxCongestion





Slingshot New features 2.1.1/SHS 2.1.2

- Auto Lane Degrade w/o recovery
  - Keep the link up in degraded state instead of taking the link down
    - Useful with long-running customer jobs which cannot tolerate a link going down
- Dynamic Kernel Module Support (DKMS)
  - Enables kernel module recompilation through DKMS on install or kernel update
- Ability to run CSM Fabric Manager as non privileged `slingshot` user
- Slingshot Orchestrated Maintenance - Phase 1
  - Added Switch Policy Enforcement and Link Policy Enforcement
- LACP – Completion
  - Supports industry standard link-resiliency and traffic switchover as per LACP short timeout
  - Detects abrupt member port speed changes and acts within seconds
  - Supports interoperability with third-party switches
- Many Slingshot 1.0.0 and 2.0.0 commands are now decommissioned in Slingshot 2.1.0
  - If customer has automation scripts or documents based on Slingshot 1.0.0 or 2.0.0 command syntax
    - See Appendix B in HPE Slingshot Release Notes
- Troubleshooting documentation changed to disapprove previous Slingshot switch fabric-agent-host restart method
  - Place switch into maintenance mode to avoid serious fabric issues

Boot Orchestration Service Version 2 (BOS v2) in CSM-1.4

BOS V2 is a new version with a new endpoint, `v2`

- URLs are versioned
- V1
- https://api-gw-service-nmn.local/apis/bos/v1/session/f91e2774-a47c-4d4f-9ba0-2ca04343a8eb
- V2
- https://api-gw-service-nmn.local/apis/bos/v2/sessions/a7a5786c-6361-4659-84a3-72398537b893
- Access different versions of BOS with different versions of the URLs
- The Cray CLI is also versioned.
  - cray bos v1 <command>
  - cray bos v2 <command>
  - CLI will default to v2

BOS V1 and BOS v2 Coexist

- BOS V1 will continue to operate until CSM-1.6
- Customers should choose to use either BOS V1 or BOS V2
  - They should NOT use both concurrently
  - BOS V1 session templates are not interoperable under BOS V2
    - However, V1 templates will be migrated to V2 templates (once)

Main Theme: Nodes Proceed at their own Pace

- In *both* BOS V1 *and* V2, BOS acts on nodes in a BOS session concurrently
  - In BOS V1, every node in a session proceeded *in lock step*
    - Slowest node slows the entire session down
    - Nodes that experience an error are dropped and not retried
  - In BOS V2, every node proceeds *independently*, at its own pace
    - More nodes reach an operable state faster
    - If a node encounters a problem, BOS V2 retries the operation for *just that node*

New Endpoints

- Components
- Options

BOS Sessions

- BOS V2 monitors the nodes in a session
- A session completes when all nodes in it are ‘done’
  - The nodes’ actual states reached their desired states
  - The nodes hit the allowed number of retries (due to errors/failures)

Status

- Session Status
  - Values
    - Pending
    - Running
    - Complete – All of the nodes have finished and/or failed
- Component Status

Session Status

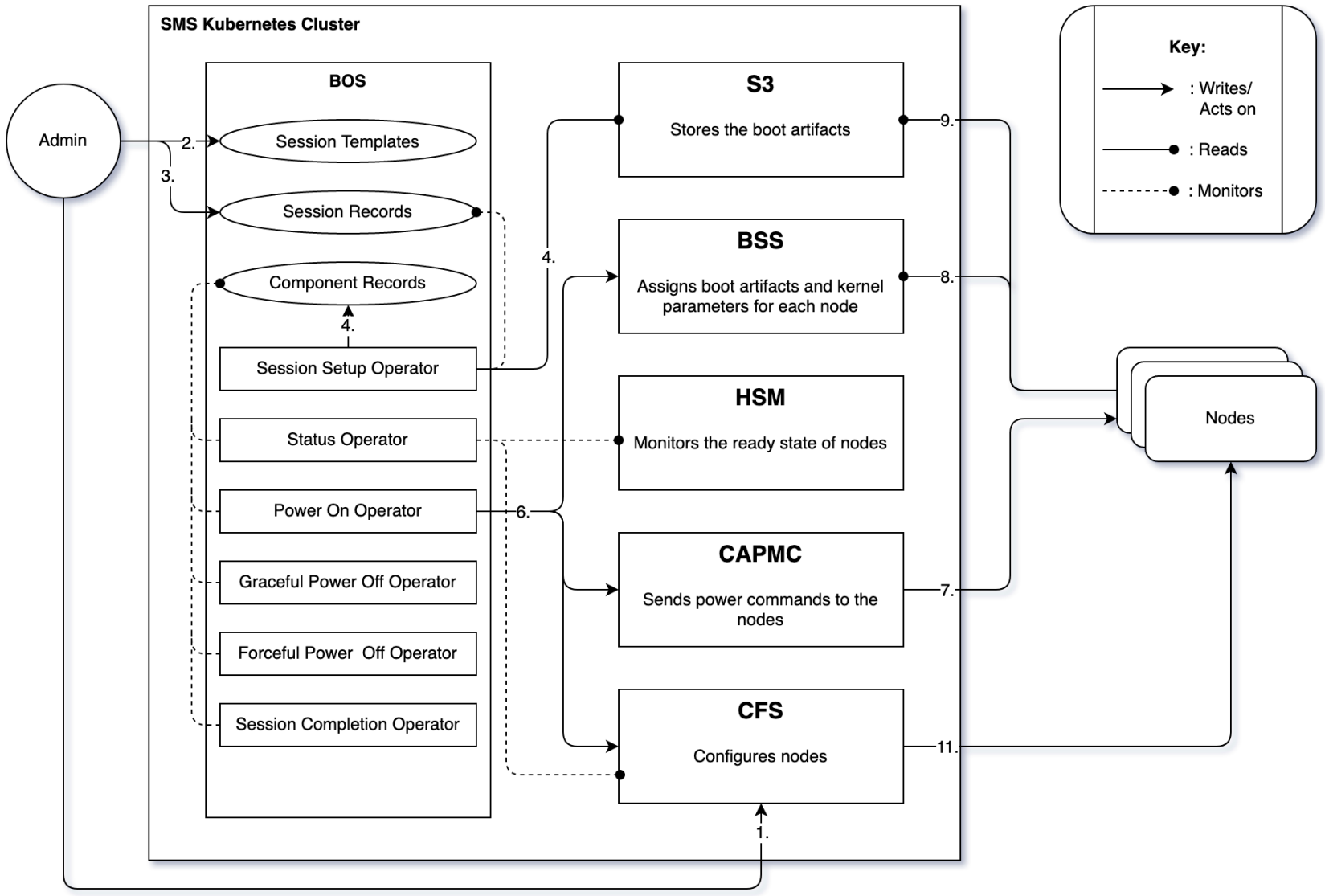
```
ncn# cray bos sessions describe 50459b1e-06d8-4708-8d55-608aea33810e --format json
{
  "components": "x1000c1s1b0n2",
  "include_disabled": false,
  "limit": "x1000c1s1b0n2",
  "name": "50459b1e-06d8-4708-8d55-608aea33810e",
  "operation": "reboot",
  "stage": false,
  "status": {
    "end_time": "2024-02-16T00:09:41",
    "error": null,
    "start_time": "2024-02-15T23:32:07",
    "status": "complete"
  },
  "template_name": "ktest-23.11.0-iscsi",
  "tenant": null
}
```

Component Status: 1 Stable and 1 Failure

```
ncn# cray bos v2 components describe x1000c1s1b0n2 --format json |
jq .status
{
  "phase": "",
  "status": "stable",
  "status_override": ""
}

ncn-m001:~/jasons # cray bos v2 components describe x1000c1s1b0n3 -
-format json| jq .status
{
  "phase": "powering_off",
  "status": "failed",
  "status_override": "failed"
}
```

BOS V2 Operators



BOS V2 Improvements

- New, higher levels of
  - Boot efficiency
  - Boot resiliency
  - Transparency

Configuration Framework Service Improvements in CSM-1.4

Two debugging improvements

- ARA – ARA Records Ansible
- CFS Debugging tool

ARA Records Ansible (ARA)

- Yes, it's another recursive acronym
- ARA is an open-source log collector, API, and UI, specifically for collecting and parsing Ansible logs
- ARA records the Ansible logs from all Configuration Framework Service (CFS) sessions
- ARA provides an Ansible friendly way to view them
- Reference: https://ara.recordsansible.org/
- ARA is installed with CSM-1.4
- ARA's GUI is accessible at: https://ara.cmn.<SYSTEM DOMAIN NAME>

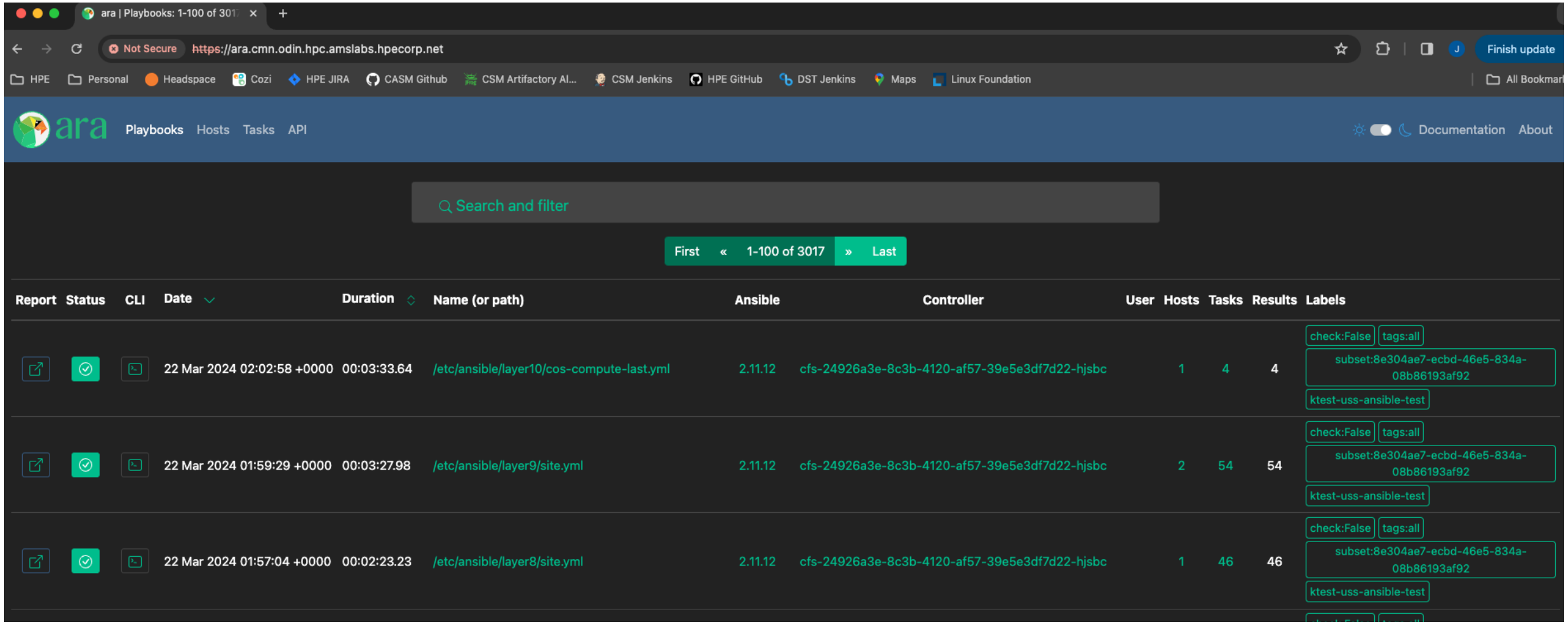
ARA Recording Workflow

- Ansible play → Ansible → ARA Callback → ARA API Server→ Database
- Graphic from: https://ara.recordsansible.org/static/recording-workflow.png

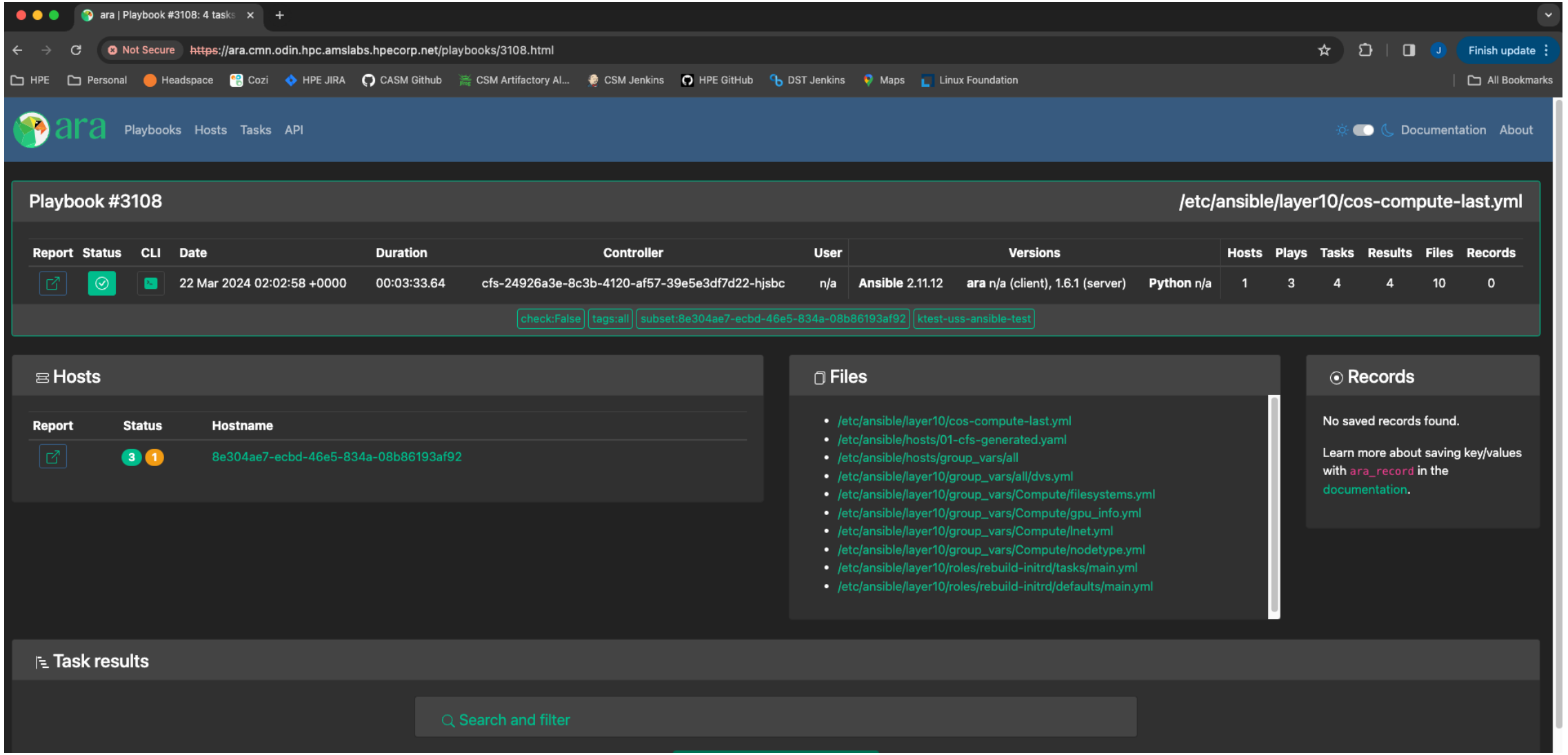
ARA User Interface

- API
- CLI
- GUI

ARA GUI Screenshot



ARA GUI Playbooks Overview



CFS Debugging Tool

- Tool to aid debugging failed CFS sessions and other CFS problems
- Invocation:
  - `ncn# cfs-debug`
- Example Display:
  - Select debugger mode. Type help for details.

```
-1) Auto-debug (default)
-2) Directed-debug
-3) Auto-debug report
-4) Collect logs
-5) Additional actions
-0) Exit
```

CFS-debug Auto-debug Output (Screen capture)

```
Running full check list...
[CFS health] The health of the CFS service
 - [ OK ] cray-cfs-api health:           healthy
 - [ OK ] cray-cfs-batcher health:       healthy
 - [ OK ] cray-cfs-operator health:      healthy
 - [ OK ] cfs-hwsync-agent health:       healthy
 - [WARN] cfs-trust health:              errors in the logs

cfs-trust health ==========================================================================================================
cfs-trust .....................  errors in logs
Would you like to see details for this service (Y/n) # Y
 - Errors found in logs for pod: cfs-trust-9f97c6c56-jkt7g container: cfs-trust
----------------------------------------------------------------------------------------------------------------------------
2024-02-20 19:16:12.906501+00:00        403 Client Error: Forbidden for url: http://cray-vault.vault:8200/v1/transit/export/signing-key/cfstrust
----------------------------------------------------------------------------------------------------------------------------


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


[CFS related health] The health of services related to CFS
 - [IMS health] The health of the IMS service
    - [ OK ] cray-ims health:              healthy
 - [Kafka health] The health of the Kafka service
    - [ OK ] kafka health:                 healthy
    - [ OK ] zookeeper health:             healthy
[CFS sessions health] The health of recent CFS sessions
 - [WARN] CFS recent session health:     Found 5 unhealthy recent sessions

CFS recent session health =================================================================================================
batcher-3d48e43f-3a58-4326-8031-aff3f1ce1f68 ......  pending   ........ 3d
batcher-1c636c6c-1840-41d6-ab6d-6a563bd53ecd ......  pending   ........ 3d
batcher-bf807057-a394-4ea7-9fd7-a2c6007d8fb3 ......  pending   ........ 3d
batcher-2d9ae6c9-62f9-49e3-a713-57b6c31bd49f ......  pending   ........ 3d
batcher-5826010c-184b-4146-808b-8710a2a777b8 ......  pending   ........ 3d
----------------------------------------------------------------------------------------------------------------------------

[CFS components health] The health of components from CFS' perspective
 - [WARN] CFS state-reporter health:     51/59 succeeded

CFS state-reporter health =================================================================================================
8 components  could not be reached  : x1000c3s2b0n3,x1000c3s2b0n2,x1000c1s1b0n3,x1000c1s2b1n0,...
----------------------------------------------------------------------------------------------------------------------------
```

CFS-debug Mode: Auto-debug Output (part 1)

```
Running full check list...
[CFS health] The health of the CFS service
 - [ OK ] cray-cfs-api health:            healthy
 - [ OK ] cray-cfs-batcher health:        healthy
 - [ OK ] cray-cfs-operator health:       healthy
 - [ OK ] cfs-hwsync-agent health:        healthy
 - [WARN] cfs-trust health:               errors in the logs

cfs-trust health
================================================================================
cfs-trust ......................  errors in logs
Would you like to see details for this service (Y/n) # Y
 - Errors found in logs for pod: cfs-trust-9f97c6c56-jkt7g container: cfs-trust
--------------------------------------------------------------------------------
2024-02-20 19:16:12.906501+00:00        403 Client Error: Forbidden for url:
http://cray-vault.vault:8200/v1/transit/export/signing-key/cfstrust
--------------------------------------------------------------------------------
```

CFS-debug Mode: Auto-debug Output (part 2)

```
[CFS related health] The health of services related to CFS
 - [IMS health] The health of the IMS service
    - [ OK ] cray-ims health:                    healthy
 - [Kafka health] The health of the Kafka service
    - [ OK ] kafka health:                       healthy
    - [ OK ] zookeeper health:                   healthy
```

CFS-debug Mode: Auto-debug Output (part 3)

```
[CFS sessions health] The health of recent CFS sessions
 - [WARN] CFS recent session health:     Found 5 unhealthy recent sessions

CFS recent session health
================================================================================
batcher-3d48e43f-3a58-4326-8031-aff3f1ce1f68 ......  pending   ........ 3d
batcher-1c636c6c-1840-41d6-ab6d-6a563bd53ecd ......  pending   ........ 3d
batcher-bf807057-a394-4ea7-9fd7-a2c6007d8fb3 ......  pending   ........ 3d
batcher-2d9ae6c9-62f9-49e3-a713-57b6c31bd49f ......  pending   ........ 3d
batcher-5826010c-184b-4146-808b-8710a2a777b8 ......  pending   ........ 3d
--------------------------------------------------------------------------------

[CFS components health] The health of components from CFS' perspective
 - [WARN] CFS state-reporter health:     51/59 succeeded

CFS state-reporter health
================================================================================
8 components  could not be reached  :
x1000c3s2b0n3,x1000c3s2b0n2,x1000c1s1b0n3,x1000c1s2b1n0,...
--------------------------------------------------------------------------------
```

CFS-debug Mode: auto-debug failure diagnosis

```
CFS recent session health
================================================================================
sat-30d0e5f0-a61b-4c07-9b0c-ac60eb0bbccc ..........  failed   ......... 1h
Would you like to see details for this session (Y/n) # Y
 - Session sat-30d0e5f0-a61b-4c07-9b0c-ac60eb0bbccc failed on playbook sma-ldms-
application.yml (container ansible)
 - Ansible failure contained task output.
 - Parsing line starting with: "fatal: [13b55ce2-14f4-4494-ab04-8437c8f8919c]: FAILED!"
--------------------------------------------------------------------------------
warning: Found NDB Packages.db database while attempting bdb backend: using ndb
backend.
--------------------------------------------------------------------------------
<?xml version='1.0'?>
<stream>
<message type="error">No provider of &apos;+sma-system-test&apos; found.</message>
</stream>
--------------------------------------------------------------------------------
- This is not a recognized failure.  Please see the owner of this role.
```

Other CFS Improvements in CSM-1.4

- Can now name the customized image from the command line
  - Allows overwriting existing images instead of always creating a new one
- CLI now allows bulk component updates
- Can now stop CFS/batcher and cancel configurations

Configuration Framework Service Improvements in CSM-1.5

Overview

- Debug_on_failure flag
- Debug playbooks
- External Repository Support
- Pagination support

Debug_on_failure Flag

- CFS sessions can be created with the **debug_on_failure** flag
- If set to true, this will cause sessions that fail during Ansible execution to remain running so that users can exec into the pod
- (ncn-mw#) kubectl -n services exec -it <pod> -c ansible -- /bin/sh
- Once debugging is complete users should touch the /tmp/complete file to complete and cleanup the session
  - If this is not done, the session will remain up until the debug_wait_time expires
- NOTE: This is only available in the v3 CFS API which was released with CSM-1.5

Debug Playbooks

- CFS supports special debug playbooks, which are part of the Ansible Execution Environment (AEE) image and always available
- These playbooks can be used without requiring a special configuration to be created
- The following playbooks are available and can be specified as the configuration name for a session if no other configuration has already been created with these names
  - **debug_fail** -
    - immediately fails
    - can be used with the **debug_on_failure** flag (previous slide) to quickl create an Ansible environment for debugging
  - **debug_facts** -
    - gathers and prints the facts for all available targets
  - **debug_noop** -
    - Way to test the CFS framework without running any Ansible tasks
    - Does not gather facts
    - Useful for skipping past the Ansible container for debugging
    - Easy way to test setting up the inventory and cloning content down

- NOTE: This is only available in the v3 CFS API which was released with CSM-1.5

External Repository Support

- CFS allows users to define optionally sources
- Sources enable cloning from external repositories
- Sources contain all the information needed to clone information from a repo
  - The username and password can be specified in source
    - CFS will store them in a Vault secret

Paging CFS Records (Pagination)

- For configurations, sessions, and templates, CFS only lists a limited number of records at a time
  - Reduces memory requirements especially on large systems
- By default, returns a number of records up to the **default_page_size**, 1000
- Can be overridden at query time
- Pages beyond the first can be requested using the **after_id** parameter, which should be set to the id of the last record in the previous page
  - CFS uses a Keyset pagination strategy

Power Control Service (PCS) in CSM-1.4

PCS Replaces CAPMC

- Cray Advanced Power Management and Control (CAPMC)

Why re-implement, re-design CAPMC?

- Architectural alignment (micro-service)
- Focus on Core functionality (resolving power states)
- Code Maintainability

CAPMC versus PCS

CAPMC	PCS	Explanation
system power monitoring	--	Split out from PCS, All environmental telemetery is part of the Shasta Monitoring Framework (SMF/SMA). SMF has the right tools and capabilities to expose the data (SQL, grafana, etc) for customer use cases.
node power/energy monitoring	--	Split out from PCS, All environmental telemetery is part of the Shasta Monitoring Framework (SMF/SMA). SMF has the right tools and capabilities to expose the data (SQL, grafana, etc) for customer use cases.
node power on/off control	component power on/off control	PCS expands power controls beyond 'compute nodes' to components more generally. This is core functionality of PCS.
power capping control	power capping control	No functional difference; API differences, but same capabilities.

PCS REST API

- Turns xnames on or off
- Performs hard and soft reset actions
- Sets and retrieves power capping parameters and capabilities

Non-blocking API

- CAPMC is mostly a blocking API; meaning that a call to CAPMC may result in a long wait time while the system resolves the operation
- PCS is a non-blocking API; returns quickly
- PCS tokenizes the request and allows the caller to get status later

Thank you

Harold Longley, harold.longley@hpe.com
Jason Sollom, jason.sollom@hpe.com