# HP 9000 Containers A.03.01 on HP Integrity – Administrator's Guide

HP-UX 11i v3

Version 03.01.04.02 (Nov 2012)

# Table of contents

# Preface

This document describes how to install HP 9000 Containers A.03.01.04 on HP-UX 11i v3, how to transition application environment from an HP 9000 server to an HP 9000 container, how to perform configuration and management tasks, and how to troubleshoot issues. It also lists the known limitations of the HP 9000 Containers and provides an overview of the file system layout.

## Intended Audience

This document is intended for system administrators, solution architects and others involved in transitioning applications from legacy HP 9000 servers to HP-UX 11i v3 on newer HP Integrity servers using HP 9000 Containers, and in managing the transitioned environment.

## Typographic Conventions

The following conventions have been used in the document.

$ A dollar sign represents the system shell prompt

## Related Information

For more information on HP 9000 Containers product, refer to
http://www.hp.com/go/hp9000-containers

For information about the products used for building HP 9000 Containers

- HP ARIES dynamic binary translator
  http://www.hp.com/go/aries
- HP-UX Containers (formerly Secure Resource Partitions – SRP)
  http://www.hp.com/go/containers

For information about the HP OverEasy portfolio of products, visit
http://www.hp.com/go/overeasy

For information on HP Integrity servers, HP-UX 11i v3 and VSE, visit
http://www.hp.com/go/integrity
http://www.hp.com/go/hpux11i
http://www.hp.com/go/vse

## Revision History

| Revision | Date | Section | Change summary |
|---|---|---|---|
| 1.05 | Oct 30, 2010 | NA | Initial version |
| 1.06.00 | Feb 3, 2011 | 2.1 | Perl and SecureShell dependencies |
| | | 3.2 | HP 9000 Containers A.01.06 enhancements |
| | | 3.4 | Updates to Ignite-UX recovery process |
| | | 3.7 | Additional configuration steps |
| | | 4.3 | Internals of configuration |
| | | 5.8 | Configuring multiple IP addresses |
| | | 5.10.2 | Workaround for current Ignite-UX limitation |
| | | 8.2 | Triaging HP 9000 container access issues |
| 03.00 | Jul 29, 2011 | All | Update for major release A.03.00 |
| 03.01.01 | Dec 1, 2011 | 1.5 | What is new in HP 9000 Containers A.03.01? |
| | | 2.4 | New depot name in installation instructions |
| | | 3.5.7 | Using tape archives with recovery tool |
| | | 3.6.3 | New option for disabling disallowed command |
| | | 3.6.22 | Using enhanced tool for kernel tunable configuration |
| | | 4.1 | Migrating from HP 9000 Containers A.03.0x |
| | | 6.5.2 | New option for configuring global VxFS mounts |
| | | 6.10 | More details on container patching |
| | | 9.4 | Edits in patching limitations |
| | | 10.2 | More tips for troubleshooting SSH issues |
| | | 10.5 | More known issues and workarounds |
| 03.01.02 | June18, 2012 | 2.2 | Recommended patch set |
| | | 3,4,5 | Split the container creation steps for *system* and *classic* types |
| | | 3.4 | More information on kernel parameter setting |
| | | 4.5.12 | Configuring X-server |
| | | 4.5.13 | Configuring additional privileges |
| | | 4.5.14 | Configuring DDFA |
| | | 4.5.15 | Disabling AUTOFS |
| | | 4.5.16 | Configuring telnet for 10.xx containers |
| | | 4.7.2 | Configuring more stack size for applications |
| | | 8.10 | Updates on patching support |
| | | 12.2 | More on troubleshooting access issues |
| 03.01.03 | July 18. 2012 | 3.3 | Clarification on backup directories |
| 03.01.04 | Oct 17, 2012 | 1.3 | Trusted system containers enabled, SMH/SAM limitation |
| | | 2.2 | More recommended patches |
| | | 3.4 | Global host name limitation |
| | | 4.5.17 | Configuring OSI Transport Services |
| | | 4.5.18 | Enabling auditing |
| | | 8.12.5 | HP Data Protector workaroround |
| | | 8.13 | Auditing with HP 9000 containers |
| | | 12.5 | More information on known issues and workarounds |
| | | 12.9.4 | Profiling emulated applications |

|  |  |  |  |
|---|---|---|---|
| 03.01.04.01 | Oct 19, 2012 | 2.2 | Additional recommended patch |
|  |  | 4.5.9 | Configuring or disabling trusted mode |
|  |  | 4.5.12 | Configuring X server with graphics adapter |
|  |  | 6.2,8.3,12.2 | Changes to remove trusted mode limitation |
| 03.01.04.02 | Nov 18,2012 | 2.2 | Additional recommended patch |
|  |  | 8.12.5 | Added more details |
|  |  | 8.13 | Using hp9000_audit_global |
|  |  | 12.8.3 | Restoring restricted HP 9000 commands |

# 1. Introduction

## 1.1 Product Overview

HP 9000 Containers is a set of tools that enable quick transition of application environment from an HP 9000/PA-RISC server to HP Integrity server. It provides a way to *re-host* the complete HP 9000 user-space environment without the need to re-compile or re-install individual applications or re-construct application ecosystem, and with minimal re-configuration and application inventory preparation effort.

The transitioned applications will reside in a `chroot` environment (called the HP 9000-container) along with HP 9000 commands, libraries and other user space components. The HP 9000 container will have its own IP address and login credentials. An HP 9000 container can be started, stopped, modified, exported, imported and deleted. However, it cannot support applications that are kernel intrusive, system administration commands and system management related applications inside it.

HP 9000 Containers is built on two key HP-UX technologies – the HP ARIES dynamic binary translator, which provides the execution layer for PA-RISC applications, and HP-UX Containers (formerly Secure Resource Partitions – SRP) which enable multiple secure isolated execution environments on the same HP-UX operating system instance.

## 1.2 HP 9000 Containers support summary

| **Supports** | **Does not support** |
|---|---|
| Transitioning HP 9000/PA-RISC application environment to a `chroot` environment on HP Integrity server | Running HP 9000 HP-UX kernel inside the container |
| HP-UX 11i  (HP 9000) to HP-UX 11i v3 (Integrity) transition | HP 9000 environments prior to HP-UX 11i v1 |
| Creation of container environment from existing HP 9000 servers | Pre-populated HP 9000 components inside containers |
| Transitioning all application binaries and configuration files in one go | New tools for data migration |
| Usage in HP Integrity VM, HP-UX vPars, nPar | Complete HP 9000 platform virtualization |
| Executables inside container are emulated by the HP ARIES dynamic binary translator | Native mode or mixed mode execution inside containers |
| Container specific IP address and login credentials | System administration and resource monitoring tools and services |
| Life cycle management for container – start, stop, export. import, modify, delete | Online migration |
| Well-behaved, pure user space applications that perform no system management tasks | Kernel intrusive applications, device drivers, system management and monitoring related applications |
| Serviceguard integration, using modified packages, for high availability | Serviceguard inside containers |

## 1.3 HP 9000 container models

HP 9000 Containers A.03.0x supports two container models – a feature rich *system container* and a limited *classic container*. The key feature differences between the two models are summarized in the following table.

| **HP 9000 *system* container** | **HP 9000 *classic* container** |
|---|---|
| Support for *inetd* services – access to container via `telnet`, `ftp`, `rlogin`, `remsh`, `rexec` (no `telnet` yet for HP-UX 10.20) | No support for *inetd* services – access only through SSH based protocols |
| SSH based access only if SSH Is available in the HP 9000 image | SSH based access is supported, even if there is no SSH configured in the HP 9000 image |
| Support for patching inside the container (with some exceptions) | Support only for non-SD patching inside the container |
| Multiple HP 9000 *system* containers can be hosted on the same HP-UX instance | Only one *classic* container supported on an HP-UX instance |
| Can co-exist with native HP-UX containers | Cannot co-exist with native HP-UX containers |
| Private HP 9000 file system | Parts of the HP 9000 file system is shared with the host (mainly `/etc`, `/dev`, `/tcb and` parts of `/var`) |
| User management inside container | User management on the host system |
| Most commands report container relative information | Some commands report system wide information (not container specific) |
| Run level support inside container | Partial run-level support |
| Mount support inside container | No mount support inside container |
| Serviceguard integration support in either of two models – SRP package model and application package model | Serviceguard integration supported only in the application package model |
| No support for user accounting and quotas | Accounting and quotas can be enabled since user management is on host system |
| Trusted container support with some differences compared to native system | Trusted mode support similar to that on a native system (managed entirely from host) |
| No support for SMH/SAM to manage users | SMH/SAM can be used to manage users |
| Emulated login process | Native, low overhead login process |

## 1.4 HP 9000 Containers limitations

Refer to HP 9000 Containers Limitations for details on known limitations with using HP 9000 containers to re-host legacy environments.

## 1.5 HP 9000 server consolidation

There are a variety of options to consolidate HP 9000 servers on HP integrity servers using HP 9000 containers in conjunction with other products in the HP Virtualization Continuum. Some of the options are

a) Multiple HP 9000 *system* containers on bare-metal hardware
b) Using HP 9000 container(s) in HP Integrity VM guests
c) Using HP 9000 container(s) in HP-UX vPars

The choice should be based on the isolation, performance and cost requirements. Multiple HP 9000 containers can be used where

a) It is possible to use the HP 9000 *system container*.
b) There is no requirement for complete isolation of these environments. Multiple containers would share the same kernel.
c) No dynamic migration of resources (memory and CPU) is needed.
d) No online migration needed at container level.
e) There are no conflicting requirements for kernel tunable parameters.
f) No conflicting manageability requirements (since management applications must run on the host system in global compartment).
g) It is easy to co-ordinate application downtimes when the server needs a reboot.
h) There are enough resources (memory, CPU) to account for emulation overhead.
i) Some legacy 32-bit PA-RISC applications require the kernel tunable parameter `shmmax` to be less than `0x40000000`. This limits the number of applications that can be stacked together if they all need to use shared memory.
j) There are no more than 30000 concurrent processes on a system hosting legacy containers. Legacy commands and applications in containers may fail with large PIDs. Hence a system hosting such containers cannot have a `process_id_max` (and hence `nproc`) kernel tunable parameter value greater than 30000.

## 1.6 Sizing an HP 9000 container

There are some guidelines to size an HP 9000 container, accounting for the ARIES emulation overhead and also for the loss in performance because the applications were not compiled to take advantage of the Itanium processor architecture. Note that these guidelines are "common case" estimates and some tuning might be needed based on the results of testing.

a) ARIES may incur an average memory overhead of 10 MB / process. In order to compute the total requirement, find the number of processes that concurrently run at peak load on the HP 9000 server and account for an additional 10 MB for each of them. Note that even user sessions are emulated in the container, so take into account "all processes" that run on the HP 9000 server at peak load.

b) Note that HP-UX 11I v3 kernel itself has a larger memory foot print (more than 20%) compared to earlier versions. This needs to be accounted for as well.

c) The CPU requirements vary with the workload characteristics. Following types of workloads are known to have a higher requirement inside HP 9000 containers

   - Applications that spawn several short lived processes or threads
   - Applications that concurrently run thousands of CPU bound processes
   - Script intensive applications
   - Java based applications
   - Floating point intensive applications

d) The CPU overhead compared to a native version of the application can be anywhere between 30% and 100%.

e) A guideline, assuming the target server uses high end Tukwila (Itanium 9350N) cores, is to start with an approximate *1:<frequency in GHz on HP 9000 server>* core ratio. This means a 1:1 ratio if the HP 9000 server is using 1 GHz PA-RISC cores, for example.

f) Adjust the core count based on the results of the Proof of Concept testing.

## 1.7 Resource entitlement

HP-UX Containers supports the ability to allocate CPU and memory usage per container. By default, each container on the system is allocated a Process Resource Manager (PRM) group. Each PRM group can be assigned CPU and memory allocations. PRM provides two allocation models for CPU cores

- **Share based**: Restrictions (excluding maximum utilization caps) are not applied until the managed resource is fully utilized, at which point the operating system scheduler or memory manager applies an algorithm to allocate resources proportional to each PRM group's share size. This model ensures that individual containers can utilize available resources without frequent tuning of allocations.
- **Dedicated**: The specified PRM group is allocated a fixed quantity of the resource for its own exclusive use. This model guarantees immediate and complete access to the resource at the expense of the ability to allow other PRM groups access to the currently unused resource. Dedicated CPU allocation can be used to limit the software license requirements for some software products.

You can apply a combination of resource allocation models on a single server. You can also choose to disable the use of PRM, either to allow the use of a different resource allocation utility such as Workload Manager (WLM) or Global Workload Manager (gWLM), or to disable per container resource management.

For more details on PRM and WLM,
http://www.hp.com/go/hpux-core-docs

## 1.8 Where to use HP 9000 containers

HP 9000 Containers is recommended to be used where

- Upgrading or porting to native Integrity version of applications is infeasible.
- ISV application license agreements allow for copy over to a new platform or can be migrated.
- Either the ISV supports the application on ARIES, or ISV support is not a critical requirement for the customer.
- Applications intended to be transitioned are pure user-space and also not related to system administration or management.
- Traditional ARIES migration is deemed to be too costly because of one or more of the following reasons
  - There's not enough information about the application inventory (list of applications, executables, libraries, configuration files, dependencies) that reside on the HP 9000 server(s).
  - The number of servers targeted for migration is large and there are not enough resources to carry out individual application transition.
  - There's a dependency on legacy stand-alone development environments which are not supported by HP XPADE (http://www.hp.com/go/xpade).
- Where limitations described in HP 9000 Containers Limitations are acceptable.

## 1.9 Using ARIES without HP 9000 containers

It is possible to just use the stand-alone ARIES emulator to run applications compiled for PA-RISC. The process involves copying over just the application related files from the HP 9000 server to the HP Integrity system. The solution is supported with other HP virtualization solutions – HP Integrity VM, HP-UX vPars and nPar as well as with HP-UX *system* and *workload* native containers.

The following table compares the two approaches – standalone ARIES and HP 9000 Containers:

| Traditional ARIES migration | HP 9000 Containers |
|---|---|
| Need to identify and transfer application dependencies individually | All dependencies included in the HP 9000 file image that is used to create a container |
| No PA-RISC environment on the Integrity server except for system libraries and the applications | PA-RISC virtualized environment inside the container |
| Need to use a separate product called XPADE for PA-RISC C/C++ code development | PA-RISC development environment comes along with the HP 9000 file image |
| Direct installation and patching of applications may need some workarounds (for example, if the HP-UX version and platform info are being checked) | Installation and patching of applications need no workarounds |
| Non kernel intrusive system management applications can be run on ARIES | System management and resource monitoring related applications generally do not run inside the container |
| Better performance compared to containers. if applications are highly script intensive | May need to switch to using native shells and commands, in script intensive environments |
| Introduces no new manageability aspects, if used without containers | There are some additional management tasks related to containers |
| Needs almost no change in customer processes, if used without containers | Some of the processes currently being followed by customer may need changes |
| Needs no change to Serviceguard packages, if used without containers (except for any migration needed between SG versions) | Changes needed to Serviceguard packages to integrate with containers |

## 1.10 ISV software licensing and support

ISV product license and support issues need to be discussed with the respective application vendors directly. HP does not own issues related to software license (LTU/RTU) migration during application transition to a new platform. If application licensing policy explicitly prohibits copying to a new server, the customer is advised to apply for fresh licenses before they start using HP 9000 Containers.

For further details, refer to "**Support**" section on HP 9000 Containers web portal (http://www.hp.com/go/hp9000-containers)

## 1.11 Support policy

Refer to "**Support**" section on HP 9000 Containers web portal (http://www.hp.com/go/hp9000-containers)

# 2. Installation and Configuration

## 2.1 Pre-requisites

### 2.1.1 HP-UX 11i v3 March 2011 update or later

HP 9000 Containers utilize the *namespace virtualization* features that come with HP-UX 11I v3 1103 Base Operating Environment.

While installing the operating environment, it is recommended that `/var` be configured to be on a separate file system from the root file system.

Ensure that the right version is available before proceeding

```
$ swlist | grep OE
```

It is highly recommended that the system be used to host applications only inside containers. Do not install or use applications on the host outside containers. The exceptions are system management related applications (such as HP OpenView and HP Serviceguard), which are not supported inside containers.

### 2.1.2 HP-UX Containers A.03.01 or later

Download and install the most recent HP-UX Containers depot from https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=HP-UX-SRP

To install and verify the product, do

```
$ swinstall -x autoreboot=true -s <HP-UX Containers depot
  path> \*
$ swverify HP-UX-SRP
```

### 2.1.3 HP ARIES patch PHSS_41423 or later

Download and install the most recent HP ARIES patch for HP-UX 11i v3 from HP IT Resource Center (HP Support Center) http://itrc.hp.com

Check the patch level

```
$ what /usr/lib/hpux32/aries32.so
```

### 2.1.4 Perl v5.8.8 or later

Check the `perl` version on the system

```
$ perl -v
```

If it is below 5.8.8, get the latest version from https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=PERL

### 2.1.5 HP-UX SecureShell A.05.00.012 or later

Check the SecureShell version on the system

```
$ swlist | grep SecureShell
```

If it is below A.05.00.012, get the latest version from https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=T1471AA

## 2.2 Recommended patches

The following (or superseding) patches address some of the known issues related to containers and are recommended

- PHKL_41967 : 11.31 fs_select cumulative patch
- PHKL_42716 : 11.31 vfs_vnops cumulative patch
- PHNE_42017 : 11.31 cumulative ARPA Transport patch
- PHSS_42623 : 11.31 mksf(1M) cumulative patch
- PHSS_42739 : 11.31 Aries cumulative patch
- PHCO_43198 : 11.31 audcmnds cumulative patch

The latest version of the following bundles is also highly recommended. These can be downloaded from HP Software Depot

- FileSystem-SRP
- HPUXTransportSRP
- HPUX-Streams-SRP
- AuditExt

## 2.3 Additional requirements for HP 9000 *classic container*

Refer to HP 9000 container models to understand the differences between the two models, *system* and *classic*, supported with HP 9000 containers.

For HP 9000 *classic* container, the HP-UX OS instance must be dedicated for hosting a single HP 9000 container. There is no support for creating other containers, or running applications outside the container on the same OS instance. The only exceptions are system management related applications.

## 2.4 Configuring HP-UX Secure Resource Partitions

Login as root user and run

```
$ srp_sys -setup
```

You will need to enable PRM only if there is an intention to host multiple containers on the system and there is a need to partition resources (memory, CPU) among these containers. In particular, if the intention is to use an HP 9000 *classic* container, PRM can be disabled because the host system needs to be dedicated for a single container. Refer to HP 9000 container models for more information on the *classic* container.

You may choose to accept default values for all parameters or choose to customize. In particular, make sure that SSHD on host is configured to listen to the HP-UX 11i v3 host IP address. For more information on the configuration parameters refer to *HP-UX Containers A.03.01 Administrator's Guide*.

Reboot the server for the configuration to take effect.

## 2.5 Installing HP 9000 Containers

To install and verify HP 9000 Containers product, change to the directory where the depot resides and run

```
$ swinstall -s <HP9KContainers depot path> \*
$ swverify HP9KContainers
```

# 3. Preparing to transition from HP 9000 server

This chapter describes the steps that are to be followed to prepare for a transition from an HP 9000 server to an HP 9000 container on an HP Integrity server running HP-UX 11i v3.

## 3.1 Overview

The transition process from an HP 9000 server to an HP 9000 container typically involves the following sequence of activities.

- Decide on which HP 9000 container model to use
- Create the HP 9000 server file system image
- Transition kernel tunable parameters
- Decide on a name for the container
- Create file systems for the container
- Setup user environment for recovery
- Recover HP 9000 files on the HP Integrity server
- Create and configure an HP 9000 container
- Start the HP 9000 container and test applications
- Tweak ARIES configuration, if needed

**Note**: Data migration related issues needs to be addressed separately. HP 9000 Containers do not provide any new tools or documentation in that space.

## 3.2 Decide on the HP 9000 container model

Refer to HP 9000 container models to understand the differences between the two models, system and *classic*, supported with HP 9000 containers.

The recommendation is to use HP 9000 *system* container, except where

- There is a need to use SMH/SAM to manage users
- There is need for user quota/accounting
- There is need for a non-emulated login process

## 3.3 Creating the HP 9000 server image

### 3.3.1 Image creation process overview

Archive all directories from the HP 9000 server except possibly the NFS mounted ones. At the minimum include `/bin`, `/dev`, `/etc`, `/lib`, `/net`, `/opt`, `/sbin`, `/stand`, `/usr and /var` in the backup.

The image creation can be done using any utility that can eventually make the files visible under an alternate root directory and preserve file ownership and permissions. Some of the common tools that can be used are `pax` and `fbackup`. For HP 9000 *classic* container, `cpio` archives are not supported.

It may be possible to re-use existing images created by standard backup applications, if they can be recovered into an alternate path, either using the application itself or by manually running some commands. For example, see Recovering HP 9000 files on how to recover from HP Ignite-UX recovery archives. Again for *classic* container, `cpio` Ignite-UX archives are not supported.

Imaging of files can be done at a system level or at individual directory level. The choice is to be made with due diligence. Some of the considerations may include availability of storage space as well as memory and I/O overhead of the archival method.

It is recommended that application data be copied over all together to prevent inconsistencies. It is also recommended that all applications on the HP 9000 server be stopped before making the image. This prevents archival of transient files, which can cause unexpected behavior when applications are re-started inside the HP 9000 container.

### 3.3.2 Using `fbackup` for image creation of HP 9000 files

The following describes how to use `fbackup` for archiving all the required directories in a single session. If image creation is being done on a live production server, consider opting for multiple sessions to reduce memory and I/O overhead. See "`man fbackup`" for more detailed information.

- Prepare a graph file with the include-exclude list.

  For example, here are contents of a graph file for a system level backup

  ```
  i /
  e /var/adm/crash
  ```

  For a directory level backup, the graph file could b

  ```
  i /var
  e /var/adm/crash
  ```

  NFS mounted directories are excluded by default. It is not necessary to specify “e” for them explicitly in graph file. If they need to be included, then there is an option to use while running `fbackup`.

- Compute the space requirement for the archive.

  Login to the HP 9000 server and run “`du -sk`" on all the directories in the list to be included/excluded and do the computation. Also reduce sizes for any sub-directories that are NFS mounted, unless they are explicitly targeted for archival.

- Decide where `fbackup` is going to write the archive.

  It could be on a tape or a local/remote file. See “`man fbackup`” for more details. If output is to a file, ensure that `largefiles` is supported on the file system and there is enough free space to copy the archive.

  ```
  $ fsadm <file system root directory>
  $ df -k <file system root directory>
  ```

- Running `fbackup`

  If there is a need to "copy" NFS mounted directories to target server (as opposed to mounting them on the new system) then an additional option "`-n`" needs to be specified in the following command line.

  ```
  $ /usr/sbin/fbackup -0 \
    -f <output device or file path> \
    -g <graph file path> -I <index file> 2>&1 | \
    tee <fbackup log file>
  ```

  Check the log file for errors. Some temporary files like those in `/var/tmp` or `/var/spool/sockets` may appear in the log but can be ignored.

  *Do not use "`kill -9`" on `fbackup`.*

### 3.3.3 Using tar, cpio for image creation

When using tar or cpio ensure that the backup is done without including the "/" prefix. This is because the backup is intended to be restored under an alternate root, and not at the system root on the Integrity system. For example,

```
     $ cd /
     $ tar -cvf archive.tar bin dev etc lib net opt sbin stand
tmp usr var
```

Note that cpio is not supported for creating HP 9000 *classic* containers.

## 3.4 Transition kernel tunable parameters

Kernel parameters on the target system may need to be altered to accommodate the requirements of the HP 9000 server being migrated. This may also need a reboot.

- Login to the HP 9000 server and get the parameter values

```
$ kmtune > /tmp/tunables_hp9000.txt
```

- Transfer the output file to the target HP-UX 11i v3 server
- Login to the target server and run the kernel parameter configuration script

```
$ /opt/HP9000-Containers/bin/hp9000_conf_tunables \
  <HP 9000 kmtune file> <HP 9000 host name>
```

  In the batch mode, a set of selected tunable parameters will be updated automatically. In the interactive mode, the user can choose the list of tunable parameters to be updated based on the values on the PA-RISC server.

- The script does not guarantee that all required changes have been applied.

  a) It ignores parameters that are deemed to not have an impact on applications and are more related to system administration.
  b) It does not modify tunable parameters where there can be conflicts when moving from multiple HP 9000 servers into containers.
  c) It does not handle inter-tunable dependencies. Hence, some of the attempted changes may error out.
  d) It does not sum up the values of parameters when run multiple times with input from multiple HP 9000 servers (when creating multiple containers).

  Hence it is necessary that users review the data in the log file /var/opt/HP9000-Containers/logs/transition_tunables_<hostname>.log and make any further changes. There is a SUMMARY section at the end of log file.

- Pre HP-UX 11I v3 environments did not have support for large PID values. This means that certain commands and applications within such containers may fail if they encounter value larger than 30000.

```
$ kctune process_id_max=30000
$ kctune nproc=30000
```

- *Tunable base page* size is not supported with HP 9000 Containers. Ensure that the kernel tunable parameter base_pagesize is set to its default value of 4 KB

```
$ kctune base_pagesize=4
```

- If there are java or other heavily multi-threaded applications being migrated, it is recommended that parameter pa_maxssiz_32bit be increased to 128 MB

```
$ kctune pa_maxssiz_32bit=128MB
```

- There is a limitation currently that the global (host) cannot have a hostname longer than 8 characters if there are legacy HP 9000 containers being hosted on the system. The workaround is to disable overflow error checking using

```
$ kctune uname_eoverflow=0
```

## 3.5 Select a container name

The container name is also used as the `node name` and `hostname` for the container, by default. If the HP 9000 environment being migrated does not have support for long node names (as is the case with HP-UX 11i v1), it is recommended that the container name be less than or equal to 8 characters in length. If a longer container name is desired, the host name can be later edited to be different from container name, though.

*The container name is referred to as* `<srp_name>` *in the chapters to follow.*

## 3.6 Creating logical volumes for the container

- It is recommended that the container root directories be hosted in separate file systems. This is the only way to assign disk quotas to containers. By placing the home for each container in its own LUN, storage performance can be optimized.
- If the container is being created on the primary node of a Serviceguard cluster and the intention is to use the *container package* model, it is mandatory for the HP 9000 root directory to be a mount point.
- If the container root is not on a separate file system, for some reason, it is advised that the container does not reside on the Integrity host root or `/usr` file system.
- Create additional logical volumes and file systems for doing future mounts under the HP 9000 root directory as necessary (for `/var`, application data etc.)
- Provision for about 4 GB of additional space under container `/var` and about 4 GB under container `/usr` for internal use by HP 9000 Containers.

## 3.7 Installing driver and management software

Any special drivers that are needed have to be installed on the host HP-UX 11i v3 server.

Any applications that do system management related activities (such as resource monitoring) have to be installed on the host. For example, *HP GlancePlus* and *Performance Agent* have to be used from the host system.

## 3.8 Creating and configuring HP 9000 container

For steps to setup an HP 9000 *system* container (recommended choice) refer to Chapter 4 - Creating an HP 9000 *system* container

For steps to setup an HP 9000 *classic* container refer to Chapter 5 - Creating an HP 9000 *classic* container

# 4. Creating an HP 9000 *system* container

This chapter describes the set of steps that are to be followed to create and configure an HP 9000 *system* type container.

## 4.1 Setting up user environment for image recovery

- If `cpio`, `tar`, `fbackup or Ignite-UX` was used to create the image, there is no need to setup any user environment prior to recovery. HP 9000 Containers provides a tool to recover such images.
- If any other tool was used for creating the image, and the tool has an option to recover files purely based on numeric UID/GID, then no user environment needs to be setup before the recovery.
- If the tool used for creating the image, gives preference to user name and group name over UID and GID respectively, then the following needs to be done on the host system before the recovery. These steps imply that no users apart from `root` can login to the system while the recovery is going on.

  Take a backup of host user related files

  ```
  $ cp -p /etc/passwd /etc/passwd.backup
  $ cp -p /etc/group /etc/group.backup
  $ cp -p /etc/nsswitch.conf /etc/nsswitch.conf.backup
  ```

  Edit `/etc/nsswitch.conf` entry for users to include only `files`

  ```
  users   files
  ```

  Delete all entries from `/etc/group` file other than `root, other. bin. sys. adm, daemon`

  Delete all entries from `/etc/passwd` file on host other than `root. daemon, bin, sys, adm`

## 4.2 Create the container root directory

- Create root for HP 9000 system container as

  ```
  $ mkdir /var/hpsrp/<srp_name>
  ```

- Mount the file system created to host the container root, if needed

  ```
  $ mount -F <fstype> <from where> /var/hpsrp/<srp_name
  ```

- Set ownership and permissions

  ```
  $ chown root:sys /var/hpsrp/<srp_name>
  $ chmod 0755 /var/hpsrp/<srp_name>
  ```

*The container root directory is referred to as `<hp9000_root>` in the sections to follow.*

## 4.3 Recovering HP 9000 files

### 4.3.1 Configure mount points

If the files within the container need to be recovered onto mount points, create them on the HP-UX 11I v3 host. However, `/dev` and `/sbin` are not supported for use as mount points since this might cause the container creation process to fail.

For example,

```
$ mkdir <hp9000_root>/var
$ chown bin:bin <hp9000_root>/var
$ mount –F vxfs <from where> <hp9000_root>/var
```

### 4.3.2 Using Ignite-UX network recovery archive

It Ignite-UX network recovery archive exists for the HP 9000 server; it can be used to get the files replicated on the target server.  However, Ignite-UX cannot itself be used to do the recovery since it cannot restore to an alternate root directory.

Steps to recover from an Ignite-UX network recovery archive

- Identify the archive – by default, it will reside on the Ignite-UX server under `/var/opt/ignite/recovery/archives/<HP 9000-host-name>`
- Copy the archive file onto the Integrity server (or make it visible via a NFS-mount). Do not keep the archive in "/" directory on the system.
- Uncompress the archive
- Recover the image

```
$ /opt/HP9000-Containers/bin/hp9000_recover_image \
  <hp9000_root> <image-file>
```

- Ignore any errors related to recovery of `/dev` directory in the log file.

### 4.3.3 Using Ignite-UX tape recovery archive

It Ignite-UX tape recovery archive exists for the HP 9000 server; it can be used to get the files replicated on the target server. However Ignite-UX itself cannot be used to do the image recovery since it cannot restore to an alternate root directory.

Steps to recover from an Ignite-UX tape recovery archive

- Insert the tape into a compatible drive
- Extract the archive into file system

```
$ copy_boot_tape -u /dev/rmt/0mn -d <directory>
```

- Identify the file in the extract that corresponds to the file system image. This will typically be the largest file in the extract. For HP-UX 11i v1, it is usually named `file0002`.
- Copy the archive file onto the Integrity server (or make it visible via a NFS-mount). Do not keep the archive in "/" directory on the system.
- Recover the archive

```
$ /opt/HP9000-Containers/bin/hp9000_recover_image \
  <hp9000_root> <image-file>
```

- Ignore any errors related to recovery of `/dev` directory in the log file.

### 4.3.4 Using cpio, tar, frecover for recovering HP 9000 files

- If it is a file archive, copy it to the HP-UX 11i v3 server (or make it visible via NFS- mount). Do not keep the archive in "/" directory on the system. Recover the archive using

```
$ /opt/HP9000-Containers/bin/hp9000_recover_image \
  <hp9000_root> <image-file>
```

- If it is a tape archive, insert the tape on the Integrity server and present it to the HP-UX 11i v3 system as, for example, /dev/rtape/tape1_BEST. Recover the archive using

```
$ /opt/HP9000-Containers/bin/hp9000_recover_image \
  <hp9000_root> </dev/rtape/tape1_BEST>
```

- Ignore any errors related to recovery of /dev directory in the log file.

### 4.3.5 Using other tools for recovery

If third party tools have been used for recovery, ensure that proper permissions and ownership (UID/GID) are preserved. Some tools are known not to preserve setuid/ setgid bits. Check /usr/sbin/sendmail for an instance of the setuid case.

### 4.3.6 Post recovery steps

After the recovery is complete,

- Manually check if all the basic directories (/etc. /opt, /sbin, /usr, /var) have been recovered properly.
- Directories that have not been copied over need to be created manually and assigned proper ownership and permissions. For example,

```
$ mkdir <hp9000_root>/var/adm/crash
$ chmod 0755 <hp9000_root>/var/adm/crash
$ chown root:root <hp9000_root>/var/adm/crash
```

- When using tools other than cpio, tar, and fbackup restore the user related files back (if they were modified prior to recovery)

```
$ cp -p /etc/passwd.backup /etc/passwd
$ cp -p /etc/group.backup /etc/group
$ cp -p /etc/nsswitch.conf.backup /etc/nsswitch.conf
```

## 4.4 Creating the HP 9000 *system* container

### 4.4.1 Preparing PRM configuration

If PRM is being employed to allocate resources between multiple containers, decide on whether FSS (fair share scheduler) or PSET (processor set) is to be used with cores. FSS provides better sizing and more flexibility, but is known to have conflicts when application internally does resource management (such as when using database resource managers). With PSETs the cores are reserved even when the container is down.

For FSS, the percentage entitlement is calculated as

$$\frac{\text{Number of shares assigned to a particular PRM Group}}{\text{Sum of the shares assigned to all PRM Groups}} \times 100$$

### 4.4.2 Adding `hp9000sys` template

To create an HP 9000 *system* container, add the `hp9000sys` template

```
$ srp -add <srp_name> -t hp9000sys
```

### 4.4.3 Configuration parameters

- Auto start setting

  This controls whether the container needs to be started at the time of server boot (through a RC script).

  If this configuration is being done on the primary node of a Serviceguard cluster and the intention is to use the *container package* model, answer the *autostart* question in the negative.

- Root user configuration

  A *system* container has a root user different from that of the HP-UX 11i v3 host system with fewer privileges. Hence the password is prompted for.

- DNS configuration

  By default, the DNS configuration for an HP 9000 *system* container is picked up from inside the HP 9000 image. It can be changed if needed, by specifying new values for the related parameters when promoted.

- Disallowed commands configuration

  HP 9000 Containers A.03.01 provides two options for disabling system administration related commands inside the container

  a) Use compartment rules to restrict commands
  b) Overwrite the disallowed commands with a dummy executable that prints an error message and exits (default)

  For HP-UX 10.xx containers, choose (a) by overriding the default

  ```
  Use rules to restrict unsupported commands [no] yes
  ```

  For HP-UX 11.xx containers, choose the default option to begin with. It may be modified later after reviewing the details presented in the HP 9000 container patching section.

- PRM configuration

  `CPU Entitlement (shares):`
  Minimum share of the CPU when the system is at peak load

  `Max CPU Usage (%):`
  Maximum percentage of the CPU that can be used by the container

  `Memory Entitlement (shares):`
  Minimum share of the private real memory when the system is at peak load

  `Max Memory (%):`
  Maximum percentage of the private real memory that can be allocated for the container from the available system's memory for user processes

  `Shared Memory (megabytes):`
  The minimum size of the real memory in megabytes allocated to the container for shared memory usage.

- Network parameters

  A static IP address is essential for the container. DHCP is not currently supported.

  Ensure that IP address, LAN interface, gateway IP and subnet mask have been configured appropriately. The LAN interface can be either private to the container or shared. The network configuration is actually performed on the host system (not inside the container).

If the HP 9000 server was using IPv4 address, the same is recommended for the HP 9000 container since the environment may not have complete IPv6 support.

If this configuration is being done on one of the nodes of a Serviceguard cluster and of the container IP address is to be managed by Serviceguard, then answer the following in negative

```
Add IP address to netconf file? [yes] no
```

The container creation can take up to 30 minutes to complete.

### 4.4.4 Error messages

If you receive an error message like "`Could not generate swlist`"

- Check if `/var/adm/sw` directory exists under `<hp9000_root>.` This is necessary to proceed.
- If so, check if `/var/adm/sw` or directories under it such as `/var/adm/sw /products` is a link to some other directory in the image, Suppose, for example, the link is to a directory `/softdepot`.
- Create a link (with same path) on the host system to point to this directory. For the above example,

```
$ ln –s /<hp9000_root>/softdepot /softdepot
```

If there is a warning "`Could not find HP 9000 HP-UX version`", do the OS version configuration manually, once `srp –add` is complete, as follows

- Open the file `<hp9000_root>/.ariesrc`
- Specify HP 9000 HP-UX version for `pa_os_ver` parameter. For example,

```
/    -pa_os_ver B.10.20
```

- Specify the same in `<hp9000_root>/.aries64rc`

### 4.4.5 Verification

```
$ srp –list <srp_name> -v | more
```

### 4.4.6 Changing configuration parameters (if needed)

```
$ srp –replace <srp_name>
```

### 4.4.7 Reverting configuration (if needed)

If there were any errors during configuration, the partial container can be deleted using

```
$ srp –delete <srp_name> delete_changes_ok=yes
```

## 4.5 Additional container configuration

### 4.5.1 Configuring host name/node name

By default, the container name is also used as the node name and host name for the HP 9000 container. Refer to Modifying hostname on details of how to modify this configuration.

A couple of cases where the host name may need to be changed

- Legacy HP 9000 environments may not support long names. If the container name is larger than 8 characters in length, the container must be given a different host name/node name.
- If Serviceguard is intended to be used with the HP 9000 container in the *application package* model, the HP 9000 container on each node will have a different IP address (but same compartment name) and hence it is advised to configure different host names on each node.

Configure applications inside the HP 9000 container with the host name. Typically, this involves editing configuration files but some applications store this name in databases or in internal formats (in which case application documentation needs to be referred to).

### 4.5.2 Configuring IP address

Configure applications inside the HP 9000 container to listen to the SRP's own IP address. Some applications store the IP address in the databases – refer to application documentation on how to do re-configuration.

If application license depends on IP address and it cannot be migrated, or if application re-configuration for IP address is too complex, use the HP 9000 system IP address for the HP 9000 container.

Refer to Modifying IP address configuration for details on how to modify configuration.

### 4.5.3 Configuring additional IP addresses

Applications inside an HP 9000 container may need to use multiple IP addresses. Analyze the configuration in `<hp9000_root>/var/opt/HP9000-Containers/etc/rc.config.d/netconf` to find the number of configured IP addresses on the HP 9000 server.

If Serviceguard is intended to be used with the HP 9000 container in the *application package* model, applications may need to use an additional floating IP address which will be managed by Serviceguard

Refer to Modifying IP address configuration for details on how to do this configuration.

### 4.5.4 Configuring additional devices

There is no support for creating and managing device files inside the container. The devices have to be provisioned on the HP-UX 11i v3 host system.

To make a device visible inside using

```
$ srp -add <srp_name> -tune device=<device_path>
```

For example, this can be used to provision raw devices used by database applications to the container. The data from raw devices has to be backed up and restored separately using standard tools such as `dd`.The raw device names may also have to be configured in application specific files.

The list of device files copied from the host into the container is recorded in

```
/var/hpsrp/<srp_name>.setup/srpdevices.lst
```

### 4.5.5 Configuring mount points

Refer to Configuring mount and export points on how to configure NFS, AUTOFS and mount points for the HP 9000 container. It is recommended that mounts inside the container root directory be configured in either container local or container pre-start fstab (and not in global fstab)

### 4.5.6 Restoring or deleting HP 9000 startup services

As part of HP 9000 container setup several daemons are deleted from the HP 9000 RC directories. The heuristic used is that all services that appear in HP 9000 swlist, except the ones that are supported inside the container, are moved out of <hp9000_root>/sbin/init.d. If there are applications in the container which were installed using Software Distributor, related daemons may also have got removed.

To restore a deleted service use the script

```
$ /opt/HP9000-Containers/bin/hp9000_restore_service
```

The tool will query the user for the container name and the name of the RC script which has been moved to /sbin-hp9000/init.d inside container. The script will also update the record /var/opt/HP9000-Containers/deleted_services.

Similarly, to delete a service from the container run

```
$ /opt/HP9000-Containers/bin/hp9000_remove_service
```

### 4.5.7 Configuring DCE services

Run tool to view or configure DCE

```
$ /opt/HP9000-Containers/bin/hp9000_dce_setup <srp_name>
```

If the tool finds DCE server configuration and if there are DCE clients on other servers that point to this server, the host name and IP address of the HP 9000 container need to be changed to the values on the HP 9000 server to avoid re-configuring all clients.

If the tool finds DCE client configuration, and if the HP 9000 container is using a different hostname/IP address from those on the HP 9000 server, add this new client to the DCE server using dce_config. If this is not feasible, then the only option is to re-use the hostname and IP address from the HP 9000 server.

Edit IP address configuration in /etc/opt/security/pe_site, if needed.

### 4.5.8 Configuring root cron jobs

As part of HP 9000 container creation, all cron jobs configured by root are moved out because they may contain system administration related jobs which may not be supported inside the container. If any of these jobs need to be run in the HP 9000 container, it can be re-configured using crontab command or by restoring entries from the backup file <hp9000_root>/var/opt/HP9000-Containers/var/spool/cron/crontabs/root.

### 4.5.9 Configuring or disabling trusted mode features

Trusted mode support with HP 9000 *system* containers has the following differences compared to a native system

(a) Audit management has to mostly happen from global and there are some known limitations. Steps to configure auditing are described in section 4.5.18.

(b) There is no SMH/SAM available inside the container to manage trusted mode
(c) Some features such as inability to reset a lost root password are not applicable

There are no additional steps needed to configure trusted mode other than what is required for auditing.

In case, there is a need to convert from trusted mode to standard mode, use

```
$ srp -start <srp_name>
$ srp_su <srp_name> root -c "/usr/lbin/tsconvert -r"
$ srp -stop <srp_name>
```

### 4.5.10 Configuring inittab

- Check `<hp9000_root>/var/opt/HP9000-Containers/etc/inittab` to see the configuration present on the HP 9000 server.
- Open the container `inittab` file `/var/hpsrp/<srp_name>/etc/inittab`.
- Check if application specific configurations have been retained. If not there, pull in the entries to container `inittab`.

### 4.5.11 Configuring printers

- Configure printer devices on the host HP-UX 11i v3 system.
- Store the configuration from the HP 9000 server and restore it into the HP-UX 11i v3 server using `lpmgr` as follows.

  On the HP 9000 server, run

  ```
  $ mkdir /tmp/lpsave
  $ /usr/sam/lbin/lpmgr -S -xsavedir=/tmp/lpsave
  ```

  Transfer `/tmp/lpsave` directory to Integrity host HP-UX 11i v3 system and run

  ```
  $ /usr/sam/lbin/lpmgr -R -xsavedir=<dir>
  ```

### 4.5.12 Configuring X server

X server with graphics adapter is not supported inside HP 9000 containers. However, X Virtual Frame Buffer is supported inside HP 9000 `system` containers. To use `XVfb`,

- Open the display screens file, for example `/etc/X11/X0screens`
- Comment out any line containing `/dev/crt`
- Add configuration
  ```
  ServerOptions
     ServerMode XVfb
  ```
- Open `/etc/dt/config/Xservers`
- Comment out line containing `/usr/bin/X11/X :0` or modify it to
  ```
     /usr/bin/X11/Xvfb :0 -fbdir /tmp
  ```

If there is only one container on the host system, and if no X-server is required to be running on the host, it may be possible to configure X server with graphics adapter using the following steps.

- Open `/etc/cmpt/<srp_name>.rules` and insert a line just before the first line containing `#include`
  ```
  # define ALLOW_RDEVOPS
  ```
- Run command to reset compartment rules
  ```
  $ setrules
  ```
- Enable graphics module on host system and reboot (if not already loaded). For example,
  ```
  $ kcmodule gvid_core=best gvid=best
  $ reboot
  ```

- Verify that module is loaded

```
    $ kcmodule | grep gvid
```

- Copy over graphics devices into the container

```
    $ srp -add <srp_name> -tune device=/dev/gvid
    $ srp -add <srp_name> -tune device=/dev/gvid0
    $ srp -add <srp_name> -tune device=/dev/gvid_info
```

- Copy over input devices into the container. For example,

```
    $ srp -add <srp_name> -tune device=/dev/hid
```

- Change `/var/hpsrp/<srp_name>/etc/X11/XF86Config` to reflect new devices. For example,

```
            Option     "Device"   "/dev/hid/hid_000"
```

### 4.5.13 Configuring additional privileges

The `setprivgrp` command is not currently supported inside an HP 9000 container. Hence privileges such as RTPRIO and MLOCK cannot be granted using configuration in `/etc/privgroup` inside the container. A workaround is to use the command and the configuration file from global after copying the group name and GID to host `/etc/group`.

**`Note:`** The above configuration will get applied to groups with same GIDs in other *system* containers on the host as well. Hence this is not recommended where there are multiple containers on the system unless it can be ensured that a unique GID will be used (across the system) for groups which need the privilege.

### 4.5.14 Configuring DDFA

If Data Communications and Terminal Controller Device File Access (`ddfa`) is needed inside container, open `/etc/cmpt/<srp_name>.rules` file and insert a line as follows just before the first line containing `#include`

```
    #define ALLOW_MKNOD
```

**`Note:`** This will enable MKNOD privilege inside the container but do ensure that `mknod` is not used for any other purpose other than by ddfa itself. Using `mknod` inside container, in general, unsupported and can result in undefined system state.

### 4.5.15 Disabling AUTOFS

If AUTOFS is not being used, it is highly recommended that the service be disabled to save container startup/shutdown time.

- Open the file `/etc/rc.config.d/nfsconf`
- Set `AUTOFS=0`.

### 4.5.16 Configuring `telnet` for HP-UX 10.xx containers

It is known that HP-UX 10.xx version of `telnetd` is incompatible with ARIES emulation on HP-UX 11I v3. The workaround is to copy the following files from an HP-UX 11i v1 or (HP-UX 11.00) system into the 10.xx container.

```
    /usr/lbin/telnetd
    /usr/lib/libc.2
    /usr/lib/libsis.1
    /etc/inetsvcs.conf
```

Also, create a link as follows inside the container

```
    $ ln -s /usr/lib/libsis.1 /usr/lib/libsis.sl
```

### 4.5.17 Configuring OSI Transport Services

If OSI Transport Services (OTS) is in use on the HP 9000 server, download the version for Integrity HP-UX 11.31 and install it on the host system (global). Copy the related devices into the container

```
$ srp -add srpname -tune device=/dev/osotipi
$ srp -add srpname -tune device=/de/otsop
```

### 4.5.18 Enabling auditing

Auditing is not supported from within an HP 9000 *system* container, but it is possible to enable auditing in global and filter records at a container granularity. It is also possible to select users for auditing from within the container.

To enable auditing in global use `audsys` (1M) command like for example

```
$ audsys -n -c /var/adm/audit_trail -N 1
```

There is a known issue when using `-N` value greater than 1 with audsys command. Check for availability of patch PHCO_43198 if `N>1` is a requirement.

To select events or system calls for auditing, use `audevent (1M)` command. The list of selected system calls is not automatically migrated from the PA-RISC file system image to global - this is a manual process.

The user selection for auditing is retained inside the container file system, so there is no additional step required. If a change is needed to the user settings, the same can be performed inside the container using `audusr` (with trusted mode) or `userdbset` (with SMSE) command. SMH/SAM is not supported inside an HP 9000 container.

Refer to Auditing with HP 9000 Containers for details on how to filter and view audit records for a container and for the list of known auditing limitations.

## 4.6 Testing the HP 9000 container

If this configuration is being done on the primary node of a Serviceguard cluster and the intention is to use the *SRP package model*, refer to Using SRP package model for details on how to start the HP 9000 container for testing.

Otherwise, start the container

```
$ srp -start <srp_name>
```

All startup messages should say `[OK]`. Look for any startup error messages in `/var/hpsrp/<srp_name>/etc/rc.log`.

Check the status of the container

```
$ srp -status <srp_name> -v
```

Login to container from host

```
$ srp_su <srp_name>
```

Also, attempt to login using `telnet` and `ssh`.

Start applications using the normal procedures and perform exhaustive functional and performance testing to ensure compatibility.

To stop the container,

```
$ srp -stop <srp_name>
```

## 4.7 Workarounds for known issues

Check the list of known issues and workarounds to see if any of them are applicable to the environment being migrated.

## 4.8 Tweaking ARIES configuration

### 4.8.1 Configuring for more threads

The number of threads that a 32-bit application can spawn under ARIES is limited by the value of kernel tunable parameter `pa_maxssiz_32bit`. With default value, `85` threads can be spawned. For every additional thread, the value needs to be `increased` by 215 KB. For example, if an application needs `300` threads, `pa_maxssiz_32bit` needs to be increased by `(300-85)*215*1024` bytes.

If the number of threads needed by the application is unknown, use a value of 128 MB which will suffice for more than 300 threads.

```
$ kctune pa_maxssiz_32bit=128MB
```

In addition to tuning the kernel tunable parameter, ARIES needs to be configured to support more threads. Add the following to `/.ariesrc` file

```
# start config for more threads
<executable path name>  -mem_tune heap_max
# end config for more threads
```

### 4.8.2 Configuring for more stack size

The main thread stack for emulated applications is allocated by ARIES and not by the kernel. Hence it is not possible to modify this using `ulimit -s`. The default stack size is 8 MB for 32-bit applications. To increase it to, say, 16 MB increase pa_maxssiz_32bit by 8 MB and configure –ssz parameter in `/.ariesrc` for the executable like

```
# start configuration for 16 MB stack
<executable path name>  -ssz 16384
# end configuration for 16 MB stack
```

### 4.8.3 Configuring machine related parameters

If applications need to match certain specific machine parameters from the HP 9000 server, ARIES (`PHSS_41423` or a later patch) provides the following configuration options (can be added to existing `/.ariesrc` and `/.aries64rc` files)

```
-machine_id        <uname -i on HP 9000 server>
-machine_ident     <getconf MACHINE_IDENT on HP 9000 server>
-machine_serial    <getconf MACHINE_SERIAL on HP 9000 server>
-partition_ident   <getconf PARTITION_IDENT on HP 9000 server>
```

```
# start config for machine specific parameters
<executable path>  <ARIES option> <HP 9000 server value>
# end config for machine specific parameters
```

To apply this configuration to all executables, use wildcard `/` in place of executable path name. For example,

```
/  -machine_id  1338625371  -machine_ident  Z3e123a334fc9cd5b  -
machine_serial SGH4632J0F -partition_ident Z3e123a334fc9cd5b
```

These parameters take effect only when ARIES option "`-pa_os_cpu`" is also specified in the ARIES RC (`/.ariesrc` or `/.aries64rc`) file. This option is automatically set in these files when an HP 9000 container is created.

**NOTE**: The ARIES options described here should be used only where legally permitted. For example, if the configuration is being used to enable re-use of an application license, approval from respective vendor may be required.

# 5. Creating an HP 9000 *classic* container

This chapter describes the set of steps that are to be followed to create and configure an HP 9000 *classic* type container.

## 5.1 Setting up user environment for image recovery

A *classic* container shares /etc directory and login mechanism with the HP-UX 11I v3 host system. Hence, HP 9000 users and groups need to be merged into the host before doing the recovery.

- Recover HP 9000 /etc directory

  The input for the user migration process is a copy of the /etc directory from the HP 9000 server. Get a tar archive of /etc and recover it under /tmp on the HP Integrity server. It may also be possible to recover /etc from the image. For example, here is how to extract /etc from a complete fbackup image

  ```
  $ mkdir /tmp/HP9000
  $ echo "i etc" > /tmp/HP9000/graph
  $ cd /tmp/HP9000
  $ frecover -x -X -f <image file> -g /tmp/HP9000/graph
  ```

- System configuration

  Enable trusted mode on HP Integrity host using SMH, if HP 9000 server was configured with trusted mode.

  Enable shadow mode on HP Integrity host using pwconv command, if HP 9000 server was configured with shadow password.

- User and group migration

  Run the user merge tool as

  ```
  $ /opt/HP9000-Containers/bin/hp9000_conf_users \
    <path to recovered /etc directory>
  ```

  Check for errors or warnings on stderr and in the log file /var/opt/HP9000-Containers/logs/user_config.log

- Install and configure user management related products on the host

  The SSH login process to a *classic* container is actually native (does not use products from the HP 9000 image). It is towards the end of the login process that SSHD does a chroot into the HP 9000 file system and invokes a PA-RISC shell. Hence, if the requirement is to use NIS, LDAP or any other Active Directory tool, the same needs to be installed and configured on the host system.

## 5.2 Create the container root directory

The root directory has to be created under / on the host

- Create root directory, for example as

  ```
  $ mkdir /hp9000-root
  ```

- Mount the file system created to host the container root

  ```
  $ mount -F <fstype> <from where> </hp9000>
  ```

- Set ownership and permissions

  ```
  $ chown root:sys  /hp9000-root

  $ chmod 0755 /hp9000-roo
  ```

## 5.3 Recovering HP 9000 image

### 5.3.1 Configure mount points inside the container root

If the files within the container need to be recovered onto mount points, create them on the HP-UX 11I v3 host. For example,

```
$ mkdir <hp9000_root>/var
$ chown bin:bin <hp9000_root>/var
$ mount –F vxfs <from where> <hp9000_root>/var.
```

### 5.3.2 Using Ignite-UX network recovery archive

It Ignite-UX network recovery archive exists for the HP 9000 server; it can be used to get the files replicated on the target server. However, Ignite-UX cannot itself be used to do the recovery since it cannot restore to an alternate root directory.

Steps to recover from an Ignite-UX network recovery archive

- Identify the archive – by default, it will reside on the Ignite-UX server under `/var/opt/ignite/recovery/archives/<HP 9000-host-name>`
- Copy the archive file onto the Integrity server (or make it visible via a NFS-mount). Do not keep the archive in "/" directory on the system.
- Uncompress the archive
- Recover the image

  ```
  $ /opt/HP9000-Containers/bin/hp9000_recover_image \
    <hp9000_root> <image-file>
  ```

- Ignore any errors related to recovery of `/dev` directory in the log file.

### 5.3.3 Using Ignite-UX tape recovery archive

It Ignite-UX tape recovery archive exists for the HP 9000 server; it can be used to get the files replicated on the target server. However Ignite-UX itself cannot be used to do the recovery since it cannot restore to an alternate root directory.

Steps to recover from an Ignite-UX tape recovery archive

- Insert the tape into a compatible drive
- Extract the archive into file system

  ```
  $ copy_boot_tape -u /dev/rmt/0mn -d <directory>
  ```

- Identify the file in the extract that corresponds to the file system image. This will typically be the largest file in the extract. For HP-UX 11i v1, it is usually named `file0002`.
- Copy the archive file onto the Integrity server (or make it visible via a NFS-mount). Do not keep the archive in "/" directory on the system.
- Recover the archive

  ```
  $ /opt/HP9000-Containers/bin/hp9000_recover_image \
    <hp9000_root> <image-file>
  ```

- Ignore any errors related to recovery of `/dev` directory in the log file.

### 5.3.4 Using tar, frecover for recovering HP 9000 files

- If it is a file archive, copy it to the HP-UX 11i v3 server (or make it visible via a NFS-mount). Do not keep the archive in "/" directory on the system. Recover the archive

```
$ /opt/HP9000-Containers/bin/hp9000_recover_image \
  <hp9000_root> <image-file>
```

- If it is a tape archive, insert the tape on the Integrity server and present it to the HP-UX 11i v3 system as, for example, /dev/rtape/tape1_BEST. Recover the archive using

```
$ /opt/HP9000-Containers/bin/hp9000_recover_image \
  <hp9000_root> </dev/rtape/tape1_BEST>
```

- You may ignore any errors related to recovery of /dev directory in the log file.

### 5.3.5 Using other tools for recovery

If third party tools have been used for recovery, ensure that proper permissions and ownership are preserved. Some tools are known not to preserve setuid/setgid bits. Check /usr/sbin/sendmail for an instance of the setuid case.

### 5.3.6 Post recovery steps

After the recovery is complete,

- Manually check if all the basic directories (/etc. /home, /opt, /tmp, /usr, /var, /stand) have been recovered properly.
- Directories that have not been copied over need to be created manually and assigned proper ownership and permissions. For example,

```
$ mkdir <hp9000_root>/var/adm/crash
$ chmod 0755 <hp9000_root>/var/adm/crash
```

## 5.4 Creating the HP 9000 *classic* container

To create the container, add the hp9000cl template

```
$ srp -add <srp_name> -t hp9000cl
```

Configuration parameters

- Auto start setting

  This controls whether the container needs to be started at the time of server boot (through a RC script).

- HP 9000 root directory

  Specify the root path <hp9000_root> where the HP 9000 files have been recovered.

- Network parameters

  A static IP address is essential for the container. DHCP is not currently supported.

  Ensure that IP address, LAN interface, gateway IP and subnet mask have been configured appropriately. The LAN interface can be either private to the container or shared. The network configuration is actually performed on the host system (not inside the container).

If the HP 9000 server was using IPv4 address, the same is recommended for the HP 9000 container since the environment may not have complete IPv6 support.

The container creation can take up to 30 minutes.

To list configuration

```
$ srp -list <srp_name> -v | more
```

To change configuration, if needed

```
$ srp -replace <srp_name>
```

To revert configuration, if needed

```
$ srp -delete <srp_name> delete_changes_ok=y
```

## 5.5 Additional container configuration

### 5.5.1 Configuring host name/node name

By default, the container name is also used as the node name and host name for the HP 9000 container. Refer to Modifying hostname on details of how to modify this configuration. Note that legacy HP 9000 environments may not support long names. If the container name is larger than 8 characters in length, the container must be given a different host name/node name.

Configure applications inside the HP 9000 container with the host name. Typically, this involves editing configuration files but some applications store this name in databases or in internal formats (in which case application documentation needs to be referred to).

### 5.5.2 Configuring IP address

Configure applications inside the HP 9000 container to listen to the SRP's own IP address. Some applications store the IP address in the databases – refer to application documentation on how to do re-configuration.

If application license depends on IP address and it cannot be migrated, or if application re-configuration for IP address is too complex, use the HP 9000 system IP address for the HP 9000 container.

Refer to Modifying IP address configuration for details on how to modify configuration.

### 5.5.3 Configuring additional IP addresses

Applications inside an HP 9000 container may need to use multiple IP addresses. Analyze the configuration `<hp9000_root>/etc-hp9000 /rc.config.d/netconf` to find the number of configured IP addresses on the HP 9000 server.

If Serviceguard is intended to be used with the HP 9000 container in the *application package* model, applications may need to use an additional floating IP address which will be managed by Serviceguard

Refer to Modifying IP address configuration

### 5.5.4 Configuring mount points

Refer to Configuring mount and export points on how to configure NFS, AUTOFS and mount points for the HP 9000 container. Mount points have to be configured in global `fstab`.

### 5.5.5 Restoring HP 9000 startup service

As part of HP 9000 container setup several daemons are deleted from the HP 9000 RC directories. The heuristic used is that all services that appear in HP 9000 `swlist`, except for those that are supported inside the container, are moved out of `<hp9000_root> /sbin/init.d`. If there are applications in the container which were installed using Software Distributor, related daemons may also have got removed.

A backup copy of the RC scripts can be found in `/sbin-hp9000/init.d` which can be restored manually.

### 5.5.6 Configuring root `cron` jobs

As part of HP 9000 container creation, all `cron` jobs configured by root are moved out because they may contain system administration related jobs which may not be supported inside the container. If any of these jobs need to be run in the HP 9000 container, it can be re-configured using `crontab` command or by restoring entries from the backup file `<hp9000_root>/var/opt/HP9000-Containers/var/spool/cron/crontabs/root`.

### 5.5.7 Configuring `inittab`

- Check `<hp9000_root>/var/opt/HP9000-Containers/etc/inittab` to see the configuration present on the HP 9000 server.
- Open the container `inittab` file `/var/hpsrp/<srp_name>/etc/inittab`.
- Copy each application related entry into the container `inittab`, but with one modification - the fourth field which contains path of the executable should be prefixed with a `chroot <hp9000_root>`. For example,

```
appdaemon:3456:respawn:chroot /hp9000 /opt/app/bin/appd
```

### 5.5.8 Configuring printers

- Configure printer devices on the host HP-UX 11i v3 system.
- Store the configuration from the HP 9000 server and restore it into the HP-UX 11i v3 server using `lpmgr` as follows.

  On the HP 9000 server, run

```
$ mkdir /tmp/lpsave
$ /usr/sam/lbin/lpmgr -S -xsavedir=/tmp/lpsave
```

  Transfer `/tmp/lpsave` directory to Integrity host HP-UX 11i v3 system and run

```
$ /usr/sam/lbin/lpmgr -R -xsavedir=<dir>
```

### 5.5.9 Configuring non-local users

The HP 9000 local users (from `/etc/passwd`) are added to a container login group automatically as part of addition of the `hp9000cl` template and this group is allowed access to the container using RBAC. If non local users have to be given access to the container, follow instructions in User account management for the same.

### 5.5.10 Configuring trusted users

Check `/tcb/files/auth` on the HP-UX 11i v3 system and see if the trusted mode users have been merged from `<hp9000_root>/tcb`

Check the file `/var/opt/HP9000-Containers/logs/migrated_users` and see if any of the UIDs have changed as part of the user merge. If there are UIDs that have changed, the same need to be reflected in `/tcb/files/auth.` For example, if the log file reports that UID of "user1" has changed, edit the `u_id` field in `/tcb/files /auth/u/user1`

Enable auditing via SMH, if needed. The audit IDs will be automatically picked up from the files in `/tcb/files/auth.`

### 5.5.11 Configuring `sendmail`

The `sendmail` daemon runs on the host HP-UX 11i v3 system (not inside the HP 9000 `classic` container) and `/etc` is shared with host. Copy any specific configuration from HP 9000 `/etc/mail/sendmail.cf`, `/etc/mail/aliases` etc to HP-UX 11i v3 `/etc/mail`

### 5.5.12 Configuring `xinetd`

A workaround for the unavailability of `inetd` services inside a `classic` model HP 9000 container is to use `xinetd`. HP has not extensively tested this configuration and hence there may be certain limitations in using it inside container. To configure `xinetd` and setup RC scripts, follow the steps below

- Stop and delete the existing container

  ```
  $ srp -stop <srp_name>
  
  $ srp -delete <srp_name> delete_changes_ok=y -b
  ```

- Install xinetd on the HP 9000 server and restore the backup again on the Integrity server as described in Recovering HP 9000 files
- Re-create the container

  ```
  $ srp -add <srp_name> -t hp9000cl
  ```

- Run the configuration tool provided with the product

  ```
  $ /opt/HP9000-Containers/bin/hp9000_xinetd_setup \
    <srp_name>
  ```

- If the script exits with errors related to `itox`, update `/etc-hp9000/inetd.conf` so that it has only entries related to the minimum required services and run the `hp9000_xinetd_setup` script again.

## 5.6 Testing the HP 9000 container

Start the HP 9000 container using

```
$ srp -start <srp_name>
```

All startup messages should say `[OK]`. Look for any startup error messages in `/var /hpsrp/<srp_name>/etc/rc.log`

To verify status, use

```
$ srp -status  <srp_name> -v
```

Login to the container using

```
$ ssh <srp_name>
```

Start applications, as normally done on the HP 9000 server, for testing.

To stop the container

```
$ srp -start <srp_name>
```

## 5.7 Configuring ARIES parameters

Some application environments may need additional ARIES configuration as described in Tweaking ARIES configuration

# 6. Migrating HP 9000 Container Versions

## 6.1 Migrating from HP 9000 Containers A.03.0x

HP 9000 Containers A.03.0y can be installed on a system which have containers created using HP 9000 Containers A.03.0x (where x<y).

Before installing the new version of HP 9000 Containers, backup any configuration files that may have been manually modified. The only known case for modification is for `/opt/HP9000-Containers/config/hp9000_switch_commands`.

```
        $ cd /opt/HP9000-Containers/config

        $ cp -p hp9000_switch_commands hp9000_switch_commands.bkp
```

Then proceed to install the new depot

```
        $ swinstall -s <path to HP9KContainers depot> \*
```

Once installation is complete, restore the configuration file backed up.

```
        $ cd /opt/HP9000-Containers/config

        $ cp -p hp9000_switch_commands.bkp hp9000_switch_commands
```

There will be no change performed for the container during the installation process. If the enhancements and defect fixes in the new release are to be availed, the container has to be shut down and re-configured. Take a backup of the container rules file `/etc/cmpt/<srp_name>.rules` if it has been modified manually after previous container creation or upgrade. Then run

```
        $ srp -stop <srp_name>
        $ srp -replace <srp_name> -s init,cmpt
        $ srp -start <srp_name>
```

## 6.2 Migrating from HP 9000 Containers A.01.0x

HP 9000 Containers A.01.0x used a model quite similar to the *classic container* in HP 9000 containers A.03.0x. The migration of such containers to *classic container* can be performed with few manual steps and with the help of scripts provided with the solution.

Migrating to the *system container* can be done only by deleting and re-configuring the container (with same file system). This may involve environment specific steps and requires more manual effort.

In either case, a downtime is required for performing the migration.

### 6.2.1 Migrating to *classic* container

HP 9000 Containers requires HP-UX 11i v3 11.31 March 2011 update (or later). If the existing container host Integrity system is using an older version of HP-UX 11i v3, the first step is to update it to the latest HP-UX 11.31 OE.

Once the HP-UX 11i v3 OE has been updated, stop the container

```
        $ srp -stop <srp_name>
```

For every additional (non-primary) IP address configured for container usage in `/etc/rc.config.d/netconf`, manually associate the compartment tag if not already there. This was not a strict requirement in the earlier version of the product, but is one with HP 9000 Containers v3. The tag format is the same as that used for primary IP address of the container. For example, suppose there is an IPv4 address configured for use in `mysrp` container like `IP_ADDRESS[3]=<2nd IP>`, associate a tag using

```
IPV4_CMGR_TAG[3]='compartment="mysrp" template="base"
service="network" id="2"'
```

The "id" value needs to be incremented for each such IP address though. For example, if there is a 3rd IP address for `mysrp IP_ADDRESS[4]=<3RD IP>`, tag should look like

```
IPV4_CMGR_TAG[4]='compartment="mysrp" template="base"
service="network" id="3"
```

Check the version of HP-UX Containers (SRP) on the system

```
$ swlist | grep HP-UX-SRP
```

If SRP version is below A.03.01, install the latest HP-UX- Containers depot. Do NOT remove existing software.

```
$ swinstall -x autoreboot=true -s <path to depot> \*
```

Post reboot, configure system wide SRP parameters again

```
$ srp_sys -setup
```

Accept the prompt to migrate existing containers

```
Migrate all existing workload containers [y] y
```

Choose to disable PRM and IPFilter

```
Disable PRM      [n]y
Enable IPfilter  [n]n
```

You may choose default values for other parameters.

Reboot the server and stop the container if it is running

```
$ srp -stop <srp_name>
```

If there was any error reported on container migration during "`srp_sys -setup`", re-run the migration script

```
$ /opt/hpsrp/bin/util/srp_migrate -c <srp_name>
```

Check the version of HP 9000 Containers on the system

```
$ swlist | grep HP9000-Containers
```

```
$ swlist | grep HP9KContainers
```

If the version is below A.03.00, remove the existing version, prior to installing A.03.0x

```
$ swremove HP9000-Containers
```

Run the HP 9000 container specific migration script

```
$ /opt/HP9000-Containers/bin/hp9000_migrate <srp_name>
```

Configure container host name in `/var/hpsrp/<srp_name>/etc/rc.config.d /netconf` if it is different from the container name.

Start the container and verify

```
$ srp -start <srp_name>
```

### 6.2.2 Migrating to *system* container

This is possible only by deleting the container configuration and re-configuring it again with *system* container type. It may also involve environment specific manual steps. Before going down this path, it is worthwhile to consider if the image can be re-created from the HP 9000 server instead of performing the migration. Of course, this is possible only if no major changes have been made after transition from HP 9000 server except for data files, which can possibly be copied.

Below are some guidelines on the procedure to migrate to a *system* container. There may be additional environment specific steps needed.

Take a backup of the HP-UX 11i v3 server image before the migration.

Stop the container

```
$ srp -stop  <srp_name>
```

Backup network configuration and rules for future reference

```
$ cp /etc/rc.config.d/netconf /etc/rc.config.d/netconf.bkp
$ cp /etc/cmpt/<srp_name>.rules \
     /etc/cmpt/<srp_name>.rules.bkp
```

Delete the container

```
$ srp -delete <srp_name>
```

Remove any additional IP addresses from `/etc/rc.config.d/netconf` that have been configured for use by the container.

Also, move the `/tcb` folder inside the HP 9000 file system

```
$ mv <hp9000_root>/tcb  <hp9000_root>/tmp
```

Once this is done, the shared directories inside containers will be restored to the original versions that came over from the HP 9000 server. However, this may not be a valid operation for `/etc` since some files might have changed post the original SRP creation. These files need to be restored.

Uncompress and expand `etc-native.tar.gz` file created inside `<hp9000_root>`

```
$ cd /tmp
$ cp <hp9000_root>/etc-native.tar.gz .
$ gunzip <hp9000_root>/etc-native.tar.gz
$ tar -xvf ./etc-native.tar
```

Now copy the files in `/etc` that might have been updated post release

```
$ cp /tmp/etc/passwd <hp9000_root>/etc/passwd
$ cp /tmp/etc/group <hp9000_root>/etc/group
```

If `/var` is not on a separate file system from root file system, make sure that `/var/hpsrp` is on a different file system.

Move the HP 9000 file system to its new root

```
$ mv <hp9000_root> /var/hpsrp/<srp_name>
```

HP 9000 Containers A.03.0x requires HP-UX 11i v3 March 2011 update (or later). Update the system to the latest HP-UX 11i v3 OE, if it is not already using March 2011 (or later) version.

Check the version of HP 9000 Containers on the system

```
$ swlist | grep HP9000-Containers

$ swlist | grep HP9KContainers
```

If HP 9000 Containers version is below A.03.00, remove the existing version, prior to installing A.03.0x

```
$ swremove HP9000-Containers
```

Check the version of SRP on the system

```
$ swlist | grep HP-UX-SRP
```

If SRP version is below A.03.01, install the latest depot.

```
$ swinstall -x autoreboot=true <new SRP depot>
```

Enable SRP using

```
$ srp_sys -setup
```

PRM can be enabled, if there is an intention to host other containers on the same system.

Reboot the system for changes to take effect. Post reboot, the container can be re-configured by following the steps in Creating the HP 9000 system container section.

There may be other application specific files in `/etc` that may have been updated after the transition from HP 9000 server. If application testing encounters errors, look for any such files that need to be restored from `/tmp/etc`.

Once the testing is complete, delete all the HP 9000 users and groups from global `/etc/passwd` and `/etc/group` on the Integrity host system.

# 7 HP 9000 Containers file system layout

## 7.1 HP 9000 *system* container file system

The HP 9000 *system* container has a private HP 9000 file system under /var/hpsrp/<srp_name> directory. There are no directories that the container has write permission to, outside its file system.

The private directories inside the *system* container contain files that have been brought over from the HP 9000 server with some exceptions:

- The HP 9000 /dev is moved to dev-hp9000. A container private /dev is created at configuration time with a set of default devices copied over from the host system. The list of devices can be found in /opt/HP9000-Containers/config/hp9000_devices.
- Unsupported system services are removed from HP 9000 /sbin. A copy of the original directory is preserved for reference as /sbin-hp9000.
- Some specific products and files are copied into the container. The list of products and individual files that are copied can be found in /opt/HP9000-Containers/config/hp9000sys_copy_products and /opt/HP9000-Containers/config/hp9000sys_copy_files respectively
- Changes are made to some specific files inside the container file system - such as /etc/fstab, /etc/inittab, /etc/rc.config.d/netconf and /etc/hosts.
- ARIES configuration files are created under the container root directory – /.ariesrc and /.aries64rc.
- crontab files owned by root inside the container are moved so that system administration related cron jobs don't get automatically enabled in the container.
- Many commands have been either overwritten or disallowed read and execute permissions inside the container (depending on the option chosen at container creation). The list of such commands can be found in /opt/HP9000-Containers/config/hp9000sys_delete_commands.

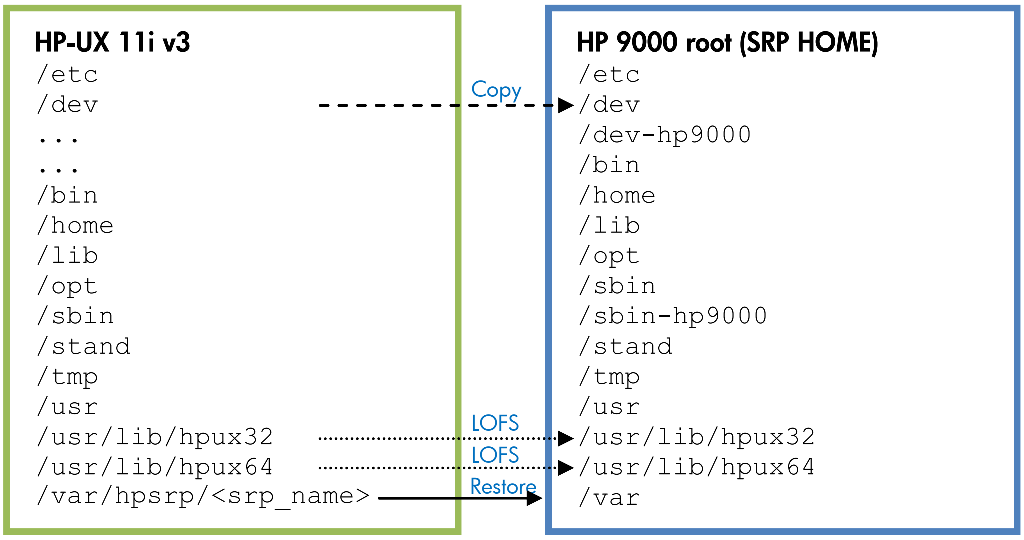


**Figure 5.1:** Overview of HP-UX 11i v3 Integrity file system configured with HP 9000 *System* Container

The file system layout, with the HP 9000 *system* container configured, is depicted in **Figure 5.1**.

There are two directories on the host system that are read-only shared (using loop back mounts) with HP 9000 system containers. These directories bring in ARIES libraries as well as other native Integrity libraries which are required for executing native commands and troubleshooting tools inside the container.

- `/usr/lib/hpux32`
- `/usr/lib/hpux64`

## 7.2 HP 9000 *classic* container file system

There are "three root directories" and corresponding file sets on an HP-UX 11i v3 host OS instance for an HP 9000 container

- The host (global compartment) HP-UX 11i v3 root (`/`)
- HP-UX SRP root (`/var/hpsrp/<srp_name>`)
- The HP 9000 container root (`/<hp9000_root>`)

The HP 9000 container file system is fairly, but not completely, isolated from the HP-UX 11i v3 file system. No HP 9000 system service is started inside the *classic* container apart from `cron` daemon. Applications inside the container interact with system services running on the HP-UX 11i v3 host system. To enable this communication, a part of the file system is shared between the container and the host system.

List of shared directories

- `/dev`
- `/etc`
- `/net`
- `/tcb`
- `/usr/lib/hpux32`
- `/usr/lib/hpux64`
- `/var/adm/syslog`
- `/var/adm/userdb`
- `/var/mail`
- `/var/news`
- `/var/opt/dce/rpc`
- `/var/uucp`
- `/var/yp`
- All subdirectories of `/var/spool`, except for `/var/spool/cron`

In addition, file system mount points may also need to be shared if they need to be accessed from within the HP 9000 container. Refer to *Configuring mount and export points* on how to implement mount point sharing.

File system sharing is implemented through local file system (`LOFS`) a.k.a. loop-back mounts to HP 9000 container directories from corresponding native directories. These `LOFS` mounts are performed as part of the HP 9000 container startup. This is enabled by configuring `/var/hpsrp/<srp_name>/etc/fstab`. These mount points are critical and need to be always active for applications to function correctly inside the HP 9000 container.

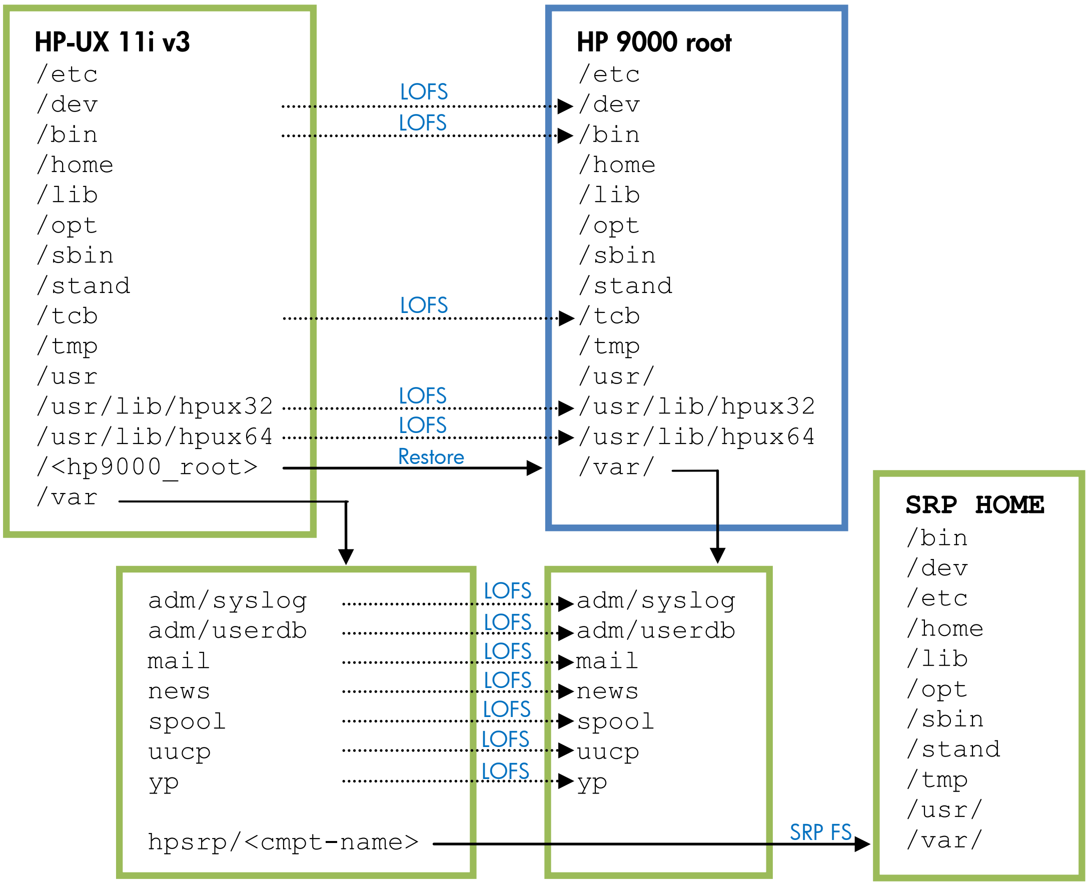


**Figure 5.2:** Overview of HP-UX 11i v3 Integrity file system configured with HP 9000 *Classic* Container

The file system layout, with the HP 9000 *classic* container configured, is depicted in **Figure 5.2**.

Some of the actions performed during a *classic* container configuration:

- Adds HP 9000 users to a login group and grants that group access to the container using `roleadm` command with role `SRPlogin-<srp_name>`.
- Configures `/var/hpsrp/<srp_name>/etc/cmpt/fstab` with the loop back mount points required to implement directory sharing.
- Merges files from `<hp9000_root>/etc` and `<hp9000_root>/tcb` into the corresponding directories on the HP-UX 11I v3 system based on some heuristics.
- Creates a set of symbolic links in `<hp9000_root>/usr/lib/security`.
- Deletes unsupported system daemons from `<hp9000_root>/sbin/init.d` (and the corresponding RC links).
- The HP 9000 container specific RC script (`hp9000_rc`) and links are copied to `/var/hpsrp/<srp_name>/sbin/init.d`.
- Move root `crontab` file so that system administration related cron jobs don't get automatically enabled inside the container.
- ARIES resource configuration files are created under the container root directory – `/.ariesrc and /.aries64rc`.

## 7.3 HP 9000 Containers directories

HP 9000 Containers depot installation creates following directories under `/opt/HP9000-Containers/`

| | | |
|---|---|---|
| `bin` | : | setup, cleanup and management scripts |
| `docs` | : | documentation |
| `config` | : | configuration for setup |
| `newconfig` | : | default configuration files which are copied into container |

The HP 9000 container configuration logs can be found at `/var/opt/HP9000-Containers/logs.`

HP 9000 Containers depot also installs files in the following directories:

```
/opt/hpsrp/etc/templates
/opt/hpsrp/bin/update-ux
/opt/hpcmgr/lib/Cmgr
/opt/hpcmgr/lib/Util
/opt/hpsmh/data/htdocs/srpgui
/usr/lbin/sw/post_session
/usr/share/man/man5.Z/container_hp9000.5
```

When an HP 9000 container is setup, records of changes made to the file system are stored under `<hp9000_root>/var/opt/HP9000-Containers`. This information is critical to be preserved in order to be able to run a proper cleanup if and when the HP 9000 container is to be deleted or re-configured.

# 8 HP 9000 Containers Administration

Most of the administration tasks for HP 9000 containers need to be performed from the HP-UX 11i v3 host system (referred to as the *global compartment* in the following sections).

## 8.1 Administrator privileges

By default, the `root` user on the host system is assigned administrator privilege for lifecycle management (`start`, `stop`, `export`, `import`, `delete`, `modify`) of the container. It is possible to allow additional users the privilege to do lifecycle management on the HP 9000 container using RBAC (Role Based Access Control).

Adding an administrator

```
$ roleadm add <user-name> SRPadmin-<srp_name>
```

Deleting an administrator

```
$ roleadm delete <user-name> SRPadmin-<srp_name>
```

## 8.2 Startup and shutdown

Starting the HP 9000 container

```
$ srp -start <srp_name>
```

Shutting down the HP 9000 container

```
$ srp -stop <srp_name>
```

If it warns that certain processes could not be terminated, re-try the operation

```
$ srp -stop  <srp_name>
```

To enable or disable auto start at system boot time

```
$ srp -replace <srp_name> -s init
```

Check `/var/hpsrp/<srp_name>/etc/rc.log` on the host system for RC logs. Inside a *system* container, this file is accessible as `/etc/rc.log`. It is not possible to access this log file from within a *classic* container.

## 8.3 User account management

### 8.3.1 HP 9000 *system* container

User management activities can be performed locally just like on the HP 9000 server. The container has a local `root` user with complete access to the HP 9000 file system.

Resetting container root password from global compartment

```
$ srp -replace <srp_name> -s init
```

The following limitations apply to user management inside *system* containers

- There is no support for user accounting and quota

It is highly recommended that there be no users configured in the global compartment, apart from the default users and any other users related to system administration or system management applications.

### 8.3.2 HP 9000 *classic* container

User management needs to be performed in the global compartment. As part of HP 9000 container configuration, a group "<srp_name>-login" is created and this group is allowed container access. All local users from HP 9000 /etc/passwd file are added to this auxiliary group.

- Adding a new user

  Login to the global compartment as root and execute "useradd" command. There's no need to prefix <hp9000_root> directory in this step while specifying the home directory.

  Create home directory inside the <hp9000_root> directory. Set permission for home directory to 0755.

  Add user to one of the HP 9000 container login groups. Names of default login groups (created at setup time) start with <srp_name>-login

  ```
  $ groupmod -a -l <username> <srp_name>-login
  ```

- Adding all users of a group

  If all members of a group should be allowed access to container, use

  ```
  $ roleadm assign \&<group-name> SRPlogin-<srp_name>
  ```

- Disallow access for all members of a group

  ```
  $ roleadm revoke \&<group-name> SRPlogin-<srp_name>
  ```

# 8.4 Configuring SSH authorization keys

## 8.4.1 HP 9000 *system* container

SSH keys can be generated and used similar to that on the HP 9000 server.

## 8.4.2 HP 9000 *classic container*

Enabling automatic login to an HP 9000 container using SSH authorization keys requires home directories to be created *outside* the container (in the *global compartment*). Follow the sequence of steps below:

- For every user who needs to use SSH authorization keys, create a home directory on the host system (*global compartment*) with same path as inside the HP 9000 container. Change permissions of home directory to 0755 and ownership of directory to the individual user.
- The user should login to the host system and create a $HOME/.ssh directory with 0700 permissions.
- The user should login to the client systems (from where automatic login is to be allowed) and generate a ssh key

  ```
  $ ssh-keygen -t dsa
  ```

- Append the contents of $HOME/.ssh/id_dsa.pub on the client system to $HOME/.ssh/authorized_keys on the target system (*global compartment*).

## 8.5 Configuring mount and export points

### 8.5.1 Configuring NFS and AUTOFS clients

- An HP 9000 *system* container has NFS and AUTOFS client support inside it. Hence the configuration can be performed inside the container file system just like on the HP 9000 server.
- Inside an HP 9000 *classic* container, there is no support for NFS or AUTOFS. The configuration can be performed on the host system and exposed to the container

  Configure /etc/fstab, /etc/auto_master and /etc/auto.direct files on the host.

  Perform actual mounts on the host system.

  Make the mount point visible inside the HP 9000 *classic* container

```
$ /opt/HP9000-Containers/bin/hp9000_link_dir \
  <directory> <srp_name>
```

### 8.5.2 Configuring VxFS mount points

- For a *system* container, there are three options.

  a) Configure container pre-start mounts. These mounts will be done from global into the container file system when the container starts up and will be unmounted as it shuts down. The configuration has to be done in /var/hpsrp/<srp_name>.setup/fstab and using the same format as used in the host /etc/fstab. For example,

```
$ echo "\n/dev/vg01/lvol2 /var/hpsrp/mysrp/mnt vxfs \
  delaylog 0 2" >> /var/hpsrp/mysrp.setup/fstab
```

  b) Make the logical volume visible inside the container and then configure the mount in the local fstab.. For example,

```
$ srp -add <srp_name> -tune device=/dev/vg01/lvol2
$ srp -add <srp_name> -tune device=/dev/vg01/rlvol2
$ echo "\n/dev/vg01/lvol2 /mnt vxfs delaylog 0 2\
  >> /var/hpsrp/mysrp/etc/fstab
```

  This facility is not currently supported for the /usr file system since the container has to do LOFS mount of /usr/lib/hpux32 and /usr/lib /hpux64 before it can start any service. It is also not supported for /var because container startup needs to read this directory.

  c) Configure the mount on the HP-UX 11i v3 host with the mount point inside the container. For example. if there is a /mnt mount point on the HP 9000 server configure

```
$ echo "\n/dev/vg01/lvol2 /var/hpsrp/mysrp/mnt vxfs \
  delaylog 0 2" >> /etc/fstab
```

  There is currently a known limitation with this approach. Post system reboot, the "bdf" and "mount" commands inside the container do not display information for these mount points. The issue is related to RC sequencing order between fstab processing and SRP initialization and does not have an immediate fix.

- For a *classic* container, there is no mount support inside it. Configure the mount point on the HP-UX 11i v3 host system, do a mount and make it visible inside the container using the `hp9000_link_dir` tool.

```
$ /opt/HP9000-Containers/bin/hp9000_link_dir \
  <directory> <srp_name>
```

  For example,

```
$ echo "\n/dev/vg01/lvol2 /mnt vxfs delaylog 0 2" \
  >> /etc/fstab
$ mount /mnt
$ /opt/HP9000-Containers/bin/hp9000_link_dir \
  <directory> <srp_name>
```

  Alternatively directly mount from global to a container file system directory

```
$ mkdir <hp9000_root>/mnt
$ echo "\n/dev/vg01/lvol2 /hp9000-root/mnt vxfs \
  delaylog 0 2" >> /etc/fstab
```

### 8.5.3 Configuring NFS exports

NFS server is not supported inside an HP 9000 container.

- Configure the NFS exports in *global compartment*.
- Specify the complete path `<hp9000_root/dir>`, where `dir` was the directory originally exported from the HP 9000 system.
- On the client systems, make sure that the host name configured is of the native HP-UX 11i v3 server, not of the HP 9000 container.

## 8.6 Modifying IP address configuration

### 8.6.1 Changing primary IP address

- Stop the HP 9000 container

```
$ srp -stop <srp_name>
```

- Re-configure the container network parameters

```
$ srp -replace <srp_name> -s network
```

  If the IP address is being managed by Serviceguard then,

```
Add IP address to netconf file? [yes]   no
```

- Open the container `sshd` configuration file, by default (`/var/hpsrp/<srp_name>/opt/ssh/sshd_config`) and change if it was configured.

```
ListenAddress <new IP-address>
```

- Re-configure applications with new IP address.

- Update `/etc/hosts` and `/var/hpsrp/<srp_name>/etc/hosts` to reflect new mapping.

- Start the HP 9000 container

```
$ srp -start <srp_name>
```

### 8.6.2 Adding a new IP address for the HP 9000 container

- Find number of IP addresses already assigned to the container

```
$ srp -list <srp_name> - s network -v
```

- Find the id value for the next IP. For instance, the next id will be 2 if no additional IP has been assigned apart from the container primary address.
- Add the new IP address using

```
$ srp -add <srp_name> -s network -id "<next id>"
```

If the IP address is being managed by Serviceguard then,

```
Add IP address to netconf file? [yes]   no
```

### 8.6.3 Changing host IP address

During `srp_sys -setup` the host `sshd` might have been configured to listen specifically to the host system IP address. So if the native IP address changes, the `sshd` configuration file needs to be updated. Edit the configuration file (default `/opt/ssh/etc/sshd_config`) for the `ListenAddress` parameter.

```
ListenAddress <new IP-address>
```

Re-start sshd

```
$ /sbin/init.d/secsh stop
$ /sbin/init.d/secsh start
```

## 8.7 Modifying hostname

- For HP 9000 *system container* the hostname and node name configuration can be performed from within the container just like on the HP 9000 server. Edit `HOSTNAME` parameter in `/etc/rc.config.d/netconf` and update `/etc/hosts`, `/etc/mail/sendmail.cw` and any application configuration. Also update `/etc/hosts` in global compartment to reflect new host name.
- For HP 9000 *classic container*, hostname and node name configuration needs to be done from *global* (the HP-UX 11i v3 host). Edit `/var/hpsrp/<srp_name>/etc/rc.Config.d/netconf`, `/etc/hosts`, `/etc/mail/sendmail.cw` and any application configuration.

## 8.8 Modifying resource entitlements

HP Process Resource Manager (PRM) can be used to manage resource entitlements for an HP 9000 *system container*. To modify the PRM configuration for the container, run

```
$ srp -replace <srp_name> -s prm
```

It is also possible to modify the PRM group configuration using the System Management Homepage (SMH) or using PRM commands directly. For more details

```
$ man prm
```

## 8.9 Monitoring and spawning processes from Integrity host

The `ps` command in *global compartment* lists processes running in all containers but with certain obfuscation in the name. For example, an `sshd` running in a container with id 3 will be displayed as `/opt/ssh/sbin/3_Sshd`.

To list only processes running inside an HP 9000 container use

```
$ srp_ps <srp_name> <ps options>
```

To invoke a command in the context of an HP 9000 `system` container

```
$ srp_su <srp_name> root -c "<command full path> <args>"
```

To invoke a command in the context of an HP 9000 *classic* container

```
$ srp_su <srp_name> root -c "chroot <hp9000_root> \
  <command full path> <args>"
```

## 8.10 HP 9000 container patching

### 8.10.1 Patching support overview

Patching applications using custom installers should work fine inside HP 9000 *system* and *classic* type containers.

Patching using Software Distributor (SD) or other custom tools is not supported with HP 9000 *classic* containers.

With HP 9000 HP-UX 11i *system* containers, patching using SD is supported but with some differences as described in the sections to follow.

### 8.10.2 Native components inside container

There is a set of products (mainly NFS) and files (commands such as `ipcs`, `mount`, `netstat`, `ioscan`, `traceroute` etc) that are copied from the host HP-UX 11i v3 system into an HP 9000 container as part of configuration. This is needed because the corresponding PA-RISC legacy components do not work with the HP-UX 11i v3 kernel and the differences cannot be bridged using ARIES user space emulation. There is also a backup copy of these native files placed in `/var/opt/HP9000-Containers/native` inside the container.

When products including these files are patched inside the container they might get overwritten by HP 9000 versions that can cause failures. An SD post session script `/usr/lbin/sw/post_session/hp9000_flag_sync` is automatically run to copy the files again from the backup copy made in `/var/opt/HP9000-Containers /native`.

The copying of files from the host is not a one-time operation. Following events trigger creation of a file `hp9000_needs_recovery` inside the container under `/var/adm/sw` inside the container. When the container is re-started, this file is detected and the set of native files is copied again. The following actions create the recovery file are

- Patching or installation of products including these files on the host
- Removing products or patches from the host
- `Update-UX` on the host
- Patching or installation of these products from within the container
- Removing products or patches from within the container

The copying can also be triggered manually (when container is in stopped state) using

```
$ srp -replace <srp_name> -s init
```

### 8.10.3 Restricted commands inside container

There is a set of commands (mostly related to system administration tasks) that are disallowed inside containers. HP 9000 Containers A.03.01 provides two ways to restrict these commands.

a) Deny execute permission for these commands using compartment rules in `/opt/HP9000-Containers/config/hp9000.disallowed.cmds`.
   This also causes the read permission on these files to be denied (compartment rules cannot distinguish between read and execute). This was the only available option in HP 9000 Containers A.03.00.

b) Replace unsupported commands with a dummy command that exits with an error message. Every time there is a `swinstall` or `swremove` operation, the commands are replaced again (using a SD post session script). Commands listed in `/opt/HP9000-Containers/config/hp9000sys_delete_commands` are replaced. This option is available starting HP 9000 Containers A.03.01 for HP-UX 11i `system` containers.

The choice can be made at the time of container creation by answering `yes` (for rules) or `no` (for replacement) to the following question

```
Use rules to restrict unsupported commands?
```

The choice may also be changed later using replace operation

```
$ srp -replace <srp_name> -s init,cmpt
```

Compartment rules provide a stricter way to restrict these commands. However, because read permission on these files is disabled, SD operations such as `swinstall`, `swverify` and `swremove` will fail for products that include these commands. For example, a quality pack may contain several products some of which contain files that belong to the list of restricted commands and installation/rollback of the pack can turn out to be tedious. A workaround is to temporarily disable the compartment rules when the operation is being performed. Open `/etc/cmpt/<srp_name>.rules` on the host HP-UX 11i v3 server and comment out (using #) the line including `hp9000. disallowed.cmds`. Then run the command

```
$ setrules
```

Once patching is complete, the rules have to be enabled again by editing the rules file to remove the comment and running `setrules` again.

If the option to replace unsupported commands (which is the default) is chosen, the SD operations for products including these files should not be hampered. However, `swinstall` and `swremove` will run a bit slower because a post session script will be run to re-do the replacement of these commands. It is important that the post session scripts be not interrupted.

### 8.10.4 Applying kernel patches

There is no active HP 9000 kernel inside an HP 9000 container. Hence applying kernel patches inside the container will have no effect. The `swinstall` operation will, however, go through and update files. There will be no automatic re-start of the container.

### 8.10.5 Command and library switch

HP 9000 Containers v3 provide options to switch to using native HP-UX commands and latest versions of system libraries using the `cmdv3` and `libv3` templates respectively. However, once such a switch is made patching products involving these files run the risk of overwriting them with the legacy components again. To get the commands and libraries back, the container has to be stopped and replaced.

```
$ srp -replace <srp_name> -t cmdv3
$ srp -replace <srp_name> -t libv3
```

### 8.10.6 `swverify` errors

The `swverify` command inside container may report errors for various reasons

a) If compartment rules are used to restrict commands, there is no read permission on the command files.

b) If command and/or library switch is done, the file attributes differ from what is stored in the SD database.

c) Post a `swremove` operation for a patch that existed when the container was created and before `swinstall` of the same or a different patch containing the same files (only if the fileset includes one of the unsupported commands and services, or files that have been copied from the global).

### 8.10.7 Post session processing

As part of container creation, there are some scripts and configuration files that get copied into the container. It is critical to have these in place for the SD post session processing to work. The post session processing takes care of deleting unsupported services, restoring native files and overwriting unsupported commands after patching operations inside the container.

List of files related to post session processing

```
/usr/lbin/sw/post_session/hp9000_flag_sync
/usr/lbin/sw/post_session/hp9000_delete_svcs
/var/opt/HP9000-Containers/hp9000sys_sd_filesets
/var/opt/HP9000-Containers/deleted_services
```

If commands are restricted using replacement, then

```
/usr/lbin/sw/post_session/hp9000_replace_cmds
/var/opt/HP9000-Containers/hp9000sys_delete_commands
```

## 8.11 Run level support

### 8.11.1 HP 9000 *system* container

The `srp_init` daemon is the container is equivalent of the system `init(1M) daemon.` It is the first process started in the container and spawns processes/monitors based on the `/etc/inittab` file.

The `srp_init` command can be run inside the container to communicate with the `srp_init` daemon and change the run level.

```
$ /sbin/srp_init 0|1|2|3|4|5|6|Q|q
```

Check `/srp.log` inside the container for messages from `srp_init.`

For more details, including differences with system `init`

```
$ man 1m srp_init
```

### 8.11.2 HP 9000 *classic* container

There is only partial support for run-levels. No system daemon, apart from `cron`, is supported. Application services are supported and the processing of RC directories follows the same order as on a physical server. However, there is no support for switching run-level inside the container.

## 8.12 Backup and Migration

### 8.12.1 Exporting and importing HP 9000 *system* containers

To just backup the container configuration (not the file system)

```
$ srp -export <srp_name> -b
```

To backup the container configuration and the file system)

```
$ srp -export <srp_name> ok_export_dirs=yes
```

To delete the container configuration and restore from backup

```
$ srp -delete <srp_name> -b
$ srp -import -xfile  <name of configuration export file>
```

To delete the container completely (including file system) and restore from backup

```
$ srp -delete <srp_name> delete_changes_ok=yes
$ mv /var/hpsrp/<srp_name> <some backup location>
$ srp -import -xfile  <path of export file including file \
  system>
```

**NOTE:** There is a known issue with HP-UX Containers A.03.01 that the ownership of imported files would change if the same users are configured on the Integrity host system with different UIDs (or same groups with different GIDs), Hence it is recommended that no users/groups be configured in the *global compartment* or make sure that the IDs match (use LDAP or NIS).

### 8.12.2 Exporting and importing HP 9000 *classic* containers

The HP 9000 *classic* container shares a part of the file system with the HP-UX 11i v3 host on which it resides. Hence it is advised that backup be taken at a system level, and not at a container level.

To just backup the container configuration (not file system)

```
$ srp -export <srp_name> -b
```

To delete and re-import just the container configuration

```
$ srp -delete <srp_name> -b
$ srp -import -xfile  <configuration export file>
```

### 8.12.3 Copying (cloning) an HP 9000 *classic* container

This is not supported.

### 8.12.4 Copying (cloning) an HP 9000 *system* container

Backup the source container configuration and the file system.

```
$ srp -export <srp_name> ok_export_dirs=yes -b
```

Use the exchange file created to import into a new container. Specify new network parameters for the container

```
$ srp -import <new_srp_name> -xfile <path of export file>
```

### 8.12.5 Backup applications with HP 9000 *system* containers

Backup applications may be run inside HP 9000 *system* containers but there are some limitations

- The backup application should not attempt tasks or commands that are unsupported inside the container – typically related to system administration.
- `Ignite-UX` is not known to work inside HP 9000 containers.
- If compartment rules are used to restrict unsupported commands, read permissions on these files will be disabled inside an HP 9000 container. Hence an image which includes command directories (`/sbin`, `/usr/bin`, `/usr /contrib/bin`, `/usr/lbin` and `/usr/sbin`) may not contain the complete set of files.
- It is advised not to recover a complete image from an HP 9000 server inside the container. If at all such a recovery happened make sure to do a re-configuration before the next container start

  ```
  $ srp -replace <srp_name> -s init
  ```

Tape devices can be exposed to the container using

```
$ srp -add <srp_name> -tune device=<tape device file>
```

Some backup applications (such as *Legato Networker*) can be configured to read `/etc/fstab` and backup all configured file systems. If this option is to be used, this means that the mount points have to be configured in container local `fstab` (not in container pre-start `fstab`). However, `/usr` and `/var` are not supported in local `fstab`. They have to be configured in container pre-start `fstab`. For these file systems the backup may have to be initiated from the host separately.

`HP Data Protector` is known to fail when used inside an HP 9000 *system* container. A workaround is to execute the backup by the IP address of the host - the DP disk agent inside the container should point to the DP Media agent running on the host.

If backup applications are used on the host 11i v3 system then the container needs to be in started state if mounts are configured in container pre-start or local `/etc/fstab`.

### 8.12.6 Backup applications with HP 9000 *classic* containers

In addition to all the caveats described for the *system* container. Additional care should be exercised because directories such as `/etc` and `/dev` are shared with the host system. Restoring a complete image from an HP 9000 server into an HP 9000 *classic* container should never be attempted as it will destroy the HP-UX 11i v3 `/etc` contents.

It is recommended that the backup applications be on the Integrity host system to avoid the described issue. If such backup applications need to run commands inside the container for any reason, the `srp_su` command may be employed

```
$ srp_su <srp_name> root -c "chroot <hp9000_root> \
  <command> <args>"
```

## 8.13 Auditing with HP 9000 Containers

The HP-UX audit subsystem is not virtualized at a container level and hence auditing cannot be managed completely from within the container. However, it is possible to enable auditing in global and filter records for specific containers.

Audit management in global is no different from that on a system without containers.

At a command level, `audsys` (1M) is used for enabling/disabling auditing, `audevent` (1M) is used to select events, `audomon` (1M) for monitoring etc.

There is a known issue when using `-N` value greater than 1 with audsys command. Check for availability of patch PHCO_43198 if `N>1` is a requirement.

With *system* containers, the selection of users has to be performed from within the container using `userdbset` (with SMSE) or `audusr` (with trusted mode).

For example,

```
$ srp_su <srp_name>
$ audusr -a <user>
$ userdbset -u <user> AUDIT_FLAG=1
```

Once configured, audit records generated by processes in all the containers are written to audit log files in the global view.

To view all audit records generated,

```
$ auditdp -r <global_log>
```

To display records for a specific *system* container from global

```
$ audisp -C <srp_name>
$ auditdp -r <global_log> -s "+cmpt=<srp_name>"
```

The records displayed in global will, however, have an incorrect mapping between user/group IDs and names. This is because the records contain only the IDs and the mapping is different in global and inside container.

To display raw audit data of all containers with IDs correctly mapped to names, run the sample script provided in `/opt/audit/AudReport/bin/hp9000_audit_global` instead of `auditdp`. This script is included in AuditExt B.11.31.04.01 (or later) which can be downloaded from HP Software Depot.

To display audit logs for a specific containers

```
$ hp9000_audit_global -C <srp_name> -a <global_log>
```

To display audit logs for all containers

```
$ hp9000_audit_global -a <global_log>
```

To copy the relevant records from global into a container

```
$ /opt/audit/AudReport/bin/srp_auditdp_copy
  -r <global_log > -R <local_log> -C <srp_name>
```

To copy the records from global to all `system` containers

```
$ /opt/audit/AudReport/bin/srp_auditdp_copy
  -r <global_log > -R <local_log>
```

To view the copied records from within a `system` container

```
$ audisp <local_log>
```

There is currently one known major auditing limitation with legacy *system* containers. Login and logoff events are not included in audit logs. There is only one known workaround for this issue – write an `init.d` startup script that executes the following commands

```
      echo ”audit_en_logins_compat/W 1" | adb -o –w
/stand/vmunix\ /dev/kmem
      echo "audit_logoff_compat/W 1" | adb -o -w /stand/vmunix\
/dev/kmem
```

With HP 9000 *classic* containers, the selection of users is also done in global (`/etc` and `/tcb` are shared between the host and the container). The audit records will contain information that for the container as well and currently there is no way to filter records for the container. Note that there can be only one such container on the system.

# 9 Using Container Manager

The Container Manager is integrated in the System Management Homepage (SMH) and provides a graphical user interface (GUI) to manage *system*, *workload* and *HP 9000* containers.

Tasks supported for HP 9000 *system* containers with the Container Manager

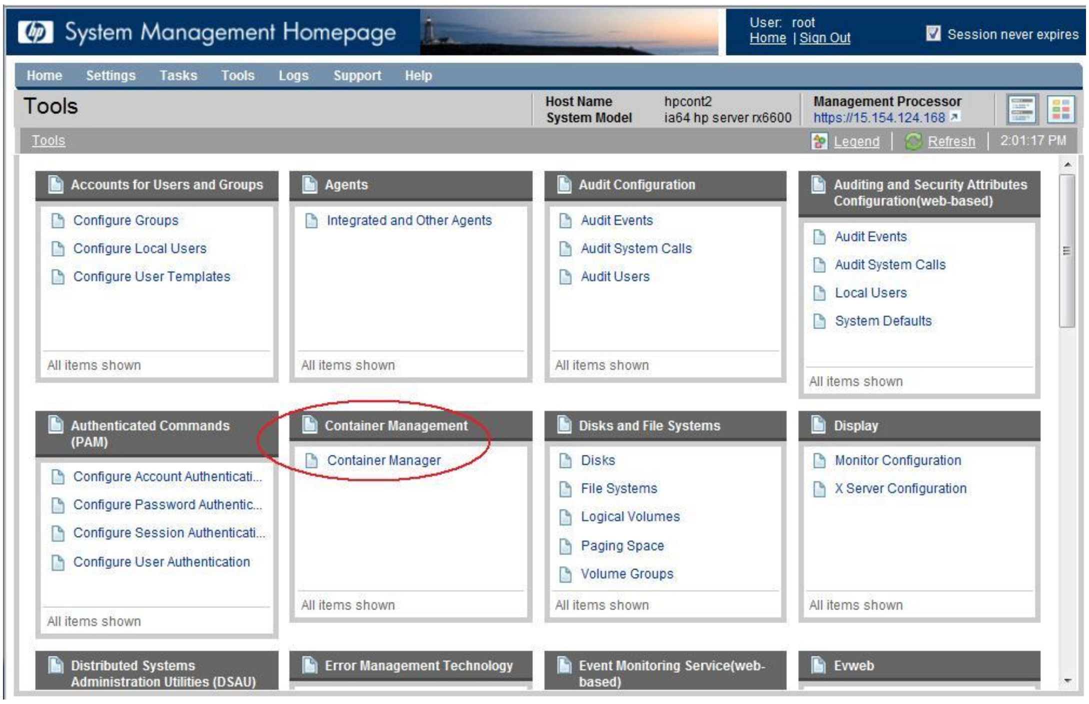
- Enable or disable system wide configuration for containers
- Monitor container status and activity
- Create and delete containers
- Start and stop containers
- Modify container configuration
- Export and import containers

## 9.1 Accessing Containers Manager from SMH

Login to SMH by using the SMH administrator or root credential:

`http://`*`<hostname>`*`:2301/`

On the `SMH Tools page`, select `Container Manager` from the `Container Management` menu box, as shown in Figure 7.1.



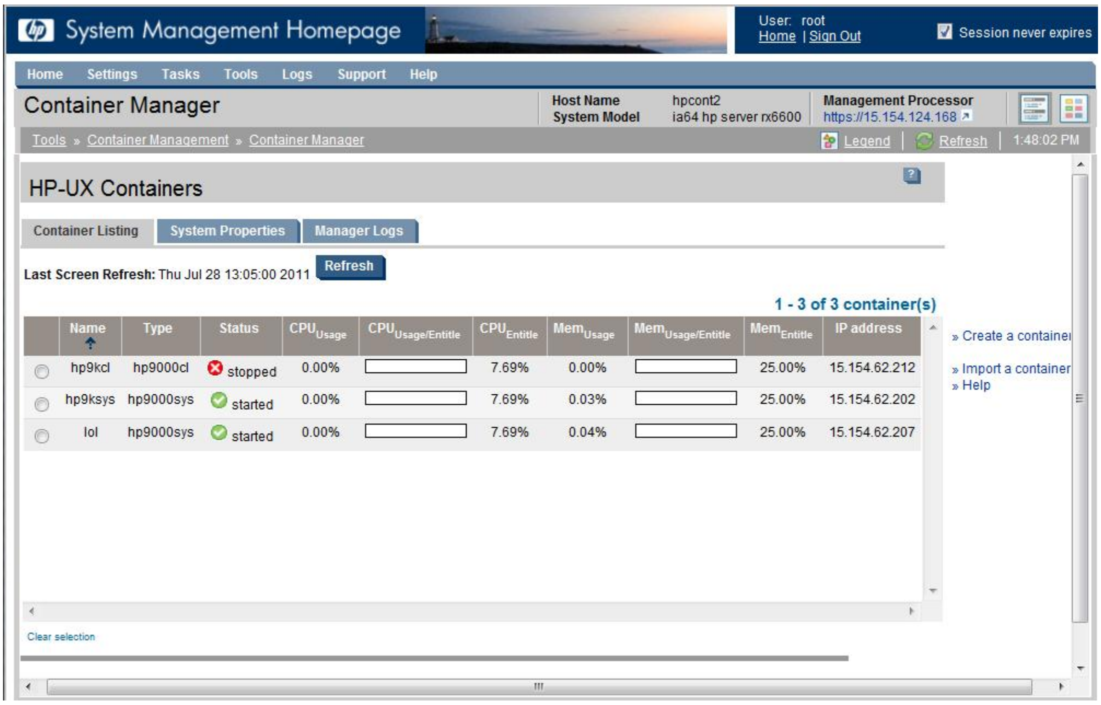
User: root
Host Name hpcont2
System Model ia64 hp server rx6600
Management Processor https://15.154.124.168
2:01:17 PM

**Figure 7.1:** SMH – Selecting Container Manager

## 9.2 Container Manager Home page

The Container Manager home page provides a view of all containers on the Integrity host system including current state and resource utilization for each container.

To access help information for any Container Manager page, click the `question mark icon "?"` located in the upper right corner of the Container Manager home page.



System Management Homepage

User: root
Home | Sign Out
☑ Session never expires

Home Settings Tasks Tools Logs Support Help

Container Manager

Host Name hpcont2
System Model ia64 hp server rx6600
Management Processor https://15.154.124.168

Tools » Container Management » Container Manager
Legend | Refresh | 1:48:02 PM

HP-UX Containers

Container Listing | System Properties | Manager Logs

Last Screen Refresh: Thu Jul 28 13:05:00 2011 Refresh

1 - 3 of 3 container(s)

| | Name | Type | Status | CPU Usage | CPU Usage/Entitle | CPU Entitle | Mem Usage | Mem Usage/Entitle | Mem Entitle | IP address |
|---|---|---|---|---|---|---|---|---|---|---|
| ○ | hp9kcl | hp9000cl | stopped | 0.00% | | 7.69% | 0.00% | | 25.00% | 15.154.62.212 |
| ○ | hp9ksys | hp9000sys | started | 0.00% | | 7.69% | 0.03% | | 25.00% | 15.154.62.202 |
| ○ | lol | hp9000sys | started | 0.00% | | 7.69% | 0.04% | | 25.00% | 15.154.62.207 |

» Create a container
» Import a container
» Help

Clear selection

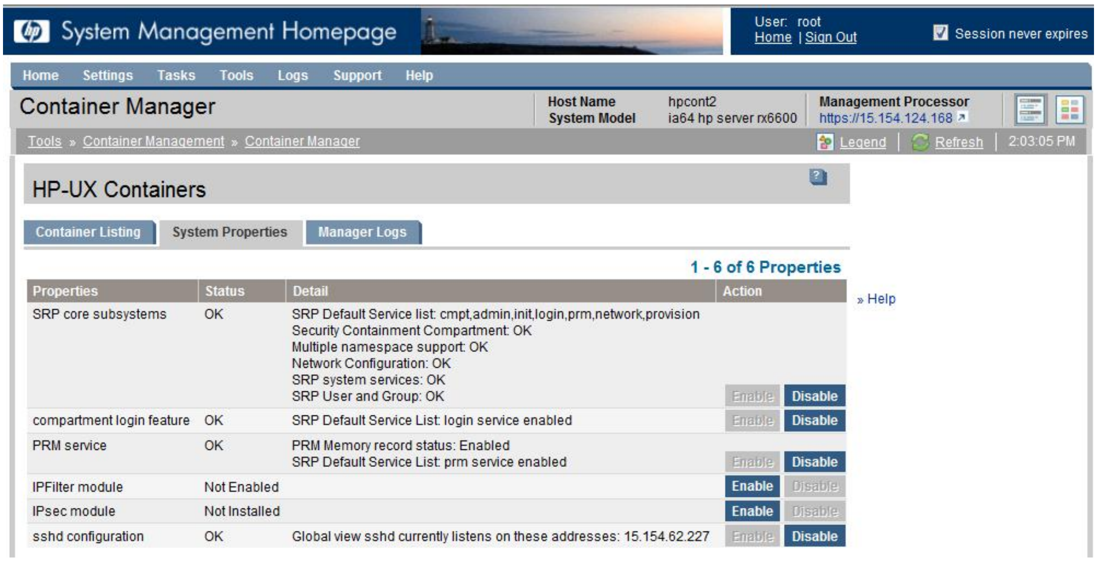
**Figure 7.2:** SMH – Container Manager home page

## 9.3 Setting up the container environment

Enable the core subsystem properties before creating containers by either using the `srp_sys` command (with the `–enable` or `–setup` option) or by performing the following actions using the `Container Manager`

a) Go to the `System Properties tab`.
b) Click the `Enable` button for the SRP core subsystems property.
c) Wait for the successful completion message to appear in the result window.
d) Enable the `compartment login` feature, the `PRM service`, and the `sshd configuration` properties. Optionally, you can enable other properties.
e) Reboot the system to enable the required properties.

The `Container Listing` tab will only be displayed if the `SRP core subsystem` property status is set to OK.



System Management Homepage

User: root
Home | Sign Out
☑ Session never expires

Home Settings Tasks Tools Logs Support Help

Container Manager

Host Name hpcont2
System Model ia64 hp server rx6600
Management Processor https://15.154.124.168

Tools » Container Management » Container Manager
Legend | Refresh | 2:03:05 PM

HP-UX Containers

Container Listing | System Properties | Manager Logs

1 - 6 of 6 Properties

| Properties | Status | Detail | Action |
|---|---|---|---|
| SRP core subsystems | OK | SRP Default Service list: cmpt,admin,init,login,prm,network,provision<br>Security Containment Compartment: OK<br>Multiple namespace support: OK<br>Network Configuration: OK<br>SRP system services: OK<br>SRP User and Group: OK | Enable Disable |
| compartment login feature | OK | SRP Default Service List: login service enabled | Enable Disable |
| PRM service | OK | PRM Memory record status: Enabled<br>SRP Default Service List: prm service enabled | Enable Disable |
| IPFilter module | Not Enabled | | Enable Disable |
| IPsec module | Not Installed | | Enable Disable |
| sshd configuration | OK | Global view sshd currently listens on these addresses: 15.154.62.227 | Enable Disable |

» Help

**Figure 7.3:** Enabling the system properties

## 9.4 Creating an HP 9000 container

To create a container, follow the steps below:

a) From the `Container Manager` home page, click `Create a container.`

b) Enter a `Container Name.`

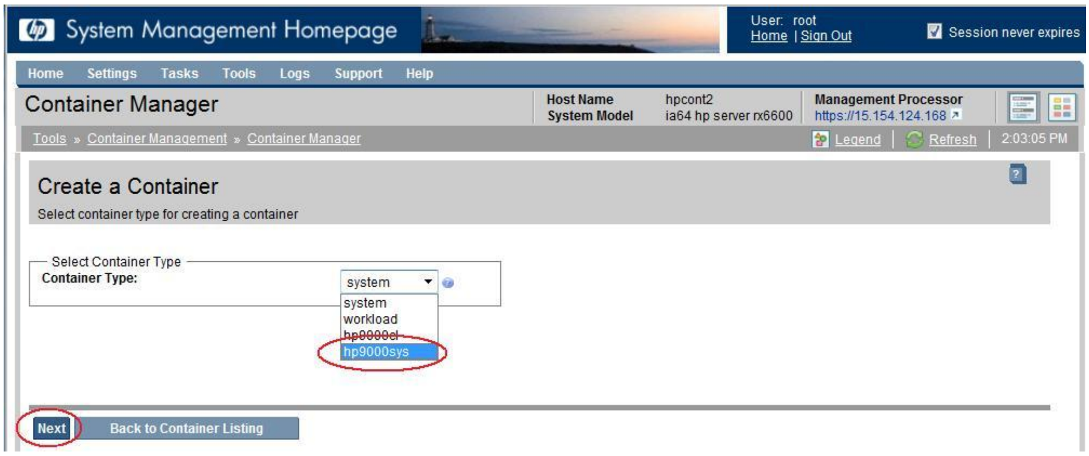
**NOTE:** You cannot use the keywords `system, workload, hp9000sys or hp9000cl` as container names.

c) By default, the `System Container` type is selected. Select `hp9000sys` for creating an HP 9000 *system* container and `hp9000cl` for HP 9000 *classic* container.

d) Modify parameter fields as desired.

e) Click `Create`. A result window pops up and shows the create command being executed and the logs being generated on the host. Once the create operation is complete, an operation success or failure message is displayed.

f) Click `Back to container listing` and close the result window.


System Management Homepage
User: root
Home | Sign Out
Session never expires
Home Settings Tasks Tools Logs Support Help
Container Manager
Host Name hpcont2
System Model ia64 hp server rx6600
Management Processor https://15.154.124.168
Tools » Container Management » Container Manager
Legend | Refresh | 2:03:05 PM
system
workload
hp9000cl
hp9000sys
Next
Back to Container Listing

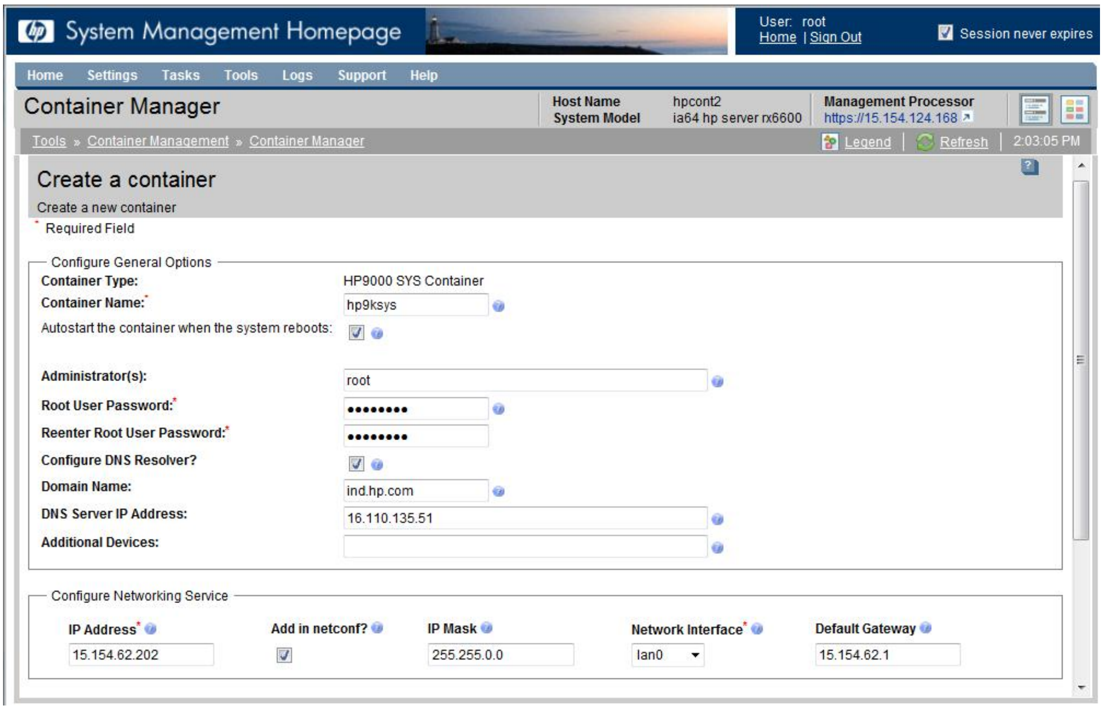
Create a Container

Select container type for creating a container

Select Container Type

Container Type: system

**Figure 7.4:** Container Manager – selecting an `hp9000sys` container


System Management Homepage
User: root
Home | Sign Out
Session never expires
Home Settings Tasks Tools Logs Support Help
Container Manager
Host Name hpcont2
System Model ia64 hp server rx6600
Management Processor https://15.154.124.168
Tools » Container Management » Container Manager
Legend | Refresh | 2:03:05 PM

Create a container

Create a new container

* Required Field

Configure General Options

| | |
|---|---|
| Container Type: | HP9000 SYS Container |
| Container Name:* | hp9ksys |
| Autostart the container when the system reboots: | ☑ |
| Administrator(s): | root |
| Root User Password:* | •••••••• |
| Reenter Root User Password:* | •••••••• |
| Configure DNS Resolver? | ☑ |
| Domain Name: | ind.hp.com |
| DNS Server IP Address: | 16.110.135.51 |
| Additional Devices: | |

Configure Networking Service

| IP Address* | Add in netconf? | IP Mask | Network Interface* | Default Gateway |
|---|---|---|---|---|
| 15.154.62.202 | ☑ | 255.255.0.0 | lan0 | 15.154.62.1 |

**Figure 7.5:** Container Manager – creating an `hp9000sys` container

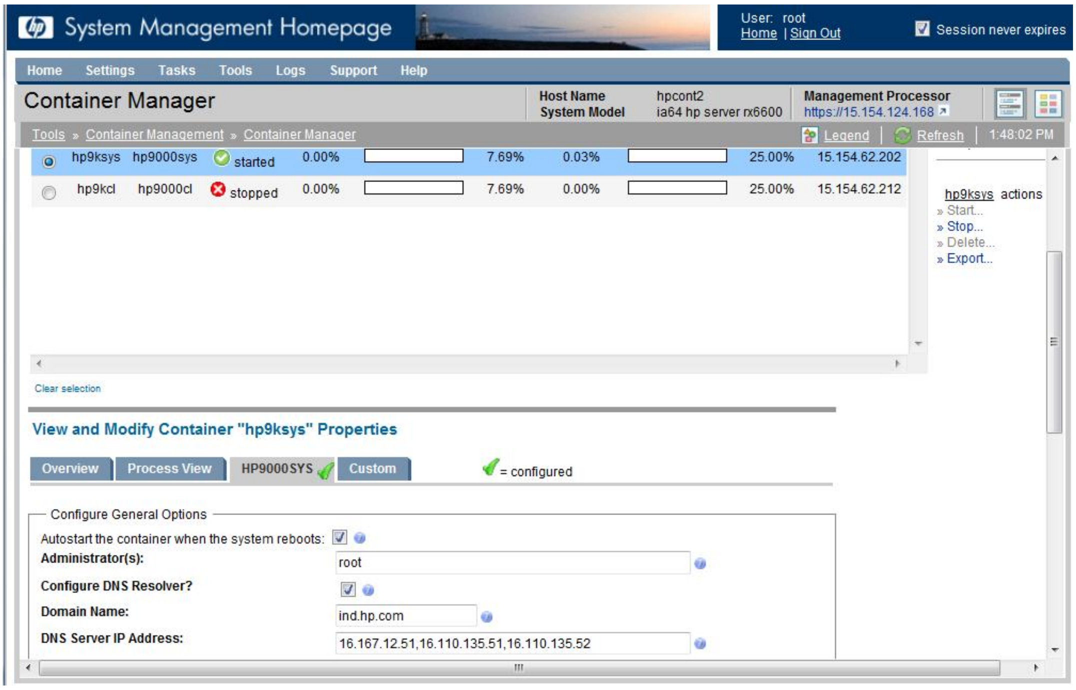
## 9.5 Viewing and modifying configuration

You can view or modify the configuration of a container once it has been created. Modifying configuration parameters requires the container be in the `stopped` state.

a) From the Container Manager home page, click the `Container Listing` tab and select the container to view or modify. The detailed view for the selected container is displayed below the `Container Listing` tab. The `Overview` tab displays the key properties. The `Process View` tab lists the processes running inside the container. The `Base` tab provides detailed configuration.

b) Select the configuration tab you wish to view or modify and make the changes.

c) To apply changes, click `Modify Container`. A result window displays output of the modification. After the operation completes, click `Close This Window`.



Host Name: hpcont2
System Model: ia64 hp server rx6600
Management Processor: https://15.154.124.168

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ● | hp9ksys | hp9000sys | started | 0.00% | 7.69% | 0.03% | 25.00% | 15.154.62.202 |
| ○ | hp9kcl | hp9000cl | stopped | 0.00% | 7.69% | 0.00% | 25.00% | 15.154.62.212 |

View and Modify Container "hp9ksys" Properties

Configure General Options

Autostart the container when the system reboots: ☑

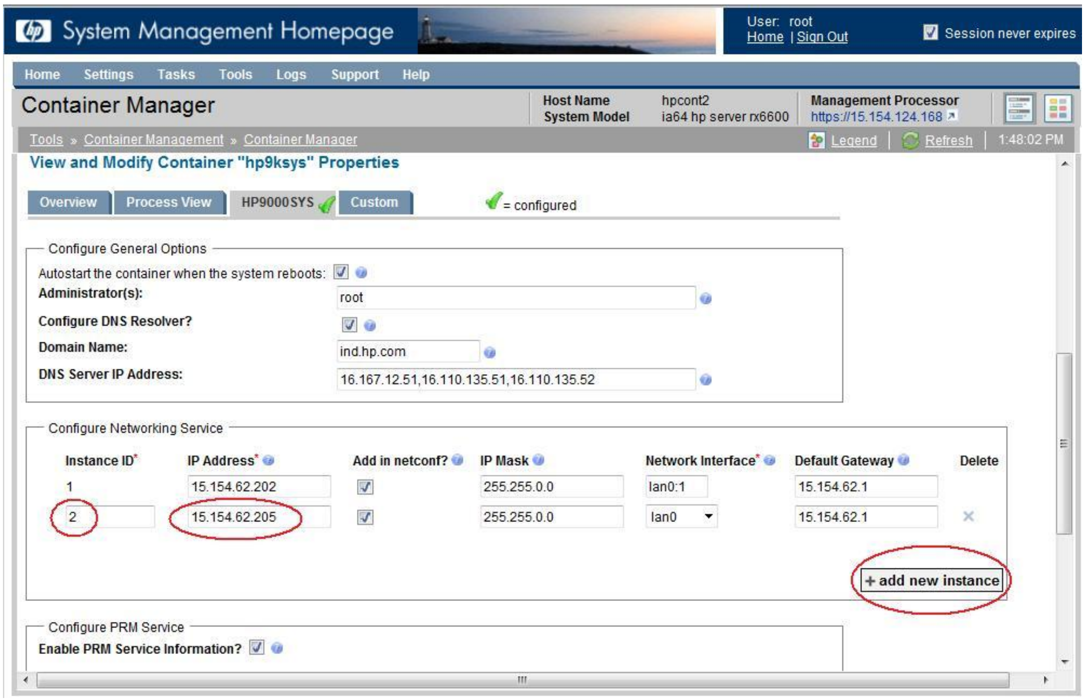
Administrator(s): root

Configure DNS Resolver? ☑

Domain Name: ind.hp.com

DNS Server IP Address: 16.167.12.51,16.110.135.51,16.110.135.52

**Figure 7.6:** Container Manager – viewing a container



Host Name: hpcont2
System Model: ia64 hp server rx6600
Management Processor: https://15.154.124.168

View and Modify Container "hp9ksys" Properties

Configure General Options

Autostart the container when the system reboots: ☑

Administrator(s): root

Configure DNS Resolver? ☑

Domain Name: ind.hp.com

DNS Server IP Address: 16.167.12.51,16.110.135.51,16.110.135.52

Configure Networking Service

| Instance ID* | IP Address* | Add in netconf? | IP Mask | Network Interface* | Default Gateway | Delete |
|---|---|---|---|---|---|---|
| 1 | 15.154.62.202 | ☑ | 255.255.0.0 | lan0:1 | 15.154.62.1 | |
| 2 | 15.154.62.205 | ☑ | 255.255.0.0 | lan0 | 15.154.62.1 | ✕ |

Configure PRM Service

Enable PRM Service Information? ☑

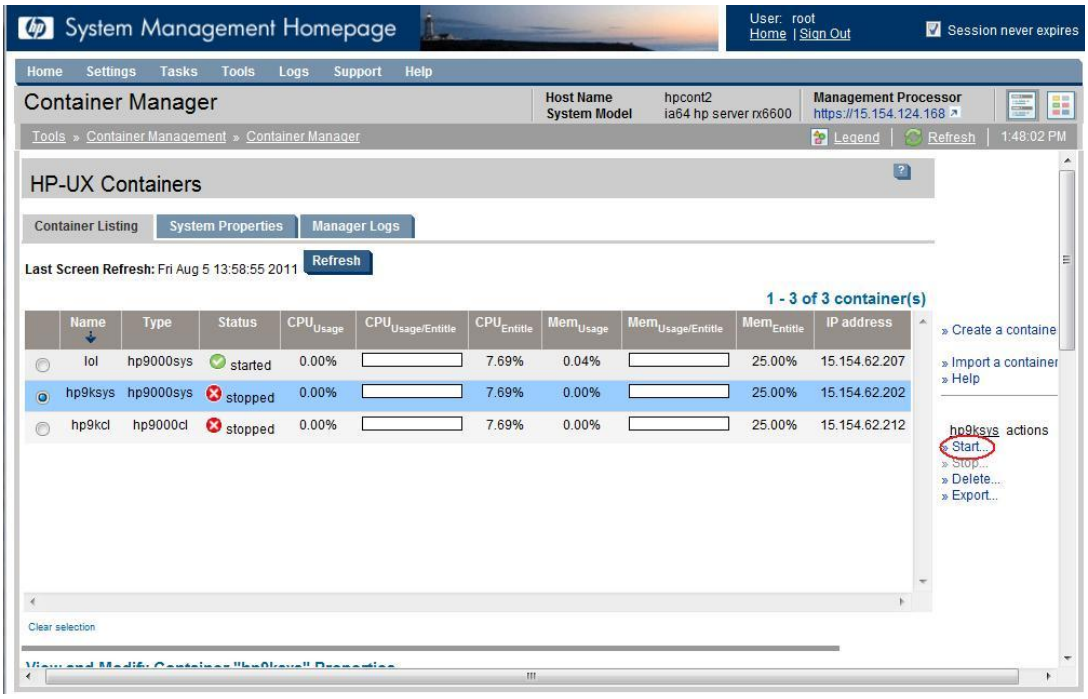
**Figure 7.7:** Container Manager – modifying a container

## 9.6 Starting and stopping a container

You can start a container if it is currently in the `stopped` state. The current state of each container is displayed on the Container Manager home page. From the `homepage`, follow these steps to start a container:

a) Click on the container that you want to start.
b) Click `Start` located in the right hand task bar.

Similarly you can stop the container if it is already in the `started` state.



Host Name: hpcont2
System Model: ia64 hp server rx6600
Management Processor: https://15.154.124.168

Last Screen Refresh: Fri Aug 5 13:58:55 2011

1 - 3 of 3 container(s)

| | Name | Type | Status | $\text{CPU}_{\text{Usage}}$ | $\text{CPU}_{\text{Usage/Entitle}}$ | $\text{CPU}_{\text{Entitle}}$ | $\text{Mem}_{\text{Usage}}$ | $\text{Mem}_{\text{Usage/Entitle}}$ | $\text{Mem}_{\text{Entitle}}$ | IP address |
|---|---|---|---|---|---|---|---|---|---|---|
| ○ | lol | hp9000sys | ✓ started | 0.00% | | 7.69% | 0.04% | | 25.00% | 15.154.62.207 |
| ● | hp9ksys | hp9000sys | ✗ stopped | 0.00% | | 7.69% | 0.00% | | 25.00% | 15.154.62.202 |
| ○ | hp9kcl | hp9000cl | ✗ stopped | 0.00% | | 7.69% | 0.00% | | 25.00% | 15.154.62.212 |

**Figure 7.8**: Container Manager – starting a container

# 10 Integration with Serviceguard

## 10.1 Overview

HP Serviceguard allows creation of high availability clusters of HP 9000 or HP Integrity servers. Serviceguard is not supported to run within an HP 9000 container. However, it can be installed in the `global` compartment (i.e. on the host system) and configure it with applications running inside containers.

With HP 9000 *system* containers, there are two models in which SG can be used

- The *application package model*, where HP 9000 containers will be up on all failover nodes but the monitored application will be running inside only one of the containers at any given time. The application will be failed over to the other container when need arises.
- The container (*SRP) package model*, where the HP 9000 container is active only on one node at a time. Failover happens at a container level – the entire container is shut down on the primary node and brought up on the failover node. RC scripts need to be written to start applications along with container startup.

With HP 9000 *classic* containers, only the *application package* model is now supported.

Configuring a Serviceguard cluster for use with HP 9000 Containers involves the following high level steps:

- Setup the Serviceguard cluster
- Do system configuration on each node in the cluster
- Decide which package model to use
- Decide which application controls file system and network interface
- Configure shared logical volumes
- Configure HP 9000 container on primary node
- Configure HP 9000 container on each failover nodes
- Create or migrate monitor script
- Create RC scripts to start applications (container (*SRP) package* only)
- Create or migrate Serviceguard package configuration
- Copy package configuration to failover nodes
- Enable the package and test failover

## 10.2 System Configuration

All nodes in the cluster need to be identical with respect to OE and patch levels. Follow installation and configuration instructions to prepare each node for HP 9000 Containers.

## 10.3 Deciding which failover model to use

- With HP 9000 *classic* container, only the application package model of Serviceguard integration is supported.
- With HP 9000 *system* container, the choice needs to be made based on the requirements. It will be easier to transition existing SG packages to the *application package model* and the failover can be quicker at application level. The *container package model* simplifies manageability and fresh packages are easier to create.

## 10.4 Deciding which application will have control

- If the *application package model* is being used, HP recommends that management of network interface and mounting of file systems for data be controlled by Serviceguard.
- If the *container package model* is being used, HP recommends that these be managed by HP-UX Containers.
- If you want to use the Serviceguard network failover capability, the network interface management should be done by Serviceguard.
- A Serviceguard managed container and a non-Serviceguard managed container on the same host should not share the same physical network interface.

## 10.5 Using container package model

This model is supported with HP 9000 *system* containers only.

### 10.5.1 Decide on the file system model

The container file system under `/var/hpsrp/<srp_name>` can be either on shared logical volume or replicated on all nodes. A shared file system provides better manageability and lower storage costs. However, it requires that the host system to be on the same OE update and patch level on all the nodes..

### 10.5.2 Configuring shared logical volumes

If the container file system is to be shared, configure a shared logical volume on the primary node, export the volume group configuration and import it on the failover nodes.

Do the same for hosting shared data, if applicable.

### 10.5.3 Configuring primary node

Steps described in Preparing to transition from HP 9000 server and in either of Creating an HP 9000 *system* container should be followed on the primary node. Note that there are specific instructions for integration with Serviceguard, when creating the container.

In particular during container creation, answer the following in negative

```
Autostart container at system boot? [yes]   no
```

If the container IP address needs to be managed by Serviceguard

```
Add IP address to netconf file? [yes] no
```

### 10.5.4 Writing RC scripts for applications

In the *container package model*, the container is failed over and started on the failover node. This does not start applications unless RC scripts are written to start them along with the container. For more details on creating RC scripts, refer to `rc` man page.

### 10.5.5 Testing applications on primary node

Follow the below steps to start and test the HP 9000 container

- If the container IP addresses are managed by Serviceguard, enable them manually, and add route entry (manually for testing)

  ```
  $ ifconfig <container-lan-interface> \
    <container-ip-addr> netmask <netmask>
  ```

  ```
  $ /usr/sbin/route add default <gateway-ip-addr> 1 \
    source <container-ip-address>
  ```

- Start the HP 9000 container

  ```
  $ srp -start <srp_name>
  ```

  Check `/var/hpsrp/<srp_name>/etc/rc.log` to verify that applications configured in RC scripts have been started properly.

- Login to the HP 9000 container and test applications.
- Once testing is complete, stop the HP 9000 container.

  ```
  $ srp -stop <srp_name>
  ```

  Check `/var/hpsrp/<srp_name>/etc/rc.log` to verify that applications configured in RC scripts have been stopped properly.

- If the container IP addresses are managed by Serviceguard, disable them and remove the route entry

  ```
  $ ifconfig <srp-lan-interface> 0
  ```

  ```
  $ /usr/sbin/route delete default <gateway-ip-addr> \
    1 source <srp-ip-addr>
  ```

### 10.5.6 Configuring failover nodes

- If the container file system is shared, export only the container configuration on primary node

  ```
  $ srp -export <srp_name> -xfile <path name of
    exchange file>
  ```

- If the container file system is not shared, include it in the export file

  ```
  $ srp -export <srp_name> ok_export_dirs=yes -xfile
    <path name of exchange file>
  ```

- Unmount the container file system and deactivate the volume group on primary node. Similarly for any shared volume data.
- Copy over the exchange file to failover nodes

  ```
  $ cmcp <exchange file> <failover node>:<exchange \
    file>
  ```

- Create root directory

  ```
  $ mkdir /var/hpsrp/<srp_name>
  $ chown root:sys /var/hpsrp/<srp_name>
  $ chmod 0755 /var/hpsrp/<srp_name>
  ```

- If the container file system resides on a logical volume

  ```
  $ mount <logical volume> /var/hpsrp/<srp_name>
  ```

- If not using shared file system, ensure that no users are configured on the failover node apart from the default set of users that come with the operating system.

- Import the container onto the failover node

```
$ srp -import -xfile <exchange file> autostart=no
```

- Configure kernel tunable parameters on failover node to match primary node
- Configure printers if applicable
- Install special device drivers, if any
- Install manageability software, if any
- Transition `/etc/privgroup` configuration from primary node (global) if applicable to the container being setup for failover.

### 10.5.7 Creating monitor scripts

The monitor scripts for applications need to be placed in some directory under `<hp9000_root>`. You may be able to just use the existing monitor scripts if they are compatible with the Serviceguard version on the Integrity server. Refer to Serviceguard documentation on how to migrate older packages.

### 10.5.8 Configuring SG package

A reference implementation of a Serviceguard package and `README` can be found under `/opt/hpsrp/example/serviceguard/srp_as_sg_package/srp_package.conf`.

The main aspects of this configuration are

- If the container file system is on shared volume, specify `/var/hpsrp/<srp_name>` as a SG managed file system.

  ```
  fs_name /dev/<vg_name>/container_lv> fs_directory \
  /var/hpsrp/<srp_name>
  ```

- Specify the monitor script to be executed inside the container.

  ```
  service_cmd "/opt/hpsrp/bin/srp_su <srp_name> <user>\
  -c "<command line for monitor script>""
  ```

- If the container IP addresses are to be managed by Serviceguard, specify the same

  ```
  ip_subnet <subnet>
  ip_address <IP address>
  ```

- If Serviceguard is managing the container IP addresses, HP recommends that the package be configured to create default routes for these. For example,

  ```
  # SG ip address
  ip_subnet  192.10.25.0
  ip_address 192.10.25.12

  # srp_route_script configures the required source
  # based routing entries for the SG managed IP
  # addresses
  external_script /etc/cmcluster/pkg1/srp_route_script
  ```

  A reference implementation of this script can be found under `/opt/hpsrp/example/serviceguard/srp_as_sg_package/srp_route_script`.

- Specify a control script for starting and stopping the container during failover.

  ```
      external_script /etc/cmcluster/pkg/srp_control_script
  ```

  A reference implementation of this script can be found under `/opt/hpsrp /example/serviceguard/srp_as_sg_package/srp_control_script`.

## 10.6 Using application package model

### 10.6.1 Configuring shared volumes

The application data may reside in shared volumes, but the container file system needs to be replicated on each node. There is no support for a shared container file system.

### 10.6.2 Configuring primary and failover nodes

HP 9000 containers need to be configured separately on each node in the cluster following the steps described in Preparing to transition from HP 9000 server and either of Creating an HP 9000 *system* container or Creating an HP 9000 classic container depending on the container type selected.  Configure a unique IP address and a unique *hostname* for the HP 9000 container on each server, but use same container name. Once the configuration is completed on each node, start each container and verify that applications run fine.

### 10.6.3 Configuring SG package

The configuration from the HP 9000 server can be re-used with minor modifications provided that the configuration is compatible with the Serviceguard version on the host system. Refer to Serviceguard documentation on details of how to migrate older packages.

Since Serviceguard is running in the `global` compartment and applications run within the HP 9000 container, the commands used to monitor and start applications (`service_cmd` configuration) need to be modified to use `srp_su`.

For HP 9000 *system* container, the configuration will look like

```
service_cmd ""/opt/hpsrp/bin/srp_su <srp_name> <user name>\
-c "<command line>""
```

For HP 9000 *classic* container, if the command is to be run as root user

```
service_cmd "/opt/hpsrp/bin/srp_su <srp_name> root -c \
"chroot <hp9000_root> <command line>""
```

For HP 9000 *classic* container, if the command is to be run as a non-root user

```
service_cmd "/opt/hpsrp/bin/srp_su <srp_name> root \
"chroot <hp9000_root> /usr/bin/su - <user> -c \
<command line>""
```

If Serviceguard is managing the container IP addresses, HP recommends that the package be configured to create default routes for these. For example,

```
# SG ip address
ip_subnet  192.10.25.0
ip_address 192.10.25.12

# srp_route_script configures the required source based
# routing entries for the SG managed IP addresses
external_script /etc/cmcluster/pkg1/srp_route_script
```

A reference implementation of this script can be found under
`/opt/hpsrp/example/serviceguard/srp_as_sg_package/srp_route_script`

## 10.7 Applying configuration

Follow the standard steps to apply configuration on primary node and test failover.

```
$ cmcheckconf –P <package configuration>
$ cmapplyconf –P <package configuration>
$ cmrunpkg –v –n <primary node name> <package>
```

Kill an application process manually, and see if the failover happens correctly.

# 11 HP 9000 Containers Limitations

## 11.1 Application Limitations

### 11.1.1 No support for kernel intrusive applications or those that use privileged instructions

Applications that are kernel intrusive and those that use privileged instructions will not work inside the HP 9000 container. This includes, for example, applications that read and write `/dev/kmem`, use DLKMs or install device drivers.

### 11.1.2 No support for system management applications

Management utilities like SAM/SMH and applications that do tasks related to system management or monitoring may not work inside the HP 9000 container. This includes HP *Openview* agents, and HP *Serviceguard*, among others. The recommendation is to run these applications outside the HP 9000 container (i.e. in the *global* compartment).

### 11.1.3 No support for system resource monitoring applications

Applications that do disk, memory or CPU monitoring and performance agents are not supported inside HP 9000 containers. These applications have to run in global.

### 11.1.4 No support for general applications in *global* compartment

It is advised that no applications be used in the global compartment when using containers. The exceptions are for those applications that are related to system administration or management including HP Serviceguard and HP Openview which are not supported inside the container.

### 11.1.5 No support for use as DHCP, DNS or NFS server as an IP router

DHCP, DNS and NFS servers are currently unsupported inside a container. Also, using the container as a router is not supported.

### 11.1.6 Other limitations

As documented in HP ARIES Limitations
http://h21007.www2.hp.com/portal/site/dspp/menuitem.863c3e4cbcdc3f3515b49c108973a801/?ciid=f6ccb52bdb779110VgnVCM100000275d6e10RCRD#limits

## 11.2 Setup Limitations

### 11.2.1 No support for DHCP

HP 9000 containers require static IP addresses. There is no support for using DHCP for container IP address currently.

### 11.2.2 No support for large base page sizes

Applications running in an HP 9000 container on an HP-UX 11i v3 system where kernel tunable parameter `base_pagesize` is configured to a non-default value may experience correctness issues with symptoms such as application hangs and aborts. The recommendation is to set kernel parameter `base_pagesize` to its default value of 4 KB.

### 11.2.3 No support for large PID, UID, host name with legacy containers

Legacy (pre HP-UX 11i v3) containers may not have the necessary user space components to support large values of PIDs, UID/GIDs, host name/node name. Hence such large values cannot be supported with HP 9000 containers built from such environments. In particular, the `process_id_max` on the HP-UX 11i v3 host system may need to be set to 30000 even though the kernel can support values up to a 1073741823.

### 11.2.4 HP 9000 *classic container* specific limitations

- No support for multiple HP 9000 containers.
- No support for co-existence with native containers.
- Some server applications which register RPC services may need a virtual hostname configuration (that matches the host) to work. Refer to Known issues and workarounds for more details.

## 11.3 Access Limitations

### 11.3.1 HP 9000 *classic container* specific limitations

Using `telnet` to login to an HP 9000 *classic* container is not supported. Use `ssh` instead. If `telnet` is used the user login will be placed in the *global* compartment and will be unable to run applications. However `telnet` from the HP 9000 container to other servers is supported.

Using remote commands such as `remsh, rlogin, rcp, rexec` to access the HP 9000 classic container is not supported. Secure shell (`SSH`) based protocols (`ssh`, `slogin`, `scp`) may be used along with authorization keys to achieve similar functionality.

## 11.4 Patching Limitations

For HP 9000 *classic container*, there is no support for SD patching inside the container

For HP 9000 *system* container, the following limitations apply

- If compartment rules are used to restrict unsupported commands (not default), patching would fail for products that include these commands.
- `swverify and check_patches` may report errors inside an HP 9000 container.
- Patching is not supported for libraries and commands that have been switched using the `libv3` and `cmdv3` templates.

For more details and specifics refer to the section on HP 9000 container patching

## 11.5 User management limitations

For HP 9000 *system* container, the following limitations apply to user management.

- There is no support for user quota.
- Do not configure users in the *global*, apart from those related to system administration or system management related applications.

For an HP 9000 *classic* container, the following limitations apply to user management

- The users have to be configured on host and access to container managed via RBAC.

- SSH authorization keys require home directories to be created on the host as well (just to store the keys).

## 11.6 Commands Limitations

The following commands are known to fail inside both `system` and `classic` HP 9000 containers

```
$ df -k .
```

The following commands are known to provide global information (not container specific data) inside an HP 9000 *classic* container, due to the file sharing described in HP 9000 classic container file system

- `bdf` (also reports errors for loopback mounts)
- `df`
- `last`
- `lastb`
- `mount`
- `who`
- `finger`

## 11.7 Unsupported Tasks

The following list captures the major tasks that are unsupported and disabled inside an HP 9000 container. Most of them are related to system administration and can be performed outside the container (in the *global* compartment).

- Assembly debugging
- Accounting enable/disable
- Auditing enable/disable
- CacheFS
- CIFS client
- Cluster management
- Compartment rule configuration
- Date and time setting
- DHCP configuration
- Device creation
- Disk management
- Driver installation and management
- Event monitoring Service
- File system management and export
- Global Instant Capacity Management
- HP-UX Containers (SRP) creation and management
- Interrupt configuration
- IP address configuration
- IPFilter. IPSec configuration
- Ignite-UX
- Kernel debugging
- Kernel make
- Kernel memory read
- Kernel module (DLKM) administration
- Kernel registry services
- Kernel tunable management
- LIF
- Logical and physical volume management
- Make kernel
- Network tunable configuration

- NIC administration
- NFS server and exports/shares
- NLIO
- NTP
- OLAR
- Partition management
- Portable file system
- Printer management (*classic container* only)
- Privilege management
- Process Resource Management (PRM)
- Processor set management
- Processor binding
- RAID control
- Reboot, shutdown system
- Resource (CPU, memory, disk etc) monitoring
- Routing configuration. advertisement
- SAM, SMH
- SCSI control
- Serviceguard
- Software Distributor based installation and patching (*classic container* only)
- Storage/Disk management
- STREAMS administration
- Support Tools Manager (STM)
- Swap space management
- System activity reporter
- System boot configuration
- System crash configuration
- System diagnostics and statistics
- Update-ux
- VxFS, VxVM, and Volume Replicator related tasks

# 12 HP 9000 Containers Troubleshooting

## 12.1 Checking HP 9000 container health

### 12.1.1 Checking container status

Login to the *global* compartment as `root` and run

```
$ srp -status <srp_name> -v
```

Verify if connectivity to the HP 9000 container is working fine both from within the *global* compartment and from another system.

### 12.1.2 Checking container startup logs

Check `/var/hpsrp/<srp_name>/etc/rc.log` to verify if the previous startup/ shutdown went fine. Search `/var/adm/syslog/syslog.log` for '`SRP`' to know the list of operations that have been applied to the compartment.

### 12.1.3 Checking container configuration

Login to the *global* compartment are run

```
$ srp –list <srp_name> -v
```

### 12.1.4 Checking PRM configuration and statistics

```
$ prmlist –g –s
$ prmmonitor
```

### 12.1.5 Checking network data

Login to the *global* compartment are run

```
$ netstat –in
```

Verify that the container network interfaces are up. Also, verify that there is a default route entry with the container IP address as gateway using

```
$ netstat –rn
```

### 12.1.6 Checking kernel parameters

Legacy containers cannot support large PIDs and large base page size.

```
$ kctune base_pagesize=4
$ kctune process_id_max=30000
$ kctune nproc=30000
```

Some legacy 32 bit applications also expect a lower value for `shmmax`

```
$ kctune shmmax=0x40000000
```

### 12.1.7 Checking host name/node name

Legacy containers are known to have issues when long host name or node name is in use. Check `/etc/rc.config.d/netconf` and `/var/hpsrp/<srp_name>/etc/rc.config.d/netconf` for these parameters.

## 12.2 Triaging HP 9000 container access issues

Check /etc/rc.config.d/netconf on the host HP-UX 11i v3 server and ensure that the IP address, gateway and subnet mask for the container is configured correctly. Refer to Modifying IP address configuration on how to change these values if needed.

Check /opt/ssh/etc/sshd_config on host and ensure that ListenAddress parameter is set to host IP address. It should not be commented out, nor set to any container IP address.

For SSH failures to HP 9000 *classic* type *containers*, check

a) ListenAddress in /var/hpsrp/<srp_name>/opt/ssh/sshd_config should be set to the container IP address.

b) ChrootDirectory in /var/hpsrp/<srp_name>/opt/ssh/sshd_config should be set to the root directory where HP 9000 files have been recovered (<hp9000_root>)

c) The host HP-UX 11i v3 root directory (/) should have 755 permissions and root:sys or root:root ownership.

d) Similarly, all other directory components of the path leading up to <hp9000_root> should have 755 permissions and root:sys or root:root ownership.

For SSH access issues to HP 9000 *system* containers, check the following

a) Does sshd_config have UseDNS set? If so, is the DNS server accessible from the container and is the container host name in DNS? Does setting UseDNS to no help?

b) Does sshd_config have PermitRootLogin set to no?

c) Do host keys need to be re-generated and used on the clients?

d) Is the routing configuration correct?

e) Does upgrading SSH version inside the container help?

There is a known problem that if the node name on global (host system) is longer than 8 characters in length (and kernel parameter expanded_node_host_names is set to 1), then only two login sessions are allowed to legacy containers on the system. The workaround is to do

```
$ kctune uname_eoverflow=0
```

For telnet failures to HP 9000 *system* containers, check the value of kernel parameters npty and nstrpty are sufficient. Also, check if there are enough /dev/pty and /dev/pts devices exposed to the container. If not, try

```
$ kctune npty=<new value>
$ insf -e -n <new npty value>
$ srp -add <srp_name> -tune device=/dev/pty/*
```

Legacy inetd services are known to fail with large PIDs. Ensure that kernel tunable parameter process_id_max is set to less than or equal to 30000.

Do note that login using srp_su <srp_name> will work if the HP 9000 *system* container is in started state, even if the login services are not functioning properly. This can be used for debugging purposes. It will be useful to get a tusc log on sshd or inetd as described in the next section.

## 12.3 Collecting application and system call logs

- Check application logs and files where stdout /stderr have been re-directed.
- Install HP-UX system call tracer utility tusc for HP-UX 11.31/Itanium on the host. It can be downloaded from http://hpux.connect.org.uk/hppd/hpux/Sysadmin/tusc-8.1

- Copy /usr/local/bin/tusc binary to the HP 9000 container root directory
- Login to the HP 9000 container and run tusc on the failing application

```
$ tusc -o <output file path> -lfpkaev     \
  -s \!sigprocmask,sigaction,sigsetreturn \
  <executable> <arguments>
```

- Search the tusc log for clues like failing system calls. Check if any of the HP 9000 container limitations are encountered. For example, analyze execve() calls to see if any unsupported command is being invoked.

## 12.4 Debugging applications

It is possible to use PA-RISC HP WDB to debug applications inside a container, just like on the HP 9000 server. The only additional requirement is to set the following environment variable before invoking gdb.

```
$ export PA_DEBUG=1
```

ARIES generates PA-RISC HP-UX core files when the application aborts. And WDB can be used to analyze these core files as well.

Note that it is also possible to re-compile and link applications as well, provided the required compilers and tools are available inside the container.

## 12.5 Known issues and workarounds

The patches and products listed in Recommended patches resolve some of the known issues when using containers. The list below captures some known issues and workarounds.

- Legacy lsof command is known to fail inside HP 9000 containers. You may need to install lsof on the host system and copy /usr/local/bin/lsof into the container.

- Some java 1.3 based legacy applications (such as TIBCO) may fail to run inside an HP 9000 container. Upgrade to java 1.4.2 generally works for such applications. This has to be attempted only after the application failure is encountered.

- UDP broadcast messages may not reach the container. This issue is frequently encountered is when using TIBCO rendezvous agent inside the container. Check with HP on availability of fix or workaround.

- Local to local container communication does not honor subnet mask when selecting source IP address. Check with HP on availability of fix or workaround.

- *HP GlancePlus* returns no information for PRM groups configured with PSETs. The alternative is to use prmmonitor command instead of GlancePlus to monitor resource usage.

- *IBM Informix Dynamic Server* is known to intermittently hang inside an HP 9000 container if parameter NUMCPUVPS is larger than 1. The workaround is to set this parameter to 1.

- *Progress Database* is known to sometimes hang or crash running in emulation mode. Check with HP on availability of fix or workaround.

- *Oracle Database* is known to have emulation issues when parameter parallel_automatic_tuning is set to TRUE.

- *Oracle Database* is sometimes known to fail with ORA-0600 errors. Check with HP on availability of any workaround.

- PRM FSS cannot be used along with *Oracle Database Resource Manager*. See http://h20338.www2.hp.com/enterprise/downloads/PRM-Oracle_white_paper.pdf. The workaround is to switch to PSETs if the resource manager is in use.

- *HP Data Protector* reports errors when run inside an HP 9000 container. Check with HP on the availability of a workaround.

- With old versions of *SAP*, the `stopsap` *command* is known to hang and produce a core dump of the `sapstart` process. Upgrading to SAP kernel **1773** patch generally solves the issue. You may also contact HP-UX support for a workaround.

- Old versions of `Connect Direct` are known to fail when kernel parameters `maxfiles` is larger than `2048`. Set `maxfiles` and `maxfiles_lim` parameters to `2048` to work around this issue.

- Old versions of java (before 1.4.2.28) may fail to run inside a container with error message like *java.lang.InternalError: URLSeedGenerator file:/dev/random generated exception: Permission denied].* This issue can be worked around by commenting out the line `securerandom.source=file:/dev/random` in the `java.security` file in `<java_home>/jre/lib/security.`

- Some of the terminal settings may be lost when moving to an HP 9000 container. For example, `Ctrl-C` may no longer interrupt processes when logged in using `telnet` or `rlogin`. If this issue is observed inside the container, edit `/etc/profile` to invoke `stty` for required settings.

- There is no `telnet` access to an HP-UX 10.20 *system* container. A workaround is to copy `telnetd` and its dependencies from HP-UX 11i v1 or HP-UX 11.00 system to the container.

- If mounts for a container are configured in global `/etc/fstab`, they do not appear in the output of `bdf` command inside the container post reboot. Also, subsequent `umount` operations may report errors. The workaround is to configure container pre-start mounts as described in Configuring mount and export points

- The `hp9000_conf_tunables` script does not add up parameters when multiple containers are being created on the same host. There may be a need to increase parameters such as `npty` or `maxfiles` to accommodate all users and applications in many containers.

- When auditing is enabled with HP 9000 *system* containers, the login and logoff events do not get recorded. The workaround is to write a init.d script that executes the following commands after reboot

  ```
  echo ”audit_en_logins_compat/W 1" | adb -o -w
  /stand/vmunix\ /dev/kmem
  ```

  ```
  echo "audit_logoff_compat/W 1" | adb -o -w
  /stand/vmunix\ /dev/kmem
  ```

- The “`srp -stop`” operation sometimes returns before all processes are killed. Issuing a second or third “`srp -stop`” generally works. The most common case for this occurs when AUTOFS is enabled. If there is no requirement, turn off AUTOFS in `/etc/rc.config.d/nfsconf` inside the container.

- The “`srp -export`” operation does not include files that are larger than 8 GB in size. These files have to be backed up separately.

- The “`srp -import`” operation is known to change ownership of files if same users exist on the host system with different UIDs (or same groups with different GIDs). It is advised that during import, no users be configured on the host system apart from the default users.

- Commands with argument strings larger than 768 KB are known to fail inside HP 9000 containers. For example, "`ls *`" on a directory with a large number of files. A workaround is to copy the command from the host into the container.
- There is no `telnet`, `ftp`, `remsh`, `rlogin`, `rexec` or `rcp` access to a *classic container*. The workaround is to install `xinetd` on the HP 9000 server and create the container again. To configure `xinetd` you can run the script

```
$ /opt/HP9000-Containers/bin/hp9000_xinetd_setup \
   <srp_name>
```

- Some server applications may fail to start up inside a *classic* container and may throw errors such as "`unable to register RPC service`". For such applications, try configuring ARIES with a virtual host name that matches the Integrity host name. Include in `/.ariesrc` (32-bit) or `/.aries64rc` (64-bit)

```
<executable path> -cmpt_host_name <name of the host 11i
v3 system>
```

## 12.6 Troubleshooting HP ARIES

- If application complains about thread creation or stack growth related failures, refer to Tweaking ARIES configuration for resolution details.
- Update HP ARIES patch to the latest version. Every patch brings in defect fixes that can save a lot of troubleshooting effort.
- Disable ARIES optimizations by including the following lines in `/.ariesrc` (for 32-bit) or `/.aries64rc` (for 64-bit)

```
# Disable optimizations
<executable-path> -noopt
# End
```

- Disable ARIES translations by including the following lines in `/.ariesrc` (for 32-bit) or `/.aries64rc` (for 64-bit). This is just for testing purposes and cannot be used as a workaround since it slows down the applications significantly.

```
# Disable translation for testing
<executable-path> -notrans
# End
```

- If ARIES complain of heap exhaustion, refer to ARIES man page (`man 5 aries`) on how to configure larger values for these parameters. ARIES man page is accessible on host system only.
- Refer to the T*roubleshooting* section on HP ARIES web page http://www.hp.com/go/ARIES.
- Contact HP-UX support center for assistance with troubleshooting ARIES issues.

## 12.7 Troubleshooting HP-UX Containers

Refer to the *Verifying and troubleshooting container section* in the *HP-UX Containers A.03.01 Administrator's Guide*. Known defects and workarounds are documented in the *HP-UX Containers A.03.01 Release Notes.*

## 12.8 Re-configuring an HP 9000 container

### 12.8.1 Re-doing HP 9000 container configuration

If there are suspected issues inside the HP 9000 container file system then login to the global and run following commands

```
$ srp -stop    <srp_name>
$ srp -replace <srp_name>
$ srp -start   <srp_name>
```

### 12.8.2 Switching to newer HP 9000 libraries

Applications inside the HP 9000 container, by default, use system libraries brought over from the HP 9000 server. However it can be configured to use PA-RISC libraries shipped with Integrity HP-UX 11.31 instead. This enables a newer set of libraries to be used in the HP 9000 container with potentially more defect fixes.

The HP 9000 container needs to be stopped before the libraries are switched. Login to the *global* compartment as root and run following commands

```
$ srp -stop <srp_name>
$ srp -add  <srp_name> -t libv3 -b
```

The process may take about 10 minutes as the PA-RISC HP-UX 11i v3 libraries are copied into the HP 9000 container and any application libraries are merged in.

**NOTE**: Once switched, patching these libraries inside the container will not be supported. If the files are overwritten as a result of patching, they need to be recovered manually. Also there is no automatic copying into the container when these libraries are patched on the host. It is possible to run the `replace` operation to copy all the latest libraries again, but it requires a container downtime,

To re-copy the set of libraries from the host system

```
$ srp -stop    <srp_name>
$ srp -replace <srp_name> -t libv3
```

To switch back to original libraries, use

```
$ srp -stop   <srp_name>
$ srp -delete <srp_name> -t libv3
```

### 12.8.3 Restoring restricted HP 9000 commands

HP 9000 Containers restricts set of commands in two ways inside the container

(a) Replace unsupported commands with a dummy command that exits with an error message. In this case, the original HP 9000 commands are backed up under `/sbin-hp9000` (for `/sbin` commands) or under `/var/opt/HP9000-Containers` inside the container. These may be copied back for testing. For example,

```
$ cp -p /var/opt/HP9000-Containers/usr/sbin/<command>
/usr/sbin/
```

(b) Use compartment rules to disallow execution. To allow execution of the command, remove the entry of command from file `hp9000.disallowed.cmds`. Then run the command

```
$ setrules
```

If the command is found to work fine, remove the entry for the command from /var/opt/HP9000-Containers/hp9000sys_delete_commands inside the container. Also, remove the entry from /opt/HP9000-Containers/config/hp9000sys_delete_commands and /opt/HP9000-Containers/config/hp9000.disallowed.cmds

## 12.9 Performance tuning

### 12.9.1 Switching to Integrity native commands

If the application is heavily script intensive, it might take significant performance hit when run inside an HP 9000 container. This is because scripts are usually short living processes and have flat execution profiles and do not suit emulation. Such applications are not very common but do exist. It is possible to switch to using Integrity native versions of shells and certain commands, if needed.

The list of commands that are copied can be found in `/opt/HP9000-Containers /config/hp9000_switch_commands`

To switch commands

```
$ srp -stop  <srp_name>
$ srp -add   <srp_name> -t cmdv3 -b
```

**NOTE**: Once switched, patching these commands inside the container will not be supported. If the files are overwritten as a result of patching, they need to be recovered manually. Also there is no automatic copying into the container when these commands are patched on the host. It is possible to run the `replace` operation to copy all the latest commands again, but this needs a container downtime,

To re-copy commands

```
$ srp -stop    <srp_name>
$ srp -replace <srp_name> -t cmdv3
```

To switch back to HP 9000 commands

```
$ srp -stop   <srp_name>
$ srp -delete <srp_name> -t cmdv3
```

### 12.9.2 Tuning ARIES emulation

For some applications it may be possible to improve performance with some ARIES tuning. The ARIES configuration parameters are specified in `/.ariesrc` for 32-bit applications or `/.aries64rc` for 64-bit applications. The configuration files may also reside in user home directory or application directory, if need be,

Enable trace scheduling

```
# Enable trace scheduling
<executable-path> -sched_trace
# End
```

Reduce memory overhead

```
# Reduce memory footprint
<executable-path> -mem_min
# End
```

**NOTE**: These configurations can have adverse impact on performance in some cases and hence proper testing is recommended before enabling these in production.

### 12.9.3 Tuning kernel parameters

- Install HP-UX kernel patch PHKL_41967, if not already available, to avoid encountering known `select()` performance issue.
- For some applications, or if there is heavy memory pressure, reducing the parameters `filecache_min` and `filecache_max` to 5-10% of physical memory might help.

```
$ kctune filecache_min=5% filecache_max=10%
```

- Compare the kernel parameters against the HP 9000 server
- Contact HP-UX support center for assistance with kernel tuning.

### 12.9.4 Profiling ARIES emulation

It is possible to use HP Caliper to profile ARIES emulation of an application running inside a container from its startup. However, it is not possible to attach caliper to an ARIES emulated process.

- Install caliper on the host system (global) and copy the files into the container

  ```
  $ cp -p -r  /opt/caliper  /var/hpsrp/<srp_name>/opt
  ```

- Create a directory to hold ARIES profiles

  ```
  $ mkdir /tmp/ARIES_profdb
  ```

- Modify the startup command line to include

  ```
    export PA_BOOT32_DEBUG=3

    export PA_BOOT64_DEBUG=3

    export CALIPER_HOME=/opt/caliper

    $CALIPER_HOME/bin/caliper fprof -r a \
    --database=/tmp/ARIES_profdb/db,unique \
    --scope=process \
    --des=all \
    --process=all \
    --thread=all \
    --output-file=/tmp/ARIES_profdb/aries_profile.txt,per-
  process,unique \
  <Application executable or startup script> <arguments>

    unset PA_BOOT64_DEBUG

    unset PA_BOOT32_DEBUG
  ```

- Run the test and save `/tmp/ARIES_profdb`
- Contact HP support for detailed analysis of the report

# For more information

- HP Integrity family
  http://www.hp.com/go/integrity
- HP-UX 11i v3
  http://www.hp.com/go/hpux11i
- HP OverEasy portfolio
  http://www.hp.com/go/overeasy
- HP ARIES dynamic binary translator
  http://www.hp.com/go/aries
- HP-UX Containers
  http://www.hp.com/go/containers
- HP 9000 Containers
  http://www.hp.com/go/hp9000-containers
- HP 9000 Containers Software Access
  http://software.hp.com
  » HP-UX 11i Software » HP-UX 11i general » HP 9000 Containers
- HP Process Resource Manager
  http://www.hp.com/go/prm
- HP ID-VSE for Integrity servers
  http://www.hp.com/go/vse
- HP Serviceguard for HP-UX 11i
  http://www.hp.com/go/serviceguard
- HP-UX Manuals
  http://www.hp.com/go/hpux-core-docs

Oct 2012