Hewlett Packard
Enterprise

MANAGING USER ACCESS WITH UAN AND UAI

Alex Lovell-Troy, CSM Cloud Architect
Harold Longley, CSM User Experience Solutions Architect

CUG 2021 May 3-5, 2021

AGENDA

- Introduction and Overview
- User Access Options
- Resource Management
- Authentication and User Lifecycles
- Logging
- Data Access
- Future Work

INTRODUCTION AND OVERVIEW

UAN and UAI can both provide access to Cray Supercomputer resources for users to develop, compile, launch jobs on the compute nodes, and analyze the results

For long-running operations or memory-intensive activities, UANs are most suitable.

For flexible, cloud-like interactions, UAIs are most suitable.

User Access Node (UAN)

- Dedicated node (HW & SW) ideal for stable long term persistent tasks
- Can be configured to match compute nodes with GPU
- Can use DVS for filesystem; not necessarily rely on local disk
- Can take full advantage of all hardware resources and swap for memory-intensive workloads

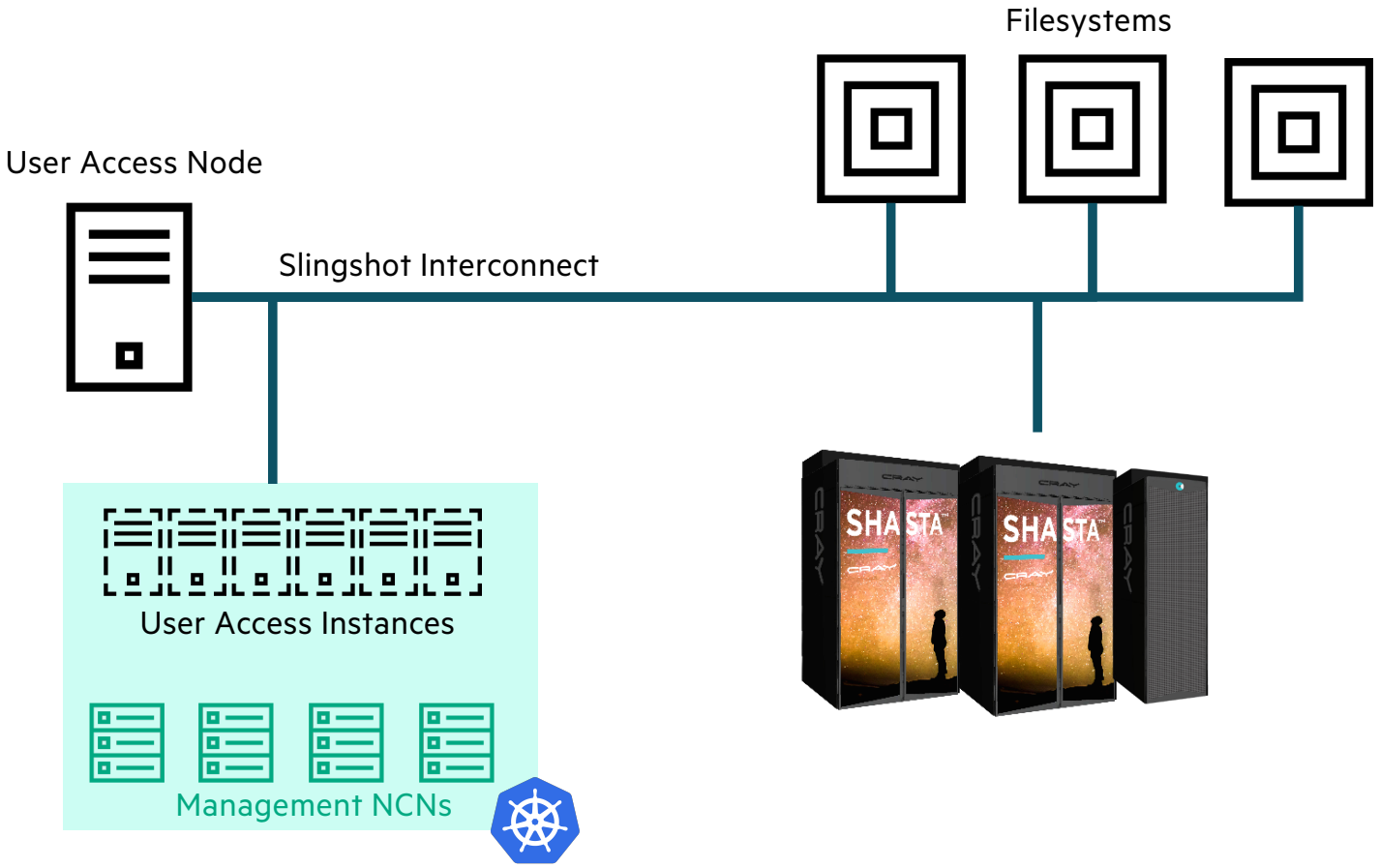
User Access Instances (UAI)

- On-demand; ideal for short-term interactive tasks
- Disposable; environments can be suspended without loss of user data
- Underlying host can be configured to match compute nodes with GPU
- Administrators can limit resources available per user

USER ACCESS OPTIONS

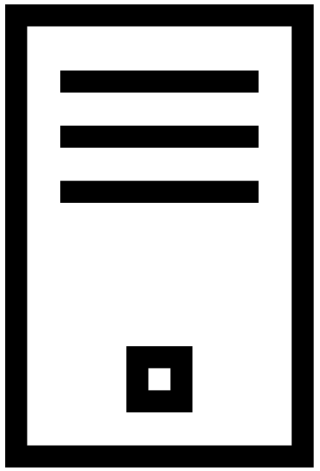
Power Users
Compile and Run

Standard Users
Run and Monitor



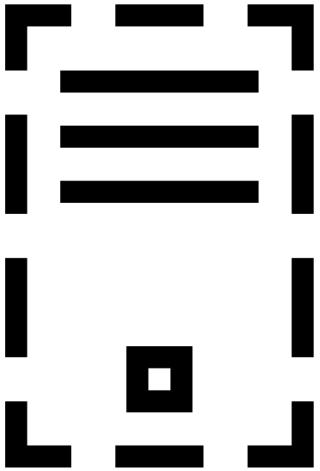
USER ACCESS NODES IN SHASTA 1.4

- Dedicated Hardware
  - Validated on:
    - HPE Proliant DL 325 & DL 385
    - Gigabyte R272-Z30
- System image from same sources as Compute Nodes
  - Image Management Service
  - Image Customization through CFS
  - Cray Operating System based on SLES15 SP1
    - Linux Kernel 5.3
- Limitations in 1.4
  - IP_VS Support
  - eBPF Support



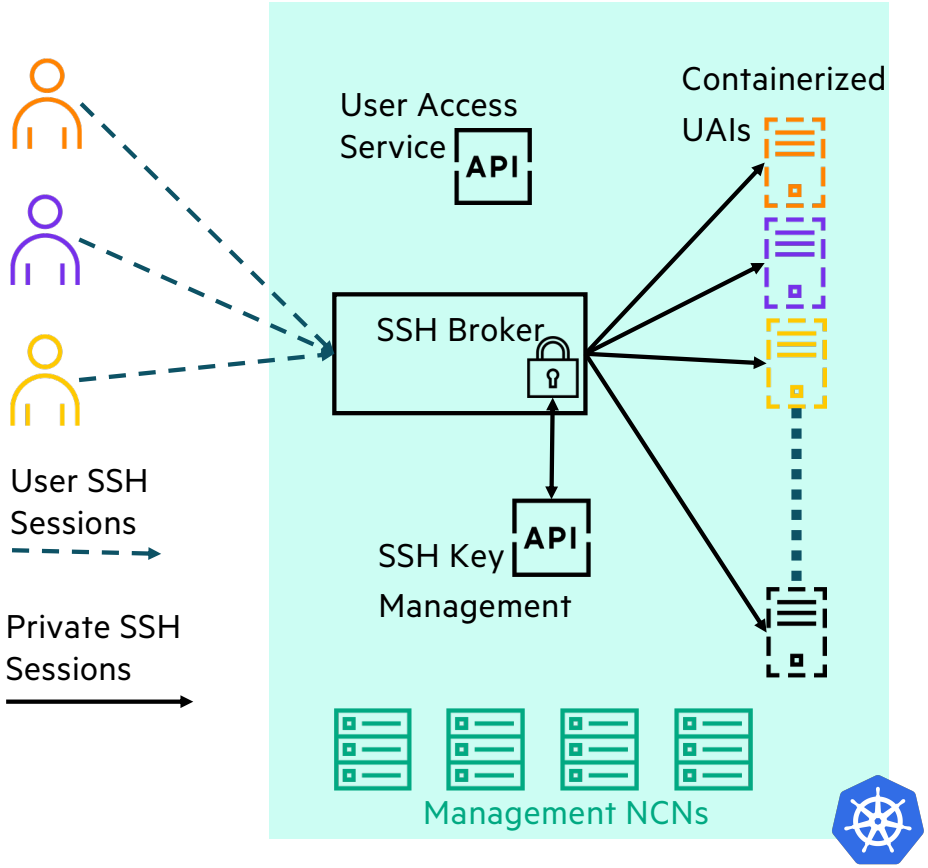
USER ACCESS INSTANCES IN SHASTA 1.4

- “Serverless”
- Containerized
- Templated
- Disposable
- Preserves User State



USER ACCESS SERVICE AND BROKER

- On-Demand containerized SSH environment "serverless"
- SSH is the only User-Facing API
- Templated UAI Pods launched and destroyed as-needed
- User state persisted only in cross-mounted filesystems (like /home)
- Internal SSH relies only on single-use SSH keys
- Broker consumes a single IP regardless of how many users
- Multiple brokers can be used to handle different user types and user groups



RESOURCE UTILIZATION

- Is a user hogging resources?
- UAN
  - Linux tools to investigate
- UAI
  - Kubernetes tools for setting and checking resource limits across all nodes
  - Memory is a HARD limit – Kubernetes nodes don't have swap
  - Possible to identify node with running UAI and use Linux tools

AUTHENTICATION OPTIONS

Linux Logins

- Standard Linux Login items
- SSH Access
- Users and Groups
- PAM

Web and API Authentication

- Web Based OAuth2 (OpenID Connect)
- Keycloak w/LDAP
- Keycloak Gateway
- Not part of UAI or UAN authentication

KEYCLOAK FOR LDAP FEDERATION

- keycloak provides access via API gateway to services (such as UAS)
  - keycloak can link to external LDAP servers
  - keycloak can have local accounts
- How long does it take for user lifecycle changes to take effect?
  - keycloak syncs from LDAP regularly
  - Sync from keycloak to S3 for Slurm/munged /etc/passwd, /etc/group
    - Nothing by default in v1.4, but could create Kubernetes cronjob
  - Sync from S3 Slurm/munged to nodes' /etc/passwd, /etc/group
    - Nothing by default in v1.4, but could create Linux cronjob on compute nodes

TEMPORARILY RESTRICT ACCESS

- UAN
  - Disable account in LDAP or set non-existent shell
  - Standard Linux methods
    - /etc/nologin to reject all non-root access
    - /etc/security/access.conf
    - PAM and SSSD controls
      - See UAN Ansible role uan_ldap
  - Can write custom Ansible play to apply Linux methods and run post-boot CFS to reconfigure UAN
- UAI
  - No restriction once SSH key pair has been used to create UAI pod

PAM VS SSH KEY LOGIN

- keycloak provides access via API gateway to services (such as UAS)
  - keycloak can link to external LDAP or AD servers
  - keycloak can have local accounts
- UAN login access via Linux SSSD/PAM
  - Can link to external LDAP or AD servers
    - Configured by Ansible plays run post-boot by CFS (Configuration Framework Service)
  - Can have local accounts
    - Configured by Ansible plays run post-boot by CFS
- Legacy UAI login access via ssh public/private key pair provided when UAI created
  - Possession of private key enables connection to UAI as that account
  - Job submission still requires Linux uid/gid on compute nodes
- New UAI broker login access via SSSD/LDAP

ACCOUNT TERMINATION

- Common
  - Remove account from LDAP
  - Delete running and queued jobs in WLM
  - Clean up files in network filesystems (NFS, Lustre, or SpectrumScale)
- UAN
  - Remove cronjobs for account
  - Kill running processes for account
  - Clean up files in local filesystems owned by account uid
- UAI
  - Delete UAI Kubernetes pods for account

ACCESS LOGS

- Common
  - Log aggregation from nodes and pods sent to SMA
    - Searchable via Kibana or ElasticSearch
    - May be forwarded to site logging infrastructure
- UAN
  - Standard syslog entries sent to SMA
  - Enable audit logging or failed login attempt logging using Linux method
    - Write custom Ansible play run by CFS to make Linux change
  - Console login attempts collected into cray-conman console log sent to SMA
- UAI
  - Pod log sent to SMA
  - No method inside pod for audit logging or failed login attempt logging

DATA ACCESS

- UAN
  - Client for remote filesystems
    - UAN mounts NFS, Lustre, or SpectrumScale filesystem
  - All client commands available
    - Lustre lfs and lctl
- UAI
  - Worker node hosting UAI is client of remote filesystem
    - Worker node mounts NFS, Lustre, or SpectrumScale filesystem
    - UAS configuration enables that mount point inside UAI pod
  - Lustre client commands not available in v1.4 (lfs, lctl)

FUTURE WORK

THANK YOU

alt@hpe.com
harold.longley@hpe.com