Hewlett Packard Enterprise

Building non-standard images for CSM systems

Harold Longley, Davide Tacchella, Dennis Walker, Isa Wazirzada, Andy Warner

May 8, 2025

Agenda

- Introduction
- Layered Image Model
- CSM 1.6 image curation, build, and boot toolchain
- Curating a stock SUSE image without COS/USS
- Curating a non-SUSE image
- Conclusion and Future work

Introduction

Diverse Linux toolchains tailored to project-specific needs enable functional innovations in HPC environments

- Distinct build chains/workflows with emphasis on provenance
- Adapting to the run-time needs of different user communities
- Varying security compliance
- Run managed nodes with existing virtual appliances
- Multi-system workload portability

CSM 1.6 now enables booting other Linux distributions

Rootfs Delivery Change in CSM 1.6.1+

Prior to CSM 1.6.x

- DVS kernel module required for root mount
- COS required for DVS

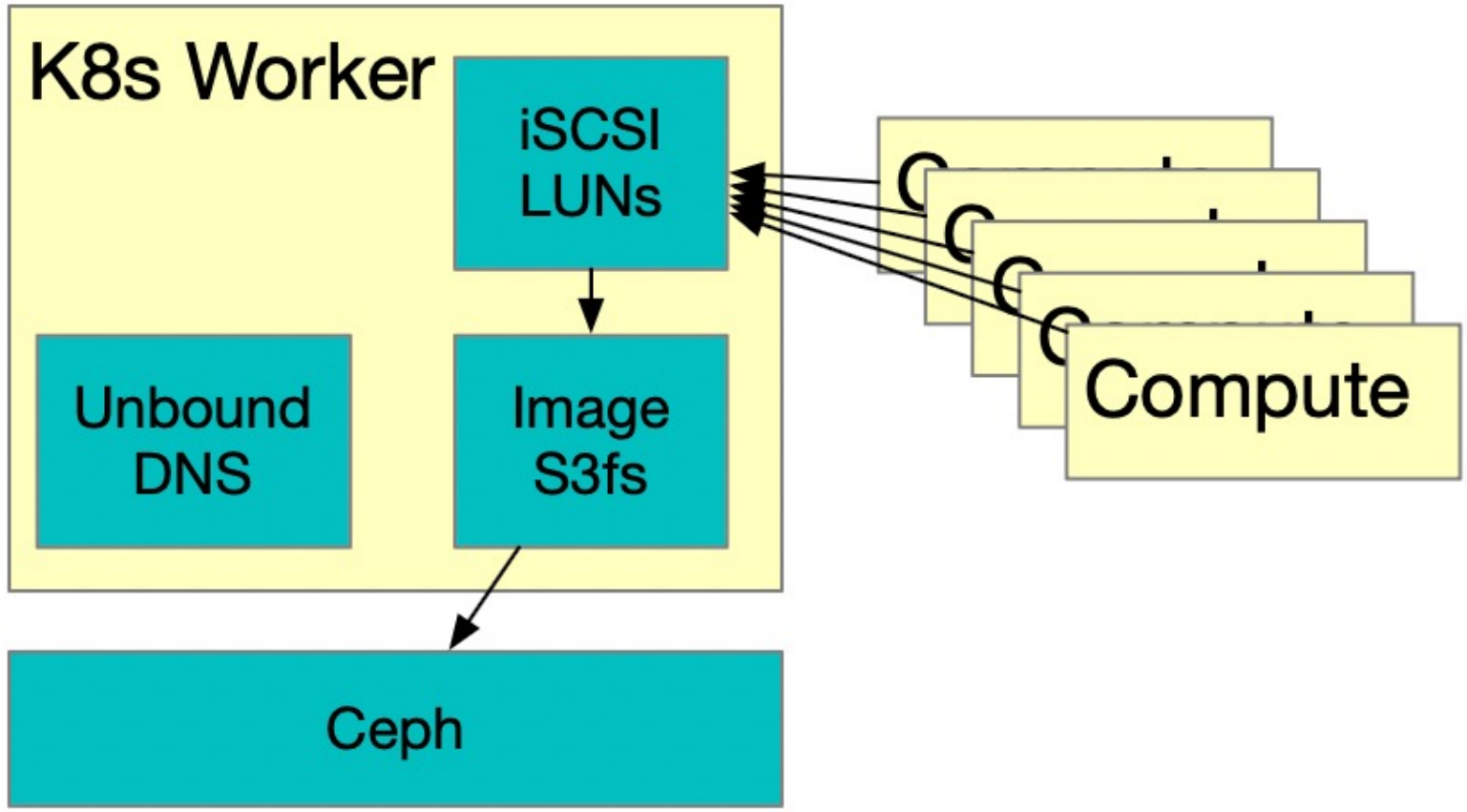
As of CSM 1.6.1

- iSCSI is now the default rootfs projection mechanism
- 1.6.1-only both DVS and iSCSI available for rootfs projection

1.7.0 and later – only iSCSI

Anatomy of iSCSI on Worker Nodes

- iSCSI installed via CFS csm_configuration Ansible play
  - Creates A records and SRV records for pointers to iSCSI services
  - Labels the node in Kubernetes
  - Installs sbps-marshall-agent
  - Creates a spire-token for IMS access
  - Populates base iSCSI targets
  - Node-exporter ingests iSCSI Linux counters
- Result: IMS images with specified label are mounted as iSCSI LUNs



How A Compute Node Mounts iSCSI

- Node boots
- Spire authenticates
- Fetches kernel/initrd
- Dracut runs
  - dig +short on DNS name provided by kernel parameters
  - For every SRV record in response
    - Mounts iSCSI target in /dev/mapper
  - Updates multipathd config to merge those
  - Mounts writable tmpfs overlay as root
- Functionality provided by cray-cps-dracut
- Images can be delivered over HSN or NMN

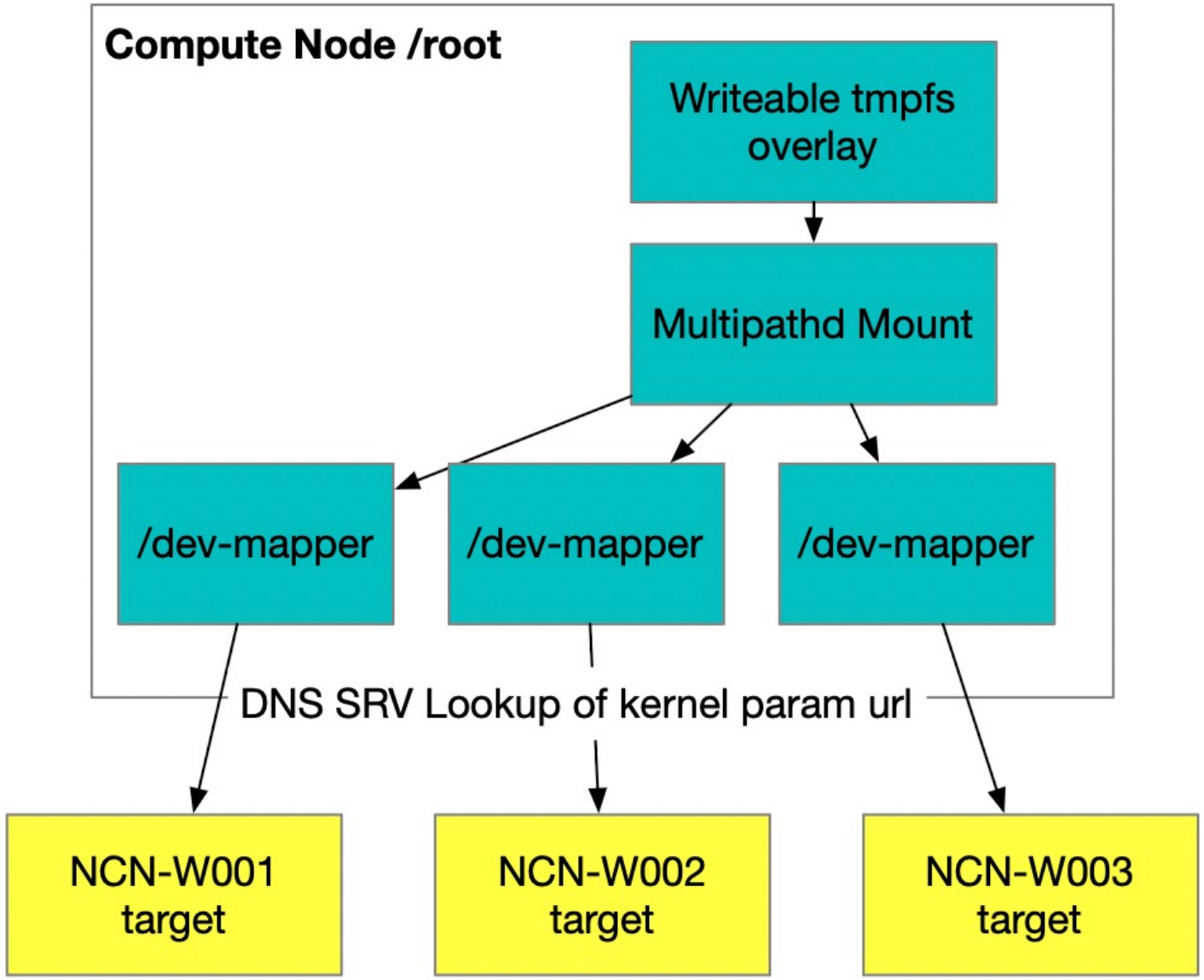
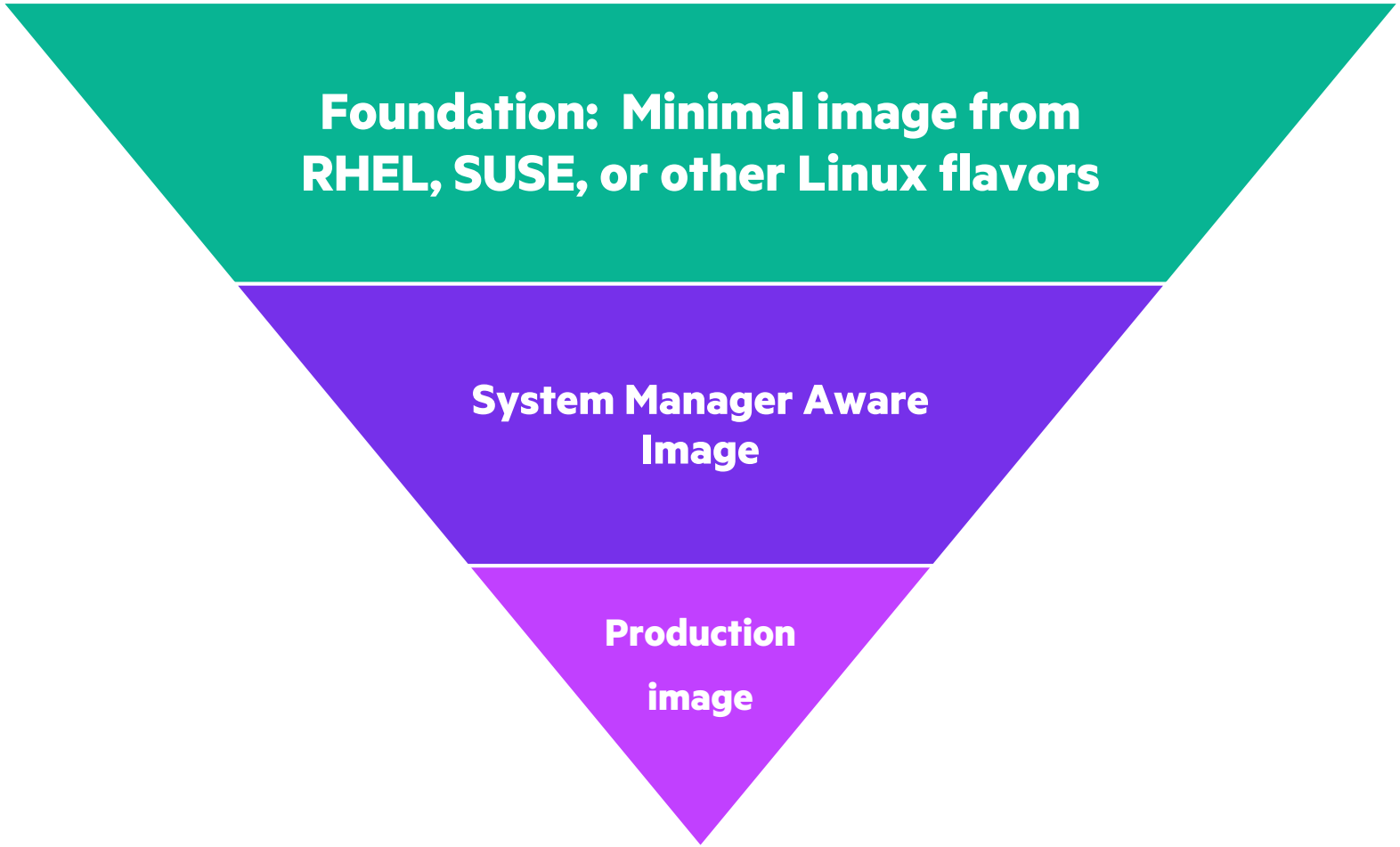


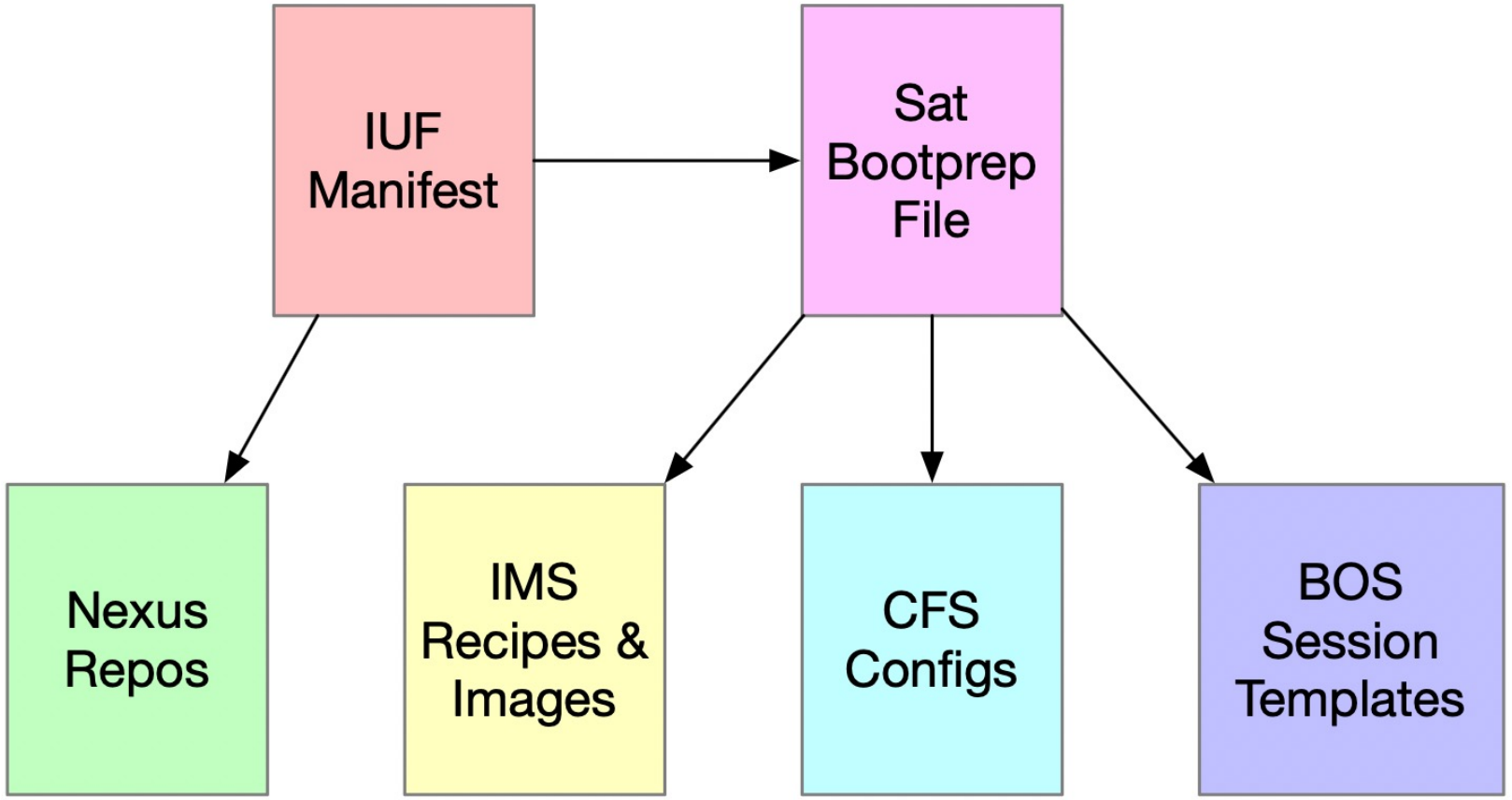
Image Maturity Model – Quality Attributes

- **Functionality**
  - Supports real user applications
- **Security**
  - TPM
  - Node attestation
- **Performance**
  - Optimized run-time configuration
- **Availability**
  - iSCSI/multipath
- **Compatibility/Interoperability**
  - Special hardware on nodes
  - Orchestration
  - Programming environment
  - Workload manager/node health
- **Observability**
  - Monitoring
  - Logging



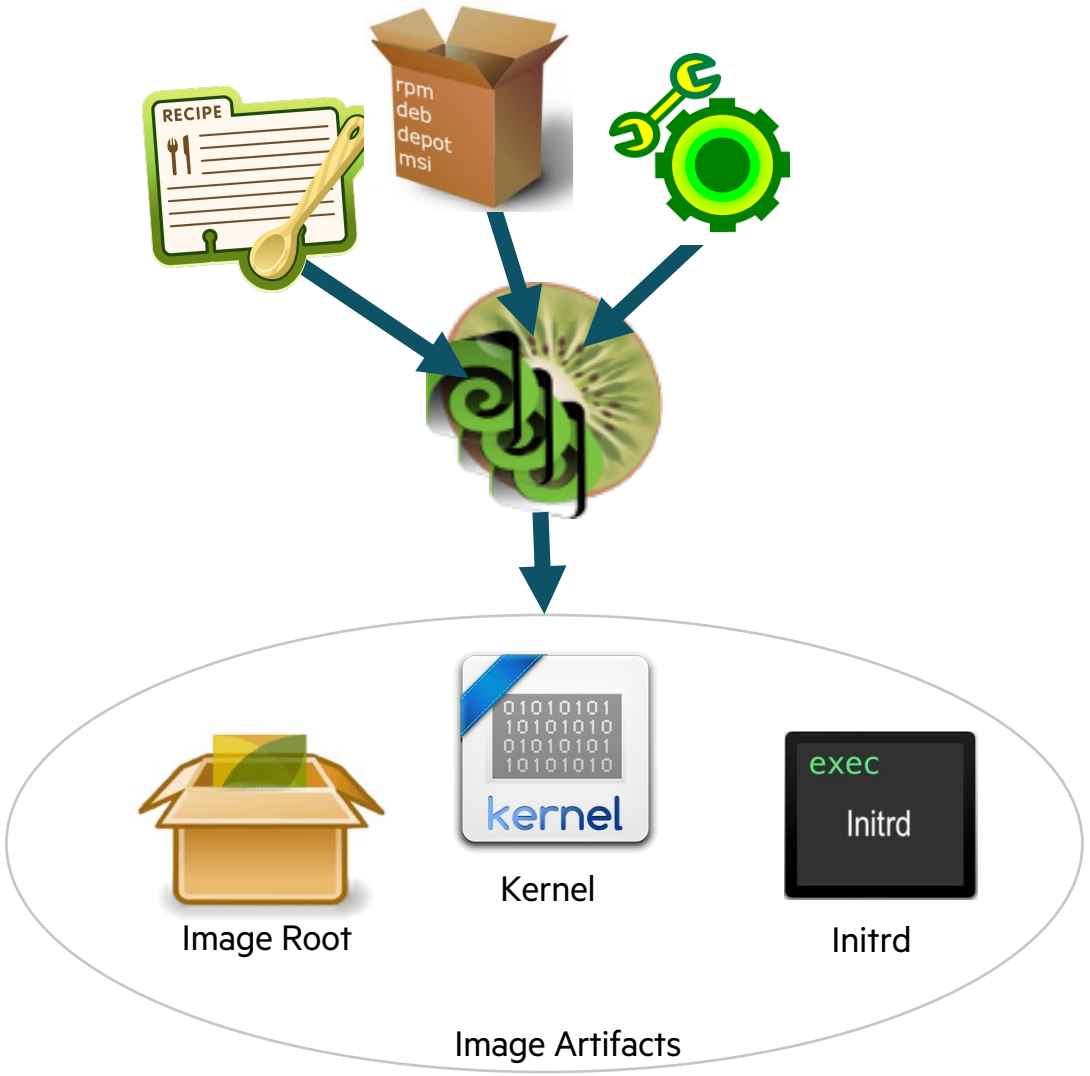
CSM Tool Overview

- IMS, CFS, BOS provide a foundation
- SAT Bootprep wraps the above
- IUF adds management of Nexus and S3 content



Example: Create a stock SUSE image

- Load Linux vendor repositories into **Nexus**
- Create a kiwi-ng recipe
  - Customize list of RPMs
- Import recipe and build base image with **IMS**
- Create a customized **CFS** configuration
- Customize image with **CFS**
- Create a **BOS** session template
- Boot nodes



SAT bootprep

- SAT bootprep streamlines
  - CFS Configuration and BOS Session Template creation
  - Image curation and creation
- SAT bootprep input file example
- Launch Example
  - **sat bootprep run bootprep_input.yaml**

```yaml
---
schema_version: 1.0.2
configurations:
- name: "{{default.note}}compute-{{recipe.version}}{{default.suffix}}"
  layers:
  - name: csm-packages-{{csm.version}}
    playbook: csm_packages.yml
    product:
      name: csm
      version: "{{csm.version}}"
  - name: shs-{{default.network_type}}_install-{{slingshot_host_software.working_branch}}
    playbook: shs_{{default.network_type}}_install.yml
    product:
      name: slingshot-host-software
      version: "{{slingshot_host_software.version}}"
      branch: "{{slingshot_host_software.working_branch}}"
    special_parameters:
      ims_require_dkms: true
   … Truncated for brevity …
images:
- name: "{{default.note}}{{base.name}}{{default.suffix}}"
  ref_name: base_uss_image.x86_64
  base:
    ims:
      type: recipe
      id: e6669c1d-8d34-41bf-8e81-69c45939c2cc

- name: "compute-{{base.name}}"
  ref_name: compute_image.x86_64
  base:
    image_ref: base_uss_image.x86_64
  configuration: "{{default.note}}compute-{{recipe.version}}{{default.suffix}}"
  configuration_group_names:
  - Compute

session_templates:
- name: "{{default.note}}compute-{{recipe.version}}.x86_64{{default.suffix}}"
  image:
    image_ref: compute_image.x86_64
  configuration: "{{default.note}}compute-{{recipe.version}}{{default.suffix}}"
  bos_parameters:
    boot_sets:
      compute:
        arch: X86
        kernel_parameters: console=ttyS0,115200 crashkernel=512M@4G ip=dhcp quiet spire_join_token=${SPIRE_JOIN_TOKEN}
        node_roles_groups:
        - Compute
        rootfs_provider: "sbps"
        rootfs_provider_passthrough: "sbps:v1:iqn.2023-06.csm.iscsi:_sbps-hsn._tcp.{{default.system_name}}.{{default.s
```

Compute Image Content (SLES example)

Required early boot, CSM, and iSCSI content

- cray-cps-Dracut
- cray-netif-Dracut
- cray-cps-utils
- iscsid
- multipathd
- cray-node-identity
- cray-heartbeat-service
- spire-agent
- cfs-state-reporter
- bos-state-reporter
- csm-node-heartbeat
- cfs-trust
- cfs-debugger
- csm-auth-utils
- cray-boot-parameters-shasta
- tpm-provisioner-client

Optional workload supporting content

- SLURM/Munge
- cray-setup-scripts
- Cray Programming Environment (CPE) – post boot*
- LNet
- bpcmdmod – Power Utilization Counters
- Network Drivers and Kernel Modules
- Lustre Client
- xpmem
- cray-crash-utility
- msr-tools
- cray-atom-energy-plugin
- cray-freemem
- cray-rasdaemon
- cray-pals
- msr-safe
- cray-hugepage-setup
- cray-libhugetlbfs

Create a non-SUSE image

- Create base OS image outside of IMS
- Import external image into IMS
- Create CFS configuration
  - CSM and SHS (Slingshot Host Software) layers will add rpms and configuration
  - Site CFS layer
    - May need to translate some USS Ansible code to be used here
      - Filesystem configuration, LDAP configuration, Slurm or PBS Pro client and configuration, etc.
- Customize imported image
- Assign image to nodes with BSS or create BOS session template
  - BOS could disable or enable post-boot configuration with CFS
- Boot nodes

Conclusion and Future Work

- Key Achievements:
  - Leveraged standard CSM tools to curate, build, and boot non COS/USS content
    - CFS, BOS, IMS
    - Brought together with `sat bootprep`

- Future work
  - Continue validating the image model
  - Incorporating GPU tools and Cray Programming Environment (CPE) into configuration of image
  - Building images for RHEL and Rocky Linux using Nexus and IMS
  - Convert boot image to OCI container for use as UAI container on UAN or on compute node via podman, Singularity, or Kubernetes

Thank you

davide.tacchella@hpe.com
dennis.walker@hpe.com
isa.wazirzada@hpe.com
harold.longley@hpe.com
andy.warner@hpe.com