HP-UX Containers

An introduction to the products and features of HP-UX Containers

Technical white paper

Table of contents

HP-UX 11i v3: The operating system of the mission-critical HP Converged Infrastructure 2

HP-UX Containers 2
Overview 3

Why HP-UX Containers? 4

Container types 5
System containers 5
Workload containers 6
HP 9000 containers 7

Choosing the right container 8
When to use an HP 9000 container 9
When to use a workload container 9
When to use a system container 10

For more information 11

HP-UX 11i v3: The operating system of the mission-critical HP Converged Infrastructure

HP-UX 11i v3 is designed to simplify and unify IT, and deliver the always-on resiliency, dynamic optimization of resources, and investment protection and stability demanded in mission-critical computing. It integrates proven UNIX® functionality with advances in high availability, security, partitioning, infrastructure and workload management, and instant-capacity-on-demand. It delivers this functionality within the industry's one of the first mission-critical converged infrastructures, to drive up flexibility while reducing risk and delivering compelling value.

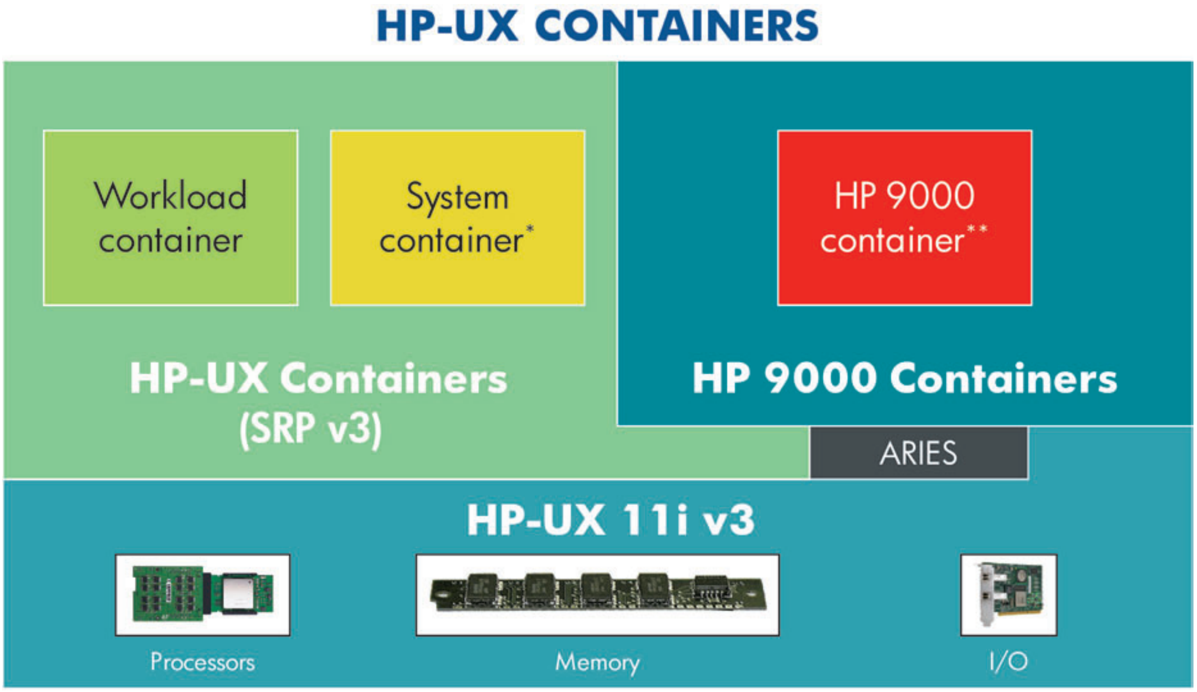
HP-UX delivers built-in integration of virtualization and management software to dynamically enhance IT infrastructure. Within the HP-UX 11i v3 Virtual Server Operating Environment, HP offers a comprehensive line of virtualization capabilities designed to help customers get the most from their HP Integrity servers, by consolidating diverse workloads to help improve ROI.

HP-UX Containers

The HP-UX Containers brand consists of two HP products: HP-UX Containers (previously known as HP-UX Secure Resource Partitions) and HP 9000 Containers.

The HP-UX Containers product provides the core foundation for containers on HP-UX, in addition to two container types: workload and system (new in HP-UX Containers v3). The HP 9000 Containers product can be added to provide a third container type: HP 9000.

Figure 1: HP-UX Containers Portfolio



* System Containers are new in HP-UX Containers (formally known as HP-UX SRP).

** HP 9000 Containers are currently supported on HP-UX SRP v2 only.

Overview

HP-UX Containers provide multiple container types used to create an isolated operating environment within a single instance of the HP-UX 11i v3 operating system. HP-UX Containers allows the enterprise to host varied application workloads in secure individual operating environments on a single physical server, thereby better utilizing server resources (CPU, memory, and network access) and data center resources (power, cooling, and space).

All HP-UX Containers-enabled systems have a global view where the system level processes run. Processes running in the global have no additional access restrictions to resources on the system; with the possible exception of cores that have been dedicated to a container using PSETs.

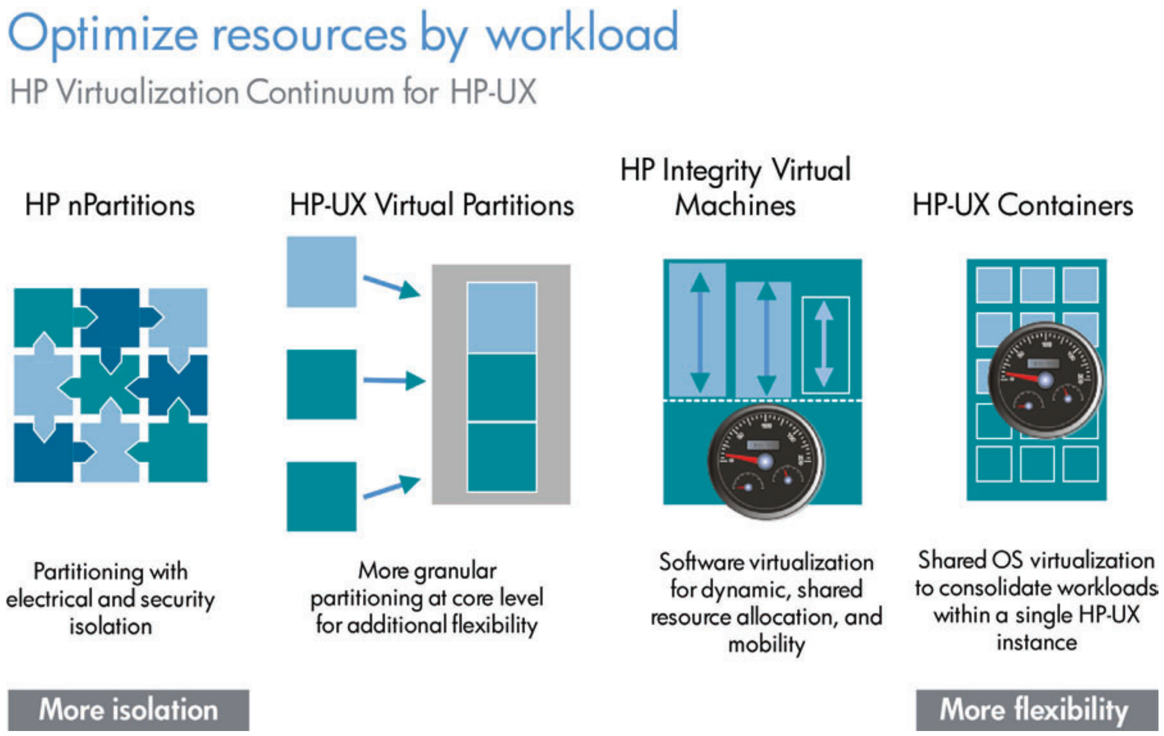
System level administration functions such as container management, software maintenance with Software Distributor (SD), device management, network interface management, kernel modifications, and system management utilities such as smh(1M) should be done from the global. Any non-management or non-system-administrative applications on the system should be hosted in a container. Some tasks such as file backup and recovery can be done from the global view or from within a container.

HP-UX Containers utilizes Process Resource Manager (PRM) to set resource entitlements for containers on the system. By assigning a container a PRM group, administrators can assign the container CPU and memory entitlements. Resource entitlements consist of a guaranteed minimum amount of the resources and can optionally include resource caps, ensuring the container does not exceed a predefined limit for the resource. CPU entitlements can utilize the Fair Share Scheduler (FSS) in which multiple CPUs are shared across containers, each container having a guaranteed minimum number of CPU shares. Container CPU entitlements can also utilize PSETs, allowing a number of cores to be dedicated to that container and unusable by other containers or the global.

In addition to container type specific capabilities, all containers can utilize the following features:

- An isolated container home directory
- A dedicated network interface and IP address
- Container-specific login environment
- Isolated inter-process communication (IPC) and process view
- Dedicated per-container CPU and memory resource allocations
- Per-container initialization and shutdown capabilities
- Container-specific network security policies
- Import and export containers between systems to facilitate workload balancing
- Cloning to support high availability environments

Figure 2: HP-UX Containers and the Virtualization Continuum for HP-UX



Optimize resources by workload

HP Virtualization Continuum for HP-UX

HP nPartitions
HP-UX Virtual Partitions
HP Integrity Virtual Machines
HP-UX Containers
Partitioning with electrical and security isolation
More granular partitioning at core level for additional flexibility
Software virtualization for dynamic, shared resource allocation, and mobility
Shared OS virtualization to consolidate workloads within a single HP-UX instance
More isolation
More flexibility

HP-UX Containers is a component of the Virtualization Continuum for HP-UX and is compatible with HP-UX nPartitions, HP-UX vPar, and Integrity Virtual Machine (VM) solutions. You can create containers in any HP-UX OS image; the OS image can exist in an nPartition, vPar, Integrity VM, or directly on non-partitioned server hardware.

Why HP-UX Containers?

HP-UX Containers provides an ideal operating environment for consolidating multiple workloads on a single system. This reduces the number of operating system environments requiring administration and support, minimizing “OS sprawl” issues encountered with other virtualization models. Container technology is built into the operating system itself, which allows containers to support small to large workloads with negligible overhead. HP-UX Containers simplifies system resource management by providing the ability to automatically balance CPU and memory needs of workloads running in containers. This “set and forget” method that lets HP-UX Containers dynamically adjust to meet capacity requirements is balanced by an ability to cap the amount of CPU or memory that a container can use.

Container types

HP-UX Containers provides multiple container types. Each container type offers unique features, which allow administrators to choose the container that best fits their workload requirements. The next section gives a brief overview of each container type. For more information on system and workload containers, visit: http://www.hp.com/go/virtualization-manuals.

For more information on HP 9000 containers, see the HP 9000 Containers Administrator's Guide at: http://www.hp.com/go/hp9000-containers.

System containers

System containers provide virtualization and private namespace capabilities that give users and applications the look and feel of a private operating system instance. The unique namespace eases application deployment with out-of-the-box implementations since it avoids name clashes that are common when consolidating workloads within one OS instance.

As with all container types, each system container has a private directory under /var/hpsrp. However, a process that runs in a system container has its file system root set, using a secure chroot() at the containers private directory (/var/hpsrp/$CONTAINER) instead of the system's file system root (/). This allows each system container to have their own copy of files that are accessed using the same file system path as other containers. For example, a system container named sys1 would access its private passwd file using the standard path of /etc/passwd when that file's real path is /var/hpsrp/sys1/etc/passwd. System container sys2 would access its private passwd file using /etc/passwd, which has a real path of /var/hpsrp/sys2/etc/passwd.

There are two types of file system layouts available for system containers (specified when you create a system container):

- **Shared:** The container shares the /usr, /sbin, and /stand directories with the global (read-only)
- **Private:** The container only shares the /stand directory with the global (read-only); /usr and /sbin are private (read/write) to the container.

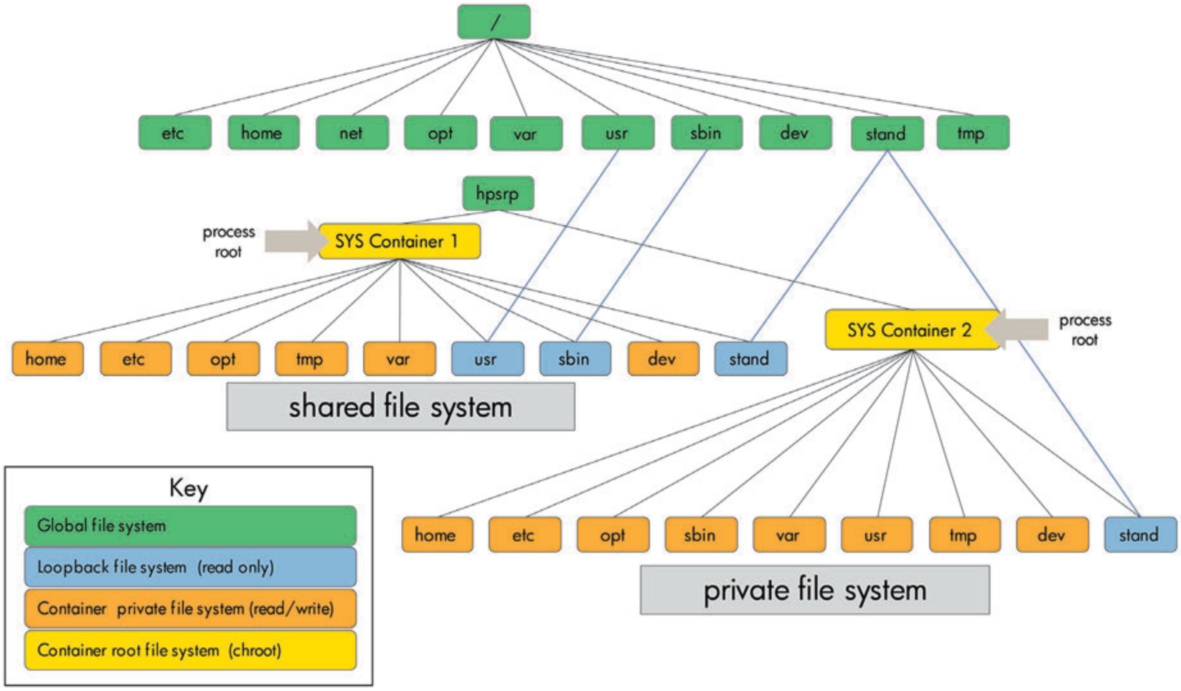
See figure 3 for a detailed layout of a system container file system access.

Each system container has:

- A unique host and node name
- Local users and groups (including a local root user)
- Local NIS or LDAP domain
- Local password policies
- Local file system view (private or shared)
- Local system services (for example, init, sshd, pwgrd, syslogd, and inetd)
- Private network interface and IP address
- Private IPC namespace
- Local NFS Client and AutoFS support

Both system and workload containers are managed using the same tools, including the SRP Manager integrated with SMH. Container management on HP-UX Containers v3 use the same commands and tools as HP-UX SRP v2, making the transition from HP-UX SRP v2 to HP-UX Containers v3 easy for administrators.

Figure 3: System container file system layout



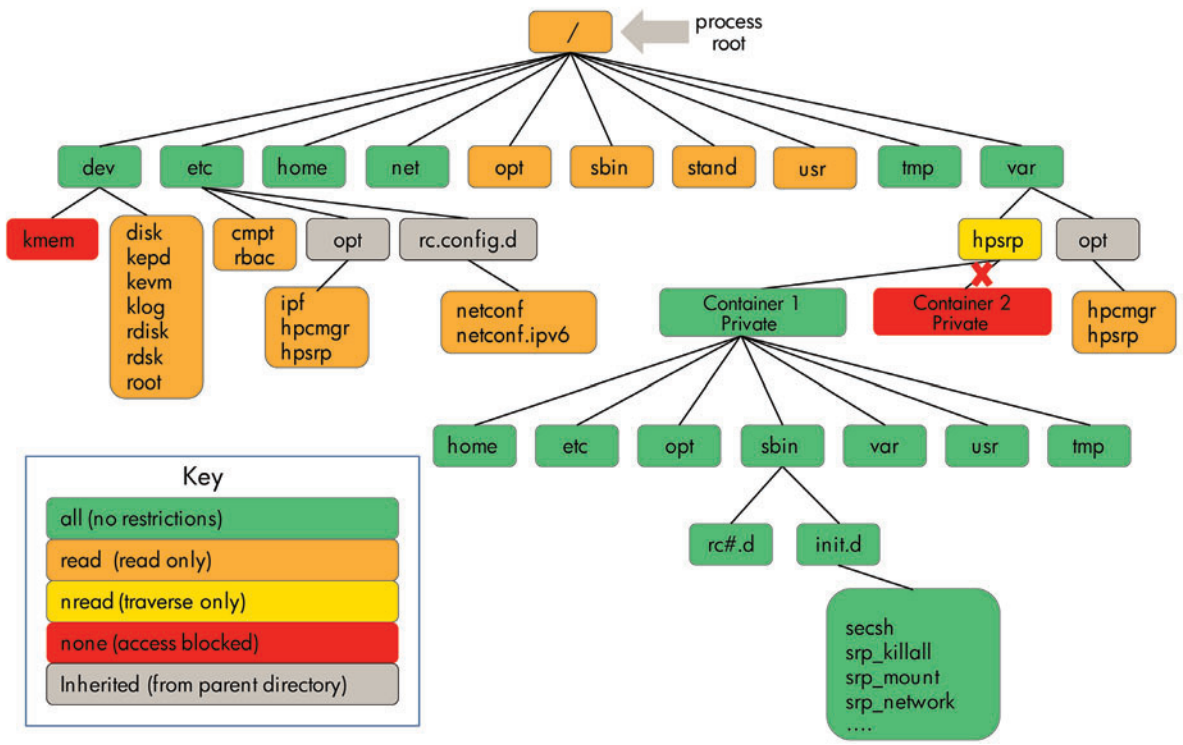
Workload containers

Workload containers provide a lightweight workload hosting environment. All workload containers share their file system view, hostname, IPC namespace, and service daemons with the global. A workload container has restricted access and view of these resources, rather than a virtualized view. This restricted view of a workload container can be customized to meet the workload's requirements. For example, it is possible to allow one workload container to open IPC communications with another workload container, something that is not supported with system containers.

Like all containers, a workload container has a private directory under `/var/hpsrp` that is only accessible by that container. Because workload containers share the entire file system with the global and other workload containers, they can share application directories with other workload containers. For example, the `/opt/my_application` directory is shared across all workload containers. See figure 4 for a detailed layout of a workload container file system access.

You can create a workload container very quickly and it requires little disk space. Upfront application deployments can require customization compared to other container types. However, ongoing maintenance of a workload container is typically less than other container types as most system administration activities for workload containers are shared with the global view.

Figure 4: Workload container file system layout



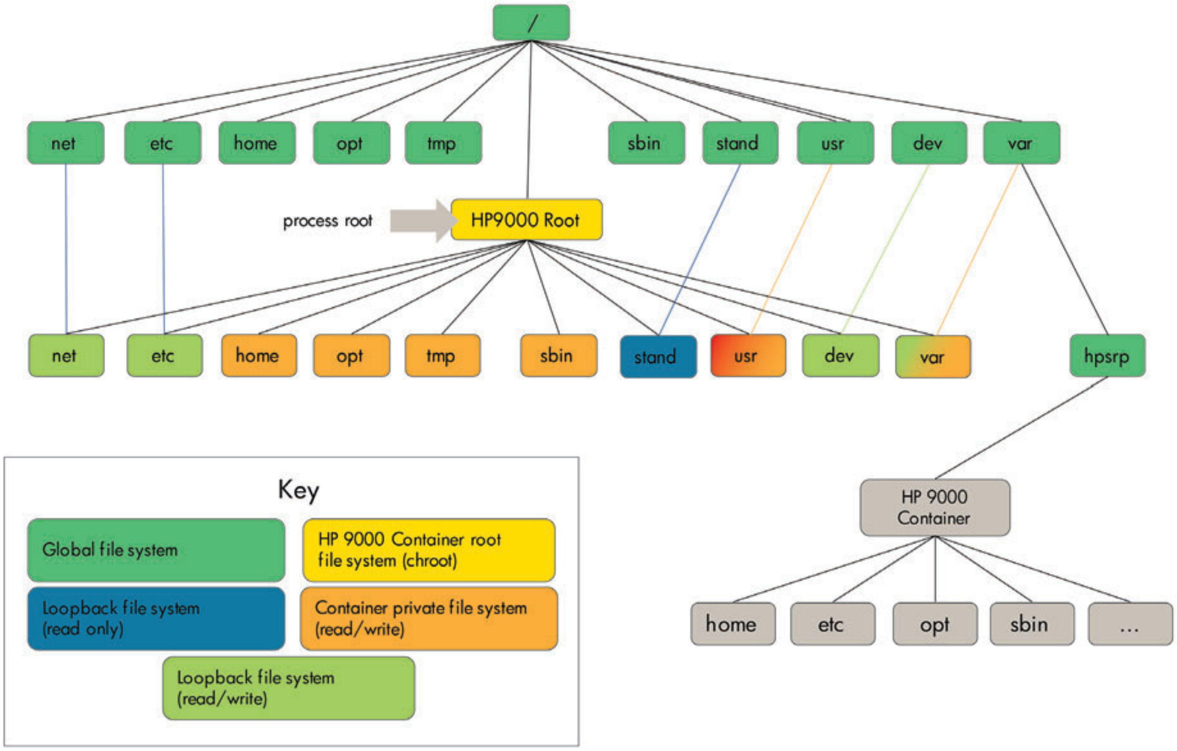
HP 9000 containers

HP 9000 containers replicate an HP 9000 HP-UX ecosystem and create a chroot isolated virtual HP 9000 file system. An HP 9000 container provides an environment to re-host the complete HP 9000 user-space environment without the need to recompile or re-install individual applications or reconstruct application ecosystem, all with minimal reconfiguration and application inventory preparation effort.

The transitioned application(s) reside in a chroot environment (HP 9000 container) along with the HP 9000 commands and libraries. The HP 9000 container has its own IP address that can be used by applications to access services running in the container, including remote SSH access. You can start, stop, export, import, and delete an HP 9000 container. However, HP 9000 containers do not support the HP 9000 HP-UX kernel, kernel intrusive applications, system administration commands, and system management related applications inside them.

NOTE:
As of the release of this document, HP 9000 Containers is only supported with HP-UX SRP v2. Support for HP 9000 Containers with HP-UX Containers v3 is currently under development.

Figure 5: HP 9000 container file system layout



Choosing the right container

HP-UX Containers provide multiple container types to meet the workload hosting needs of your environment. In many situations more than one container type may be utilized to meet your needs. In order to select the best container type for the workload, you will need to compare the application deployment, administrative, high availability, and compliance characteristics of the container types with those of your workload.

A workload can be defined as a related set of applications and supporting services—which is far more than just an application. A consolidation project should start at the workload level, and not at the application level, to ensure that all environmental variables are considered during the planning phase.

The following section provides an initial guide to help you determine which type of container is best for your workload. Consult the HP 9000 or HP-UX Containers administration manuals for a detailed description of each container features.

When to use an HP 9000 container

HP 9000 containers are ARIES-based PA-RISC execution environments. The applications under transition consideration must be suitable for ARIES emulation. For example, only user space applications are supported. If any application depends on kernel modules or specific device drivers, it will not work correctly under the ARIES and HP 9000 Containers environment. For more information, refer to the ARIES binary compatibility and product support statement.

While stand-alone ARIES-based transition can be the preferred path for standard ISV applications, the HP 9000 Containers solution is a better alternative for the transition of legacy environments that include mostly custom applications. Use HP 9000 Containers solution if:

- The application environment is very old and the information about application components and dependencies is not readily available.
- The application environment is very complex, such that it is difficult to extract the applications along with relevant dependencies.
- The transition project aims to modernize large number of applications mostly not up to date with latest/current versions of ISV software.
- The transition project aims to modernize a large fleet of PA-RISC based HP 9000 HP-UX servers.
- Ideal for running PoC (proof of concept) tests of applications being considered for use with ARIES.

When to use a workload container

Workload containers provide a filtered view of all resources on the system, including processes, file systems, and network interfaces. Workload container configuration is highly flexible and allows administrators to set access controls for a workload container that meets the workloads needs. Workload containers can help simplify synchronizing environments across servers when utilizing Serviceguard, or similar high availability solutions. You should also consider utilizing workload containers when the level of isolation for the container needs to be customized to meet application compatibility or compliance purposes.

Because of this flexibility, workload containers are ideal for workloads that:

- Require IPC access to other applications running in a separate workload container
- Interact with services provided in the global
- Need to share common IP address access with the global or other workload containers
- Have minimum disk space available
- Need a quick setup time
- Share container directory across Serviceguard nodes
- Have minimal software dependencies on migration across HA nodes
- Have no requirement for:
  - Unique hostname
  - Private file system
  - Private IPC namespace

When to use a system container

System containers provide a look and feel of a standalone system and many of the capabilities of a virtual machine guest without the associated management and performance overhead. System containers allow you to migrate more of the workload's operating environment, minimizing the process and administrative changes required. When users log into a system container, they have private versions of many of the system services that they have in a standalone operating system. System containers are ideal for hosting environments that require per container administration similar to a standalone system. System containers provide an easier up front migration path for workload consolidation by providing a similar environment to the original workload's environment. Use system containers when:

- The workload requires a unique hostname
- Users/group names need to be private to the container
- Application installation does not allow for alternate installation path
- Private system services are required
- Migrating from other container technologies
- Application deployment and modifications are common

A system container provides virtualization features not found in a workload container allowing for more services to be run in on a per-container basis.

For more information

To read more about:

- HP-UX Containers, go to:
  www.hp.com/go/containers
- HP-UX virtualization and infrastructure management products, go to:
  www.hp.com/go/vse
- HP-UX, go to:
  www.hp.com/go/hp-ux

To learn more about how HP-UX Containers allow enterprises to host varied application workloads in secure individual operating environments in a single OS instance, visit www.hp.com/go/containers.

4AA3-4845ENW, Created July 2011