Charon-PAR Version 3.0 User's Guide

(Product version 3.0.6 and higher)

Contents

About this Guide

Purpose of this Document

This document covers **Charon-PAR version 3.0** (version 3.0.6 and higher). For other versions, please refer to the relevant documentation of your product version.

This user's guide covers all aspects of selecting a suitable environment (hardware and software) to run the Charon-PAR emulator. It also describes all tasks involved with installing, configuring, and running the Charon-PAR emulator.

The following topics are not covered by this guide:

- While this document will cover guest system specific configuration aspects where required, it does not cover the management of the guest system (MPE/iX and HP-UX).
- It also does not cover the migration of a physical PA-RISC system to an emulated system. If you require migration support, please contact your Stromasys representative or a Stromasys partner.

Product naming conventions:

- **Charon-PAR**: general name for this Charon emulator product.
- **Charon-PAR/PA3**: functionality of the Charon-PAR product emulating historic PA-RISC systems for MPE/iX.
- **Charon-PAR/PA9-64 and Charon-PAR/PA9-32**: functionality of the Charon-PAR product emulating historic PA-RISC systems for HP-UX. Here, PA9-64 denotes the 64-bit emulator, PA9-32 the 32-bit emulator.

Intended Audience

This document is intended to be used by any end-user or Stromasys partner required to install, configure, and manage a Charon-PAR installation. It is assumed that a reader of this document has a basic knowledge of managing Linux systems and is familiar with the guest operating system used.

Content Overview

The following are the main topics covered in this user's guide. The order in the document may be different - as the items in the present section are arranged by subject.

- Introduction to Charon-PAR
- Emulated Models Supported by Charon-PAR: basic overview of amount of memory and maximum number of CPUs supported by the different emulated models. Details for each model are provided in Emulated Model Hardware Configuration Details and Configuration File Reference.
- Charon-PAR Prerequisites: requirements that must be fulfilled by the host system before Charon can be installed.
- Charon-PAR Software Installation: installation overview and installation steps.
- Initial Emulator License Configuration: overview of licensing options and initial license configuration.
- Charon-PAR Basic Guest Configuration and Installation: basic steps of setting up a CHARON-PAR/PA3 system (emulates historic PA-RISC hardware for MPE/iX), or a Charon-PAR/PA9 system (emulates historic PA-RISC hardware for HP-UX), and a sample HP-UX installation.
- Detailed configuration information:
  - Serial Line Emulation Notes: information about how to set up different terminal emulators.
  - Guest System Autoboot
  - Running Charon-PAR as a Service
  - Adding Additional Disks
  - Physical and Virtual Tape Access
  - Adding Generic SCSI Devices
  - HP-UX Graphical Connection via X-Server: X11 configuration details for host and guest system.
  - Ethernet Configuration Notes
  - Configuring SuperIO Devices
  - Configuration File Reference: detailed description of available configuration parameters.
  - Emulated Model Hardware Configuration Details
- Charon-PAR Command-Line Options: command-line options provided by the Charon-PAR emulator image.
- File Transfer to and from Guest Operating System: hints on different file transfer methods that may be useful during a migration situation.
- Upgrading the Charon-PAR Software and Deinstalling the Charon-PAR Software
- License Management Overview for Charon-PAR: overview of the different tools available for license management, and important configuration options.
- Charon-PAR Utilities: description of additional command-line utilities.
- Emulated System Firmware Functions: overview of options available on the console of an emulated system.
- HP3000 Command Files & UDCs: additional tools provided on the MPE pre-installed disk.

Obtaining Documentation

The latest released version of this manual and other related documentation are available on the Stromasys support website at Product Documentation and Knowledge Base.

Obtaining Technical Assistance or General Product Information

Obtaining Technical Assistance

Several support channels are available to cover the Charon virtualization products.

If you have a support contract with Stromasys, please visit http://www.stromasys.com/support/ for up-to-date support telephone numbers and business hours. Alternatively, the support center is available via email at support@stromasys.com.

If you purchased a Charon product through a Value-Added Reseller (VAR), please contact them directly.

Obtaining General Product Information

If you require information in addition to what is available on the Stromasys Product Documentation and Knowledge Base and on the Stromasys web site y ou can contact the Stromasys team using https://www.stromasys.com/contact/, or by sending an email to info@stromasys.com.

For further information on purchases and the product best suited to your requirements, you can also contact your regional sales team by phone:

Region	Phone	Address
Americas	+1 919 239 8450	Stromasys LLC 871 Marlborough Ave, suite 100, Riverside CA 92507 USA
Europe, Middle-East and Africa	+41 22 794 1070	Avenue Louis-Casai 84 1216 Cointrin Switzerland

Conventions

Notation	Description
$	The dollar sign in interactive examples indicates an operating system prompt for VMS. The dollar sign can also indicate non superuser prompt for UNIX / Linux.
#	The number sign represents the superuser prompt for UNIX / Linux.
>	The right angle bracket in interactive examples indicates an operating system prompt for Windows command (cmd.exe).
`User input`	Bold monospace type in interactive examples indicates typed user input.
<**`path`**>	Bold monospace type enclosed by angle brackets indicates command parameters and parameter values.
`Output`	Monospace type in interactive examples, indicates command response output.
[ ]	In syntax definitions, brackets indicate items that are optional.
...	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
`dsk0`	Italic monospace type, in interactive examples, indicates typed context dependent user input.

Definitions

Term	Description
Host	The system on which the emulator runs, also called the Charon server
Guest	The operating system running on a Charon instance, for example, Tru64 UNIX, OpenVMS, Solaris, MPE or HP-UX

Related Documents

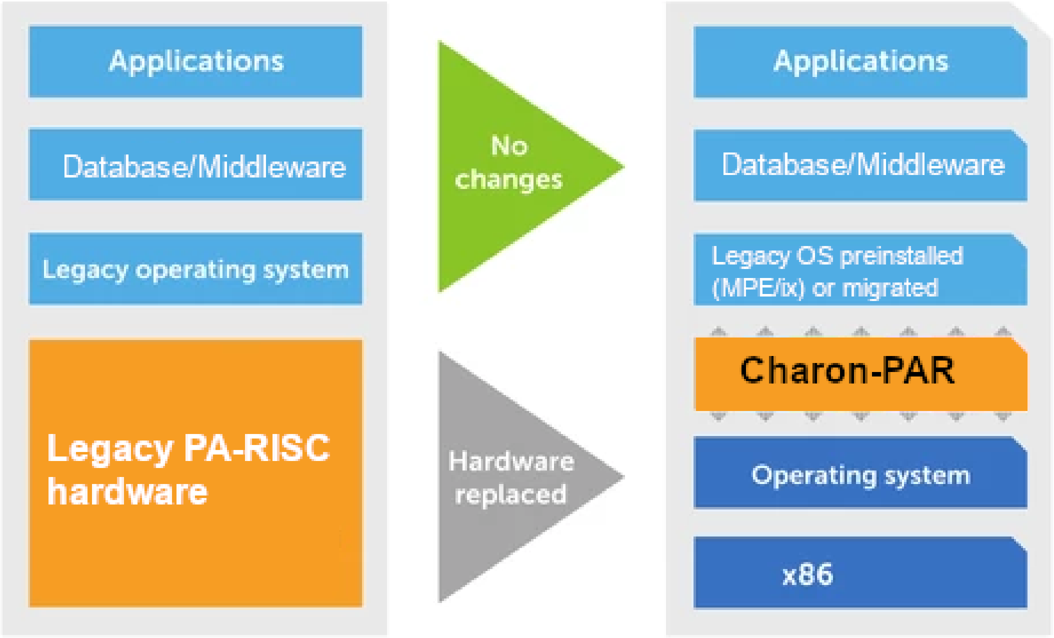
- CHARON-PAR/PA3/PA9 for Linux
- Charon-PAR for Linux - Older Versions

Introduction to Charon-PAR

Charon-PAR is a hardware virtualization layer running under Linux on industry standard servers. It emulates a range of historic 64-bit and 32-bit PA-RISC hardware and allows existing users of such systems to move to modern Intel-based server hardware.

Modern software operating systems contain a hardware abstraction layer (HAL). The HAL creates a software layer on top of the hardware to "virtualize" the functionality of the hardware components. The Charon-PAR products are essentially HALs of complete legacy hardware, including its PCI-based I/O devices. They are precise models of the legacy hardware, and contain modules which emulate the legacy hardware CPUs, console subsystem, buses and I/O adapters, disks and tapes.

The installation of Charon-PAR on a general purpose host platform provides an exact model of the historic PA-RISC hardware. On this 'virtual' system you install your legacy operating system and the associated applications, just as though you were using the original hardware. In most cases, no changes of the software are required. The Charon-PAR emulated systems run the same binary code and the same I/O drivers as the original hardware. This is illustrated in the following image:



Emulated Models Supported by Charon-PAR

Contents

Emulated Models Supported by Charon-PAR/PA9

The following table provides a short overview of the historic PA-RISC models for HP-UX emulated by the Charon-PAR software. For more details regarding each model, please refer to *Emulated Model Hardware Configuration Details* in the Charon-PAR user's guide. For configuration details and restrictions, please refer to the section *Configuration File Reference* in the Charon-PAR user's guide.

Please note: some emulated devices (e.g., tape drives, serial, and parallel ports) can be linked to physical devices on the Charon host system. As the physical devices on a cloud-based Charon host system typically do not exist, such configurations are not available in cloud environments.

Model	Maximum Memory	Number of CPUs	SCSI contr.	Ethernet contr.	Serial/ par. ports	Expansion slots
64-bit models						
A400 [1], rp2400	2GB	1	Dual SCSI-2 contr. (LSI 53C8xx) and Dual SCSI-3 LVD contr. (LSI 53C8xx)	DEC 21143-PD Tulip 10 /100 Mbit/s	Diva serial PCI card (2 ports) and optional Super-IO with 2 serial and one par. port	2
A400 [1], rp2430						
(A400), rp2405						
A500 [1], rp2450	8GB	1 - 2				4
A500 [1], rp2470						
(A500), rp2405						
rp5400	8GB	1 -2				6
rp5450	16GB	1 - 4				10
rp5430	8GB	1 - 2				6
rp5470	16GB	1 - 4				10
N4000 [1], rp7400	32GB [3]	1 - 8 [3]				12
rp3410 (from v3.0.5)	6GB	1 dual-core	Dual SCSI-3 LVD contr. (LSI 53C1010)			6
rp3440 (from v3.0.5)	32GB [2]	1-2 dual-core [2]				
rp4410 (from v3.0.5)	128G	1-2 dual-core				
rp4440 (from v3.0.5)	128GB [2]	1-4 dual-core [2]				

Model	Maximum Memory	Number of CPUs	SCSI contr.	Ethernet contr.	Serial/ par. ports	Expansion slots
32-bit models						
9000/720 [6]	265MB	1	Cobra Core SCSI (53C7xx)	Cobra Core LAN (802.3)	2 Cobra Core RS-232	0
9000/B132L [4]	2GB [5]	1 [4]	LSI 53C710	Intel 82596, 10 Mbit/s	1 RS-232 serial port	0

[1] N4000, A400, and A500 models are historic PA-RISC models for MPE/iX. While they can run HP-UX, this is not recommended due to different Ethernet controller requirements between MPE and HP-UX.

[2] Oversized versions of the models may be available (up to 128 CPUs and 512GB of RAM). Please check availability with your Sales contact.

[3] Starting with version 3.0.5, oversized versions of the rp7400 may be available (up to 64 CPUs and 512GB of RAM). Please check availability with your Sales contact. The N4000 models can have 1-4, 6, or 8 emulated CPUs.

[4] Supported starting with Charon-PAR version 3.0.7. Oversized versions with up to 16 CPUs and up to 3840MB RAM (starting with version 3.0.14) may be available. **HP-UX limitations**: 10.20 supports 16 CPUs; 11.00 and 11.11 support 15 CPUs. 10.20, 11.00 and 11.11 support up to 3840MB RAM. Please check availability with your Sales contact.

[5] 2GB supported starting with Charon-PAR version 3.0.8-22201. Earlier versions support 1.5GB.

[6] Oversized versions with up to 4 CPUs (starting with Charon-PAR 3.0.13) and up to 3840MB of RAM (starting with version 3.0.14) may be available. **HP-UX limitations**: 9.x supports 1 CPU and up to 768 MB of RAM; 10.20 and 11.00 support 4 CPUs and up to 3840 MB of RAM. Please check availability with your Sales contact.

Emulated Models Supported by Charon-PAR/PA3

The following table provides a short overview of the historic PA-RISC models for MPE/iX emulated by the Charon-PAR Software. For more details regarding each model, please refer to *Emulated Model Hardware Configuration Details* in the Charon-PAR user's guide. For configuration details and restrictions, please refer to the section *Configuration File Reference* in the Charon-PAR user's guide.

Please note: some emulated devices (e.g., tape drives, serial, and parallel ports) can be linked to physical devices on the Charon host system. As the physical devices on a cloud-based Charon host system typically do not exist, such configurations are not available in cloud environments.

Model	Maximum Memory	Number of CPUs	SCSI contr.	Ethernet contr.	Serial/ par. ports	Expansion slots
A400	2GB	1	Dual SCSI-2 contr. (LSI 53C8xx) and Dual SCSI-3 LVD contr. (LSI 53C8xx)	DEC 21143 Tulip 10/100 Mbit/s	Diva serial PCI card (2 ports)	2
A500	8GB	1 - 2				4
N4000	16GB	1 - 4				12

Charon Licensing Options Overview

Charon Emulator products require a product license to run. Stromasys offers several different licensing methods from which the one best suited to the customer requirements must be selected. This page provides an overview of the different methods and their applicability to different emulator environments.

The **three basic licensing technologies** that will be described in the following sections are:

- **Sentinel HASP licensing**, a third-party solution offering local and network-based hardware and software licenses - now owned by Thales.
- **VE (Virtual Environment) licensing** developed and owned by Stromasys offering local and network-based software licenses.
- **Automatic licensing for cloud instances** developed and owned by Stromasys (currently for Charon-SSP only) offering cloud-specific metered licensing.

Additional information:

- Sentinel HASP Licensing Documentation: detailed information about HASP licensing and managing the licensing option on different emulator products.
- Virtual Environment (VE) License Server Documentation: detailed information about installing, configuring, and managing a VE license server, and about managing the licensing option on different emulators.
- Charon-AXP, Charon-VAX, Charon-PAR, and Charon-SSP user's guides: detailed information about the product installation, in particular license drivers and utilities included in the product distribution. Detailed information about the configuration file content of each product.

Contents

Product Support Overview and General Recommendations

The different licensing types are described in more detail below. This section shows which options are supported by the individual products, and for which environments the options are best suited.

Your product	Supported licensing options	Suitable environment for licensing option
All emulator products	Sentinel HASP USB hardware license	On-premises installation on physical servers. On-premises installation in virtualized environment with USB port pass-through or USB port server. Backup license strongly recommended.
	Sentinel HASP Software license	On-premises installation on physical servers with stable hardware and operating system configuration. On-premises installation in virtualized environment with stable virtual hardware and operating system configuration. Backup license indispensable.
Charon-AXP/VAX for Linux starting with version 4.12 Charon-PAR starting with version 3.0.6 Charon-SSP starting with version 4.2	VE licensing with general VE mode	On-premises installation on physical servers (VE license server on physical Linux host or on Linux instance in a supported cloud). On-premises installation in VMware environments with VE license server in same vCenter management sphere. Cloud installations (VE license server on physical Linux host or on Linux instance in a supported cloud). Backup license server indispensable.
Charon-SSP starting with version 4.2	Automatic licensing	Supported for Charon-SSP AL marketplace imaged provided on selected clouds. Public license servers operated by Stromasys.
Charon-SSP starting with version 5.4	Automatic licensing with AutoVE server	Supported for Charon-SSP AL marketplace imaged provided on selected clouds. Private AutoVE license server operated by the customer. Backup license server indispensable.

Licensing Options Characteristics

Common Characteristics of HASP and VE Licenses

Both, Sentinel HASP and VE licenses, have some common characteristics despite the differences in the technical implementation:

- Several products can be licensed on one license.
- Several emulator instances can be licensed on one license.
- A license can be limited by a specified expiration date.
- A license can be limited by a specified number of runtime hours (mostly used as backup licenses).
- The license used by the emulator can be on the Charon emulator host itself or served over the network. Since software licenses are very sensitive to any hardware change or backup restoration, dedicated stable license server systems are recommended in production environments using software licenses.

Sentinel HASP Licenses (Thales) - Specific Characteristics

Sentinel HASP Overview

Sentinel HASP licenses are the "traditional" licensing method for Charon emulator products. Their main characteristics are:

- Software and hardware (dongle) licenses.
- Based on third-party vendor solution.
- Require special third-party license driver software which is distributed with the Charon emulator products.
- Installed on Charon host or separate license server.
- Problematic in cloud environments and somewhat difficult to use in VMware environments.
- Hardware dongles are a flexible and host-hardware independent solution for on-premises installations (as long as there is a free USB port).
- The customer is billed by Stromasys depending on the number and type of the emulated systems allowed by the installed license(s). The license driver software itself is free of charge.

Please refer to the Charon License Handbook (Licensing Documentation) for details about these licenses.

USB Hardware Dongles

Prerequisites	• USB hardware dongles require an available USB port on the system on which they are installed, or a supported USB port server. • Supported Linux or Microsoft Windows system. • Supported license drivers. • In case of a network dongle: functioning network and firewall configuration between license client and license server.
Emulator support	All emulator products.
Typical use cases	• **On-premises installations** - in particular if physical servers are used as Charon emulator hosts. For on-premises installations in VMware and similar environments, dongles can be used, but require additional configuration (e.g., passing a dongle through to a Charon host VM) and/or hardware (e.g., USB port server). • Backup licenses for other HASP licenses. As there is no link to the hardware or operating system on which the dongle is used, dongles can be moved easily between systems.
Not suited for	Cloud environments. Even though theoretically an emulator host could be linked to an on-premises network license server offering a network license installed on a dongle, this is not a sustainable solution for a production environment.

HASP Software License

Prerequisites	• Supported Linux or Microsoft Windows system. • Supported license drivers. • If offered as a network license: functioning network and firewall configuration between license client and license server. • Stable hardware configuration of the system the license is installed on. Since a software license is tied to the hardware characteristics of the system it is installed on, changes in the hardware can easily invalidate the license. • Backup license to support continued operation should there be an accidental invalidation of a software license.
Emulator support	All emulator products.
Typical use cases	• **On-premises installations** if the hardware of the system the license is installed on is stable. Special attention to hardware stability is needed if the system runs under a hypervisor. • Used as primary license with USB dongle or a network-based license server as backup license solution.
Not suited for	• Cloud environments. Cloud instance hardware is not stable with respect to the characteristics relevant to a Charon HASP software license. The Charon emulator license is likely to be invalidated quickly. • Other environments where the license would be installed on a system with frequently changing hardware.

Charon VE (Virtual Environment) Licenses - Specific Characteristics

VE License Basic Overview

The **main characteristics** of VE (Virtual Environment) licenses are the following:

- Software licenses only.
- Developed by Stromasys.
- Installed on the Charon host or a separate license server. Since software licenses are very sensitive to any hardware change or backup restoration, dedicated stable license server systems are recommended in production environments using software licenses.
- Require the Charon VE license server software.
- Require matching VE-capable Charon emulator software.

Prerequisites	• Supported Linux system to be used for license server. • Supported platform (supported cloud, VMware, or physical server) for the license server. • If used across the network, correctly configured connectivity and firewall settings. • Additional platform-dependent requirements described in the Virtual Environment (VE) License Server Documentation.
Emulator support	• Charon-SSP • Charon-PAR • Charon-AXP/VAX for Linux
Typical use cases	• Cloud installations • On-premises installation where a hardware dongle is not required.
Not suited if	• License server hardware changes frequently. Or frequent data restoration is expected. • Licensing via hardware dongle is required.

Modes of Operation

VE licensing can be used in two different modes described below.

General VE Mode

Operational information:

- The license server is owned and operated by the customer.
- The customer requests a license from Stromasys and installs it on the license server.
- The customer is billed for the license by Stromasys depending on the number and type of the emulated systems allowed by the installed license(s).
- The license server software itself is free of charge.

AutoVE Mode

Operational information:

- Offered on selected clouds only (requires AL marketplace image).
- At the time of writing Charon-SSP AL marketplace image only.
- The license server is owned and operated by the customer.
- The customer requests a license from Stromasys and installs it on the license server (due to the sensitivity to hardware changes and data restoration, the license server should be a stable dedicated instance).
- The customer is billed for the license and the cloud instance by the cloud provider.
- The license server software itself is free of charge.

AutoVE mode is an extension of automatic licensing and introduces metered billing (by the cloud-provider) for VE licenses in cloud environment. It defines how many Charon emulator cloud instances can be run based on the respective license. The number of emulated systems on each host instance is limited by the host resources, not the license.

Supported Platform Information

Support at the time of writing:

VE license server availability	Supported clouds (at the time of writing: AWS, OCI, Azure, GCP, IBM, and Nutanix), supported VMware environments, and physical servers.
AutoVE support	• VE License server starting with version 1.1.21. • Charon-SSP AL (Automatic Licensing) marketplace images version 5.3.8 or higher on AWS, OCI, GCP, and Azure only.
Charon emulator product support for General VE Mode	Charon-SSP and Charon-PAR (starting with Charon-PAR version 3.0.6 and VE license server version 1.1.19). Support for Charon-AXP/VAX is planned starting with version 4.12.

Charon-SSP Automatic Licensing (AL) for Cloud Environments

Main characteristics:

- For special Charon-SSP AL marketplace image in selected clouds. The cloud instance automatically receives a license upon first launch.
- The customer is billed for the cloud instance and the Stromasys product by the cloud provider.
- Software license.
- Licenses served by a Stromasys-operated public license server.
- Licensing is not based on the number of Charon emulator instances, but per cloud instance - independent of the number of emulators run on the instance. The number of emulated systems on each host instance is limited by the host resources, not the license.

Prerequisites	• Charon-SSP AL marketplace image. • Stable virtual hardware of instance (changing number of CPUs will invalidate the license and require that a new instance be launched). • Working Internet access of instance.
Emulator support	At the time of writing only Charon-SSP based on an AL marketplace image available on selected clouds.
Typical use cases	• Cloud-only. • Typically used by customers without a large cloud deployment who don't have a large cloud environment yet.
Not suited for	Anything outside a supported cloud. Large customer-adapted cloud environment. Privacy requirements prohibit Internet access

For a similar solution, but with a private, customer-operated license server, see AutoVE in the VE licensing section.

Charon-PAR Prerequisites

Contents

Host System Prerequisites

Hardware Prerequisites

The host hardware on which Charon-PAR runs must fulfill at least the following requirements:

- Intel or AMD x86-64 hardware platform (AMD support starting with Charon-PAR version 3.0.6)
- At least 3GHz, 3.4GHz or higher is recommended
- SSE 4.2 and FMA required
- **CPU cores required**:
  - at least one CPU core for the host operating system, **and**
  - at least 2 cores per emulated CPU (3 cores for future advanced DIT)
- **RAM requirements**:
  - 4GB RAM plus 1.1 times the emulated RAM size for the emulator. Additional memory is required for I/O buffering and other Linux processes (2 to 4 GB can be used as a starting point).
  - at least 24GB RAM for N4000 models
- One Ethernet interface for the host system and one Ethernet interface for each emulated Ethernet NIC. TAP interfaces connected to a virtual Linux bridge can also be used. Please note that the use of TAP interfaces is restricted to internal bridges in cloud environments (that is, a bridge cannot be linked to a NIC connected to the cloud LAN). The latest interface type supported is MACVTAP mode.
- At least one available USB port if HASP hardware licenses are to be used.
- Disable NUMA balancing (see section *Disabling NUMA Balancing* below).
- If possible, turn off hyper-threading in BIOS.

Please note:

- The sizing guidelines above—in particular regarding number of host CPU cores and host memory—show the **minimum requirements**. Every use case has to be reviewed and the actual host sizing has to be adapted as necessary.
- If the host CPU does not support the required extensions (SSE and FMA), the emulator will not start. Instead, it will stop with an error message indicating the missing features.
- To identify the capabilities of the host system CPU, use one of the following commands: **`lscpu`** or **`cat /proc/cpuinfo`**.

Disabling NUMA Balancing

One prerequisite for Charon-PAR is to disable NUMA balancing. There are two basic methods to do so. They are described below.

General Information

As the first step, **check the status of NUMA balancing**:

`# cat /proc/sys/kernel/numa_balancing` (0 = off, 1 = on)

If it is enabled, use **one of the methods** below to disable it.

Caveat: a different kernel parameter, **numa** (values **on** or **off**), allows disabling the NUMA awareness of the kernel. In a system with several NUMA nodes, this can cause serious **performance degradation**. Disabling NUMA awareness of the kernel is not a prerequisite for Charon-PAR.

Disabling NUMA Balancing Using *sysctl*

To **disable NUMA balancing temporarily in a running system**, use the command: `# sysctl -w kernel.numa_balancing=0`

To **make the configuration permanent** you must create a **sysctl** configuration entry:

- Create a file in **/etc/sysctl.d/** (e.g., 90-numa.conf)
- Add the line `kernel.numa_balancing=0`
- Save the file.
- **Please note**: this method overwrites any NUMA settings made via the kernel boot option method below.

As this method does not require a reboot of the system, it is the less disruptive one.

Disabling NUMA Balancing via a Kernel Boot Option

To disable NUMA balancing via a kernel boot parameter, perform the following steps as the root user:

- Open the file **/etc/default/grub** in a text editor.
- Add the following parameter to the variable GRUB_CMDLINE_LINUX:
  - `numa_balancing=disable`
- Save the file and regenerate the grub.conf file.
  - For systems with BIOS boot:
    `# grub2-mkconfig -o /boot/grub2/grub.cfg`
  - For systems with UEFI boot (example for RHEL; check the location if you use a different distribution):
    `# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg`
  - Reboot the system.

Please note: if the numa_balancing parameter is set via **sysctl** (see above) it will override the value in the grub configuration.

Software Prerequisites

Host Operating System

The following host operating systems are supported by Charon-PAR:

- Red Hat (64-bit): versions 7.x starting with 7.4, versions 8.x, and (starting with Charon-PAR 3.0.8) 9.x
- Oracle Linux (64-bit): versions 7.x starting with 7.4, versions 8.x, and (starting with Charon-PAR 3.0.8) 9.x
- Rocky Linux (64-bit): versions 8.x, and (starting with Charon-PAR 3.0.8) 9.x
- CentOS (64-bit): versions 7.x starting with 7.4, and versions 8.x.
  Please note that as of 1 January 2022 CentOS 8 is EOL; for new deployments, it is strongly recommended to use a non-EOL alternative. For existing installations, the possible negative impacts of staying with an EOL host operating system should be carefully evaluated. Future versions of Charon-PAR many no longer run on CentOS 8 systems.

General limitation:

- Do not hibernate or suspend the host system while Charon-PAR instances are active.
- Charon-PAR 3.0.5 build 21906 implemented real-time priority for the timer task. **This changes requires that CPU accounting on the Linux host be disabled**.

Please note: Stromasys may offer prepackaged cloud marketplace images for Charon-PAR. At the time of writing, this is planned for AWS. Such marketplace images contain the supported host operating system, the required additional software and the Charon-PAR emulator product.

Other Host Software Prerequisites

Older HASP license driver packages (e.g., 7.6.3) require the 32-bit glibc package. This is available in the standard CentOS and Red Hat repositories. Sometimes it is not possible to use an online repository for the installation of 32-bit glibc package. In this case the procedure described in How to install the glibc.i686 package without an Internet connection can be used.

On RHEL 7.x and CentOS 7.x, the **libev** package is required. If it is reported as missing during the Charon installation on RHEL 7.x check that the repository **extras** is included and enabled, if not, include and enable it. Please refer to your Linux distribution administrator's guide. Example for RHEL 7.x:

```
# yum-config-manager --enable rhel-7-server-extras-rpms
```

Supported Hypervisors

The Linux host system can run inside a **VMware** virtual machine. Supported VMware versions:

- VMware ESXi 5.5, 6.x, and 7.x

Please note:

- Using the VMware vSphere Hypervisor (ESXi) to dynamically move an active Charon-PAR instance to a different server is not supported (vMotion). In particular, moving to a system that does not have identical hardware (including clock speed) will cause problems.
- Prerequisites of additional products may limit the choice of hypervisors. For example, a VE license server VM requires VMware ESXi 6.5 or higher. Please refer to the appropriate documentation.

Should you require a different hypervisor, please contact your Stromasys representative to discuss your requirements.

Licensing Requirements

Charon-PAR requires a valid license to run any emulated systems. The license can be provided in several ways:

- Sentinel HASP licenses:
  - local hardware license (USB dongle),
  - software license, or
  - network license (every software license can also be a network license).
- VE (Virtual Environment) license server (starting with Charon-PAR version 3.0.6). The license server can be on the same system as the emulator or on a separate system. If the license server is to be installed on the same system as the emulator, the emulator must run in an environment supported by the license server (see Licensing Documentation).

Without the presence of a valid license key, the emulated system will not start. Should the license become invalid or be removed during operation, the emulated system will stop after a grace period. The lengths of this grace period depends on the version of the Charon-PAR software and the license configuration.

In a virtualized or cloud environment, there maybe additional challenges for user-friendly and stable product licensing. For such environments, please contact Stromasys or your Stromasys VAR to discuss the best option for your requirements.

Guest System Prerequisites

Currently supported Charon-PAR/PA9 guest system versions:

The currently implemented **64-bit models** support the following HP-UX versions:

- HP-UX 11v1 (11.11), 11v2 (11.23)
- In addition, emulated systems configured with 360 and 440 MHz CPUs can also run HP-UX 11.00 (e.g., rp2400-1-360, rp2400-1-440, rp7400-1-440, etc.)
- HP-UX 11v3 (11.31), on rp34xx and rp44xx only, is supported starting with Charon-PAR version 3.0.5.

The currently implemented **32-bit models** support the following 32-bit HP-UX versions:

- The 9000/720: HP-UX 9.05, 9.07, 10.20, and 11.00
- The 9000/B132L: HP-UX 10.20, 11.00, and 11v1 (11.11)

Currently supported Charon-PAR/PA3 guest system versions:

- MPE/iX 7.5 (Backward compatible)

HP Operating System Licensing Requirements

The user is responsible for clearing any licensing requirements regarding the guest operating system and the applications running on the guest operating system with the original vendor(s).

Charon-PAR Software Installation

Contents

Important information for Charon-PAR software upgrades:

- Upgrading to this Charon-PAR from an older version requires a license update (independent of license type used). Please contact your VAR or Stromasys representative to plan the update.
- Applicable to HASP licenses only: Charon-PAR version 3.0 and higher is **incompatible** with licenses of earlier versions. Earlier product versions are incompatible with licenses for Charon-PAR 3.0 and higher. Charon-PAR/PA3 licenses for version 3.0 are not backward compatible for emulators running with version 2.x.

Charon-PAR Software Packaging and Installation

The Charon-PAR software can be provided in different formats. At the time of writing, the following software packaging was available:

- A set of RPM packages for installations on all supported platforms and environments.
- A marketplace AMI containing the base operating system and the preinstalled Charon-PAR software. At the time of writing, this format was planned for AWS and Azure. Please contact your Stromasys representative for details about availability on AWS and other cloud platforms. The Appendix of this document contains basic information about setting up a cloud instance.

Installing Charon-PAR from RPM Installation Packages

Charon-PAR RPM Software Packages Overview

Charon-PAR is provided as a **set of RPM-packages**. This set is different for versions before 3.0.10 and version 3.0.10 and later.

Charon-PAR Versions 3.0.10 and Later

In **version 3.0.10**, Charon-PAR is provided as **one self-extracting archive** containing **all** the RPM packages:

> **charon-par-***<version>***.sh**

Starting with **version 3.0.11**, Charon-PAR is provided as **one self-extracting archive per Linux major version**: one archive each for Linux 7.x, 8.x, and 9.x.

- **charon-par-***<version>***.el7.sh**
- **charon-par-***<version>***.el8.sh**
- **charon-par-***<version>***.el9.sh**

To **unpack the archive**, perform the following steps:

- Go to the directory where the package is stored.
- Run the archive shell script: **# sh** *<archive-name>*
- Accept the EULA. To successfully unpack the archive, the end-user license agreement must be accepted.
- After this, the software packages making up the Charon-PAR kit will be extracted into a version-specific sub-directory of the current working directory of the user.

Example:

```
$ sh charon-par-3.0.11-22700-54a67939.el8.sh
Verifying archive integrity...  100%   MD5 checksums are OK. All good.
Uncompressing Stromasys product packages installer  100%
End User License Agreement for : STROMASYS SOFTWARE.

NOTICE TO USER ("LICENSEE", “YOU” or “YOUR”): THIS AGREEMENT GOVERNS

<lines removed>

Please confirm EULA (yes/no) > yes
EULA accepted
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase

$ ls charon-par-3.0.11-22700-54a67939
aksusbd-7.63-1.i386.rpm
aksusbd-8.13-1.x86_64.rpm
Charon_Idle_0.02.depot
charon-license-1.2.0-1.x86_64.rpm
charon-license-certs-1.2.0-1.x86_64.rpm
charon-license-hasp-1.2.0-1.el8.x86_64.rpm
charon-mtd-4.11-20406.el8.x86_64.rpm
charon-par-3.0.11-22700.54a67939.el8.x86_64.rpm
EULA.txt
xhpterm-free-3.0.11-22700.54a67939.el8.x86_64.rpm
```

Please note that **gzip**, **md5sum**, **cksum**, **gpg**, **tar**, and **openssl** are required to successfully unpack the RPMs.

The following table lists the installation packages available in the archives.

Placeholders:

- ***<version>*** - version of the package, e.g. 3.0.11-22700.54a67939
- ***<linux-version>*** - major version of the Linux host operating system, i.e., **el7**, **el8**, or **el9**

Package	Description
Charon-PAR legal end-user license agreement	
`charon-license-<version>.x86_64.rpm`	Mandatory package. End-user license agreement text will be installed in */opt/charon/doc.*
Sentinel runtime environment and utilities. Required for licensing the Charon-PAR software using Sentinel HASP licenses.	
`aksusbd-7.63-1.i386.rpm`	32-bit package for CentOS/RHEL/Oracle Linux 7.x; requires 32-bit glibc package
`aksusbd-8.13-1.x86_64.rpm`	64-bit package for RHEL 7.7 and higher; RHEL/CentOS/Oracle/Rocky LInux 8. (see also https://docs.sentinel.thalesgroup.com/ldk/LDKdocs/rte/linux-rh-suse-centos.html)
`charon-license-hasp-<version>.<linux-version>.x86_64.rpm`	HASP license management utilities and libraries.
Charon-PAR emulator	
`charon-par-<version>.<linux-version>.x86_64.rpm`	Charon-PAR emulator executable, template configuration files for the different supported hardware models, PuTTY configurations, etc.
Charon-PAR utilities	
`charon-mtd-<version>.<linux-version>.x86_64.rpm`	Charon-PAR tape utility for Linux
`xhpterm-free-<version>.<linux-version>.x86_64.rpm`	Terminal emulation for block mode applications
Additional Packages	
`charon-license-certs-<version>.x86_64.rpm`	Sample certificates for the new custom certificate feature of the VE license server. Support planned for Charon-PAR 3.0.11 and later.
`Charon_Idle_0.02.depot`	Optional HP-UX package to enable the power save feature in HP-UX running in an emulator.

Please note: the VE license server RPM package is not part of the Charon-PAR set of RPM packages. Stromasys or your Stromasys VAR will provide you with the software or a download link. Prepackaged cloud marketplace images include the VE license server RPM (in */charon/storage*).

Charon-PAR Versions 3.0.9 and Older

The following table lists the available installation packages.
Support for RHEL 9.x and derivatives was added in Charon-PAR version 3.0.8.

Placeholders:

- ***<version>*** - version of the package
- ***<linux-version>*** - major version of the Linux host operating system, i.e., **el7**, **el8**, or **el9**

Package	Description
Sentinel runtime environment. Required for licensing the Charon-PAR software using Sentinel HASP licenses.	
`aksusbd-7.63-1.i386.rpm`	32-bit package for CentOS/RHEL/Oracle Linux 7.x; requires 32-bit glibc package
`aksusbd-8.13-1.x86_64.rpm`	64-bit package for RHEL 7.7 and higher; RHEL/CentOS/Oracle/Rocky LInux 8. (see also https://docs.sentinel.thalesgroup.com/ldk/LDKdocs/rte/linux-rh-suse-centos.html)
Charon-PAR emulator	
`charon-par-<version>.<linux-version>.x86_64.rpm`	Charon-PAR emulator executable, template configuration files for the different supported hardware models, PuTTY configurations, etc.
Charon-PAR utilities	
`charon-mtd-<version>.<linux-version>.x86_64.rpm`	Charon-PAR tape utility for Linux
`xhpterm-free-<version>.<linux-version>.x86_64.rpm`	Terminal emulation for block mode applications
Additional Packages	
`Charon_Idle_0.02.depot`	Optional HP-UX package to enable the power save feature in HP-UX running in an emulator.

Please note: the VE license server RPM package is not part of the Charon-PAR set of RPM packages. Stromasys or your Stromasys VAR will provide you with the software or a download link. Prepackaged cloud marketplace images include the VE license server RPM (in */charon/storage*).

Important if there is more than one emulator product installed on the same Charon host system: Charon-PAR versions 3.0 up to 3.0.9 contain the *h aspvlib* library and the *hasp_srm_view* and *hasp_update* utilities. This means that there may be installation conflicts with other Charon emulator products on the same host systems. Such conflicts can be overcome either by not installing the conflicting modules of the other products (if possible), or by forcing the installation of the Charon-PAR kit. Please refer to the documentation of your host system (**man rpm**) in such cases and ensure that this will not negatively influence any other installed products. If you use the **`--force`** option for an installation, please uninstall older packages of the same product before

Additional Information

Obtaining the Charon-PAR software:

The software can be downloaded from the Stromasys fileserver. If required, please contact your Stromasys representative or your VAR partner. They will provide the download information or the software packages to you.

Charon-PAR cloud marketplace images can be obtained from the respective cloud marketplaces.

Additional requirements for a successful installation:

- The packages above have installation dependencies that are normally satisfied automatically from the standard software repositories when installing packages with **yum** or **dnf**. If the standard repositories are not available, please refer to How to install the glibc.i686 package without an Internet connection. The document explains - among other points - how to create a local repository.
- Additional packages may be required to be installed on the host operating system. They will be discussed in the sections where they are relevant.
- If additionally required packages are not available in the standard repositories of the host operating system, additional repositories can be installed using, for example, the EPEL repository (more information: https://fedoraproject.org/wiki/EPEL).

Additional requirements for initial emulator setup:

- For the installation of a Charon-PAR/PA3 emulated system, a preinstalled MPE/iX 7.5 disk image (**mpe75a.dsk.bz2**) is required. This will be provided by Stromasys on demand.
- For the installation of a Charon-PAR/PA9 emulated system, a HP-UX ISO is required. It must be provided by the customer.

Charon-PAR RPM Installation Command Overview

The following table provides an overview of the installation commands for the supported host operating systems. For details, please refer to the relevant man-pages on Linux. The table only lists command-line installation options for Linux. There are also graphical installation tools. To describe all of them is outside the scope of this document.

Installation Option	Description
Package manager (uses repositories, takes care of dependencies, etc.)	`# yum install <package-name>` On Linux 8.x, replace **yum** with **dnf**. if a filename is specified, locally stored RPM packages can be installed.
Command to install individual local packages.	`# rpm -i </path/to/package>`

Sentinel Runtime RPM Installation

This package is mandatory for older Charon-PAR emulator versions. However, the runtime software installed is **only needed for HASP licensing.** It is **not relevant for VE licensing.** For later versions the installation is optional if HASP licensing is not used.

Important points:

- The **aksusbd package version 7.63** requires the 32-bit glibc-library. On supported Linux distributions, this library is available through the standard repositories. The relevant package is **glibc.i686** on Red Hat Enterprise Linux and CentOS. Unless version 7.63 is needed to solve a specific problem, aksusbd version 8.13 which does not have this limitation should be used.
- Normally, the **aksusbd** package installation with **yum** or **dnf** installs the 32-bit glibc and other dependency automatically (provided package repositories are available). Red Hat Enterprise must be registered to access the Red Hat repositories, or the installation CD must be mounted. Should the Charon host system not have access to the Internet, please refer to How to install the glibc.i686 package without an Internet connection for further information.
- When installing the **aksusbd package version 8.13** in some CentOS 7.x environments it was observed that the aksusbd service did not start automatically after a new installation or an upgrade. The following error was displayed during the installation: ***Failed to restart aksusbd.service: Access denied***. If this happens, uninstall the aksusbd package and reboot the system. After rebooting the system, re-install the package. The cause of this sporadic problem has not yet been identified.

The **aksusbd** package installs the programs **aksusbd** and **hasplmd** in **/usr/sbin**.

The example below shows the installation of the **aksusbd** package (version 8.13):

```
# dnf install aksusbd-8.13-1.x86_64.rpm
Last metadata expiration check: 0:59:01 ago on Mi 18 Mai 2022 15:16:11 CEST.
Dependencies resolved.
================================================================================
 Package          Architecture    Version           Repository             Size
================================================================================
Installing:
 aksusbd          x86_64          8.13-1            @commandline          7.1 M

Transaction Summary
================================================================================
Install  1 Package

Total size: 7.1 M
Installed size: 14 M
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Running scriptlet: aksusbd-8.13-1.x86_64                                  1/1
  Installing       : aksusbd-8.13-1.x86_64                                  1/1
  Running scriptlet: aksusbd-8.13-1.x86_64                                  1/1
Created symlink /etc/systemd/system/multi-user.target.wants/aksusbd.service  /etc/systemd/system/aksusbd.service.
Created symlink /etc/systemd/system/multi-user.target.wants/hasplmd.service  /etc/systemd/system/hasplmd.service.

  Verifying        : aksusbd-8.13-1.x86_64                                  1/1

Installed:
  aksusbd-8.13-1.x86_64

Complete!
```

Installing the Charon-PAR RPM Packages

To install the Charon-PAR packages perform the following steps:

	Step	Command
1	Log in as the privileged user on the system ("root").	
2	Go to the directory in which the rpm packages were stored.	# **cd** *<path-to-kits-dir>*
3	Install the Charon-PAR emulator and utilities packages.	# **yum install charon*.rpm xhpterm*.rpm** (Linux 7.x) **# dnf install charon*.rpm xhpterm*.rpm** (Linux 8.x and 9.x) or **# rpm -i *.rpm**

The following shows a sample installation of the Charon-PAR emulator package on CentOS 8:

```
# dnf install charon-par-3.0.0-21400.ed1d4d8.el8.x86_64.rpm
Last metadata expiration check: 1:27:20 ago on Wed 29 Jul 2020 07:39:18 PM CEST.
Dependencies resolved.
================================================================================
 Package        Arch       Version                       Repository        Size
================================================================================
Installing:
 charon-par     x86_64     3.0.0-21400.ed1d4d8.el8       @commandline     6.0 M

Transaction Summary
================================================================================
Install  1 Package

Total size: 6.0 M
Installed size: 32 M
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Running scriptlet: charon-par-3.0.0-21400.ed1d4d8.el8.x86_64              1/1
  Installing       : charon-par-3.0.0-21400.ed1d4d8.el8.x86_64              1/1
  Running scriptlet: charon-par-3.0.0-21400.ed1d4d8.el8.x86_64              1/1
  Verifying        : charon-par-3.0.0-21400.ed1d4d8.el8.x86_64              1/1
Installed products updated.

Installed:
  charon-par-3.0.0-21400.ed1d4d8.el8.x86_64

Complete!
```

Location of Charon-PAR Files

Charon-PAR is installed under **/opt/charon**.

- **/opt/charon/bin**: executable of emulator and utilities
- **/opt/charon/cfg**: template configuration files for emulated systems and PuTTY session definitions (to connect to the emulated console).
- **/opt/charon/doc**: license agreement and Release Notes (WHATSNEW file)

To add the installation path to your PATH variable, become the root user and create the file **/etc/profile.d/charon.sh** with the following content:

```
PATH=$PATH:/opt/charon/bin; export PATH
```

The updated PATH variable will become active after the next login. You can also execute this command directly: `# . /etc/profile.d/charon.sh`

Post-Installation Tasks

Installing PuTTY

A recommended post-installation task is the installation of PuTTY to connect to the emulated console. The associated package is part of the EPEL repository.

PuTTY installation steps CentOS/Red Hat:

	Step	Command
1	Log in as the privileged user on the system ("root").	
2	If the additional repository has not yet been installed, install it. The rpm package to install the EPEL repository is included in the extras repository of Red Hat and CentOS.	`# yum --enablerepo=extras install epel-release` Up-to-date information can be found on the following page: https://fedoraproject.org/wiki/EPEL
3	Install PuTTY.	`# yum install putty`
4	If you used PuTTY before, you will have a **.putty** or **.config/putty** directory under your home directory already. If the PuTTY configuration directory does not already exist, create the **.config/putty** directory in the home directory of the root user.	Start PuTTY: `# putty` Select and **load** the default settings. Click on **Save**. Exit PuTTY.

5	Copy the Charon-PAR profiles to the PuTTY sessions directory of the root user (depending on where you stored you session data).	`# cp /opt/charon/cfg/PAR* /root/.putty/sessions`
		or
		`# cp /opt/charon/cfg/PAR* /root/.config/putty/sessions`

Please note:

- In some versions of CentOS/RHEL 8 PuTTY was not included in the EPEL repository. In such cases, the version from RHEL/CentOS 7 could be used.
- If you want to use PuTTY from a different user, perform the profile copy action above for the user in question as well.

Charon-PAR Cloud-specific Marketplace Image Overview

Cloud-specific marketplace images can be used to launch a Charon-PAR host instance in a cloud instead of installing the RPM packages on a Linux instance in the cloud. Marketplace images contain the following software:

- The underlying Linux operating system.
- The preinstalled Charon emulator product.
- Utilities required by the emulator product.
- Additional management utilities (Linux Toolkit).

For further information, please refer to the Appendix, the Virtual Environment (VE) License Server Documentation, and the documentation of your cloud provider for additional information.

At the time of writing, such a marketplace image was planned for AWS. Please contact your Stromasys representative with any questions regarding availability.

Installing the Linux Toolkit (Optional)

This is not a mandatory, but a recommended step. The toolkit is included in Charon-PAR cloud marketplace images.

The Linux Toolkit support the management of emulator related tasks, such as

- License Management
- Starting and stopping the emulator
- Log file monitoring
- Alert management
- Information collection to facilitate product support

Stromasys will provide you with a download link for the Linux Toolkit.

Please refer to Charon-PAR Linux Toolkit for further information.

Sentinel HASP License Driver Post-Installation Tasks

Not required for VE licenses.

The following post-installation tasks are strongly recommended to improve security and to enable additional trouble-shooting options:

- Protect remote access to the Sentinel Admin Control center: see *Security Settings for Sentinel Admin Control Center* in the *HASP Licensing handbook* under Licensing Documentation.
- Enable access and error logging for the license driver: see Enabling logging in Sentinel Admin Control Center

VE License Server Installation

This step is only required under the following conditions:

- You plan to use VE licensing for your installation, **AND**
- you cannot use an existing VE license server making it necessary to install the license server software locally or on a dedicated system.

In this case, use the steps describe below.

1. The Charon VE License Server package is delivered as an installation package that is **not part of the Charon-PAR set of RPM packages**. Stromasys or your Stromasys VAR will provide you with the software or a download link.

The packaging is different based on the VE license server version:

- VE license server version 2.2.3 and older: `license-server-<version>.rpm`
  RPM package for installation on a supported Linux system.
- VE license server version 2.2.4 and higher: `license-server-<version>.rpm.sh`
  Self-extracting archive containing the end-user license agreement (EULA) and the RPM package.

In both cases *<version>* indicates the version of the software, for example, 2.2.1

2. As a privileged user (**root**) go to the directory where you stored the installation package and install the package. On an instance installed from a prepackaged Charon marketplace image, the installation package is stored under */charon/storage*.

- For VE license server 2.2.4 and above, unpack the archive and agree to the end-user license agreement:
  - `# sh license-server-<version>.rpm.sh`
    This will display the EULA. After agreeing to it, the RPM installation package will be unpacked in the current directory.

- Install the package:
  - Linux 7.x: `# yum install license-server*.rpm`
  - Linux 8.x and 9.x: `# dnf install license-server*.rpm`

Please note: the VE license server software has **specific software requirements** (especially in the different cloud environments and on VMware). Please refer to the *VE License Server Guide* in Licensing Documentation section on the Stromasys documentation page.

Initial Emulator License Configuration

Each Charon emulator product requires a valid license to run. This license can be a HASP or a VE license. Different steps are required to configure the emulator to use the license depending on the license type:

- Initial License Configuration for VE Licenses
- Initial License Configuration for Sentinel HASP Licenses

Please note: when configuring the emulator, VE licenses and HASP licenses are **mutually exclusive**. If a configuration file contains the definition of a VE license, any HASP license definition is ignored (that is, the VE configuration takes precedence).

Initial License Configuration for VE Licenses

If you use a Charon VE license server and the matching Charon emulator packages (minimum version Charon-PAR 3.0.6), the following steps are required:

- Configure the license server address, the TCP port number (if license not offered on the default port 8083), and the passphrase in the license section of the emulator configuration using the following syntax:

  ```
  license_key_id "VE://<license-server-IP>[:port]/<passphrase>/"
  ```

- If this has not been done already:
  - Create a C2V (customer-to-vendor) file **on the license server**. Use this file to request a license from Stromasys.
  - The license received in response (V2C file) must be installed on the license server.
- If a backup license server is to be used, add a second `license_key_id` line for the backup server. The first line indicates the primary server, the second the backup server. Only **one** backup server can be configured.

Currently, the VE license server is available in supported cloud and VMware environments, and on physical servers.

Firewall considerations:

If the VE license server is not installed on the same system as the emulator, any intermediate firewall must allow at least the port on which the license is served. Optionally, the firewalls must allow the port on which the web-based GUI is available. These ports are **configurable** on the VE license server. The default values are the following:

- Default port on which licenses are served by the VE license server: TCP 8083.
- Default port on which the web-based GUI runs: TCP 8084.

Further information:

Please refer to the Licensing Documentation on the Stromasys documentation website and the section License Management Overview for Charon-PAR in this document for information about how to request and install a license on a VE license server.

Initial License Configuration for Sentinel HASP Licenses

To access a HASP license, the emulator needs the Sentinel runtime software. This software is part of the Charon-PAR for Linux installation kits. Refer to the software installation section for software installation instructions.

Please note: Charon-PAR version 3.0.0 and higher is **incompatible** with licenses of earlier versions. Earlier product versions are incompatible with licenses for Charon-PAR 3.0.0 and higher. Charon-PAR licenses with version 2.1.0 are not compatible with emulators running Charon-PAR version 2.0.0

If you have a hardware USB license dongle, simply connect it to the system after the Charon-PAR software has been installed and verify that the license can be read correctly by entering the following command:

```
$ /opt/charon/bin/hasp_srm_view -all
```

Please note that the above command must be run from a local connection. If you are connected over a remote connection (e.g., ssh), you will receive an error. As a workaround, you can display the license contents with the following command (adapt the path of the command if your installation location is different):

```
$ ssh localhost /opt/charon/bin/hasp_srm_view
```

Verify that the license contains the correct product, the correct number of emulator instances, and the correct expiration date (if applicable).

Example of a Charon-PAR/PA9-64 license:

```
$ hasp_srm_view -all

<lines removed>

The Physical KeyId: 1202236799
License type: License Dongle (Network capable)
CHARON Sentinel HASP License key section
Reading 4032 bytes

The License Number: N10.0064
The License KeyId: 1202236799
The Master KeyId: 1645066348

<lines removed>

End User name: Stromasys

Purchasing Customer name: Stromasys

<continued to the right>
```

```
Product License Number: Def_1
Virtual Hardware: PA-RISC
HP/PA Physical Key ID: 12345678
Product Name: Charon-PA9-64-L4
Product Code: CHPA9-64-L4-IP
Major Version: 3
Minor Version: 0
32bit and 64bit OS requirements: 64bit
Host Operating System required: LINUX
CPU's allowed: 8
Maximum virtual memory: 32768MB
License expiration date: 03-Jun-2021
Instances allowed: 9
Feature number: 1
Check interval: 60 minutes

<lines removed>
```

If you purchased a software license or if you want to use a network license server, or if you need more information about licensing in general and how to handle licenses in Charon-PAR, please refer to the section License Management Overview for Charon-PAR and in the Licensing Documentation. The license chapter contains important information about security and more complicated use cases (for example, multiple license keys, software licenses, and network licenses).

Charon-PAR Basic Guest Configuration and Installation

This section shows a basic configuration and setup of emulator instances for Charon-PA3 and Charon-PA9.

Please note: The examples are for illustrative purposes only. For a production installation, make sure to plan your host and guest system configurations according to your needs.

Contents

Preparatory Steps

The following points should be considered before starting with a basic test setup:

For a production setup, there are many additional topics to consider and decisions to be made. These are outside the scope of this document. Please contact your Stromasys representative if you need support for migrating production environments.

Storage Directory Structure

Before you begin to configure Charon-PAR emulator instances and to install guest systems, it is advisable to create a directory structure for storing the data required by Charon-PAR emulator instances and guest systems.

Space required for the guest operating systems:

- Charon-PAR/PA3: a minimum of 10GB
- Charon-PAR/PA9-64: a minimum of 16GB
- Charon-PAR/PA9-32: 1-2GB for HP-UX 9.0.7 and 10.20, a minimum of 10GB for HP-UX 11.00.

The numbers above are just basic guidelines for the minimum operating system installation. The space actually required depends on the individual configuration of the system to be migrated.

The following shows a **sample directory structure.** A structure similar to this will be used for the rest of the examples in this section:

```
# mkdir -p /data/Stromasys/host1

# mkdir /data/Stromasys/host1/Data

# mkdir /data/Stromasys/host1/Data/Disk

# mkdir /data/Stromasys/host1/Data/Tape
```

The Disk and Tape directories will be used to store emulated logical devices, for example, disk containers.

Please note:

- Each instance of Charon-PAR must be **started in a separate directory** because certain files (SSTORAGE.DAT, NVOLATILE.DAT, UA_SSTORAGE.DAT, and console.dat) will always be created in the current working directory of the emulator instance. Starting multiple emulator instances in the same directory will lead to errors due to file locking problems.
- **It is not recommended** to place emulator storage devices (in particular vdisks) on NFS as this will have a significant impact on performance. However, if any of the storage (e.g., ISO files or vdisks) is on an NFS share, NFS locking must be enabled and all intermediate firewalls between client and server must allow the port used by the **lockd** and **statd**. Failure to do so will cause the emulator to hang at startup.

Ethernet Interface Preparation

Cloud Considerations

If running Charon-PAR in a cloud environment please consider that every cloud environment has specific networking characteristics that could conflict with interface configurations made by the user of the Charon-PAR host system. Please refer to the documentation provided by the cloud provider to understand the networking behavior of your cloud instance before you change any interface settings (failing to do so may end with your cloud instance being permanently unreachable).

With respect to the supported Charon-PAR emulated Ethernet configurations, please bear in mind the following when deploying Charon in a cloud environment:

- If dedicating a host NIC to the emulator, the emulated system and the guest system must use the same MAC address and IP address on the interface that were assigned to the interface by the cloud provider. On the Linux level, the IP address must be removed from the interface. A dedicated NIC can be a RAW or MACVTAP configuration (MACVTAP support was added in Charon-PAR 3.0.9). For cloud settings, MACVTAP is recommended. Please refer to Ethernet Configuration Notes for more information.
- If using a TAP interface, the bridge must be an internal bridge (not connected to the cloud LAN via a host NIC). The bridge must use a private address range different from the cloud LAN. The emulator can then communicate with the outside world via NAT, or with other other private networks via a routing configuration.

Red Hat and CentOS Version 7.x

Even though it is not mandatory, it is good practice to exclude the Ethernet interfaces that will be used by the guest system running on Charon-PAR from being managed by the NetworkManager. This can be done for physical NICs dedicated to the guest operating system, or for virtual bridge and TAP configurations (that allow the host and the guest to share a physical interface).

In a network environment controlled by the NetworkManager, the NetworkManager—if configured accordingly—will create and manage the interface configuration files. This may create conflicts with the requirements of the Charon-PAR guest system. If this is the case, a manual configuration should be considered.

There are several options to prevent such problems:

- If the NetworkManager is not needed for other purposes, you can disable (commands: **# systemctl stop NetworkManager; systemctl disable NetworkManager**) and create the initial ifcfg-<interface> files in **/etc/sysconfig/network-scripts/** manually.
- If the NetworkManager is required for other purposes,
  - make sure it uses the ifcfg-files (**plugins=ifcfg-rh** must be enabled in section **[main]** of **/etc/NetworkManager/NetworkManager.conf**),
  - stop the NetworkManager (command**: # systemctl stop NetworkManager**),
  - exclude all interfaces required for the Charon-PAR guest(s) from NetworkManager control by adding the line **NM_CONTROLLED=no** to the respective ifcfg-<interface> file(s),
  - restart the NetworkManager (command: **# systemctl start NetworkManager**).

Example of a minimal ifcfg-<interface> file in **/etc/sysconfig/network-scripts/**:

```
NM_CONTROLLED=no
DEVICE=eth0
HWADDR=00:11:22:33:44:55
BOOTPROTO=none
ONBOOT=yes
```

Example of a minimal bridge configuration via ifcfg-<interface> files in **/etc/sysconfig/network-scripts/**:

Bridge definition	Physical NIC	TAP interface

DEVICE=br_enp0s8
NAME=br_enp0s8
TYPE=Bridge
ONBOOT=yes
DEFROUTE=yes
STP=no
BOOTPROTO=none
IPADDR=10.0.0.1
NETMASK=255.255.255.0
GATEWAY=10.0.0.10
NM_CONTROLLED=no
	DEVICE=enp0s8
	NAME=enp0s8
	TYPE=Ethernet
	ONBOOT=yes
	NM_CONTROLLED=no
	BRIDGE=br_enp0s8
		DEVICE=tap0_enp0s8
		NAME=tap0_enp0s8
		BRIDGE=br_enp0s8
		TYPE=Tap
		ONBOOT=yes
		NM_CONTROLLED=no

Please refer to your host system's man-pages for additional information about the NetworkManager.

There are other settings required to disable certain off-load functions on interfaces dedicated to a Charon-PAR guest system. These settings are configured in the emulator configuration file and described further down in this document.

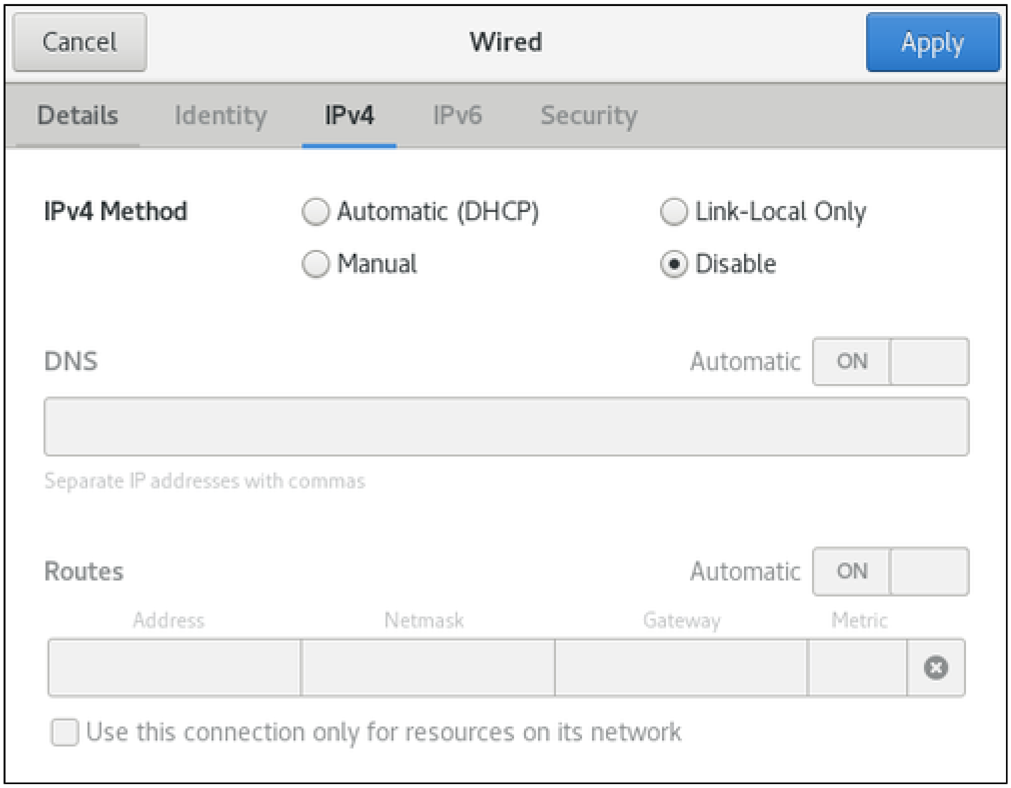
Red Hat Version 8.x and Higher

Starting with CentOS/Red Hat/Oracle Linux 8.x, the network-scripts package has been deprecated and the bridge-utils package is no longer available in the standard repositories. Network management for these Linux versions has been changed to use the NetworkManager capabilities. Network management can be handled via the Gnome GUI, via **nmcli** commands, or via the **nmtui** utility. Please refer to the documentation of your Red Hat or CentOS version for more information.

In RHEL 8, the network-scripts package can be installed optionally to maintain the same network management methods as used in Linux version 7. In RHEL 9, the package is no longer available.

Example for Graphical User Interface

The following image shows a sample configuration that disables the IPv4 configuration for an interface that is to be dedicated to the emulator using the GNOME network settings.



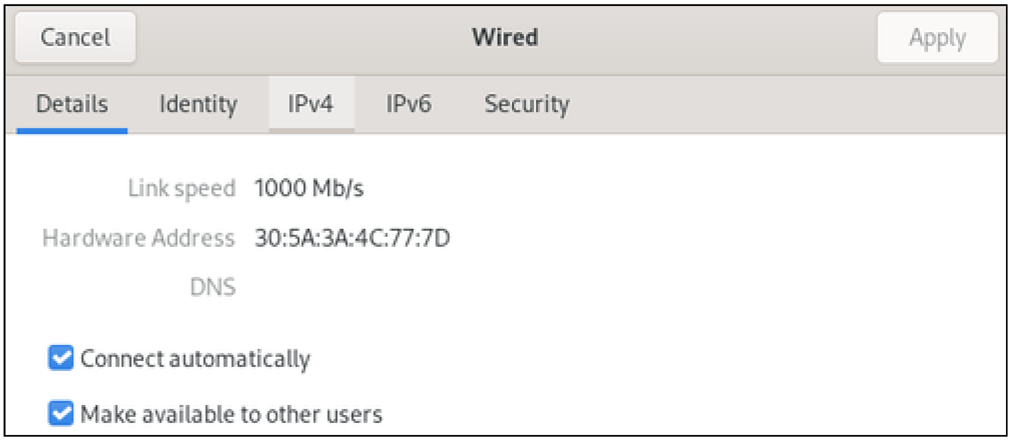
IPv4 Method ○ Automatic (DHCP) ○ Link-Local Only ○ Manual ● Disable

DNS Automatic ON

Routes Automatic ON

□ Use this connection only for resources on its network

Also, make sure to include the option **Make available to other users** in the **Details** tab:



Link speed 1000 Mb/s

Hardware Address 30:5A:3A:4C:77:7D

DNS

☑ Connect automatically

☑ Make available to other users

You can use the following command to disable IPv6 for an interface (enp0s8 in the example):

```
# sysctl -w net.ipv6.conf.enp0s8.disable_ipv6=1
```

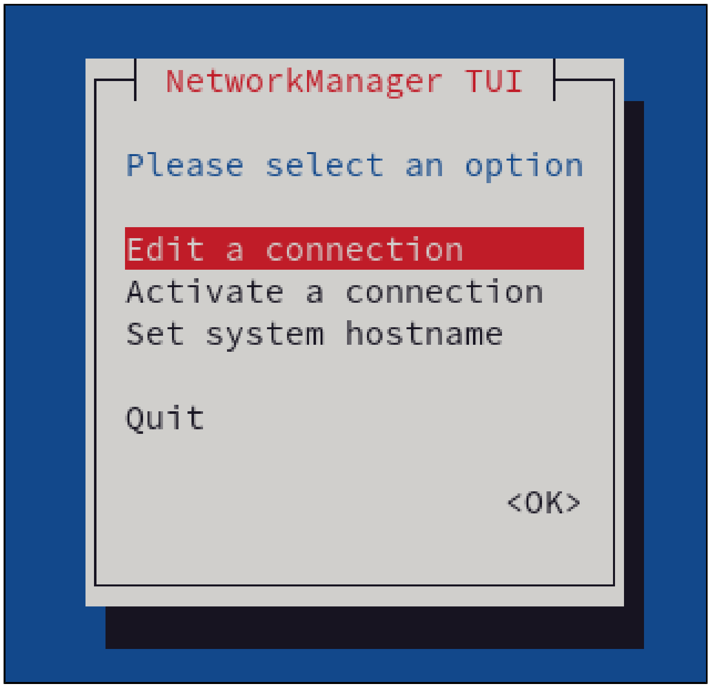
To make this setting permanent, add it to */etc/sysctl.conf*.

Example for Non-Graphical User Interface

On a non-graphical interface, you can use the **nmcli command-line utility** to create a network connection as shown in the following example:

```
# nmcli conn add type ethernet autoconnect yes \
con-name <physical-int-name> \
ifname <pysical-int-name> ip4 0.0.0.0
```

Or you can use the **nmtui text-based utility** (separate RPM package). It provides a text-based configuration menu.



Please note: after changes via **nmcli** or **nmtui** you must restart the NetworkManager (`systemctl restart NetworkManager.service`).

Charon-PAR/PA3 Guest - Basic Configuration and Installation

Contents

Please note: All Linux sample commands in this section are performed as the root user (prompt = #). This may not be necessary for all commands in all environments depending on the local permission settings. The commands can be executed on the Charon host graphical console or via a non-graphical login (e.g., SSH). However, for accessing the running emulator, the use of PuTTY is strongly recommended because it works best with the MPE/iX environment.

The Charon-PAR/PA3 functionality of Charon-PAR emulates historic PA-RISC systems for MPE/iX. This section shows a sample installation of an MPE /iX guest system

Emulator Licensing Considerations

The emulator needs a valid license to run. The license can be a Sentinel HASP license or a Virtual Environment (VE) license. Please refer to Initial Emulator License Configuration for the basic license configuration steps of both options. The license(s) to be used are defined in the configuration file. If running in a cloud environment, a VE license is strongly recommended (supported from Charon-PAR version 3.0.6).

Downloading and Unpacking the Provided MPE/IX Disk Container

If not already done, please download the image with the preconfigured MPE/IX 7.5 disk container from Stromasys. Contact your Stromasys representative or your VAR for download instructions. The preconfigured LDev1 image allows the emulated system to boot without having to separately install or update MPE/iX.

Once the image has been downloaded, copy it to the Disk directory and unpack it:

	Step	Command
1	Copy/uncompress the image to the Disk directory.	`# bzcat mpe75a.dsk.bz2 > /data/Stromasys/host1/Data/Disk/ldev1.dsk`

After this step, if the image is not needed for other systems, you can delete the compressed image to save disk space.

Adapting the Configuration File Template

The configuration file template must be adapted to the customer environment and the required guest system characteristics. This section describes the minimum number of changes required to configure a Charon-PAR/PA3 emulator instance. Please refer to Configuration File Reference for a detailed description of the configuration file options.

Perform the following steps to create a basic configuration file for a Charon-PAR/PA3 emulator instance:

	Step	**Description (configuration file settings are examples!)**

1	Create a copy of the configuration file template.	Example:
		`# cd /data/Stromasys/host1/`
		`# cp  /opt/charon/cfg/pa3.cfg host1.cfg`

2	Open the configuration file in a text editor.	`# vi host1.cfg`
		or
		`# gedit host1.cfg`

3	Configure the system model.	`model "A400-100-110"`
		`Configured model must be covered by your license.`

4	Configure the RAM for the emulated system.	`memory 2G`
5	Configure LDEV 1 (preconfigured system disk).	Change definition of DKA0 to: `DKA0.image="/data/Stromasys/host1/Data/Disk/ldev1.dsk"`

6	Configure a tape devices.	Change definition of MKAx to:
		`MKA600.image="/data/Stromasys/host1/Data/Tape/ldev7.img"`
		`MKA500.image="/data/Stromasys/host1/Data/Tape/ldev8.img"`

7	Configure a network card based on a physical host NIC. This NIC is dedicated to the guest operating system and cannot be used for network communication by the host system. To share an interface, a virtual bridge would have to be used (see Ethernet configuration sections in this document).	In the configuration file template, comment out the dummy interface and the tap interface configuration. Then uncomment the interface configuration of the physical interface example. Change the definition of EWA0 to the interface that is assigned to the emulated system (example: eth1) and disable the offload parameters for the interface:
		`EWA0.mapping_mode="RAW"`
		`EWA0.iface="eth1"`
		`EWA0.initialize_command="ethtool -K $IFACE rx off ; ethtool -K $IFACE tx off ; ethtool -K $IFACE sg off ; ethtool -K $IFACE gso off ; ethtool -K $IFACE gro off ; ethtool -K $IFACE txvlan off ; ethtool -K $IFACE rxvlan off"`
		Depending on the capabilities of the Ethernet device, not all offload parameters maybe be available. This will create a warning message. The variable **IFACE** is set by the system before executing the initialize command.
		For cloud environments, please note:
		• The MAC address on an interface must be the same as the one configured by the cloud provider on the dedicated interface. This should be the case by default for Charon-PAR. The IP address used by the guest OS must be the one configured by the cloud provider.
		• Virtual bridge configurations that are linked to a host NIC are not possible in cloud environments. Only "internal virtual bridge" configurations are possible. Such bridges can be used for host-guest communication. External communication for the guest can be achieved either by NAT or by routing configurations (depending on requirements and cloud-specific restrictions). A less complex, TAP based solution is to use a MACVTAP interface (using MAC and IP address assigned by the cloud provider).

8	Save the configuration file.	

Starting the Charon-PAR/PA3 Emulator Instance

Please note: The examples in this section assume that a **graphical interface** is used, on which a PuTTY session can be run. **Should this not be possible**, you can access the console of the emulated system by using, for example, a telnet client and pointing it to the TCP port on the Charon host defined for the serial console port in the configuration file. The default port is 30000 (sample command to connect to an emulator on the local system: $ `telnet` *`localhost`* `30000`). In this case, do not start the PuTTY program automatically from the emulator configuration.

To start the Charon-PAR/PA3 emulator instance, perform the following steps:

	Step	Command
1	Go to the directory where the configuration file is located.	`# cd /data/Stromasys/host1/`
2	Start the emulator instance.	`# /opt/charon/bin/charon-par -f host1.cfg`

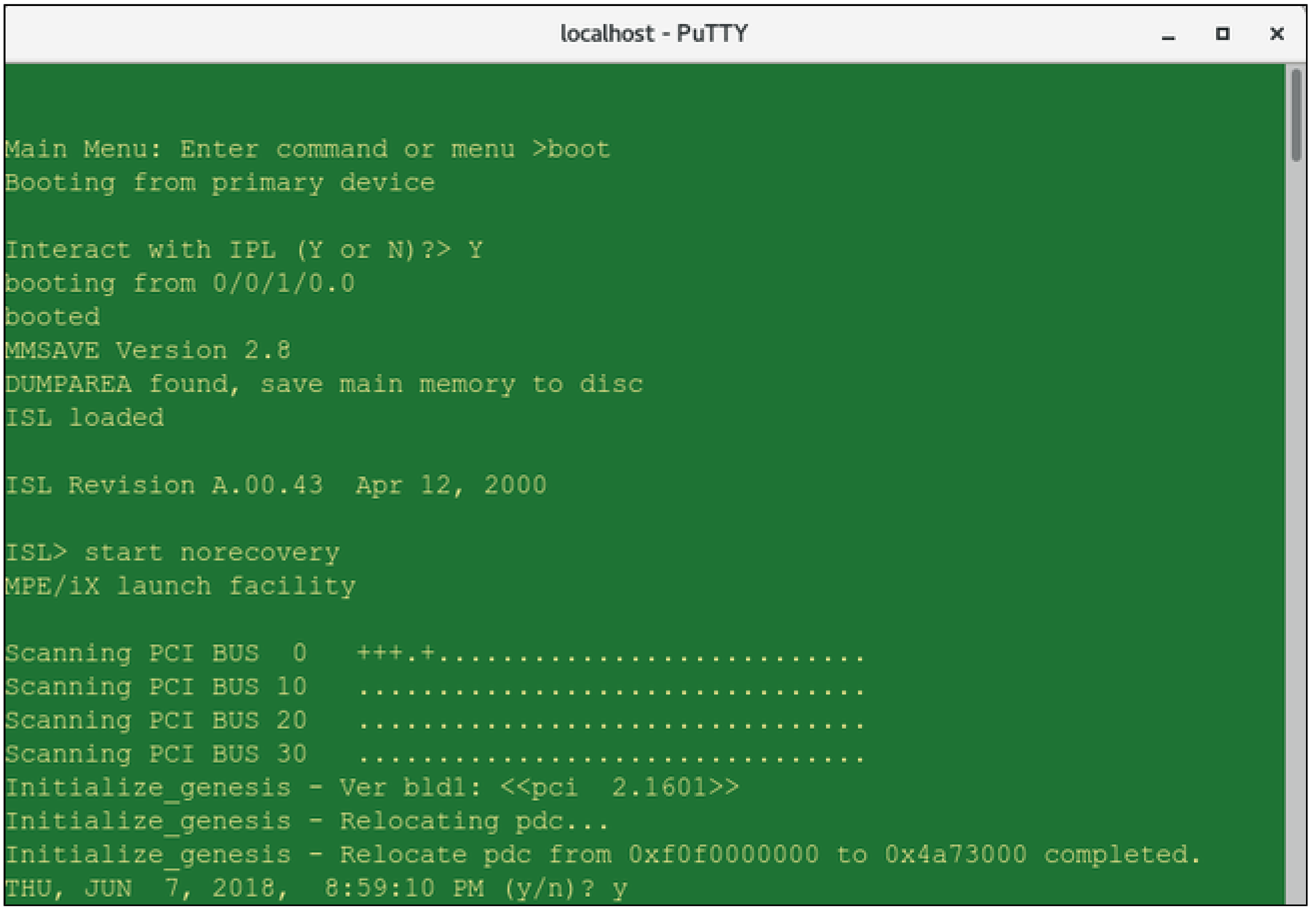
If everything works, you will see log output in the current terminal window ending in a **pa3>** prompt. This is the Charon-PAR emulator console. **Please note**: the prompt may be obscured by log messages and not be immediately visible. In this case, press the **Enter** key once to display the prompt.

Should you receive an error indicating that PuTTY cannot load the preconfigured font, open PuTTY as the root user, load the **PAR-Telnet** profile, select a font that is available on your host system and save the profile.

At the same time, a **green PuTTY console window** will open. In this green console window perform the following steps:

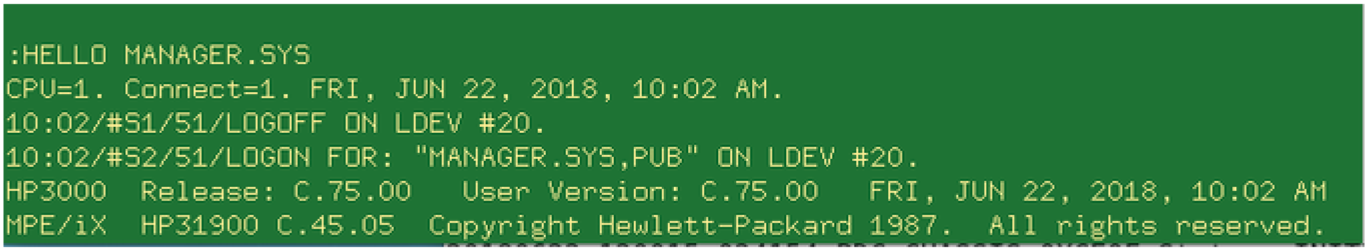
- Press the **Enter** key.
- This should produce the prompt **Main Menu: Enter command or menu >**.
- Enter the command **boot** and confirm by pressing the **Enter** key.
- This should produce the prompt **Interact with IPL (Y or N)?>**
- Press **Y.**
- The system should stop at the **ISL>** prompt.
- Type **start norecovery** and press the **Enter** key.
- The system will boot. It may prompt you for the correct date/time.
- After a successful boot you will see the **:** prompt.

The image below shows sample console interaction:



```
Main Menu: Enter command or menu >boot
Booting from primary device

Interact with IPL (Y or N)?> Y
booting from 0/0/1/0.0
booted
MMSAVE Version 2.8
DUMPAREA found, save main memory to disc
ISL loaded

ISL Revision A.00.43  Apr 12, 2000

ISL> start norecovery
MPE/iX launch facility

Scanning PCI BUS  0   +++.+...........................
Scanning PCI BUS 10   ................................
Scanning PCI BUS 20   ................................
Scanning PCI BUS 30   ................................
Initialize_genesis - Ver bld1: <<pci  2.1601>>
Initialize_genesis - Relocating pdc...
Initialize_genesis - Relocate pdc from 0xf0f0000000 to 0x4a73000 completed.
THU, JUN  7, 2018,  8:59:10 PM (y/n)? y
```

The image below shows how to use the **HELLO** command to log into the booted guest system:



```
:HELLO MANAGER.SYS
CPU=1. Connect=1. FRI, JUN 22, 2018, 10:02 AM.
10:02/#S1/51/LOGOFF ON LDEV #20.
10:02/#S2/51/LOGON FOR: "MANAGER.SYS,PUB" ON LDEV #20.
HP3000  Release: C.75.00   User Version: C.75.00   FRI, JUN 22, 2018, 10:02 AM
MPE/iX  HP31900 C.45.05  Copyright Hewlett-Packard 1987.  All rights reserved.
```

Basic Network Configuration

Default configuration

The preconfigured MPE/iX disk also includes preconfigured network settings:

- Hostname: demo.charon.com
- Network interface name: LAN
- IP address: 192.168.111.25

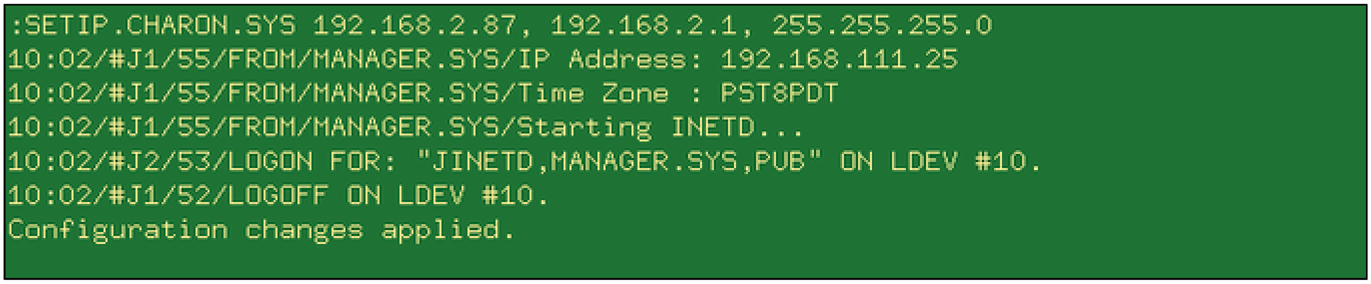
The default network configuration also includes the following configuration files:

- HOSTS.NET.SYS: contains aliases for 192.168.111.25.
- RESLVCNF.NET.SYS: contains the public DNS server IP addresses.
- NSSWITCH.NET.SYS: configures the name lookup to use HOSTS file, followed by DNS.

Changing the default configuration

Stromasys provides a special command file (**SETIP.CHARON.SYS**) in the preconfigured MPE image. This command file enables the user to configure the network without using a block-mode terminal that would otherwise be required to access the relevant MPE utility (**NMMGR**).

The command file takes three parameters: host IP address, default gateway address, and netmask. The image below shows a sample:



```
:SETIP.CHARON.SYS 192.168.2.87, 192.168.2.1, 255.255.255.0
10:02/#J1/55/FROM/MANAGER.SYS/IP Address: 192.168.111.25
10:02/#J1/55/FROM/MANAGER.SYS/Time Zone : PST8PDT
10:02/#J1/55/FROM/MANAGER.SYS/Starting INETD...
10:02/#J2/53/LOGON FOR: "JINETD,MANAGER.SYS,PUB" ON LDEV #10.
10:02/#J1/52/LOGOFF ON LDEV #10.
Configuration changes applied.
```

After changing the configuration, the command file will automatically restart the network to activate the changes.

For cloud environments, please note: in case of a dedicated Ethernet interface, the IP address of the emulator must be set to the value assigned by the cloud provider to the corresponding host NIC.

Stopping the Emulator

In order to stop the emulator in an orderly manner, perform the following steps:

1. Shut down the guest operating system:

    a) Go to the console of the emulated system.

    b) Press the key combination **CTRL+A**. This will display the **=** prompt.

    c) At the **=** prompt, type **`SHUTDOWN SYSTEM`**.

    d) Wait until the system displays the "`Shutdown of operating system complete. (Shut 6)`" message.

    Please note: The **=SHUTDOWN** command only works on the system console. The **:SHUTDOWN** command does not have this requirement. However, it is not as safe a method as the **=SHUTDOWN** command.

2. Starting with version 3.0.5, the emulator is stopped automatically after the guest operating system shutdown is complete (unless the configuration parameter `system.stop_on_halt` has been set to **`false`**). If the emulator is not stopped automatically after the shutdown of the guest operating system, different methods must be used to stop it in different situations:

- **The emulator was started in interactive mode and the Charon-PAR console is available in current terminal**: enter the **exit** command at the **pa3>** prompt and hit **Enter**. This will cause the emulator to close the disk images of the emulated system and to exit.
- **The Charon-PAR console is available on a local TCP port (emulator started with -c or --console-port parameter)**: this situation will exist most frequently if the emulator has been started as a daemon or service. To stop the emulator access the Charon-PAR console via **telnet** to *localhost* on the port specified when starting the emulator, enter the **exit** command at the **pa3>** prompt, and hit **Enter**.
  Access to the Charon-PAR console via a TCP port is available starting with Charon-PAR version 3.0.1. The **CTRL+B** key combination has been disabled on the emulator console starting with this version.
- **The emulator was started as a daemon or service and the Charon-PAR console is not available via a local TCP port**: This situation can occur if you run a version before 3.0.1 or you started the emulator without the **`-c`** option.
  - For versions 3.0.1 and higher: the emulator must be stopped using the **kill** command from the root user on the Charon host system. Perform the following steps to stop the Charon-PAR process:
    - Find the PID (process id) of the process: `#` **`ps -ef | grep -i charon-par`**
    - Stop the process: `#` **`kill`** *`<charon-par-pid>`*
  - For versions before 3.0.1: on the emulator console, after shutting down the guest operating system, press **CTRL+B**. After a few seconds, the PDC console prompt (**Main menu:**) will appear. Type **exit** to stop the emulator.

For more information about the Charon-PAR console, please refer to Charon-PAR Console. For more information about the Charon-PAR command-line options, please refer to Charon-PAR Command-Line Options.

Charon-PAR/PA9-64 Guest - Basic Configuration and Installation

Contents

Please note: All Linux and HP-UX sample commands in this section are performed as the root user (prompt = #). This may not be necessary for all commands in all environments depending on the local permission settings. The commands can be executed on the Charon host graphical console or via a non-graphical login (e.g., SSH). However, for accessing the running emulator, the use of PuTTY is strongly recommended because it works best with the emulator environment.

The Charon-PAR/PA9-64 functionality of Charon-PAR emulates historic 64-bit PA-RISC hardware. This section shows a sample installation of a 64-bit guest system.

Emulator Licensing Considerations

The emulator needs a valid license to run. The license can be a Sentinel HASP license or a Virtual Environment (VE) license. Please refer to Initial Emulator License Configuration for the basic license configuration steps of both options. The license(s) to be used are defined in the configuration file. If running in a cloud environment, a VE license is strongly recommended (supported from Charon-PAR version 3.0.6).

HP-UX ISO Installation Medium

To install a basic HP-UX system running as a guest system in an emulator instance, you need to provide an HP-UX ISO installation medium.

Example:

	Step	Command
1	Change to the directory where you store your virtual disks.	`# cd /data/Stromasys/host1/Data/Disk/`
2	Unpack an ISO file contained in a compressed tar archive into this directory (**example only**: the actual command will vary depending on the location and archive format of your ISO file).	`# tar -xzf /path-to-file/HPUX_11v1_Sept2005_FOE.tgz`

If the compressed tar archive of the ISO file is no longer needed for other purposes, it can be removed.

Creating a Virtual Disk Container for the System Disk

The emulated system requires a disk on which the HP-UX system will be installed. Create a system disk as shown in the following example:

	Step	Command
1	Change to the directory where you store your virtual disks.	`# cd /data/Stromasys/host1/Data/Disk/`

2	Create an empty disk container. Note that the actual size of the disk will depend on the requirements of your emulated system. The disk name shown is just an example.	Example virtual disk creation (20GB):
		With **dd**:
		`# dd if=/dev/zero of=ldev1-v11.dsk bs=1G count=20`
		With **fallocate** (very fast alternative):
		`# fallocate --length 21474836480` *ldev1-v11.dsk*
		Please note: the size of the disk must be such that it is aligned to a 512-byte boundary. You can check this with the following bash expression:
		`$ bytes=<number-of-bytes>; echo $(( bytes - bytes/512*512 ))`
		If it returns 0, the alignment is correct.

Adapting the Configuration File Template

The configuration file template must be adapted to the customer environment and the required guest system characteristics. This section describes the minimum number of changes required to configure a Charon-PAR/PA9-64 emulator instance. Please refer to Configuration File Reference for a detailed description of the configuration file options.

Perform the following steps to create a basic configuration file for a Charon-PAR emulated historic 64-bit PA-RISC system for HP-UX (rp2400 sample):

	Step	Description (configuration file settings are examples only!)

1	Create a copy of the configuration file template.	Example:
		`# cd /data/Stromasys/host1/`
		`# cp /opt/charon/cfg/rp2400.cfg host1.cfg`

2	Open the configuration file in a text editor.	`# vi host1.cfg`
		or
		`# gedit host1.cfg`

3	Configure the system model.	`model "rp2400-1-650"`
		`Configured model must be covered by your license.`

4	Configure the RAM for the emulated system.	`memory 2G`
5	Configure the serial console for telnet via PuTTY.	Use the method 2 section in the configuration file and make sure the definition of the serial.uart0. device is: `serial.uart0.device.command="putty -load PAR-Telnet-VT100"`
6	Configure disk 1 (system disk).	Change definition of DKA0 to: `DKA0.image="/data/Stromasys/host1/Data/Disk/ldev1-v11.dsk"`

7	Add a definition for disk 2 (ISO file) or (if it already exists) modify the definition to point to your ISO file.	`load DKA100`
		`DKA100.image="/data/Stromasys/host1/Data/Disk/HP-UX_11v1_Sept2005_FOE.iso"`

8	Configure a tape devices.	Change definition of MKAx to:
		`MKA600.image="/data/Stromasys/host1/Data/Tape/ldev7.img"`
		`MKA500.image="/opt/Stromasys/host1/Data/Tape/ldev8.img"`

9	Configure a network card based on a physical host NIC. This NIC is dedicated to the guest operating system and cannot be used for network communication by the host system. To share an interface, a virtual bridge would have to be used (see Ethernet configuration sections in this document).	In the configuration file template, comment out the dummy interface and the tap interface configuration. Then uncomment the interface configuration of the physical interface example. Change the definition of EWA0 to the interface that is assigned to the emulated system (example: eth1) and disable the offload parameters for the interface: **`EWA0.mapping_mode="RAW"`** **`EWA0.iface="eth1"`** **`EWA0.initialize_command="ethtool -K $IFACE rx off ; ethtool -K $IFACE tx off ; ethtool -K $IFACE sg off ; ethtool -K $IFACE gso off ; ethtool -K $IFACE gro off ; ethtool -K $IFACE txvlan off ; ethtool -K $IFACE rxvlan off"`** Depending on the capabilities of the Ethernet device, not all off-load parameters maybe be available. This will create a warning message. The variable **IFACE** is set by the system before executing the initialize command. **For cloud environments, please note**: • The MAC address on an interface must be the same as the one configured by the cloud provider on the dedicated interface. This should be the case by default for Charon-PAR. The IP address used by the guest OS must be the one configured by the cloud provider. • Virtual bridge configurations that are linked to a host NIC are not possible in cloud environments. Only "internal virtual bridge" configurations are possible. Such bridges can be used for host-guest communication. External communication for the guest can be achieved either by NAT or by routing configurations (depending on requirements and cloud-specific restrictions). A less complex, TAP based solution is to use a MACVTAP interface (using MAC and IP address assigned by the cloud provider).
10	Save the configuration file.	

Starting the Emulated System and Installing the Guest System

This section shows how to start the emulator and install a HP-UX system.

Please note:

- The examples in this section assume that a **graphical interface** is used, on which a PuTTY session can be run.
- **Should this not be possible**, you can access the console of the emulated system by using, for example, a telnet client and pointing it to the TCP port on the Charon host defined for the serial console port in the configuration file. The default port is 30000. Sample command to connect to an emulator on the local system:

  ```
  $ telnet localhost 30000.
  ```

  In this case, do not start the PuTTY program automatically from the emulator configuration.

Starting the Emulator

To start the emulated Charon-PAR/PA9-64 system, perform the following steps:

	Step	Command
1	Go to the directory where the configuration file has been stored.	`# cd /data/Stromasys/host1/`
2	Start the emulator instance.	`# /opt/charon/bin/charon-par -f host1.cfg`

If everything works, you will see log output in the current terminal window ending in a **pa9-64>** prompt. This is the Charon-PAR emulator console. It displays log information, allows to set configuration parameters and to stop the emulator. **Please note**: the prompt may be obscured by log messages and not be immediately visible. In this case, press the **Enter** key once to display the prompt.

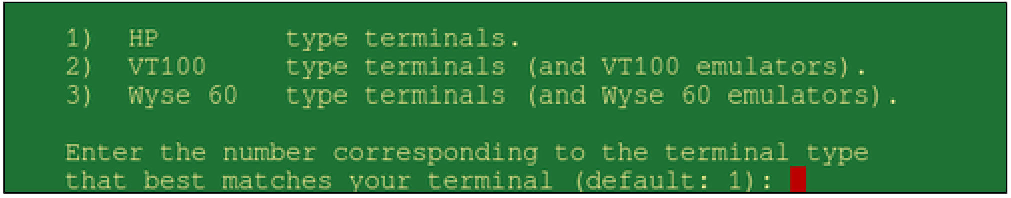
At the same time, a **green PuTTY console window** will open.

Please note: Should there be an error from PuTTY about missing fonts, start PuTTY as the root user, load the **PAR-Telnet-VT100** profile, set a font that is available on the host system, save the configuration.

Booting from the Installation ISO File

In green PuTTY console window perform the following steps to boot the installation ISO file:

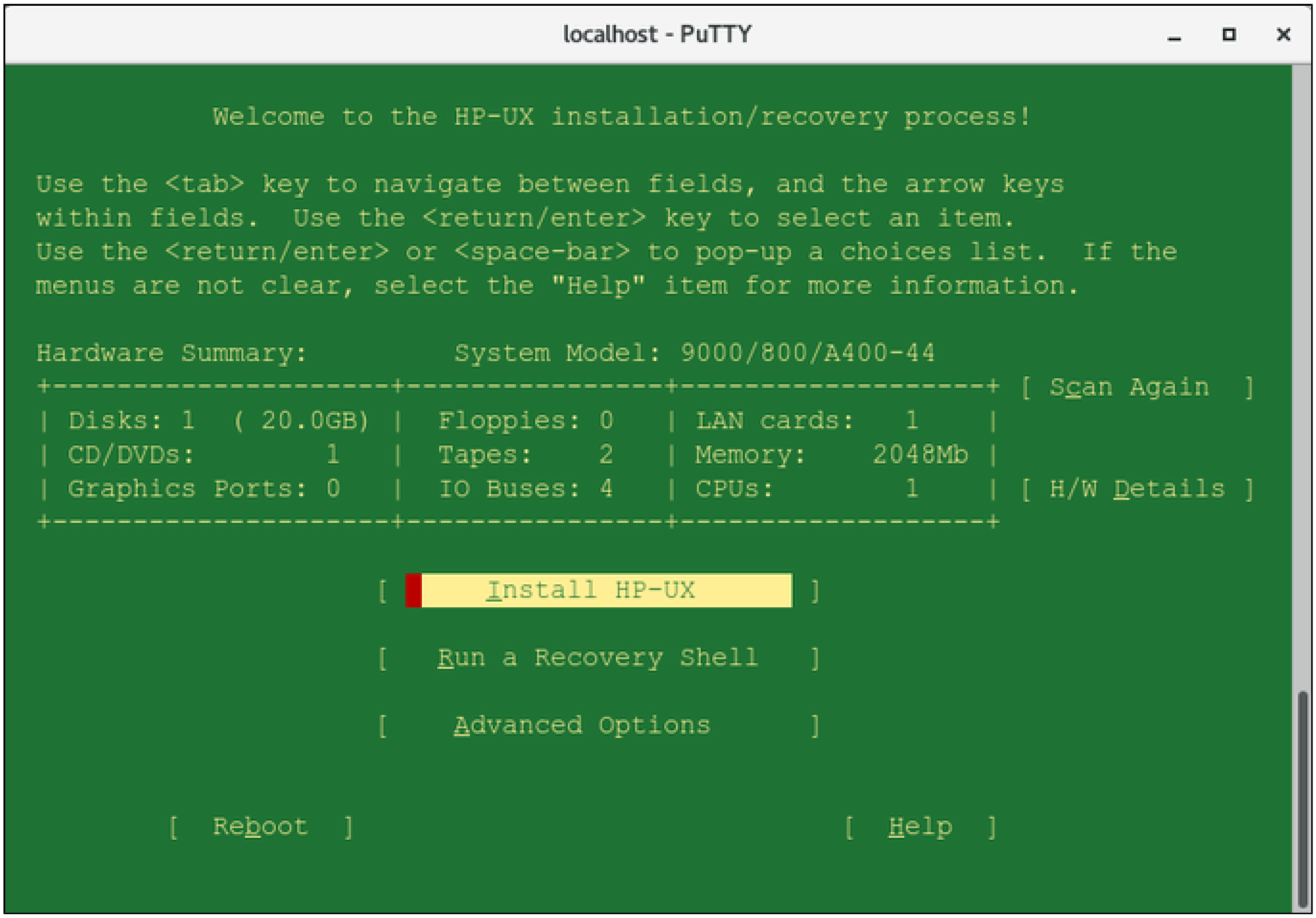
- Press the **Enter** key.
- This should produce the prompt **Main Menu: Enter command or menu >**.
- Enter the command **`boot 0/0/1/0.1.0`** to boot from the ISO and confirm by pressing the **Enter** key.
- At the prompt Interact with IPL (Y or N)?> answer **N**. The system will continue to boot.
- At the next prompt, select **VT100** as the terminal (option 2).



```
   1)  HP        type terminals.
   2)  VT100     type terminals (and VT100 emulators).
   3)  Wyse 60   type terminals (and Wyse 60 emulators).

   Enter the number corresponding to the terminal type
   that best matches your terminal (default: 1):
```

HP-UX Installation Phase 1: Pre-installation Dialogs

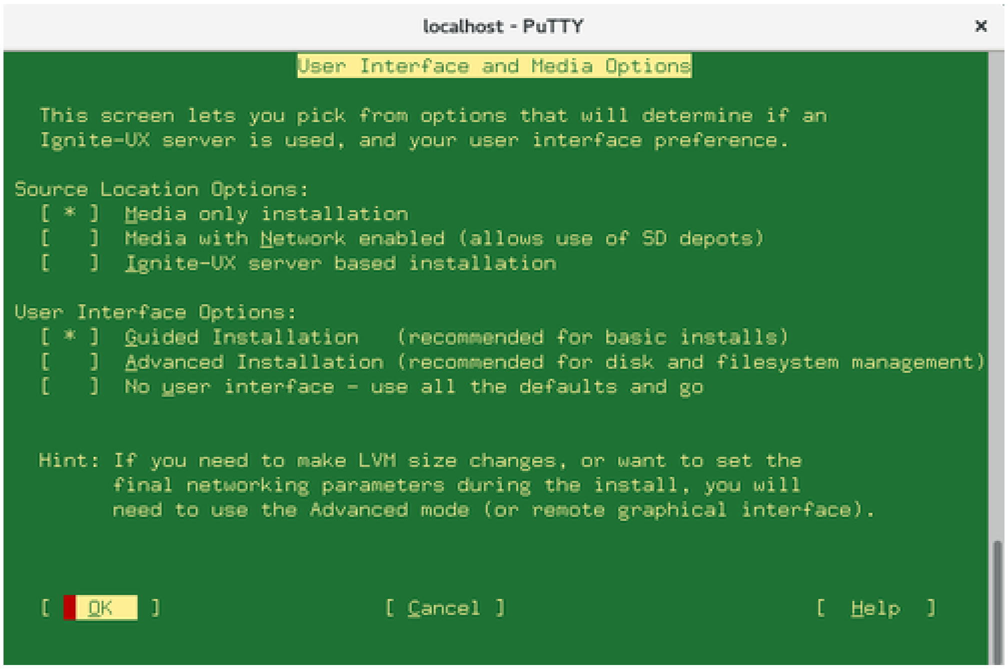
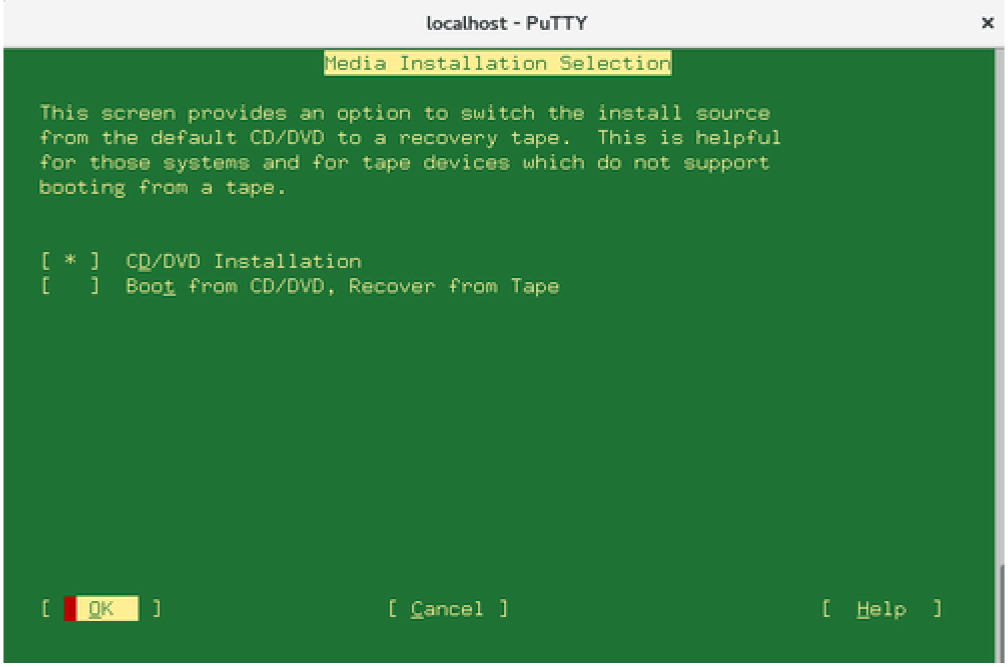
At the screen following the terminal selection you can select to **install HP-UX** as shown in the following image:



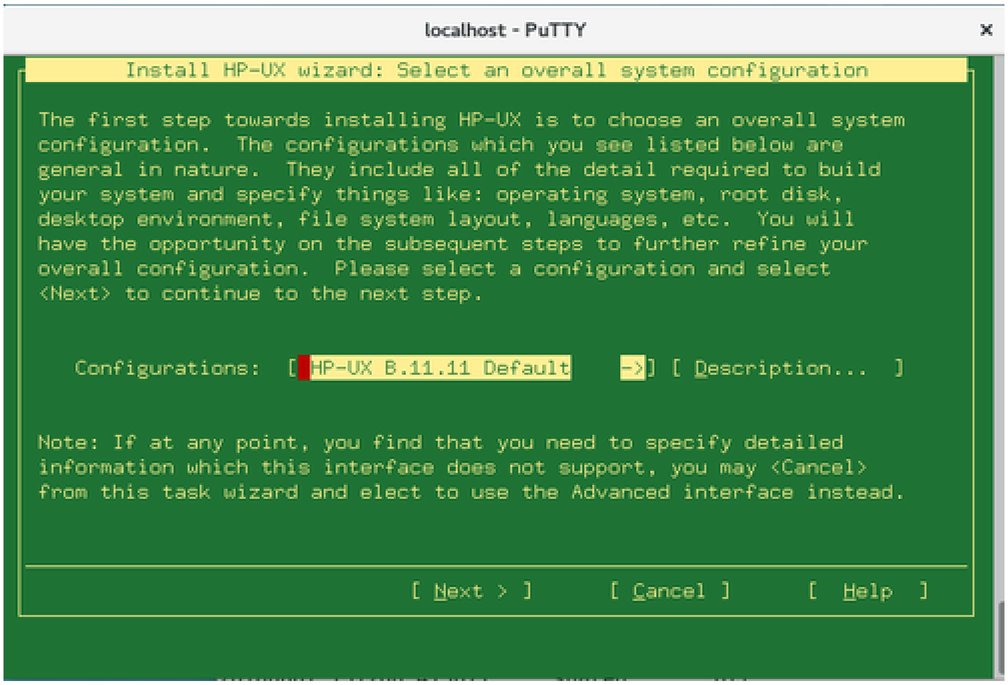
```
            Welcome to the HP-UX installation/recovery process!

 Use the <tab> key to navigate between fields, and the arrow keys
 within fields.  Use the <return/enter> key to select an item.
 Use the <return/enter> or <space-bar> to pop-up a choices list.  If the
 menus are not clear, select the "Help" item for more information.

 Hardware Summary:         System Model: 9000/800/A400-44
 +----------------------+----------------+-------------------+ [ Scan Again  ]
 | Disks: 1  ( 20.0GB)  |  Floppies: 0   | LAN cards:   1    |
 | CD/DVDs:         1   |  Tapes:    2   | Memory:   2048Mb  |
 | Graphics Ports: 0    |  IO Buses: 4   | CPUs:        1    | [ H/W Details ]
 +----------------------+----------------+-------------------+

                      [     Install HP-UX      ]

                      [  Run a Recovery Shell  ]

                      [   Advanced Options     ]


          [  Reboot  ]                            [  Help  ]
```

- Check if all configured devices were found as expected. If something is missing, check your configuration file.
- For a basic test installation use the default choices in this and the other initial configuration menus. You can type the **underlined character** to select a choice. Sometimes you will have to confirm the selection by pressing the **Enter** key, sometimes the selection will also activate the choice.
- Continue with **Install HP-UX**.

The following table provides an overview of the information that will be collected.
Please note: the number and content of screens may vary depending on your selections.

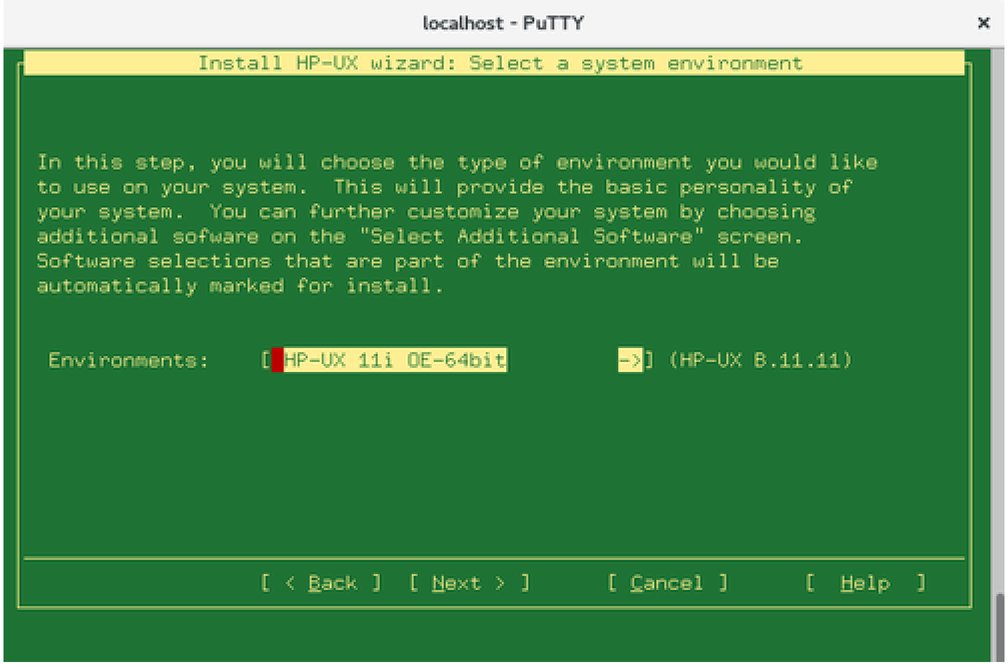
Information requested	Illustration
Select the type of installation and media. Continue with **OK**. In this basic example, an installation from local media is selected. Alternatively, you can choose to enable the network and use software depots for additional software to be installed, or you can use an Ignite installation server. For these advanced options, please refer to the HP-UX documentation.	 localhost - PuTTY User Interface and Media Options This screen lets you pick from options that will determine if an Ignite-UX server is used, and your user interface preference. Source Location Options: [*] Media only installation [ ] Media with Network enabled (allows use of SD depots) [ ] Ignite-UX server based installation User Interface Options: [*] Guided Installation (recommended for basic installs) [ ] Advanced Installation (recommended for disk and filesystem management) [ ] No user interface - use all the defaults and go Hint: If you need to make LVM size changes, or want to set the final networking parameters during the install, you will need to use the Advanced mode (or remote graphical interface). [OK] [Cancel] [Help]
Select if you want to install from the media or recover the system. Continue with **OK**.	 localhost - PuTTY Media Installation Selection This screen provides an option to switch the install source from the default CD/DVD to a recovery tape. This is helpful for those systems and for tape devices which do not support booting from a tape. [*] CD/DVD Installation [ ] Boot from CD/DVD, Recover from Tape [OK] [Cancel] [Help]

Select the basic system configuration. Continue with **Next**.

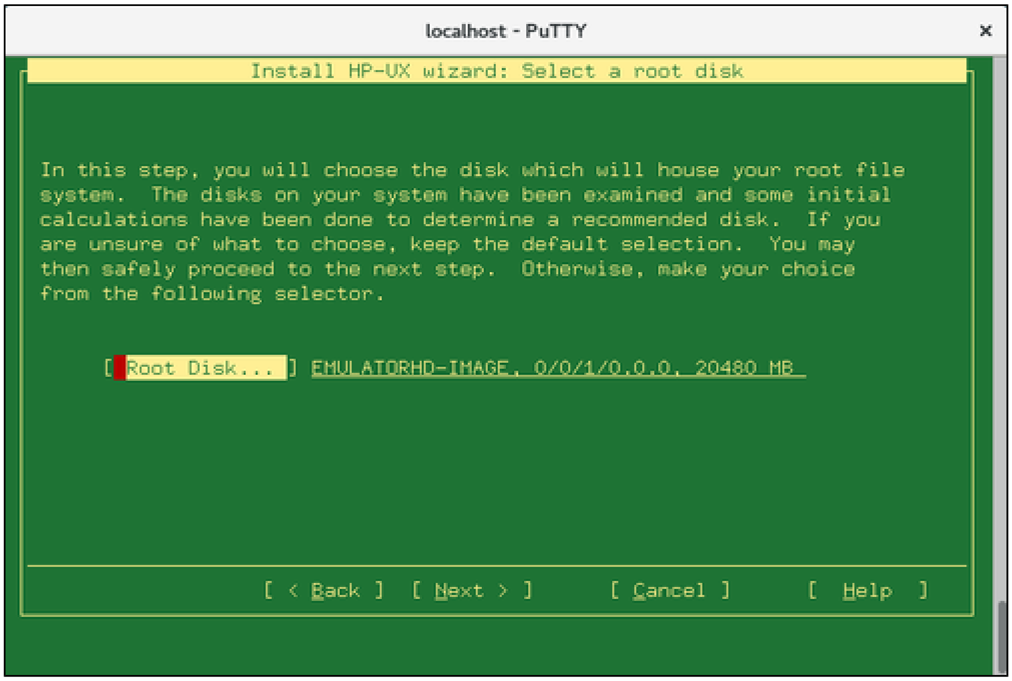
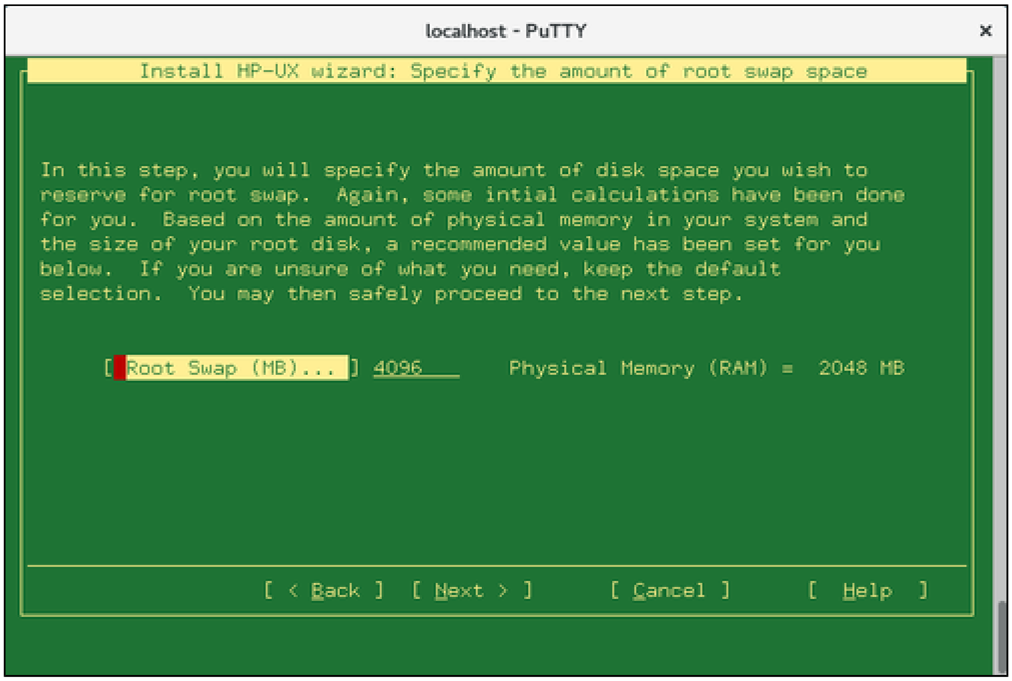


```
Install HP-UX wizard: Select an overall system configuration

The first step towards installing HP-UX is to choose an overall system
configuration.  The configurations which you see listed below are
general in nature.  They include all of the detail required to build
your system and specify things like: operating system, root disk,
desktop environment, file system layout, languages, etc.  You will
have the opportunity on the subsequent steps to further refine your
overall configuration.  Please select a configuration and select
<Next> to continue to the next step.


   Configurations:  [ HP-UX B.11.11 Default     ->] [ Description...  ]


Note: If at any point, you find that you need to specify detailed
information which this interface does not support, you may <Cancel>
from this task wizard and elect to use the Advanced interface instead.


                              [ Next > ]      [ Cancel ]      [  Help  ]
```

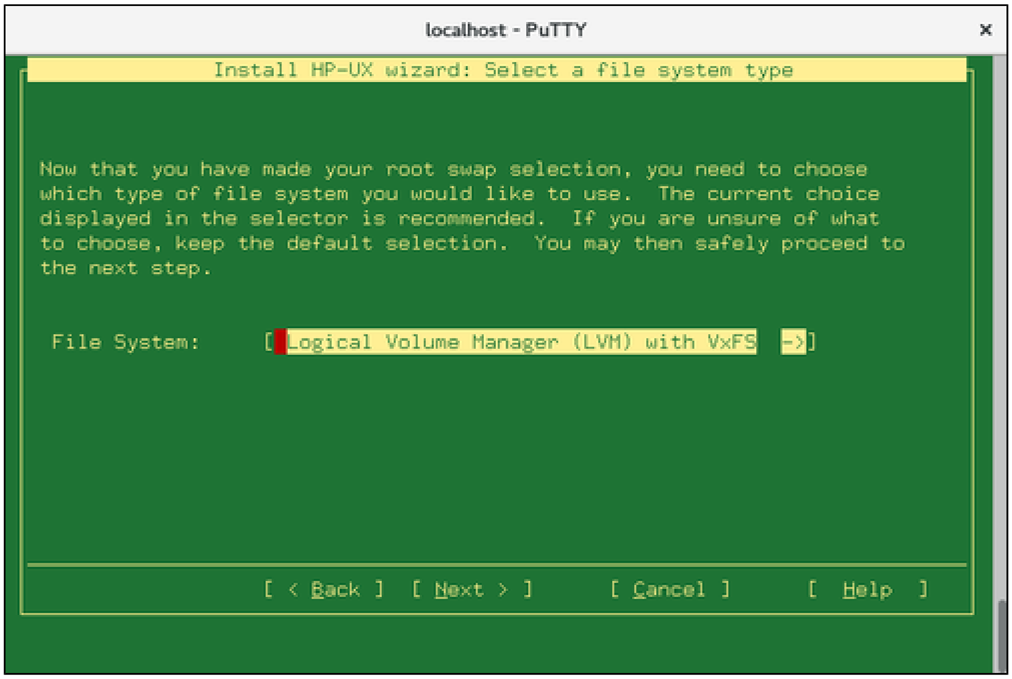
Select the operating system environment. Continue with **Next**.



```
Install HP-UX wizard: Select a system environment



In this step, you will choose the type of environment you would like
to use on your system.  This will provide the basic personality of
your system.  You can further customize your system by choosing
additional sofware on the "Select Additional Software" screen.
Software selections that are part of the environment will be
automatically marked for install.


 Environments:     [ HP-UX 11i OE-64bit         ->] (HP-UX B.11.11)








                   [ < Back ]  [ Next > ]      [ Cancel ]      [  Help  ]
```

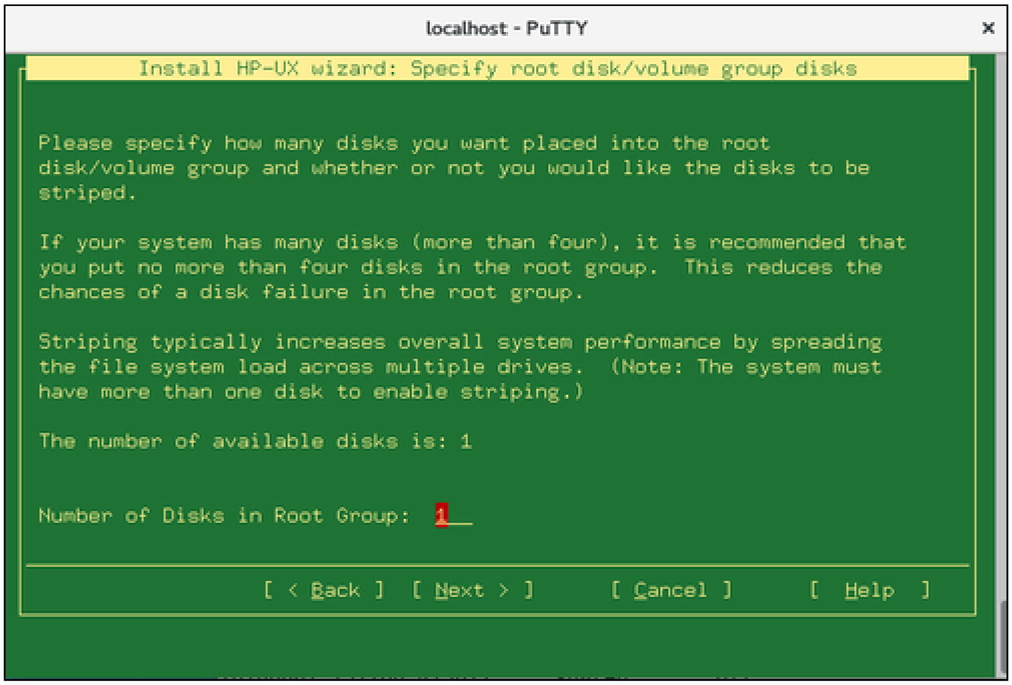
Select the root disk. Continue with **Next** .	 localhost - PuTTY Install HP-UX wizard: Select a root disk In this step, you will choose the disk which will house your root file system. The disks on your system have been examined and some initial calculations have been done to determine a recommended disk. If you are unsure of what to choose, keep the default selection. You may then safely proceed to the next step. Otherwise, make your choice from the following selector. [Root Disk...] EMULATORHD-IMAGE, 0/0/1/0.0.0, 20480 MB [< Back] [Next >] [Cancel] [Help]
Configure the swap space size. Continue with **Next**.	 localhost - PuTTY Install HP-UX wizard: Specify the amount of root swap space In this step, you will specify the amount of disk space you wish to reserve for root swap. Again, some intial calculations have been done for you. Based on the amount of physical memory in your system and the size of your root disk, a recommended value has been set for you below. If you are unsure of what you need, keep the default selection. You may then safely proceed to the next step. [Root Swap (MB)...] 4096 Physical Memory (RAM) = 2048 MB [< Back] [Next >] [Cancel] [Help]

Select the filesystem type for the root disk. Continue with **Next**.



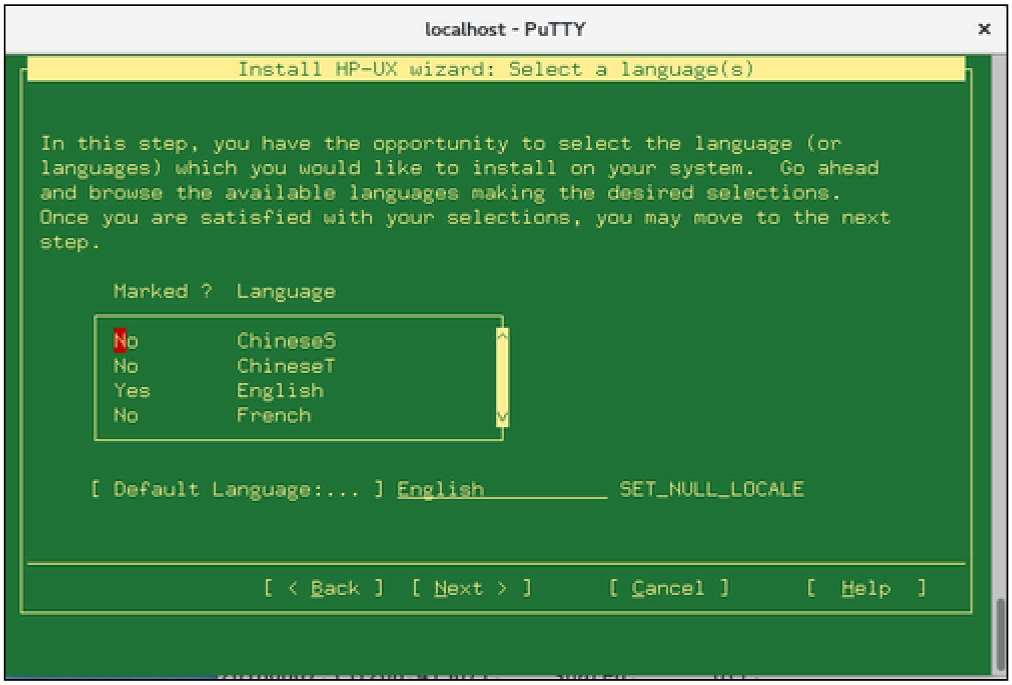
```
                    Install HP-UX wizard: Select a file system type


 Now that you have made your root swap selection, you need to choose
 which type of file system you would like to use.  The current choice
 displayed in the selector is recommended.  If you are unsure of what
 to choose, keep the default selection.  You may then safely proceed to
 the next step.


  File System:     [ Logical Volume Manager (LVM) with VxFS  ->]

            [ < Back ]  [ Next > ]     [ Cancel ]     [  Help  ]
```

If you selected LVM in the step above, configure the number of disks included in the root volume group. Continue with **Next** (use the TAB key to get to the options at the bottom of the screen).



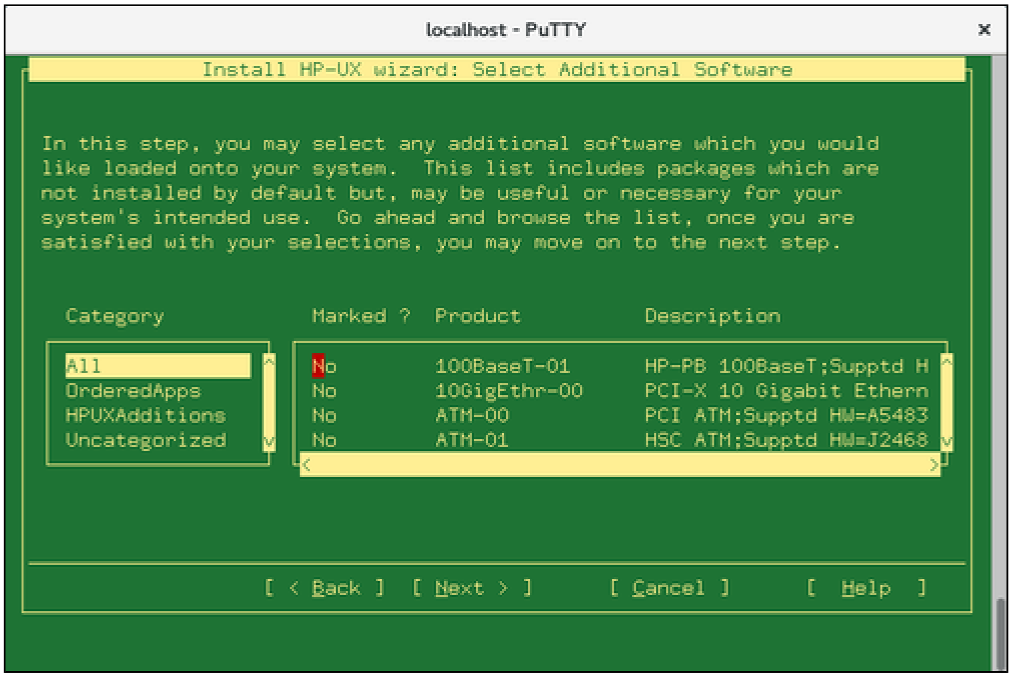
```
              Install HP-UX wizard: Specify root disk/volume group disks

 Please specify how many disks you want placed into the root
 disk/volume group and whether or not you would like the disks to be
 striped.

 If your system has many disks (more than four), it is recommended that
 you put no more than four disks in the root group.  This reduces the
 chances of a disk failure in the root group.

 Striping typically increases overall system performance by spreading
 the file system load across multiple drives.  (Note: The system must
 have more than one disk to enable striping.)

 The number of available disks is: 1


 Number of Disks in Root Group:  1__

            [ < Back ]  [ Next > ]     [ Cancel ]     [  Help  ]
```

Configure your language setting. Continue with **Next**.



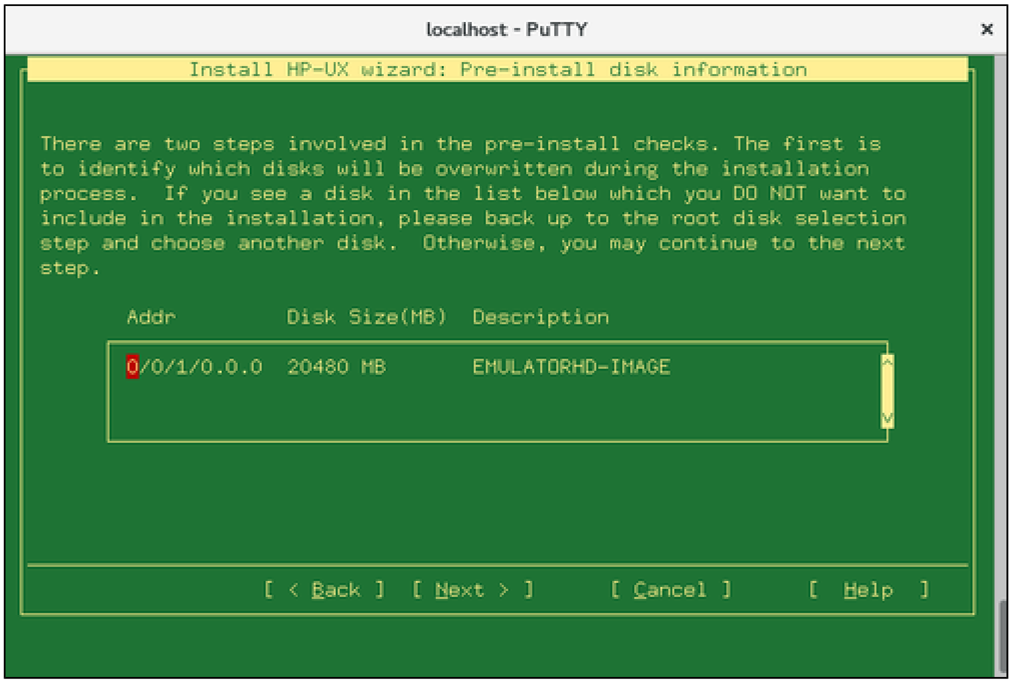
```
localhost - PuTTY

                 Install HP-UX wizard: Select a language(s)

  In this step, you have the opportunity to select the language (or
  languages) which you would like to install on your system.  Go ahead
  and browse the available languages making the desired selections.
  Once you are satisfied with your selections, you may move to the next
  step.

        Marked ?  Language

        No        ChineseS
        No        ChineseT
        Yes       English
        No        French

      [ Default Language:... ] English           SET_NULL_LOCALE

                  [ < Back ]  [ Next > ]      [ Cancel ]      [  Help  ]
```

Select additional software to install, if required. Continue with **Next**.



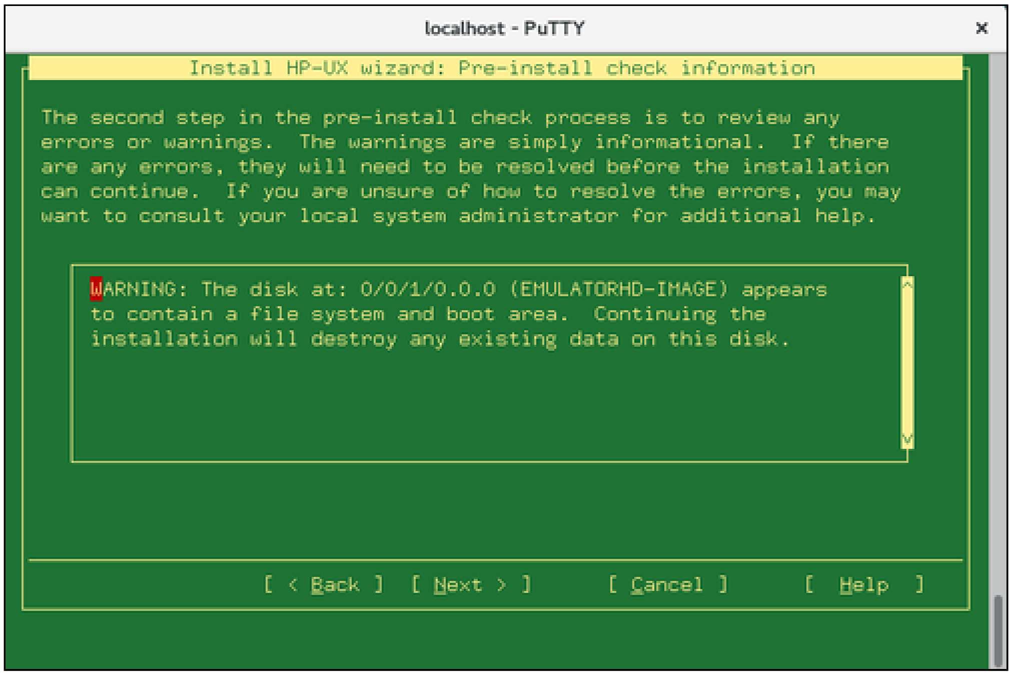
```
localhost - PuTTY

              Install HP-UX wizard: Select Additional Software

  In this step, you may select any additional software which you would
  like loaded onto your system.  This list includes packages which are
  not installed by default but, may be useful or necessary for your
  system's intended use.  Go ahead and browse the list, once you are
  satisfied with your selections, you may move on to the next step.

    Category            Marked ?  Product           Description

    All                 No        100BaseT-01       HP-PB 100BaseT;Supptd H
    OrderedApps         No        10GigEthr-00      PCI-X 10 Gigabit Ethern
    HPUXAdditions       No        ATM-00            PCI ATM;Supptd HW=A5483
    Uncategorized       No        ATM-01            HSC ATM;Supptd HW=J2468

                  [ < Back ]  [ Next > ]      [ Cancel ]      [  Help  ]
```

Confirm the selection of the target disk for installation. It lets you check that the correct disks have been selected to be overwritten. If the correct disk is displayed, continue with **Next**. Otherwise, go back and correct your configuration.



```
                 Install HP-UX wizard: Pre-install disk information

  There are two steps involved in the pre-install checks. The first is
  to identify which disks will be overwritten during the installation
  process.  If you see a disk in the list below which you DO NOT want to
  include in the installation, please back up to the root disk selection
  step and choose another disk.  Otherwise, you may continue to the next
  step.

         Addr          Disk Size(MB)  Description

         0/0/1/0.0.0   20480 MB       EMULATORHD-IMAGE

                  [ < Back ]  [ Next > ]     [ Cancel ]     [  Help  ]
```

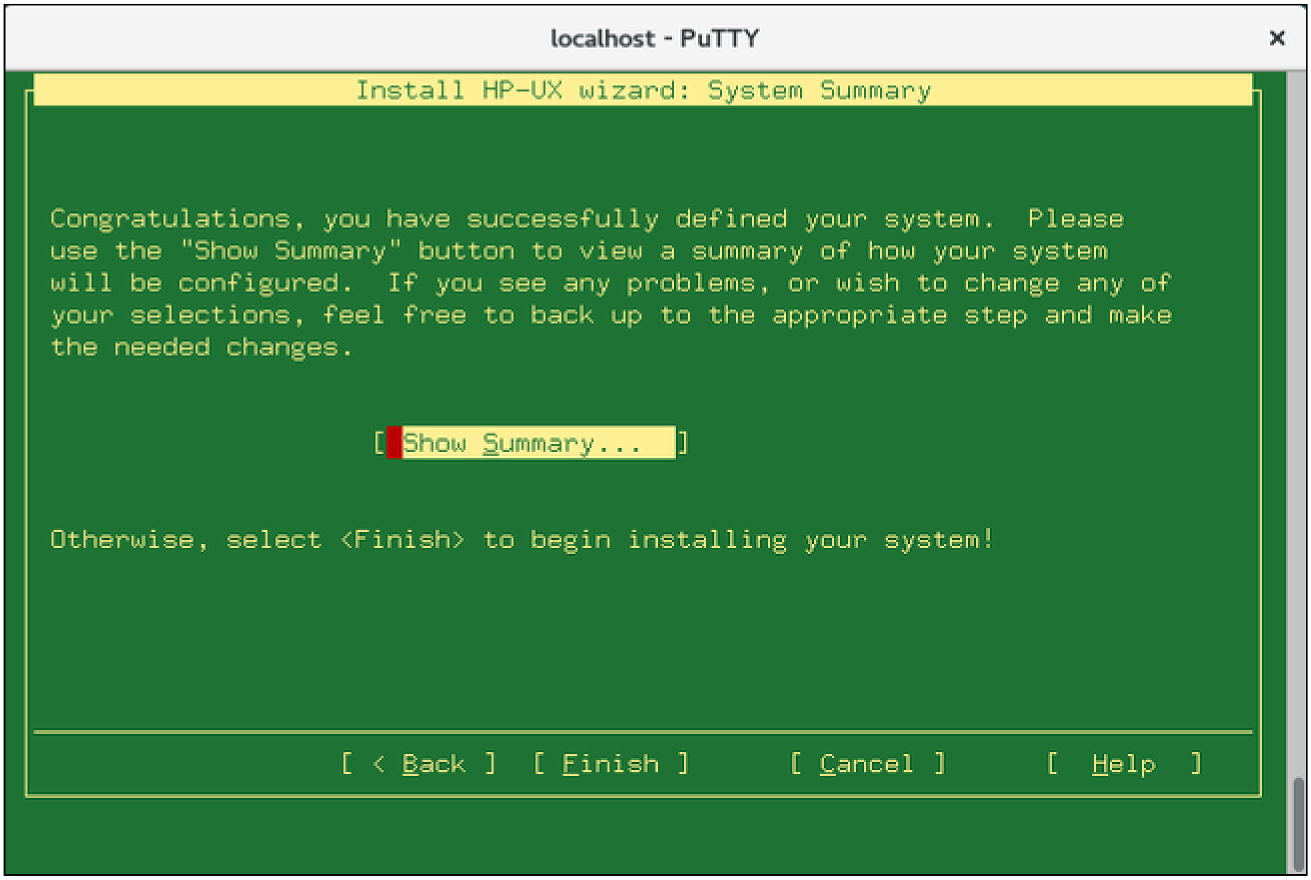
If disks to be overwritten contain an operating system, you may be asked for an explicit confirmation. If it is OK to overwrite the disk, continue with **Next**. Otherwise, go back and correct your configuration.



```
                 Install HP-UX wizard: Pre-install check information

  The second step in the pre-install check process is to review any
  errors or warnings.  The warnings are simply informational.  If there
  are any errors, they will need to be resolved before the installation
  can continue.  If you are unsure of how to resolve the errors, you may
  want to consult your local system administrator for additional help.

      WARNING: The disk at: 0/0/1/0.0.0 (EMULATORHD-IMAGE) appears
      to contain a file system and boot area.  Continuing the
      installation will destroy any existing data on this disk.

                  [ < Back ]  [ Next > ]     [ Cancel ]     [  Help  ]
```

Eventually, you will be offered to display a summary of your selection and to start the installation.

The following image shows the last screen before the installation is started:

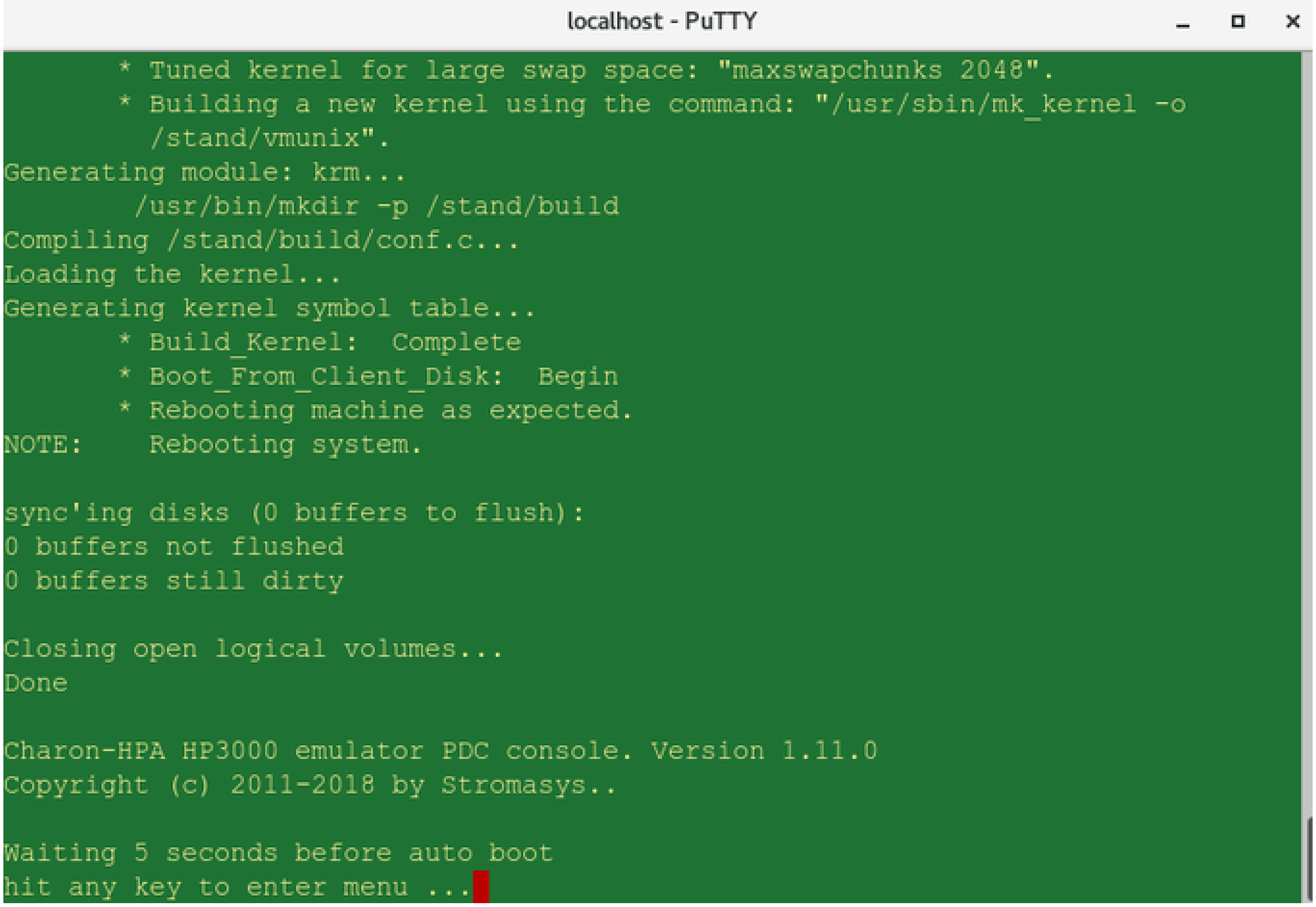


```
                     Install HP-UX wizard: System Summary


 Congratulations, you have successfully defined your system.  Please
 use the "Show Summary" button to view a summary of how your system
 will be configured.  If you see any problems, or wish to change any of
 your selections, feel free to back up to the appropriate step and make
 the needed changes.


                     [ Show Summary... ]


 Otherwise, select <Finish> to begin installing your system!




                   [ < Back ]  [ Finish ]      [ Cancel ]      [  Help  ]
```

Once you select **Finish**, the HP-UX installation will start.

HP-UX Installation Phase 2: Kernel Build and Reboot

Once the initial operating system installation has finished, the system will build a new kernel and reboot as shown below:

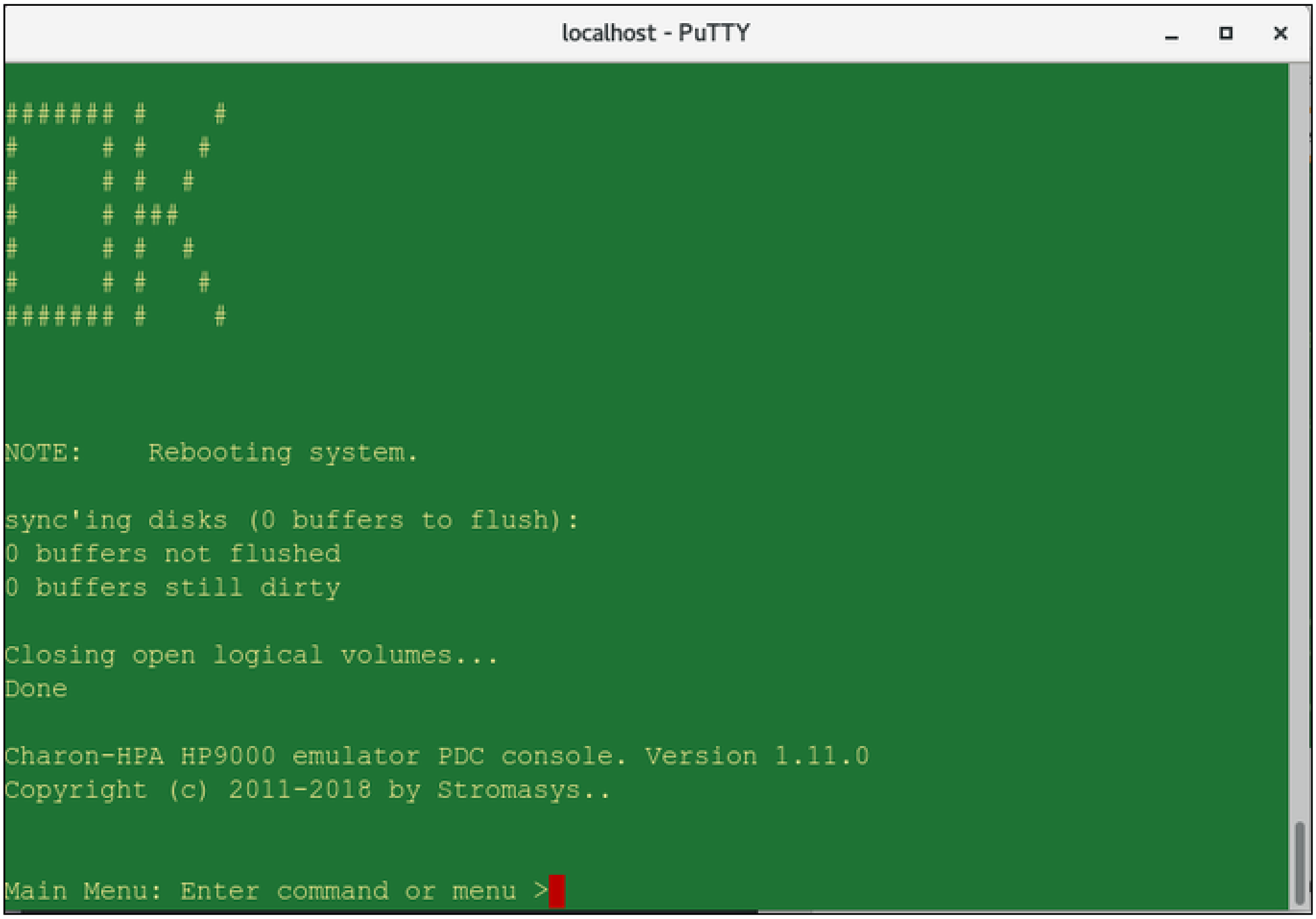


```
       * Tuned kernel for large swap space: "maxswapchunks 2048".
       * Building a new kernel using the command: "/usr/sbin/mk_kernel -o
         /stand/vmunix".
Generating module: krm...
        /usr/bin/mkdir -p /stand/build
Compiling /stand/build/conf.c...
Loading the kernel...
Generating kernel symbol table...
       * Build_Kernel:  Complete
       * Boot_From_Client_Disk:  Begin
       * Rebooting machine as expected.
NOTE:    Rebooting system.

sync'ing disks (0 buffers to flush):
0 buffers not flushed
0 buffers still dirty

Closing open logical volumes...
Done

Charon-HPA HP3000 emulator PDC console. Version 1.11.0
Copyright (c) 2011-2018 by Stromasys..

Waiting 5 seconds before auto boot
hit any key to enter menu ...
```

The system will restart and and run the fileset and software configuration tasks. During this run, it will display many messages starting with the string **NOTE:** to show progress and describe the actions taken. At the end of this configuration run, the system will again shut down and reboot to the boot prompt again.

HP-UX Installation Phase 3: Setup Dialogs and Login

At the screen shown below, enter the command **boot 0/0/1/0.0.0** at the prompt **Main Menu: Enter command or menu >** to boot from the new system disk:



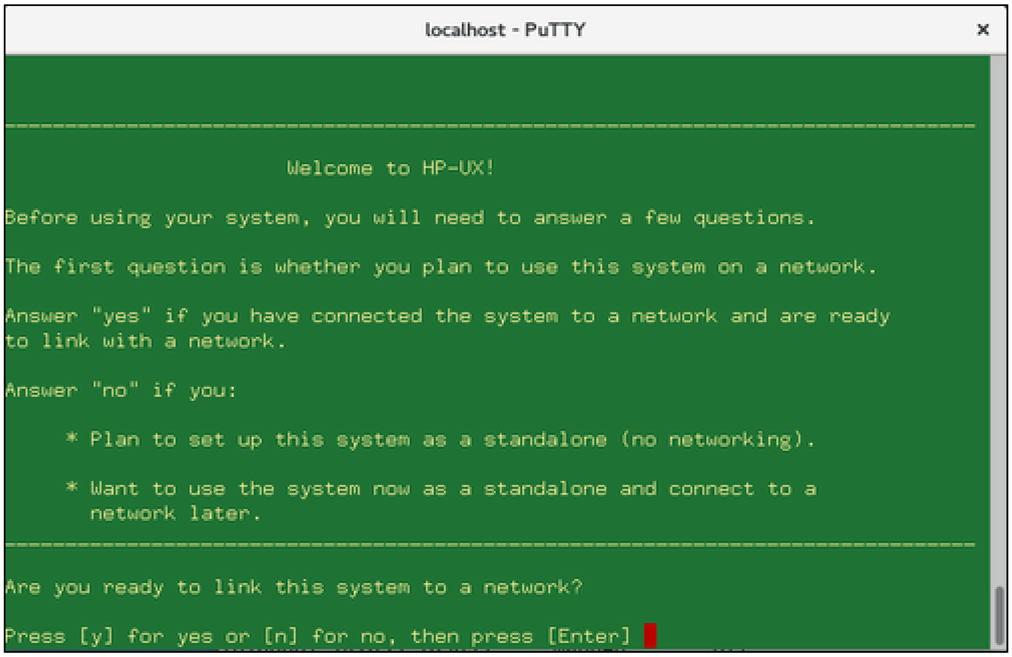
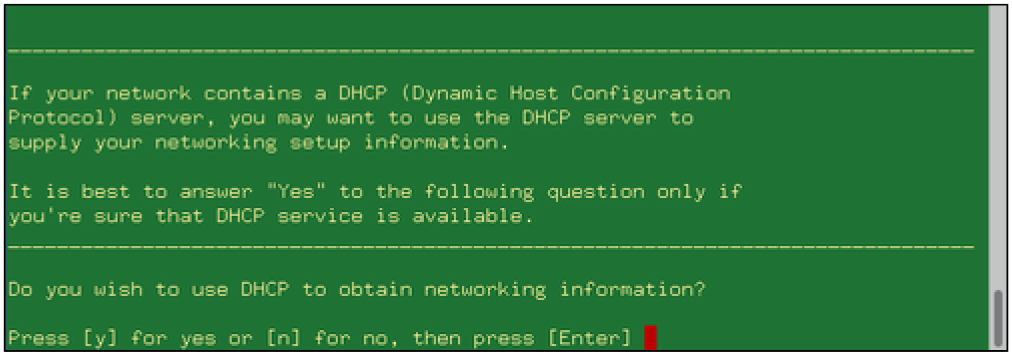
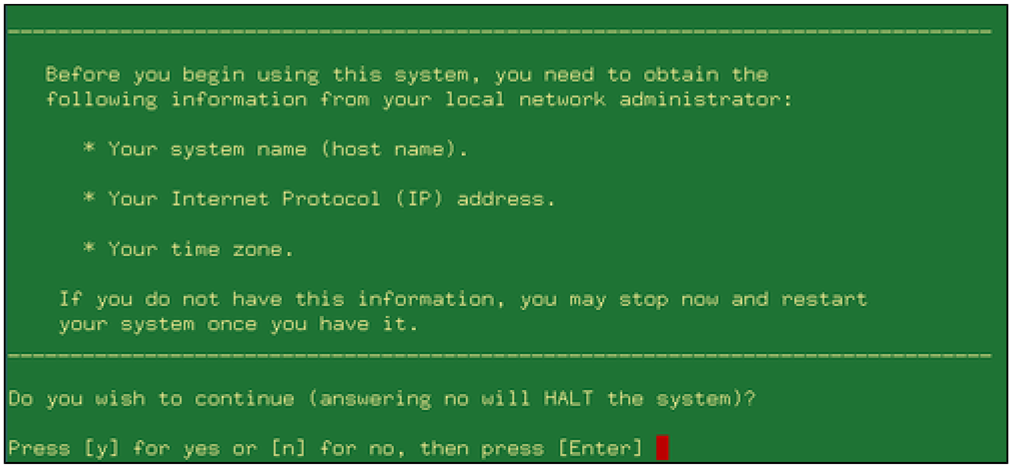
```
####### #    #
#     # #   #
#     # #  #
#     # ###
#     # #  #
#     # #   #
####### #    #



NOTE:    Rebooting system.

sync'ing disks (0 buffers to flush):
0 buffers not flushed
0 buffers still dirty

Closing open logical volumes...
Done

Charon-HPA HP9000 emulator PDC console. Version 1.11.0
Copyright (c) 2011-2018 by Stromasys..


Main Menu: Enter command or menu >
```

Respond with **N** to the **Interact with IPL** question.

This will boot the system and start a series of setup dialogs.

The following table provides an overview of the setup dialogs.

Please note: Depending on your setup, the sequence, number, and content of the screens will vary. **Confirmation dialogs (to confirm the data entered) are not shown**.

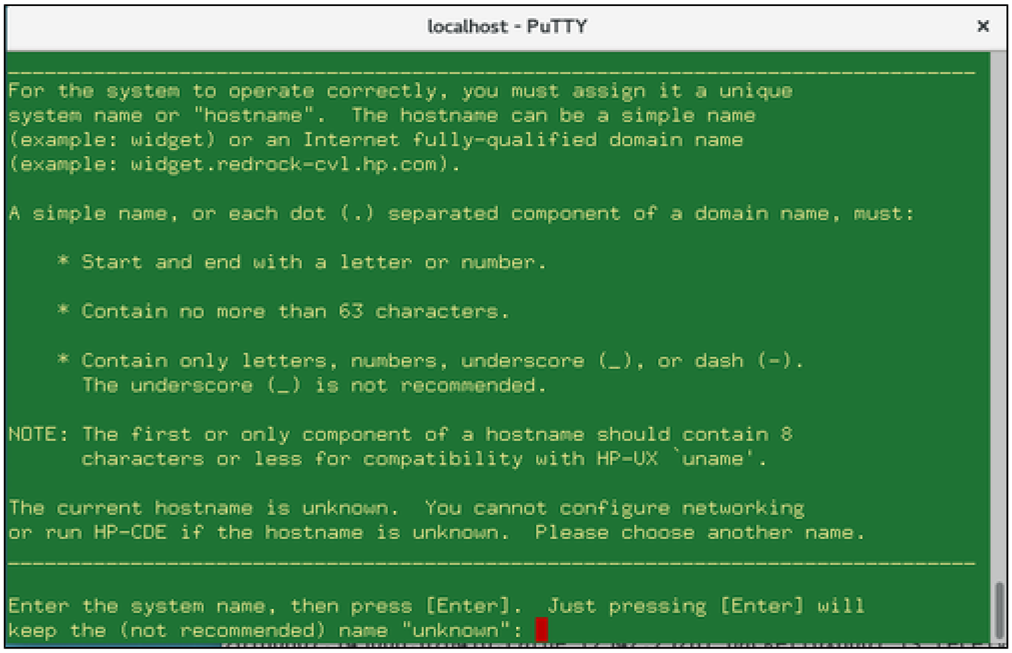
Configuration step	Illustration
Specify whether the system will be connected to a network. Answer **y** to configure networking.	
Specify whether a DHCP server is to be used. Please answer **n** to this question. As DHCP is not recommended for Charon-PAR.	
The next screen is to confirm that the user has the required configuration information at hand. Answering no will halt the system.	

```
-------------------------------------------------------------------------------

                       Welcome to HP-UX!

Before using your system, you will need to answer a few questions.

The first question is whether you plan to use this system on a network.

Answer "yes" if you have connected the system to a network and are ready
to link with a network.

Answer "no" if you:

     * Plan to set up this system as a standalone (no networking).

     * Want to use the system now as a standalone and connect to a
       network later.
-------------------------------------------------------------------------------

Are you ready to link this system to a network?

Press [y] for yes or [n] for no, then press [Enter]
```

```
-------------------------------------------------------------------------------

If your network contains a DHCP (Dynamic Host Configuration
Protocol) server, you may want to use the DHCP server to
supply your networking setup information.

It is best to answer "Yes" to the following question only if
you're sure that DHCP service is available.
-------------------------------------------------------------------------------

Do you wish to use DHCP to obtain networking information?

Press [y] for yes or [n] for no, then press [Enter]
```

```
-------------------------------------------------------------------------------

   Before you begin using this system, you need to obtain the
   following information from your local network administrator:

      * Your system name (host name).

      * Your Internet Protocol (IP) address.

      * Your time zone.

    If you do not have this information, you may stop now and restart
    your system once you have it.
-------------------------------------------------------------------------------

Do you wish to continue (answering no will HALT the system)?

Press [y] for yes or [n] for no, then press [Enter]
```

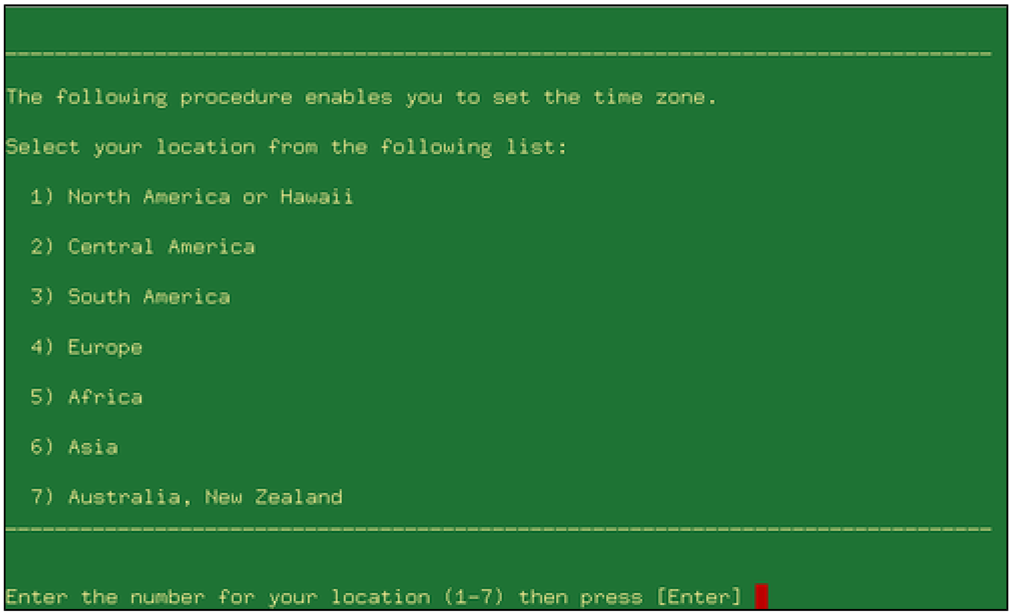
Hostname configuration. Enter the hostname of the emulated system. Continue with **Enter**.



```
--------------------------------------------------------------------------------
For the system to operate correctly, you must assign it a unique
system name or "hostname".  The hostname can be a simple name
(example: widget) or an Internet fully-qualified domain name
(example: widget.redrock-cvl.hp.com).

A simple name, or each dot (.) separated component of a domain name, must:

    * Start and end with a letter or number.

    * Contain no more than 63 characters.

    * Contain only letters, numbers, underscore (_), or dash (-).
      The underscore (_) is not recommended.

NOTE: The first or only component of a hostname should contain 8
      characters or less for compatibility with HP-UX `uname'.

The current hostname is unknown.  You cannot configure networking
or run HP-CDE if the hostname is unknown.  Please choose another name.
--------------------------------------------------------------------------------

Enter the system name, then press [Enter].  Just pressing [Enter] will
keep the (not recommended) name "unknown":
```

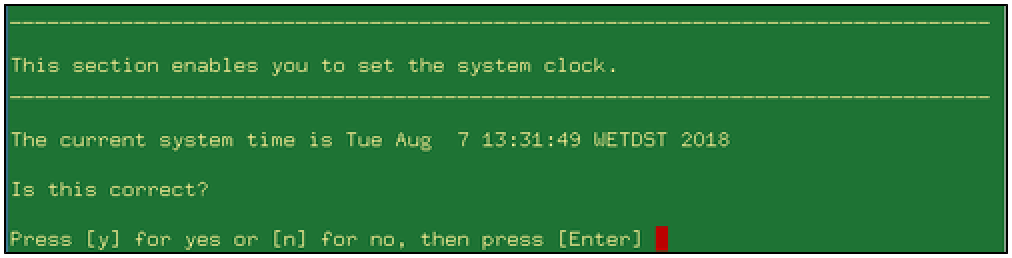
The **next two screens** allow you to configure the timezone for your system. The first screen (*displayed here*) selects the general region, the second (*not displayed*), the exact timezone within the region.

Make the appropriate selection and continue with **Enter**.



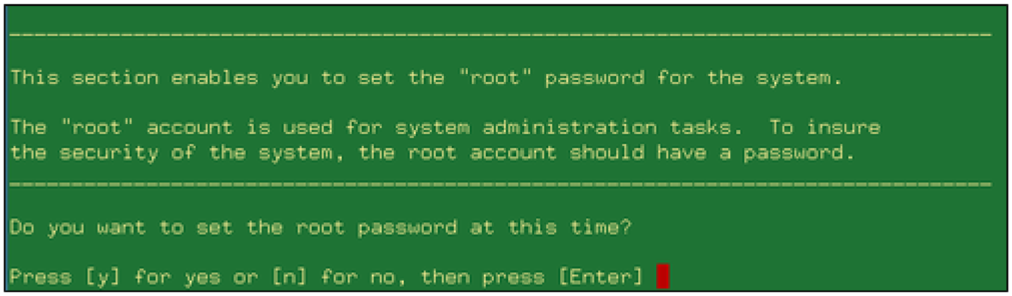
```
--------------------------------------------------------------------------------

The following procedure enables you to set the time zone.

Select your location from the following list:

  1) North America or Hawaii

  2) Central America

  3) South America

  4) Europe

  5) Africa

  6) Asia

  7) Australia, New Zealand
--------------------------------------------------------------------------------


Enter the number for your location (1-7) then press [Enter]
```

Confirm/correct the system time.



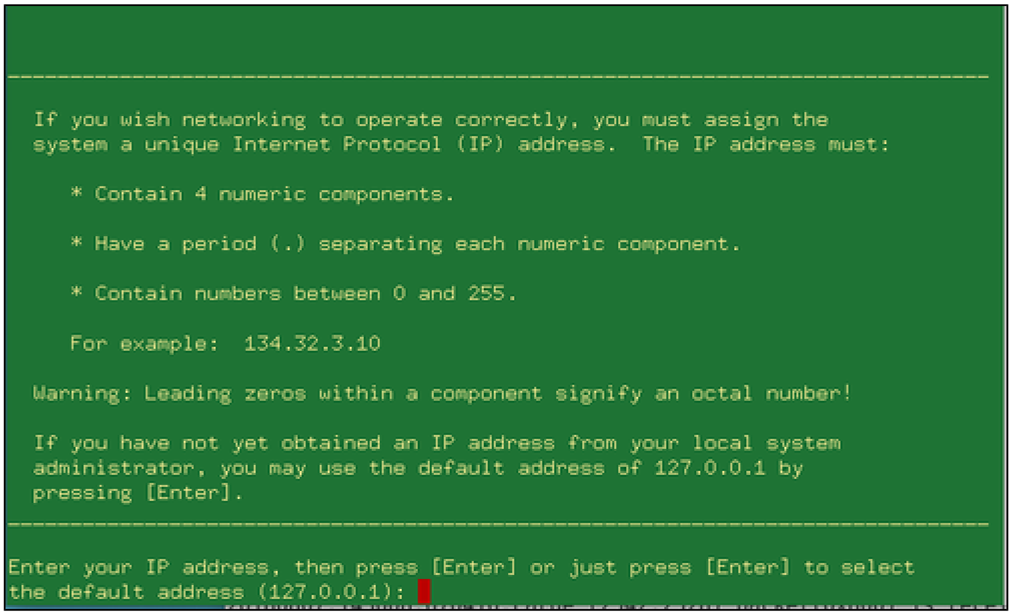
```
--------------------------------------------------------------------------------

This section enables you to set the system clock.
--------------------------------------------------------------------------------

The current system time is Tue Aug  7 13:31:49 WETDST 2018

Is this correct?

Press [y] for yes or [n] for no, then press [Enter]
```

Set the root password. Confirm that you want to set the password. Then the system will prompt you for the password.



```
--------------------------------------------------------------------------------

This section enables you to set the "root" password for the system.

The "root" account is used for system administration tasks.  To insure
the security of the system, the root account should have a password.
--------------------------------------------------------------------------------

Do you want to set the root password at this time?

Press [y] for yes or [n] for no, then press [Enter]
```

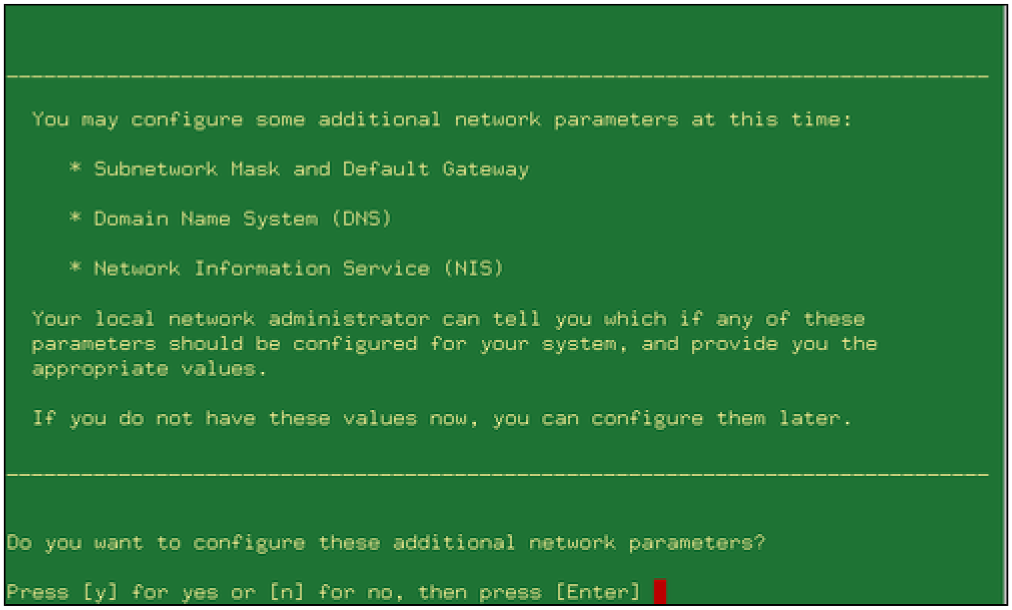
Configure the IP address and continue with **Enter**.



```
--------------------------------------------------------------------------------

  If you wish networking to operate correctly, you must assign the
  system a unique Internet Protocol (IP) address.  The IP address must:

     * Contain 4 numeric components.

     * Have a period (.) separating each numeric component.

     * Contain numbers between 0 and 255.

     For example:  134.32.3.10

  Warning: Leading zeros within a component signify an octal number!

  If you have not yet obtained an IP address from your local system
  administrator, you may use the default address of 127.0.0.1 by
  pressing [Enter].
--------------------------------------------------------------------------------

Enter your IP address, then press [Enter] or just press [Enter] to select
the default address (127.0.0.1):
```

Additional network settings overview. Confirm with **y** if you want to configure one or more of these network settings.

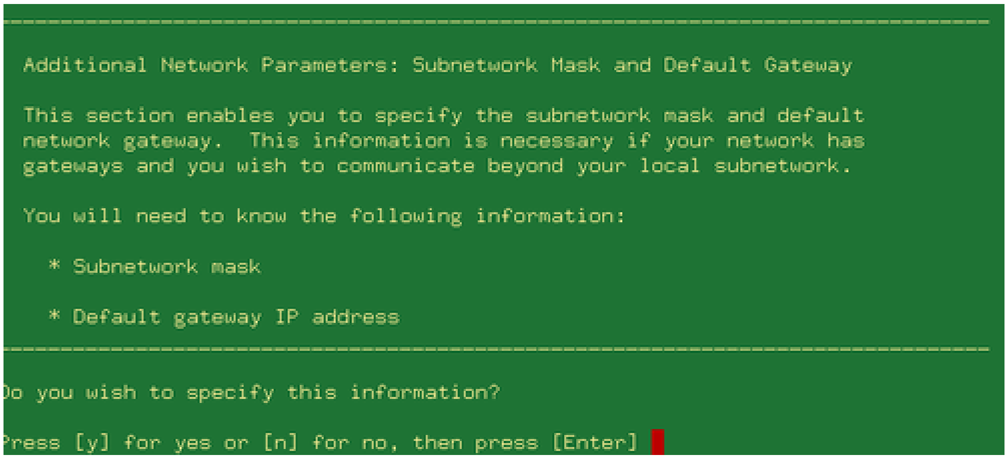
For cloud environments, please note: in case of a dedicated Ethernet interface, the IP address of the emulator must be set to the value assigned by the cloud provider to the corresponding host NIC.



```
--------------------------------------------------------------------------------

  You may configure some additional network parameters at this time:

     * Subnetwork Mask and Default Gateway

     * Domain Name System (DNS)

     * Network Information Service (NIS)

  Your local network administrator can tell you which if any of these
  parameters should be configured for your system, and provide you the
  appropriate values.

  If you do not have these values now, you can configure them later.

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


Do you want to configure these additional network parameters?

Press [y] for yes or [n] for no, then press [Enter]
```

If you answered **yes** in the step above, you will be asked to configure subnet mask and default gateway.

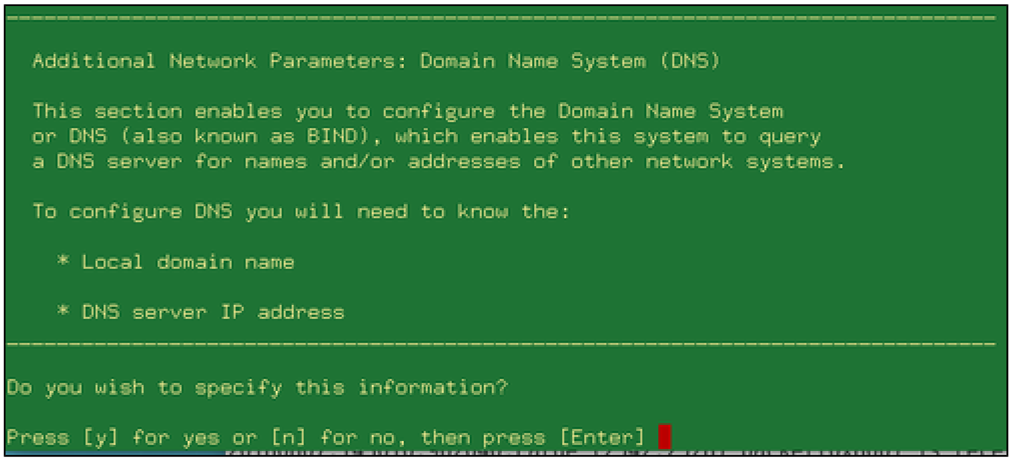
If you answer **yes** to this question, the system will display additional screens (*not displayed here*) guiding you through the configuration steps.



```
--------------------------------------------------------------------------------

  Additional Network Parameters: Subnetwork Mask and Default Gateway

  This section enables you to specify the subnetwork mask and default
  network gateway.  This information is necessary if your network has
  gateways and you wish to communicate beyond your local subnetwork.

  You will need to know the following information:

    * Subnetwork mask

    * Default gateway IP address
--------------------------------------------------------------------------------

Do you wish to specify this information?

Press [y] for yes or [n] for no, then press [Enter]
```

If you answered **yes** when asked whether you wanted to configure additional network options, the option to configure DNS will be offered.

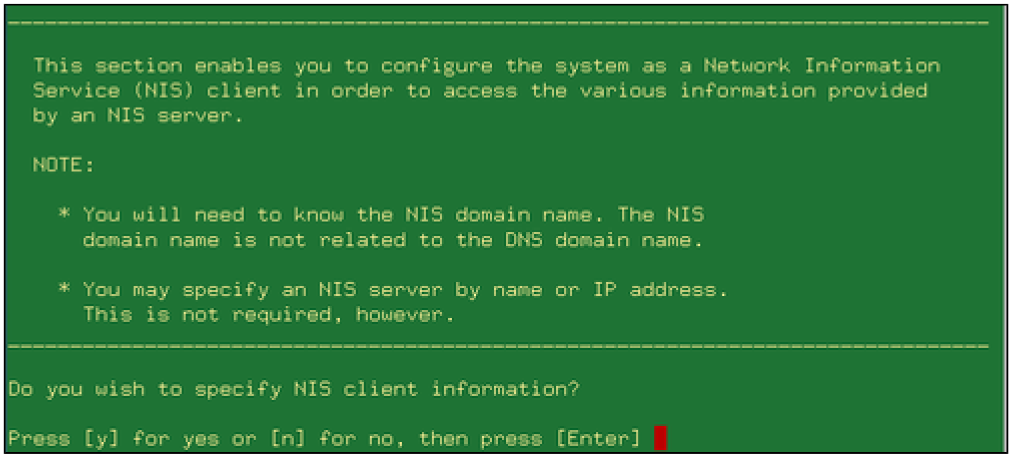
If you answer **yes** to this question, the system will display additional screens (*not displayed here*) guiding you through the configuration steps.



```
-------------------------------------------------------------------------------

  Additional Network Parameters: Domain Name System (DNS)

  This section enables you to configure the Domain Name System
  or DNS (also known as BIND), which enables this system to query
  a DNS server for names and/or addresses of other network systems.

  To configure DNS you will need to know the:

    * Local domain name

    * DNS server IP address

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

Do you wish to specify this information?

Press [y] for yes or [n] for no, then press [Enter]
```

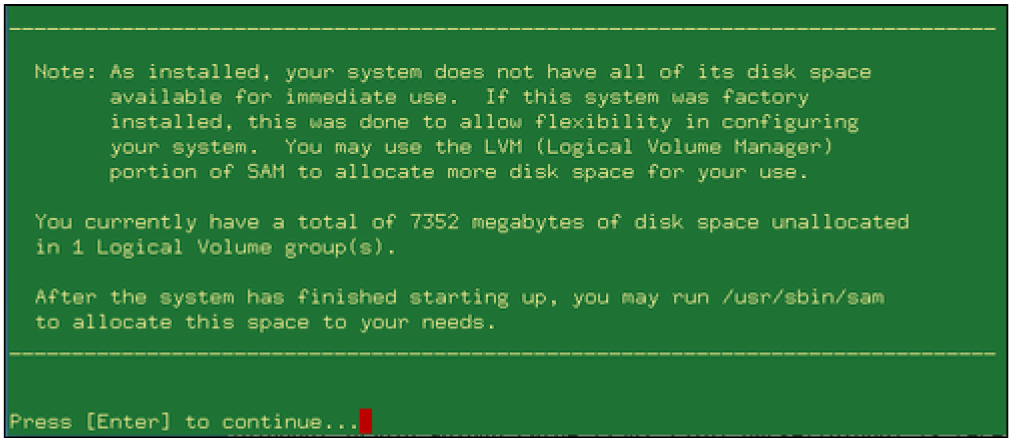
If you answered **yes** when asked whether you wanted to configure additional network options, the option to configure NIS will be offered.

If you answer **yes** to this question, the system will display additional screens (*not displayed here*) guiding you through the configuration steps.

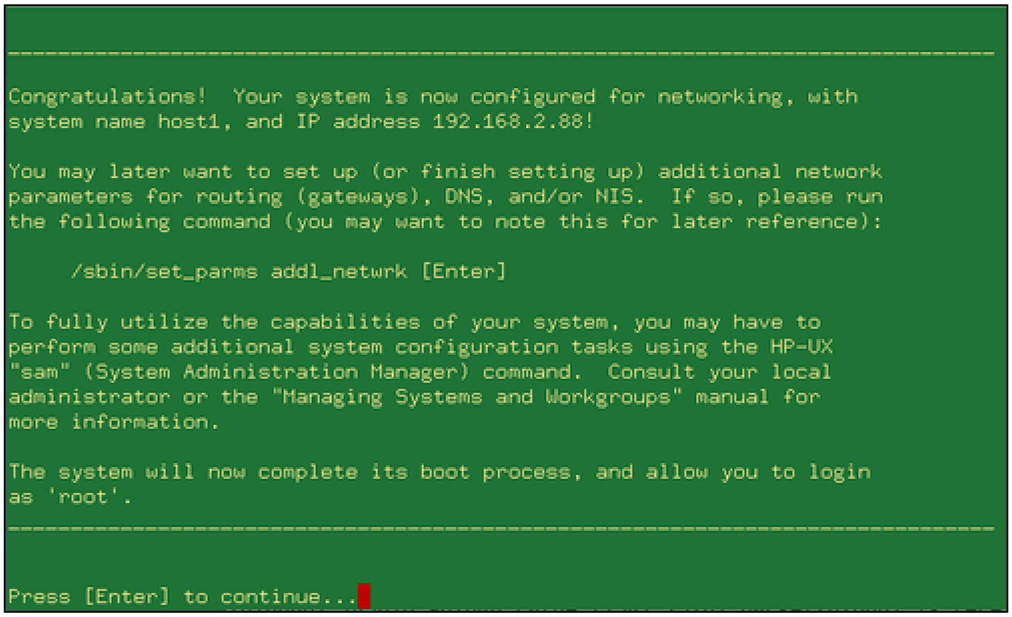


```
-------------------------------------------------------------------------------

  This section enables you to configure the system as a Network Information
  Service (NIS) client in order to access the various information provided
  by an NIS server.

  NOTE:

    * You will need to know the NIS domain name. The NIS
      domain name is not related to the DNS domain name.

    * You may specify an NIS server by name or IP address.
      This is not required, however.

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

Do you wish to specify NIS client information?

Press [y] for yes or [n] for no, then press [Enter]
```

An additional screen will inform you about currently unassigned disk space and how to make it usable later.

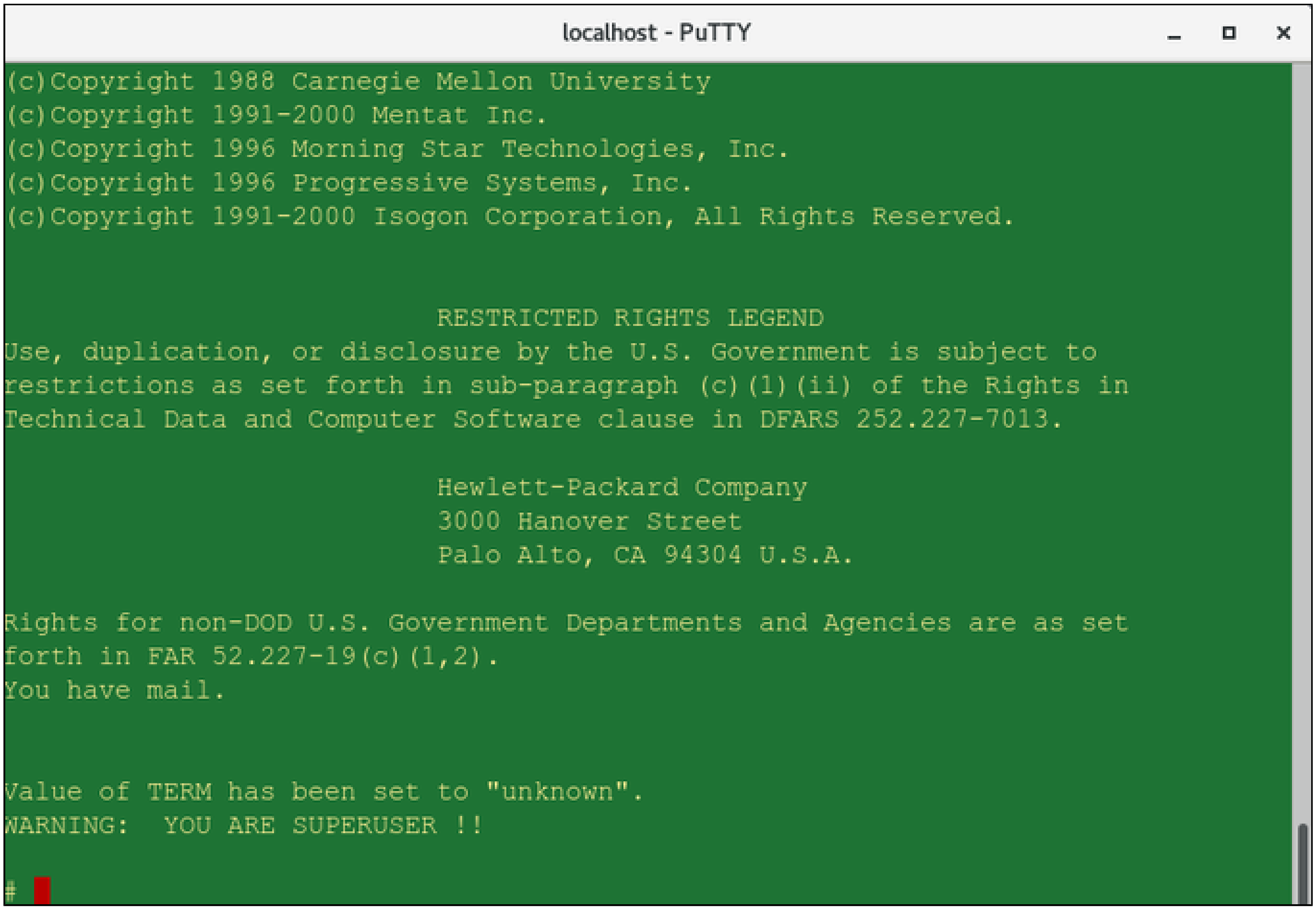


```
-------------------------------------------------------------------------------

  Note: As installed, your system does not have all of its disk space
        available for immediate use.  If this system was factory
        installed, this was done to allow flexibility in configuring
        your system.  You may use the LVM (Logical Volume Manager)
        portion of SAM to allocate more disk space for your use.

  You currently have a total of 7352 megabytes of disk space unallocated
  in 1 Logical Volume group(s).

  After the system has finished starting up, you may run /usr/sbin/sam
  to allocate this space to your needs.

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

Press [Enter] to continue...
```

Confirmation that system configuration is complete. Upon pressing **Enter**, the system will continue to boot to the the login prompt.	

```
-------------------------------------------------------------------------------

Congratulations!  Your system is now configured for networking, with
system name host1, and IP address 192.168.2.88!

You may later want to set up (or finish setting up) additional network
parameters for routing (gateways), DNS, and/or NIS.  If so, please run
the following command (you may want to note this for later reference):

     /sbin/set_parms addl_netwrk [Enter]

To fully utilize the capabilities of your system, you may have to
perform some additional system configuration tasks using the HP-UX
"sam" (System Administration Manager) command.  Consult your local
administrator or the "Managing Systems and Workgroups" manual for
more information.

The system will now complete its boot process, and allow you to login
as 'root'.
-------------------------------------------------------------------------------


Press [Enter] to continue...
```

During the boot process the system will display startup messages for the individual system components and finally the log-in prompt. You can log into the system using the root password configured above:



```
(c)Copyright 1988 Carnegie Mellon University
(c)Copyright 1991-2000 Mentat Inc.
(c)Copyright 1996 Morning Star Technologies, Inc.
(c)Copyright 1996 Progressive Systems, Inc.
(c)Copyright 1991-2000 Isogon Corporation, All Rights Reserved.


                           RESTRICTED RIGHTS LEGEND
Use, duplication, or disclosure by the U.S. Government is subject to
restrictions as set forth in sub-paragraph (c)(1)(ii) of the Rights in
Technical Data and Computer Software clause in DFARS 252.227-7013.

                           Hewlett-Packard Company
                           3000 Hanover Street
                           Palo Alto, CA 94304 U.S.A.

Rights for non-DOD U.S. Government Departments and Agencies are as set
forth in FAR 52.227-19(c)(1,2).
You have mail.


Value of TERM has been set to "unknown".
WARNING:  YOU ARE SUPERUSER !!

#
```

Stopping the Emulator

In order to stop the emulator in an orderly manner, perform the following steps:

1. Shut down the guest operating system:

    ```
    # shutdown -h now
    ```

    Starting with Charon-PAR version 3.0.5 this will, by default, automatically stop the emulator when it detects that the operating system has been halted. See the system.stop_on_halt parameter in the Configuration File Reference. If the emulator is not stopped automatically, continue with step 2 below.

2. Starting with version 3.0.5, the emulator is stopped automatically after the guest operating system shutdown is complete (unless the configuration parameter `system.stop_on_halt` has been set to **`false`**). If the emulator is not stopped automatically after the shutdown of the guest operating system, different methods must be used to stop it in different situations:

- **The emulator was started in interactive mode and the Charon-PAR console is available in current terminal**: enter the **exit** command at the **pa9-64>** prompt and hit **Enter**. This will cause the emulator to close the disk images of the emulated system and to exit.
- **The Charon-PAR console is available on a local TCP port (emulator started with -c or --console-port parameter)**: this situation will exist most frequently if the emulator has been started as a daemon or service. To stop the emulator access the Charon-PAR console via **telnet** to *localhost* on the port specified when starting the emulator, enter the **exit** command at the **pa9-64>** prompt, and hit Enter.
  Access to the Charon-PAR console via a TCP port is available starting with Charon-PAR version 3.0.1. The **CTRL+B** key combination has been disabled on the emulator console starting with this version.

- **The emulator was started as a daemon or service and the Charon-PAR console is not available via a local TCP port**: This situation can occur if you run a version before 3.0.1 or you started the emulator without the **`-c`** option.
  - For versions 3.0.1 and higher: the emulator must be stopped using the **kill** command from the root user on the Charon host system. Perform the following steps to stop the Charon-PAR process:
    - Find the PID (process id) of the process: **`# ps -ef | grep -i charon-par`**
    - Stop the process: **`# kill`** *`<charon-par-pid>`*
  - For versions before 3.0.1: on the emulator console, after shutting down the guest operating system, press **CTRL+B**. After a few seconds, a traceback may be shown. At the end of this output, you will be asked to select a dump type. Select **N** for no dump. The PDC console prompt (**Main menu:**) will appear. Type **exit** to stop the emulator.

For more information about the Charon-PAR console, please refer to Charon-PAR Console. For more information about the Charon-PAR command-line options, please refer to Charon-PAR Command-Line Options.

Charon-PAR/PA9-32 Guest - Basic Configuration and Installation

Contents

Please note: All Linux and HP-UX sample commands in this section are performed as the root user (prompt = #). This may not be necessary for all commands in all environments depending on the local permission settings. The commands can be executed on the Charon host graphical console or via a non-graphical login (e.g., SSH). However, for accessing the running emulator, the use of PuTTY is strongly recommended because it works best with the emulator environment.

The Charon-PAR/PA9-32 functionality of Charon-PAR emulates historic 32-bit PA-RISC hardware. This section shows a sample installation of a 32-bit guest system.

Emulator Licensing Considerations

The emulator needs a valid license to run. The license can be a Sentinel HASP license or a Virtual Environment (VE) license. Please refer to Initial Emulator License Configuration for the basic license configuration steps of both options. The license(s) to be used are defined in the configuration file. If running in a cloud environment, a VE license is strongly recommended (supported from Charon-PAR version 3.0.6).

HP-UX ISO Installation Medium

To install a basic HP-UX system running as a guest system in an emulator instance, you need to provide an HP-UX ISO installation medium.

Charon-PAR/PA9-32 for the model 720 currently supports HP-UX 9.05, 9.07, 10.20, and 11.00. The installation procedure for versions 10.20 and 11.00 is very similar to the one for Charon-PAR/PA9-64. However, the procedure for the older HP-UX versions can be quite different. Hence, 9.07 is used for this example. For this version of HP-UX, **two** ISO files are required:

- HP-UX 9.07 **install** ISO (mini-system installation)
- HP-UX 9.07 **Core_OS** ISO (operating system filesets)

Copy the ISO files to the directory you prepared for the disk containers of your emulated instance.

Creating a Virtual Disk Container for the System Disk

The emulated system requires a disk on which the HP-UX system will be installed. Create a system disk as shown in the following example:

	Step	Command
1	Change to the directory where you store your virtual disks.	`# cd /data/Stromasys/host1/Data/Disk/`

2	Create an empty disk container. Note that the actual size of the disk will depend on the requirements of your emulated system. The disk name shown is just an example.
	Please note: HP-UX 9.07 allows a maximum root disk size of 2GB. The number of blocks should be a multiple of 512.
		With **dd:**
		`# dd if=/dev/zero of=ldev1.dsk bs=1K count=2091520`
		With **fallocate** (very fast alternative):
		`# fallocate --length 2147483648 ldev1.dsk`
		Please note: the size of the disk must be such that it is aligned to a 512-byte boundary. You can check this with the following bash expression:
		`$ bytes=<number-of-bytes>; echo $(( bytes - bytes/512*512 ))`
		If it returns 0, the alignment is correct.

Adapting the Configuration File Template

The configuration file template must be adapted to the customer environment and the required guest system characteristics. This section describes the minimum number of changes required to configure a Charon-PAR/PA9-32 emulator instance. Please refer to Configuration File Reference for a detailed description of the configuration file options.

Perform the following steps to create a basic configuration file for a Charon-PAR/PA9-32 emulated system:

	Step	Description

1	Create a copy of the configuration file template.	Example:
		`# cd /data/Stromasys/host1/`
		`# cp /opt/charon/cfg/pa9-32.cfg host1.cfg`

2	Open the configuration file in a text editor.	`# vi host1.cfg`
		or
		`# gedit host1.cfg`

3	Configure the system model.	**model "720"** Configured model must be covered by your license.
4	Configure the RAM for the emulated system (in MB).	**memory 64**
5	Configure the serial console for telnet via PuTTY.	Use the method 2 of the serial line section in the configuration file and make sure the definition of the asp.uart0.device is: `asp.uart0.device.command="putty -load PAR-Telnet-VT100"`
6	Configure disk 1 (system disk).	Change definition of DKA0 to: `DKA0.image="/data/Stromasys/host1/Data/Disk/ldev1.dsk"`

7	Add a definition for disk 2 (ISO file) or (if it already exists) modify the definition to point to your ISO installation file for the mini-system (the **install** disk).	`load DKA100`
		`DKA100.image="/data/Stromasys/host1/Data/Disk/HP-UX_9.07_Install_S700.iso"`

8	Configure a network card based on a physical host NIC. This NIC is dedicated to the guest operating system and cannot be used for network communication by the host system. To share an interface, a virtual bridge would have to be used (see Ethernet configuration sections in this document).	In the configuration file template, comment out the dummy interface and the tap interface configuration. Then uncomment the interface configuration of the physical interface example. Change the definition of EWA0 to the interface that is assigned to the emulated system (example: eth1) and disable the offload parameters for the interface:
		`system.lan0.card.mapping_mode="RAW"`
		`system.lan0.card.iface="eth1"`
		`system.lan0.card.initialize_command="ethtool -K $IFACE rx off ; ethtool -K $IFACE tx off ; ethtool -K $IFACE sg off ; ethtool -K $IFACE gso off ; ethtool -K $IFACE gro off ; ethtool -K $IFACE txvlan off ; ethtool -K $IFACE rxvlan off"`
		Please note: Depending on the capabilities of the Ethernet device, not all offload parameters maybe be available. This will create a warning message. The variable IFACE is set by the system before executing the initialize command.
		For cloud environments, please note:
		• The MAC address on an interface must be the same as the one configured by the cloud provider on the dedicated interface. This should be the case by default for Charon-PAR. The IP address used by the guest OS must be the one configured by the cloud provider.
		• Virtual bridge configurations that are linked to a host NIC are not possible in cloud environments. Only "internal virtual bridge" configurations are possible. Such bridges can be used for host-guest communication. External communication for the guest can be achieved either by NAT or by routing configurations (depending on requirements and cloud-specific restrictions). A less complex, TAP based solution is to use a MACVTAP interface (using MAC and IP address assigned by the cloud provider).

9	Save the configuration file.	

Starting the Emulated System and Installing the Guest System

This section shows how to start the emulator and install a HP-UX system.

Please note:

- The examples in this section assume that a **graphical interface** is used, on which a PuTTY session can be run.
- **Should this not be possible**, you can access the console of the emulated system by using, for example, a telnet client and pointing it to the TCP port on the Charon host defined for the serial console port in the configuration file. The default port is 30000. Sample command to connect to an emulator on the local system:

  `$ telnet localhost 30000`).

  In this case, do not start the PuTTY program automatically from the emulator configuration.

Starting the Emulator

To start the emulated Charon-PAR/PA9-32 system, perform the following steps:

	Step	Command
1	Go to the directory where the configuration file has been stored.	`# cd /data/Stromasys/host1/`
2	Start the emulator instance.	`# /opt/charon/bin/charon-par -f host1.cfg`

If everything works, you will see log output in the current terminal window and eventually a **pa9-32>** prompt. This is the Charon-PAR emulator console. It displays log information, allows to set configuration parameters and to stop the emulator. **Please note**: the prompt may be obscured by log messages and not be immediately visible. In this case, press the **Enter** key once to display the prompt.

At the same time, a **green PuTTY console window** will open.

Please note: Should there be an error from PuTTY about missing fonts, start PuTTY as the root user, load the **PAR-Telnet-VT100** profile, set a font that is available on the host system, save the configuration.

Booting from the Install ISO File

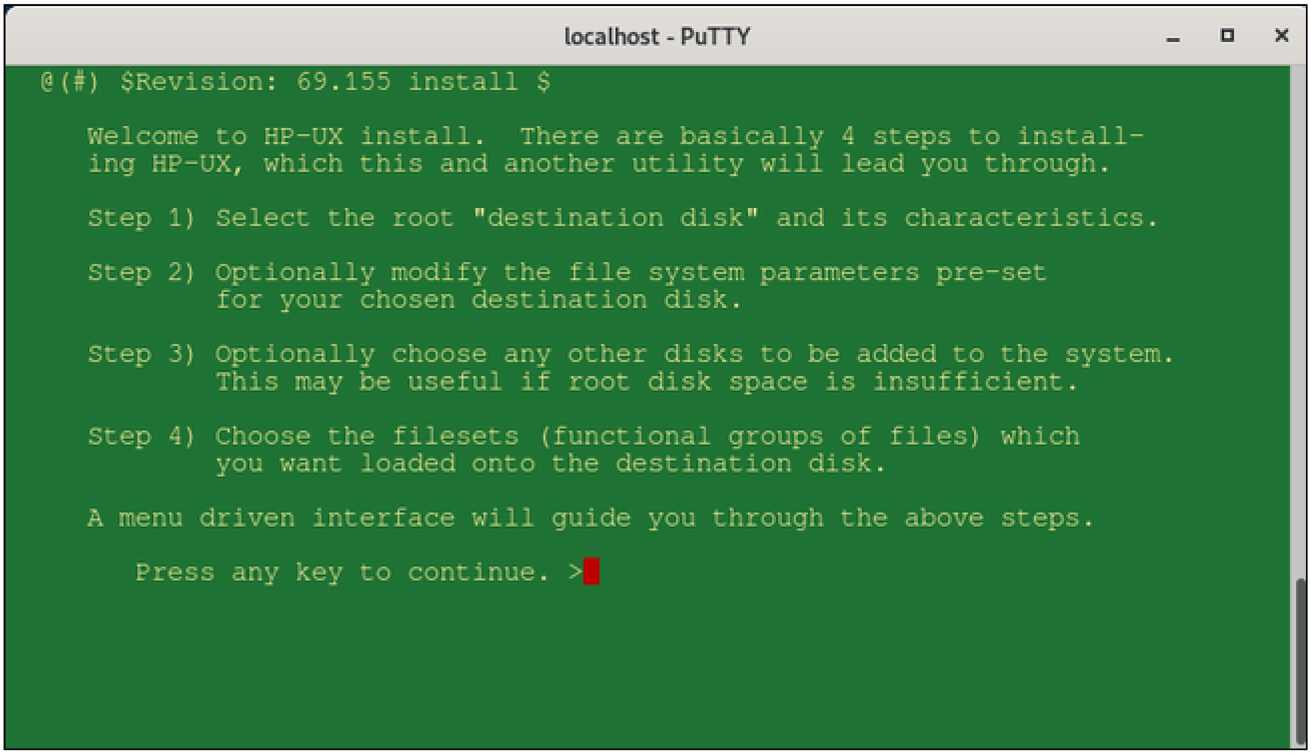
In green PuTTY console window perform the following steps to boot the installation ISO file (containing a very basic system):

- Press the **Enter** key.
- This should produce the prompt **Main Menu: Enter command or menu >**.
- Enter the command **`boot 2/0/1.1.0`** to boot from the ISO and confirm by pressing the **Enter** key.
- At the prompt Interact with IPL (Y or N)?> answer **N**.
- At the next prompt, press the **Enter** key to proceed to the installation overview. Please note that it may take a little while until this prompt is displayed.

HP-UX 9.07 Installation Phase 1: Installing the Basic System

During this phase, a basic system is installed to the root disk that will enable the installation of the rest of the operating system.

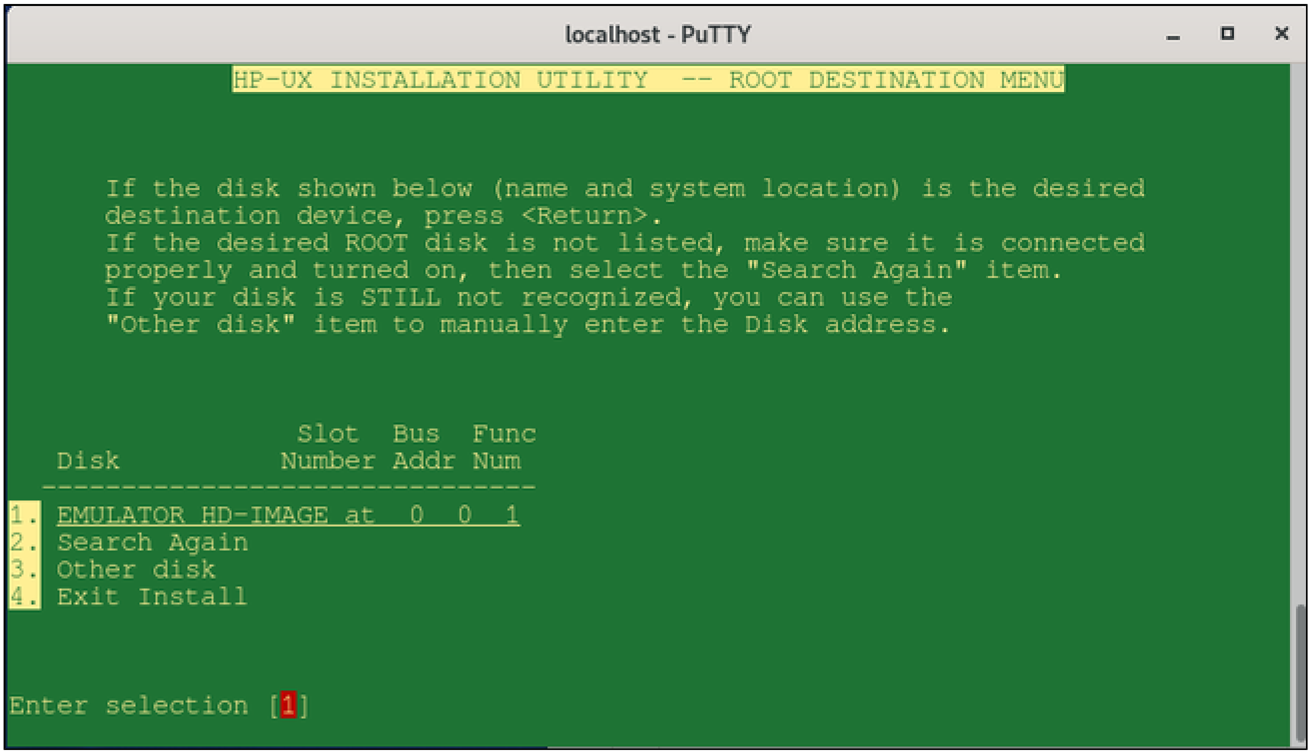
After booting the **install** ISO file as described above, you will eventually see an overview of the installation steps:



```
 @(#) $Revision: 69.155 install $

    Welcome to HP-UX install.  There are basically 4 steps to install-
    ing HP-UX, which this and another utility will lead you through.

    Step 1) Select the root "destination disk" and its characteristics.

    Step 2) Optionally modify the file system parameters pre-set
            for your chosen destination disk.

    Step 3) Optionally choose any other disks to be added to the system.
            This may be useful if root disk space is insufficient.

    Step 4) Choose the filesets (functional groups of files) which
            you want loaded onto the destination disk.

    A menu driven interface will guide you through the above steps.

       Press any key to continue. >
```

After you've read the information, press any key to continue with the next step: the root disk selection.

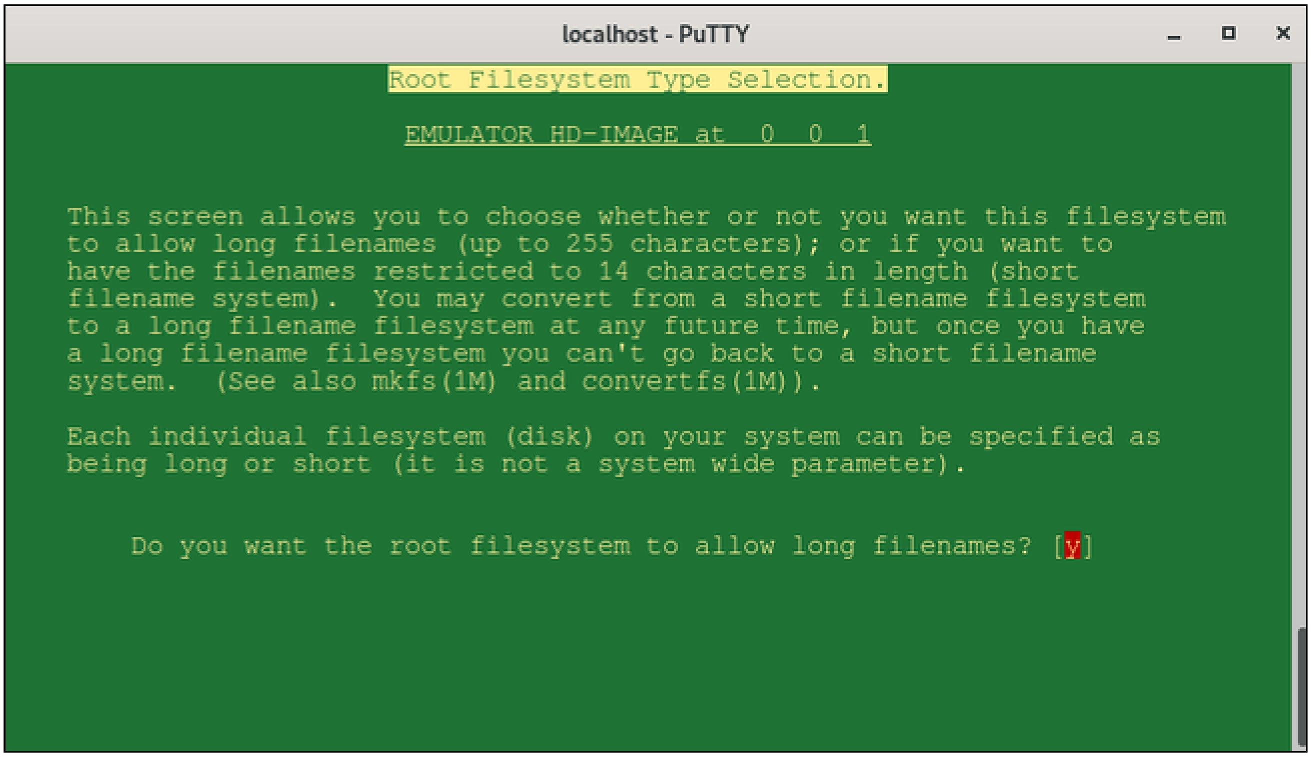
The following image shows a sample:



```
              HP-UX INSTALLATION UTILITY  -- ROOT DESTINATION MENU



      If the disk shown below (name and system location) is the desired
      destination device, press <Return>.
      If the desired ROOT disk is not listed, make sure it is connected
      properly and turned on, then select the "Search Again" item.
      If your disk is STILL not recognized, you can use the
      "Other disk" item to manually enter the Disk address.



                 Slot   Bus  Func
   Disk         Number  Addr  Num
  -------------------------------
1. EMULATOR HD-IMAGE at  0   0   1
2. Search Again
3. Other disk
4. Exit Install


Enter selection [1]
```

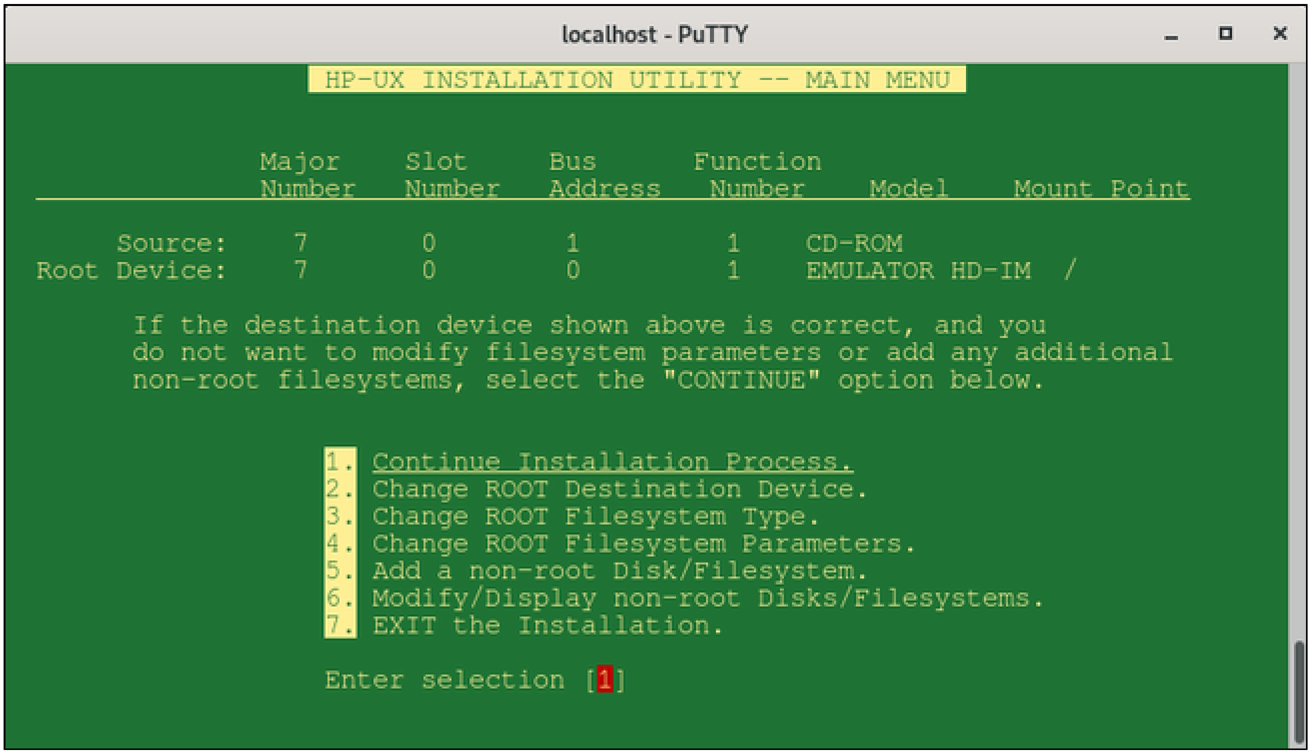
In the above sample, select the offered disk by entering its number and press **Return**.

Before the installation starts, you have to select if the root disk should support **long file names**:



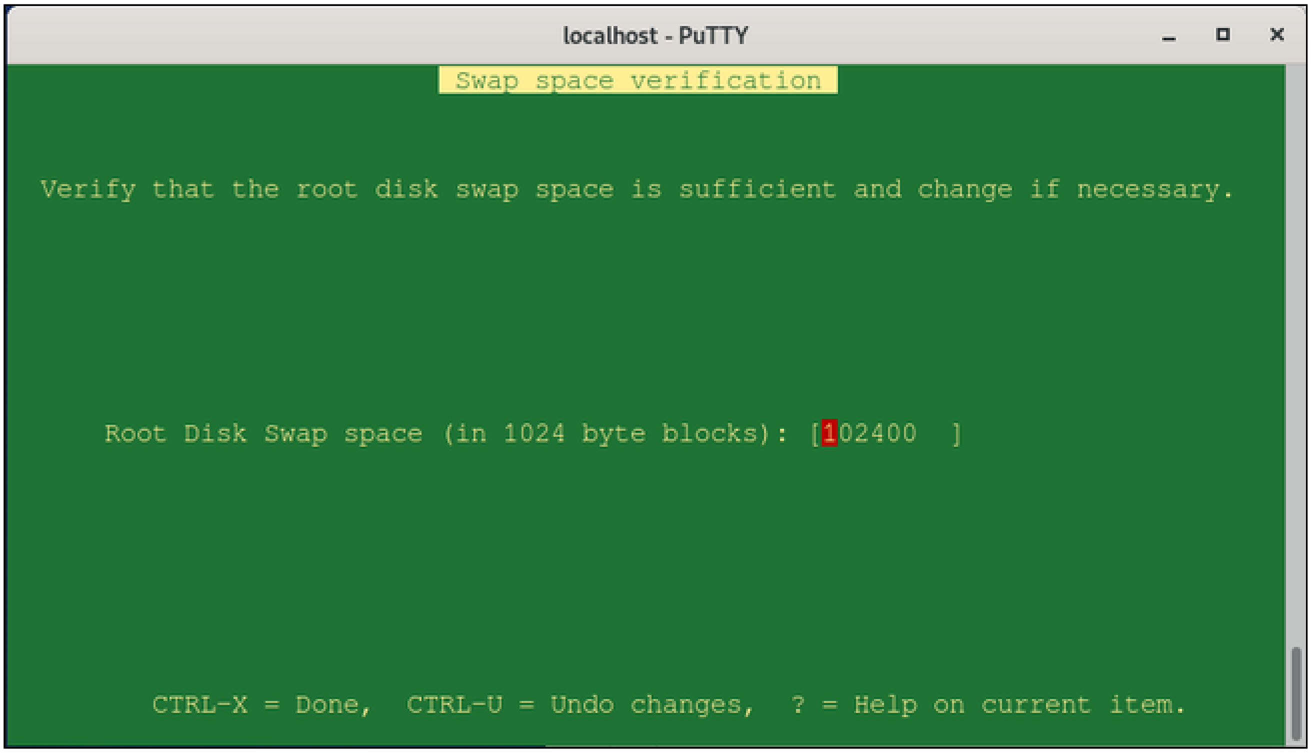
```
                        Root Filesystem Type Selection.

                         EMULATOR HD-IMAGE at  0  0  1


   This screen allows you to choose whether or not you want this filesystem
   to allow long filenames (up to 255 characters); or if you want to
   have the filenames restricted to 14 characters in length (short
   filename system).  You may convert from a short filename filesystem
   to a long filename filesystem at any future time, but once you have
   a long filename filesystem you can't go back to a short filename
   system.  (See also mkfs(1M) and convertfs(1M)).

   Each individual filesystem (disk) on your system can be specified as
   being long or short (it is not a system wide parameter).


       Do you want the root filesystem to allow long filenames? [y]
```

You will be offered additional installation options as shown in the sample below:



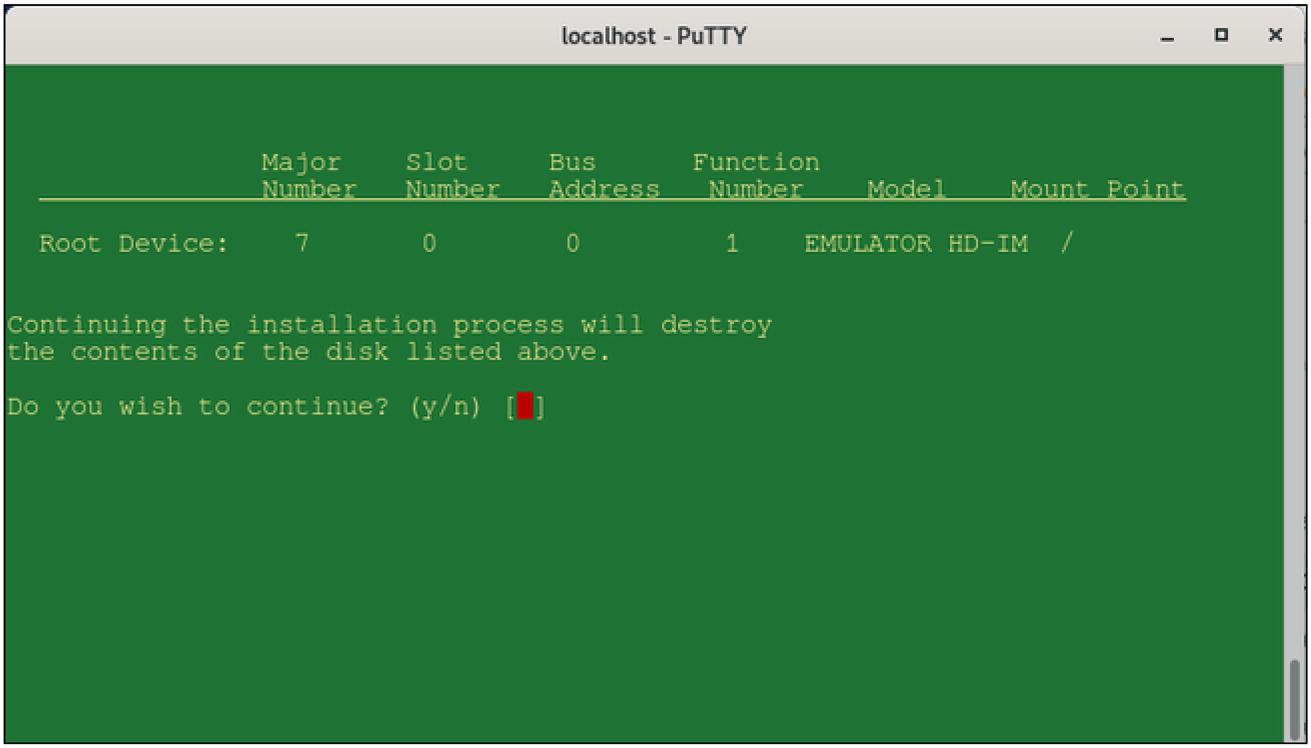
```
                    HP-UX INSTALLATION UTILITY -- MAIN MENU


              Major    Slot     Bus      Function
              Number   Number   Address   Number    Model    Mount Point

     Source:    7        0        1         1     CD-ROM
 Root Device:   7        0        0         1     EMULATOR HD-IM   /

        If the destination device shown above is correct, and you
        do not want to modify filesystem parameters or add any additional
        non-root filesystems, select the "CONTINUE" option below.


                    1. Continue Installation Process.
                    2. Change ROOT Destination Device.
                    3. Change ROOT Filesystem Type.
                    4. Change ROOT Filesystem Parameters.
                    5. Add a non-root Disk/Filesystem.
                    6. Modify/Display non-root Disks/Filesystems.
                    7. EXIT the Installation.

                    Enter selection [1]
```

Then you are asked to configure the **swap space** for the system:



```
                    Swap space verification


  Verify that the root disk swap space is sufficient and change if necessary.



      Root Disk Swap space (in 1024 byte blocks): [102400   ]



         CTRL-X = Done,   CTRL-U = Undo changes,   ? = Help on current item.
```

And eventually, you have to confirm that the installation should be performed:



```
                Major     Slot      Bus       Function
                Number    Number    Address    Number     Model     Mount Point

  Root Device:    7         0         0          1     EMULATOR HD-IM  /

Continuing the installation process will destroy
the contents of the disk listed above.

Do you wish to continue? (y/n)  [ ]
```

After the confirmation, the basic system with the essential operating system components will be installed and the system is automatically shut down.

You will be returned to the console prompt.

Type **exit** at the **pa9-32>** prompt to stop the emulator.

HP-UX 9.07 Installation Phase 2: Installing the Operating System Filesets

During this step, the basic system will be used to install the other components of the operating system.

Step 1: replace the **install** ISO file in the emulator configuration with the **Core_OS** ISO file as described below:

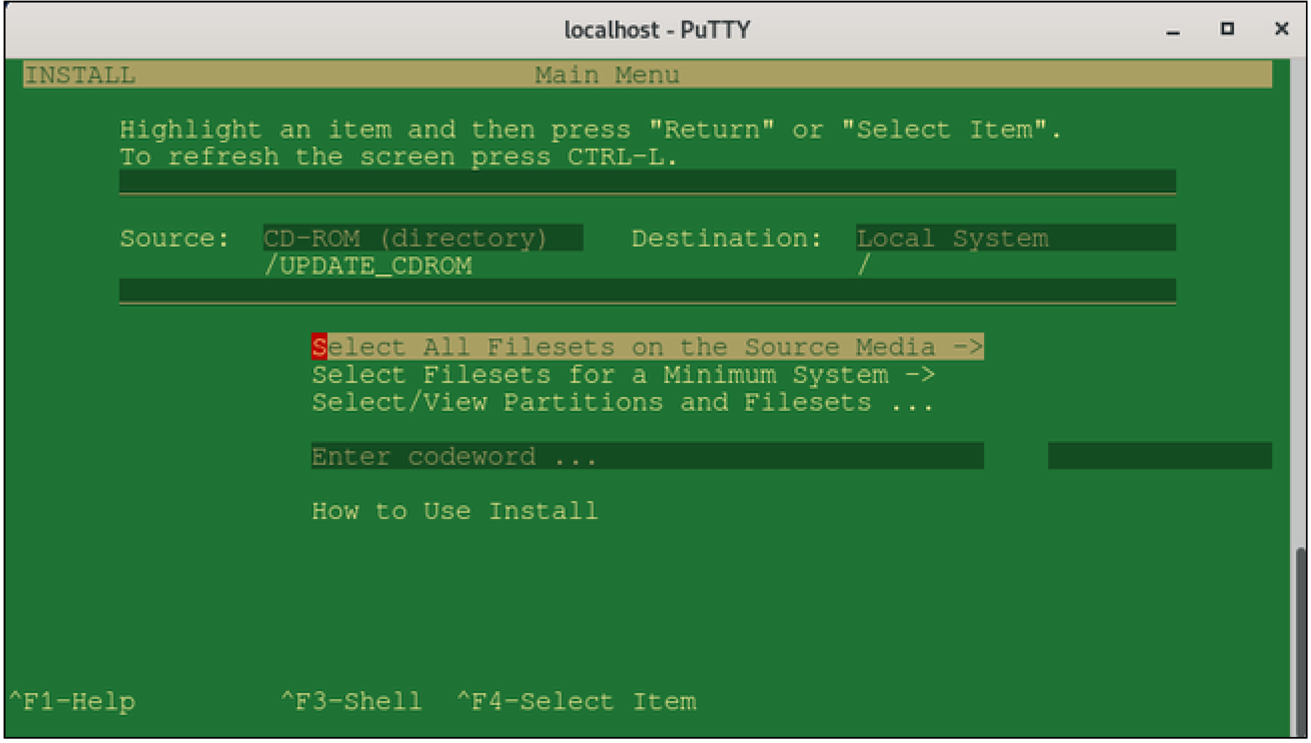
- Open the emulator configuration file in a text editor.
- Find the definition for the virtual CD-ROM (DKA100 in this example).
- Modify the definition to point to your **Core_OS** ISO file for the mini-system.
  Example:
  `DKA100.image="/data/Stromasys/host1/Data/Disk/HP-UX_9.07_Core_OS_S700.iso"`

Step 2: start the emulator and boot the basic HP-UX system as described below:

- Go to the directory where the emulator configuration file is stored.
- Start the emulator.
  Example:
  `# /opt/charon/bin/charon-par -f host1.cfg`
- In the green PuTTY window, at the prompt **Main Menu: Enter command or menu >,** enter the boot command **`boot 2/0/1.0.0`** and press **Return** to boot from the system disk (DKA0 in this example).
- Enter **N** at the **Interact with IPL** prompt.
- The basic HP-UX system should boot.

Step 3: install the Core OS filesets as described below:

- Confirm that you have the correct ISO installed.
- Enter your choice for the fileset selection (normally all filesets of the Core OS).



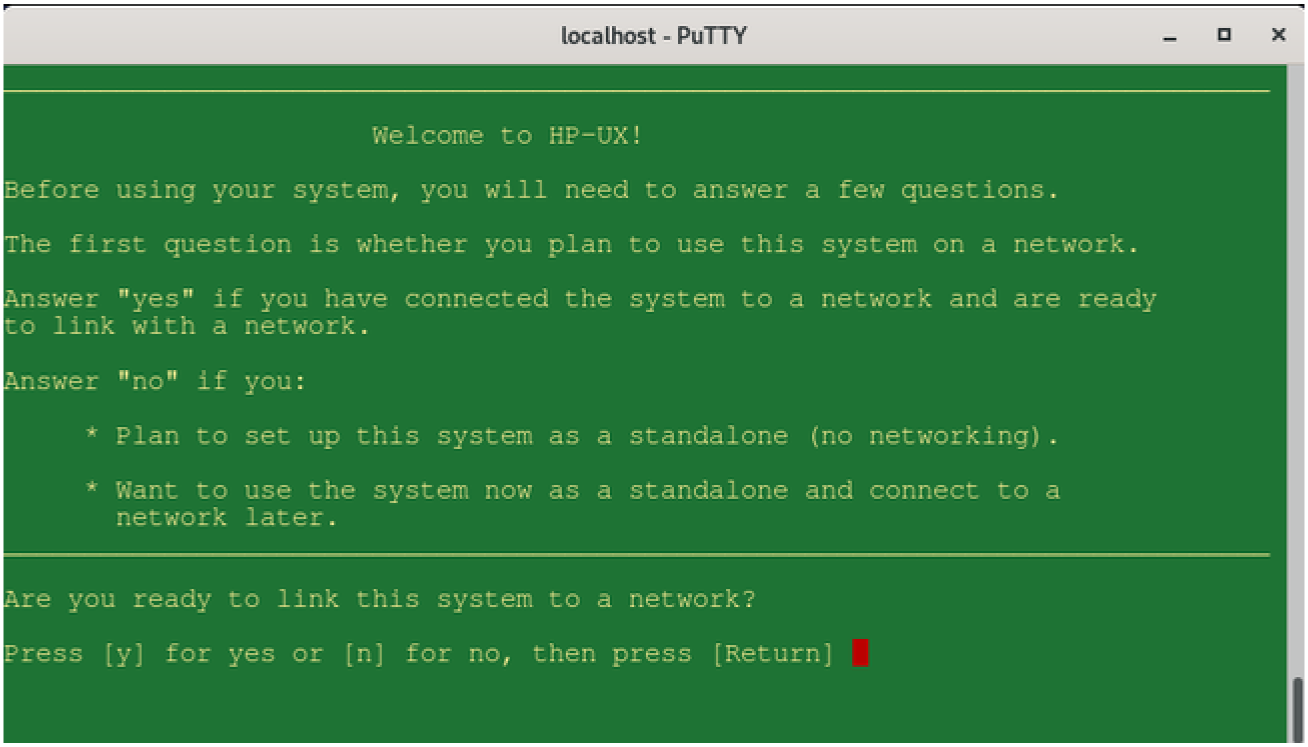
```
INSTALL                                 Main Menu

       Highlight an item and then press "Return" or "Select Item".
       To refresh the screen press CTRL-L.

       Source:  CD-ROM (directory)    Destination:  Local System
                /UPDATE_CDROM                       /

                    Select All Filesets on the Source Media ->
                    Select Filesets for a Minimum System ->
                    Select/View Partitions and Filesets ...

                    Enter codeword ...

                    How to Use Install

^F1-Help         ^F3-Shell  ^F4-Select Item
```

- You will be asked to confirm your choice twice.
- Then, the loading of the filesets begins.
- At the end, the HP-UX system is shut down and returns to the console environment.

HP-UX 9.07 Installation Phase 3: Booting and Configuring HP-UX

Upon boot, you will be asked a few questions for the initial configuration of your system.

Step 1: decide if the system should be connected to a network:



```
------------------------------------------------------------------------------

                       Welcome to HP-UX!

Before using your system, you will need to answer a few questions.

The first question is whether you plan to use this system on a network.

Answer "yes" if you have connected the system to a network and are ready
to link with a network.

Answer "no" if you:

     * Plan to set up this system as a standalone (no networking).

     * Want to use the system now as a standalone and connect to a
       network later.

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

Are you ready to link this system to a network?

Press [y] for yes or [n] for no, then press [Return]
```

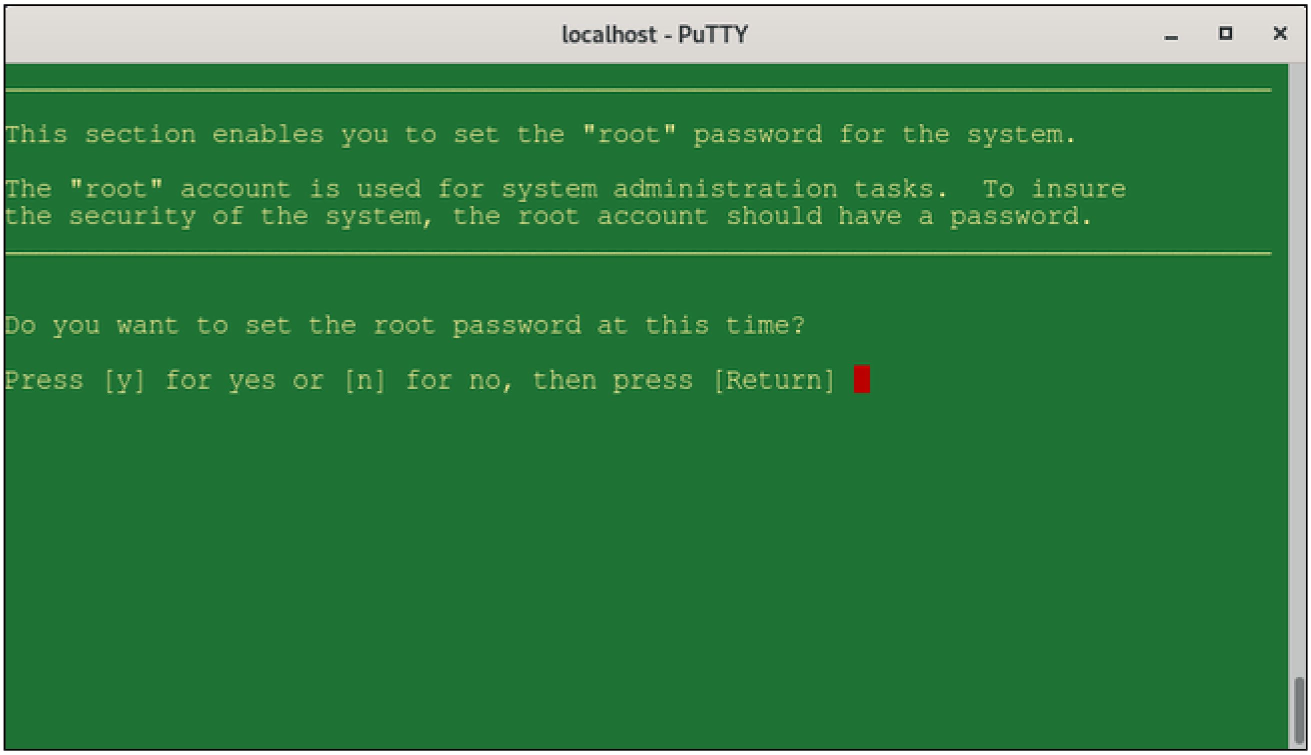
If you answer yes to this question, you will be lead through a series of questions and dialogs allowing you to configure

- hostname
- IP address and netmask
  For cloud environments, please note: in case of a dedicated Ethernet interface, the IP address of the emulator must be set to the value assigned by the cloud provider to the corresponding host NIC.
- default gateway
- DNS
- fontserver

Step 2: configure the timezone and time.

A set of questions and dialogs will lead you through this step. Note that HP-UX 9.07 may not allow you to set the current year (last year to set may be 1999). You can set the date later using the date command:
# date [**-u**] [*mmddhhmm*[[*cc*]*yy*]]
(Format: month / day of month / hour / minutes / century / year)

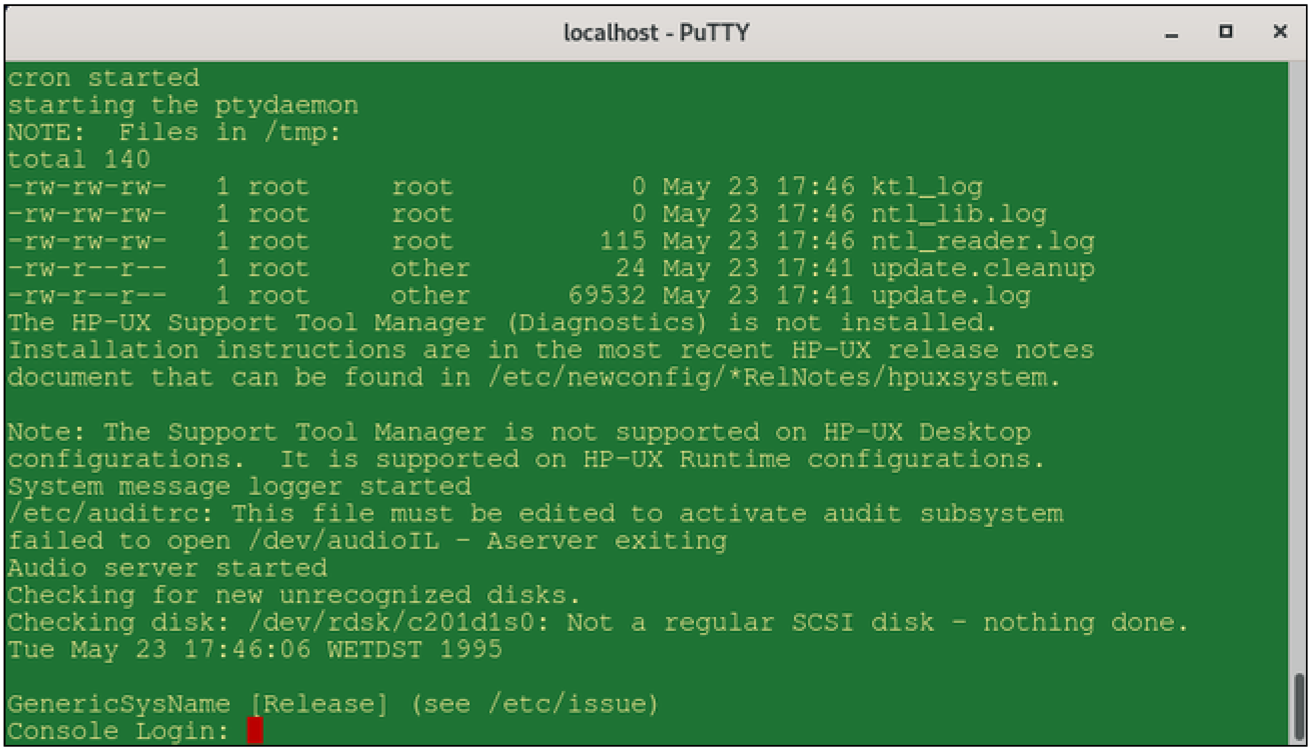
Step 3: set the root password.



```
This section enables you to set the "root" password for the system.

The "root" account is used for system administration tasks.  To insure
the security of the system, the root account should have a password.


Do you want to set the root password at this time?

Press [y] for yes or [n] for no, then press [Return]
```

Step 4: login and terminal configuration.

After the above steps have been completed, you will be able to login as the root user and start using the system. At the beginning, you have to select your terminal settings.



```
cron started
starting the ptydaemon
NOTE:  Files in /tmp:
total 140
-rw-rw-rw-   1 root     root           0 May 23 17:46 ktl_log
-rw-rw-rw-   1 root     root           0 May 23 17:46 ntl_lib.log
-rw-rw-rw-   1 root     root         115 May 23 17:46 ntl_reader.log
-rw-r--r--   1 root     other         24 May 23 17:41 update.cleanup
-rw-r--r--   1 root     other      69532 May 23 17:41 update.log
The HP-UX Support Tool Manager (Diagnostics) is not installed.
Installation instructions are in the most recent HP-UX release notes
document that can be found in /etc/newconfig/*RelNotes/hpuxsystem.

Note: The Support Tool Manager is not supported on HP-UX Desktop
configurations.  It is supported on HP-UX Runtime configurations.
System message logger started
/etc/auditrc: This file must be edited to activate audit subsystem
failed to open /dev/audioIL - Aserver exiting
Audio server started
Checking for new unrecognized disks.
Checking disk: /dev/rdsk/c201d1s0: Not a regular SCSI disk - nothing done.
Tue May 23 17:46:06 WETDST 1995

GenericSysName [Release] (see /etc/issue)
Console Login:
```

Stopping the Emulator

In order to stop the emulator in an orderly manner, perform the following steps:

1. Shut down the guest operating system:

```
# shutdown -h now
```

Starting with Charon-PAR version 3.0.5 this will, by default, automatically stop the emulator when it detects that the operating system has been halted. See the system.stop_on_halt parameter in the Configuration File Reference. If the emulator is not stopped automatically, continue with step 2 below.

2. Starting with version 3.0.5, the emulator is stopped automatically after the guest operating system shutdown is complete (unless the configuration parameter `system.stop_on_halt` has been set to **`false`**). If the emulator is not stopped automatically after the shutdown of the guest operating system, different methods must be used to stop it in different situations:

- **The emulator was started in interactive mode and the Charon-PAR console is available in current terminal**: enter the **exit** command at the **pa9-32>** prompt and hit **Enter**. This will cause the emulator to close the disk images of the emulated system and to exit.
- **The Charon-PAR console is available on a local TCP port (emulator started with -c or --console-port parameter)**: this situation will exist most frequently if the emulator has been started as a daemon or service. To stop the emulator access the Charon-PAR console via **telnet** to *localhost* on the port specified when starting the emulator, enter the **exit** command at the **pa9-32>** prompt, and hit Enter.
  Access to the Charon-PAR console via a TCP port is available starting with Charon-PAR version 3.0.1. The **CTRL+B** key combination has been disabled on the emulator console starting with this version.

- **The emulator was started as a daemon or service and the Charon-PAR console is not available via a local TCP port**: This situation can occur if you run a version before 3.0.1 or you started the emulator without the **`-c`** option.
  - For versions 3.0.1 and higher: the emulator must be stopped using the **kill** command from the root user on the Charon host system. Perform the following steps to stop the Charon-PAR process:
    - Find the PID (process id) of the process: **`# ps -ef | grep -i charon-par`**
    - Stop the process: **`# kill`** *`<charon-par-pid>`*
  - For versions before 3.0.1: on the emulator console, after shutting down the guest operating system, press **CTRL+B**. After a few seconds, a traceback may be shown. Then the PDC console prompt (**Main menu:**) will appear. Type **exit** to stop the emulator.

For more information about the Charon-PAR console, please refer to Charon-PAR Console. For more information about the Charon-PAR command-line options, please refer to Charon-PAR Command-Line Options.

Charon-PAR Console

Contents

General Information

The Charon-PAR console has several functions:

- It shows log output during the time the emulator is active.
- It allows to stop the emulator using the **exit** command at the console prompt.
- It accepts certain commands that are interpreted by the running emulator.

Accessing the Charon-PAR Console

The Charon-PAR Console can be accessed in two different ways:

- When starting the emulator interactively without any console-specific parameters, the console will open in the terminal in which the emulator was started.
- Starting with version 3.0.1, there is an additional command-line parameter when starting the emulator. It allows the redirection of the Charon-PAR console to a TCP port (local access only).

Charon-PAR Console in Current Host Terminal Window

The example below shows how a Charon-PAR emulator is started interactively and the Charon-PAR console is displayed in the current host terminal window:

```
# /opt/charon/bin/charon-par -f ./host1-hpux-physeth.cfg
20201026:134810.542014:Charon-PAR
20201026:134810.542188:Version 3.0.1, build 21.500 38927fc Sep 29 2020 17:28:55 Prod
20201026:134810.542231:Copyright (c) 2011-2020 by Stromasys.
20201026:134810.547210:Checking the available license key "445532399".
20201026:134810.660840:Found license key: "445532399".

<lines removed>

20201026:134815.114034:Product Name = Charon-PA9-64-L4 License key ID = 445532399.
20201026:134815.115400:Product DIT level adv
20201026:134815.117806:config load: ./host1-hpux-physeth.cfg
20201026:134815.119083:System model: rp7400-1-650

<lines removed>

20201026:134816.511458:cpu0: run cpu loop
20201026:134816.511534:Host CPU freq: 3461
20201026:134816.638887:starting process 'putty -load PAR-Telnet-VT100'

20201026:134816.640643:uart2: port address: 0.0.0.0:30002
pa9-64>20201026:134816.742516:uart0: Client connected

pa9-64>
pa9-64>
pa9-64>
```

Charon-PAR Console on Network Port

This feature is available starting with Charon-PAR version 3.0.1. The example below shows how a Charon-PAR emulator is started in daemon mode and the Charon-PAR console is reachable on TCP port 12345 (the command-line parameters are described in Charon-PAR Command-Line Options):

```
# /opt/charon/bin/charon-par -x  -f ./host1-hpux-physeth.cfg -c 12345
20201026:135307.060655:Charon-PAR
20201026:135307.060881:Version 3.0.1, build 21.500 38927fc Sep 29 2020 17:28:55 Prod
20201026:135307.060943:Copyright (c) 2011-2020 by Stromasys.
20201026:135307.062010:daemon mode pid: 17198
20201026:135307.066758:Checking the available license key "445532399".
20201026:135307.168306:Found license key: "445532399".

<lines removed>

20201026:135311.623945:Product Name = Charon-PA9-64-L4 License key ID = 445532399.
20201026:135311.624498:Product DIT level adv
20201026:135311.625298:config load: ./host1-hpux-physeth.cfg
20201026:135311.626446:System model: rp7400-1-650

<lines removed>

20201026:135312.232223:cpu0: run cpu loop
20201026:135312.232297:Host CPU freq: 3461
20201026:135312.351846:starting process 'putty -load PAR-Telnet-VT100'

20201026:135312.355624:uart2: port address: 0.0.0.0:30002
20201026:135312.357104:Telnet console address: 127.0.0.1:12345
20201026:135312.457464:uart0: Client connected

[root@redhat7 9KTraining]#
[root@redhat7 9KTraining]# telnet localhost 12345
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

pa9-64>
```

Important Commands on the Charon-PAR Console

There is a small number of commands that are relevant to the user and can be entered on the Charon-PAR console:

- **`exit`**: stops the emulator. Ensure that the guest operating system has been cleanly shut down before entering this command.
- **`signal_toc`**: equivalent to pressing CTRL+B on a physical PA-RISC system. It brings the emulated system to the emulated PDC console, for example, to access the emulator console without restarting the emulator after the guest operating system has been shut down. You may have to answer a question about writing a dump file before the PDC console can be accessed. The CTRL+B key combination is blocked on the emulator console itself (starting with version 3.0.1) to avoid unintentional damage.
- <**MK***XN*>**`.load`**: load a tape for a HP-UX guest operating system if the **autoload** parameter has not been configured.

Installing Client Systems via an Ignite-UX Server (from ver. 3.0.9)

Contents

Starting with version 3.0.9, it is planned that the operating system running in a Charon-PAR emulator can be installed from an Ignite-UX server.

Please note:

- This option does not implement a full network boot, but it allows storing the installation media on the Ignite server and perform the HP-UX installation using this data.
- This chapter describes the basic step required on the client side. It does not describe how to set up the Ignite server.

Setting up the Boot Environment on the Client Side

To enable an operating system installation using an Ignite server, the boot directory with all contents must be copied from the server to the client:

- On the Ignite server:
  - Create an archive (e.g., a tar archive) including all files under **/opt/ignite/boot**.
- Copy the archive to the Charon emulator host.
- Unpack the archive on the Charon emulator host in a place accessible by the emulator.

Sample of the directory content (the operating specific directories contain additional files):

```
# ls H9KIgniteClient/opt/ignite/boot/
AUTO              boot_lif            fpswa.efi        hpux.efi  Rel_B.11.00  Rel_B.11.22
auto_globals  EFI_CD_image  fs_cfg.def        nbp.efi    Rel_B.11.11  Rel_B.11.23
```

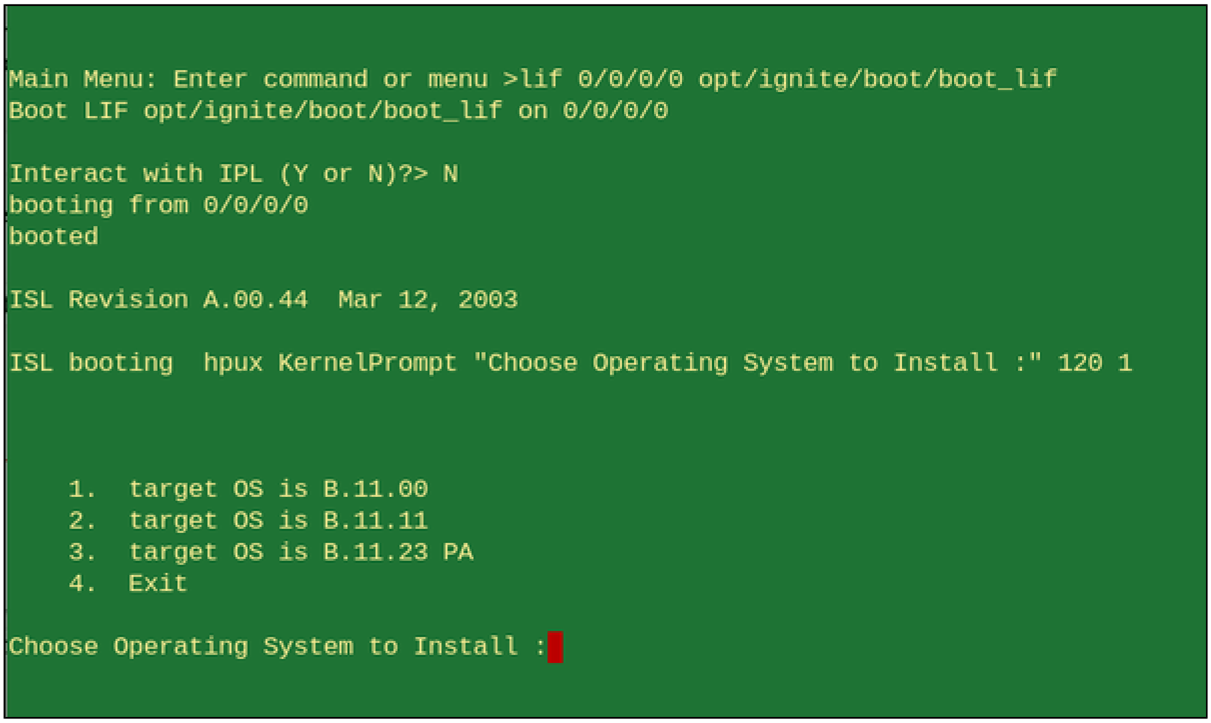
Booting and Installing the Client

To install a Charon-PAR emulator client from an Ignite-UX server, perform the following steps:

- Perform the preparatory steps to set up emulator storage and networking as needed.
- Configure the emulator as needed. In particular
  - configure a network interface in a network connected to the Ignite server,
  - if client MAC addresses are defined for the clients on the server, set a static MAC address in the emulator configuration.
- Start the emulator.
- On the PDC console, use the following command to boot the client system:

```
lif 0/0/0/0 /path-to-local-ignite-boot-directory/boot_lif
```

The example below shows the first steps of the boot sequence:

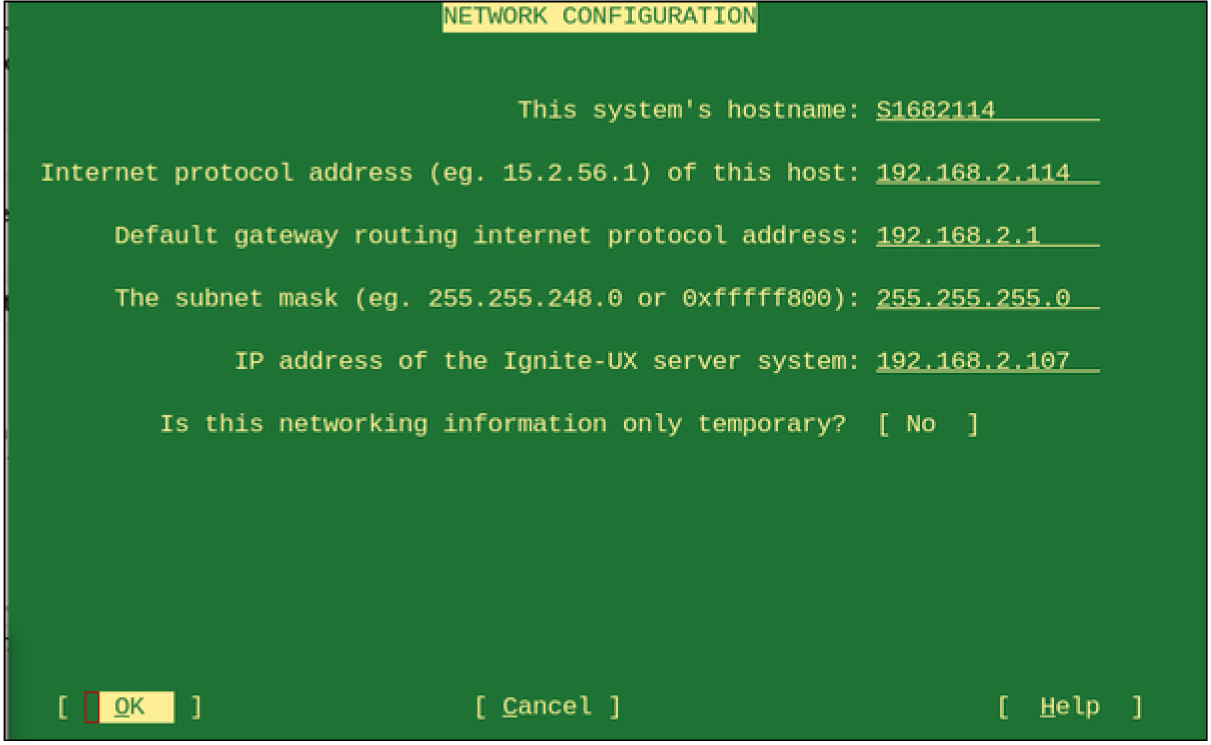


```
Main Menu: Enter command or menu >lif 0/0/0/0 opt/ignite/boot/boot_lif
Boot LIF opt/ignite/boot/boot_lif on 0/0/0/0

Interact with IPL (Y or N)?> N
booting from 0/0/0/0
booted

ISL Revision A.00.44  Mar 12, 2003

ISL booting  hpux KernelPrompt "Choose Operating System to Install :" 120 1



    1.  target OS is B.11.00
    2.  target OS is B.11.11
    3.  target OS is B.11.23 PA
    4.  Exit

Choose Operating System to Install :
```

After selecting the desired operating system, further selections are:

- The terminal emulation to use (VT100).
- The action desired (install HP-UX).
- The type of installation (e.g., guided).

Then, the network data of the Ignite server and the client is displayed similar to the example below:



```
                                  NETWORK CONFIGURATION


                                  This system's hostname: S1682114

  Internet protocol address (eg. 15.2.56.1) of this host: 192.168.2.114

       Default gateway routing internet protocol address: 192.168.2.1

       The subnet mask (eg. 255.255.248.0 or 0xfffff800): 255.255.255.0

               IP address of the Ignite-UX server system: 192.168.2.107

          Is this networking information only temporary?  [ No  ]








  [  OK  ]                    [ Cancel ]                           [  Help  ]
```

After confirming the network configuration, the normal operating system installation starts.

Additional Configuration Options

Contents

Serial Line Emulation Notes

Table of contents

Serial Lines for Emulated Systems

Most models emulated by Charon-PAR have two serial lines (LDev 20 and 21). One of these emulated lines is mapped to the serial console, the other to a second serial port. Some models may have more serial lines. Please refer to Emulated Model Hardware Configuration Details for more information. This section provides an overview of the serial line configuration. Please also refer to the Configuration File Reference.

Configuration Parameters Overview

Each emulated line has a device path denoting bus and slot information in the emulated system (see configuration file template or chapter *Emulated Model Hardware Configuration Details*) and a configuration path. The configuration path is used to set up the serial line configuration in the emulator configuration file.

The following table shows the device and configuration paths used for the two serial lines:

Charon-PAR 64-bit models		**Charon-PAR 32-bit model 720**		**Charon-PAR 32-bit model B132L**	
Serial line	Config. path	Serial line	Config. path	Serial line	Config. path
1 = Console	**`serial.uart0.device`**	1 = Console	**`asp.uart0.device`**	1 = Console	**`gsc.lasi.uart.device`**
2	**`serial.uart2.device`**	2	**`asp.uart1.device`**	2	**`gsc.wax.uart.device`**

Serial port configuration options are specified using the following syntax:

```
serial.uartX.device.<option>="<value>"
```

or (for model 720)

```
asp.uartX.device.<option>="<value>"
```

or (for model B132L)

```
gsc.lasi.uart.device.<option>="<value"
```

If the Super/IO module has been loaded, there are two additional serial lines. In the configuration, they are identified by

- **superio_001.uart0** and
- **superio_001.uart1**.

The basic configuration options are the same as for the default serial lines.

The most important configuration options are listed below. For additional parameters and more detailed information, please refer to the Configuration File Reference.

Serial port option	Description
type	The serial port type determines the protocol used on the serial port. Possible values are: • **DUMMY**: serial interface presented to emulator is not connected. • **socket**: the port is set to TCP raw mode. • **telnet**: the port supports the telnet protocol. • **pty**: indicates that the emulated serial line is connected to a Linux pseudo-terminal. • **tty**: emulated serial line is connected to physical hardware (serial hardware must be supported by Linux). • **RFC2217**: emulated serial line is connected to a serial line server across the IP network.
port	Each port is mapped to a TCP socket on the emulator host or to a physical line. This parameter defines the **TCP port** used, the name of the **physical device**, or the port and address of a serial line server. A port number must be unique among all active port numbers on the host system. A **TCP port** is specified as `"<where-to-listen>:<portnumber>"`. It requires a port type of **socket** or **telnet**. Default values (listening on all interfaces): • ":30000" for line 1 • ":30002" for line 2 To restrict console connections to certain host interfaces or host IP addresses, set *where-to-listen* to the respective interface name or IP address. The settings **`lo:`***<portnumber>* and *<portnumber>* (without the colon) are equivalent. A **physical device** is specified as "`<device-file>`". It requires a port type of **device**. Example: • "/dev/ttyS0" • "/dev/ttyUSB0" A **serial line server** is specified as "`<ip-address>`:`<port>`".
command	This parameter is optional. It can be used to specify a terminal emulation program that is started automatically when the emulated system is started. Charon-PAR provides preconfigured profiles for PuTTY (PAR-Socket PAR-Telnet PAR-Telnet-VT100) that can be used to connect to the emulated system via a serial line.

The configuration file templates contain the following default configurations:

Line	Charon-PAR/PA3	Charon-PAR/PA9
	Emulated serial console on port 30000 using the telnet protocol. PuTTY started automatically with preconfigured profile.	

1	`serial.uart0.device.type="telnet"`
	`serial.uart0.device.port=":30000"`
	`serial.uart0.device.command="putty -load PAR-Telnet"`
		PA9-64
		`serial.uart0.device.type="telnet"`
		`serial.uart0.device.port=":30000"`
		`serial.uart0.device.command="putty -load PAR-Telnet-VT100"`
		PA9-32 (model 720)
		`asp.uart0.device.type="telnet"`
		`asp.uart0.device.port=":30000"`
		`asp.uart0.device.command="putty -load PAR-Telnet-VT100"`
		PA9-32 (model B132L)
		`gsc.lasi.uart.device.type = "telnet"`
		`gsc.lasi.uart.device.port = ":30000"`
		`gsc.lasi.uart.device.command="setsid putty -load PAR-Telnet-VT100"`

	Emulated second serial line in raw mode on port 30002.	

2	`serial.uart2.device.type="socket"`
	`serial.uart2.device.port=":30002"`
		PA9-64
		`serial.uart2.device.type="socket"`
		`serial.uart2.device.port=":30002"`
		PA9-32 (model 720)
		`asp.uart1.device.type="socket"`
		`asp.uart1.device.port=":30002"`
		PA9-32 (model B132L)
		`gsc.wax.uart.device.type = "socket"`
		`gsc.wax.uart.device.port = ":30002"`

Please note:

By default, **pressing Ctrl+C in the Charon-PAR console** is passed to the emulator's child processes and kills the PuTTY console process. If this is not desired, you can modify the command line to include the **setsid** command to start the telnet command in a new session.
For example: `serial.uart0.device.command="setsid putty -telnet -P 30000 localhost"`

HP-UX Interactive Login Activation

To enable interactive login on a non-console serial line in HP-UX, a GETTY process has to be started for the terminal line.

In versions before 11.31, you can add an additional terminal via SAM (**Peripheral Devices > Terminals and Modems > Actions > Add Terminal**). This will also add the necessary entries in to **/etc/inittab**.

You can also (as the root user) edit **/etc/inittab** to add the configuration manually or to modify an existing configuration.

To identify available serial lines, you can use the ioscan command:

```
# ioscan -fn -C tty

Class I H/W Path Driver S/W State H/W Type Description

==================================================================

tty 0 0/0/4/0 asio0 CLAIMED INTERFACE PCI Serial (103c1048)

                   /dev/GSPdiag1   /dev/tty0p0  /dev/tty0p4

                   /dev/diag/mux0  /dev/tty0p1

                   /dev/mux0       /dev/tty0p2
```

Sample inittab entry:

```
ttp2:234:respawn:/usr/sbin/getty -h tty0p2 9600
```

After editing the file, **any changes must be activated** by running the command

```
# init q
```

The command

```
# ps -ef | grep -i getty
```

can be used to verify that the processes are running. Note that a GETTY process is only shown for lines without active login session.

After this configuration, when connecting to the serial line, a login prompt should be displayed. If you use the Linux telnet client to connect to an emulated line mapped to a TCP socket, use **mode character** to avoid local echo.

Please note: in version 11.31, SAM was replaced by SMH. This tool does not have the same configuration option, so only the manual configuration option is available.

Terminal Emulation Considerations

The emulated serial lines can be accessed via a terminal emulation program running on the emulator host system or on a remote system.

Please note: if the terminal emulation program runs on a remote system, any firewall between the remote system and the emulator host system must permit the configured ports.

Stromasys supplies PuTTY profiles with the Charon-PAR kit. However, customers can select another terminal emulation program better suited to their purposes. This is especially true for MPE/iX guest systems where applications often use block mode terminals. This mode is not supported by PuTTY.

If not done so already during the installation, copy the provided PuTTY profiles to the correct directory on the host system. See *Post-Installation Tasks* in Charon-PAR Software Installation.

Accessing Emulated Charon-PAR/PA3 Systems via Terminal Emulation

Charon-PA3 guest systems can be accessed directly from the Linux emulator host using either the PuTTY or xhpterm terminal emulator described below, both of which run directly on the Linux host desktop, or by simply using the **telnet** command. Each has limitations, however. Neither terminal emulator provides a reliable way to paste multiple lines of text into :EDITOR , for example.
Once the network of the Charon-PAR/PA3 guest system has been configured configured, it may be preferable to connect to the emulated system from a networked PC using any of the robust commercially-available terminal emulators.

Using Non-Graphical Telnet Client

In the absence of a graphical user environment, terminal emulators such as PuTTY and xhpterm cannot be used. In such cases, you can use the telnet command on the Charon host or from a remote system to access the serial console port of the emulated PA3 system.

The basic command syntax is as follows:

```
$ telnet <charon-host> <console-tcp-port>
```

Parameters:

- *charon-host*: IP address or hostname of the Charon-PAR host system on which the emulator runs
- *console-tcp-port*: The TCP port configured for the serial console port in the configuration file of the emulated PA3 system

Please note: Line-editing is very limited. To delete a character on the command-line, use CTRL+BACKSPACE. Note that the deleted character will still be visible until it is overwritten. Block-mode applications are not supported.

Using PuTTY

The PuTTY terminal emulator is the preferred method to use as the Charon-PAR/PA3 console. It does not, however, support the terminal escape sequences used by the legacy hardware, so any inverse or highlighted text codes will be ignored, and block mode applications cannot be run. Using PuTTY in telnet mode (as supplied) allows use of the **Break** key. PuTTY supports copy and paste. To copy text to the clipboard, just select it with the left mouse button (this automatically copies the selection to the clipboard). To paste the clipboard into a PuTTY window, use Shift-Ins . Unfortunately, PuTTY cannot be used to reliably paste large blocks of text (even with MPE type-ahead enabled), as data overruns can easily occur.

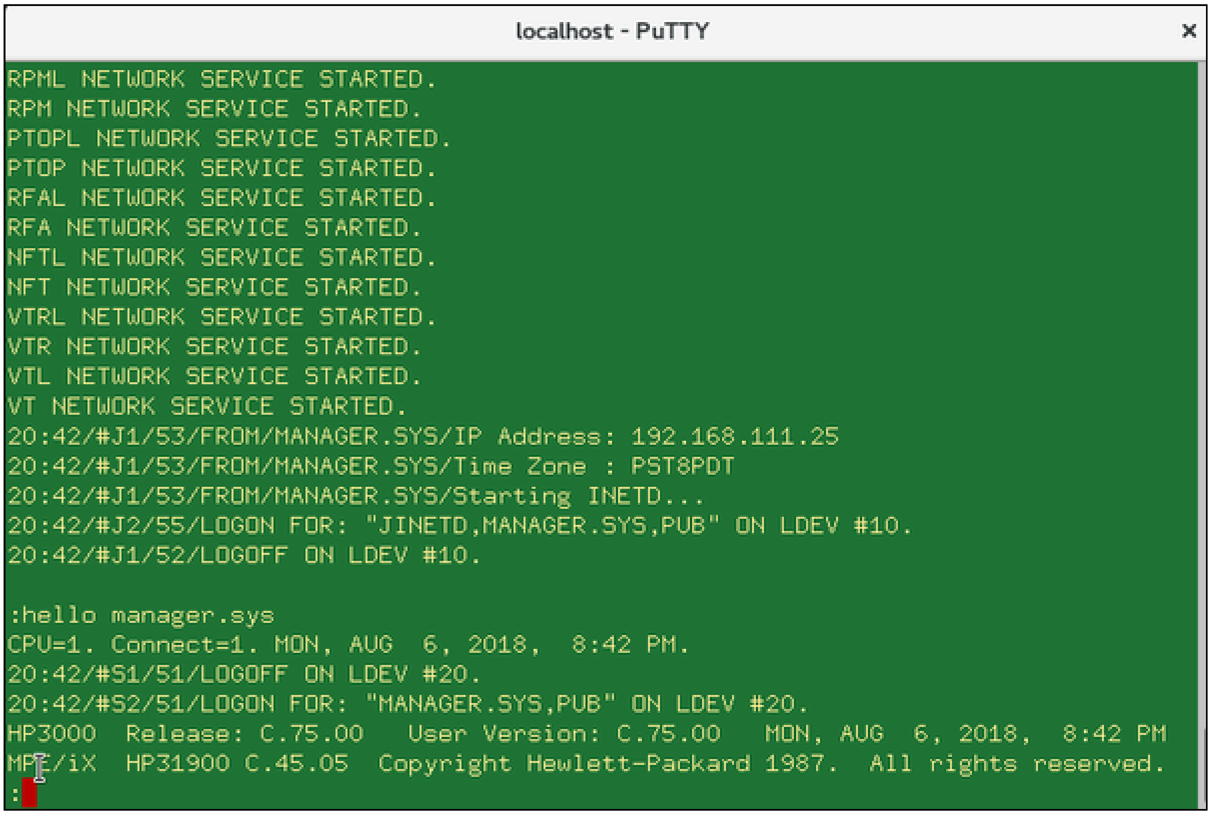
By default, the emulator configuration is set up to start PuTTY on the console port with a command similar to the following example:

`serial.uart0.device.command="putty -load PAR-Telnet"` (default port 30000).

To start an terminal emulator manually, perform the following steps:

	Step	Command / Action
1	Make sure the preconfigured PuTTY profiles are in the directory **.putty/sessions** or **`.config/putty/sessions`** in your home directory.	(see *Post-Installation Tasks* in Charon-PAR Software Installation)
2	Start PuTTY from the command-line.	**`$ putty -load PAR-Telnet`**
3	Should you get an error that a font was not found, change the font.	Start PuTTY from the command-line. **`$ putty`** • Select and load the required profile. • Select **Fonts** on the left and select a monospace font. • Select **Session** on the left and save the profile.
4	To connect from a remote host, change localhost in the template to the correct hostname or IP address.	Start PuTTY from the command-line. **`$ putty`** • Select and load the required profile. • Change the hostname to the one of the Charon-PAR host as required. • Connect to the remote Charon-PAR system.

The image below shows a PuTTY terminal window connected to the guest MPE/iX system console:



```
RPML NETWORK SERVICE STARTED.
RPM NETWORK SERVICE STARTED.
PTOPL NETWORK SERVICE STARTED.
PTOP NETWORK SERVICE STARTED.
RFAL NETWORK SERVICE STARTED.
RFA NETWORK SERVICE STARTED.
NFTL NETWORK SERVICE STARTED.
NFT NETWORK SERVICE STARTED.
VTRL NETWORK SERVICE STARTED.
VTR NETWORK SERVICE STARTED.
VTL NETWORK SERVICE STARTED.
VT NETWORK SERVICE STARTED.
20:42/#J1/53/FROM/MANAGER.SYS/IP Address: 192.168.111.25
20:42/#J1/53/FROM/MANAGER.SYS/Time Zone : PST8PDT
20:42/#J1/53/FROM/MANAGER.SYS/Starting INETD...
20:42/#J2/55/LOGON FOR: "JINETD,MANAGER.SYS,PUB" ON LDEV #10.
20:42/#J1/52/LOGOFF ON LDEV #10.

:hello manager.sys
CPU=1. Connect=1. MON, AUG  6, 2018,  8:42 PM.
20:42/#S1/51/LOGOFF ON LDEV #20.
20:42/#S2/51/LOGON FOR: "MANAGER.SYS,PUB" ON LDEV #20.
HP3000  Release: C.75.00   User Version: C.75.00   MON, AUG  6, 2018,  8:42 PM
MPE/iX  HP31900 C.45.05  Copyright Hewlett-Packard 1987.  All rights reserved.
:
```

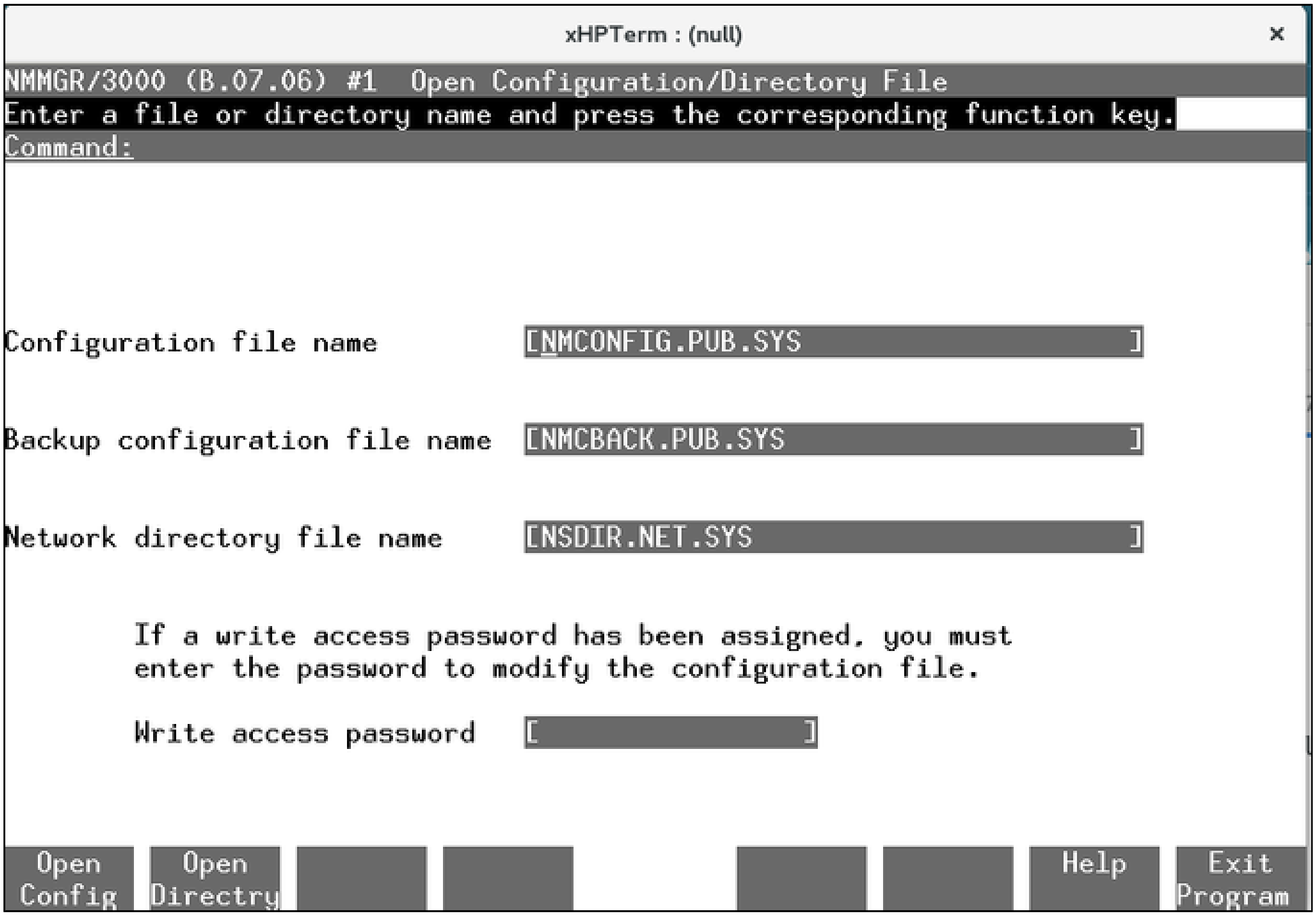
Please note: Line-editing is very limited. To delete a character on the command-line, use BACKSPACE. Note that the deleted character will still be visible until it is overwritten. Block-mode applications are not supported.

Using xhpterm

The **xhpterm** terminal emulation (an X Windows version of freevt3k) does not support many terminal escape sequences used by historic PA-RISC hardware for MPE/iX, but it can be used to run block-mode applications. It must be used on a serial line set up for raw mode.

Sample command (run on Charon host): `/opt/charon/bin/xhpterm -port 30002 -clean -font 10x20`

The following image shows an xhpterm window connected to the second serial line of an emulated MPE/iX system (running NMMGR):



```
NMMGR/3000 (B.07.06) #1  Open Configuration/Directory File
Enter a file or directory name and press the corresponding function key.
Command:




Configuration file name         [NMCONFIG.PUB.SYS                      ]


Backup configuration file name  [NMCBACK.PUB.SYS                       ]


Network directory file name     [NSDIR.NET.SYS                         ]


        If a write access password has been assigned, you must
        enter the password to modify the configuration file.

        Write access password   [                  ]
```

Working without Block-Mode Enter key

If working without a block-mode Enter key, you can navigate using the function keys of your keyboard. They correspond to the selection fields at the bottom of the screen.

For example, in the above window,

- **F1** would correspond to **Open Config**,
- **F8** would correspond to **Exit Program**.

Mapping the Block-Mode Enter key

The primary purpose for including this terminal emulator is to allow block-mode applications (like NMMGR) to be run from the Linux desktop. xhpterm uses the numeric keypad **Enter** key to simulate block-mode **Enter**. If your keyboard does not have a numeric key pad, the **Enter** key can be mapped to another key or key combination using **xmodmap** on Linux.

Example mapping the numeric key pad **Enter** key to **Right-Ctr**l:

	Step	Command

1	Determine the key code of the **Right-Ctr**l key.	**`$ xmodmap -pk | grep "Control_R"`**
		You will receive an output similar to:
		`105 0xffe4 (Control_R) 0x0000 (NoSymbol) 55 0xffe4 (Control_R)`

2	Redefine the key code to the numeric keypad **Enter** key.	**`$ xmodmap -e "keycode 105 = KP_Enter"`**
3	If original key mapping is required, reverse step 2.	**`$ xmodmap -e "keycode 105 = Control_R"`**

Please note: these settings are not persistent across X-sessions. To always use a certain setting when running **xhpterm**, you could, for example, use a small script to set and reset the key mapping. An example (only for illustrative purposes only) is shown below.

```
#!/usr/bin/bash
#
# parameter: -f <font> -p <port>
#
while getopts "f:p:" opt; do
    case "$opt" in
    (f) font="-font $OPTARG" ;;
    (p) port="-port $OPTARG" ;;
    (*) print "flag [$opt] unknown. abort." && exit 1 ;;
    esac
done

if [[ -z "$port" ]]; then
        echo "Port (-p) is mandatory"
        exit 1
fi

xmodmap -e "keycode 105 = KP_Enter"
/opt/charon/bin/xhpterm -clean $font $port
xmodmap -e "keycode 105 = Control_R"
```

Using Different Fonts

The default command-line to start xhpterm is: **`$ xhpterm -port 30000 -clean -font 10x20`**

If the 10x20 font is not found, you can try **`-font 6x13`** instead.

Additional fonts (e.g. the 9x15 font) can be installed using the command

```
# yum install xorg-x11-fonts-misc
```

Please note: When using xhpterm , avoid resizing the window - it can result in input and output fields in block-mode programs being misaligned. Changing the screen font will change the window size automatically, without causing this problem.

Accessing Emulated Charon-PAR/PA9 Systems via Terminal Emulation

Using Non-Graphical Telnet Client

In the absence of a graphical user environment, terminal emulators such as PuTTY and xhpterm cannot be used. In such cases, you can use the telnet command on the Charon host or from a remote system (if the serial line configuration permits it) to access the serial console port of the emulated PA3 system.

The basic command syntax is as follows:

```
$ telnet <charon-host> <console-tcp-port>
```

Parameters:

- *charon-host*: IP address or hostname of the Charon-PAR host system on which the emulator runs
- *console-tcp-port*: The TCP port configured for the serial console port in the configuration file of the emulated PA3 system

Please note: To make character deletion via the BACKSPACE key work on the command-line of HP-UX, you may have to enter the command`$ stty erase <backspace-key>` (and add it to your shell startup file, e.g., $HOME/.profile).

Using PuTTY

Stromasys provides preconfigured PuTTY profiles to access the serial console of an HP-UX system running on Charon-PAR. PuTTY does not support all control characters used by HP-UX, but many utilities (e.g., SAM) will work with some restrictions.

By default, the emulator configuration is set up to start PuTTY on the console port with a command similar to the following example:
`serial.uart0.device.command="putty -load PAR-Telnet-VT100"`.

To start an terminal emulator manually, perform the following steps:

	Step	Command / Action
1	Make sure the preconfigured PuTTY profiles are in the directory **.putty/sessions** or `.config/putty/sessions` in your home directory.	(see *Post Installation Tasks* in Charon-PAR Software Installation)
2	Start PuTTY from the command-line.	`$ putty -load PAR-Telnet-VT100`
3	Should you get an error that a font was not found, change the font.	Start PuTTY from the command-line. `$ putty` • Select and load the required profile. • Select **Fonts** on the left and select a monospace font. • Select **Session** on the left and save the profile.
4	To connect from a remote host, change localhost in the template to the correct hostname or IP address.	Start PuTTY from the command-line. `$ putty` • Select and load the required profile. • Change the hostname to the one of the Charon-PAR host as required. • Connect to the remote Charon-PAR system.

Instead of PuTTY, you can also use the standard telnet command on your local or remote Linux system (**`telnet`** *`<charon-par-host> <port>`*). However, Stromasys does not provide profiles for this. To make character deletion via the BACKSPACE key work on the command-line of HP-UX, you may have to use the command `$ stty erase <backspace-key>`.

Additional configuration steps to improve the behavior of the terminal emulation:

Step	Command
To enable better line drawing with the VT100 setting do the following on system where PuTTY runs	
PuTTY versions before 0.71	
Temporary set locale to **C** before starting PuTTY if you use a different locale.	`$ export LC_ALL=C` Then start PuTTY. This setting will enable proper line-drawing.
PuTTY versions starting with 0.71	
Use the new **UTF8linedraw** parameter in the PuTTY configuration.	Either add the line **UTF8linedraw=1** to your session configuration file or Start PuTTY from the command-line, load the **PAR-Telnet-VT100** session, enable VT100 line drawing for UTF-8 (the last parameter in **Window > Translation**), and save the session.
On the HP-UX guest system	

Set the terminal type to **vt100** after logging in to the system (this can be added to the login profile, if required).	`$ TERM=vt100`
	`$ export TERM`
	This will configure a terminal type known to HP-UX. Without a known terminal type many screen-based applications (e.g., editor, SAM) will not work.

Additional information: Cannot get correct line drawing on Linux/UNIX with Putty.

Other Terminal Emulation Considerations

Accessing the Emulated System Console Directly via telnet from a Remote System

This is not recommended due to the inherent security risk of using an unencrypted program across a potentially insecure network. This option is described for completeness.

Please note:

- The serial line port configuration must allow the remote connection.
- intermediate firewalls must allow the ports used.
- There can be only one connection to an emulated serial line at one time.

The following two options provide two examples of how to connect to the console of an emulated system from a remote system using **telnet**.

Option 1: using PuTTY installed on the remote system

If PuTTY is installed on the remote system, you can use the session profiles provided by Charon-PAR (copy them from the emulator host) to connect to the emulated system:

- Start PuTTY.
- Load the correct session profile (see sections above).
- Change **`localhost`** to the name or IP-address of the emulator host.
- Start the session.

Option 2: use the telnet program on the remote system

Run the following program from the command-line:

```
$ telnet <emulator-host> <serial-line-port>
```

`emulator-host` is the name or IP-address of the Linux emulator host.

`serial-line-port` is the port of the emulated serial line using the telnet protocol (default 30000).

To exit the session, use the telnet escape key sequence **`CTRL+}`** and type **`quit`**.

Running the telnet Terminal Emulation through an SSH Connection

Even though a plain telnet connection can be used to connect to the emulated system's console, this plaintext connection creates security problems. As an **alternative**, you run a telnet terminal emulation through an SSH connection using one of several options. The following sections show examples.

Please note: Any intermediate firewall must permit the port used for SSH (default 22).

Start Telnet from an Interactive SSH Login to the Charon Host

This is the easiest way to secure your connection:

- Login via SSH to the Charon host.
- Start a local telnet session from the command-line (sample uses the default console port):
  `telnet localhost 30000`

Display PuTTY via X11 through an SSH Tunnel

Prerequisite: X11 is enabled on the emulator host and the remote system.

Use the following steps to start a PuTTY session across an SSH tunnel:

	Step	Command
1	Connect to the emulator host via SSH specifying that X11 should be tunneled.	**$ ssh -X** *<user>@<emulator-host>*
2	Enter the password of *user* to log into the emulator host system.	
3	Start the terminal emulation program (e.g., PuTTY, assuming the Charon-PAR session configurations have been installed for the user).	**$ putty -load PAR-Telnet-VT100**

Use SSH Port-Forwarding to Create a Secure Connection

SSH port-forwarding creates a secure communication path for insecure protocols like telnet. The following example shows how to create such a basic tunnel on a Linux system for a serial console connection to a Charon-PAR emulator running on a remote Charon host:

- Step 1: initiate the port-forwarding connection:
  ssh -i ~/.ssh/*<my-private-key>* **-N -L50000:localhost:30000** *<user>@<charon-host>*

  In the local port 50000 is forwarded to the remote port 30000 (default console port) on the charon host system.
  The example assumes a password-less login via an SSH key-pair, but this is not mandatory.
  The parameter -N prevents an interactive session from being created. This is also not mandatory.
  Please refer to the SSH manpages of your Linux system for details on port-forwarding.
- Step 2: use the tunnel for your telnet connection:
  - Either from the command-line: **telnet localhost 50000**
  - or point your PuTTY telnet session to localhost port 50000

Guest System Autoboot

In many cases it is desirable to automatically start the guest OS when the emulator is started. This section will show the required steps in the following sections:

Please note: Charon-PAR can also be configured to start automatically as a daemon when the host system boots and to be stopped when the host system is shut down. However, such a configuration requires that the user shut down the guest OS cleanly before the emulator is stopped. Failure to do so can cause data corruption in the guest system. Contact your Stromasys representative for support if you require such a configuration.

Autoboot for Charon-PAR/PA9 Systems

By default, the newly installed guest HP-UX system is not booted automatically when the emulator starts. A Charon-PAR/PA9 system running HP-UX can be configured for autoboot in two different ways:

- Configuration from the PDC console.
- Configuration from HP-UX.

Please note: Autoboot will be interrupted if a configuration change occurs which causes reduced performance to allow manual intervention. This can be overridden by enabling **autostart** in addition to **autoboot** on the PDC console. However, this is not recommended as it is important to understand the cause of the problem.

Setting Autoboot on the Console

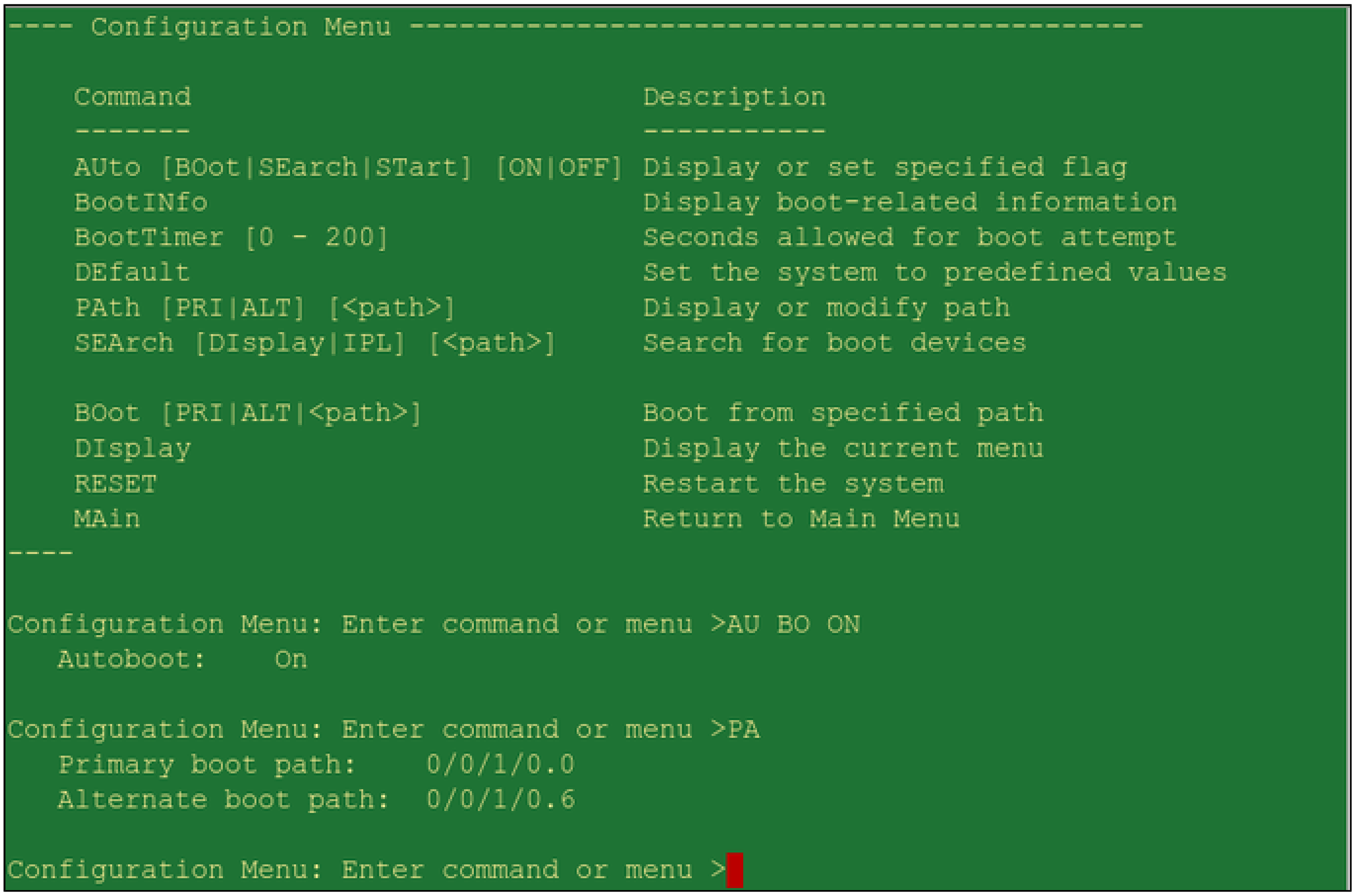
To enable autoboot from the Charon-PAR/PA9 PDC console before system boot, perform the following steps:

	Step	Description / Command
1	Start the emulator.	`# /opt/charon/bin/charon-par -f` *`</path-to/configuration-file>`*
2	Enter menu mode.	At the prompt **Main Menu: Enter command or menu >** type **menu**. This will display the available menu options.
3	Enter the configuration menu.	Type **`CO`** to enter the configuration menu. This will display the available configuration options.
4	Activate autoboot.	Type **`auto boot on`** to enable autoboot (or use the abbreviation **`au bo on`**). Autoboot is now enabled for the next boot.
5	Restart the emulator.	Stop the emulator: enter the **exit** command at the **pa9-64>** or **pa9-32>** prompt and hit **Enter**. Start the emulator again as shown in step 1.

To **interrupt autoboot**, press any key as soon as the system comes up. As there are only 5 seconds time to do this, this requires to have the PuTTY console started automatically.

To **disable autoboot**, follow the steps above, but use the command **`auto boot off`** instead.

The following image shows the configuration menu with autoboot set to on and the command to display the boot paths:



```
---- Configuration Menu ---------------------------------------------

    Command                            Description
    -------                            -----------
    AUto [BOot|SEarch|STart] [ON|OFF]  Display or set specified flag
    BootINfo                           Display boot-related information
    BootTimer [0 - 200]                Seconds allowed for boot attempt
    DEfault                            Set the system to predefined values
    PAth [PRI|ALT] [<path>]            Display or modify path
    SEArch [DIsplay|IPL] [<path>]      Search for boot devices

    BOot [PRI|ALT|<path>]              Boot from specified path
    DIsplay                            Display the current menu
    RESET                              Restart the system
    MAin                               Return to Main Menu
----

Configuration Menu: Enter command or menu >AU BO ON
   Autoboot:     On

Configuration Menu: Enter command or menu >PA
   Primary boot path:     0/0/1/0.0
   Alternate boot path:   0/0/1/0.6

Configuration Menu: Enter command or menu >
```

Setting Autoboot from HP-UX

Once the HP-UX system has booted, you can use the **setboot** command to enable or disable autoboot.

Syntax:

```
setboot -b on|off
```

Parameters:

- **on** enables autoboot
- **off** disables autoboot
- entering the command without parameters shows the current settings for autoboot and boot paths.

The following example shows the output of a **setboot** command without parameters:

```
# setboot
Primary bootpath : 0/0/1/0.0.0
Alternate bootpath : 0/0/1/0.6.0

Autoboot is ON (enabled)
Autosearch is OFF (disabled)
```

After enabling autoboot, shut down the guest system and restart the emulator. The system will boot automatically.

To **interrupt autoboot**, press any key as soon as the system comes up. As there are only 5 seconds time to do this, this requires to have the PuTTY console started automatically.

Autoboot for Charon-PAR/PA3 Systems

Setting a Charon-PAR/PA3 system with MPE/iX to autoboot requires two separate steps:

- Applying an autoboot file to the guest operating system that enables automatic execution of the **`start norecovery`** command.
- Enabling autoboot on the Charon-PAR/PA3 PDC console.

Applying an Autoboot File to the System

Enabling Autoboot normally involves first writing an SLT (System Load Tape) to tape, and then updating the system from tape. A Charon virtual tape can be used for this purpose. These instructions describe how to enable autoboot using a virtual tape. Charon is supplied with a preformatted autoboot file that contains "start norecovery". You can modify this file, or supply your own. Follow the steps below to create and apply the SLT:

	Step	Commands

1	Boot the system and log in as the Manager.
	Then copy the supplied autoboot file, and load the virtual tape file on LDev 7. LDev 7 is preconfigured in SYSGEN as device 0/0/1/0.6.0 (virtual tape MKA600 using container ldev7.img in the example).
	Wait for the message "`Vol (unlabelled) mounted on LDEV# 7`" to appear on the console, and continue.
		`:HELLO MANAGER.SYS`
		`:COPY AUTOBOOT.CHARON, AUTOBOOT.PUB`
		`:DEVCTRL 7;LOAD=ONLINE`

2	Use **sysgen** to specify the new autoboot file and write the SLT.
	The line with the ? is a prompt asking for the tape number and a PIN. The PIN is represented in the sample here by the string XXX. Use the number shown in place of the XXX.
		`:SYSGEN`
		`sysgen> sysfile`
		`sysfile> aauto autoboot.pub.sys type=disc`
		`sysfile> hold`
		`sysfile> exit`
		`sysgen> tape`
		`?19:00/#S2/XXX/ LDEV# FOR "SYSGTAPE" ON TAPE (NUM)?`

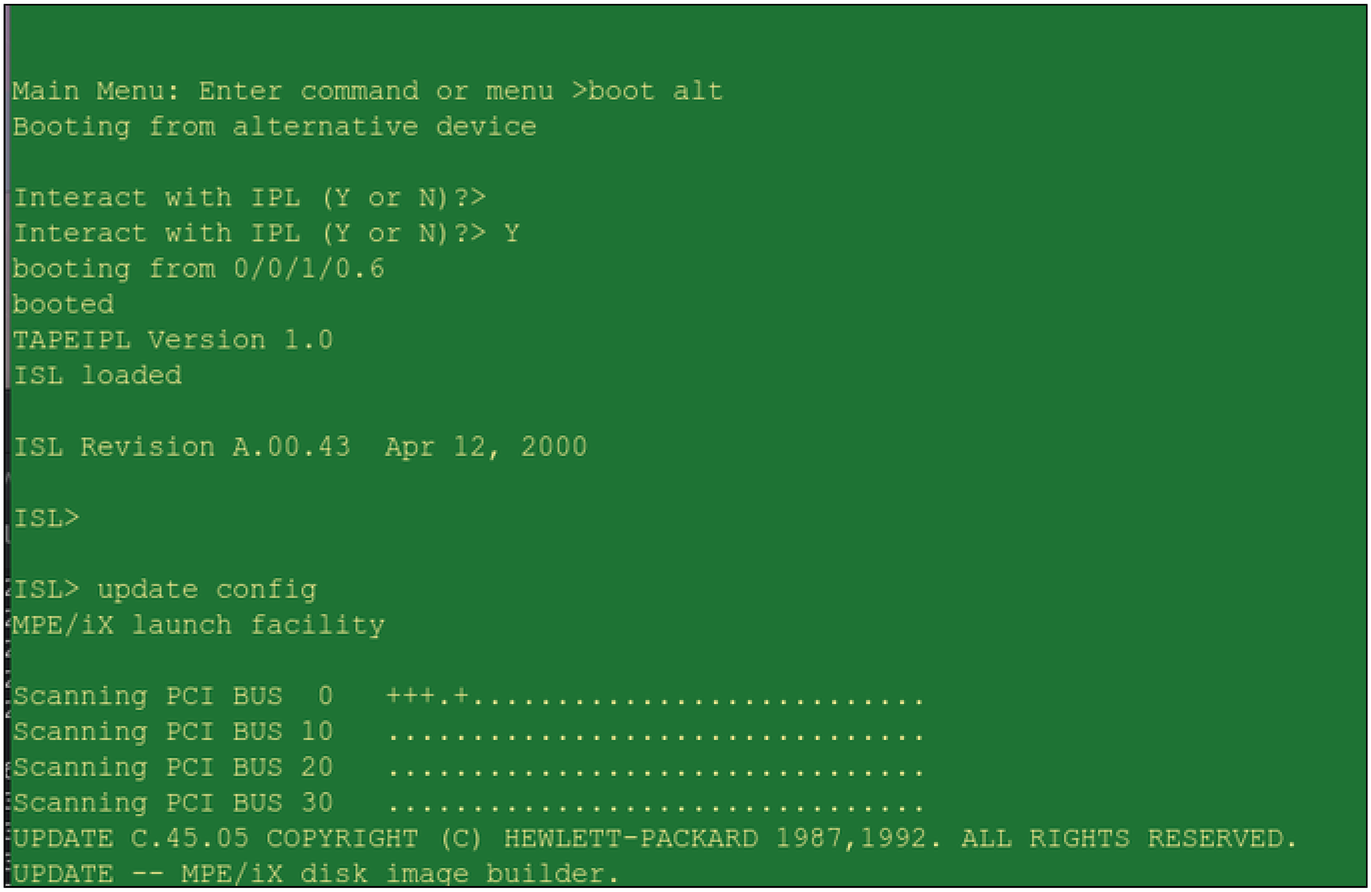
3	Respond to the prompt.	• Press **CTRL-A**. This will display an = character. • Type **`REPLY XXX,7`** after the = character replacing XXX with the number displayed in the prompt. • Confirm with the **Enter** key.
4	Wait for the message "`**boot tape is successfully built**`" before you continue.	

5	Exist **sysgen**, respond **yes** to the question if you really want to exit, and then shut down the system.	**`sysgen> exit`**
		`(...)`
		`Configuration changes are not kept yet!`
		`Still want to exit (yes/no) ?` **`yes`**
		`: shutdown system`

6	Stop and restart the emulator.	To stop, type **exit** at the Charon-PAR **pa3>** console prompt. Then restart the emulator.
7	Reboot the system from the virtual tape on LDev 7. Then update the system with the files from the virtual tape.	• At the prompt **`Main Menu: Enter command or menu`** type **boot alt** (the default alternate boot path is LDev 7; if you change it, boot using the device path of the tape). • At the prompt **Interact with IPL (Y or N)?** type **Y**. • At the ISL prompt type **`update config`**.
8	The system will display the list of files that are updated.	
9	Boot normally. IPL interaction should not be needed.	• At the prompt **Main Menu: Enter command or menu** type **`boot`**. • At the prompt **`Interact with IPL (Y or N)?`** type **`N`**.

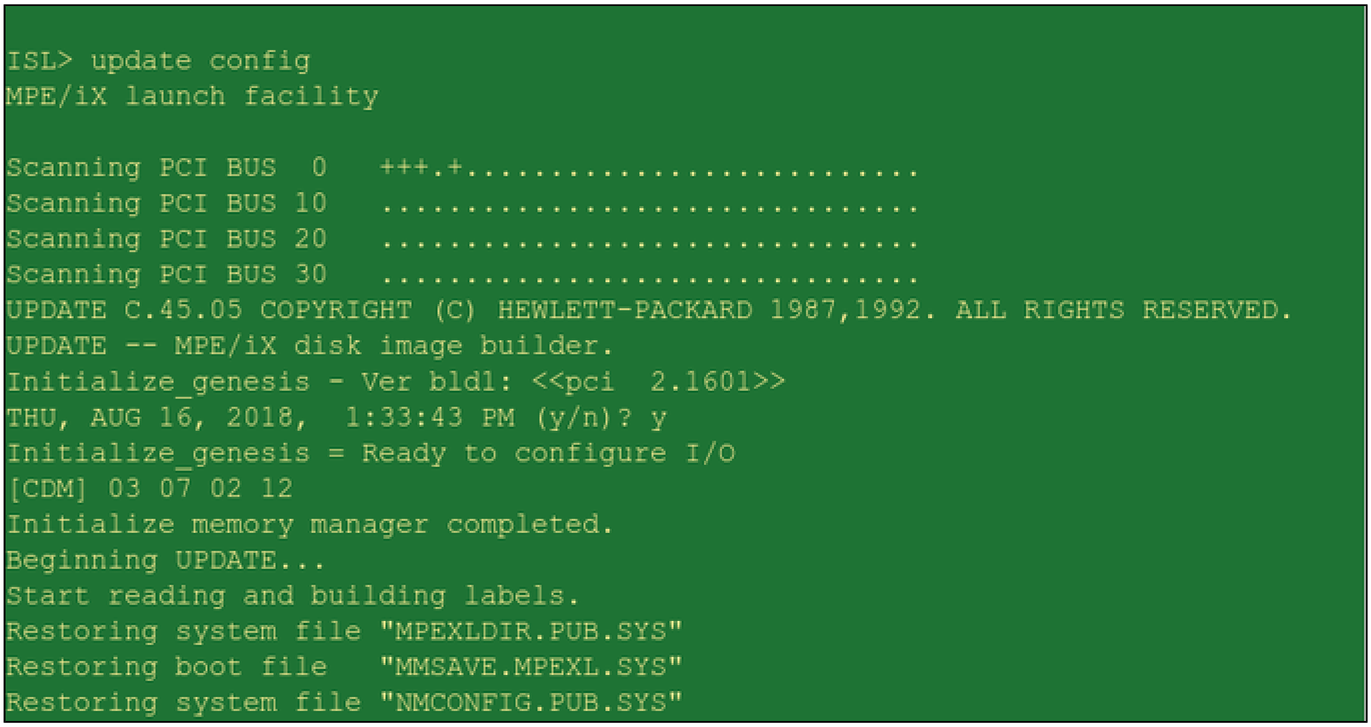
Please note: After this step, the virtual tape container (ldev7.img in the example) can be removed or archived. It is no longer needed.

The following image illustrates the **boot alt** and **update config** command sequence of step 7 above:



```
Main Menu: Enter command or menu >boot alt
Booting from alternative device

Interact with IPL (Y or N)?>
Interact with IPL (Y or N)?> Y
booting from 0/0/1/0.6
booted
TAPEIPL Version 1.0
ISL loaded

ISL Revision A.00.43  Apr 12, 2000

ISL>

ISL> update config
MPE/iX launch facility

Scanning PCI BUS  0   +++.+.........................
Scanning PCI BUS 10   ..............................
Scanning PCI BUS 20   ..............................
Scanning PCI BUS 30   ..............................
UPDATE C.45.05 COPYRIGHT (C) HEWLETT-PACKARD 1987,1992. ALL RIGHTS RESERVED.
UPDATE -- MPE/iX disk image builder.
```

The following image illustrates a part of the system file application stage (step 8 above):



```
ISL> update config
MPE/iX launch facility

Scanning PCI BUS  0   +++.+...........................
Scanning PCI BUS 10   ................................
Scanning PCI BUS 20   ................................
Scanning PCI BUS 30   ................................
UPDATE C.45.05 COPYRIGHT (C) HEWLETT-PACKARD 1987,1992. ALL RIGHTS RESERVED.
UPDATE -- MPE/iX disk image builder.
Initialize_genesis - Ver bldl: <<pci  2.1601>>
THU, AUG 16, 2018,  1:33:43 PM (y/n)? y
Initialize_genesis = Ready to configure I/O
[CDM] 03 07 02 12
Initialize memory manager completed.
Beginning UPDATE...
Start reading and building labels.
Restoring system file "MPEXLDIR.PUB.SYS"
Restoring boot file   "MMSAVE.MPEXL.SYS"
Restoring system file "NMCONFIG.PUB.SYS"
```

Enabling Autoboot on the PDC Console

To enable autoboot from the Charon-PAR/PA3 PDC console before system boot, perform the following steps:

	Step	Description / Command
1	(Re)start the emulator (shut down the guest OS and emulator before, as needed).	`# /opt/charon/bin/charon-par -f </path-to/configuration-file>`
2	Enter menu mode.	At the prompt **Main Menu: Enter command or menu >** type **menu**. This will display the available menu options.
3	Enter the configuration menu.	Type **`CO`** to enter the configuration menu. This will display the available configuration options.
4	Activate autoboot.	Type **`auto boot on`** to enable autoboot (or use the abbreviation **`au bo on`**). Autoboot is now enabled for the next boot.
5	Restart the emulator.	Stop the emulator: enter the **exit** command at the **pa3>** prompt and hit **Enter**. Start the emulator again as shown in step 1.

To **interrupt autoboot**, press any key as soon as the system comes up. As there are only 5 seconds time to do this, this requires to have the PuTTY console started automatically.

To **disable autoboot**, follow the steps above, but use the command **`auto boot off`** instead.

Running Charon-PAR as a Service

Contents

General Information

Charon-PAR can be started as a service manually and automatically then the Charon host system boots. The service can be stopped manually or automatically when the host system shuts down.

Please note: each instance of Charon-PAR must be started in a separate directory because certain files (SSTORAGE.DAT, NVOLATILE.DAT, UA_SSTORAGE.DAT, and console.dat) will always be created in the current working directory of the emulator instance. Starting multiple emulator instances in the same directory will lead to errors due to file locking problems.

Important points to note when when starting and stopping a Charon-PAR service:

- Only start Charon-PAR and the guest system automatically, if the guest system and guest system applications can be brought up in unattended mode. Before setting up a service for automatic start, review your system to verify it is suitable for this type of configuration.
- Stopping a Charon-PAR service **does not include a clean shut-down** of the guest operating system. The guest operating system must be shut down cleanly before the Charon-PAR service is stopped because stopping the service is the equivalent of pulling the power plug. Failing to shut down the operating system cleanly can lead to **data corruption, data loss, and even an unbootable guest system**. Contact Stromasys or your Stromasys VAR if you require additional information.
- Only activate Charon-PAR as a service during system boot once the correct functioning of all guest system services has been tested manually and a procedure for a clean guest system shutdown has been tested and put in place.
- When starting Charon-PAR as a service automatic boot of the guest operating system may also be required. Please refer to Guest System Autoboot for more information.
- When starting Charon-PAR as a services, you have to manually start the terminal emulator to connect to the console serial port (e.g., PuTTY).

Configuring a systemd Service

The Charon-PAR installation kit contains a sample **systemd** service file. This file can be modified and installed as a systemd service.

Installing and Testing a Charon-PAR systemd Service

- Ensure that all paths in your Charon-PAR file are absolute paths.
- Test your Charon-PAR configuration manually until satisfied that it is working correctly and your guest operating system services start as expected.
- Copy **/opt/charon/bin/charon-par.service** to a local file. For example: *<guest-name>*.**service**
- Adapt the service file to contain the correct paths to the Charon-PAR executable (/opt/charon/bin/charon-par) and the Charon-PAR configuration file. If you edit the file **after** it has been copied to the **systemd** directory, use **`systemctl daemon-reload`** to activate the changes.
- Copy the service file to **/etc/systemd/system**.
- Test a manual start of the service:
  ```
  # systemctl start <guest-name>
  ```
- Check the status of the service:
  ```
  # systemctl status <guest-name>
  ```
- Shut down the guest operating system if running.
- Test a manual stop of the service:
  ```
  # systemctl stop <guest-name>
  ```
- If everything worked as expected, enable the service for automatic startup:
  ```
  # systemctl enable <guest-name>
  ```

Please note: The console log output of a service is redirected to **journalctl**.

The following example shows the content of a service file, starting the service, and checking the status:

Sample service file

```
$ cat myguest.service

[Unit]
Description="Charon emulator service"
After=syslog.target network.service aksusbd.service
Wants=aksusbd.service

[Service]
Type=forking
ControlGroup=cpu:/
WorkingDirectory=/opt/charon/log
PIDFile=/var/run/myguest.pid
ExecStart=/opt/charon/bin/charon-par -f /Stromasys/Charon-PAR/myguest/myguest.cfg --daemon --pidfile /var/run
/myguest.pid

[Install]
WantedBy=multi-user.target
```

Starting the service

```
# systemctl start myguest
```

Service status

```
$ systemctl status myguest

 myguest.service - "Charon emulator service"
Loaded: loaded (/etc/systemd/system/myguest.service; disabled; vendor preset: d>
Active: active (running) since Fri 2020-03-20 12:03:09 CET; 17s ago
Process: 514524 ExecStart=/opt/charon/bin/charon-par -f /Stromasys/Charon-PAR/myguest/>
Main PID: 514528 (charon-par)
Tasks: 78 (limit: 76997)
Memory: 4.2G
CPU: 1.612s
CGroup: /system.slice/myguest.service
514528 /opt/charon/bin/charon-par -f /Stromasys/Charon-PAR/myguest/myguest.cfg
```

<lines removed>

Stopping and Disabling a systemd Service

To stop and disable the Charon-PAR systemd service perform the following steps:

- Shut down the guest operating system if running.
- Stop the service:
  `# systemctl stop <guest-name>`
- Disable the service:
  `# systemctl disable <guest-name>`

Adding Additional Disks

Adding disk space to the Charon-PAR emulated guest systems systems is a process consisting of several steps:

- Build disk container files in the Linux host file system.
- Link these disk container files with the corresponding emulator configuration paths in the Charon configuration file.
- Enable the disk for use by the guest operating system.
- For Charon-PAR/PA3:
    - Define virtual disk devices using SYSGEN on MPE/iX.
    - Create the virtual volume using VOLUTIL on MPE/iX.

There is no direct correspondence between the number of virtual disks that are added to the emulated system and the number of Linux disks on which they reside. The storage on the host system just must provide enough capacity to store the disk containers used as virtual disks by the emulated system.

The sections below describe the process in more detail.

Please note that it is **not recommended** to place emulator storage devices (in particular vdisks) on NFS as this will have a significant impact on performance. However, if any of the storage (e.g., ISO files or vdisks) is on an NFS share, NFS locking must be enabled and all intermediate firewalls between client and server must allow the port used by the **lockd** and **statd**. Failure to do so will cause the emulator to hang at startup.

Creating Disk Container Files

The disk container files are created using the **dd** command. This command takes an **input file** name, an **output file** name, a **block size**, and a **count** as parameters. When instructed to read from **/dev/zero** , the command creates the output file and fills it with (block-size * count) zero bytes. Each file may be built as large as needed.

Please note: The maximum disk size supported by MPE/iX is 512GB. The maximum size for HP-UX is 2TB.

Use the following steps to create a new disk container file:

	Step	Command
1	Change to the directory where you store your virtual disks.	Example: `# cd /data/Stromasys/host1/Data/Disk/`
2	Create an empty disk container. Note that the actual size of the disk will depend on the requirements of your emulated system.	Example virtual disk creation (size 20GB): `# dd if=/dev/zero of=mydiskname.dsk bs=1G count=20`

Adding a Virtual Disk to the Configuration File

The new virtual disk must be added to the configuration file of the Charon-PAR instance. This requires two commands:

	Step	Configuration file entry
1	Load the SCSI device.	`load DKXnnn`
2	Link device with disk container file.	`DKXnnn.image="/path/to/disk-container-file"`

The device name **DK***Xnnn* encodes the **device type** and the **device path** of the emulated SCSI device presented to the guest operating system:

A disk device name has the following components:

- Value **DK** at the beginning of the device name: specifies that device type is disk.
- **X**: value consists of an uppercase letter encoding the device path. Please refer to Emulated Model Hardware Configuration Details and the relevant configuration file templates for system specific device path information.
  Example (rp4000):
  A = device path 0/0/1/0
  B = device path 0/0/1/1
  C = device path 0/0/2/0
  D = device path 0/0/2/1
- **nnn**: value encodes the SCSI device connected to the SCSI controller.
  Formula: (SCSI target ID*100)+LUN

The following example adds a SCSI disk with device path 0/0/1/0.0.0 to the configuration:

```
load DKA0

DKA0.image="ldev1.dsk"
```

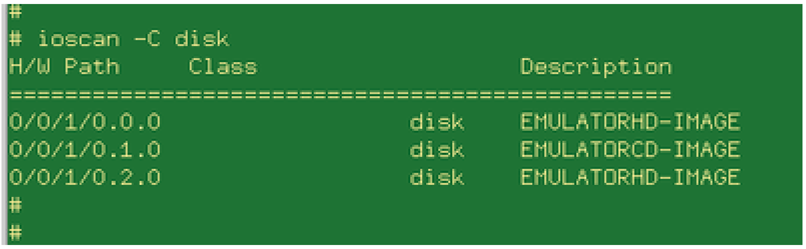
Start Using the Disk in the Guest OS

Before the disk can be used by guest operating system applications, there are some required steps that need to be performed on the guest OS. The following examples are only used for illustrative purposes. They are intended to provide a general impression of the necessary actions.

Enable a Disk for Use by HP-UX

After booting the emulated system with the new disk added to the configuration file, check if HP-UX did recognize it:

Command	Example

To list all known disks on the system, enter the following command:
`# ioscan -C disk`
	The disk with device path 0/0/1/0.2.0 (DKA200) is the new disk:
	
	`#`
	`# ioscan -C disk`
	`H/W Path     Class                       Description`
	`==================================================`
	`0/0/1/0.0.0                    disk     EMULATORHD-IMAGE`
	`0/0/1/0.1.0                    disk     EMULATORCD-IMAGE`
	`0/0/1/0.2.0                    disk     EMULATORHD-IMAGE`
	`#`
	`#`

HP-UX Device Nodes

During HP-UX boot-up, **/sbin/insf** is executed to create the character and block device special */dev* files that allow communication with the disk. HP-UX uses the following device file naming system. For each disk device, the following special files are created:

- Block device file:
  /dev/dsk/c<*number-of-card*>**t**<*SCSI-target-ID*>**d**<*device-LUN*>
- Character device file:
  /dev/rdsk/c<*number-of-card*>**t**<*SCSI-target-ID*>**d**<*device-LUN*>

Where card is the (SCSI) controller number, target is the SCSI ID number, and device is the logical unit number, or LUN. The LUN is 0 for the majority of devices. An example for a disk at controller 0, target 2, LUN 0, is */dev/dsk/c0t2d0*.

A disk is usually divided into file systems, areas that can hold files, or raw data areas such as swap. File systems are created in disk partitions or logical volumes. To view the size of the drive to be configured use the command, *diskinfo*.

Example:

```
# diskinfo /dev/rdsk/c0t2d0

SCSI describe of /dev/rdsk/c0t2d0:

     vendor: EMULATOR

     product id: HD-IMAGE

     type: direct access

     size: 2097152 Kbytes

     bytes per sector: 512
```

Creating and Mounting a Filesystem on the new Disk

On HP-UX disks are typically used as part of LVM volume groups. Alternatively, a disk can be formatted using the traditional **HFS** filesystem. LVM logical volumes can be used with **HFS** or **JFS** filesystems. JFS is HP's version of the Veritas journaling filesystem. Below, two samples show the two different approaches.

Please refer to your system's documentation and the man-pages for detailed information about the commands used in the examples.

HFS filesystem on standalone disk

The following example, which shows how to create a filesystem and mount the disk, will use **HFS** on a standalone disk:

	Step	Command
1	Create a filesystem on the new disk. Please note: LVM is now the preferred method to use disks on HP-UX. This example uses a traditional filesystem for simplicity.	Example to create a hfs filesystem on DKA200: `# newfs -F hfs /dev/rdsk/c0t2d0`
2	Mount the disk to a mount point (create mount point first if it does not exist yet).	Example to mount DKA200 on /mnt: `# mount /dev/dsk/c0t2d0 /mnt`

LVM volume group with JFS file system

	Step	Command
1	Create a physical (non-bootable) volume.	`# pvcreate -f /dev/rdsk/c0t0d0`

2	Create the volume group file structure.	Example to create structure for volume group 1:
		`# mkdir /dev/vg01`
		`# mknod /dev/vg01/group c 64 0x010000`
		0x*NN*0000 is the minor device number.
		To identify all minor device numbers already used, run the command
		`# ll /dev/*/group`

3	Create volume group.	Example to create volume group 1 with physical volume created above: `# vgcreate vg01 /dev/dsk/c0t0d0`
4	Create logical volume.	Example to create a logical volume named *myname* of *size-MB* megabytes: `# lvcreate -L size-MB -n myname vg01`
5	Create filesystem on logical volume.	Example to create a JFS (vxfs) filesystem on logical volume *myname*: `# newfs -F vxfs /dev/vg01/rmyname`
6	Mount filesystem to a mount point (create mount point first if it does not exist yet).	Example to mount the new filesystem on /mnt: `# mount -F vxfs /dev/vg01/myname /mnt`

Enable a Disk for Use by MPE/iX

After booting the emulated system with the new disk added to the configuration file, it needs to be enabled for use by the operating system as described below.

Adding the Device via Sysgen

After booting the MPE/iX guest system in the emulator, add the device using SYSGEN; specify ID=ST19171N and the path that you specified in the configuration file **load** command in the previous step. For example:

`PATH=0/0/1/0.1.0` for LDEV 2 (specified by "load DKA100" in pa3.cfg)
`PATH=0/0/1/0.2.0` for LDEV 3 (specified by "load DKA200" in pa3.cfg), etc.

This example shows one way to add LDEV 2:

	Step	Commands
1	Log in to **Manager.sys**.	`:HELLO MANAGER.SYS`
2	Start **sysgen**.	`:SYSGEN`

3	Enter the following commands.	`sysgen> io`
		`io> adev 2 id=ST19171N path=0/0/1/0.1.0`
		`io> hold`
		`io> exit`
		`sysgen> keep`
		`keeping to group CONFIG.SYS`
		`Purge old configuration (yes/no)? yes`
		`sysgen>`

4	Exit **sysgen**.	`sysgen> exit`
5	Activate the configuration.	`:DOIONOW`

When using the SYSGEN **adev** command, depending on the path chosen and your current configuration, you may first need to create parent controller entries using the SYSGEN **apath** command. Each disk device is created beneath PCI_DEVICE , A5150A , and PSEUDO path entries as shown in the following example:

```
apath id=PCI_DEVICE path=0/0/<device>

apath id=A5150A path=0/0/<device>/<function>

apath id=PSEUDO path=0/0/<device>/<function>.<target>

adev <ldev number> id=ST19171N path=0/0/<device>/<function>.<target>.<lun>
```

You do not need to indicate to SYSGEN how large a disk is. Use the ST19171N device ID whatever size your disk image is; MPE will automatically detect (and use) the size of the physical disk image file.

Creating an MPE Volume

The final stage involves creating the MPE volume (and, if needed, volume set) and initializing the disk. When assigning new volume names, using one of these conventions is recommended:

- Name each volume, starting with the master, as MEMBERx , where ' x ' starts at ' 1 ' for the master volume and is incremented for each additional volume in the set.
- Name each volume MEMBERxxx , where ' xxx ' is the LDev number of the disk

The command **:VOLUTIL** should be used to create and initialize new volumes, and also (if needed) new volume sets, as shown in the following example. You will need to be logged in as MANAGER.SYS .

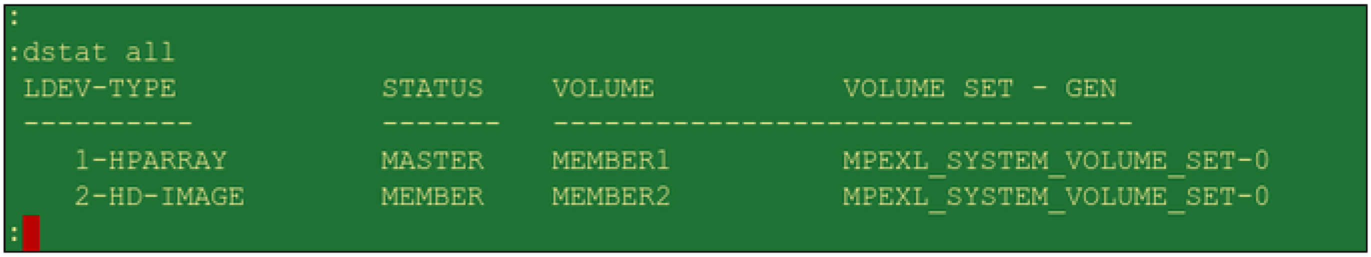
The following example shows how to add a new LDev 2 named MEMBER2 to the System Volume Set:

	Step	Command
1	Start **volutil**.	`:VOLUTIL`
2	At the voltutil prompt add the new disk as a volume to the system volume set.	`volutil: newvol MPEXL_SYSTEM_VOLUME_SET:MEMBER2 2 100 100`
3	Exit **volutil**.	`volutil: exit`

The following example shows how to create a new volume set with LDev 2 as the master:

	Step	Command
1	Start **volutil**.	`:VOLUTIL`
2	At the voltutil prompt create a new volume set with the new disk as the master volume.	`volutil: newset USERSET MASTER 2 100 100`
3	Exit **volutil**.	`volutil: exit`

To display the status of the volumes, use the **dstat** command as shown below:



```
:
:dstat all
 LDEV-TYPE                STATUS     VOLUME               VOLUME SET - GEN
 ----------               -------    ----------------------------------
    1-HPARRAY             MASTER     MEMBER1              MPEXL_SYSTEM_VOLUME_SET-0
    2-HD-IMAGE            MEMBER     MEMBER2              MPEXL_SYSTEM_VOLUME_SET-0
:
```

Physical and Virtual Tape Access

Adding tape devices to the systems emulated by Charon-PAR is a process consisting of several steps:

- Link the tape container files and/or physical tape devices with the corresponding configuration paths in the Charon configuration file and load the devices.
- Enable the tape for use by the guest operating system.

The storage on the host system must provide enough capacity to store the tape containers used as virtual tapes by the emulated system.

Please note: physical tapes are not supported in cloud environments.

The sections below describe the process in more detail.

Adding Tape Devices to the Configuration File

The new emulated tape device must be added to the configuration file of the Charon-PAR instance. This requires two commands:

	Step	Configuration file entry
1	Load the SCSI device.	`load MKXnnn`

2	Link the device with tape container file
	
	or
	
	physical tape device
		`MKXnnn.image="/path/to/tape-container-file"`
		The path can point to an existing tape container file, but this is not mandatory. The emulator can create the container file if required.
		`MKXnnn.image="/dev/stN"`
		You can identify the Linux tape device name using the command **`dmesg |grep -i tape`**.
		N stands for the device number, e.g., /dev/st0.

3	Enable loading the virtual tape automatically (for virtual tapes based on container file).	`MKXnnn.autoload=yes` **Please note**: • Multi-volume backup software may not work correctly when autoload is enabled. If a new tape is requested by the software, it may not wait for a new tape to load but overwrite the existing file. If data is must be written to several tapes, this needs to be implemented manually, e.g., by a customized script. • Without autoload enabled, MPE/iX can load a tape using the DEVCTRL command. There is no such option for HP-UX guest systems. The autoload command can help to overcome this deficiency. Please note the note above!

The parameter **MK***Xnnn* encodes the **device type** and the **device path** of the emulated SCSI device presented to the guest operating system:

A tape device name has the following components:

- Value **MK** at the beginning of the device name: specifies that device type is tape.
- **X**: value consists of an uppercase letter encoding the device path. Please refer to Emulated Model Hardware Configuration Details for system specific device path information.
  Example (rp4000):
  A = device path 0/0/1/0
  B = device path 0/0/1/1
  C = device path 0/0/2/0
  D = device path 0/0/2/1
- **nnn**: value encodes the SCSI device connected to the SCSI controller.
  Formula: (SCSI target ID*100)+LUN
  The LUN is always 0 for tape devices.

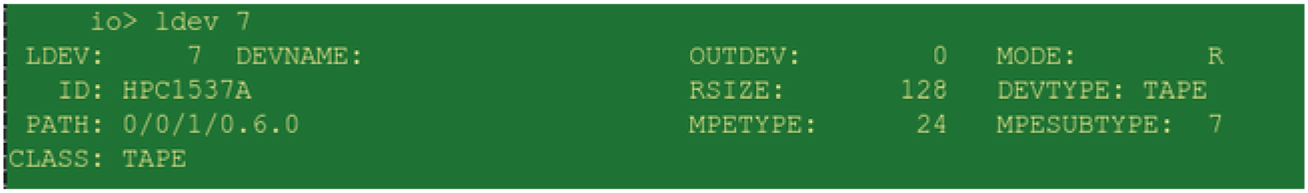
The following example adds a SCSI tape device with device path 0/0/1/0.6.0 (controller A, SCSI target ID 6, LUN 0) to the configuration:

```
load MKA600

MKA600.image="ldev7.img"
```

Using Tapes on MPE/iX Guest Systems

Displaying Tape Characteristics on MPE/iX Systems

You can use the SYSGEN > IO to display tape characteristics as shown in the following example for LDev 7:



```
     io> ldev 7
 LDEV:      7   DEVNAME:                   OUTDEV:        0    MODE:         R
   ID: HPC1537A                            RSIZE:       128    DEVTYPE: TAPE
 PATH: 0/0/1/0.6.0                         MPETYPE:      24    MPESUBTYPE:   7
CLASS: TAPE
```

Using :DEVCTRL to Load and Eject a Tape on MPE/iX Systems

:DEVCTRL is an script that allows several aspects of physical tape devices to be controlled: in-drive data compression, tape loading, and automatic tape eject. This script is also used with Charon-PAR to load virtual tape image files. When :DEVCTRL is used to load a virtual tape file, it causes Charon-PAR to create (if necessary) and open the virtual tape file configured in the Charon-PAR configuration file.

The :DEVCTRL syntax is show below:

```
DEVCTRL [DEV=] LDev [COMPRESSION= ENABLE | DISABLE | NOCHANGE ]
                    [EJECT= ENABLE | DISABLE | NOCHANGE]
                    [LOAD= ONLINE | OFFLINE | NOCHANGE]
```

For example, to load an emulated type device on LDev 7, use the following command:

```
:DEVCTRL 7; LOAD=ONLINE
```

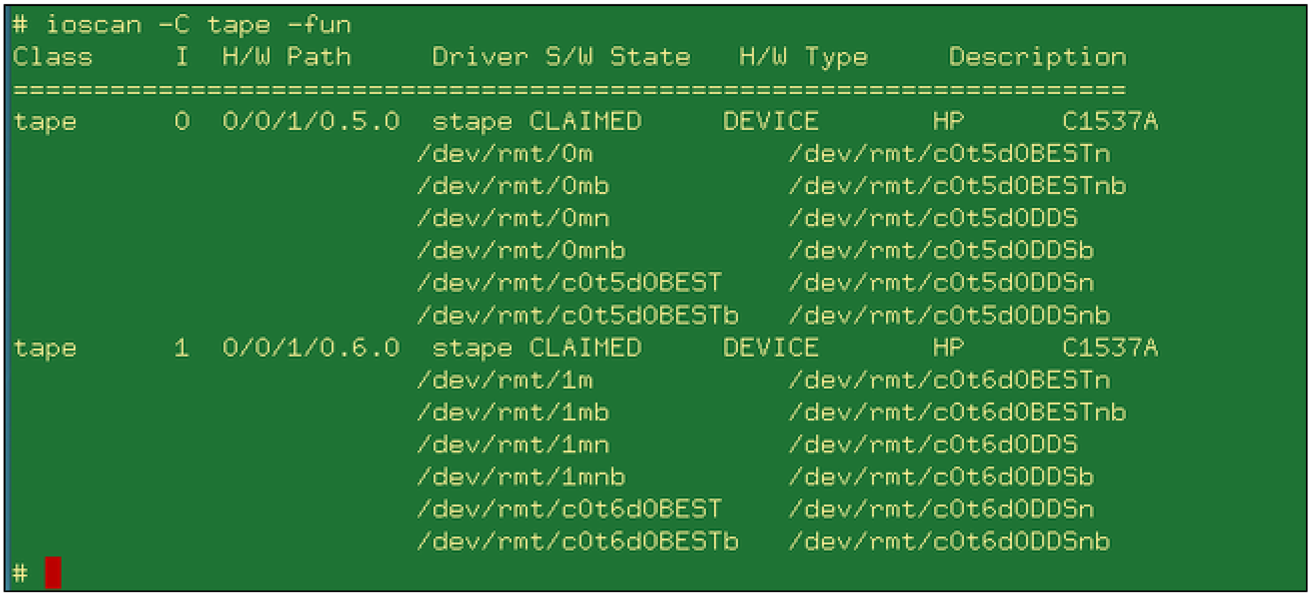
Reading and Writing Tapes in MPE/iX

To read and write tapes under MPE/iX, the :STORE and :RESTORE commands are used. Please contact your Stromasys representative or partner if you need support with data backup and restore under MPE/iX.

Using Tapes on HP-UX Guest Systems

Displaying the Tape Configuration in HP-UX

After booting HP-UX with the virtual tape device configuration, you can use the command **`ioscan -C tape -fun`** to verify if the device was found and the driver was loaded. The following image shows an example:



```
# ioscan -C tape -fun
Class     I  H/W Path     Driver S/W State   H/W Type     Description
=====================================================================
tape      0  0/0/1/0.5.0  stape CLAIMED     DEVICE       HP       C1537A
                         /dev/rmt/0m                /dev/rmt/c0t5d0BESTn
                         /dev/rmt/0mb               /dev/rmt/c0t5d0BESTnb
                         /dev/rmt/0mn               /dev/rmt/c0t5d0DDS
                         /dev/rmt/0mnb              /dev/rmt/c0t5d0DDSb
                         /dev/rmt/c0t5d0BEST        /dev/rmt/c0t5d0DDSn
                         /dev/rmt/c0t5d0BESTb       /dev/rmt/c0t5d0DDSnb
tape      1  0/0/1/0.6.0  stape CLAIMED     DEVICE       HP       C1537A
                         /dev/rmt/1m                /dev/rmt/c0t6d0BESTn
                         /dev/rmt/1mb               /dev/rmt/c0t6d0BESTnb
                         /dev/rmt/1mn               /dev/rmt/c0t6d0DDS
                         /dev/rmt/1mnb              /dev/rmt/c0t6d0DDSb
                         /dev/rmt/c0t6d0BEST        /dev/rmt/c0t6d0DDSn
                         /dev/rmt/c0t6d0BESTb       /dev/rmt/c0t6d0DDSnb
#
```

The example shows two type devices and their associated special devices.

The **`lssf`** command can be used to verify that the special files point to the correct device paths, SCSI target ID and LUN.

```
# lssf /dev/rmt/0m

stape card instance 0 SCSI target 5 SCSI LUN 0 at&t best density
available at address 0/0/1/0.5.0 /dev/rmt/0m
```

Loading and Ejecting the Virtual Tape in HP-UX

Using the autoload Configuration Option

If **autoload** has been enabled in the configuration, the virtual tape is "loaded" and the container file created, if required, when the tape device is accessed.

Please note: Multi-volume backup software may not work correctly when autoload is enabled. If a new tape is requested by the software, it will not wait for a new tape to load but overwrite the existing file. If data is to be written to several tapes, this needs to be implemented manually (e.g., by a customized script).

For example, the following command will cause the container file for device **/dev/rmt/0mnb** (default tape device) to be created and opened:

```
# mt status
Drive: HP C1537A
Format:
Status: [41114701] BOT online compression immediate-report-mode
File: 0
Block: 0
```

The tape can be "ejected" by setting it to offline (*device-name* is the full path to the special file):

```
# mt -f <device-name> offl
```

After the offline command, the container file can be removed (i.e., the tape can be archived).

To simulate the "swapping" of tapes, use the sequence:

- Release a loaded tape so its container file can be moved away: **`mt -f`** `<`*`devicename`*`>` **`offl`**
- Write to the tape as required (this will create a new container file, if needed).
- Release the loaded tape so its container file can be moved away: **`mt -f`** `<`*`devicename`*`>` **`offl`**

Using Manual Tape Loading from the PA9 Console

In some cases it may be useful to load/unload tapes manually from the **pa9-64** or **pa9-32** console instead of using autoload. This is achieved by the following command (using PA9-64 as an example):

Load a tape:

```
pa9-64> MKXnnn.load
```

Unload a tape:

```
pa9-64> MKXnnn.unload
```

Where **MK***Xnnn* stands for the virtual tape configuration name, for example, MKA500.

If no tape is "loaded" in the virtual tape device, the command

```
    mt -f <device-name> status
```

will show a status of 0.

Reading and Writing Tapes in HP-UX

There are several commands that can be used to write data to / read data from the virtual tapes, for example,

- tar
- fbackup/frecover
- dump/restore
- vxdump/vxrestore

Please refer to your HP-UX documentation for details about the usage of these utilities.

Adding Generic SCSI Devices

When adding **MK***Xnnn* and **DK***Xnnn* SCSI devices to the configuration of the emulated system, Charon-PAR uses the Linux SCSI disk and tape interface. The emulator provides an emulated disk or tape device to the guest system and translates between this emulated device and the Linux device interface.

On the other hand, when adding a generic SCSI device (**GK***Xnnn*) to the configuration of the emulated system, Charon-PAR passes the SCSI commands of the guest system through to the generic Linux SCSI device (/dev/sg*X*). The guest system communicates directly with the device. This may be helpful if the device used is not a standard disk or tape device.

Please note:

- Only a disk or tape generic SCSI device can be used in the Charon-PAR configuration.
- It is not certain that a generic SCSI device will work together with the guest operating system. This depends on the compatibility of the guest system SCSI commands and the capabilities of the device.

In contrast to **MK** and **DK** devices, the generic device configuration uses **`devname`** instead of **`image`** to specify the linked Linux device.

The following example adds SCSI generic device **/dev/sg0** with device path 0/0/2/0.6.0 (controller C, SCSI target ID 6, LUN 0) to the configuration:

```
load GKC600

GKC600.devname="/dev/sg0"
```

HP-UX Graphical Connection via X-Server

This section describes two methods to run graphical applications on the HP-UX guest system by using an X-Server installed on the emulator host system or a remote system:

- Xsession started via ssh
- Xsession started via XCMCP

The following points are discussed in more detail:

X-Servers

To apply the two methods described, the remote system needs to run an X-Server. For the examples in this section two free applications are used:

- Linux: Xephyr, a nested X-Server able to run on an existing graphical desktop.
- Microsoft Windows: Xming.

There are several other free and commercial X-Servers. To describe all of them is outside the scope of this document.

Install Xephyr on Linux

Xephyr is available in the standard repositories of the supported Linux distributions. If it has not already been installed, install it using the following command on an RPM-based system:

```
# yum install xorg-x11-server-Xephyr
```

Install Xming on Microsoft Windows

There are several commercial products. However, there are also free X-server packages, for example the X-server integrated in Cygwin, VcXsrv, or Xming. The following examples use Xming.

The installer for Xming and more product information are available on http://www.straightrunning.com/XmingNotes/.

Running an X-Session via SSH on Linux

Please note: any firewall between the HP-UX and the X-Server system must allow the SSH connection between the two systems.

Advantages compared to using XDMCP:

- encrypted communication between the guest HP-UX system and the X-Server,
- no font problems.

Disadvantages compared to using XDMCP:

- more complicated handling,
- does not use the general CDE login mechanism.

Perform the following steps to start an Xsession via SSH.

	Step	Command
1	Start Xephyr.	**`$ Xephyr :20 -ac -screen 950x540`** Parameters: • :20 - the DISPLAY variable is set to :20 • -ac - no access control to the X-Server • -screen - the dimensions of the X-display (example only) For more information refer to the man-pages of your Linux system.

2	Start an xterm inside Xephyr and create a SSH connection to HP-UX.	**`$ xterm -display :20  \`**
		`   -e ssh -Yf <user@hp-ux-ip-address> /usr/dt/bin/Xsession`

3	Wait for the password prompt inside the xterm window and enter the password for the HP-UX user.	
4	An X-Session for the user will open in the Xephyr window.	

Using XDMCP

Before using the X-server, XDMCP must be enabled on the guest system. XDMCP allows the HP-UX dtlogin screen to be displayed on remote X-Servers upon request of the X-Server.

Please note: Any intermediate firewall must allow access to UDP port 177 on HP-UX, and to TCP ports 6000~60xx from the remote host to your PC (depending on the X-Server settings).

Enabling XDMCP

Perform the following steps to enable XDMCP on HP-UX:

	Step	Description
1	Create a system-specific copy of the Xconfig file	**`# cp /usr/dt/config/Xconfig /etc/dt/config/Xconfig`**
2	Enable XDMCP.	Edit the file: **`# vi /etc/dt/config/Xconfig`** Locate the following line and insert a comment character, '**#**', at the beginning of the line. `Dtlogin.requestPort: 0` Save the file.

3	Restart the X-server.	**`# /sbin/init.d/dtlogin.rc stop`**
		`# /sbin/init.d/dtlogin.rc start`
		(If there is no dtlogin.rc file in **/sbin/init.d**, you first have to run **`/usr/dt/bin/dtconfig -e`**)

Font Considerations

Often, the system running the X-Server does not have all the necessary fonts to run HP-UX X-applications. In such cases, the graphical applications will not start. You can check if this is the case by monitoring the log file **/usr/dt/Xerrors** on the HP-UX guest system.

If fonts are missing, there are two ways to overcome the problem:

- Configure and activate the fontserver on HP-UX and include it in the fontpath on the system running the X-Server. **This is the preferred solution**.
- Copy the missing fonts to the system running the X-Server and include them in the local fontpath.

Configuring HP-UX as a Fontserver

To configure and activate the fontserver on HP-UX, perform the following steps:

	Step	Command

1	Include the CDE fonts in the fontserver's catalog.	Use a text editor to open the file
		`/etc/X11/fs/config`
		If not already there, append
		`,/usr/dt/config/xfonts/C`
		to the end of the line beginning with
		`catalogue =`

2	Enable the automatic start of the fontserver at boot.	Use a text editor to open the file
		`/etc/rc.config.d/xfs`
		If not already set, set the following variable to 1:
		`RUN_X_FONT_SERVER=1`

3	(Re)start the fontserver.	**`# /sbin/init.d/xfs stop`**
		`# /sbin/init.d/xfs start`

4	Test if the fontserver responds to queries.	Enter the following command on the system running the X-Server:
		`$ fslsfonts -server` *`<font-server-ip>`***`:7000 -fn '-dt*' | head`**
		This should produce output similar to the following:
		`-dt-application-bold-i-normal-sans-10-100-75-75-p-60-hp-roman8`
		`-dt-application-bold-i-normal-sans-10-100-75-75-p-60-iso8859-1`

Please note: By default, the fontserver runs on port 7000. This port must be allowed through any intermediate firewalls.

Once the fontserver responds to queries, it should be possible to add it to the fontpath of the X-Server (see below in the operating system specific sections).

Copying Missing Fonts to a Linux X-Server

Please note: This should only be attempted, if the use of a fontserver is not possible for some reason.

To add HP-UX fonts to the Linux system running the X-Server perform the following steps:

	Step	Command

1	Create a directory to store the additional fonts.	For example:
		`# mkdir /usr/share/fonts/fonts1`
		`# mkdir /usr/share/fonts/fonts2`

2	Copy the missing fonts from the HP-UX guest to the new directory.	Example using the most likely missing fonts:
		`# cd /usr/share/fonts/fonts1`
		`# scp <hp-ux-ip>:/usr/lib/X11/fonts/hp_roman8/75dpi/* .`
		`# cd /usr/share/fonts/fonts2`
		`# scp <hp-ux-ip>:/usr/dt/config/xfonts/C/* .`

3	Create an updated **fonts.dir** file.	For each newly created fonts directory execute the following command:
		`# cd <directory containing the fonts>`
		`# mkfontdir`

4	Check if fontpath can be added to X-Server.	Assuming you run a graphical user session on Linux, use the following command to add a new font directory to your server (until next restart):
		`# xset fp+ <directory containing the fonts>`
		To display the fontpath of the currently used X-Server, use the following command:
		`# xset q`

5	Refresh the font cache.	Run the following command to refresh the font cache: `# fc-cache -fv`
6	Verify whether your X-Server finds the fonts.	Assuming you run a graphical user session on Linux, use the following command to query the X-Server for known fonts (using HP-UX fontnames as an example): `# xlsfonts -fn '-dt*' | head`

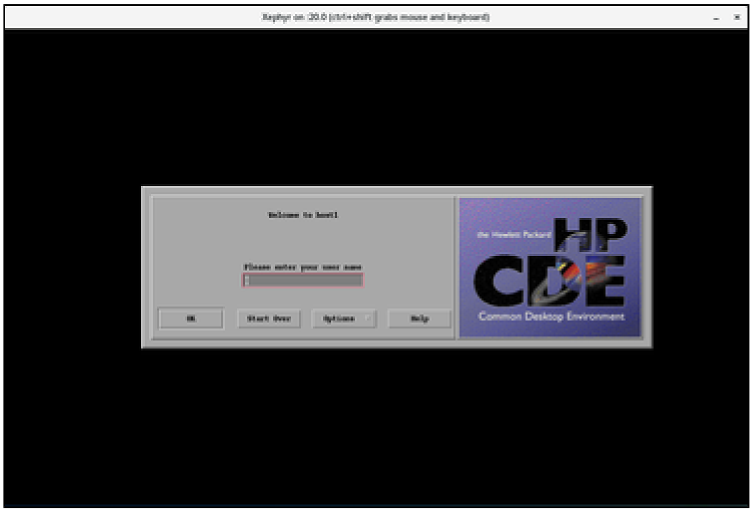
Connect to HP-UX from Linux

This section shows how to use Xephyr to connect to a HP-UX guest system using XDMCP.

Perform the following steps to start an Xsession via XDMCP.

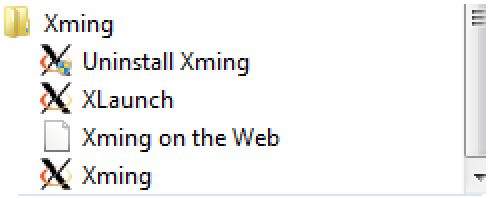
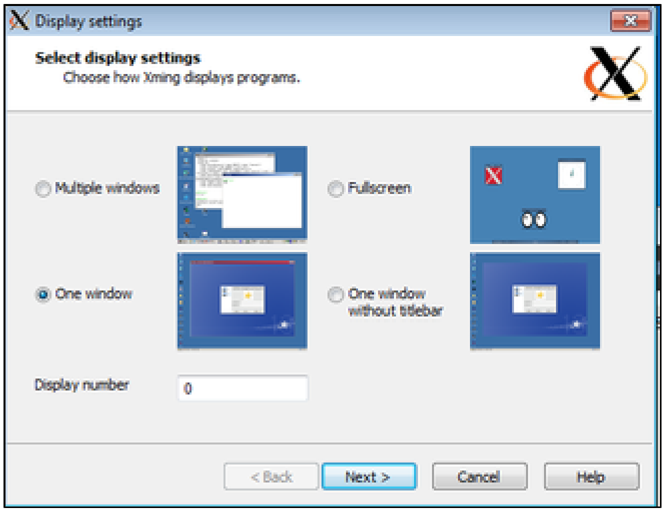
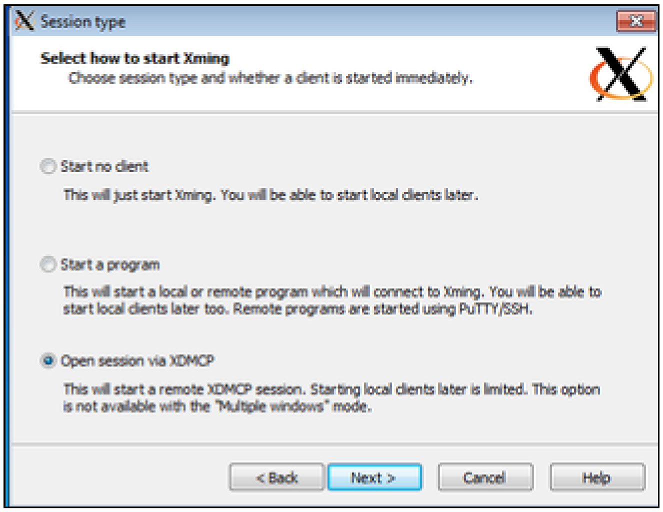
	Step	Command

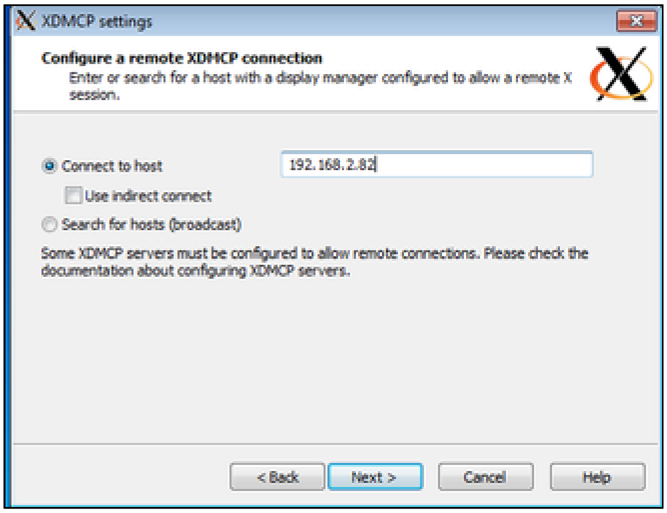
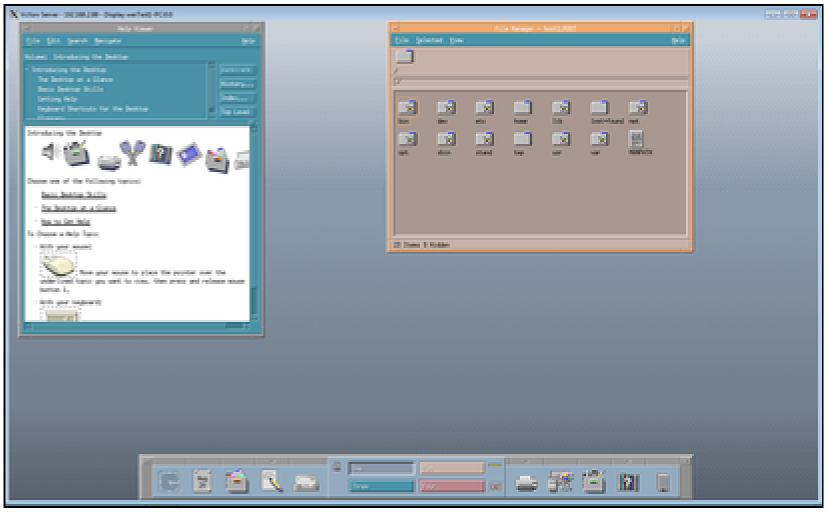
1	Start Xephyr.	Use local font directories:
		`$ Xephyr :20 -ac -screen 950x540 -query <hp-ux-guest-ip> \`
		`  -fp <new-font-directory1>, <new-font-directory2>,...`
		Use fontserver:
		`$ Xephyr :20 -ac -screen 950x540 -query <hp-ux-guest-ip> \`
		`  -fp tcp/<hp-ux-guest-ip>:7000/all`
		Parameters:
		• :20 - the DISPLAY variable is set to :20
		• -ac - no access control to the X-Server
		• -screen - the dimensions of the X-display (example only)
		• -query - requests an XDMCP session from the HP-UX guest
		• -fp - add the local fontpath for the required fonts or a fontserver
		For more information refer to the man-pages of your Linux system.

2	A CDE login window will appear in the Xephyr window.	 Xephyr on :20.0 (ctrl+shift grabs mouse and keyboard) Welcome to host1 Please enter your user name OK Start Over Options Help the Hewlett Packard HP CDE Common Desktop Environment

Connect to HP-UX from Windows

This section shows how to use Xming to connect to a HP-UX guest system using XDMCP.

	Step	Description
1	Start **XLaunch** from the Xming section in the **Start** menu. Make sure that Xming is not already running before you start	 Xming Uninstall Xming XLaunch Xming on the Web Xming
2	- Select the option **One Window**. This is one of the options supporting XDMCP. - Press **Next** to continue	 Display settings **Select display settings** Choose how Xming displays programs. ○ Multiple windows ○ Fullscreen ● One window ○ One window without titlebar Display number: 0
3	- Select **XDMCP** for the session. - Press **Next** to continue.	 Session type **Select how to start Xming** Choose session type and whether a client is started immediately. ○ Start no client This will just start Xming. You will be able to start local clients later. ○ Start a program This will start a local or remote program which will connect to Xming. You will be able to start local clients later too. Remote programs are started using PuTTY/SSH. ● Open session via XDMCP This will start a remote XDMCP session. Starting local clients later is limited. This option is not available with the "Multiple windows" mode.

4	• Enter the IP address or hostname of the virtual HP-UX system. • Press **Next** to continue.	 XDMCP settings **Configure a remote XDMCP connection** Enter or search for a host with a display manager configured to allow a remote X session. ● Connect to host 192.168.2.82 ☐ Use indirect connect ○ Search for hosts (broadcast) Some XDMCP servers must be configured to allow remote connections. Please check the documentation about configuring XDMCP servers. < Back | Next > | Cancel | Help
5	Finish the setup by continuing through the remainder of the sections (for this example all the defaults were accepted) and press **Finish** on the last configuration screen of XLaunch.	
6	The HP-UX graphical login screen is displayed and after logging in, you will see the graphical desktop.	

Ethernet Configuration Notes

Contents

General Information

This section provides information on basic aspects of configuring an Ethernet interface for the emulator. For details on the configuration parameters used and for additional parameters, please refer to the Configuration File Reference.

The emulated Ethernet interfaces of Charon-PAR can be linked either to a

- physical host interface,
- a TAP interface connected to a virtual bridge on the host, or
- (since Charon-PAR 3.0.9) a MACVTAP interface linked to a physical NIC on the host.

Notes for physical interfaces:

- If used directly as a guest interface, offloading parameters must be turned off using the initialization command parameter in the configuration file.
- In cloud environments, the necessary interface settings are not always reflected correctly on the cloud-provided NICs. For such installation settings a MACVTAP interface can be used to provide a dedicated connection without connecting the emulator directly to the physical interface.

Notes for TAP interfaces:

- A TAP interface can either be created by the user or will be created automatically by the emulator if it does not already exist.
- For automatically created TAP interfaces, the user can specify a name or let the emulator select a name.
- A automatically created TAP interface is not automatically added to a bridge, this must be configured via the initialize_command (see the Ethernet Configuration Parameter table below).
- An automatically created TAP interface is deleted automatically upon emulator stop.
- If used in a cloud environment, only internal virtual bridges (no connection to the cloud LAN) are supported.

Notes for MACVTAP interfaces:

- Introduces a transparent bridge interface associated with on physical NIC on the host, and provides a dedicated interface to the emulator.
- Provides an abstraction layer between physical NIC and emulator.
- Physical interface requires no changes in the offloading configuration.
- Fragmentation is performed by Linux on the MACVTAP interface. Jumbo frames on the physical interface will not cause problems.
- Uses same MAC address as the physical interface.
- Suited for associating a dedicated host Ethernet interface to the emulator in cloud environments.

Notes for network configurations on VMware:

- ESXi has configuration parameters that improve security but may block certain emulator NIC configurations.
- One configuration option allows or blocks promiscuous mode (the default mode of operation for Charon-PAR guest NICs).
- Other configuration options allow or block the change of a MAC address or the use of a different, additional MAC address on the same NIC (*MAC address changes*, *Forged transmits*).

By default, emulated models have one Ethernet device. Depending on the model, more Ethernet devices can be added.

A guest HP-UX system will perceive the network card as a 10/100 Mbit/s controller running at 10 Mbit/s half-duplex. If the user tries to change this setting using SAM or the **lanadmin** command, the command will be accepted but the displayed interface settings will not change. However, the throughput of the emulated network card depends on the combination of network performance, physical network card characteristics, host system and guest system performance - it is not tied to the displayed interface settings. So the actual throughput will be what can be achieved depending on the conditions listed above.

The emulated network devices

- DE 500 PCI based cards (for 64-bit systems), and
- LASI-82596 cards (for the 32-bit system)

do not support Jumbo frames. For physical interfaces, this feature must be disabled in the emulator configuration (together with any other offloading parameters) using the **initialize_command** parameter.

Considerations for Cloud Environments

Every cloud environment has specific characteristics that could conflict with the interface configurations made for Charon-PAR. Please refer to the documentation provided by the cloud provider to understand the networking behavior of your cloud instance **before you change any interface settings**.

Important points that apply to most cloud environments:

- If the Charon host is configured with more than one active IP interface, asymmetric routing can cause connectivity problems. In such cases, policy-based routing (per interface routing tables with associated IP rules) is required.
- Only IP unicast traffic is supported. Non-IP traffic or multicast/broadcast traffic is not supported and requires traffic tunneling.
- Promiscuous interface mode is not supported.
- Only traffic with the MAC address assigned by the cloud provider is allowed across an interface.
- Routing requires special configuration steps (source/destination check disabling) on the cloud instances. Enabling IP forwarding on the Linux host is not enough.
- Cloud specific security rules must allow the relevant traffic. Configuring the Linux firewall correctly is not enough.
- If a host NIC is dedicated to a guest system, the MAC address and IP address assigned to the interface by the cloud provider must be used by the guest.

Basic Configuration Components

Ethernet Interface Names

Names on 64-bit systems:

The name of the Ethernet interface in the emulator configuration file has the format **EW***XN* with the following definitions:

- *X* - an uppercase letter starting with **A** for the first interface and then continues with **B**, **C**, etc. for additional interfaces. The possible number of network cards depends on the features of the original physical system. The absolute maximum number is 16.
- *N* - the device number of the card starting with 0 for each value of *x*.

Names on 32-bit systems:

The currently supported 32-bit system supports only one Ethernet card:

- Model 720: **system.lan0.card**
- Model B132L: **gsc.lasi.lan.card**

Minimal Configuration Parameters

The configuration of each network interface must at least have two basic components as shown below:

64-bit system (example shows the first Ethernet card EWA0):

- Interface definition (**`EWA0.iface`**)
- Interface initialization command (**EWA0.initialize_command**)

32-bit system:

- Interface definition (**system.lan0.card`.iface`** or **gsc.lasi.lan.card**.**iface**)
- Interface initialization command (**system.lan0.card.initialize_command** or **gsc.lasi.lan.card**.**initialize_command**)

The default configuration assumes that a physical network interface will be used.

Please refer to the Configuration File Reference and the configuration template of your model for more information.

Basic Configuration Example

Example of a basic network card configuration using a physical interface on a 64-bit emulated system:

```
EWA0.iface="enp0s20f0u3u4"
EWA0.initialize_command="ethtool -K $IFACE rx off ; ethtool -K $IFACE tx off ; ethtool -K $IFACE sg off ; ethtool -K $IFACE
gso off ; ethtool -K $IFACE gro off ; ethtool -K $IFACE txvlan off ; ethtool -K $IFACE rxvlan off"
```

The variable **IFACE** is set automatically by the emulator upon start. The **initialize_command** parameter is used to turn off all relevant offloading parameters on the interface used by the emulator.

Adding Additional Emulated Network Cards

Please note: if additional Ethernet controllers can be loaded - and how many - depends on the emulated model.

This section will show an example. Please refer to the section *I/O Slot Configuration* in the Configuration File Reference chapter and the Emulated Model Hardware Configuration Details chapter for details.

To add an additional Ethernet controller to your emulator configuration, you must perform the following steps:

- Identify a free PCI slot (model configuration template or Emulated Model Hardware Configuration Details).
- Add a **load** command to the configuration to load an Ethernet module into the identified emulator slot (see the Configuration File Reference chapter for details).
- Add the network card configuration to your configuration file.
- Start the emulator and configure the interface in the guest system.

Step 1: identify a free PCI slot for your emulated model. The available slots depend on your model and the existing configuration.

Example: model "rp7400-1-650" has a PCI I/O expansion slot 4 (device path 0/8/0)

Step 2: Add the load command for the Ethernet module to the configuration file.

Sample configuration command for slot 4: **load ETH tulip PCI 4**

Step 3: Add the network card configuration to your configuration file selecting the correct host interface name (example: *eth2*). Note that the name of the second emulated interface is **EWB0**.

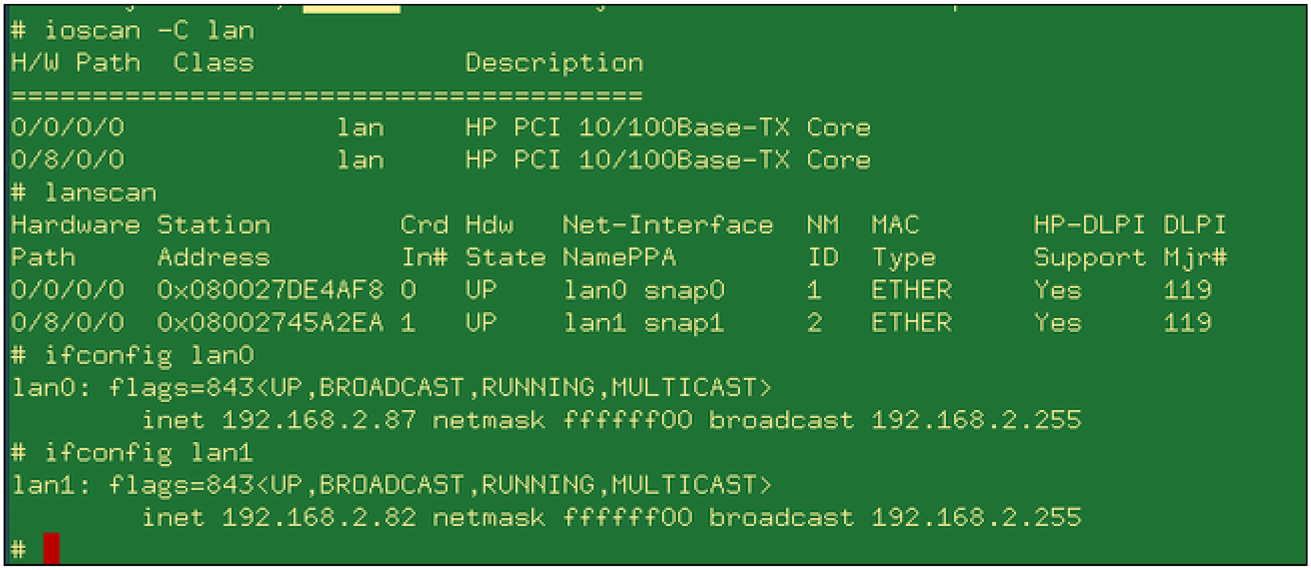
Example (**important:** the initialization command must be in one continuous line):

```
EWB0.iface="eth2"
EWB0.initialize_command="ethtool -K $IFACE rx off; ethtool -K $IFACE tx off; ethtool -K $IFACE sg off; ethtool -K $IFACE gso off; ethtool -K $IFACE gro off; ethtool -K $IFACE txvlan off; ethtool -K $IFACE rxvlan off"
```

Step 4: Start the emulator, boot the guest system and configure the second interface.

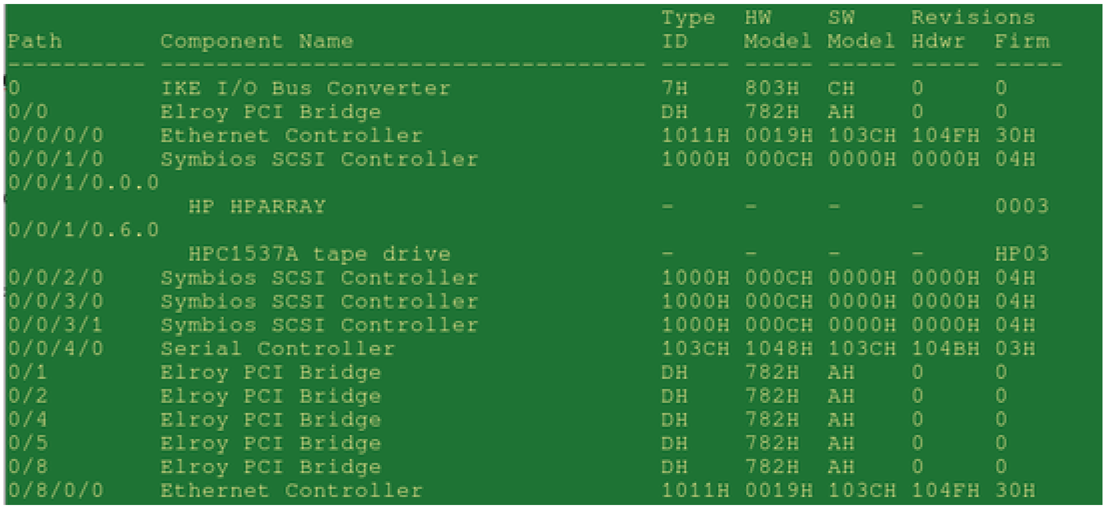
- On PA3, use the `ISL > ODE > RUN MAPPER2` command (access from console via: Interaction with IPL) to verify the existence of the new interface and the correct device path.
- The above command also works on PA9, but on HP-UX you can also use the `lanscan` command to verify the existence of the new interface (on HP-UX it is normally lan1).
- Configure the interface.
  - The easiest way on HP-UX is the following:
    - HP-UX before 11.31: use **SAM > Network and Communication > Network Interface Cards** (*Actions* > *Configure*).
    - HP-UX 11.31: use **SMH > Networking and Communications-> Network Interfaces Configuration > Network Interface Cards**. Please note that at the time of writing SMH only worked with the emulator when being used over the network (not on the serial console).
  - On MPE/iX, the interface must be added via NMMGR (for interactive use, **xhpterm** can be used to access this tool), and the interface must be started using the NETCONTROL command. Please refer to the documentation of your operating system for further information.

The following image shows a sample of a HP-UX system with two Ethernet interfaces:



```
# ioscan -C lan
H/W Path  Class            Description
=======================================
0/0/0/0             lan     HP PCI 10/100Base-TX Core
0/8/0/0             lan     HP PCI 10/100Base-TX Core
# lanscan
Hardware Station        Crd Hdw   Net-Interface  NM  MAC       HP-DLPI DLPI
Path     Address        In# State NamePPA        ID  Type      Support Mjr#
0/0/0/0  0x080027DE4AF8 0   UP    lan0 snap0     1   ETHER     Yes     119
0/8/0/0  0x08002745A2EA 1   UP    lan1 snap1     2   ETHER     Yes     119
# ifconfig lan0
lan0: flags=843<UP,BROADCAST,RUNNING,MULTICAST>
        inet 192.168.2.87 netmask ffffff00 broadcast 192.168.2.255
# ifconfig lan1
lan1: flags=843<UP,BROADCAST,RUNNING,MULTICAST>
        inet 192.168.2.82 netmask ffffff00 broadcast 192.168.2.255
#
```

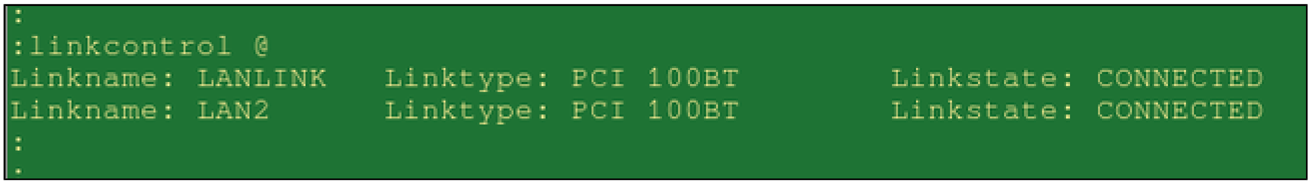
The following shows a sample of the second Ethernet interface being configured on device path 0/8/0/0:



```
                                                    Type  HW    SW    Revisions
Path       Component Name                           ID    Model Model Hdwr  Firm
---------- ---------------------------------------- ----- ----- ----- ----- -----
0          IKE I/O Bus Converter                    7H    803H  CH    0     0
0/0        Elroy PCI Bridge                         DH    782H  AH    0     0
0/0/0/0    Ethernet Controller                      1011H 0019H 103CH 104FH 30H
0/0/1/0    Symbios SCSI Controller                  1000H 000CH 0000H 0000H 04H
0/0/1/0.0.0
             HP HPARRAY                             -     -     -     -     0003
0/0/1/0.6.0
             HPC1537A tape drive                    -     -     -     -     HP03
0/0/2/0    Symbios SCSI Controller                  1000H 000CH 0000H 0000H 04H
0/0/3/0    Symbios SCSI Controller                  1000H 000CH 0000H 0000H 04H
0/0/3/1    Symbios SCSI Controller                  1000H 000CH 0000H 0000H 04H
0/0/4/0    Serial Controller                        103CH 1048H 103CH 104BH 03H
0/1        Elroy PCI Bridge                         DH    782H  AH    0     0
0/2        Elroy PCI Bridge                         DH    782H  AH    0     0
0/4        Elroy PCI Bridge                         DH    782H  AH    0     0
0/5        Elroy PCI Bridge                         DH    782H  AH    0     0
0/8        Elroy PCI Bridge                         DH    782H  AH    0     0
0/8/0/0    Ethernet Controller                      1011H 0019H 103CH 104FH 30H
```

On an MPE/iX system, the device path is needed to configure the new network link.

The following shows a sample of two active Ethernet interfaces on MPE/iX:



```
:
:linkcontrol @
Linkname: LANLINK   Linktype: PCI 100BT        Linkstate: CONNECTED
Linkname: LAN2      Linktype: PCI 100BT        Linkstate: CONNECTED
:
:
```

Supplemental Information: Virtual Bridge and TAP Interfaces

The following sections describe the basic steps for the configuration of virtual bridges and TAP interfaces.

Please note:

- The examples describe a virtual bridge that connects host and guest system to the local LAN via a physical NIC.
- The described virtual bridge configuration is mostly suited for **on-premises** installations where the host system runs on a physical system.
- For **VMware** (or similar environments) it is recommended to add additional vNICs for the guest as needed - instead of using a bridge configuration on the host.
- In most **cloud environments**, a virtual bridge cannot directly link bridge traffic to the LAN via a host NIC. There are usually restrictions regarding the MAC addresses allowed on the NICs of a cloud instance. Also promiscuous mode is usually not possible. Hence, in such environments, traffic between a virtual bridge and the rest of the network must usually be NAT'ed or routed. The virtual bridge is then internal to the Linux host and not connected directly to the LAN.

Basic Bridge Configuration Using *ip* Commands

The following table shows a simple example of how to create a virtual bridge on Linux and how to add TAP interfaces to it using **ip commands**:

Step	Command

Gather information about the existing network configuration.	List all existing interfaces, their status and their MAC addresses:
	`# ip link show`
	List all configured IP addresses and netmasks:
	`# ip addr show`
	List the routing table entries:
	`# ip route show`

Create a TAP interface (not needed if emulator automatic creation is used).	`# ip tuntap add dev my_tap0 mod tap`
Activate TAP interface (can also be done via the initialize command in emulator configuration).	`# ip link set my_tap0 up`
Create a new bridge - if needed.	`# ip link add name my_bridge type bridge`
Activate the bridge - if needed	`# ip link set my_bridge up`
If needed, add a physical interface to the bridge. This interface will connect the bridge to the physical LAN (not supported for cloud installations).	`# ip link set <physical-int> master my_bridge`
Remove any configured IP address from the physical interface.	`# ip addr delete <ip-address>/<netmask> dev <physical-int>`
Add the IP address previously configured on the physical interface (or another IP address to be used by the host system) to the bridge interface.	`# ip addr add <ip-address>/<netmask> dev my_bridge`

Add any routes (including the default gateway) to be routed across the bridge interface - if needed.	`# ip route add <network>/<netmask> via <gwy-IP> dev my_bridge`
	`# ip route add default via <gwy-IP> dev my_bridge`

Add the TAP interface to the bridge (can also be done via the initialize command in the emulator configuration).	`# ip link set my_tap0 master my_bridge`

Please note: The commands above are not persistent. They would need to be scripted and added to the host system startup as required.

To delete TAP and bridge interfaces, you can use the command `ip link delete <interface-name>`.

Basic Bridge Configuration Using *nmcli*

The following table shows a simple example of how to create a virtual bridge on Linux and how to add TAP interfaces to it using nmcli commands (NetworkManager command-line interface):

Step	Description

Gather information about the existing network configuration.	List all existing interfaces, their status and their MAC addresses:
	`# ip link show`
	List all configured IP addresses and netmasks:
	`# ip addr show`
	List the routing table entries:
	`# ip route show`

If needed, create a new connection for the physical interface to be used with the bridge.	`# nmcli conn add type ethernet autoconnect yes \`
	`con-name <physical-int-name> \`
	`ifname <pysical-int-name> ip4 0.0.0.0`

If needed, create a bridge and set a static IP address for it.	`# nmcli con add type bridge autoconnect yes \`
	`con-name mybr0 ifname mybr0 \`
	`ipv4.method manual ip4 <ipaddr/mask> stp off`

If needed, activate bridge.	`# nmcli con up bridge-mybr0`

If needed, add physical interface to bridge (not supported for cloud installations).	`# nmcli conn modify <physical-int-conn> master mybr0 \`
	`slave-type bridge`

Activate changes to interface.	`# nmcli conn up <physical-int-conn>`

After the bridge has been created, you can let the emulator create a TAP interface automatically and use the initialize command in the emulator configuration to add the TAP interface for the emulator to the bridge.

Please note: By default, **nmcli** commands are persistent. The connection information is stored in */etc/sysconfig/network-scripts/* if the default plugin in */etc/NetworkManager/NetworkManager.conf* is *ifcfg-rh*. It is stored in */etc/NetworkManager/system-connections/* if the default plugin is *keyfile*. Permanent TAP connection information is always stored in the NetworkManager directory unless the network-scripts package is used for configuration.

Basic Emulator Configuration Using a TAP Interface

The following example shows how to use a TAP interface in the emulator configuration, where the interface is automatically created and removed by the emulator. The initialization command adds the interface to an existing bridge named *my_bridge*:

```
EWA0.mapping_mode="TAP"
EWA0.initialize_command="ip link set ${IFACE} master my_bridge"
```

Please refer to the Configuration File Reference and the configuration template of your model for more information.

Linux Firewalls and Virtual Bridges

If firewall rules are to be used for bridged traffic, the kernel can be instructed to apply iptables (also arptables and ip6tables) rules to bridged traffic. In older versions, this option was included in the bridge functionality itself. Starting with kernel 3.18, the filtering functionality in the form of the br_netfilter module was moved into a separate module that can be loaded by the user if required. If the module is not loaded, no firewall rules are applied to bridge traffic and no further actions are required to pass the bridged traffic through the Linux host system.

To check, if the module is loaded use the command

```
# lsmod | grep netfilter
```

To use the firewall for bridged traffic on newer Linux kernels, the module must be loaded using the command

```
# modprobe br_netfilter
```

or by defining an **iptables** rule that uses the **physdev** module.

If the module has been loaded, the following system configuration parameters are available:

- `net.bridge.bridge-nf-call-iptables`
- `bridge-nf-call-arptables`
- `bridge-nf-call-ip6tables`

They are **set to 1 by default** (equivalent to `echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables`). This value enables iptables rules for bridged traffic (and by default blocks bridged traffic).

Setting the parameters to 0 will disable the firewall for bridged traffic. They can be set permanently via */etc/sysctl.conf*.

To allow bridged traffic through the enabled firewall, use commands like the following:

```
# firewall-cmd --permanent --direct --add-rule ipv4 filter INPUT 1 \

-m physdev --physdev-is-bridged -j ACCEPT

# firewall-cmd --permanent --direct --add-rule ipv6 filter INPUT 1 \

-m physdev --physdev-is-bridged -j ACCEPT

# firewall-cmd --reload
```

Please refer to the documentation of your host system for more detailed information.

Please note: at the time of writing, this feature is not yet available for nftables.

Supplemental Information: MACVTAP Interfaces (since version 3.0.9)

In some cloud environments, interface settings on the Linux level are not passed properly to the NIC on the cloud level. This can cause problems with network connectivity and, in particular, with network performance. The problem can be solved by using a virtual bridge between host and guest, and by configuring routing or NAT to allow guest communication with other systems. However, such configurations are often complex due to the fact that only the cloud-assigned MAC and IP addresses can be used. On a virtual bridge to which host and guest are attached, this condition cannot be met.
MACVTAP interfaces provide an alternative solution - they are linked to a host NIC and are TAP based. However, when used in passthrough mode, they can use the same MAC and IP address as assigned by the cloud provider.

Basic (non-persistent) configuration of a MACVTAP interface on the Linux host system:

Activate the NIC that is dedicated to the emulator:

```
# ip link set <nic-name> up
```

Remove IPv4 and IPv6 addresses from the NIC:

```
# ip addr flush <nic-name>

# ip -6 addr flush <nic-name>
```

Create a MACVTAP interface linked to the host NIC:

```
# ip link add link <nic-name> name <macvtap0> type macvtap mode passthru
```

Configure the correct MTU on the new interface:

```
# ip link set dev macvtap0 mtu 1500
```

Activate the new interface:

```
# ip link set macvtap0 up
```

Set the correct permissions on the underlying TAP interface:

```
# chown ${USER} /dev/tap`cat /sys/class/net/macvtap0/ifindex`
```

Basic (persistent) configuration of a MACVTAP interface on the Linux host system using nmcli commands:

If desired rename connection:

```
# nmcli conn mod <old-connection-name> con-name <new-connection-name>
```

Remove IP address from physical interface and configure automatic interface activation at boot:

```
# nmcli conn mod <new-connection-name> ipv4.method "disabled" ipv6.method "disabled"

# nmcli conn mod <new-connection-name> connection.autoconnect yes
```

Activate changes on physical interface (this command may time-out if IP connection test is configured)

```
# nmcli con up <new-connection-name>
```

Create MACVTAP interface, disable the IP configuration and activate the interface:

```
# nmcli conn add type macvlan mode passthru dev <physical-NIC> tap yes autoconnect yes con-name macvtap0
ifname macvtap0

# nmcli conn mod macvtap0 ipv4.method "disabled" ipv6.method "disabled"

# nmcli con up macvtap0
```

Basic emulator configuration using a MACVTAP interface:

```
EWA0.mapping_mode = "MACVTAP"
EWA0.iface = "macvtap0"
```

Configuring SuperIO Devices

Contents

General Information

The SuperIO is the emulation of a PCI device containing PC style peripherals used in legacy 64-bit PA-RISC systems for HP-UX:

- parallel port
- 2 serial ports
- dual-channel IDE controller
- floppy disk controller
- USB 1.1 controller
- timer
- PIC interrupt controller

Please note:

- The current version of **Charon-PAR emulates only a subset** of these devices.
- **Not supported** on Charon-PAR/PA9-32 model 720.

Currently supported are (depending on support by the emulated model and guest operating system)

- two serial ports (physical ports not available in cloud environments)
- one parallel port (not available in cloud environments)

SuperIO Module Configuration

Loading the SuperIO device

Before any devices of the module can be configured and used, the module must be loaded in the emulator configuration file.

The SuperIO module can be loaded in any system model with PCI bus support.

Use the following syntax to load this module into a slot on the default bus:

```
load SUPERIO sio <bus-number> <slot-number>
```

Example:

```
load SUPERIO sio 0 6
```

Use the following syntax to load this module into an PCI expansion slot:

```
load SUPERIO sio PCI <slot-number>
```

where the slot number is the number listed in the configuration file template or in Emulated Model Hardware Configuration Details for the expansion I/O slot selected.

Please note:

- Only one SuperIO module can be loaded in the configuration. Multiple SuperIO instances will not work properly.
- The usable bus numbers and slot numbers depend on the emulated model and the already loaded devices.
- If another PCI device is already installed in the specified PCI slot, the load command will fail with an error message.
- **The correct bus/slot location of the device is important to preserve the correct GSP (service processor) console configuration**:
  - If the SuperIO module is inserted in a slot before the normal system serial console, the service processor console (GSP console) is set to the first SuperIO serial port.
  - HP-UX does not support the SuperIO serial port as GSP console. With such a configuration, it will crash early-on in the boot process. Therefore, take care to **always use a slot number higher than the one where the correct system console line has been loaded**.

After booting the guest HP-UX system, the example above will result in an **ioscan** output similar to the following:

```
ba          1  0/0/6/1      superio   CLAIMED     BUS_NEXUS    PCI Core I/O Adapter
tty         1  0/0/6/1/1    asio0     CLAIMED     INTERFACE    Built-in RS-232C
tty         2  0/0/6/1/2    asio0     CLAIMED     INTERFACE    Built-in RS-232C
ext_bus     4  0/0/6/1/3    SCentIf   CLAIMED     INTERFACE    Built-in Parallel Interface
unknown    -1  0/0/6/1/4              UNCLAIMED   UNKNOWN      Built-in Floppy Drive
```

SuperIO Serial Ports

Serial ports installed on SuperIO module can be configured the same as other serial ports. Please see chapter Serial Line Emulation Notes for more information.

Path names for serial devices in the emulator configuration:

- superio_001.uart0 - COM0 port
- superio_001.uart1 - COM1 port

Serial ports configuration example:

```
superio_001.uart0.device.type="telnet"
superio_001.uart0.device.port=":30001"

superio_001.uart1.device.type="telnet"
superio_001.uart1.device.port=":30002"
```

SuperIO Parallel Port

The parallel port installed on a SuperIO module connects to the host system parallel port device **`/dev/parport0.`** The parallel port output from the guest OS is redirected to the host parallel port. No additional configuration for the parallel port is required.

However, it may be necessary to install the required kernel drivers in HP-UX and rebuild the kernel to activate the interface.

If the **ioscan** command shows the parallel port as **UNCLAIMED**, perform the following steps:

1. Use the kcmodule command to verify the status of the required drivers:
   ```
   # kcmodule |grep CentIf
   CentIf static explicit
   SCentIf static explicit
   ```
2. If the status is unused, load the modules into the kernel using the commands:
   ```
   # kcmodule CentIf=best
   # kcmodule SCentIf=best
   ```
3. You will be informed that this change can only become active after the next reboot. If you confirm, the kernel will be modified and you can reboot the system.

Once the **ioscan** command shows that the parallel port as **CLAIMED** by the correct driver, you can, for example, use **SAM** to configure a printer on the parallel port
(**Printers and Plotters > LP Spooler > Printers and Plotters > Action > Add Local Printer > Add Parallel Printer**). On HP-UX 11.31, use **SMH** for this task. Please note that at the time of writing SMH only worked with the emulator when being used over the network (not on the serial console).

If no parallel port device exists on the host system, an error message similar to the one below will be printed to the emulator log. It does not affect system operation, but the parallel port redirection will not work.

```
err:open('/dev/parport0', O_RDWR) is failed (errno 2) No such file or directory
err:ioctl(handle, PPCLAIM) is failed (errno 9) Bad file descriptor
```

Modern host systems often have no physical parallel port. In such cases, a USB-LPT adapter or a software redirector (such as LPT-over-IP) can be used. The emulated parallel port operation depends on the host parallel port operation and some USB-LPT adapters may not produce reliable results.

Charon Power Save Feature (IDLE)

Contents

Charon IDLE Overview

Charon IDLE is a power save feature. It is supported on HP-UX guest systems and provides an HP-UX service which forces the Charon emulator to release unused CPU cycles to the Linux host. This significantly reduces the Charon-PAR CPU usage in cases where the guest OS is idle or does not requires 100% CPU utilization.

In cloud environments where customer might pay for CPU utilization this will reduce the cost of the Charon-PAR solution.

Charon IDLE Installation on HP-UX

The Charon-PAR kit contains the HP-UX software depot with the Charon IDLE package. Perform the following steps to install it on your HP-UX guest system:

1. Locate the software depot **`Charon_Idle_0.01.depot`** in your Charon-PAR kit.
2. Copy the file to the HP-UX guest system (for example using scp or ftp).
3. Log into the HP-UX system as the **root** user.
4. Install the Charon_IDLE package:
   ```
   # swinstall -s /<absolute-path>/Charon_Idle_0.01.depot Charon_Idle
   ```
5. Reboot the HP-UX guest system or start the service manually:
   ```
   # /sbin/init.d/charon_agent start
   ```

You can check if the package is already installed by entering the command: **`# swlist Charon_Idle`**.

The Linux host will now show the actual CPU usage of the emulated system instead of always 100% CPU utilization. Please note that the service is started with HP-UX startup. Hence the IDLE service will not be active before booting the system.

Charon IDLE Management and Deinstallation

The following commands can be used to **start and stop** the service:

Start the Charon IDLE service on HP-UX:

```
# /sbin/init.d/charon_agent start
```

Stop the Charon IDLE service on HP-UX:

```
# /sbin/init.d/charon_agent stop
```

The following command can be used to **deinstall** the Charon IDLE service:

```
# swremove Charon_Idle
```

Charon-PAR Command-Line Options

Please note:

- Each instance of Charon-PAR must be started in a separate directory because certain files (SSTORAGE.DAT, NVOLATILE.DAT, UA_SSTORAGE.DAT, and console.dat) will always be created in the current working directory of the emulator instance. Starting multiple emulator instances in the same directory will lead to errors due to file locking problems.
- If Charon-PAR is to be started by a non-root user (introduced in version 3.0.5 build 21906), certain capabilities must be set for the Charon-PAR image, for example via the following command:
  ```
  # setcap 'cap_sys_nice+eip cap_ipc_lock+eip cap_net_admin+eip cap_net_raw+eip' /opt/charon/bin/charon-par
  ```
  Please note: if a VE license is used, the emulator must always be started with root privileges.

When starting Charon-PAR (the command **/opt/charon/bin/charon-par**), there are several command-line options that can be used:

Options (1)	**Description**
`-D`, `--dump`	Do not use in a production environment unless instructed to do so by Stromasys support! Emulator runs in core dump mode. If stopped with SIGQUIT or other stop signal, it will write a core file in the location specified by **/proc/sys/kernel/core_pattern**.
`-F`, `--fast-fpu`	Fast FPU mode provides better floating-point performance at the expense of minor deviation in floating-point status reporting. When this option is enabled, all floating-point operations produce exactly the same values as the real hardware. The differences cause FPU tests (ODE WDIAG and llmc_em) to report errors. However, it is highly unlikely that any real system or application software is impacted by this deviation. Configuration show in emulator log as (FPU DIT Mode: Fast - vs Precise).
`-G` *<mode>* `--ditgcsync` *<mode>*	Do not use in a production environment unless instructed to do so by Stromasys support! Possible values: • **sync**: synchronous garbage collector mode • **async**: asynchronous garbage collector mode (default)
`-S` *<mode>* `--ditsync` *<mode>*	Do not use in a production environment unless instructed to do so by Stromasys support! Possible values: • **sync**: synchronous DIT execution mode • **async**: asynchronous DIT execution mode (default) • **no_dit**: DIT disabled
`-K` *<dit-level>* `--dit-kind` *<dit-level>*	Do not use in a production environment unless instructed to do so by Stromasys support! Each Charon-PAR model has a built-in maximal DIT optimization level. This parameter allows to reduce this level. As result, the emulation will run with reduced performance. Possible values (from lowest to highest): **dispatch**, **slow**, **medium**, **basic**, **adv**
`-X` *<mode>* `--collect-statistics <mode>`	Collect the available statistics. Mode format: *<level>*`,json,`*<output-descriptor>*, where *level* = 0-9 The *output-descriptor* can be one of the following: • A filename • A named pipe (e.g., created with the **mkfifo** command); added in 3.0.5-21904 • A Unix socket in the format **unix://**<*socket-name*>; added in 3.0.5-21904
`-a` *<name>* `--instance-name` *<name>*	Assigns a logical name to the emulated system instance at start (reserved for future use).
`-c` *<console-port>* `--console-port` *<console-port>*	Redirects the Charon-PAR console to a TCP/IP port that can be accessed via telnet (from local system only). This makes the console accessible also when the emulator runs in daemon mode. The port number can be in the range of 0-65535. However, port numbers 0-1023 are generally reserved for specific applications and should not be used.
`-f` *<filename>* `--config` *<filename>*	Specifies the path to the emulator configuration file.
`-h` `--help`	Displays a short help information.

`-l` *<logpath>* **`--logfile-path`** *<logpath>*	Specifies the path to the logfile.
`-p` *<filename>* **`--pidfile`** *<filename>*	File to store the process ID of the emulator process. Used mostly when running emulator as a daemon.
`-x` **`--daemon`**	The emulator is started as a daemon. This mode can be used when the emulator is to be started at system boot. It can also be used to start the emulator while connected to the system via the network to avoid license problems.
`--mmu-assist` *<setting>*	Do not use unless advised to do so by Stromasys support! MMU assist type. Possible values: **none, simple, advanced**. Used in debugging situations to modify Charon optimization modes.
`--no-libc-assist`	Do not use unless advised to do so by Stromasys support! Disable libc assist mode. Used in debugging situations.
Added in 3.0.5 build 21904 (do not use unless advised to do so by Stromasys support)**:**	
`--dit_page_headers` *<number>*	DIT code headers number (default 0 - automatic)
`--dit_page_size` *<number>*	DIT code page size KB (default 48). The parameter defines the size of a single translated page chunk size.
`--dit_pool_size` *<number>*	DIT pool size GB (default 2; maximum value 128). Any increase will increase the emulator's memory requirements accordingly.

File Transfer to and from Guest Operating System

This section provides some examples of how to transfer data to/from the guest operating system. These examples are intended for illustrative purposes only. The actual file transfer (especially in the context of a migration from physical to emulated hardware) depends on the exact customer environment and requirements.

Transferring Files to/from an HP-UX Guest

HP-UX offers various commands and methods to copy files between the guest system and other systems.

Examples:

- Copying data using physical media, e.g., a tape drive, to transfer data between HP-UX running on Charon-PAR and another HP-UX system.
- NFS: the HP-UX can mount a remote filesystem or a remote system can mount a filesystem exported by the guest HP-UX. Then data can be exchanged between the two systems.
- File transfer programs, e.g., **sftp** and **scp**.
- General backup and recovery programs in conjunction with a file transfer program, e.g., **tar**, **cpio**, **pax**, **fbackup/frecover**.

The following section will provide some examples. Please refer to your system's documentation for additional information and methods not covered in the examples.

Please note:

When using SSH-based applications (ssh, scp, sftp) for connecting to a HP-UX system from a modern Linux system, you may run into a problem caused by a mismatch of available key exchange methods. The error shown will be similar to the sample below:

```
Unable to negotiate with 192.168.2.87 port 22: no matching key exchange method found. Their offer: diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1
```

In such cases, you can use the following workaround:

> Add the SSH option **`-oKexAlgorithms=+diffie-hellman-group1-sha1`** to your command.

Transferring Files with SCP

With SCP, single files or directory structures (recursive copy) can be copied to and from HP-UX.

The following table shows some SCP syntax examples

Task	Command
Copy single file to another system	$ **scp** *<local-file> <user>@<remote-host>:<remote-file-path>*
Copy a directory recursively to another system	$ **scp -r** *<local-path> <user>@<remote-host>:<remote-path>*
Copy a single file from another system	$ **scp** *<user>@<remote-host>:<remote-file-path> <local-file>*
Copy a directory recursively from another system	$ **scp -r** *<user>@<remote-host>:<remote-path> <local-path>*

The following example shows how a small directory tree is copied to a user account named **charon** from another system:

```
$ scp -r ./Tmp charon@192.168.2.107:

charon@192.168.2.107's password:

file4.txt  100%    0     0.0KB/s   00:00

file3.txt  100%    0     0.0KB/s   00:00

file1.txt  100%    0     0.0KB/s   00:00

file2.txt  100%    0     0.0KB/s   00:00
```

Transferring Files with SFTP

SFTP is another file transfer program that can be used for secure file transfers between the HP-UX guest system and other systems. The SFTP program can be used interactively (similarly to FTP) and in non-interactive mode. When used in interactive mode, you can use the help command to learn about the command syntax. SFTP can resume interrupted file transfers.

The following example shows how a directory structure is recursively copied from the guest system (IP address 192.168.2.107 in the example) by another system:

```
$ sftp charon@192.168.2.107

charon@192.168.2.107's password:

Connected to 192.168.2.107.

sftp> ls Tmp

Tmp/A  Tmp/B  Tmp/C

sftp>

sftp> get -r Tmp/

Fetching /home/charon/Tmp/ to Tmp

Retrieving /home/charon/Tmp

Retrieving /home/charon/Tmp/B

Retrieving /home/charon/Tmp/B/b

Retrieving /home/charon/Tmp/B/a

Retrieving /home/charon/Tmp/A

Retrieving /home/charon/Tmp/A/b

Retrieving /home/charon/Tmp/A/a

Retrieving /home/charon/Tmp/C

sftp>

sftp> bye
```

Using General Backup and Restore Commands

When transferring larger amounts of data (e.g., when moving application data from one HP-UX system to another), it is more efficient create an archive of the required data before transferring them and unpacking the data on the target system.

HP-UX offers a large number of such programs. For example:

- **cpio**
- **tar**
- **dump/restore** for HFS filesystems
- **vxdump/vxrestore** for VxFS filesystems
- **fbackup/frecover** for selective file backup and restore
- **make_net_recovery** / **make_tape_recovery** when working with Ignite
- **pax** to create and restore archive files of different formats; there is also an **pax_enh** package to support file sizes larger than 8GB in version 11.23 (default in HP-UX 11.31).

Please note:

- The **cpio** and the standard **tar** command on HP-UX have restrictions with respect to the largest file size they can handle: the maximum file size for **cpio** is 2GB, the maximum for **tar** is 8GB.
- The utilities **tar**, **cpio**, and **pax** can be used to transfer archives between HP-UX and other operating systems.

When creating an archive using one of the above commands, you can copy the archive to the target HP-UX system and restore it there. Alternatively, you can pipe the archive command on the source system through **ssh** to the respective restore command on the target.

The following example shows how to use **fbackup**/**frecover** through **ssh** to copy files between two HP-UX systems:

On the source system:

```
# fbackup -i <source-path> -f - | ssh <remote-user>@<target-ip> "(cd <destination-dir>; frecover -r -f -)"
```

The following example shows how to use **pax** through **ssh** to copy files between two HP-UX systems or between an HP-UX system and, for example, a Linux system:

On the source system:

```
# pax -w <source-path> | ssh <remote-user>@<target-ip> "(cd <destination-dir>; pax -v -r -pe)"
```

Please note: Take care not to overwrite any files on the target system! Please read the respective man-pages carefully to learn about additional parameters of the commands and their proper use.

Transferring Files to/from an MPE/iX Guest

If you have enough disk space on the MPE/iX guest, you can copy existing MPE/iX software, data, and settings to the guest MPE/iX system. Charon-PAR comes with MPE/iX 7.5 already installed and patched. So you avoid the most difficult part of setting up a new system.

Please note:

- Charon-PAR comes with MPE/iX 7.5. Note that there are significant restrictions for earlier MPE/iX releases. Charon-PAR may not run with such releases. When using the provided system disk, you must not overwrite any MPE files with copies from earlier MPE releases. As a general rule, specify the `;KEEP` keyword on `:RESTORE` commands, especially when using `STORE` tapes that may contain `PUB.SYS` files. It is strongly recommended to use the MPE/iX 7.5 disk provided by Stromasys. Should you have different requirements, please discuss them with your Stromasys representative.
- This section only describes general methods for file transfer. The specific data transfer required to migrate a physical system to an emulated system (e.g. copying the account structure) is not covered by the user's guide. To obtain support for system migration, contact your Stromasys representative or Stromasys partner.

File Transfer Options

The traditional methods used to move files between systems are available also available for Charon-PAR emulated systems.

When using the original vendor or third-party Store, you should specify the `;COMPRESS` (or equivalent) keyword to ensure that the resulting archive files are as small as possible. Compression can reduce space required to store TurboIMAGE files quite considerably. The `;PARTDB` keyword is also recommended, to include any partial database files you may have.
To ensure that all files in the Hierarchical File System (HFS) are included store the " / " fileset; Store also interprets the " @.@.@ " fileset as " / ".

Physical Tapes

There are two ways you can connect a tape device to a Charon-PAR host system. If you build your server with a SCSI card, you can connect any SCSI tape device that works with MPE/iX. You can also connect a HP USB DAT 72 drive to the Charon-PAR host system which offers a low-cost solution if you currently use DDS3 or DDS4 media.

Any original vendor or third-party backup product may be used to restore files on the Charon-PAR guest system.

Using FTP

FTP can be used to transfer individual files. However, it is far more efficient to use it in conjunction with any STORE product that can store-to-disk, preferably compressed. Use FTP to transfer the resulting archive file(s) to the MPE/iX guest system and restore-from-disk to extract the contents. Restore-from-disk results in files being created with correct attributes and security settings which is not necessarily the case when transferring individual files with FTP. This method does require enough free disk space on both systems to hold the store-to-disk archive file(s). If disk space is tight, individual accounts (or even groups) can be processed instead of storing the whole system. Any original vendor or third-party backup product may be used to restore files from disk on the Charon-PAR guest system.

Other Methods

When available, any other MPE/iX file transfer can be used, e.g., :DSCOPY , Samba, or NFS.

Cloud-specific File Transfer Options

Depending on the customer requirements, the configuration of the original system, and the amount of data, different data transfer options may have to be applied. The various cloud providers offer add-on services to facilitate the transfer of large amounts of data from the customer premises to the cloud instances. This section provides a brief list of such services. However, these services are independent of the Stromasys product offering. So always refer to the documentation of your cloud provider for up-to-date information.

- **Microsoft Azure data transfer offering**: for large data transfers, Azure offers special data transfer services.
  Please refer to the description of the Azure Data Transfer Solutions for more information.
- **OCI data transfer offering**: for large data transfers, Oracle offers its Data Transfer Services.
- **AWS data transfer offering**: for large data transfers, Amazon offers a special service, AWS Snowball.
- **Google data transfer offering**: for large data transfers, Google offers its Data Transfer services.
- **IBM data transfer offering**: for large data transfers IBM offers its Mass Data Migration services.

Upgrading the Charon-PAR Software

This section describes how to upgrade the Charon-PAR software to a new version. It covers the following points:

Please note:

- Upgrading to this Charon-PAR from an older version requires a license update. Please contact your VAR or Stromasys representative.
- Charon-PAR version 3.0.0 and higher is incompatible with HASP licenses of earlier versions. Earlier product versions are incompatible with licenses for Charon-PAR 3.0.0 and higher. Charon-PAR licenses of version 2.1.0 are not compatible with emulators running Charon-PAR version 2.0.0.
- VE licensing is supported starting with Charon-PAR version 3.0.6.

Upgrading Charon-PAR Using RPM packages

Charon-PAR Software Packages Overview

Charon-PAR is provided as a **set of rpm-packages**.

In **version 3.0.10**, Charon-PAR is provided as **one self-extracting archive** containing **all** the RPM packages:

```
charon-par-<version>.sh
```

Starting with **version 3.0.11**, Charon-PAR is provided as **one self-extracting archive per Linux major version**: one archive each for Linux 7.x, 8.x, and 9.x.

- `charon-par-<version>.el7.sh`
- `charon-par-<version>.el8.sh`
- `charon-par-<version>.el9.sh`

To **unpack the archive**, perform the following steps:

- Go to the directory where the package is stored.
- Run the shell script: `# sh <archive-name>`
- Accept the EULA. To successfully unpack the archive, the end-user license agreement must be accepted.
- After this, the software packages will be extracted into a version-specific sub-directory of the current working directory of the user.

Please note that **gzip**, **md5sum**, **cksum**, **gpg**, **tar**, and **openssl** are required to successfully unpack the RPMs.

For o**lder versions**, the individual RPM packages are provided.

The following table lists the required installation packages (the placeholder <*version*> denotes the version of the package, the placeholder <linux-version> denotes the version 7, 8, or 9 of the supported Linux distributions):

Package	Description
aksusbd-<*version*>.i386.rpm	Sentinel runtime environment. Required for licensing the Charon-PAR software via HASP licenses (not required for VE licenses).
charon-mtd-<*version*>**.**<*linux-version*>**.x86_64.rpm**	Charon-PAR tape utility
charon-par-<*version*>**.**<*linux-version*>**.x86_64.rpm**	Charon-PAR emulator executable, template configuration files, etc.
xhpterm-free-<*version*>**.**<*linux-version*>**.x86_64.rpm**	Charon-PAR emulator executable, template configuration files, etc.
charon-license-<*version*>**.x86_64.rpm**	Included **starting with version 3.0.10**. Mandatory package. End-user license agreement text will be installed in */opt/charon/doc.*
charon-license-hasp-<*version*>**.**<*linux-version*>**.x86_64.rpm**	Included **starting with version 3.0.10.** HASP license management utilities and libraries. Previous versions included these utilities and libraries in the Charon-PAR kit, which could cause installation conflicts with other Charon products.
charon-license-certs-<*version*>**.x86_64.rpm**	Sample certificates for the new custom certificate feature of the VE license server. Support planned for **Charon-PAR 3.0.11** and later.

Obtaining the Charon-PAR software: Stromasys will provide the download information or the software packages to you.

Upgrade Commands Overview

The following table provides an overview of the upgrade commands for the supported host operating systems. For details, please refer to the relevant man-pages on Linux. The table only lists command-line options for Linux. There are also graphical tools. To describe all of them is outside the scope of this document.

Upgrade Options	Description
Package manager (uses repositories, takes care of dependencies, etc.)	`# yum update <package-name>` On Linux 8.x and Linux 9.x replace **yum** with **dnf**.
Command to upgrade individual local packages	`# rpm -U </path/to/package>`

Upgrading the Charon-PAR Software

To install the Charon-PAR packages perform the following steps:

	Step	Command
1	Log in as the privileged user on the system ("root").	
2	Go to the directory in which the rpm packages were stored.	`# cd<path-to-kits-dir>`
3	Update the Charon-PAR packages.	`# yum update *.rpm` (Linux 7.x) `# dnf update *.rpm` (Linux 8.x and 9.x) or `# rpm -U *.rpm`

Cloud Image Upgrade

The Charon emulator software running on a Charon host installed from a cloud-specific marketplace image can be updated **using RPM packages** as shown above.

Alternatively, **after careful evaluation**, such a host can be upgraded by launching a new Charon host instance from the Charon marketplace image.

This may be useful if

- all Charon instance files (e.g., vdisks and ISO images) are on a separate disk storage volume that can easily be moved to a new instance,
- the overall configuration of the Charon host system is not very complex, i.e., can be recreated without much time and effort,
- a major host operating system upgrade is required.

Steps (only meant for illustration - the details could vary depending on the customer environment):

- Create an instance with the new Charon version in the same subnet as the old instance (same security group and key pair as old instance).
- Shut down guest systems and stop running emulator instances on the old host system.
- Back up disk containers used for emulated disks and the configuration data on the old instance to the separate disk volume.
- If applicable, back up important system configuration files to the separate disk volume.
- Stop the old instance.
- Move the disk volume(s) and (if applicable) static IP addresses to the new instance.
- Import the emulated system configurations on the new system.
- Adapt the host system configuration as needed.
- Start the guest systems.
- If everything works, terminate the old instance.

Please refer to your cloud provider's documentation for cloud-specific details.

Deinstalling the Charon-PAR Software

If the Charon-PAR software is no longer required, it can be deinstalled following the description below.

RPM Package Deinstallation

Software Deinstallation Commands

The following table provides an overview of the deinstallation commands for the supported host operating systems. For details, please refer to the relevant man-pages on Linux. The table only lists command-line deinstallation options for Linux. There are also graphical tools. To describe all of them is outside the scope of this document.

Deinstallation Option	Description
Package manager (uses repositories, takes care of dependencies, etc.)	`# yum erase <package-name>`
Command to remove individual local packages	`# rpm -e </path/to/package>`

Deinstalling the Software

The following table provides an overview of the steps required to remove the Charon software:

	Step	Command
1	Log in as the privileged user on the system ("root").	
2	Shutdown all running guest operating systems and running emulators.	

3	Remove the Charon-PAR packages (assuming that no other Charon products are installed).
	Please note: The packages **aksusbd**, and **charon-license-hasp** may also be used by other Charon products. Make sure it is safe to remove them!
		`# yum erase aksusbd charon* xhpterm-free`
		or
		`# rpm -e aksusbd charon* xhpterm-free`

Software Deinstallation on Cloud-Specific Images

You can deinstall the Charon RPM packages as described above.

To permanently remove your Charon cloud instance, select your instance from the list of active virtual machines and select to **terminate** or **delete** the instance. Please refer to your cloud provider documentation for details about this operation.

This will stop the instance and remove it. Unless your data (configuration files, vdisk containers, etc.) was stored on a separate disk volume, it will also be removed (this may be dependent on cloud-specific settings).

Please note: make sure you backup any data you wish to retain before terminating an instance.

License Management Overview for Charon-PAR

A license can be installed locally on the system or—in case of a network license—it can be served to clients on the network by a license server. To use the products, you must have **one of the following**:

- Valid physical license key (HASP dongle)
- Valid Sentinel HASP software license
- Valid VE license

Please note: when configuring the emulator, VE licenses and HASP licenses are **mutually exclusive**. If a configuration file contains the definition of a VE license, any HASP license definition is ignored (that is, the VE configuration takes precedence).

The complete licensing documentation can be found in the Licensing Documentation section of the Stromasys Product Documentation page. Please refer to this documentation for any topics not covered in this overview guide and for more detailed information on the topics discussed here.

This page provides an overview of the basic steps to set up the licensing environment for a system running Charon-PAR for Linux using either **Sentinel HASP** licensing, or **Virtual Environment (VE)** licensing.

General Licensing Aspects

Creating and updating a license prior and during the operation of Charon emulators involves the following basic steps:

1. Generate a C2V (customer-to-vendor) file. It contains the data required to generate a license.
2. Submit the C2V information to Stromasys Orders Administration.
3. Receive one or more V2C (vendor-to-customer) license file. It contains the new or updated license data.
4. Apply the V2C files to the local system or the license server.

The following sections describe how to perform these tasks using the different tools available.

Sentinel HASP Licensing

General Description

Please note:

- Upgrading to this Charon-PAR from an older version requires a license update. Please contact your VAR or Stromasys representative to plan the update.
- Applicable to HASP licensing only: Charon-PAR version 3.0.0 and higher is incompatible with licenses of earlier versions. Earlier product versions are incompatible with licenses for Charon-PAR 3.0.0 and higher. Charon-PAR licenses of version 2.1.0 are not compatible with emulators running Charon-PAR version 2.0.0.

A Charon emulator product will not run without a valid license. This license can be provided on a USB hardware dongle (a Sentinel HASP key) or as a software license bound to the hardware of the host system or the network license server. A loss or defect of a license will cause the emulator to stop.

To check the validity of a license, Charon performs the following steps:

- The Charon emulator checks the text block of the license to determine if it is allowed to run (if there is a product section which corresponds to the emulator).
- Then Charon checks the feature associated with the product to see if the feature is valid - that is, not expired, runtime not counted to 0, and if there are still some instance-slots available to run the emulator.
- If there is no matching product section or if the feature is not valid, Charon reports the problem in the log and shuts down.

Charon products **check the availability of a valid license** under several conditions:

1. At startup:

If no license is found, an error message will be written to the emulator log file and the emulator will not start.

In some emulator products it is possible to configure the number of retries and the waiting time between them by adding parameters to the emulator configuration file. Please refer to the product documentation for the details regarding the relevant parameters: `license_retry_period` and `license_retry_count` parameters (obsolete starting with version 1.10)

2. At regular intervals during the runtime of the emulator (the default license check period of 1 hour can be changed by Stromasys using the appropriate license parameters):

- If the previously used valid license has disappeared, is defect, or has become invalid, the emulator will report the missing license in the log file and continue operation for a limited amount of time as described below.
- If there is another valid license, for example a backup license defined in the configuration file, it will be used.
- *Starting Charon-PAR 1.10*, Charon allows for a grace period of 12 hours during which the software checks for the presence of a valid license every 10 minutes until a valid license is found. If no valid license is found after the grace period has expired, the emulator will stop.
- *Earlier versions*: should there still be no valid license at the next regular license check (this default interval may be changed by individual Charon products), the emulator will stop.

- If a time-restricted license is used and it expires, the Charon instance tries to find its replacement automatically and, if found, proceeds using the replacement license.

Under certain circumstances existing licenses must be updated.

License updates are required for:

- Charon product version upgrade
- Product change
- License validity (time) extension

License updates, for example a new expiration date or a new execution counter, can be performed without interrupting operation, that is, without shutting down active Charon instances, **provided that the resulting license is valid for the running Charon instance**. If any parameters of the updated license are not valid for the running Charon instance (for example, the wrong model or version), the Charon instance will stop running.

Licenses are **backwards compatible to some extent**. A license **update to a more recent version** will allow some previous Charon versions to run as well. The details depend on the Charon product:

The license check for Charon-PAR will pass if the version on the license is higher than the one of the running Charon instance, **as long as** the major version (e.g., the 2 in 2.00) is the same. Please note that license are not compatible between product versions 1.x, 2.x, and 3.x.

License Type Overview

Depending on customer requirements, Stromasys can provide several different license types.

Local Hardware License

Local hardware licenses are USB dongles and work on the system they are physically connected to. For Windows systems supporting AnywereUSB®, hardware licenses can also be connected over the network. They can easily be moved to a different system, if required. The content of the dongle can be updated if a change to the license or an extension of a time-based license is required. The necessary steps will be described later in this document.

Please note:

- Hardware dongles require the Sentinel HASP run-time (driver) installation before the dongle can be connected to and used by the system.
- Hardware dongles, apart from HL-MAX dongles, are equipped with a **battery** and a clock, which makes them independent of the host clock. The battery is not rechargeable. However, the dongle can use the power provided by the host system while it is plugged in. By doing this, the depletion of the battery can be slowed down. Check the dongle at regular intervals if it is not permanently connected to a system. If the battery becomes completely depleted, the dongle will be permanently unusable and must be replaced. See also: How long does the license USB dongle battery last upon a full charge.

Software License

A software license is a "virtual" key with functionality very similar to a HASP network-enabled hardware dongle.

A software license does not require any special hardware but it still **requires installation of the Sentinel runtime environment**.

Please note:

- To avoid unexpected problems, do not use any Sentinel runtime software that was not provided by Stromasys without being advised to do so by your Stromasys representative.
- Software licenses are best suited for stable environments, because their correct function depends on certain characteristics of the host system. Changing any of these characteristics will invalidate the license.
  - If the Charon host runs on real hardware, software licenses are by default **tightly bound to the hardware** for which they were issued. If major hardware characteristics of the system are changed, the license will be disabled.
  - If the Charon host runs in a **virtual environment** (e.g., VMware), software licenses are normally bound to the virtual machine ID and a set of additional characteristics of the virtual machine. If any of these parameters are changed, the license will be disabled.
- Software licenses are very sensitive to even small changes on the host system. Therefore, it is especially important to provide for a backup license that will ensure continued operation should there be a problem with the software license. See *Handling Multiple License Keys and Product Licenses* for details.

For a more detailed description of the restrictions, please refer to Software Licensing restrictions or contact your Stromasys representative.

Network License

The network Sentinel HASP key (**red USB dongle**) can be shared between several hosts running a Charon emulator product (including the host on which the network license is installed).

All **software licenses** are also network licenses.

If the Charon emulator product is installed on the host where the network license is connected, no additional steps are required. The Sentinel driver is installed as part of the Charon product installation. If the host does not have a Charon emulator product installed, the host can still distribute the connected network license to emulator instances running on other hosts.

The Sentinel HASP runtime software must be installed on such a "license server" and on the client system. For details regarding the installation, please refer to the software installation section in this document. Once both the Sentinel runtime software and the network license are installed, the Charon emulator product can be started on any appropriate client host on the LAN, provided access to the license is enabled.

- The network license will be visible to all hosts that can access the license server over IP. Access to the license server must be possible on **port 1947** v ia **UDP** (discovery process) and **TCP** (actual access to license). Further information (e.g., use of additional UDP ports) can be found in the section Firewall Considerations.
- The license server and the client must both allow access to the network license using the appropriate management tools.

The maximum **number of concurrently active** Charon instances is determined by the parameters of the license.

If you need to install a **standalone license server**, please refer to the installation section of the Licensing Handbook.

Overview of Initial License Installation Steps

Each Charon emulator product requires a valid license to run. To access the license, the emulator needs the Sentinel runtime software. This software is part of the Charon-PAR for Linux installation kits. Refer to the software installation section of the Charon product user's guide for software installation instructions.

After the installation of this software, the license can be installed on the system. The following steps will be described in more detail below:

1. Add a password for the Sentinel Admin Control Center (ACC).
2. If you purchased a **hardware license**, you can simply plug the dongle into a free USB port on the system.
3. If you purchased a **software license**, you need to create a fingerprint file in C2V (customer-to-vendor) format containing the system characteristics. Use this file to request a license for your Charon product from Stromasys.
4. If your license is a **network license** served by a license server, make sure that the access of the client system to the license server is not blocked by the configuration or a firewall.
5. Optionally, define how **multiple licenses** will be used (selecting primary/backup license, defining license priorities).
6. If you have an **existing license** that needs to be updated, you need to create a customer-to-vendor (C2V) file and use this file to request a license update from Stromasys.

Overview of License Management Tools

The following list shows the main tools used to manage licenses on Linux:

- **Sentinel Admin Control Center (ACC)**: A web-based interface providing important configuration options with respect to licenses.
- The **hasp_srm_view** program: A command-line tool to display the detailed license contents and generate C2V and fingerprint files. Cannot be used over a remote connection when using local hardware licenses.
- The **hasp_update** program: A command-line tool to install new and update existing licenses.

Accessing the License Management Tools

The command-line tools are installed under `/opt/charon/bin/`. If this directory is not part of your PATH variable, you have to specify the full path to access the command.

To run the commands, use the following syntax for **hasp_srm_view** and **hasp_update** respectively.

```
# /opt/charon/bin/hasp_srm_view <option>

                                   and

# /opt/charon/bin/hasp_update <option>
```

The relevant options will be specified with the tasks described below, as needed.

The **Sentinel ACC** on the local system is accessed by starting a web-browser and pointing it to the URL: **http://localhost:1947**.

Sentinel Admin Control Center (ACC) Security Settings

Stromasys strongly recommends performing the steps described below to reduce the risk of unauthorized access to the Sentinel ACC.

Define a Password for the Sentinel ACC

By default, anyone on the local system with access to port 1947 can access the GUI. If remote access is enabled, users on the network with access to port 1947 can also access the GUI. To protect access to the GUI with a password, perform the following steps:

Step	Description
1	Open a web browser and navigate to **http://localhost:1947/.**
2	**Click** on the left-hand menu item labeled **Configuration**.
3	**Click** on the **Basic Settings** tab.
4	Under the entry labeled **Password Protection, click** the **Change Password** button.
5	At the **Change Password** window: • Leave the **Current Admin Password** field blank (there is no password set by default). • Enter the desired password into the **New Admin Password** field. • Repeat the desired password in the **Re-enter new Admin Password** field**.** • **Click** the **Submit** button**.**
6	Back at the **Basic Settings** tab: • Under the section labeled **Password Protection,** select the **All ACC Pages** radio button**.** • **Click** the **Submit** button to save this change.
7	Optional: to allow **remote access** to the Sentinel HASP GUI: • **Click** the **Basic Settings** tab. • **Select** the **Allow Remote Access to ACC** check box. • **Click** the **Submit** button**.** **Additional information:** For remote access to the Sentinel HASP GUI, • the firewall must permit access to port 1947/TCP, and • network visibility on the ACC network configuration tab must be set to **All Network Adapters.** If required, remote access can also be enabled on Linux by editing the file */etc/hasplm/hasplm.ini* and setting the parameter **ACCremote** to **1** . Should the file not yet exist, refer to the Sentinel ACC selected *hasplm.ini* parameters chapter in the *Tools Reference* section of the licensing handbook.

Please note: With these settings, when you connect to the HASP GUI from a remote system, you may be prompted for a username and password. It is enough to just enter the configured password and leave the username field empty.

Setting Linux File Protections

To prevent unprivileged access and modifications to the Sentinel HASP configuration file on Linux, enter the following commands:

```
# chmod 0700 /etc/hasplm

# chmod 0600 /etc/hasplm/*
```

Please note:

- The file **hasplm.ini** on Windows is readable by normal users but cannot be modified. Password information is encrypted.
- Should it not be possible to open a local browser on a **Linux server**, the remote access setting can also be modified by editing the file `/etc/hasplm/hasplm.ini` and changing the value of the parameter **ACCremote** from **0** (access disabled) to **1** (access enabled). The parameter to enable network visibility is **bind_local_only**. It must be set to **0**. Changed settings are recognized automatically by **aksusbd**.

Viewing Existing Licenses

It is important to know which licenses are visible on a system. For example, the user can

- determine if the correct license is installed,
- identify the expiration date on time-based licenses,
- identify the remaining hours of run-time on backup licenses,
- identify the license ID of primary and backup license which are needed for the emulator configuration file,
- identify licenses that conflict with the currently used product and therefore may need to be removed.

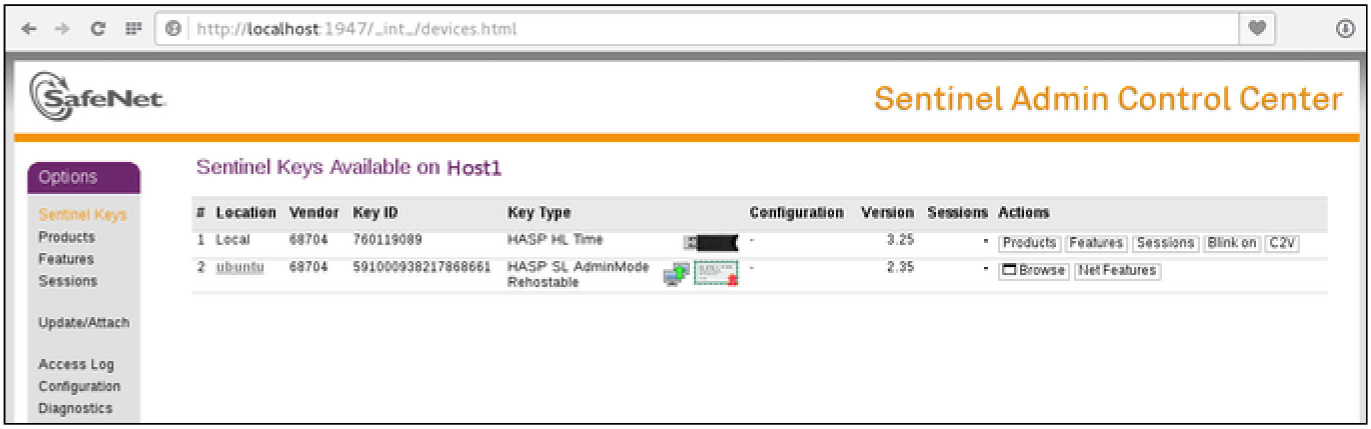
Licenses can be viewed using two tools:

- Sentinel ACC: shows important information, but not the product specific license parameters.
- Command-line tool **hasp_srm_view**: shows all product details contained on the license. Can only be run from a local connection for local hardware licenses.

Viewing a License with Sentinel ACC

To view available licenses using Sentinel ACC, start the web interface as described above.

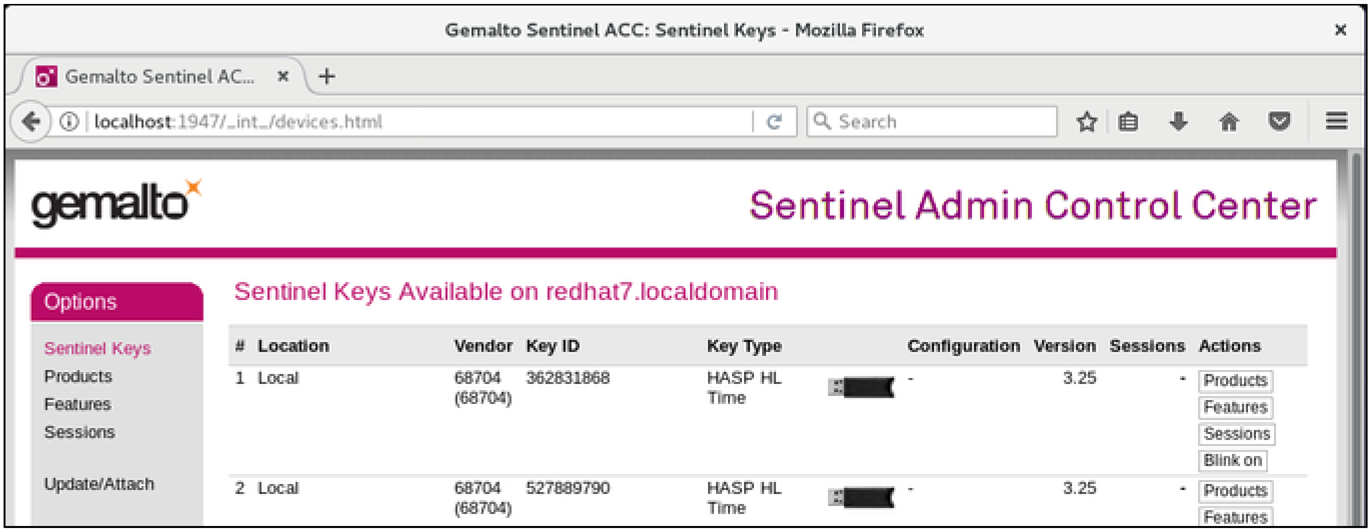
To get to the Sentinel Keys screen, click on the corresponding menu item or access the URL **http://localhost:1947/_int_/devices.html** directly. A screen similar to the following opens and displays the available license keys:



Sentinel Keys Available on Host1

#	Location	Vendor	Key ID	Key Type	Configuration	Version	Sessions	Actions
1	Local	68704	760119089	HASP HL Time	-	3.25	-	Products Features Sessions Blink on C2V
2	ubuntu	68704	591000938217868661	HASP SL AdminMode Rehostable	-	2.35	-	Browse Net Features

For Sentinel drivers version 7.60 and above, this screen looks similar to the following:


Gemalto Sentinel ACC: Sentinel Keys - Mozilla Firefox
Gemalto Sentinel AC...
localhost:1947/_int_/devices.html
Search
gemalto
Sentinel Admin Control Center
Options
Sentinel Keys
Products
Features
Sessions
Update/Attach

Sentinel Keys Available on redhat7.localdomain

#	Location	Vendor	Key ID	Key Type	Configuration	Version	Sessions	Actions
1	Local	68704 (68704)	362831868	HASP HL Time	-	3.25	-	Products Features Sessions Blink on
2	Local	68704 (68704)	527889790	HASP HL Time	-	3.25	-	Products Features

This page displays important information about the available licenses, including

- **Location**: Shows whether the license is local or remote. If the license is a network license, the hostname of the remote system is specified. You can access the remote license manager by clicking on the hostname, provided that access is permitted.
- **Key ID**: The unique identification of the license.
- **Key Type**: Hardware keys are marked by the abbreviation HL (hardware license) and a picture of the license dongle. Software licenses are marked by the abbreviation SL (software license).
- **Sessions**: Shows the number of active sessions opened for the specific key.

The buttons on the right-hand side can be used to retrieve more information about the license or to extract the C2V file for a license update. The **Browse** button shown for network licenses will connect to the remote license manager to show the license features.

A C2V file can only be extracted if the license in question is local to the current license manager, i.e., the license manager to which the web browser is connected. In the example above, the network license on a different host does not have the option to create a C2V file. You can connect to the remote license manager clicking on the hostname in the **Location** column (if connections are allowed).

Please note:

- The option to create a C2V file is not available in older versions of the Charon emulator software.
- Starting with ACC version 7.60, the option to create C2V files for USB dongles is only available if it has been enabled in the basic configuration section (under the *Configuration* menu item).

The menu options *Products*, *Features* and *Sessions* on the left-hand side provide the same information as the buttons. However, they show the information for all licenses.

Viewing a License with hasp_srm_view

On Linux, the license content is displayed using the **hasp_srm_view** command. For displaying the license, the following parameters are relevant:

- Display the default license: run the command without options or with **`-l`**
- Display all licenses: run the command with the option **`-all`**
- Display a license with a specific ID: run the command with the option **`-key`**

Please note: Local hardware licenses can only be displayed from a local connection to the system, for example via the console. If you are connected via a remote connection, for example via **ssh**, the **hasp_srm_view** command will return an error. Network licenses do not have this problem. A workaround is described below.

Workaround when logged in via a remote connection:

When connected to the system via a remote connection, the command to display a local hardware license will return an error. As a workaround, you can display the license contents with the following command (adapt the path of the command if your installation location is different):

```
$ ssh localhost /opt/charon/bin/hasp_srm_view
```

Please note: Starting with Charon-PAR 1.10, the hasp_srm_view utility on Linux does not follow the settings in the Sentinel ACC with respect to querying remote license servers and network visibility. The utility performs a broadcast search for network licenses even if this has been disabled in the Sentinel ACC. If this behavior has to be prevented for specific reasons, the network access of the system must be temporarily restricted or disabled, for example by blocking the relevant traffic with a firewall. Alternatively, access to the network license at the license server side can be blocked. Note that such methods can negatively impact other functions of the system or, in case of blocking access to a network license on the server, even the functions on other client host systems.

The following shows sample output of the **hasp_srm_view** command on Linux (to display all available licenses, use the **`-all`** parameter):

```
$ hasp_srm_view
License Manager running at host: host1.example.com
License Manager IP address: 127.0.0.1

The Physical KeyId: 760119089
CHARON Sentinel HASP License key section
Reading 4032 bytes
The License Number: 1000.639
The License KeyId: 760119089
The Master KeyId: 2131943932
Release date: 09-JUN-2016
Release time: 14:33:59
Update number: 6
End User name: Stromasys - User1
Purchasing Customer name: Stromasys SA
Virtual Hardware: AlphaServer_DS10, AlphaServer_DS10L, AlphaServer_DS15, AlphaServer_DS20
Instances allowed: 5
Product Name: CHARON-AXP
Product Code: CHAXP-470xx-WI-LI
Major Version: 4
Minor Version: 7
Maximum Build: 99999
Minimum Build: 1
Host CPU supported: X64
Host Operating System required: WINDOWS, LINUX
CPU's allowed: 32
Maximum virtual memory: 65536MB
Released product expiration date: 12-Jan-2017
--- output truncated ---
```

Starting with license driver versions >= 7.60, for example, in Charon-PAR version 2.00, there are some small changes in the output of **hasp_srm_view** . The parameters are described in the License Content section of the Licensing Handbook. The following sample shows the most important changes (in blue):

```
License Manager running at host: Host1

License Manager IP address: 192.168.2.001

HASP Net key detected

The Physical KeyId: 1015925129

License type: License Dongle (Network capable)

CHARON Sentinel HASP License key section

Reading 4032 bytes

The License Number: 000.TEST.CENTER

The License KeyId: 1015925129

The Master KeyId: 827774524

Release date: 11-APR-2018

Release time: 16:52:33

Update number: 68

End User name: STROMASYS

Purchasing Customer name: STROMASYS

Product License Number: AXP_4_9_TEST

Virtual Hardware: AlphaServer_DS10, AlphaServer_DS10L, AlphaServer_DS15, <items removed>

Product Name: CHARON-AXP

Product Code: CHAXP-490xx-LI

Major Version: 4

Minor Version: 9

< Parameters Maximum Build and Minimum Build are no longer shown>

32bit and 64bit OS requirements: 64bit  <replaces parameter Host CPU supported>

Host Operating System required: LINUX

CPU's allowed: 16

Maximum virtual memory: 65536MB

CHAPI enabled

License expiration date: 01-Nov-2018

Instances allowed: 3

Feature number: 1

Check interval: 15 minutes
```

Installing the License

This section provides a short overview of the initial license installation. For more in-depth information, please refer to the licensing handbook.

Please note: If a conflicting or obsolete license is visible to the system, it can be **(temporarily) removed or disabled**. If you need to remove a hardware license, simply unplug it. If you need to remove a software license or disable access to a network license, please refer to the relevant chapter in the licensing handbook. Before removing a license or disabling access to it ensure that it is not required by another currently active product.

Installing a Local Hardware License

A local hardware license (USB dongle) is installed by inserting the USB license key into a free USB port of the host system.

After this step, verify that the license is visible to the system by following the steps for viewing a license as described above. Please bear in mind that a local hardware license cannot be read when connected to the system via a remote connection (for example, ssh).

Installing a Software License

To install a new software license, perform the following steps:

Step 1: Create a fingerprint file using the **hasp_srm_view** command:

Execute the following command to create the fingerprint file:

```
# /opt/charon/bin/hasp_srm_view -fgp <filename.c2v>
```

The fingerprint will be written to the filename specified.

Please note:

- **Charon-PAR before 1.10**: the above command will terminate with the error message "`Can not retrieve the C2V (host fingerprint mode) data`" or with "`Can not retrieve the C2V (host fingerprint mode) data`" if a network-wide software license is visible to the system. Access to such a license needs to be temporarily disabled before creating the fingerprint file.
- **Charon-PAR versions 1.10 and higher** can create a fingerprint file even in the presence of a network-wide software license.

Step 2: Send the resulting fingerprint file to Stromasys orders administration using the email address that Stromasys will provide to you.

Step 3: After receiving the V2C file from Stromasys, copy the file to the system where the license needs to be installed and install the new license:

In addition to the Sentinel ACC, the command **hasp_update** can be used on Linux to apply V2C files.

The following example shows the use of the **hasp_update** command:

```
# /opt/charon/bin/hasp_update u /path/filename.v2c
```

This section describes the installation of a new software license. However, the commands to install a V2C file are identical when updating a hardware license.

Important caveat:

- When updating a hardware license you will in most cases receive two V2C files, a ***_fmt.v2c** file and a ***.v2c** file. The *_fmt.v2c file formats the dongle and the *.v2c file contains the updated license data. In such cases the ***_fmt.v2c file** must be applied **first**.

Installing a Network License

For a network license to be provided to a client host on the network, a license server must have been set up either with a network-enabled hardware license (red dongle) or a software license (software licenses are always network enabled).

If the license server also runs a Charon emulator product, follow the steps in the user's guide to install the software. To set up a standalone license server, please refer to the licensing handbook.

For the client to access a license on a license server the following steps are required:

- Any firewall between license server and client must permit the necessary communication.
- The license server must be configured to allow access from the client.
- The client must be configured to allow access to the license server.

Firewall Considerations

The following ports are used for the communication between license server and client hosts:

- On the **server side** (where network license has been installed), port 1947 must be open for incoming TCP and UDP traffic to allow client access to the license.
- On the **client side**, traffic is initiated using ports 30000 through 65535 as the source ports and port 1947 as the target port. If broadcast search for remote licenses is to be used, the client must also permit UDP traffic initiated from port 1947 of the license server to ports 30000 through 65535 of the client.

If a host on the network cannot find the license server even though the server is operational, you can **temporarily** disable the firewall to determine whether it blocks the traffic.

For details on how to configure the firewall in your network, please consult your operating system documentation and make sure to adhere to your company's security policies.

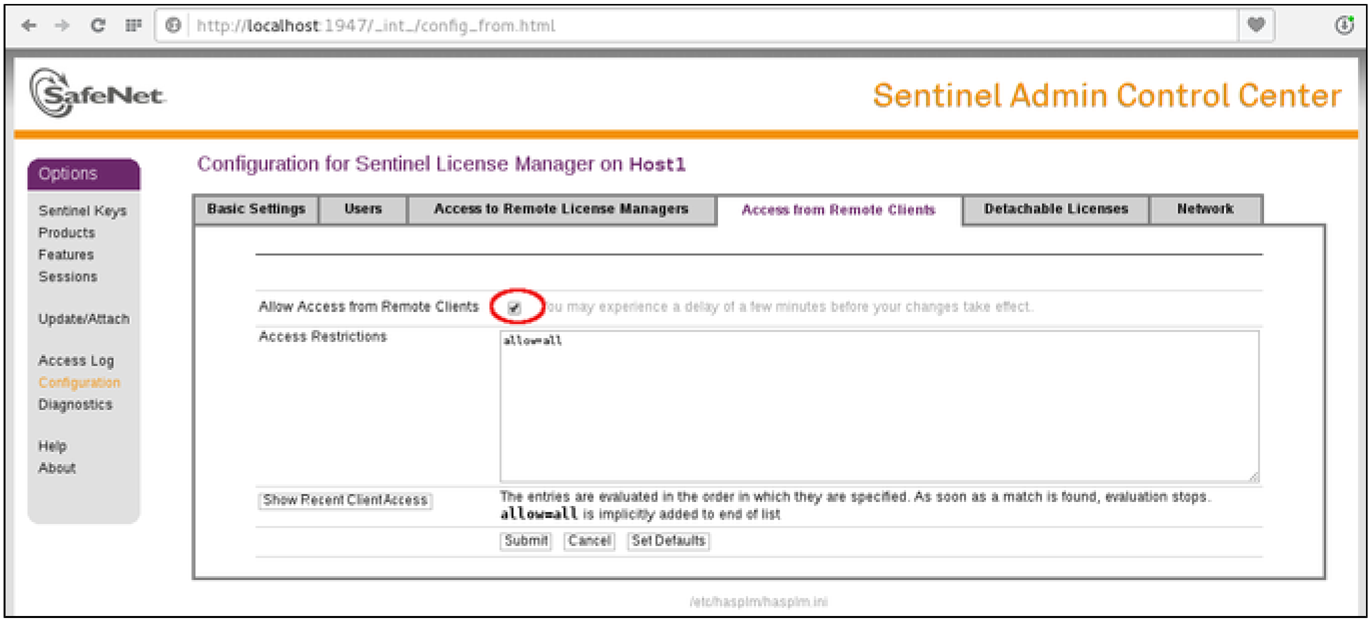
Allowing Client Access on the License Server

The Sentinel license manager on the license server can be configured to **allow** or **disallow** access from remote clients to the network licenses installed on the license server. To access this configuration option, perform the following steps **on the license server**:

1. Open a web-browser and go to the URL **http://localhost:1947/_int_/config_from.html** (option: Configuration / Access from Remote Clients).

2. This will open a configuration page similar to the following. **Please note**: newer Charon emulator products (e.g., Charon-AXP/VAX version 4.9 and Charon-PAR 1.10 and higher) have newer versions of the Sentinel license drivers. The Sentinel ACC pages of these versions look different, but the functionality remains mostly the same.

Old ACC version:



Configuration for Sentinel License Manager on Host1

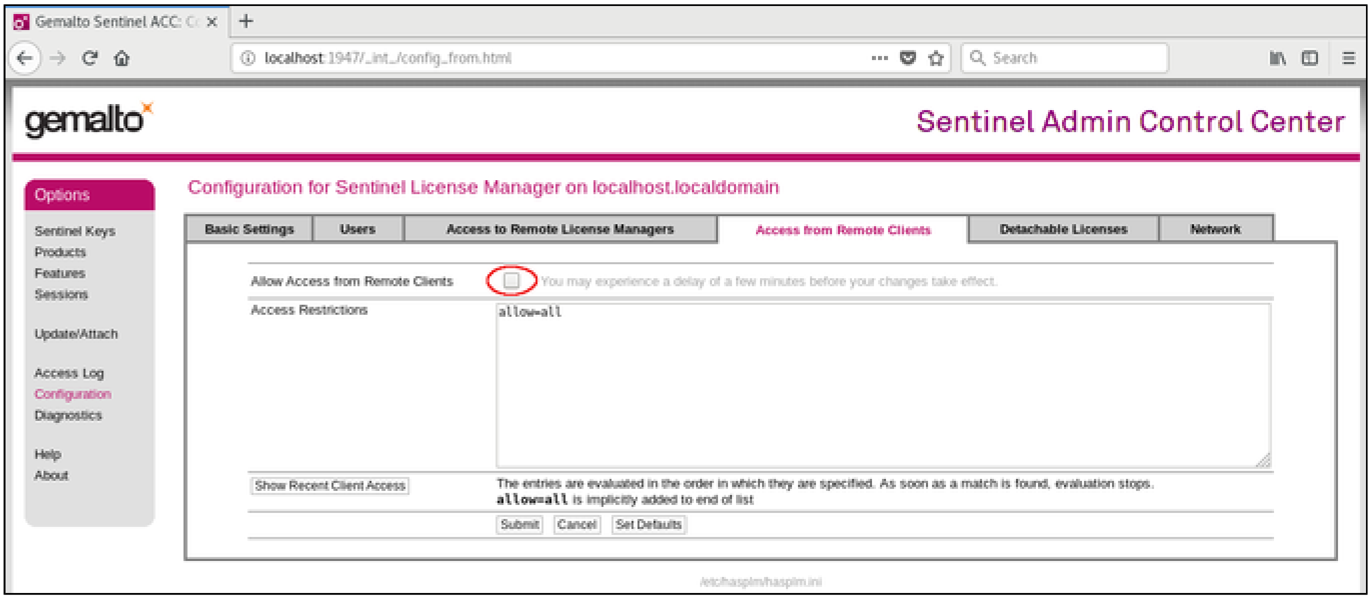
Allow Access from Remote Clients ☑ You may experience a delay of a few minutes before your changes take effect.

Access Restrictions `allow=all`

The entries are evaluated in the order in which they are specified. As soon as a match is found, evaluation stops. **allow=all** is implicitly added to end of list

/etc/hasplm/hasplm.ini

New ACC version:



Configuration for Sentinel License Manager on localhost.localdomain

Allow Access from Remote Clients ☐ You may experience a delay of a few minutes before your changes take effect.

Access Restrictions `allow=all`

The entries are evaluated in the order in which they are specified. As soon as a match is found, evaluation stops. **allow=all** is implicitly added to end of list

/etc/hasplm/hasplm.ini

3. Possible actions:

- To **allow access** from remote clients, *activate* the check-box next to the field **Allow Access from Remote Clients** and press **Submit** at the bottom of the page.
  Please note: to allow access from remote clients, network visibility on the "Network" tab must be set to **All Network Adapters.**
- To **refuse access** from remote clients, *clear* the check-box next to the field **Allow Access from Remote Clients** and press **Submit** at the bottom of the page.
- **Access Restrictions** allow refining access rules, e.g., by specifying IP addresses. Please refer to Sentinel ACC help for details.

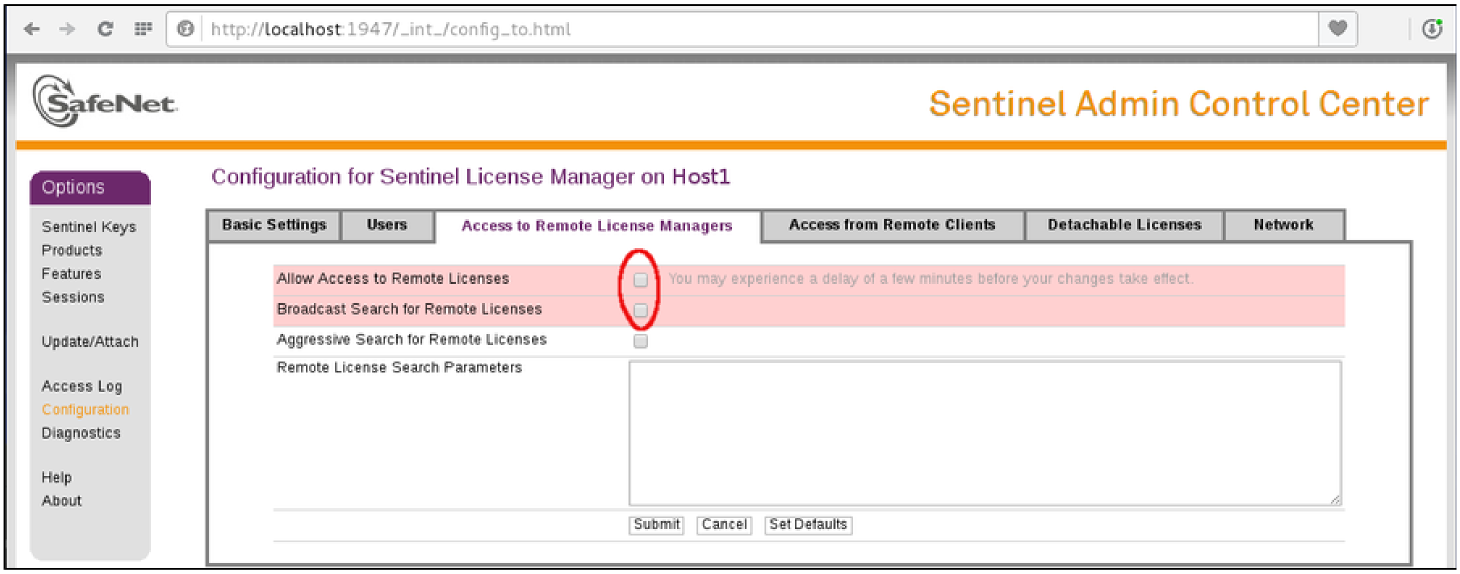
Allowing Access to a License Server on the Client

The Sentinel ACC can be configured to enable or prevent that the client host discovers network licenses and to change the options used to discover and access network licenses provided by a license server.

1. Open a web-browser on the **client host** and go to the URL **http://localhost:1947/_int_/config_to.html**
 (option: Configuration / Access to Remote License Managers).

2. This will open a configuration page similar to the following. **Please note**: newer Charon emulator products (e.g., Charon-AXP/VAX version 4.9 and Charon-PAR 1.10 and higher) have newer versions of the Sentinel license drivers. The Sentinel ACC pages of these versions look different, but the functionality remains mostly the same

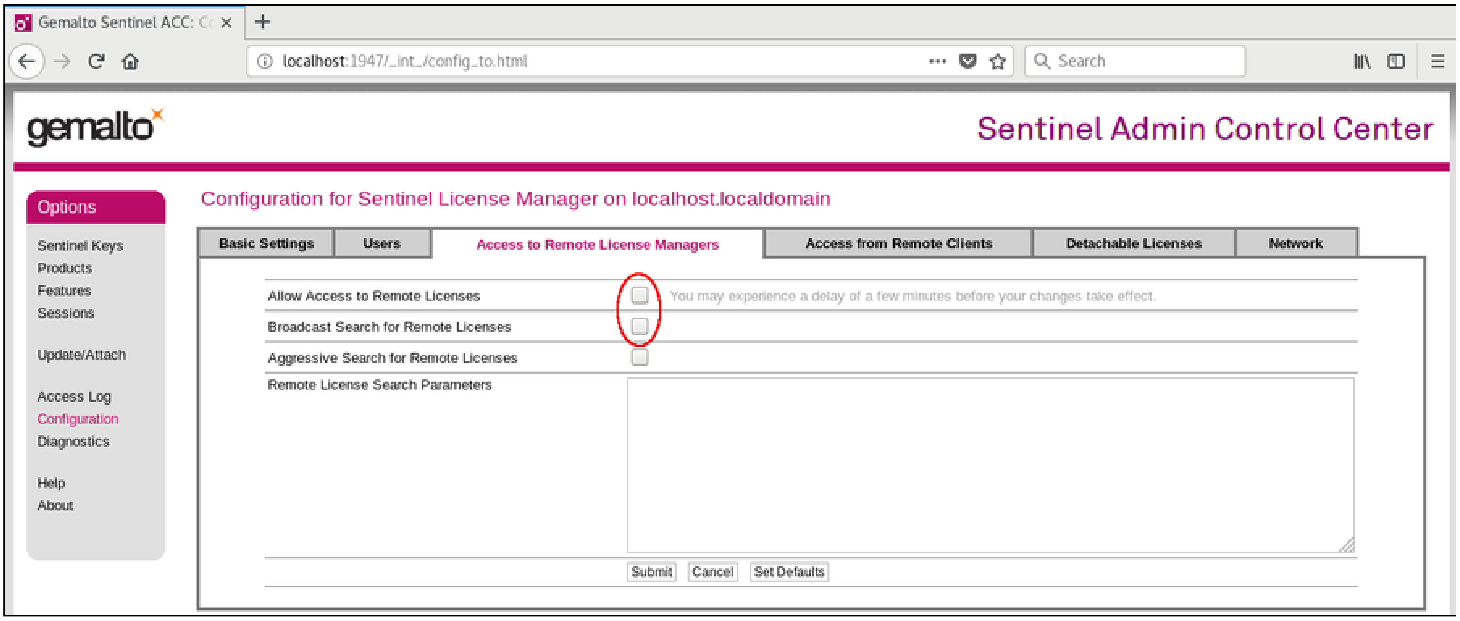
Old ACC version:



Configuration for Sentinel License Manager on Host1

Allow Access to Remote Licenses ☐ You may experience a delay of a few minutes before your changes take effect.
Broadcast Search for Remote Licenses ☐
Aggressive Search for Remote Licenses ☐
Remote License Search Parameters

New ACC version:



Configuration for Sentinel License Manager on localhost.localdomain

Allow Access to Remote Licenses ☐ You may experience a delay of a few minutes before your changes take effect.
Broadcast Search for Remote Licenses ☐
Aggressive Search for Remote Licenses ☐
Remote License Search Parameters

3. Possible actions:

- *Activate* the check-box next to the field **Allow Access to Remote Licenses** to **enable** access to license servers. Press **Submit** to save the setting.
- *Clear* the check-box next to the field **Allow Access to Remote Licenses** to **disable** access to license servers. Press **Submit** to save the setting.
- The option **Broadcast Search for Remote Licenses**, when activated, enables a broadcast search for license servers on the local network without having to enter the address of a license server.

Please note:

- If the option **Broadcast Search for Remote Licenses** is not enabled or cannot be used in the customer specific setting, you can enter specific IP addresses or host names that should be queried for network licenses in the **Remote License Search Parameters** field. Please refer to the Sentinel ACC help function for more information.
- To allow access to remote license managers, network visibility on the "Network" tab must be set to **All Network Adapters**.
- Starting with Charon-PAR 1.10, the Charon emulator products do not follow the settings in the Sentinel ACC with respect to querying remote license servers and network visibility. They perform a **broadcast search** for network licenses even if this has been disabled in the Sentinel ACC. If this behavior has to be prevented for specific reasons, the network access of the system has to be temporarily restricted or disabled, for example by blocking the relevant traffic in a firewall. Another possibility would be to block access to the network license at the license server side. Note that such methods can negatively impact other functions of the system or, in the case of blocking access to a network license on the server, even the functions on other license clients.

Verifying License Availability

After installing a license on the system, verify the availability of the license as described in the section *View Existing Licenses*.

Check if the license shows the correct product, expiration date etc.

Handling Multiple License Keys and Multiple Products per License

A Charon host system can have access to several local and remote license keys. Each license key can contain one or more product licenses.

The Charon emulator products can only use one **active** license at one time. Without additional configuration, they cannot make decisions about which product license or license key to use. If there is more than one available license key, the default license key will be used. The default license key is determined by the Sentinel software. As this can lead to undesirable effects, newer Charon emulator products have configuration parameters that allow the definition of a **primary** (or production) and a **secondary** (or backup) key, or the creation of a **list of license keys or product licenses sorted** by priority.

The following sections describe the options available to achieve a more deterministic license selection. If only one license is available, either locally installed or via the network, this section does not apply.

Please note: For Charon-PAR before version 1.10: the parameters described in this section can **only** be used to define a primary key and a backup key, or to specify the correct key if there is a conflicting license for a different Charon product on the same system. It is **not possible** with these versions of Charon to combine a local license (black dongle) with other licenses (local or network) to increase the number of concurrent instances of the same Charon product on one host system. Newer versions allow a more flexible use of multiple licenses. Both variants are described in the following sections.

Backup License Characteristics

Backup license keys are provided by Stromasys in addition to standard license keys.

It is strongly recommended to order a backup key to recover immediately from damage or loss of the main license key. A backup license key can also help in situations where the Charon host hardware fails and the software must be moved to a different system, thus invalidating the original software license.

Backup keys use a counter (integer) value programmed in the key. This integer value corresponds to a number of hours the Charon software is allowed to run. Each time the Charon software checks the license (at start and then every hour), the value is decreased (by 1 hour). Please note that backup keys have additional restrictions when compared to regular license keys:

- The runtime is typically limited to 720 hours (30 days). This is the time available to get a replacement license from Stromasys.
- A backup license key may be valid only until a certain date.
- If you start and stop the emulator frequently (e.g., frequent runs with a duration of under one hour), the runtime may be significantly less than 30 days, because the license check during the start of an emulator will reduce the counter by one.

Primary and Backup License for Charon-PAR prior to Version 1.10

If more than one key is visible to the system, you can define which is the primary and which the backup key. To do this, add the following parameters to the configuration file of the Charon instance:

```
primary_license <primary-key-id>

secondary_license <secondary-key-id>
```

To identify the relevant key IDs, display the available licenses as described in section *View Existing Licenses*.

To get information about modifying the Charon configuration files, refer to the appropriate sections in the user's guide.

Prioritizing Licenses - Charon-PAR Version 1.10 and Higher

Prioritized List of License Keys

If more than one key is visible to the system, you can define a prioritized list of license keys. To do this, add the following parameter to the configuration file of the Charon instance:

```
license_key_id <key-id prio1>

license_key_id <key-id prio2>     (more than two key ID entries are possible)

license_use_any_key false|true   (optional parameter, default = false
```

The first entry in the configuration file has the highest priority. The emulator instance will try to use this key first. If it is not available, it will try the second key in the list and so on. If the parameter **`license_use_any_key`** is set to **true**, the emulator instance will search for any keys not part of the list should none of the listed keys be available.

To identify the relevant key IDs, display the available licenses as described in section *View Existing Licenses*.

To get information about modifying the Charon configuration files, refer to the appropriate sections in the user's guide.

Prioritized List of Product License IDs

Every license key can contain one or more product sections. Newer licenses can contain a product license ID identifying each product section. A prioritized list of product license IDs can be added to the emulator instance configuration file to specify which product sections to use and which should have the highest priority. To do this, add the following parameter to the configuration file of the Charon instance:

```
license_id "<lic-id-prio1>"

license_id "<lic-id-prio2>"        (more than two entries are possible
```

The emulator instance will scan the available license keys for the listed product license IDs. Then it will try to use the first defined product section. If it is not available, it will try the next and so on. If none of the listed product license IDs are found, the emulator will stop.

Please note: without this parameter, the emulator will try to use the first applicable product section found. If this is not the correct one, the emulator may not start.

To identify the relevant product license IDs, display the available licenses as described in section *View Existing Licenses*.

To get information about modifying the Charon configuration files, refer to the appropriate sections in the user's guide.

Updating an Existing License

If you need to update an **existing hardware or software** license, for example because the time limit on the license has expired or to upgrade to a new product versions, perform the following tasks:

1. Generate the C2V file for the existing license. This step is the same for a hardware license or an existing software license. This Customer-to-Vendor (C2V) file contains the license characteristics necessary for creating the license update.
2. Send the C2V file to Stromasys. Stromasys will use the data to create the necessary license update. You will receive one (software license) or two (most hardware license updates) V2C files (the Vendor-to-Customer file).
3. Apply the license data from the V2C file(s). This will install and activate the update for your license.

These tasks are described below:

Step 1: Generate the C2V file using the **hasp_srm_view** command:

Execute the following command to create the C2V file:

```
# /opt/charon/bin/hasp_srm_view -c2v <filename.c2v>
```

If there is more than one license visible to the system, newer product versions (starting from Charon-PAR 1.10) allow selecting a specific license using the **-key** parameter:

```
$ hasp_srm_view -key <license-id> -c2v <filename.c2v>
```

The C2V file will be written to the filename specified.

Please note: The license content cannot be read if you are connected to the system via a remote connection (e.g., via ssh). The hasp_srm_view command will return an error. You can use the following workaround:

```
$ ssh localhost /opt/charon/bin/hasp_srm_view -c2v <filename>
```

Step 2: Send the resulting C2V file to Stromasys orders administration using the email address that Stromasys will provide to you.

Step 3: After receiving the V2C file(s) from Stromasys, copy the file(s) to the system where the license needs to be installed and install the new license:

Important caveat:

- If a hardware USB key is to be updated, in most cases you will have received **two** files: a ***_fmt.v2c** file and a ***.v2c** file. The *_fmt.v2c file formats the dongle and the *.v2c file contains the updated license data. In such cases the ***_fmt.v2c** file must be applied **first**.

On Linux, the command **hasp_update** can be used to apply V2C files.

The following example shows the use of the **hasp_update** command:

```
# /opt/charon/bin/hasp_update u /path/filename.v2c
```

Alternatively, you can use the Sentinel ACC to apply V2C files (use the section Update/Attach). Refer to the license handbook for more details.

Virtual Environment (VE) Licensing

The Charon emulator kit itself does not include VE license management tools. The following actions **must be performed on the VE license server**. However, the VE license server can be on the same system as the Charon emulator.

Please note:

- If you have not installed the license server yet (and for any more in-depth information), please refer to the VE license server user's guide (see Licensing Documentation).
- The information below shows the **command-line tools** for license management. Starting with version 1.1.16 of the VE license server, these activities can also be performed using a **web-based management GUI**. Please refer to the appropriate VE license server user's guide (see chapter *VE License Server Web-based Management GUI* in the VE license server documentation under Licensing Documentation).

VE Licensing Certificates Overview

This section applies to Charon-PAR version 3.0.11 and later.

This section provides a short overview of the certificates used by the VE license server and Charon-PAR. Please refer to the VE license server documentation for details.

The VE license server uses certificates for different purposes:

- License server operation: encrypted communication between license server and license clients (emulators).
  - New certificate support in the VE license server started with version 2.1.3. Changed certificate names starting with VE license server 2.2.2.
  - New certificate support for Charon-PAR started with version 3.0.11.
- Web-based management GUI: encrypted (HTTPS) communication between the integrated license server web server and web browsers. Starting with VE license server version 2.1.4, the name of the certificate and its management changed. Please refer to the VE license server documentation.

Important information:

- General VE license server configuration:
  - The VE license server will – by default – use the old certificates. Therefore, compatibility with existing Charon clients will be maintained during an upgrade of the license server.
  - If the new certificates (using pre-defined names) are present in */opt/license-server/certs,* these will be used and clients will have to use matching certificates. Please refer to the VE license server documentation for information how to activate the new certificates and, if desired, create custom certificates.
- Checking if the new certificates are enabled in a Charon-PAR installation:
  - Certificate location: */opt/charon/bin/certs*
  - Sample certificate names: *ca.crt.sample*, *charon.crt.sample*, and *charon.key.sample*
  - If the directory contains the above files **without** the *.sample* suffix (e.g., *ca.crt*, *charon.crt*, *charon.key*), the new certificates have been enabled. On the license server, the sample files (for root CA and license server) are in */opt/license_server/certs.* Please see the *VE License Server guide* in License Documentation for more information.
- **Make sure you understand the implications and possible side-effects before changing the certificate configuration**. Incorrect configurations can lead to the loss of license access and interruptions in operation.

Firewall Considerations

If the VE license server is not installed on the same system as the emulator, any intermediate firewall must allow at least the port on which the license is served. Optionally, the firewalls must allow the port on which the web-based GUI is available. These ports are **configurable** on the VE license server. The default values are the following:

- Default port on which licenses are served by the VE license server: TCP 8083.
- Default port on which the web-based GUI runs: TCP 8084.

Creating a C2V File on a VE License Server

Running esxi_bind before First C2V Creation on VMware

The **esxi_bind** command sets up the necessary communication connection between the VE license server and the ESXi host / the vCenter Server.

It must be run on the license server (and the backup license server, if applicable):

- **once** before the first license is requested, **and**
- **again** should the user credentials, the password, or the address data for the access to the ESXi host / the vCenter Server change. Please make sure that the password of the selected user account does not automatically expire after a certain time period. This would cause disruptions in the license server operation and make it impossible for clients to receive their license.

Perform the following steps:

1. Use **ssh** to log in on the license server instance (assuming that username/password login is possible for an on-premises VMware installation).
   `# ssh <user>@<license-server-ip>`
   where
   a. *<user>* is the user for interactive login associated with your license server system
   b. *<license-server-ip>* is the ip address of your license server system
2. Become the privileged user on the license server and run the **esxi_bind** program.
   a. Become the root user: `# sudo -i`
   b. Run the **esxi_bind** program:
      `# /opt/license-server/esxi_bind -a <address> -u <username> -p <password>`
      where
      i. *<address>* is the IP address of the ESXi host or vCenter Server
      ii. *<username>* is a user on the ESXi host or vCenter Server (**see notes below**).
      iii. *<password>* is the password of the user
3. If the command is successful, it will create the file */opt/license-server/config.ini* containing the connection data (the password is encrypted).

Important notes regarding the user on the ESXi host or the vCenter Server:

1. The **username on the vCenter Server** can take different forms:
   - Simple username
     esxi_bind parameter example: `-u myusername`
   - Username includes a domain name in one of the following two formats:
     - *<domain>\<username>*
       esxi_bind parameter example (quotes are mandatory): `-u 'mydomain\myusername'`
     - *<username>@<domain>*
       esxi_bind parameter example: `-u myusername@mydomain`
2. The user must have at least the following **global permissions** (i.e. the permissions cannot be limited to a specific VM):
   - Datastore > Allocate Space
   - VirtualMachine > Config > AddNewDisk
   - VirtualMachine > Config > RemoveDisk

Please note: if username and/or password contain Unix shell meta-characters, these characters must be escaped (enclose the string in single quotes, or add a backslash character in front of the meta-character).

Creating a VE C2V File and Sending it to Stromasys

The fingerprint is collected on the license server using the **c2v** utility.

Perform the following steps to collect the fingerprint on the license server and (if applicable) the backup license server:

1. Use **ssh** to log in on the license server instance.
   # ssh -i ~/.ssh/*<mykey> <user>*@*<license-server-ip>*
   where
   a. *<mykey>* is the private key of the key-pair you associated with your cloud instance
      (for an on-premises VMware installation where login with username/password is allowed, it is not needed)
   b. *<user>* is the user for interactive login associated with your license server instance (e.g., *opc* on OCI, *centos* for a CentOS instance on AWS, or the custom user on your VMware virtual machine or physical server; for an instance installed from a prepackaged Charon VE marketplace image, use *sshuser*)
   c. *<license-server-ip>* is the ip address of your license server system
2. Become the privileged user and run the **c2v** program.
   a. Become the root user: **# sudo -i**
   b. Run the c2v program: **# /opt/license-server/c2v --filename** *<my-file>*.**c2v --platform** *<my-platform>*
      where
      i. *<my-file>.c2v* is the path and name under which you want to store the fingerprint. The file type is C2V (customer-to-vendor)
      ii. *<my-platform>* indicates the platform on which the license server runs (possible values: **physical**, **aws**, **oci, gcp, azure**, **ibm**, **nutanix**, or **esxi**).
3. Copy the resulting C2V file to your local system (unless you can send email from the license server system).
4. Send the C2V file to the Stromasys orders department (email address will be provided by Stromasys).

Installing a VE V2C File on a VE License Server

In response to the C2V file, Stromasys will send you a V2C file. This file contains the license data and is installed on the license server using the **v2c** utility.

Perform the following steps to install the license on the license server:

1. Copy the V2C file to the license server (e.g., with SFTP).
2. Use **ssh** to log in on the license server instance.
   # ssh -i ~/.ssh/*<mykey>* *<user>*@*<license-server-ip>*
   where
   a. *<mykey>* is the private key of the key-pair you associated with your license server instance
      (for an on-premises VMware installation where login with username/password is allowed, it is not needed)
   b. *<user>* is the user for interactive login associated with your license server instance (e.g., *opc* on OCI, *centos* for a CentOS instance on AWS, or the custom user on your VMware virtual machine or your physical server; for an instance installed from a prepackaged Charon VE marketplace image, use *sshuser*)
   c. *<license-server-ip>* is the ip address of your license server system
3. Become the privileged user and run the **v2c** program.
   a. Become the root user: **# sudo -i**
   b. Run the v2c program: **# /opt/license-server/v2c -f** *<my-file>*.**v2c**
      where *<my-file>.v2c* is the path and name under which you want to store the fingerprint. The file type is V2C (vendor-to-customer).

After the installation of the V2C file, the license server will be restarted.

Viewing the License on a VE License Server

The license data can be viewed via the web-based GUI of the VE license server (see Licensing Documentation). It can also be viewed with the **`license_viewer`** program using the following steps:

1. Use **ssh** to log in on the license server instance.
   # ssh -i ~/.ssh/*<mykey>* *<user>*@*<license-server-ip>*
   where
   a. *<mykey>* is the private key of the key-pair you associated with your license server instance
      (for an on-premises VMware installation where login with username/password is allowed, it is not needed)
   b. *<user>* is the user for interactive login associated with your license server instance (e.g., *opc* on OCI, *centos* for a CentOS instance on AWS, or the custom user on your VMware virtual machine or your physical server; for an instance installed from a prepackaged Charon VE marketplace image, use *sshuser*)
   c. *<license-server-ip>* is the ip address of your license server system
2. Become the privileged user and run the **license_viewer** program.
   a. Become the root user: **# sudo -i**
   b. Run the license_viewer program: **# /opt/license-server/license_viewer**

Charon-PAR Emulator License Configuration

The license server must be added to the Charon-PAR emulator configuration. Please refer to the section Initial Emulator License Configuration.

Updating an Existing License

If you need to update an **existing license**, for example because the time limit on the license has expired or to upgrade to a new product versions, perform the following tasks:

1. Generate the C2V file for the existing license. This Customer-to-Vendor (C2V) file contains the license characteristics necessary for creating the license update.
2. Send the C2V file to Stromasys. Stromasys will use the data to create the necessary license update. You will receive a V2C file (the Vendor-to-Customer file).
3. Apply the license data from the V2C file(s) on the license server. This will install and activate the update for your license.

License Troubleshooting

The most important tool for identifying a license problem are the log files of the Charon emulator, the VE license server, and the Linux system. Always check them first in case of a problem.

Log Files for License Troubleshooting

Charon emulator log file location:

The default location of the emulator log files is the directory in which the emulator was started.

Please note: the path of the log emulator log files can be configured by the user to a non-default value.

VE license server log file location:

The path to the VE license server log on the license server is **/opt/license-server/log/license.log**.

The path to the VE integrated web server is **/opt/license-server/log/webserver.log**.

Linux system log:

The Linux logs can be viewed with the journalctl program. Examples:

- Identify VE license server entries (newest first): **journalctl -r -t license_server**
- Identify HASP runtime daemon entries (newest first): **journalctl -r -t aksusbd**

System Processes for Licensing

Correct license operation requires the corresponding system processes.

Sentinel HASP System processes

HASP licenses require the **aksusbd** process and the **hasplmd** process. In aksusbd version 7.63, both processes are started and stopped by the same service (/etc/init.d/aksusbd). In aksusbd version 8.13, there are two systemd services:

- the **aksusbd** service
- the **hasplmd** service

Checking the status of the services:

```
# systemctl status aksusbd
# systemctl status hasplmd
```

Starting and stopping the services:

- The two services have a dependency such that starting one of them will start the other, stopping one of them will stop the other, and restarting one of them will restart the other.
- Starting the services:
  ```
  # systemctl start aksusbd
  # systemctl start hasplmd
  ```
- Stopping the services:
  ```
  # systemctl stop aksusbd
  # systemctl stop hasplmd
  ```

VE License Server Process

VE licensing requires that the **licensed** service must run on the license server.

Checking the status of the service:

```
# systemctl status licensed
```

Starting and stopping the service:

- Starting the service:
  # systemctl start licensed
- Stopping the service:
  # systemctl stop licensed

Further Information

Sentinel HASP licenses:

Should there be a failure when trying to display or load the license key, review the error codes listed on the Stromasys website and the associated solutions.

VE Licenses:

Refer to the VE license server guide.

If the problem cannot be solved, please contact your VAR or the Stromasys Customer Support Center (maintenance contract required) using the details in the section Obtaining Technical Assistance.

Charon-PAR Utilities

Charon-PAR provides the following set of utilities:

- MTD Utility
- hasp_srm_view
- hasp_update

MTD Utility

Charon-PAR provides the utility MTD , a Magnetic Tape Dump utility which can copy Charon-PAR virtual tape image files to or from a physical tape device connected to the host system. MTD is a Linux command line utility.

The MTD utility is used to:

- Create a Charon-PAR tape image from a physical tape
- Write a tape image to a physical tape.

Command syntax:

```
$ mtd [options] <tape device name> <tape container name>   (tape to file)

$ mtd [options] <tape container name> <tape device name>   (file to tape)
```

Parameters:

```
MTD - CHARON Magnetic Tape Dump & Restore utility, Version 2.3 (Build 18302)
Copyright (C) 2009-2018 STROMASYS SA. All rights reserved.

Usage: mtd [options] <tape-drive-name> <file-name> - dump tape content to file
mtd                  <file-name> <tape-drive-name> - restore dump to tape
mtd                  <file-name> <file-name>       - convert formats
mtd                  <file-name>                   - examine tape dump and check integrity

<tape-drive-name> - tape drive
      <file-name> - name of tape container file (.mtd or .vtape)

Options:
-l <file-name> - log file name (.log)
-n             - do not rewind tape
-r <number>    - number of attempts to retry failing tape reads
-i             - ignore failing tape reads (implies -r 0)
-p             - disable progress reporting
-v             - enable verbose trace of data transfer (implies -p)
-s             - output tape image in SMA format (not relevant for Charon-PAR)
-g             - gather statistics and print upon completion
-a             - do not print logo
```

Example physical tape to container:

```
$ mtd -l tape1.txt -r 10 /dev/st5 /charon/tapes/tape1.vtape
```

Example container to physical tape:

```
$ mtd /charon/tapes/tape1.vtape /dev/st5
```

hasp_srm_view

General notes:

The **`hasp_srm_view`** utility provides a simple command-line utility for gathering Sentinel license information.

Please note: Local hardware licenses can only be displayed from a local connection to the system, for example via the console. If you are connected via a remote connection, for example **ssh**, the **hasp_srm_view** command will return an error. Network licenses do not have this problem.

Workaround when connected via remote connection:

When connected to the system via a remote connection you can display the license contents with the following command (adapt the path of the command if your installation location is different):

```
$ ssh localhost /opt/charon/bin/hasp_srm_view [OPTION]
```

Name:

hasp_srm_view – Charon Sentinel HASP Utility

Syntax:

```
hasp_srm_view [OPTION]
```

Parameters:

The command options of the **`hasp_srm_view`** utility are listed below. If no options are specified, **`-l`** is assumed.

Parameter	Description
`-?, -h, -help`	Display the usage message.
`-all`	Show the product license details for all available keys.
`-c2v` *`FILENAME`*	Collect the Sentinel HASP key status information for the default key and write it to *FILENAME*. This C2V file is used to request an update for an existing license. Can be used together with the **`-key`** parameter to select a particular key for information collection.
`-fgp` *`FILENAME`*	Collect the host fingerprint information for generating a Sentinel software license and write it to *FILENAME*.
`-key` *`KEYNUMBER`*	Assumes a show command and displays only the product license details for the key with the ID specified in *KEYNUMBER* . This parameter can be used together with the **`-c2v`** option to select a specific license for which a C2V file should be created. Required if more than one license is visible to the system.
`-l`	Show the product license details for the default key attached to the host.

hasp_update

Name:

hasp_update – Sentinel HASP Update and Transfer Utility

Syntax:

```
hasp_update u filename
```

Parameters:

The **`hasp_update`** utility provides a simple command-line interface for manipulating the HASP License Key.

Parameter	Description
`u` *`filename`*	Apply the HASP key update found in *filename*.
The following parameters are obsolete:	
`i` *`filename`*	Retrieve Sentinel protection key information and write it to *filename*.
`d`	Detach a license from a Sentinel Software License (SL).
`r`	Rehost a license from a Sentinel Software License (SL) key.

Please note: Only the **u** option of this command should be used. For all other functions, please use the Sentinel Admin Control Center or the **`hasp_srm_view`** utility.

Emulated Model Hardware Configuration Details

Table of contents

Charon-PAR Model Names

Charon-PAR/PA9-64 model name details

The model names of Charon-PAR emulated historic PA-RISC systems for HP-UX consist of three parts:

```
<family-name>-<number-of-cpus>-<clock-speed>
```

Parameters:

- ***family-name*** describes the model family, e.g., rp2400,
- ***number-of-cpus*** describes the number of emulated CPUs, and
- ***clock-speed*** describes the CPU clock frequency in MHz.

Example: *rp2470-2-750* stands for an rp2470 model with two 750-MHz CPUs.

Abbreviations are possible, e.g,, **rp2470-1** stands for an **rp2470-1-750** model. See configuration file templates for permitted abbreviations.

Charon-PAR/PA9-32 model name details

Currently, there are the following 32-bit models:

- The PA9-32/720 (PA-RISC v1.1a) with single CPU at 50MHz, model string: "720"
- The PA9-32/B132L with 1 CPU at 132MHz (oversized models with up to 16 CPUs may be available), model string: B132L-*<nr-of-cpus>.* Supported starting with Charon-PAR 3.0.7.

Charon-PAR/PA3 model name details

The model names of Charon-PAR emulated historic PA-RISC systems for MPE/iX consist of three parts:

```
<family-name>-<number-of-cpus>-<clock-speed>
```

Parameters:

- ***family-name*** describes the model family, e.g., A400,
- ***number-of-cpus*** describes the number of emulated CPUs multiplied by 100, and
- ***clock-speed*** describes the CPU clock frequency in MHz.

Example: *A400-100-110* is a single-CPU A400 system running at 110 MHz.

Device Addressing Terminology

In the following sections, an important part of the hardware configuration of an emulated model are the so called **device paths**.

A **device path** has the following structure: ***`SBA/LBA/device-number/function[.target.LUN]`***. These components have the following meaning:

- ***SBA***: system bus adapter
- ***LBA***: local bus adapter
- ***Device number***: identification of an individual device (e.g., an Ethernet card).
- ***Function***: identification of a particular unit of functionality (e.g., one Ethernet port or one SCSI bus) on a specific device. There can be up to four functions per device.
- ***Target***: optional further identification. Its use depends on the type of device connected. For example, a SCSI target ID.
- ***LUN***: logical unit number of SCSI devices (e.g., disks and tapes). Used to address a logical unit associated with a SCSI target ID.

Examples:

- SBA 0, LBA 0, device number 1, function 0 results in a device path of 0/0/1/0
- SBA 0, LBA 2, device number 1, function 1 results in a device path of 0/2/1/1
- SBA 0, LBA 0, device number 1, function 0, SCSI target ID 1, LUN 0 results in a device path of 0/0/1/0.1.0

Charon-PAR/PA9-64 - Supported 64-bit Model Configurations

This chapter describes the hardware configuration of the supported emulated historic 64-bit PA-RISC models for HP-UX. This information will be referred to in other section, e.g., the configuration file reference.

A400 Model Family

A400 Family Overview

Model name	No. of CPUs	Max. RAM in GB	CPU Freq. MHz	HP-UX Model String	Configuration model name (examples for 1 CPU)
rp2400	1	2	360 440	9000/800/A400-36 9000/800/A400-44	rp2400-1-360 rp2400-1-440
rp2430	1	2	550 650	9000/800/A400-5X 9000/800/A400-6X	rp2430-1-550 rp2430-1-650
rp2405	1	2	650	9000/800/A400-6X	rp2405-1-650-A7121A

A400 Family Hardware Configuration

Models of the A400 family have 3 PCI buses.

Built-in controllers (all on PCI bus 0):

Controller	Device path(s)	Controller character code (for SCSI device names)
DEC 21143-PD Tulip 10/100 Mbit/s Ethernet controller	0/0/0/0	

Dual SCSI-2 controller (53C875)	0/0/1/0 and 0/0/1/1	A = 0/0/1/0
		B = 0/0/1/1

Dual SCSI-3 LVD controller (53C896)	0/0/2/0 and 0/0/2/1	C = 0/0/2/0
		D = 0/0/2/1

Diva serial PCI card with 2 available serial ports	0/0/4/0.0 and 0/0/4/0.1	

There are two additional PCI slots:

Slot number	Device path (SBA/LBA/device)
1	0/2/0
2	0/4/0

A500 Model Family

A500 Family Overview

Model name	No. of CPUs	Max. RAM in GB	CPU Freq. MHz	HP-UX Model String	Configuration model name (examples for 1 CPU)
rp2450	1-2	8	360 440 550	9000/800/A500-36 9000/800/A500-44 9000/800/A500-5X	rp2450-1-360 rp2450-1-440 rp2450-1-550
rp2470	1-2	8	650 750	9000/800/A500-6X 9000/800/A500-7X	rp2470-1-650 rp2470-1-750
rp2405	1-2	8	650	9000/800/A500-6X	rp2405-1-650

A500 Family Hardware Configuration

Models of the A500 family have 4 PCI buses.

Built-in controllers (all on PCI bus 0):

Controller	Device path(s)	Controller character code (for SCSI device names)
DEC 21143-PD Tulip 10/100 Mbit/s Ethernet controller	0/0/0/0	

Dual SCSI-2 controller (53C875)	0/0/1/0 and 0/0/1/1	A = 0/0/1/0
		B = 0/0/1/1

Dual SCSI-3 LVD controller (53C896)	0/0/2/0 and 0/0/2/1	C = 0/0/2/0
		D = 0/0/2/1

Diva serial PCI card with 2 available serial ports	0/0/4/0.0 and 0/0/4/0.1	

There are four additional PCI slots:

Slot number	Device path (SBA/LBA/device)
1	0/2/0
2	0/4/0
3	0/6/0
4	0/6/2

L-Class and rp54xx Model Families

L-Class and rp54xx Families Overview

Model name	No. of CPUs	Max. RAM in GB	CPU Freq. MHz	HP-UX Model String	Configuration model name (examples for 1 CPU)
rp5400	1-2	8	360 440 550	9000/800/L1000-36 9000/800/L1000-44 9000/800/L1000-5X	rp5400-1-360 rp5400-1-440 rp5400-1-550
rp5450	1-4	16	360 440 550	9000/800/L2000-36 9000/800/L2000-44 9000/800/L2000-5X	rp5450-1-360 rp5450-1-440 rp5450-1-550
rp5430	1-2	8	360 440 550 650 750 875	9000/800/L1500-36 9000/800/L1500-44 9000/800/L1500-5X 9000/800/L1500-6X 9000/800/L1500-7X 9000/800/L1500-8X	rp5430-1-360 rp5430-1-440 rp5430-1-550 rp5430-1-650 rp5430-1-750 rp5430-1-875
rp5470	1-4	16	550 650 750 875	9000/800/L3000-5X 9000/800/L3000-6X 9000/800/L3000-7X 9000/800/L3000-8X	rp5470-1-550 rp5470-1-650 rp5470-1-750 rp5470-1-875

L1000 (rp5400) Hardware Configuration

L1000 systems have 8 PCI buses.

Built-in controllers (all on PCI bus 0):

Controller	Device path(s)	Controller character code (for SCSI device names)
DEC 21143-PD Tulip 10/100 Mbit/s Ethernet controller	0/0/0/0	

Dual SCSI-2 controller (53C875)	0/0/1/0 and 0/0/1/1	A = 0/0/1/0
		B = 0/0/1/1

Dual SCSI-3 LVD controller (53C896)	0/0/2/0 and 0/0/2/1	C = 0/0/2/0
		D = 0/0/2/1

Diva serial PCI card with 2 available serial ports	0/0/4/0.0 and 0/0/4/0.1	

Six PCI slots are available for expansion:

Slot number	Device path (SBA/LBA/device)
3	0/1/3
8	0/2/0
9	0/6/0
10	0/3/0
11	0/7/0
12	0/4/0

L2000 (rp5450) Hardware Configuration

L2000 systems have 8 PCI buses.

Built-in controllers (all on PCI bus 0):

Controller	Device path(s)	Controller character code (for SCSI device names)
DEC 21143-PD Tulip 10/100 Mbit/s Ethernet controller	0/0/0/0	

Dual SCSI-2 controller (53C875)	0/0/1/0 and 0/0/1/1	A = 0/0/1/0
		B = 0/0/1/1

Dual SCSI-3 LVD controller (53C896)	0/0/2/0 and 0/0/2/1	C = 0/0/2/0
		D = 0/0/2/1

Diva serial PCI card with 2 available serial ports	0/0/4/0.0 and 0/0/4/0.1	

Ten PCI slots are available for expansion:

Slot number	Device path (SBA/LBA/device)
3	0/1/3
4	0/1/2
5	0/1/1
6	0/1/0
7	0/5/0
8	0/2/0
9	0/6/0
10	0/3/0
11	0/7/0
12	0/4/0

L1500 (rp5430) Hardware Configuration

L1500 systems have 7 PCI buses.

Built-in controllers (all on PCI bus 0):

Controller	Device path(s)	Controller character code (for SCSI device names)
DEC 21143-PD Tulip 10/100 Mbit/s Ethernet controller	0/0/0/0	

Dual SCSI-2 controller (53C875)	0/0/1/0 and 0/0/1/1	A = 0/0/1/0
		B = 0/0/1/1

Dual SCSI-3 LVD controller (53C896)	0/0/2/0 and 0/0/2/1	C = 0/0/2/0
		D = 0/0/2/1

Diva serial PCI card with 2 available serial ports	0/0/4/0.0 and 0/0/4/0.1	

Six PCI slots are available for expansion:

Slot number	Device path (SBA/LBA/device)
3	0/4/2
8	0/3/0
9	0/9/0
10	0/8/0
11	0/12/0
12	0/10/0

L3000 (rp5470) Hardware Configuration

L3000 systems have 10 PCI buses.

Built-in controllers (all on PCI bus 0):

Controller	Device path(s)	Controller character code (for SCSI device names)
DEC 21143-PD Tulip 10/100 Mbit/s Ethernet controller	0/0/0/0	

Dual SCSI-2 controller (53C875)	0/0/1/0 and 0/0/1/1	A = 0/0/1/0
		B = 0/0/1/1

Dual SCSI-3 LVD controller (53C896)	0/0/2/0 and 0/0/2/1	C = 0/0/2/0
		D = 0/0/2/1

Diva serial PCI card with 2 available serial ports	0/0/4/0.0 and 0/0/4/0.1	

Ten PCI slots are available for expansion:

Slot number	Device path (SBA/LBA/device)
3	0/4/2
4	0/4/0
5	0/2/0
6	0/5/0
7	0/1/0
8	0/3/0
9	0/9/0
10	0/8/0
11	0/12/0
12	0/10/0

N4000 (rp7400) Model Family

N4000 (rp7400) Family Overview

Model name	No. of CPUs	Max. RAM in GB	CPU Freq. MHz	HP-UX Model String	Configuration model name (examples for 1 CPU)
rp7400	1-8 *	32 *	360 440 550 650 750	9000/800/N4000-36 9000/800/N4000-44 9000/800/N4000-55 9000/800/N4000-65 9000/800/N4000-75	rp7400-1-360 rp7400-1-440 rp7400-1-550 rp7400-1-650 rp7400-1-750

* Starting with version 3.0.5, the models above may also be available in “oversized” versions up to 64 CPUs and 512GB RAM. Please check the availability for your model with your Sales contact.

Starting with version 3.0.5, models with a CPU frequency of 750 MHz (model name in configuration: **rp7400X-*NN*-750**) will be able to make use of the extended TLB feature.

N4000 (rp7400) Hardware Configuration

N4000 systems have 14 PCI buses.

Built-in controllers (all on PCI bus 0 of system bus adapter 0):

Controller	Device path(s)	Controller character code (for SCSI device names)
DEC 21143-PD Tulip 10/100 Mbit/s Ethernet controller	0/0/0/0	
SCSI-2 controller (53C875)	0/0/1/0	A = 0/0/1/0

Dual SCSI-3 LVD controller (53C896)	0/0/2/0 and 0/0/2/1	B = 0/0/2/0
		C = 0/0/2/1

Diva serial PCI card with 2 available serial ports	0/0/4/0.0 and 0/0/4/0.1	

Twelve PCI slots are available for expansion:

Slot number	Device path (SBA/LBA/device)
1	0/5/0
2	0/4/0
3	0/12/0
4	0/8/0
5	0/10/0
6	0/2/0
7	1/12/0
8	1/10/0
9	1/4/0
10	1/2/0
11	1/8/0
12	1/0/0

rp34xx Model Family (Supported Starting with Charon-PAR Version 3.0.5)

rp34xx Family Overview

Model name	No. of CPUs *	Max. RAM in GB *	CPU Freq. MHz*	HP-UX Model String	Configuration model name (examples for 1 CPU)
rp3410	1 dual-core	6GB	800 1000 "+"-Versions: 800 1000	9000/800/rp3410 9000/800/rp3410 9000/800/rp3410 9000/800/rp3410	rp3410-1-800 rp3410-1-1000 rp3410+-1-800 rp3410+-1-1000
rp3440	1-2 dual-core	32GB	800 1000 "+"-Versions: 800 1000	9000/800/rp3440 9000/800/rp3440 9000/800/rp3440 9000/800/rp3440	rp3440-1-800 rp3440-1-1000 rp3440+-1-800 rp3440+-1-1000

* The **rp3440** models may also be available in “oversized” versions up to 128 CPUs and 512GB RAM. Please check the availability for your model with your Sales representative. The +-versions emulate PA-8800 CPUs, the non-plus versions PA-8900 CPUs.

rp34xx Family Hardware Configuration

The models of this family have 8 PCI buses. However, for Charon-PAR, 7 buses were implemented. All built-in controllers are on PCI bus 0.

Built-in controllers (all on PCI bus 0 of system bus adapter 0):

Controller	Device path(s)	Controller character code (for SCSI device names)
DEC 21143-PD Tulip 10/100 Mbit/s Ethernet controller	0/0/0/0	

SCSI-3 LVD controller (531010)	0/0/1/0 and 0/0/1/1	A = 0/0/1/0
		B = 0/0/1/1

Dual SCSI-3 LVD controller (531010)	0/0/2/0 and 0/0/2/1	C = 0/0/2/0
		D = 0/0/2/1

Diva serial PCI card with 2 available serial ports	0/0/4/0.0 and 0/0/4/0.1	
Diva serial PCI card with 2 available serial ports	0/0/5/0.0 and 0/0/5/0.1	

Six PCI slots are available for expansion:

Slot number	Device path (SBA/LBA/device)
1	0/1/0
2	0/2/0
3	0/3/0
4	0/4/0
5	0/5/0
6	0/6/0

rp44xx Model Family (Supported Starting with Charon-PAR Version 3.0.5)

rp44xx Family Overview

Model name	No. of CPUs *	Max. RAM in GB *	CPU Freq. MHz*	HP-UX Model String	Configuration model name (examples for 1 CPU)
rp4410	1-2 dual-core	128GB	800 1000 "+"-Versions: 800 1000	9000/800/rp4410 9000/800/rp4410 9000/800/rp4410 9000/800/rp4410	rp4410-1-800 rp4410-1-1000 rp4410+-1-800 rp4410+-1-1000
rp4440	1-4 dual-core	128GB	800 1000 "+"-Versions: 800 1000	9000/800/rp4440 9000/800/rp4440 9000/800/rp4440 9000/800/rp4440	rp4440-1-800 rp4440-1-1000 rp4440+-1-800 rp4440+-1-1000

* The **rp4440** models may also be available in “oversized” versions up to 128 CPUs and 512GB RAM. Please check the availability for your model with your Sales representative. The +-versions emulate PA-8800 CPUs, the non-plus versions PA-8900 CPUs.

rp44xx Family Hardware Configuration

The models of this family have 8 PCI buses. However, for Charon-PAR, 7 buses were implemented. All built-in controllers are on PCI bus 0.

Built-in controllers (all on PCI bus 0 of system bus adapter 0):

Controller	Device path(s)	Controller character code (for SCSI device names)
DEC 21143-PD Tulip 10/100 Mbit/s Ethernet controller	0/0/0/0	

SCSI-3 LVD controller (531010)	0/0/1/0 and 0/0/1/1	A = 0/0/1/0
		B = 0/0/1/1

Dual SCSI-3 LVD controller (531010)	0/0/2/0 and 0/0/2/1	C = 0/0/2/0
		D = 0/0/2/1

Diva serial PCI card with 2 available serial ports	0/0/4/0.0 and 0/0/4/0.1	
Diva serial PCI card with 2 available serial ports	0/0/5/0.0 and 0/0/5/0.1	

Six PCI slots are available for expansion:

Slot number	Device path (SBA/LBA/device)
1	0/1/0
2	0/2/0
3	0/3/0
4	0/4/0
5	0/5/0
6	0/6/0

Charon-PAR/PA9-32 - Supported 32-bit Model Configurations

This chapter describes the hardware configuration of the supported emulated historic 32-bit PA-RISC models for HP-UX. This information will be referred to in other section, e.g., the configuration file reference.

PA9-32/720 (9000/720)

PA9-32/720 Overview

Model name	No. of CPUs	Max. RAM in MB	CPU Freq. MHz	Configuration model name
HP 9000/720	1 *	265	50	720

* Oversized versions with up to 4 CPUs (starting with Charon-PAR 3.0.13) and up to 3840MB of RAM (starting with version 3.0.14) may be available. **HP-UX limitations**: 9.x supports 1 CPU and up to 768 MB of RAM; 10.20 and 11.00 support 4 CPUs and up to 3840 MB of RAM. Please check availability with your Sales contact.

PA9-32/720 Hardware Configuration

Built-in controllers:

Controller	Device path(s)	Controller character code (for SCSI device names)
Coral SGC Graphics [(1)]	1	
Cobra Core BA	2	
Cobra Core SCSI (controller model 53C700)	2/0/1	A = 2/0/1
Cobra Core LAN (802.3)	2/0/2	
Cobra Core HIL [(1)] (connector for input devices)	2/0/3	
Cobra Core RS-232	2/0/4	
Cobra Core RS-232	2/0/5	
Cobra Core Centronics [(1)]	2/0/6	
Cobra (720)	8	
Memory	9	

[(1)] Not yet implemented.

This model has no expansion slots.

PA9-32/B132L (9000/778) - Supported from Charon-PAR 3.0.7

PA9-32/B132L Overview

Model name	No. of CPUs	Max. RAM in GB	CPU Freq. MHz	Configuration model name
HP 9000/778	1 (1)	2 (2)	132	B132L-<*nr-of-cpu*>

(1) Supported starting with Charon-PAR version 3.0.7. Oversized versions with up to 16 CPUs and up to 3840MB RAM (starting with version 3.0.14) may be available. **HP-UX limitations**: 10.20 supports 16 CPUs; 11.00 and 11.11 support 15 CPUs. 10.20, 11.00 and 11.11 support up to 3840MB RAM. Please check availability with your Sales contact.

(2) 2GB are supported starting with Charon-PAR version 3.0.8-22201. Earlier versions support 1.5GB.

PA9-32/B132L Hardware Configuration

Built-in controllers:

Controller	Device path(s)	Controller character code (for SCSI device names)
Merlin 132 Core BA	8/16	
Merlin 132 Core RS-232	8/16/4	
Merlin 132 Core SCSI (LSI 53C710)	8/16/5	A = 8/16/5
Merlin 132 Core LAN (802.3; 82596)	8/16/6	
Merlin 132 Core Centronis (1)	8/16/0	
Merlin 132 Core Audio (1)	8/16/1	
Merlin 132 Core PS/2 Port (1)	8/16/7	
Merlin 132 Core PS/2 Port (1)	8/16/8	
Merlin 132 Wax BA	8/20	
Merlin 132 Wax EISA BA (1)	8/20/5	
Merlin 132 Wax HIL (1) (connector for input devices)	8/20/1	
Merlin 132 Wax RS-232	8/20/2	
Gecko GSC Core Graphics (1)	8/24	

(1) Not yet implemented.

This model has no expansion slots.

Charon-PAR/PA3 - Supported Model Configurations

This chapter describes the hardware configuration of the supported emulated historic PA-RISC models for MPE/iX. This information will be referred to in other section, e.g., the configuration file reference.

A400 Model Family

A400 Family Overview

Model name	No. of CPUs	Max. RAM in GB	CPU Freq. MHz	Configuration model name (examples for 1 CPU)
A400	1	2	110 150	A400-100-110 A400-100-150

A400 Family Hardware Configuration

Models of the A400 family have 3 PCI buses.

Built-in controllers (all on PCI bus 0):

Controller	Device path(s)	Controller character code (for SCSI device names)
DEC 21143 Tulip 10/100 Mbit/s Ethernet controller	0/0/0/0	

Dual SCSI-2 Controller (53C875)	0/0/1/0 and 0/0/1/1	A = 0/0/1/0
		B = 0/0/1/1

Dual SCSI-3 LVD controller (53C896)	0/0/2/0 and 0/0/2/1	C = 0/0/2/0
		D = 0/0/2/1

Diva serial PCI card with 2 available serial ports	0/0/4/0.0 and 0/0/4/0.1	

There are two additional PCI slots:

Slot number	Device path (SBA/LBA/device)
1	0/2/0
2	0/4/0

A500 Model Family

A500 Family Overview

Model name	No. of CPUs	Max. RAM in GB	CPU Freq. MHz	Configuration model name (examples for 1 CPU)
A500	1-2	8	140 200	A500-100-140 A500-100-200

A500 Family Hardware Configuration

Models of the A500 family have 4 PCI buses.

Built-in controllers (all on PCI bus 0):

Controller	Device path(s)	Controller character code (for SCSI device names)
DEC 21143 Tulip 10/100 Mbit/s Ethernet controller	0/0/0/0	

Dual SCSI-2 Controller (53C875)	0/0/1/0 and 0/0/1/1	A = 0/0/1/0
		B = 0/0/1/1

Dual SCSI-3 LVD controller (53C896)	0/0/2/0 and 0/0/2/1	C = 0/0/2/0
		D = 0/0/2/1

Diva serial PCI card with 2 available serial ports	0/0/4/0.0 and 0/0/4/0.1	

There are four additional PCI slots:

Slot number	Device path (SBA/LBA/device)
1	0/2/0
2	0/4/0
3	0/6/0
4	0/6/2

N4000 Model Family

N4000 Family Overview

Model name	No. of CPUs	Max. RAM in GB	CPU Freq. MHz	Configuration model name (examples for 1 CPU)
N4000	1-4, 6, or 8	16	220 330 380 440 500 550 750	N4000-100-220 N4000-100-330 N4000-100-380 N4000-100-440 N4000-100-500 N4000-100-550 N4000-100-750

Starting with version 3.0.5, models with a CPU frequency of 750 MHz (model name in configuration: N4000X-*NN*-750) will be able to make use of the extended TLB feature.

N4000 Hardware Configuration

N4000 systems have 14 PCI buses.

Built-in controllers (all on PCI bus 0 of system bus adapter 0):

Controller	Device path(s)	Controller character code (for SCSI device names)
DEC 21143 Tulip 10/100 Mbit/s Ethernet controller	0/0/0/0	
SCSI-2 Controller (53C875)	0/0/1/0 and 0/0/1/1	A = 0/0/1/0

Dual SCSI-3 LVD controller (53C896)	0/0/2/0 and 0/0/2/1	B = 0/0/2/0
		C = 0/0/2/1

Diva serial PCI card with 2 available serial ports	0/0/4/0.0 and 0/0/4/0.1	

Twelve PCI slots are available for expansion:

Slot number	Device path (SBA/LBA/device)
1	0/5/0
2	0/4/0
3	0/12/0
4	0/8/0
5	0/10/0
6	0/2/0
7	1/12/0
8	1/10/0
9	1/4/0
10	1/2/0
11	1/8/0
12	1/0/0

Configuration File Reference

Contents

Overview

The hardware of the emulated system is defined in the emulator configuration file. Charon-PAR comes with a number of template files that are stored in **/opt/charon/cfg**. Each template file contains the settings for a particular Charon-PAR/PA3 or Charon-PAR/PA9 hardware model or model family. The files also contain important descriptions for all parameters listed.

The name of the Charon-PAR/PA3 template file is **pa3.cfg**. The names of the Charon-PAR/PA9-64 templates are **rp*.cfg,** for Charon-PAR/PA9-32 **pa9-32.cfg**.

Before starting an emulator instance,

- copy the appropriate template file to the directory where you store the configuration data of your emulator instance, and
- modify the template file according to your requirements.

Comments in the file are preceded by the **#** character.

Please note that some entries require an **equal-sign** (=) and for other parameters an equal-sign is not allowed. The emulator will not start if a required equal-sign is missing or an equal-sign is placed where it is not permitted.

If a parameter value is a string, it must be enclosed in double quotes.

The configuration file is structured in several sections that are described below.

Please note the following if running the emulator in a cloud enviroment: some emulated hardware can be linked to physical host hardware. For example, serial lines, tape drives, optical drives. **These options are not available in a cloud environment as the physical devices do not exist there**.

License Configuration

This section defines the license specific parameters.

Parameter license_key_id

Optional parameter for HASP licenses, mandatory for VE licenses.

Please note: when configuring the emulator, VE licenses and HASP licenses are **mutually exclusive**. If a configuration file contains the definition of a VE license, any HASP license definition is ignored (that is, the VE configuration takes precedence).

Parameter Use for HASP Licenses

There can be several **license_key_id** parameters in this section. They define a prioritized list of license keys. The first entry has the highest priority. If no key with this ID is available, the next key in the list will be tried. If a higher priority key becomes available, the emulator will switch to this license at the next periodic license check.

Syntax:

```
license_key_id "key-id-prio1"

license_key_id "key-id-prio2"

(...)
```

Parameters:

The *key-id-prioX* parameter is a numerical value that can be determined using the **hasp_srm_view -all** command and identifying the value of the **License KeyId** parameter in the output. As long as the ID is a numerical value, the quotes are optional.

Parameter Use for VE Licenses

The configuration of primary and (optionally) backup license server must be entered into the emulator configuration file using a text editor.

Relevant parameter:

```
license_key_id "VE://<license-server-IP>[:<port>]/<passphrase>/"
```

Where the following parameters are used:

- *license-server-IP*: the IP address of the VE license server (127.0.0.1 if the VE license server is on the same host).
- *port:* the TCP port on which the license is served (if not specified, the default port 8083 will be used).
- *passphrase*: the passphrase of the correct product section on the license. This is required for the emulator to identify the correct section.

To configure a **backup license server,** repeat the line, but with the address and passphrase of the backup license server. The first line is treated as the primary server, the second as the backup server.

Parameter license_use_any_key

Please note: this parameter is not supported by the VE license server. If it is in the configuration file, comment it out.

Optional parameter for HASP licensing. This parameter is closely related to the **license_key_id** parameter. If the license_key_id parameter is used, it determines what happens if none of the defined license keys are available.

Syntax:

```
license_use_any_key false | true
```

Parameters:

Default value: **false**

If this parameter is set to **true**, the emulator will try to find any other suitable licenses should none of the licenses specified by the **license_key_id** parameter(s) be available. By default, if the **license_key_id** parameter is used, only the license keys specified there will be considered.

Parameter license_id

Please note: this parameter is not supported by the VE license server. If it is in the configuration file, comment it out.

Optional parameter for HASP licensing. There can be several **license_id** parameters in this section. They define a prioritized list of product license IDs. The first entry has the highest priority. If no key with this product license ID is available, the next product license ID in the list will be attempted. If a higher priority license becomes available, the emulator will switch to this license at the next periodic license check.

Syntax:

```
license_id "lic-id-prio1"

license_id "lic-id-prio2"

(...)
```

Parameters:

The *lic-id-prioX* parameter is a string value that must be enclosed in quotes. It can be determined using the **hasp_srm_view -all** command and identifying the value of the **Product License Number** parameter in the output. If none of the specified product licenses is available, the emulator will not start, or a running emulator will stop.

Problem Handler Parameters

The Charon-PAR configuration file supports the following parameters that can be used to alert the user to potential license problems:

Parameter	Description
hasp_lost_usr_cmd (this parameter will be renamed to **license_lost_usr_cmd** to reflect the VE licensing option)	Runs a user-defined script or a executable if Charon disconnects from current valid license and no other valid license is found. This action is called every hour during the 12 hours grace period (at the end of which Charon-PAR stops unless a license has been found). The script or command started receives three parameters from Charon-PAR: *key-id*, *product-license-id*, and *t ermination time*. Usage example: `hasp_lost_usr_cmd=“./my-license_connection_lost.sh"`
license_changed_usr_cmd	Runs a user-defined script or an executable when Charon-PAR disconnects from one license and connects to another for any reason. The script or command started receives four parameters from Charon-PAR: *old-key-id*, *old-product-license-id*, *new-key-id*, and *new-product-license-id.* Usage example: `license_changed_usr_cmd="./my-license_changed.sh"`
license_expiration_warning_usr_cmd	Runs a user-defined script or an executable when the license expiration time is within few hours. The default setup is: starting 24 hours before expiration this script is invoked each hour. The script or command started receives three parameters from Charon-PAR: *key-id*, *product-license-id*, *expir ation time*. Usage example: `license_expiration_warning_usr_cmd=“./my-license_expiration_warning.sh”`

Machine State Parameters

Optional parameter: allows a non-default path to store the machine state.

Syntax:

```
machine_state_path "path"
```

Parameters:

The value of *path* is a Unix file system path. It must be enclosed in double quotes.

System Model Configuration

The system model configuration section contains one parameter defining the emulated historic PA-RISC model.

Syntax:

```
model "model-name"
```

The configured model must be covered by your license.

Charon-PAR Model Names

Model Name Rules

Charon-PAR/PA9-64 model name details

The model names of Charon-PAR emulated historic PA-RISC systems for HP-UX consist of three parts:

```
<family-name>-<number-of-cpus>-<clock-speed>
```

Parameters:

- ***family-name*** describes the model family, e.g., rp2400,
- ***number-of-cpus*** describes the number of emulated CPUs, and
- ***clock-speed*** describes the CPU clock frequency in MHz.

Example: *rp2470-2-750* stands for an rp2470 model with two 750-MHz CPUs.

Abbreviations are possible, e.g,, **rp2470-1** stands for an **rp2470-1-750** model. See configuration file templates for permitted abbreviations.

Charon-PAR/PA9-32 model name details

Currently, there are the following 32-bit models:

- The PA9-32/720 (PA-RISC v1.1a) with single CPU at 50MHz, model string: "720"
- The PA9-32/B132L with 1 CPU at 132MHz (oversized models with up to 16 CPUs may be available), model string: B132L-*<nr-of-cpus>.* Supported starting with Charon-PAR 3.0.7.

Charon-PAR/PA3 model name details

The model names of Charon-PAR emulated historic PA-RISC systems for MPE/iX consist of three parts:

```
<family-name>-<number-of-cpus>-<clock-speed>
```

Parameters:

- ***family-name*** describes the model family, e.g., A400,
- ***number-of-cpus*** describes the number of emulated CPUs multiplied by 100, and
- ***clock-speed*** describes the CPU clock frequency in MHz.

Example: *A400-100-110* is a single-CPU A400 system running at 110 MHz.

Model Name Examples

Each configuration file template is preconfigured for a certain family of models. The following table shows a sample *model-name* for each of the configuration file templates:

Template	Sample model names *model-name* (example for one CPU)	Description
pa3.cfg	A400-100-110, A500-100-200, N4000-100-750	
rp2400.cfg	rp2400-1-360	1 CPU, 360MHz, model string: 9000/800/A400-36
	rp2400-1-440	1 CPU, 440MHz, model string: 9000/800/A400-44
rp2430.cfg	rp2430-1-550	1 CPU, 550MHz, model string: 9000/800/A400-5X
	rp2430-1-650	1 CPU, 650MHz, model string: 9000/800/A400-6X
rp2450.cfg	rp2450-1-360	1 CPU, 360MHz, model string: 9000/800/A500-36
	rp2450-1-440	1 CPU, 440MHz, model string: 9000/800/A500-44
	rp2450-1-550	1 CPU, 550MHz, model string: 9000/800/A500-5X
rp2470.cfg	rp2470-1-650	1 CPU, 650MHz, model string: 9000/800/A500-6X
	rp2470-1-750	1 CPU, 750MHz, model string: 9000/800/A500-7X
rp3410[+].cfg	rp3410[+]-1-800	1 CPU at 800MHz, model string:"9000/800/rp3410"
	rp3410[+]-1-1000	1 CPU at 1000MHz, model string:"9000/800/rp3410"
rp3440[+].cfg (1)	rp3440[+]-1-800	1 CPU at 800MHz, model string:"9000/800/rp3440"
	rp3440[+]-1-1000	1 CPU at 1000MHz, model string:"9000/800/rp3440"
rp4410[+].cfg	rp4410[+]+-1-800	1 CPU at 800MHz, model string:"9000/800/rp4410"
	rp4410[+]-1-1000	1 CPU at 1000MHz, model string:"9000/800/rp4410"
rp4440[+].cfg (1)	rp4440[+]-1-800	1 CPU at 800MHz, model string:"9000/800/rp4440"
	rp4440[+]-1-1000	1 CPU at 1000MHz, model string:"9000/800/rp4440"
rp5400.cfg	rp5400-1-360	1 CPU, 360MHz, model string: 9000/800/L1000-36
	rp5400-1-440	1 CPU, 440MHz, model string: 9000/800/L1000-44
	rp5400-1-550	1 CPU, 550MHz, model string: 9000/800/L1000-5X
rp5430.cfg	rp5430-1-360	1 CPU, 360MHz, model string: 9000/800/L1500-36
	rp5430-1-440	1 CPU, 440MHz, model string: 9000/800/L1500-44
	rp5430-1-550	1 CPU, 550MHz, model string: 9000/800/L1500-5X
	rp5430-1-650	1 CPU, 650MHz, model string: 9000/800/L1500-6X
	rp5430-1-750	1 CPU, 750MHz, model string: 9000/800/L1500-7X
	rp5430-1-875	1 CPU, 875MHz, model string: 9000/800/L1500-8X
rp5450.cfg	rp5450-1-360	1 CPU, 360MHz, model string: 9000/800/L2000-36
	rp5450-1-440	1 CPU, 440MHz, model string: 9000/800/L2000-44
	rp5450-1-550	1 CPU, 550MHz, model string: 9000/800/L2000-5X
rp5470.cfg	rp5470-1-550	1 CPU, 550MHz, model string: 9000/800/L3000-5X
	rp5470-1-650	1 CPU, 650MHz, model string: 9000/800/L3000-6X

	rp5470-1-750	1 CPU, 750MHz, model string: 9000/800/L3000-7X
	rp5470-1-875	1 CPU, 875MHz, model string: 9000/800/L3000-8X
rp7400.cfg [2]	rp7400-1-360	1 CPU, 360MHz, model string:"9000/800/N4000-36
	rp7400-1-440	1 CPU, 440MHz, model string:"9000/800/N4000-44
	rp7400-1-550	1 CPU, 550MHz, model string:"9000/800/N4000-55
	rp7400-1-650	1 CPU, 650MHz, model string:"9000/800/N4000-65
	rp7400-1-750	1 CPU, 750MHz, model string:"9000/800/N4000-75
pa9-32.cfg	720	1 CPU, 50MHz, model string: "720"
To be added in v3.0.8. Contact Stromasys.	B132L-<*nr-of-cpus*>	1 CPU (up to 16 for oversized models), model string "9000/778"

[1] The **rp3440** and **rp4440** models may also be available in “oversized” versions up to 128 CPUs and 512GB RAM. Please check the availability for your model with your Sales representative. The +-versions of the rp34xx and rp44xx models emulate PA-8800 CPUs, the non-plus versions PA-8900 CPUs.

[2] Starting with version 3.0.5, the models above may also be available in “oversized” versions up to 64 CPUs and 512GB RAM. Please check the availability for your model with your Sales representative.

Memory Configuration

This section defines the RAM size for the emulated system. It only contains one parameter.

Syntax:

```
memory memsizeG
   or
memory memsize
```

Memory size can be specified in gigabytes (denoted by the letter **G**) or megabytes.

Parameters:

Default *memsize*: 2 gigabytes

Maximum *memsize*: maximum allowed RAM size for the emulated system. See section Emulated Models Supported by Charon-PAR for the maximum allowed values.

The configurable memory also depends on the amount of memory available on the host system.

I/O Slot Configuration

This section is used to load additional controllers for the emulated system.

The available expansion PCI slots depend on the configured hardware model (see configuration template files and chapter Emulated Model Hardware Configuration Details). The *Emulated Model Hardware Configuration* chapter also lists the **device paths** for all expansion slots of a model.

Supported controller types: SCSI controllers and Ethernet cards.

Syntax for Ethernet cards:

```
load ETH tulip PCI slot-number
```

Parameters for Ethernet cards:

The parameter *slot-number* identifies the number of PCI slot (see configuration file template and Emulated Model Hardware Configuration Details). All other parameters are fixed in the current version of the software.

Syntax for SCSI controllers:

```
load SCSI controller-model PCI slot-number
```

Parameters for SCSI controllers:

- *slot-number* identifies the number of PCI slot (see configuration file template and Emulated Model Hardware Configuration Details).
- *controller-model* can be
  53C875 for single-chip boards (assigned to PCI device function 0), or
  53C896 for a dual 53C895 controller (assigned to PCI device function 0 and 1)

System Console and Serial Line Configuration

Built-in Serial Lines

Most of the emulated systems have 2 serial lines. Please refer to Emulated Model Hardware Configuration Details for the device path associated with the built-in serial lines.

Configuration path for built-in serial lines on Charon-PAR/PA3 and Charon-PAR/PA9-64	Comment
serial.uart0.device	Console
serial.uart2.device	
Configuration path for built-in serial lines on Charon-PAR/PA9-32 (720)	
asp.uart0.device	Console
asp.uart1.device	
Configuration path for built-in serial lines on Charon-PAR/PA9-32 (B132L)	
gsc.lasi.uart.device	Console
gsc.wax.uart.device	

The configuration path string is used in the configuration file to specify parameters for the virtual serial line.

Super/IO Module

If the Super/IO module has been loaded, there are two additional serial lines. In the configuration, they are identified by **superio_001.uart0** and **superio_001.uart1**. The basic configuration options are the same as for the default serial lines.

Configuration Syntax

Syntax:

```
serial.uartX.device.<option>="<value>"

or

superio_001.uartX.device.<option>="<value>"
```

The options available for configuration are described below.

Serial port options

type

The **type** option specifies the serial line connection type.

Possible values:

DUMMY

- Default value
- The serial line is not connected to a port.
- The line has no input.
- The output will be discarded but can be logged to a file.

telnet

- Port is a network socket with telnet protocol support.
- Data is transmitted as a character stream encapsulated in the telnet protocol.
- UART baud rate and parity modes are ignored.
- If no client is connected, the output will be discarded.
- Only one client connection is allowed.

socket

- Raw network socket port mode.
- Data is transmitted as a character stream without UART signals.
- UART baud rate and parity modes are ignored.
- If no client is connected, the output will be discarded.
- Only one client connection is allowed.

pty

- Linux pseudo terminal device.
- UART baud rate and parity modes are ignored.
- Ignores serial port control signals changes.

tty

- Linux serial interface device.
- This port type implements all serial port control signals, baud rate changes, parity modes. Actual capabilities depend on the capabilities of the host hardware.

RFC2217

- Port is a network telnet client with RFC2217 support.
- The RFC2217 COM Port Control (a telnet protocol extension) allows sharing modems and other serial devices over a TCP/IP network.
- The emulated serial interface connects to a serial port server.
- This port type implements all serial port control signals, baud rate changes, parity modes. Actual capabilities depend on the capabilities of the host hardware.

port

The format and values of the **port** option, depend on the **type** of the port.

Type	Port settings
DUMMY	Any existing port settings are ignored.
socket, **telnet**	The port is mapped to a TCP socket on the emulator host. A TCP port number must be unique on the host system. The **TCP port number** is specified in the following format: `<where-to-listen>:<portnumber>` Possible values for *where-to-listen*: - Empty: - If the port number is specified **without** a colon (**:**), the interface process will only be accessible on the localhost address. Equivalent `lo:<portnumber>`. - If the port number is specified **with** a colon, the interface process will listen on all host interfaces. - Interface name: the interface process will only listen on the specified host IP interface. - IP address: the interface process will only listen on the specified host IP address. Default values: - ":30000" for line 1 - ":30002" for line 2
pty, **tty**	The port is mapped to a serial host interface (e.g., /dev/ttyS0, /dev/ttyUSB0). For type **pty**, a pseudoterminal device is created automatically if no device is specified. The format is `<device-path>` (as shown in the examples above). **Please note**: physical serial lines are not available in cloud environments.
RFC2217	The port is mapped to a serial port server on the network. The format is `<ip-address>`:`<tcp-port>`

command

This parameter is optional. It can be used to specify a terminal emulation program that is started automatically when the emulated system is started. Charon-PAR provides preconfigured profiles for PuTTY (PAR-Socket, PAR-Telnet, PAR-Telnet-VT100) that can be used to connect to the emulated system via a serial line.

stop_command

Boolean flag. If true, the command started on the serial line (parameter command) will be stopped when exiting from the emulator.

out_log

File name for logging the output character stream from the CPU to the device. Does not depend on the port being operational (the actual transmission is not logged). Depending on the data sent, the file may best be viewed using a tool such as **hexdump** or **od**.

in_log

File name for logging the input character stream from the port to the device. Best be viewed using a tool such as **hexdump** or **od**.

keep_on_reset

New in version 3.0.6. For physical serial ports. If set to 1, the emulator will keep the port speed, parity, stop bits, data bits on reset.

port_defaults

New in version 3.0.6. For physical serial ports: string defining the port default settings after a reset.

The format of the string is *<speed> <bits><parity><stop>* with the following possible values:

- speed: **300**, **600**, **1200**, **2400**, **4800**, **9600**, **19200**, **38400**, **57600**, or **115200**
- bits: **5**, **6**, **7**, or **8**
- parity values and the corresponding **stty** parameters:
  - **N:** `-parenb` (none)
  - **E**: `parenb -parodd -cmspar` (even)
  - **O**: `parenb parodd -cmspar` (odd)
  - **M**: `parenb -parodd cmspar` (mark)
  - **S**: `parenb parodd cmspar` (space)
- stop values and the corresponding stty parameters:
  - **1**: `-cstopb` (one stop bit)
  - **2** : `cstopb` (two stop bits)

Example: **`system.uart0.device.port_defaults = "115200 8N1"`**

Console Communication

Charon-PAR provides preconfigured PuTTY session templates for connecting to a serial emulator console port. The templates must be copied to **/root/.putty/sessions** or **/root/.config/putty/sessions** because the emulator must be started as the root user.

Method 1: TCP port 30000, raw data mode

```
serial.uart0.device.type="socket"
serial.uart0.device.port=":30000"
serial.uart0.device.command="putty -load PAR-Socket"
```

Method 2: TCP port 30000, Telnet protocol

```
serial.uart0.device.type="telnet"
serial.uart0.device.port=":30000"
serial.uart0.device.command="putty -load PAR-Telnet-VT100"
```

To connect to the console port without loading a PuTTY template, the command parameter can be set to

```
serial.uart0.device.command="putty"
```

To start the xhpterm emulator (Charon-PAR/PA3), the following setting should be used:

```
serial.uart0.device.command="./xhpterm -port 30000 -clean -font 10x20"
```

Please note::

- **If no command is specified**, a connection to the TCP port is possible by starting another terminal emulator manually.
- For device naming on Charon-PAR/PA9-32 refer to the serial line configuration section above.
- By default, **pressing Ctrl+C in the Charon-PAR console** is passed to the emulator's child processes and kills the PuTTY console process. If this is not desired, you can modify the command line to include the **setsid** command to start the telnet command in a new session.
  For example: **`serial.uart0.device.command="setsid putty -telnet -P 30000 localhost"`**

SCSI Controllers and Devices

Every Charon-PAR emulated model has preconfigured SCSI controllers that can be used to configure emulated disks for the guest system. All models also have the option to load additional controllers. Please refer to the section *I/O Slot Configuration* in this chapter.

Please note:

- The description below shows how to work with preloaded SCSI controllers. Once an additional SCSI controller has been loaded, the steps below also apply to such a controller.
- **It is not recommended** to place emulator storage devices (in particular vdisks) on NFS as this will have a significant impact on performance. However, if any of the storage (e.g., ISO files or vdisks) is on an NFS share, NFS locking must be enabled and all intermediate firewalls between client and server must allow the port used by the **lockd** and **statd**. Failure to do so will cause the emulator to hang at startup.

Preloaded SCSI Device Controllers

Charon-PAR implements I/O using controllers. Each controller

- supports up to 15 SCSI devices,
- is identified by a character code,
- is assigned a path corresponding to the leading for elements of the device path.

The tables below show the preconfigured SCSI controllers of an **rp2400 model as an example**.

Please note: Other models may have more or less preconfigured controllers with different device paths and code mappings. For details regarding your model refer to the configuration file template or the chapter Emulated Model Hardware Configuration Details.

Code	Module path (rp2400)	Controller model
A	0/0/1/0	53C875
B	0/0/1/1	53C875
C	0/0/2/0	53C896
D	0/0/2/1	53C896

Load Device Command for SCSI Devices

The **load** command is used to define SCSI devices.

Syntax:

```
load DDXnnn
```

DDXnnn represents the device name. It has the following components:

Component	Value	Description
DD	**SCSI device type**	
	DK	Disk device mapped to file or physical device
	MK	Tape device mapped to file or physical device
	GK	Generic SCSI device mapped to physical SCSI target
X	**Controller Code** The rp2400 model is used as an example below; refer to the template configuration file of your model and to Emulated Model Hardware Configuration Details for model specific details.	
	A	rp2400 controller device path: 0/0/1/0
	B	rp2400 controller device path: 0/0/1/1
	C	rp2400 controller device path: 0/0/2/0
	D	rp2400 controller device path: 0/0/2/1
nnn	**SCSI target ID and LUN**	
	(SCSI ID * 100) + LUN	Identifies the SCSI device connected to a SCSI controller

Examples:

To load a disk on device path 0/0/1/0.1.2 (SCSI target ID 1, LUN 2), add the command

```
load DKA102
```

To load a tape on device path 0/0/1/1.12.0 (SCSI target ID 12, LUN 0), add the command

```
load MKB1200
```

The image Subcommand

After a SCSI device has been loaded, the associated container file or device must be specified. **Please note**: physical tape or CD-ROM drives are not available in cloud environments.

Please note: the **image** subcommand is for **MK** and **DK** devices only.

Format:

```
device.image="<path-to-device-or-file>"
```

Parameters:

- *device*: the virtual SCSI device created by the **load** command. See the **load** command for details about the device name syntax.
- *path-to-device-or-file*: the path to the Linux device (a physical disk device or disk partition) or container file. For physical devices it is strongly recommended to use a persistent device name from
  - /dev/disk/by-id, or
  - /dev/disk/by-uuid
    instead of a non-persistent /dev/sdX device name.

Example:

To associate a disk device on 0/0/1/0.0.0 (controller code A, SCSI target ID 0, LUN 0) with a container file called *ldev1.dsk*, add the command

```
DKA0.image="ldev1.dsk"
```

The devname Subcommand

Use the **devname** subcommand to specify the device to be used for a generic SCSI device (**GK** device). **Please note**: physical tape or CD-ROM drives are not available in cloud environments.

Please note: A generic SCSI device can be a tape or disk device. Guest system SCSI commands are passed through to the device directly and not handled by the Linux disk or tape driver. The functionality depends on the device understanding the guest system SCSI commands.

Format:

```
GKXnnn.devname="<path-to-device>"
```

Parameters:

- **GK***Xnnn:* the generic virtual SCSI device created by the **load** command. See the **load** command for details about the device name syntax.
- *path-to-device*: special filename of the generic SCSI device (/dev/sg*X*) on Linux, for example: /dev/sg0.

Tape autoload Subcommand

To enable **tape container files** to be automatically loaded as tapes into a virtual tape device, the **autoload** command can be added for a specific tape device in the configuration file. Not applicable to physical tape devices.

Format:

```
MKXnnn.autoload=yes|no
```

Parameters:

MK*Xnnn*: the emulated tape device. Possible values for X and nnn see above.

Default: autoload is off by default.

Please note:

- The autoload command works well in many situations but can cause problems in some cases, for example with multi-volume backups. Any applications using tape devices should be tested thoroughly to verify if the autoload option is working as expected.
- MPE/iX can load a tape using operating system utilities. HP-UX cannot. Hence, the autoload command may be more often useful in an HP-UX environment.

Tape size Subcommand

The **size** command limits the maximum size of a **tape container file**. Not applicable to physical tape devices.

Format:

```
MKXnnn.size=<size-in-bytes>
```

Parameters:

- **MK***Xnnn*: the virtual SCSI tape device created by the **load** command. See the **load** command for details about the device name syntax.
- *size-in-bytes*: maximum size of the tape container file.

Default: tape container file size is not limited.

Tape Subcommands for Physical Tape Devices

The following two commands can only be used for physical tape devices:

Subcommand	Description
MK*Xnnn***.fast=true|false**	Optimization for writing EOF marks. Do not use for old tape drives. Default is off.
MK*Xnnn***.nounload=true|false**	Set to true for DDS tapes. Set to false for DLT tapes. Default is on.

Typical System Disk Configuration

The system disk is usually configured on device path 0/0/1/0.0.0. To load this device and to map it to a disk image file, add the following commands:

```
load DKA0

DKA0.image="/data/virtual-disks/ldev1.dsk"
```

Network Card Configuration

General information

The emulated Ethernet interfaces of Charon-PAR can be linked either to a

- physical host interface,
- a TAP interface connected to a virtual bridge on the host, or
- (since Charon-PAR 3.0.9) a MACVTAP interface linked to a physical NIC on the host.

Notes for physical interfaces:

- If used directly as a guest interface, offloading parameters must be turned off using the initialization command parameter in the configuration file.
- In cloud environments, the necessary interface settings are not always reflected correctly on the cloud-provided NICs. For such installation settings a MACVTAP interface can be used to provide a dedicated connection without connecting the emulator directly to the physical interface.

Notes for TAP interfaces:

- A TAP interface can either be created by the user or will be created automatically by the emulator if it does not already exist.
- For automatically created TAP interfaces, the user can specify a name or let the emulator select a name.
- A automatically created TAP interface is not automatically added to a bridge, this must be configured via the initialize_command (see the Ethernet Configuration Parameter table below).
- An automatically created TAP interface is deleted automatically upon emulator stop.
- If used in a cloud environment, only internal virtual bridges (no connection to the cloud LAN) are supported.

Notes for MACVTAP interfaces:

- Introduces a transparent bridge interface associated with on physical NIC on the host, and provides a dedicated interface to the emulator.
- Provides an abstraction layer between physical NIC and emulator.
- Physical interface requires no changes in the offloading configuration.
- Fragmentation is performed by Linux on the MACVTAP interface. Jumbo frames on the physical interface will not cause problems.
- Uses same MAC address as the physical interface.
- Suited for associating a dedicated host Ethernet interface to the emulator in cloud environments.

Notes for network configurations on VMware:

- ESXi has configuration parameters that improve security but may block certain emulator NIC configurations.
- One configuration option allows or blocks promiscuous mode (the default mode of operation for Charon-PAR guest NICs).
- Other configuration options allow or block the change of a MAC address or the use of a different, additional MAC address on the same NIC (*MAC address changes*, *Forged transmits*).

By default, emulated models have one Ethernet device. Depending on the model, more Ethernet devices can be added.

A guest HP-UX system will perceive the network card as a 10/100 Mbit/s controller running at 10 Mbit/s half-duplex. If the user tries to change this setting using SAM or the **lanadmin** command, the command will be accepted but the displayed interface settings will not change. However, the throughput of the emulated network card depends on the combination of network performance, physical network card characteristics, host system and guest system performance - it is not tied to the displayed interface settings. So the actual throughput will be what can be achieved depending on the conditions listed above.

The emulated network devices

- DE 500 PCI based cards (for 64-bit systems), and
- LASI-82596 cards (for the 32-bit system)

do not support Jumbo frames. For physical interfaces, this feature must be disabled in the emulator configuration (together with any other offloading parameters) using the **initialize_command** parameter.

Ethernet Card Device Names

Names on 64-bit systems:

The name of the Ethernet interface in the emulator configuration file has the format **EW***xn* with the following definitions:

- *x* is an uppercase letter starting with **A** for the first interface and then continues with **B**, **C**, etc. for additional interfaces. The possible number of network cards depends on the features of the original physical system. The absolute maximum number is 16.
- *n i*s the device number of the card starting with 0 for each value of *x*.

Names on 32-bit systems:

The currently supported 32-bit system supports only one Ethernet card:

- Model 720: **system.lan0.card**
- Model B132L: **gsc.lasi.lan.card**

Ethernet Configuration Parameters

This section lists the available Ethernet interface configuration parameters.

Basic syntax:

<device-name>.<parameter>="<value>"

The *device-name* specifies the name of the Ethernet card as described above in the *Ethernet Card Device Names* section.

The parameters and their values are described in the following table:

Parameter	Values
mapping_mode	Possible values: • **RAW**: the emulated Ethernet card is mapped to a physical interface on the Charon host that is dedicated to the Charon guest system. *This setting is the default value.* • **TAP**: the emulated Ethernet card is mapped to a TAP device on the Charon host. If the parameter **iface** is not set or the specified interface does not exist, it is created by the emulator. • **MACVTAP**: links the emulated NIC to a physical NIC on the host via an automatically configured TAP interface. Different from a bridge, it allows using the same MAC as the host NIC when used in passthrough mode. This makes it suitable for cloud environments. MACVTAP mode requires at least Charon-PAR version 3.0.9. • **DUMMY**: presents a network interface without any actual network connectivity to the guest system. All other Ethernet parameters are ignored if this parameter is set. It will be set automatically, if • **iface**="dummy", • or **iface**="" and **mapping_mode**="RAW".
iface	Specifies the name of the host interface mapped to the emulated interface. **For mapping_mode="RAW" or "MACVTAP"**: this parameter must be set to an interface dedicated to the emulator. **For mapping_mode="TAP"**: the parameter can be • *empty*: the emulator will choose a name for the TAP interface and create it automatically at startup (and delete it when stopped). • *name of non-existing TAP interface*: the emulator will create the interface automatically at startup (and delete it when stopped). • *name of an existing TAP interface*: the emulator will use this interface. **For any mapping_mode:** the value **dummy** will cause the emulator to present a dummy NIC to the guest system. Upon emulator start, Charon-PAR will set a variable **IFACE** to contain the interface name. This variable can then be used in the initialization command to refer to the interface name.
macaddr	Can be set to override the default (same as host interface) MAC address of the interface. By default**,** if this parameter is used together with mapping_mode RAW, the interface is put into promiscuous mode, and the original MAC address of the interface is not overwritten.

	If the **legacy_mode** is configured for the interface, the interface is not set into promiscuous mode, and the MAC address of the corresponding host NIC is changed to the configured address. The original MAC address is restored upon emulator exit. The format of the MAC address is: XX-XX-XX-XX-XX-XX where X is a hexadecimal number. Please note: the above behavior is new starting with version 3.0.1 build 21.500. In older versions, when the MAC address was changed, it was not reset automatically when the emulator was stopped.
initialize_command	Command(s) to initialize the host interface for use with the Charon emulator. - **For physical interfaces (RAW)**: all offload parameters active on the interface should be turned off, as shown by the example in the configuration file template. - **For TAP interfaces**: the TAP interface should be activated, if needed, and added to the appropriate bridge interface (example: **`"ip link set $IFACE up; ip link set $IFACE master br0"`**); the exact configuration depends on the customer requirements. - Not needed if a MACVTAP interface is used. The variable **IFACE** is set by the emulator and defined automatically for the execution of the initialization command.
adapter_mode	Defines the speed and duplex settings of the interface that are reported to the guest system. Possible values: **auto**, **10BaseT-HD**, **10BaseT-FD**, **100BaseT-HD**, **100BaseT-FD** Default: **100BaseT-FD**
legacy_mode	New starting with version 3.0.1 build 21.500. **Only applicable to mapping_mode RAW with a manually configured MAC address**. Possible values: - **true** - non-promiscuous mode; if a custom MAC address was set, it will be set on the physical NIC during emulator operation and reset to the burned-in MAC address when the emulator stops. - **false** - promiscuous mode; if a custom MAC address was set, it will not be be set on the physical NIC (not required for the guest communication due to promiscuous mode) Default: **false**
pkg_dump	Can be used to enable a packet dump to the log file. Possible values: **true** or **false** Default: **false**
rx_fifo_size	Can be used to set the size of the RX FIFO buffer (in KB). Do not change unless advised to do so by Stromasys support. Possible values: **0,** or **16** to **1024** Default: **0** = disabled
ignore_tx_start_error	Possible values: **true** or **false** Default: **true** By default, this parameter enables a workaround for DMA failures at the adapter startup. If needed, the workaround can be disabled by setting the parameter to **false**. Introduced by build 3.0.6-22001.

Logging configuration

This section defines how Charon-PAR handles logging. Note that the default log file location is the directory in which the emulator was started. This can be changed with the emulator command-line option **`-l`** (or **`--logfile-path`**).

Log File Parameters

The following table shows the parameters relevant to the log file and their values:

Parameter	Description
`log.name="`*`path/log-file-name`***`"`**	*log-file-name* specifies the base name of the log file. The full log file name will be created using the base name combined with date, time and serial number. Default: **charon-par.*X*_log**. A link with the name *`log-file-name`***`.log`** will be created that points to the active log file. After a log rotation, the link will be set to the new active log file. The *`path`* specification is optional. It can be absolute or relative and indicates where the log file and the log-file link will be created. Default is the directory in which the emulator is started. The specified directory must exist - otherwise the emulator will not start.
`log.on_console= false | true`	Defines if logging output should be sent to the Charon-PAR console. Default: **true**.
`log.disable= false | true`	Enable or disable logging. Default: **false**.
`log.no_rotate= false | true`	If set to **false**, once the line limit for a log file has been reached, the old log file will be closed and a new one will be opened with the last number in the log-file name incremented.. Default: **false**.
`log.line_limit =` *`value`*	Maximum number of lines in the log file before rotation. Default: **100000**.

Log File Format

Starting with version 3.0.1 the log file has the following format:

```
YYYYMMDD:hhmmss.uuuuuu:<severity><message>

   where

YYYY   - year
MM     - month
DD     - day
hh     - hour
mm     - munites
ss     - seconds
uuuuuu - usecs

<severity> := '(warn|err|ERR):' or empty

warn   - warning
err    - error
ERR    - fatal error

<message> the component's message in free form
```

Parameter system.license_logging_level

Starting with Charon-PAR version 1.11, periodic (once per hour) log messages indicating a successful periodic license check are disabled by default to avoid cluttering the log file. They can be re-enabled by adding this parameter to the emulator configuration file.

Syntax:

```
system.license_logging_level = value
```

Parameters:

- If *value* is set to 2, the messages are enabled.
- If *value* is set to 0 or 1, the messages are disabled.

Please note that this parameter must be placed towards the end of the configuration file and will not work if placed before the memory configuration.

SuperIO Configuration

General Information

The SuperIO module is included in the Astro chipset and emulates a PCI board with PC style peripherals used in historic 64-bit PA-RISC systems:

- parallel port
- 2 serial ports
- dual-channel IDE controller
- floppy disk controller
- USB controller
- timer
- PIC interrupt controller

The current version of Charon-PAR supports only a subset of these devices. **And the SuperIO module is only supported on models that are based on the Astro chipset (rp24xx and rp54xx).**

Currently supported are (depending on support by the emulated model and guest operating system)

- two serial ports
- one parallel port

Loading the SuperIO device

Before any devices of the module can be configured and used, the module must be loaded in the emulator configuration file.

The SuperIO module can be loaded in any system model with PCI bus support.

Use the following syntax to load this module into a slot on the default bus:

```
load SUPERIO sio <bus-number> <slot-number>
```

Example:

```
load SUPERIO sio 0 6
```

Use the following syntax to load this module into an PCI expansion slot:

```
load SUPERIO sio PCI <slot-number>
```

where the slot number is the number listed in the configuration file template or in Emulated Model Hardware Configuration Details for the expansion I/O slot selected.

Please note:

- Only one SuperIO module can be loaded in the configuration. Multiple SuperIO instances will not work properly.
- The usable bus numbers and slot numbers depend on the emulated model and the already loaded devices.
- If another PCI device is already installed in the specified PCI slot, the load command will fail with an error message.
- **The correct bus/slot location of the device is important to preserve the correct GSP (service processor) console configuration**:
  - If the SuperIO module is inserted in a slot before the normal system serial console, the service processor console (GSP console) is set to the first SuperIO serial port.
  - HP-UX does not support the SuperIO serial port as GSP console. With such a configuration, it will crash early-on in the boot process. Therefore, take care to **always use a slot number higher than the one where the correct system console line has been loaded**.

After booting the guest HP-UX system, the example above will result in an **ioscan** output similar to the following:

```
ba          1  0/0/6/1      superio   CLAIMED     BUS_NEXUS    PCI Core I/O Adapter
tty         1  0/0/6/1/1    asio0     CLAIMED     INTERFACE    Built-in RS-232C
tty         2  0/0/6/1/2    asio0     CLAIMED     INTERFACE    Built-in RS-232C
ext_bus     4  0/0/6/1/3    SCentIf   CLAIMED     INTERFACE    Built-in Parallel Interface
unknown    -1  0/0/6/1/4              UNCLAIMED   UNKNOWN      Built-in Floppy Drive
```

SuperIO Serial Ports

Serial ports installed on SuperIO module can be configured the same as other serial ports. Please see chapter Serial Line Emulation Notes for more information.

Path names for serial devices in the emulator configuration:

- superio_001.uart0 - COM0 port
- superio_001.uart1 - COM1 port

Serial ports configuration example:

```
superio_001.uart0.device.type="telnet"
superio_001.uart0.device.port=":30001"

superio_001.uart1.device.type="telnet"
superio_001.uart1.device.port=":30002"
```

SuperIO Parallel Port

The parallel port installed on a SuperIO module connects to the host system parallel port device **`/dev/parport0.`** The parallel port output from the guest OS is redirected to the host parallel port. No additional configuration for the parallel port is required.

However, it may be necessary to install the required kernel drivers in HP-UX and rebuild the kernel to activate the interface.

If the **ioscan** command shows the parallel port as **UNCLAIMED**, perform the following steps:

1. Use the kcmodule command to verify the status of the required drivers:
   ```
   # kcmodule |grep CentIf
   CentIf static explicit
   SCentIf static explicit
   ```
2. If the status is unused, load the modules into the kernel using the commands:
   ```
   # kcmodule CentIf=best
   # kcmodule SCentIf=best
   ```
3. You will be informed that this change can only become active after the next reboot. If you confirm, the kernel will be modified and you can reboot the system.

Once the **ioscan** command shows that the parallel port as **CLAIMED** by the correct driver, you can, for example, use **SAM** to configure a printer on the parallel port
(**Printers and Plotters > LP Spooler > Printers and Plotters > Action > Add Local Printer > Add Parallel Printer**). On HP-UX 11.31, use **SMH** for this task. Please note that at the time of writing SMH only worked with the emulator when being used over the network (not on the serial console).

If no parallel port device exists on the host system, an error message similar to the one below will be printed to the emulator log. It does not affect system operation, but the parallel port redirection will not work.

```
err:open('/dev/parport0', O_RDWR) is failed (errno 2) No such file or directory
err:ioctl(handle, PPCLAIM) is failed (errno 9) Bad file descriptor
```

Modern host systems often have no physical parallel port. In such cases, a USB-LPT adapter or a software redirector (such as LPT-over-IP) can be used. The emulated parallel port operation depends on the host parallel port operation and some USB-LPT adapters may not produce reliable results.

Other Parameters

Parameter system.do_timer_correction

If the system time of the emulated system deviates too much from the correct time, it can cause application problems in the emulated system. If this cannot be solved by other means (e.g., NTP), the parameter described here can be used to adjust the system time of the emulated system based on the host system's NTP adjusted time.

Syntax:

```
system.do_timer_correction = false | true
```

Parameters:

If set to **false**, no time correction will take place.
If set to **true**, the time of the emulated system will be corrected as described above.

Default: false

Parameter fma_check

Charon-PAR requires a host CPU that supports the FMA capability. Normally, an emulator will not run when trying to start it on a CPU without the required capabilities.

This parameter allows you to disable the FMA check. However, this may lead to unexpected problems with the guest operating system.

Please note: Do not use this parameter if you run PA9 instances. For PA3 instances, use the configuration option at your own risk.

Syntax:

```
fma_check = false | true
```

Parameters:

If set to **false**, Charon-PAR will not check the FMA support of the host CPU.

Default: the check is enabled.

Parameters system.stop_on_halt and system.stop_on_halt_timeout

These parameters are applicable to emulated systems in which a HP-UX guest operating system runs.

When HP-UX is shut down with the `shutdown -h` command, the guest system will be halted. By default, the emulated system will be powered off automatically when a halt is detected (that is, the Charon-PAR process will be stopped).

Depending on the Charon-PAR version one of the following two parameters can be used to influence this behavior.

Charon-PAR Versions 3.0.5 to 3.0.12 (system.stop_on_halt)

The **system.stop_on_halt** parameter has the following syntax:

Syntax:

```
system.stop_on_halt = false | true
```

Parameters:

If set to **false**, Charon-PAR will not stop the emulated system automatically after the guest operating system has been halted. The emulator must be stopped from the Charon-PAR console (at the pa9-64> or pa9-32> prompt).

Default: **true**. the emulated system will be stopped automatically after the HP-UX guest operating system has been halted.

Charon-PAR Versions 3.0.13 and Higher (system.stop_on_halt_timeout)

The **system.stop_on_halt_timeout** parameter has the following syntax:

Syntax:

```
system.stop_on_halt_timeout = <number-of-seconds>
```

Parameter:

If *number-of-seconds* is set to **0**, Charon-PAR will not stop the emulated system automatically after the HP-UX guest operating system has been halted. The emulator must be stopped from the Charon-PAR console (at the pa9-64> or pa9-32> prompt).

If *number-of-seconds* is set to **a value higher than 0**, Charon-PAR will stop the emulated system automatically after the HP-UX guest operating system has been halted using a timeout corresponding to the specified number of seconds.

This parameter has no default value. To restore the default behavior of the system, comment out or remove the parameter from the emulator configuration file.

Parameter system.do_affinity

If enabled, the emulator will calculate a CPU affinity mask based on the host system and the emulator configuration. The affinity mask shows which CPUs will be used for emulated CPUs and which for I/O.

Syntax:

`system.`**do_affinity = false | true**

Parameters:

Default: **false**. The emulator will not follow any preferred host CPU affinity settings.

If set to **true**, Charon-PAR will calculate a hexadecimal affinity mask according to which emulated CPU and I/O threads will be distributed across the Charon host CPUs. The calculated mask is shown in the emulator log file.

Every bit in the calculated affinity mask corresponds to one host CPU (counting from 0). For example: 00000001 corresponds to to CPU 0, 00000003 corresponds to CPU 0 and 1.

The CPUs that can be used by the emulator can be limited by starting the emulator with an explicit affinity mask defined by the **taskset** command.
Example: `# taskset 2f /opt/charon/bin/charon-par -f myconfigfile.cfg`
This would set the affinity mask such that the emulator would only use CPUs 0-3 and 5. If **do_affinity** is enabled, the automatic calculation will be restricted to the CPUs allowed by the **taskset** command.

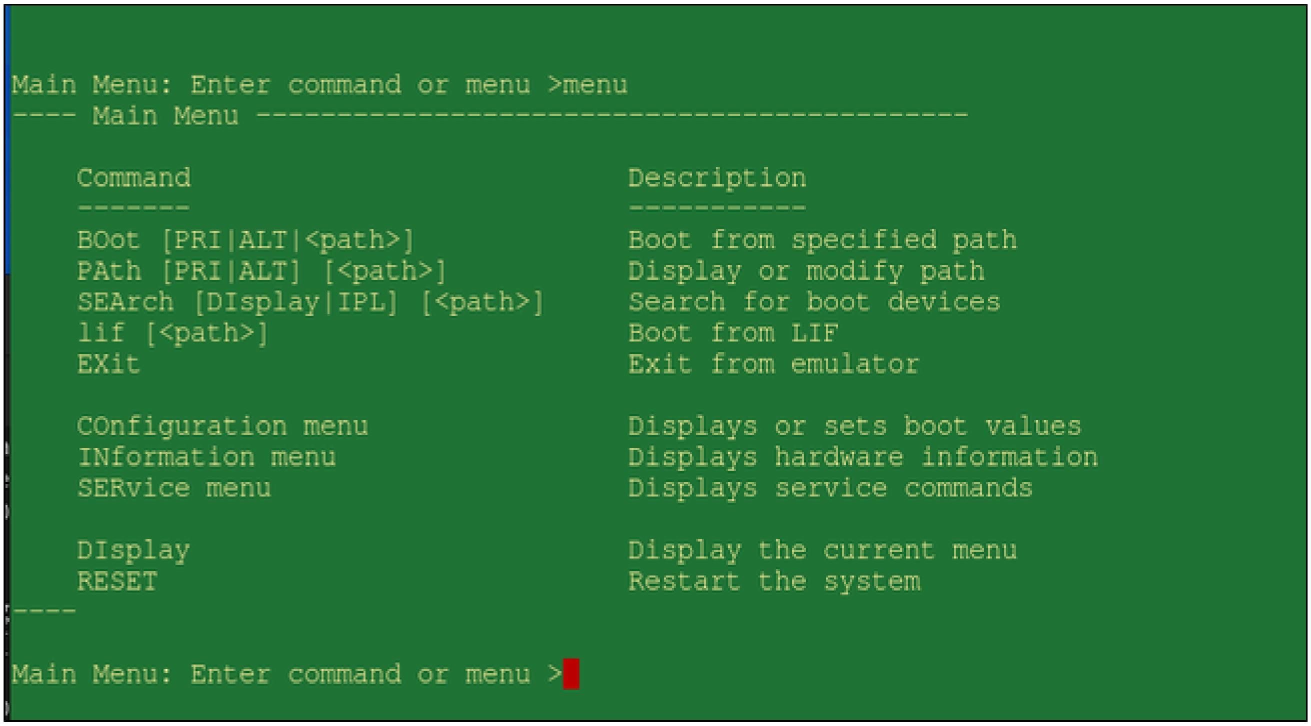
Emulated System Firmware Functions

When a Charon-PAR instance is started, the first interface visible to the user is the PDC (processor dependent code) console of the emulated system (i.e., the firmware of the system). It provides some important function that will be described below:

- PDC Main and Configuration Menus
- ODE MAPPER2 Program

PDC Main and Configuration Menus

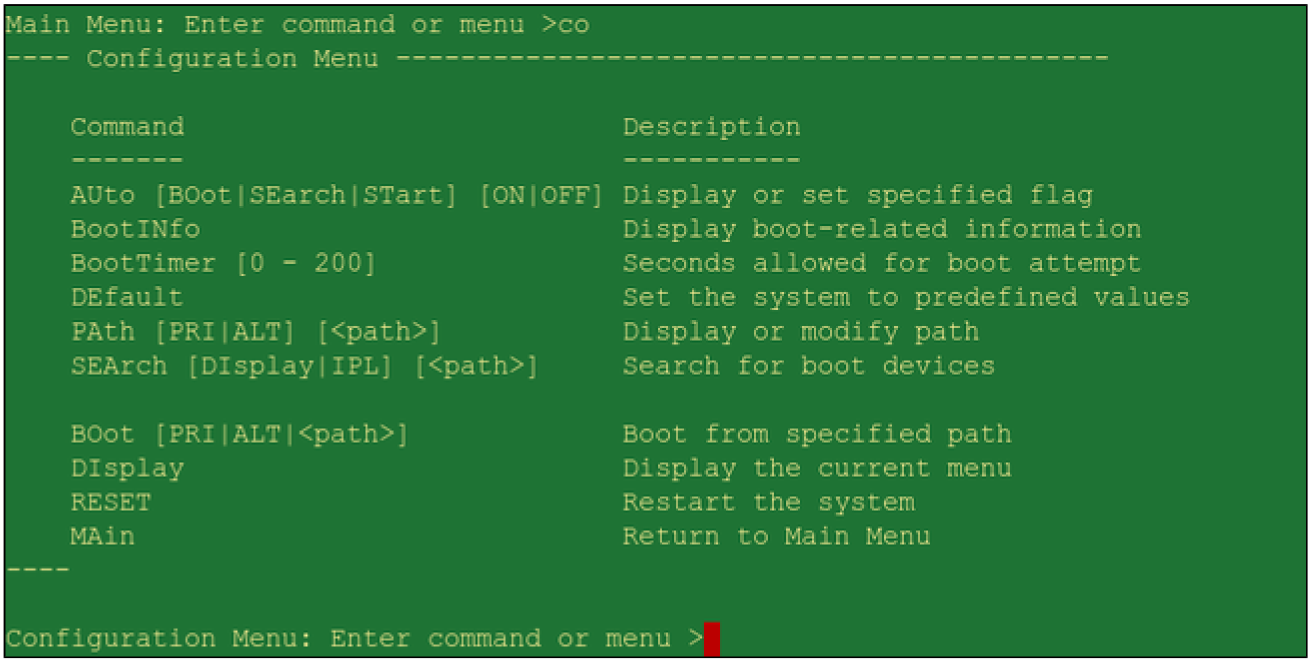
At the console prompt **`Main menu: Enter command or menu >`** the user can enter either one of the available commands directly or display the **main menu** by entering **`menu`**:



```
Main Menu: Enter command or menu >menu
---- Main Menu --------------------------------------------

    Command                           Description
    -------                           -----------
    BOot [PRI|ALT|<path>]             Boot from specified path
    PAth [PRI|ALT] [<path>]           Display or modify path
    SEArch [DIsplay|IPL] [<path>]     Search for boot devices
    lif [<path>]                      Boot from LIF
    EXit                              Exit from emulator

    COnfiguration menu                Displays or sets boot values
    INformation menu                  Displays hardware information
    SERvice menu                      Displays service commands

    DIsplay                           Display the current menu
    RESET                             Restart the system
----

Main Menu: Enter command or menu >
```

Please note: Commands can be entered by **typing the full command or the abbreviation indicated by the uppercase letters**. The commands must be entered in lowercase.

There is an **overlap** between menus, as shown by the configuration menu (type **co** or more letters of the command) below:



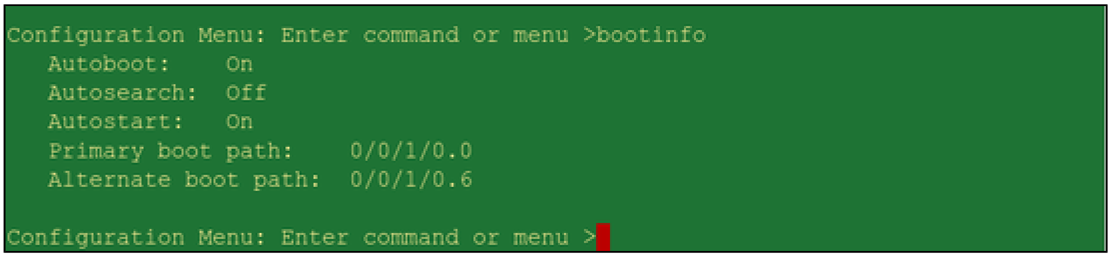
```
Main Menu: Enter command or menu >co
---- Configuration Menu --------------------------------------------

    Command                            Description
    -------                            -----------
    AUto [BOot|SEarch|STart] [ON|OFF]  Display or set specified flag
    BootINfo                           Display boot-related information
    BootTimer [0 - 200]                Seconds allowed for boot attempt
    DEfault                            Set the system to predefined values
    PAth [PRI|ALT] [<path>]            Display or modify path
    SEArch [DIsplay|IPL] [<path>]      Search for boot devices

    BOot [PRI|ALT|<path>]              Boot from specified path
    DIsplay                            Display the current menu
    RESET                              Restart the system
    MAin                               Return to Main Menu
----

Configuration Menu: Enter command or menu >
```

Important commands in the main menu:

Command	Description
`boot pri`	Boot from the primary boot device.
`boot alt`	Boot from the alternative boot device.
`boot` *device*	Boot from device with specified device path. Example device path: 0/0/1/0.0.0
`path pri`	Display primary boot path.
`path alt`	Display alternative boot path.
`path` <**pri** | **alt**> *path*	Set the boot path. If neither PRI nor ALT is specified, the primary path will be set.
`search`	Search for boot devices
`lif` *path lifimage*	Boot a lif file. There are currently two use cases: • When used to boot a single image, it is mostly used for testing. Use only as advised by Stromasys. The *path* parameter is a SCSI device path for 64-bit systems, on a 32-bit system it can be the device path of a network card or a SCSI device path. The *lifimage* parameter contains the path of the bootable lif file (absolute path or path relative to the directory in which the emulator runs). • Used to emulate booting a system from a Ignite-UX server (for example, to install the operating system). In this case, the content of the TFTP boot directory is downloaded from the Ignite server to a path accessible by the emulator. The feature is planned for version 3.0.9. Please refer to *Installing a client system from an Ignite-UX server* for more information.
`reset`	Restart the system.

Important commands only available in the configuration menu:

Command	Description
`auto`	Display the autoboot settings.
`auto boot on | off`	Enable or disable system autoboot.

`bootinfo`	Displays a summary of boot parameter and paths as shown below:
	
	`Configuration Menu: Enter command or menu >bootinfo`
	`   Autoboot:     On`
	`   Autosearch:   Off`
	`   Autostart:    On`
	`   Primary boot path:     0/0/1/0.0`
	`   Alternate boot path:   0/0/1/0.6`
	
	`Configuration Menu: Enter command or menu >`

`default`	Reset the parameter values to the predefined settings.

Please note: the service menu and the information menu currently do not contain important additional commands.

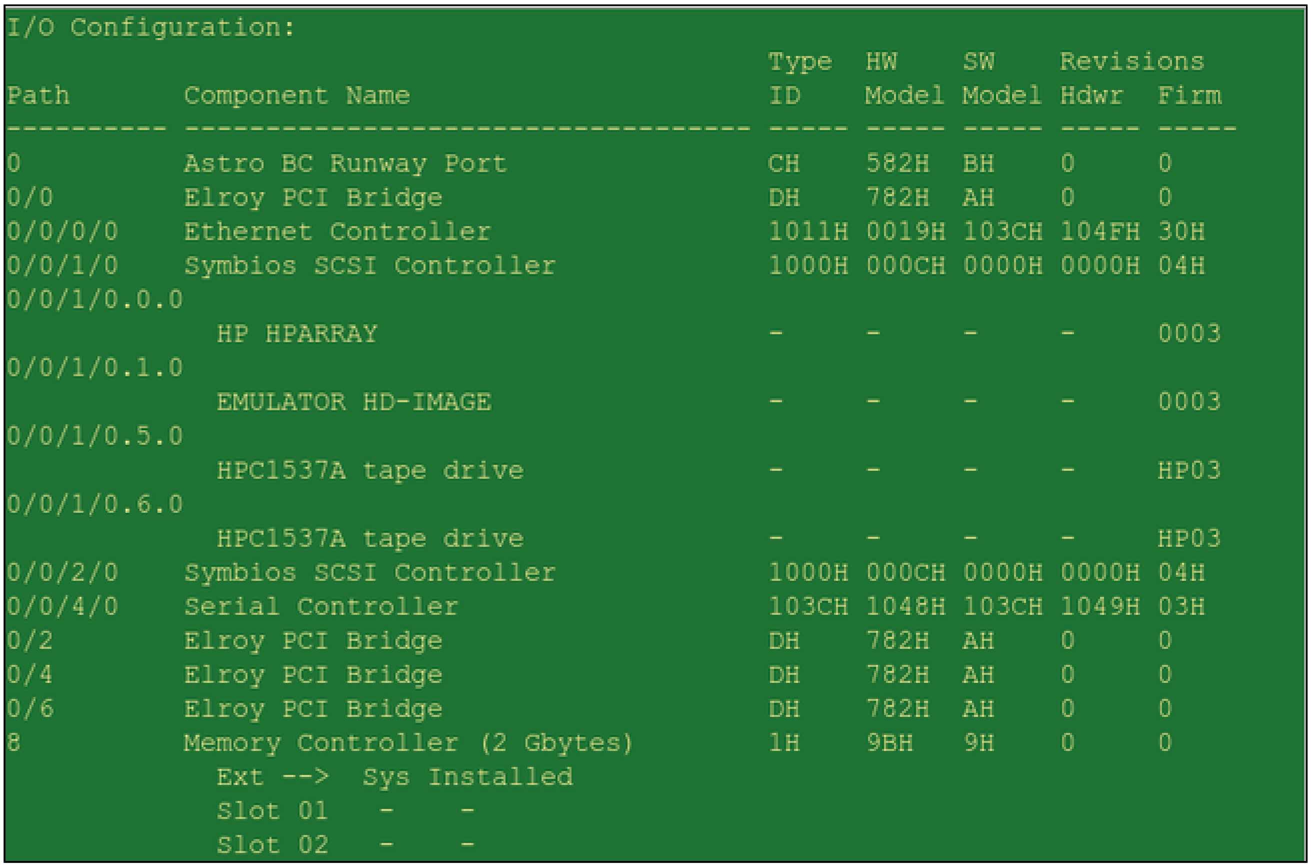
ODE MAPPER2 Program

The MAPPER2 program can be used to report all the emulated hardware of the emulated historic PA-RISC system.

To start it, perform the following steps:

1. Start the Charon instance.
2. At the PDC menu, enter **boot pri** to boot from the main path.
3. At the prompt **`Interact with IPL (Y or N)? >`**, enter **`Y.`**
4. At the **`ISL>`** prompt, enter the command **`ODE`**.
5. At the ODE> prompt, enter **`RUN MAPPER2`**.

The program will display the hardware found on the system. A partial output is shown below:



```
I/O Configuration:
                                               Type  HW    SW    Revisions
Path       Component Name                      ID    Model Model Hdwr  Firm
---------- ----------------------------------- ----- ----- ----- ----- -----
0          Astro BC Runway Port                CH    582H  BH    0     0
0/0        Elroy PCI Bridge                    DH    782H  AH    0     0
0/0/0/0    Ethernet Controller                 1011H 0019H 103CH 104FH 30H
0/0/1/0    Symbios SCSI Controller             1000H 000CH 0000H 0000H 04H
0/0/1/0.0.0
             HP HPARRAY                        -     -     -     -     0003
0/0/1/0.1.0
             EMULATOR HD-IMAGE                 -     -     -     -     0003
0/0/1/0.5.0
             HPC1537A tape drive               -     -     -     -     HP03
0/0/1/0.6.0
             HPC1537A tape drive               -     -     -     -     HP03
0/0/2/0    Symbios SCSI Controller             1000H 000CH 0000H 0000H 04H
0/0/4/0    Serial Controller                   103CH 1048H 103CH 1049H 03H
0/2        Elroy PCI Bridge                    DH    782H  AH    0     0
0/4        Elroy PCI Bridge                    DH    782H  AH    0     0
0/6        Elroy PCI Bridge                    DH    782H  AH    0     0
8          Memory Controller (2 Gbytes)        1H    9BH   9H    0     0
             Ext -->  Sys Installed
             Slot 01   -    -
             Slot 02   -    -
```

After the program has completed, type **`exit`** at the ODE prompt to return to the ISL prompt, or type **`exit`** in the Charon-PAR console to stop the emulator.

MPE/iX Command Files & UDCs

The preconfigured MPE/iX system for Charon-PAR/PA3 provides several command files and UDCs (User Defined Commands). For the most part, all customizations are located in the CHARON.SYS group.

This section provides an overview of the following utilities:

- Configuring the IP Address
- Configuring the Time Zone
- Synchronizing the System Time
- Default UDCs

Configuring the IP Address

The emulator needs a valid license to run. The license can be a Sentinel HASP license or a Virtual Environment (VE) license. Please refer to Initial Emulator License Configuration for the basic license configuration steps of both options. The license(s) to be used are defined in the configuration file. If running in a cloud environment, a VE license is strongly recommended (supported from Charon-PAR version 3.0.6).

Configuring the Time Zone

The preconfigured MPE/iX system is configured with TZ="PST8PDT" .

The supplied LOGON UDC (which executes out of UDC.CHARON.SYS) runs the command file TZ.CHARON.SYS which in turn sets the TZ system variable local to every Job and Session:

```
:PRINT TZ.CHARON.SYS
setvar tz 'PST8PDT'
```

To change the time zone, edit the file TZ.CHARON.SYS, and adjust the system clock using the commands:

```
:SETCLOCK TIMEZONE=Wxx:yy
:SETCLOCK ;CANCEL
```

xx:yy stands for the time offset of your timezone.

Synchronizing the System Time

The preconfigured system contains the :SYNC utility to update the system time. It must be run as manager.

```
:HELLO MANAGER.SYS
:SYNC.CHARON
```

If :SYNC detects that the system date was updated, it displays the old and new timestamp:

```
:sync.charon.sys
Time changed from SAT, JAN 1, 2000, 12:00 PM
               to SAT, FEB 2, 2013, 9:40 PM
```

Otherwise it will just display the current date.

Notes:

- By default, :SYNC performs a :SETCLOCK;NOW to adjust any large discrepancies between the system clock, and the current time. It should only be used when starting or resuming an emulated system, and should not be used while date-aware programs or command files are running.
- :SYNC should not be used at or close to midnight - this could result in an incorrect date being set.
- :SYNC is supplied 'as-is', with no warranty as to correct performance.

Default UDCs

The Charon-PAR preconfigured MPE/iX system includes the following predefined UDCs.

Stromasys provided UDCs in UDC.CHARON.SYS:

Command	Description
LOGON	Logon UDC
SH	Posix Shell access
ABORTCON	ABORTCON script front-end
FSCHECK	Shortcut to FSCHECK
SHOWCONN	SHOWCONN script
SHOWCLKS	Shortcut to SHOWCLKS
NSLOOKUP	Shortcut to NSLOOKUP
ABORTJ	ABORTJ script
PRINTO	"Tail" on running/terminated job
HELLO	HELLO command with default ID
L	Shortcut to LISTF,2
SHUTDOWN	Access to :SHUTDOWN command

The original vendor provided Posix "smoothing" commands in HPPXUDC.PUB.SYS: PLISTF, FINDDIR, FINDFILE, LISTDIR, DISKUSE, SH, HPMPETOHFS, HPLISTFCLEANUP, HPPARSEFEQ.

Command to access SNMPCONTROL in SNMPUDC.NET.SYS.

Reporting Problems

Including all the relevant information when you report a problem with Charon-PAR will facilitate the problem solution.

Please note: log files may contain sensitive information. **It is the responsibility of the user to verify that the information may be shared with Stromasys support according to the regulations of the user's organization.**

To include the most important information, perform the following steps described in the following sections:

- Running charon-report
- Collecting Additional Information
- Providing the Collected Data to Stromasys Support

Running charon-report

The charon-report tool (/opt/charon/bin/charon-report) collects important information about the Charon-PAR installation and the environment in which Charon-PAR runs. For example, it includes the host operating system version, a list of installed packages, the output of **lscpu**, **lshw**, **lspci**, **lsusb**, and **dmesg**, and the networking configuration.

The tool has several parameters to influence its behavior:

Parameter	Description
`-h` `--help`	Displays a short help text and a summary of the available parameters.
`-c` `--crash-report`	Utility runs in non-interactive mode. Reports are stored in **/opt/charon/reports**. Without this parameter, the utility will prompt you for a short description of the problem and store the report in **/tmp**.

`-f PATH [PATH ...]`
`--add-file PATH [PATH ...]`
	File/directory to include in report. Can be used multiple times. May contain wildcards for files /directory selection. If empty, the Charon config and log directories (/opt/charon/cfg, /opt/charon/log) will be included in the report. Important parameter to include the correct directories containing the relevant emulator configuration and log files.
	Please **do not include vdisk or vtape container files** in the report. Upload them separately if advised to do so by Stromasys support.

`-e` `--email`	Send report on email. Default address: support@stromasys.com. It is strongly recommended to review the content of the produced report before sending it via email to make sure the content of the report can be shared externally without violating company regulations.

`-t EMAIL`
`--email-to EMAIL`
	Email address to send report. Can be used multiple times.

Collecting Additional Information

In addition to the data collected by the **charon-report** utility, prepare the following information:

- The Charon-PAR configuration file used and the log files showing the problem.
- A detailed problem description.
- Detailed instructions on how to reproduce the problem.

Providing the Collected Data to Stromasys Support

The data can be provided to Stromasys support via email or via an upload to a location provided by Stromasys support.

Please note: log files may contain sensitive information. **It is the responsibility of the user to verify that the information may be shared with Stromasys support according to the regulations of the user's organization.**

Appendix

Contents

Setting up a Linux Instance in AWS

Contents

This page reflect the AWS GUI changes in spring 2022. If you still use the older GUI, please refer to the Appendix of the Charon-SSP AWS Getting Started guide.

General Prerequisites

As this description shows the basic setup of a Linux instance in AWS, it does not list specific prerequisites. However, depending on the use case, the following prerequisites should be considered:

- **Amazon account and Marketplace subscriptions**:
  - To set up a Linux instance in AWS, you need an Amazon AWS account with administrator access.
  - Identify the AWS region in which you plan to launch your instance. If planning to use an AWS service, use the following link to check if this service is available in the desired region:
    https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services
  - Identify the VPC and subnet in which you plan to launch your instance.
  - If your instance requires Internet access, ensure that the route table associated with your VPC has an Internet Gateway. If your instance requires a VPN access to your on-premises network, ensure that a VPN gateway is available. The exact configuration of your VPC and its subnets will depend on your network design and application requirements.
  - To subscribe to a specific marketplace service select **AWS Marketplace Subscriptions** in the management console and then select **Manage Subscriptions**.
  - Search for the service you plan to use and subscribe to it (accepting the terms and conditions). After a successful subscription, you will find the subscription in the Manage Subscriptions section. From there you can directly launch a new instance.
  - The AWS service providing metered Charon-SSP emulator instance is called *AWS Mainframe Modernization - Virtualization for SPARC*.

- **The instance hardware and software prerequisites will be different depending on the planned use of the instance**:
  - Option 1: the instance is to be used as a **Charon emulator host system**:
    - Refer to the hardware and software prerequisite sections of the *User's Guide* and/or *Getting Started* guide of your Charon product to determine the exact hardware and software prerequisites that must be fulfilled by the Linux instance. The **image** you use to launch your instance and the **i nstance type** you chose determine the software and hardware of your cloud instance.
    - If you use Charon emulator marketplace image, the software prerequisites are already fulfilled.
    - A Charon product **license** is required to run emulated legacy systems. Refer to the licensing information in the documentation of your Charon product, or contact your Stromasys representative or Stromasys VAR for additional information. Emulator marketplace images with Automatic Licensing use public license servers and will create their license automatically at first launch of the instance.
  - Option 2: the instance is to be used as a dedicated **VE license server**:
    - Refer to the *VE License Server Guide* for detailed prerequisites.
- Certain legacy operating systems that can run in the emulated systems provided by Charon emulator products require a license of the original vendor of the operating system. The user is responsible for any licensing obligations related to the legacy operating system and has to provide the appropriate licenses.

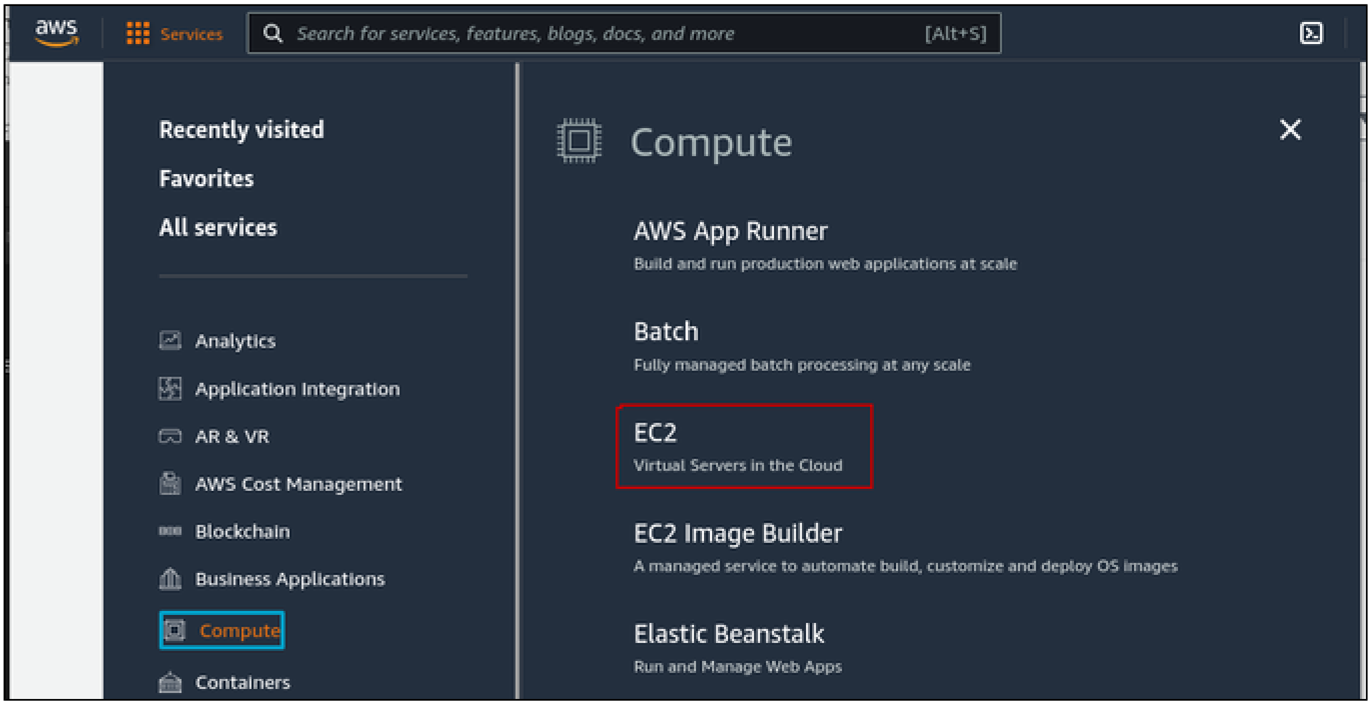
AWS Login and New Instance Launch

Please note that the AWS GUI occasionally changes. This may lead to screenshots not always reflecting the exact appearance of an configuration screen.

To start the creation of a new cloud instance, perform the following steps:

1. **Log in** to your **AWS management console.**

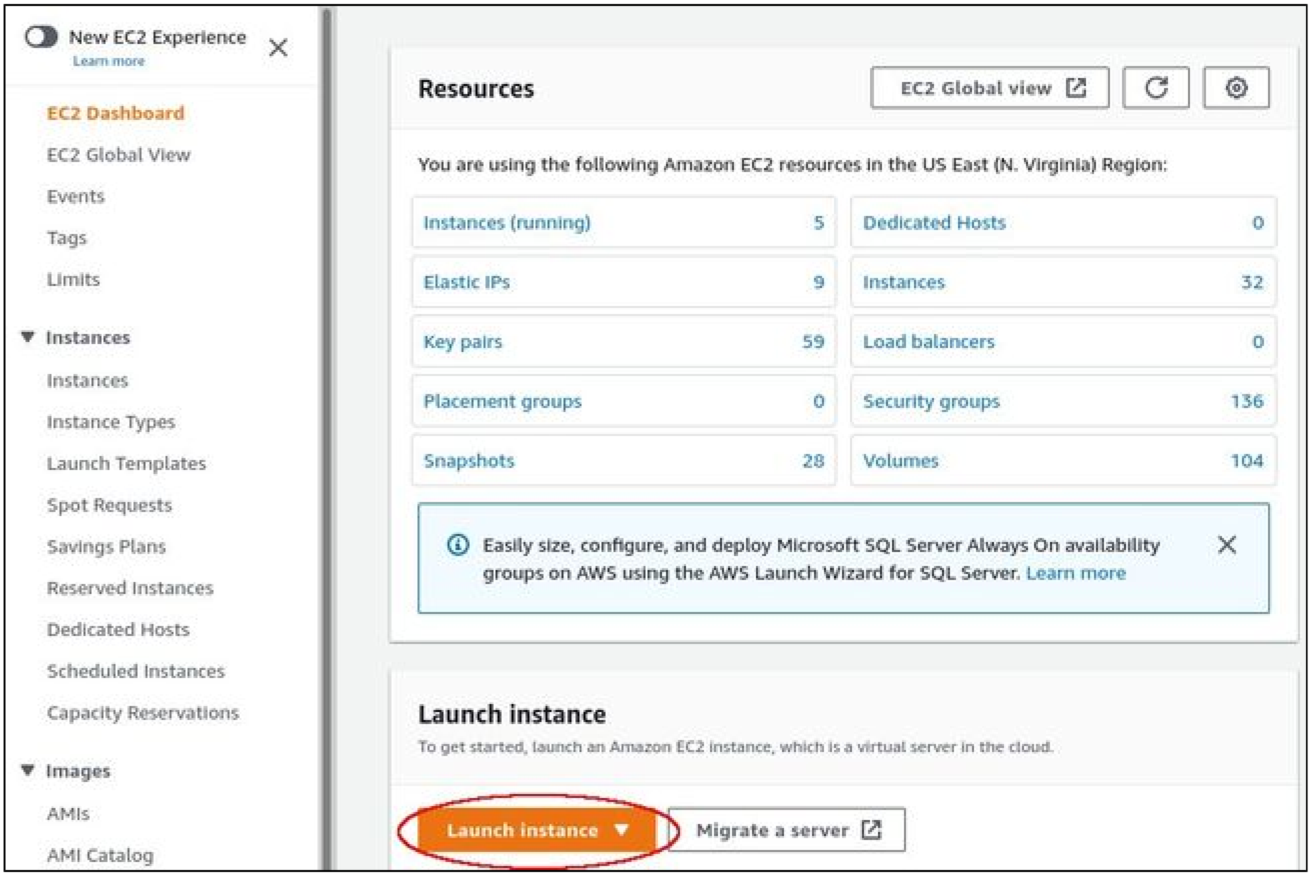
2. Find and select the **EC2 service**. You can find it in the **Recently visited section**, or use the services drop down menu (alternatively, you can also start from your **Manage Subscriptions** page and launch the instance there):



This will open the E2C dashboard.

Please note: The following sample image shows the new E2C dashboard. The old dashboard looks somewhat different, but still has the **Launch instance** button.

3. On the EC2 dashboard click on the **Launch Instance** button.



You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	5	Dedicated Hosts	0
Elastic IPs	9	Instances	32
Key pairs	59	Load balancers	0
Placement groups	0	Security groups	136
Snapshots	28	Volumes	104

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Clicking on **Launch Instance** and selecting the launch instance option will allow you to initiate the instance creation process consisting of seven steps:

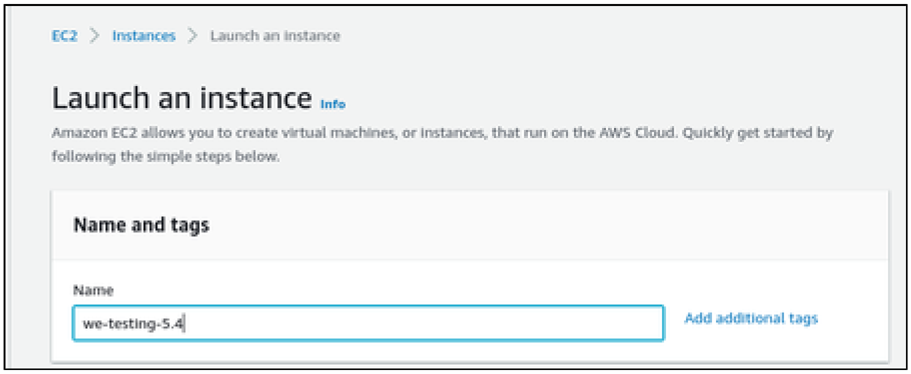
1. Enter an instance name
2. Choose AMI
3. Choose Instance Type
4. Key pair configuration
5. Network and security group configuration
6. Storage configuration
7. Advanced details
8. Launch instance

These steps are described in the next section.

New Instance Configuration

The instance creation and configuration process will guide you through a number of configuration steps and allow you to start the new instance when done.

1. Enter an instance name:



Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: we-testing-5.4

If needed, you can add additional tags to the instance.

When done, **proceed** to the **Application and OS Images** section to choose an AMI (Amazon Machine Image).

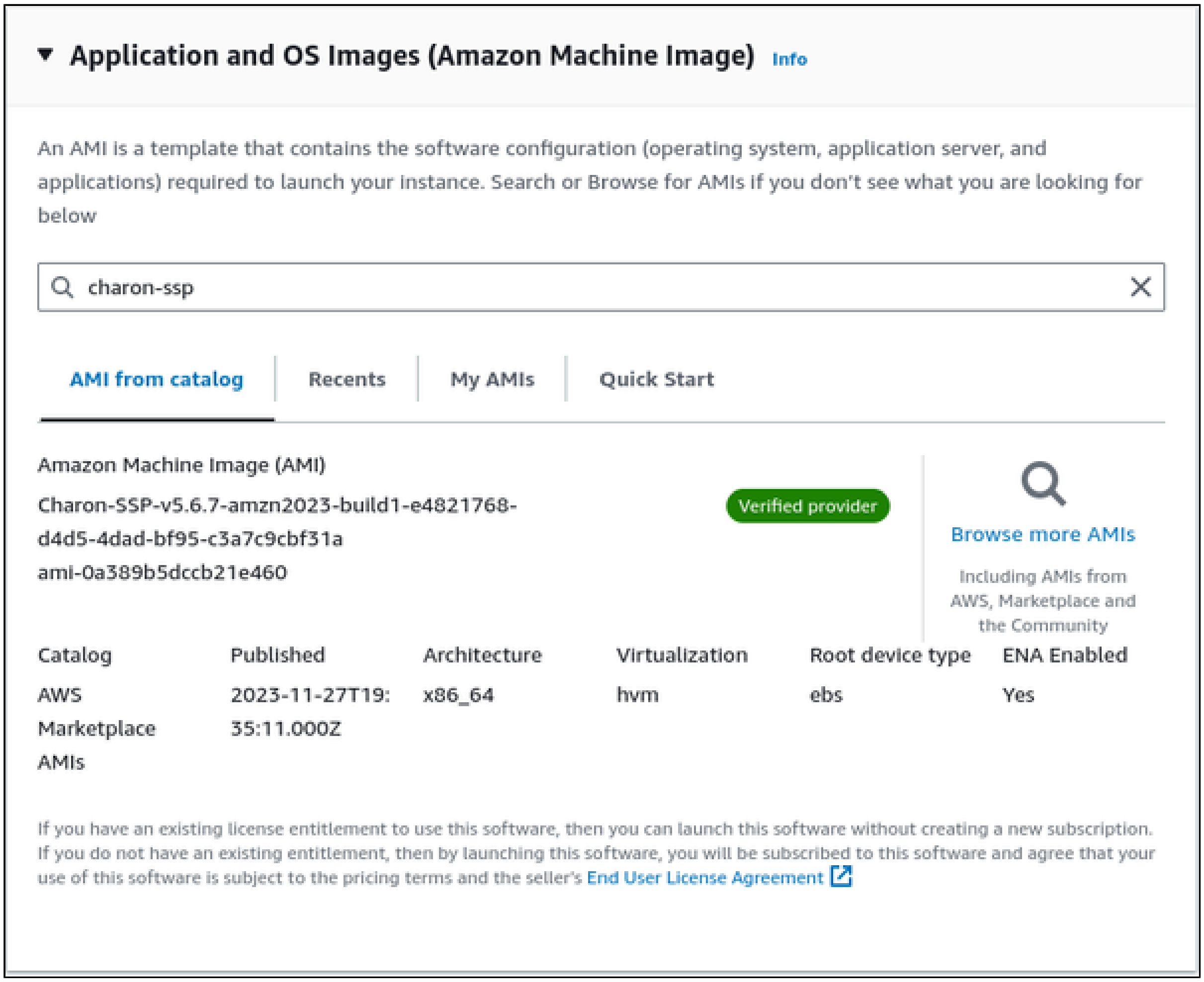
2. Choose AMI:

AMIs are prepackaged images used to launch cloud instances. They usually include the operating system and applicable application software.

Which AMI you select depends on the planned use of the instance:

- If the instance is to be used as a **Charon emulator host system** several AMI choices are possible:
  - **Installing the Charon host system from a prepackaged Charon marketplace image**: they contain the underlying operating system and the preinstalled Charon software.
    - Please check with your Stromasys representative which options are currently available in your cloud providers marketplace.
    - Depending on the cloud provider and the Stromasys product release plans, there may be two variants:
      - *Automatic licensing (AL)* for use with a public, Stromasys-operated license server. Please contact your Stromasys representative if you require a private, customer-operated AutoVE license server
      - *Virtual environment (VE)* for use with a private, customer-operated VE license server
  - **Installing the Charon host system using a conventional Charon emulator installation** with the Charon emulator installation RPM packages for Linux:
    - Choose a Linux AMI of a distribution supported by your selected Charon product and version (see the **user's guide** of your product on the Stromasys documentation site).
- If the instance is to be used as a dedicated **VE license server**:
  - Please refer to the *VE License Server Guide* in Licensing Documentation for the requirements of the Linux instance.

After deciding on which AMI is required, select a matching Linux or Charon product AMI in the Marketplace or (depending on your environment) from My AMIs.



▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

charon-ssp

Amazon Machine Image (AMI)

Charon-SSP-v5.6.7-amzn2023-build1-e4821768-d4d5-4dad-bf95-c3a7c9cbf31a
ami-0a389b5dccb21e460

Catalog	Published	Architecture	Virtualization	Root device type	ENA Enabled
AWS Marketplace AMIs	2023-11-27T19:35:11.000Z	x86_64	hvm	ebs	Yes

If you have an existing license entitlement to use this software, then you can launch this software without creating a new subscription. If you do not have an existing entitlement, then by launching this software, you will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's End User License Agreement

You can use the search field or select one of the categories displayed to start your search. **Select the Linux AMI appropriate to your planned use of the instance**, that is,

- a prepackaged Charon VE marketplace image (as shown in the example above - note the string "**ve**" in the AMI name), or
- a prepackaged Charon AL marketplace image for Automatic Licensing or AutoVE, or
- a Linux version supported for an RPM product installation, or
- a Linux version supported for the VE license server.

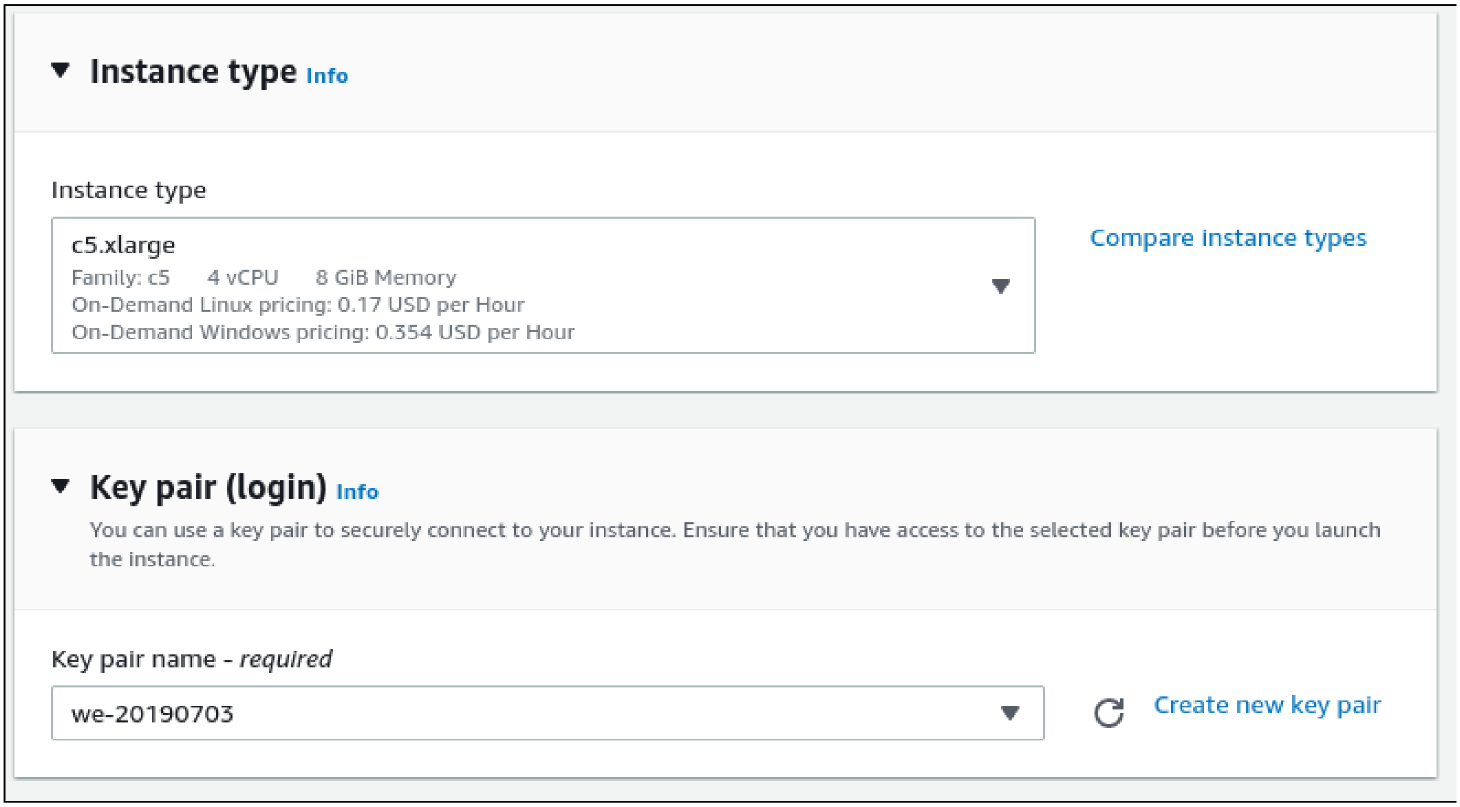
Then **proceed** to the next section, the **Instance type** selection.

3. Choose Instance Type:

Amazon EC2 offers instance types with varying combinations of CPU, memory, storage, and networking capacity.

Select an instance type that matches the requirements of the Charon product to be used. Please note that some marketplace images have a restricted selection of instance types.

When done, **proceed** to the **Key pair** configuration.



▼ Instance type Info

Instance type

c5.xlarge
Family: c5 4 vCPU 8 GiB Memory
On-Demand Linux pricing: 0.17 USD per Hour
On-Demand Windows pricing: 0.354 USD per Hour

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

we-20190703

4. SSH key pair configuration:

In this section, you can

- either **create a new SSH key pair** and download the private key, or
- you can **select an existing key pair** to use for logging in to the new instance. If you select an existing key pair, make sure you have the matching private key. Otherwise, you will not be able to log in.

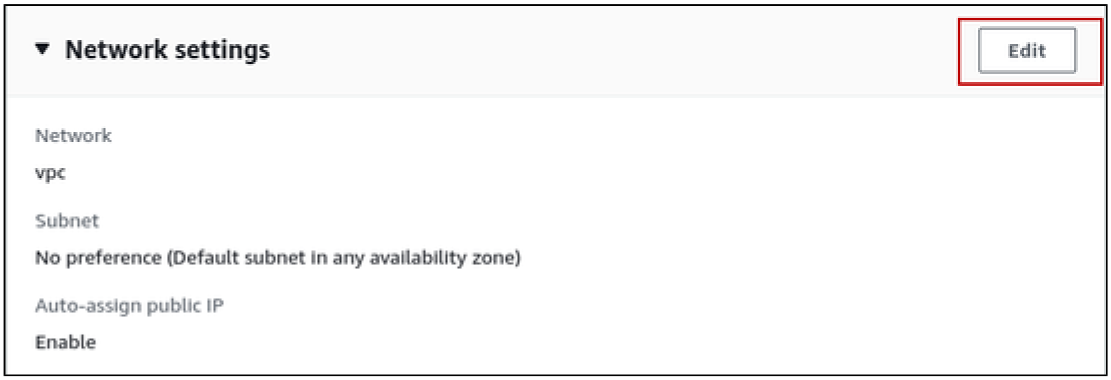
Please note: if your management system supports it, for RHEL 9.x, Rocky Linux 9.x, and Oracle Linux 9.x use SSH key types ECDSA or ED25519. This will allow connecting to these Charon host Linux systems using an SSH tunnel without the default crypto-policy settings on the Charon host having to be changed for less secure settings. This is, for example, important for the Charon-SSP Manager. See also: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/security_hardening/using-the-system-wide-cryptographic-policies_security-hardening.

After configuring your key pair, **proceed** to the **Network settings** section.

5. Network configuration:

This section offers basic default settings to connect your instance to the network. However, in most cases, you will have to adjust the settings to your environment.

To do this, click on the **Edit** button at the top of the section:



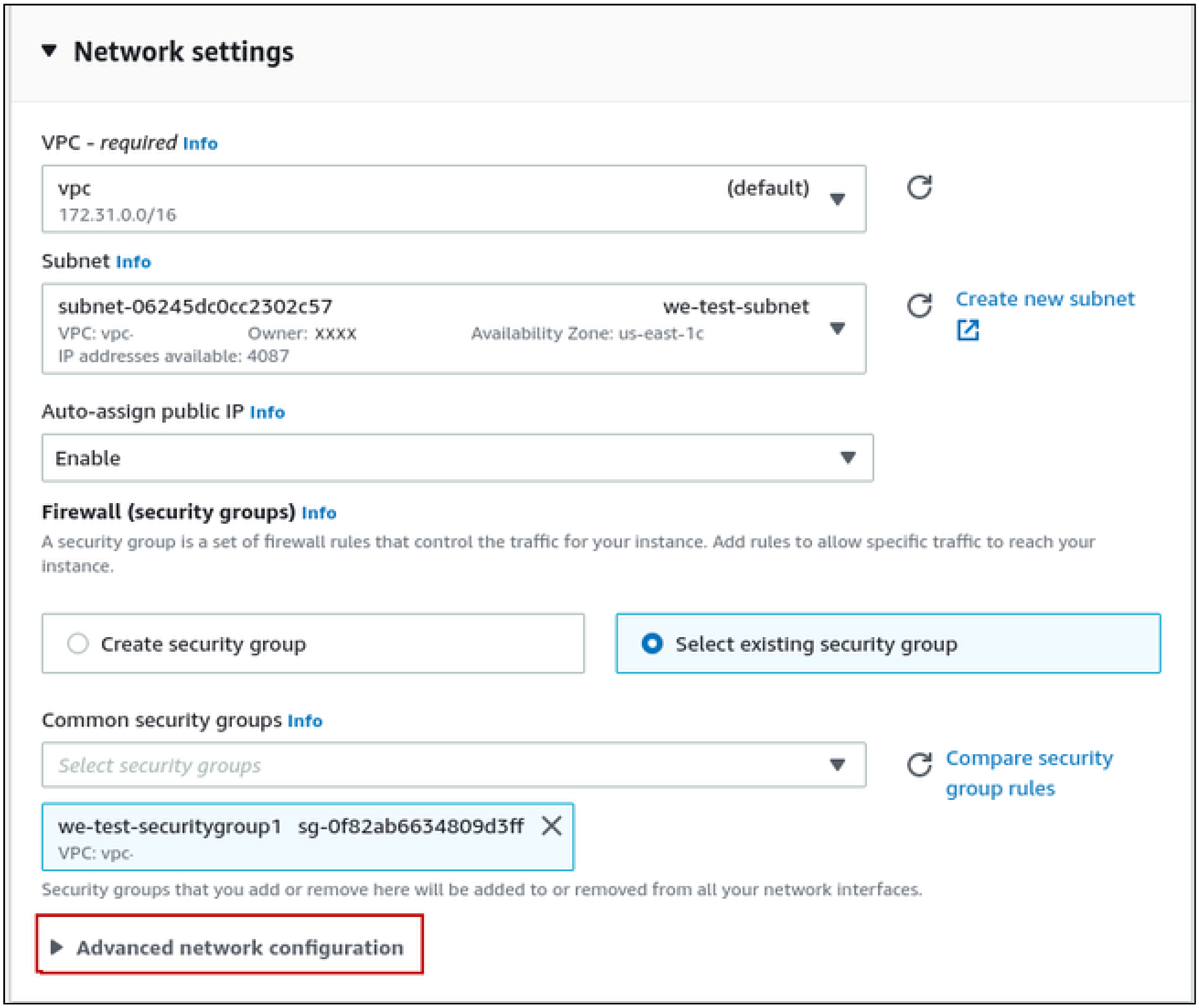
Network
vpc

Subnet
No preference (Default subnet in any availability zone)

Auto-assign public IP
Enable

This will open the edit window and allow additional settings:

- The VPC (if a non-default VPC is to be used)
- The desired subnet (either an existing one or a newly created subnet)
- Enable or disable the automatic assignment of a public IP address to the primary interface (**automatic assignment is only possible if a single network interface is selected for the instance**)
- Assign an existing or new custom security group (cloud-provided firewall). The security group must allow at least SSH to access the instance. Any ports required by applications planned to run on the instance must also be allowed (the security group can be modified at any time after the instance has been created).


▼ Network settings
Info
Create new subnet
Compare security group rules
▶ Advanced network configuration

VPC - *required*
vpc (default)
172.31.0.0/16

Subnet
subnet-06245dc0cc2302c57 we-test-subnet
VPC: vpc- Owner: xxxx Availability Zone: us-east-1c
IP addresses available: 4087

Auto-assign public IP
Enable

Firewall (security groups)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group ● Select existing security group

Common security groups
Select security groups

we-test-securitygroup1 sg-0f82ab6634809d3ff
VPC: vpc-

Security groups that you add or remove here will be added to or removed from all your network interfaces.

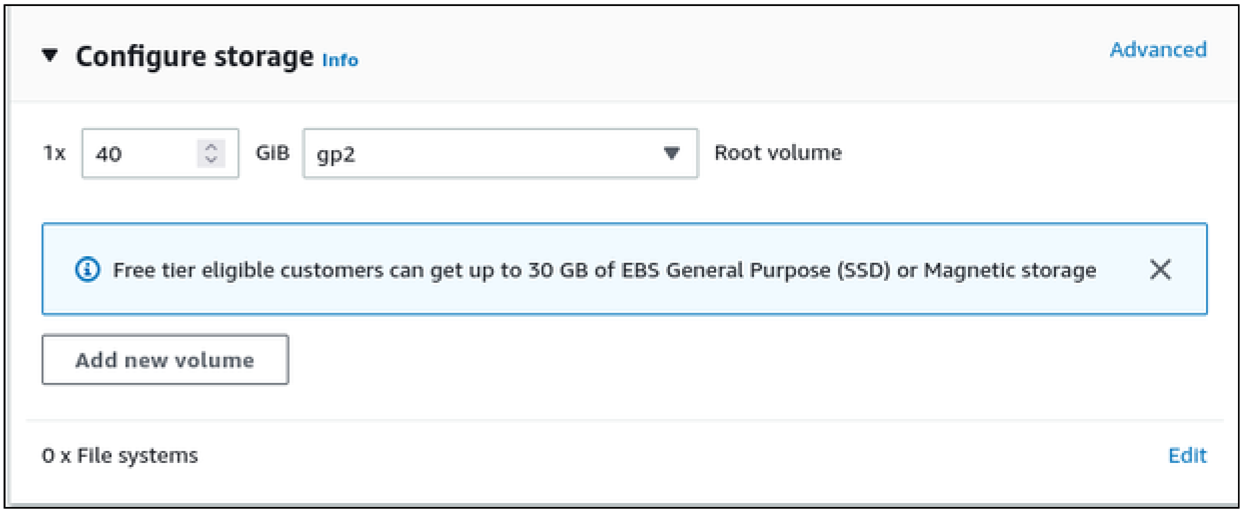
The **Advanced network configuration** option at the bottom of the section opens an additional configuration section in which you can set more advanced interface options and add additional network interfaces (automatic assignment of a public IP address only works if there is **only one network interface** attached to the instance). Additional interfaces can also be added to the instance after it has been first launched.

Once you are done with the network configuration, **proceed** to the **Configure storage** section.

6. Storage configuration:

The size of the root volume (the system disk) must be appropriate for your environment (recommended minimum system disk size for the Linux system: 30GB). You can add more storage now or later to provide space for virtual disk containers and other storage requirements, but the system disk size should cover the Linux system requirements including any applications/utilities planned to be installed on it.

Please note: It is recommended to **create separate storage volumes for Charon application data** (e.g., disk images). If required, such volumes can later easily be migrated to another instance.


▼ Configure storage Info

1x 40 GiB gp2 Root volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

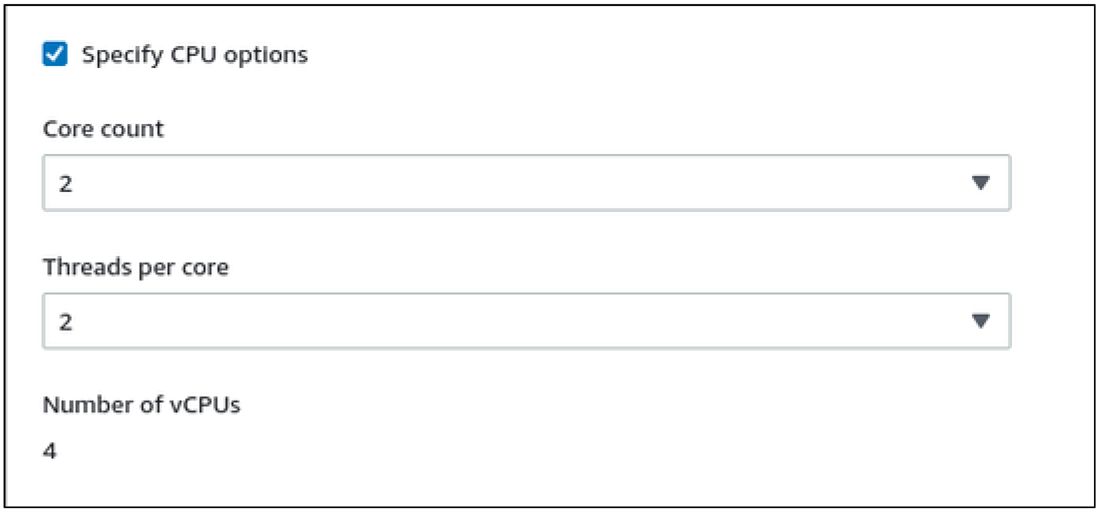
0 x File systems

If needed, open the **Advanced details** section to access additional settings.

7. Advanced details:

In this section, you can set many parameters. Three that are more likely to be useful to a Charon emulator environment are shown here as examples:

CPU characteristics (enable or disable more than one thread per CPU core, options depend on the selected instance type). **This can only be set at instance launch. It cannot be changed later**.


☑ Specify CPU options

Core count: 2

Threads per core: 2

Number of vCPUs: 4

IAM role

Only for a VE license server system with a **version earlier than 1.1.23**, you must assign the required IAM role (allowing the ***ListUsers*** action) to the instance. For more information see the Virtual Environment (VE) License Server Documentation.

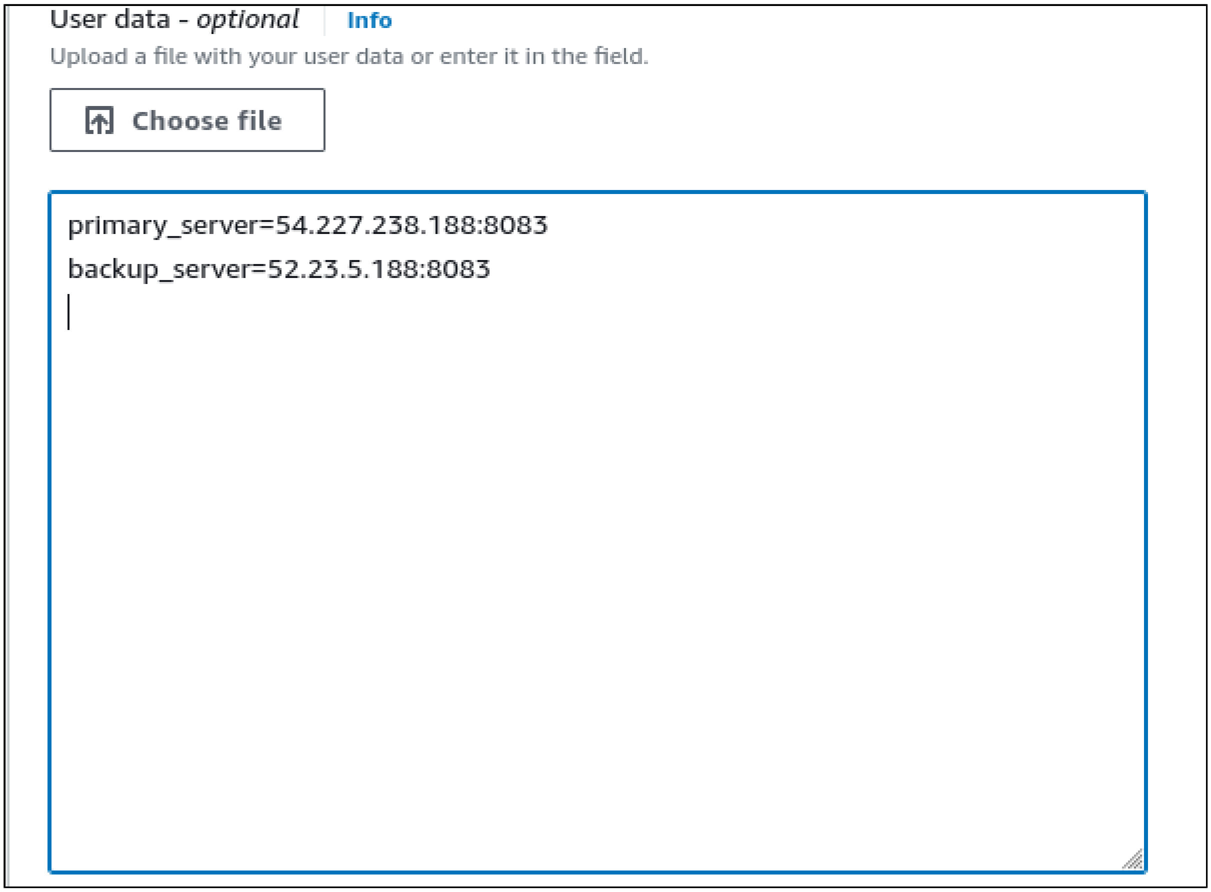
User data

If your instance is based on a **Charon AL marketplace image** and planned to be used for **AutoVE licensing** (instead of the Stromasys-operated public license servers) or based on the **Charon-SSP Amazon Linux image**, you must add the corresponding information to the instance configuration **before** the first launch of the instance.

Please note:

- Should you use the SSP Amazon Linux AMI with **SSP version 5.6.8 or higher** as provided by the *AWS Mainframe Modernization - Virtualization for SPARC* service, the instance will by **default** connect to the **public, Stromasys-operated AutoVE license servers** (defined in */opt/charon-license-server*). You only need the **user data definition for older versions or to override the default** with your private AutoVE servers.
- The example below shows the appearance of the AutoVE license server information that is entered as **User Data** in the **Advanced Details** configuration section at the bottom of the **Launch an Instance** window during the initial configuration of an instance. **Scroll down** to the bottom of the configuration window to open and display the user data section in the **Advanced Details**.
- In the older GUI version, the **Advanced Details** section is part of the **Configure Instance** window - the layout is somewhat different, but the configuration options are the same.

Enter the information for the AutoVE license server as shown in the example below (it shows the public AutoVE servers):



```
primary_server=54.227.238.188:8083
backup_server=52.23.5.188:8083
```

Valid User Data configuration options:

- `primary_server=<ip-address>[:<port>]`
- `backup_server=<ip-address>[:<port>]`

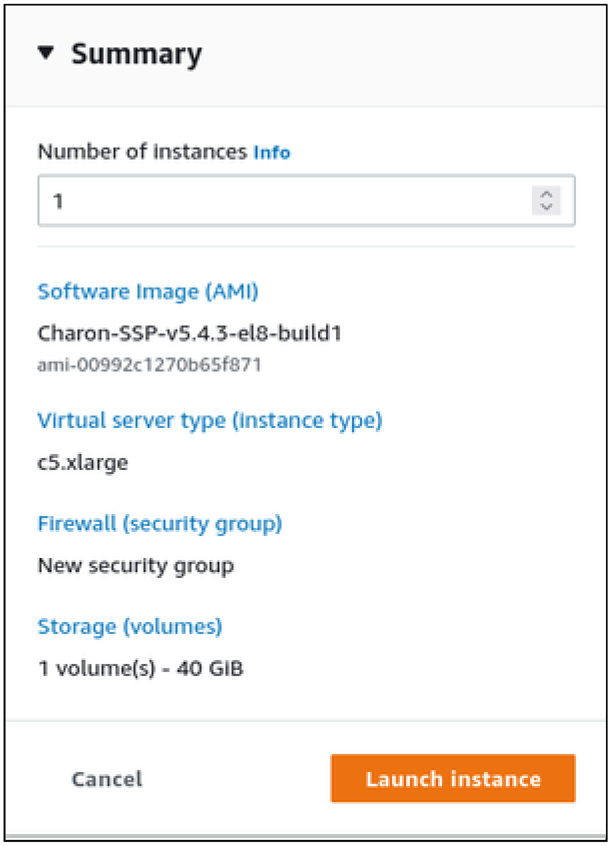
where

- *<ip-address>* stands for the IP address of the primary and the backup server as applicable, and
- *<port>* stands for a non-default TCP port used to communicate with the license server (default: TCP/8083).

Please note: at least one license server must be configured at initial launch to enable AutoVE mode. This can be via the */opt/charon-license-server* file with the default public servers (SSP 5.6.8 or higher) or via the manual user data configuration. **Otherwise, the instance will bind to one of the public AL license servers operated by Stromasys.**

8. Launch your instance:

Click on Launch instance in the right-hand pane to launch your instance (if the launch button is not visible, you may have to close overlaying text panes first):



▼ Summary

Number of instances Info

1

Software Image (AMI)

Charon-SSP-v5.4.3-el8-build1

ami-00992c1270b65f871

Virtual server type (instance type)

c5.xlarge

Firewall (security group)

New security group

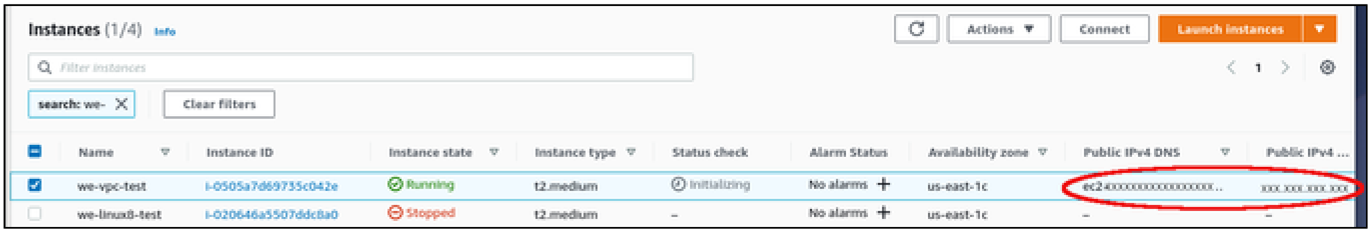
Storage (volumes)

1 volume(s) - 40 GiB

Verify that instance is running:

After starting your instance for the first time, you will see it in the initializing state in the list of your AWS instances. It will take a bit of time to get to the running state.You will eventually see a launch success message with a link to your instance. Clicking on this link will take you to your new instance in the instance overview list.

In addition to the instance state, important information, for example, the public IP address and public DNS name (marked in red) of the instance will also be displayed. The following image shows an example:



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Availability zone	Public IPv4 DNS	Public IPv4 ...
☑	we-vpc-test	i-0505a7d69735c042e	Running	t2.medium	Initializing	No alarms +	us-east-1c	ec2-xxxxxxxxxxxxxxxxx...	xxx.xxx.xxx.xxx
☐	we-linux8-test	i-020646a5507ddc8a0	Stopped	t2.medium	–	No alarms +	us-east-1c	–	–

The following sections will show you how to access the instance and how to perform additional storage and network configurations.

Please note:

- If you select your instance, the bottom of the screen will show a detailed description and status information of your instance.
- You can rename your instance after creating it by clicking on the pencil symbol that will appear next to the instance name when placing the mouse pointer over it, or by editing the name tag of the instance.

Initial Access to the Instance

Once you have access to the instance, you can create the access you require for your applications. This section just shows the basic steps for initial access to the instance.

SSH Interactive Access

To connect to the instance interactively, you must connect as the management user of your instance. Use the following command:

```
$ ssh -o ServerAliveInterval=30 -i <path-to-your-private-key> <management-user-name>@<cloudhost-IP-address>
```

The parameter `ServerAliveInterval` will protect the connection from timing out.

Please note:

- Depending on the type of connection, you will have to use either the public IP address of the cloud system or its address in a customer-specific VPN.
- The **private key** used must correspond to the public key installed in the *authorized_keys* file of the cloud instance management user. This is usually done during initial cloud instance launch.
- The management user account normally allows sudo access to privileged commands (use **sudo -i**).
- If the instance was created using a Stromasys-provided AL or VE marketplace image, the management user for **interactive login** is the user **sshuser**.

File Transfer with SFTP

SFTP enables file transfers to and from the cloud instance. Use the management user of your instance. The security rules must allow SSH access to allow SFTP access to the cloud instance.

Please note: Depending on the type of connection, you will have to use either the public IP address of the cloud system or its address in a customer-specific VPN.

To connect to the instance, use the following command:

```
$ sftp -i <path-to-your-private-key> <management-user-name>@<cloudhost-IP-address>
```

Please note:

- Depending on the type of connection, you will have to use either the public IP address of the cloud system or its address in a customer-specific VPN.
- The **private key** used must correspond to the public key installed in the *authorized_keys* file of the cloud instance management user. This is usually done during initial cloud instance launch.
- If the instance was created using a Stromasys-provided AL or VE marketplace image, the management user for **file transfer** is the user **charon**.
- If the user **charon** is used to transfer files, the home directory for the file transfer will be */charon/storage.*

Setting up a Linux Instance in Azure

Contents

General Prerequisites

As this description shows the basic setup of a Linux instance in Azure, it does not list specific prerequisites. However, depending on the use case, the following prerequisites should be considered:

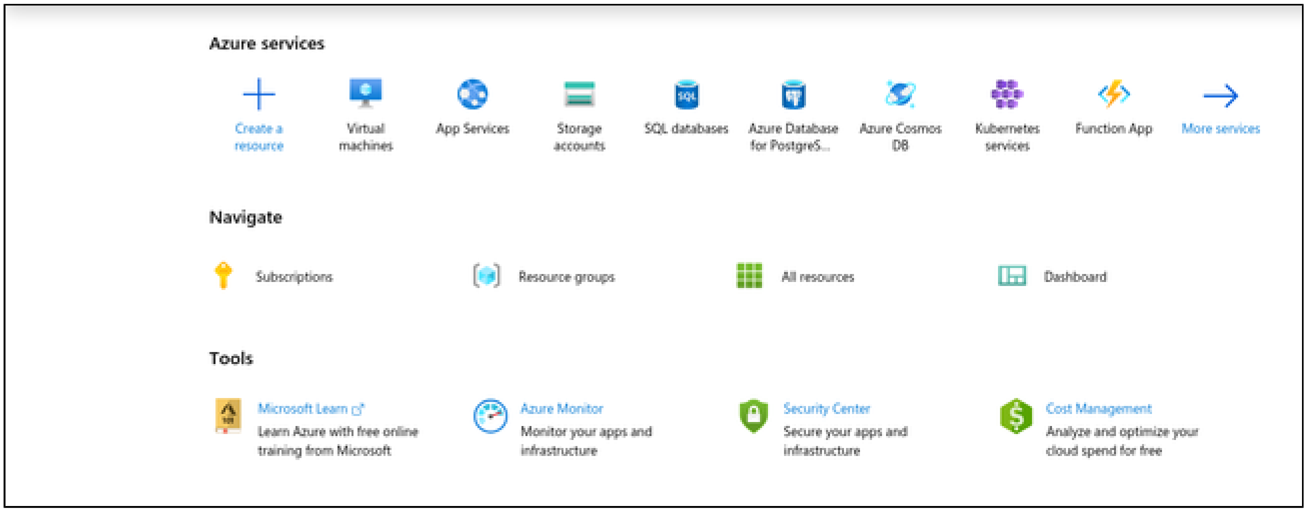
- To set up a Linux instance in Azure, you need an Azure account.
- Secondly, **prerequisites will be different depending on the planned use of the instance**:
  - Option 1: the instance is to be used as a **Charon emulator host system**:
    - Refer to the hardware and software prerequisite sections of the *User's Guide* and/or *Getting Started guide* of your Charon product to determine the exact hardware and software prerequisites that must be fulfilled by the Linux instance.
      The **image** you use to launch your instance and the **instance type** you chose determine the software and hardware of your cloud instance.
    - A Charon product **license** is required to run emulated legacy systems. Contact your Stromasys representative or Stromasys VAR for details.
  - Option 2: the instance is to be used as a dedicated **VE license server**:
    - Refer to the VE License Server Guide for detailed prerequisites.
- Certain legacy operating systems that can run in the emulated systems provided by Charon emulator products require a license of the original vendor of the operating system. The user is responsible for any licensing obligations related to the legacy operating system and has to provide the appropriate licenses.

Azure Login and New Instance Launch

Logging in to your Azure account

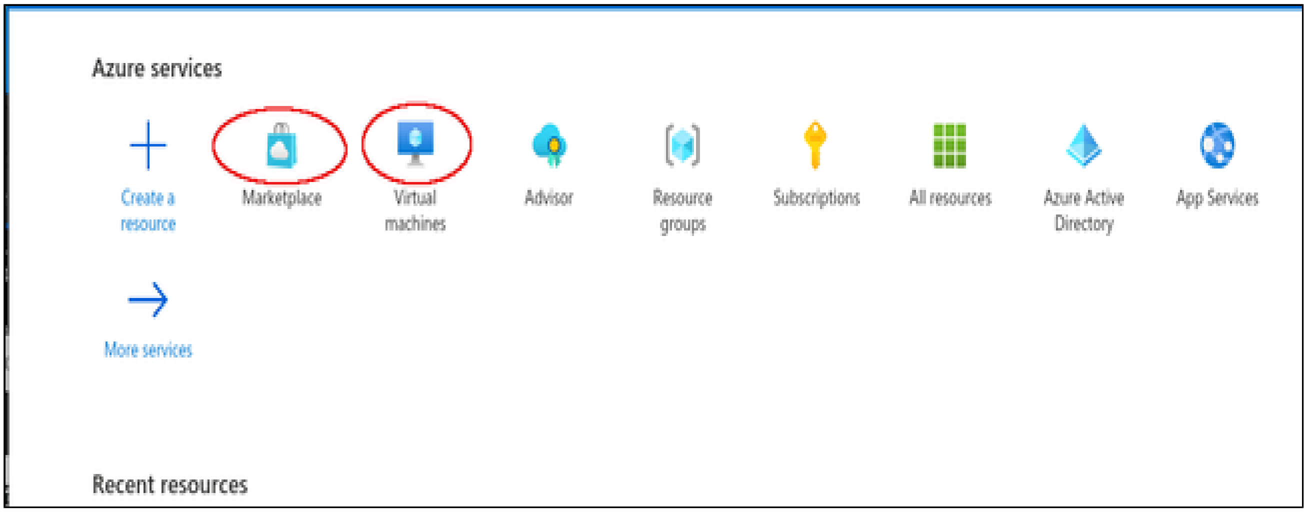
To log in perform the following steps:

- Go to **portal.azure.com**. You will see a Microsoft Azure login screen.
- Enter your login credentials.
- Upon successful login, the Azure home screen will be displayed as shown in the example below:



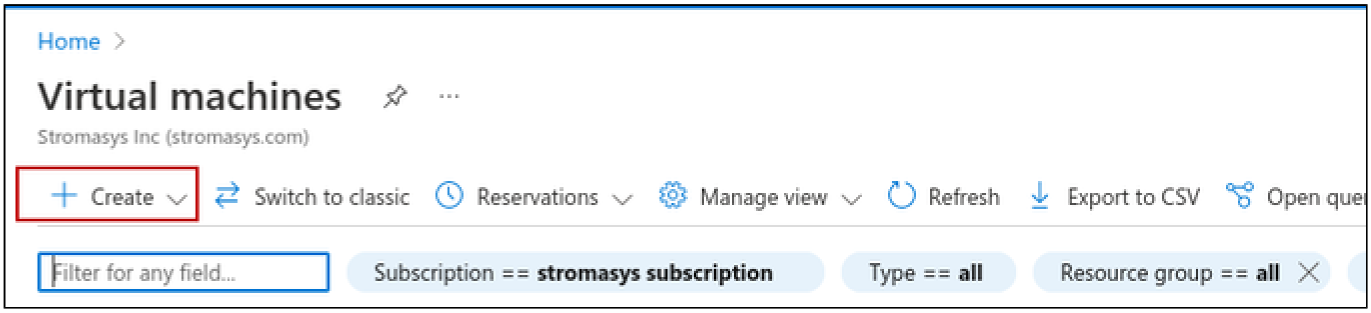
Creating a Virtual Machine

Step 1: Click on the **Virtual machines** or on the **Marketplace** icon on the home page. If you create your instance via the Marketplace icon, please select the Charon listing from the Marketplace offerings, select to create an instance, and continue with Step 3.



Clicking on **Virtual machines** opens the virtual machines overview list.

Step 2: Click on the **Create** link in the overview list.



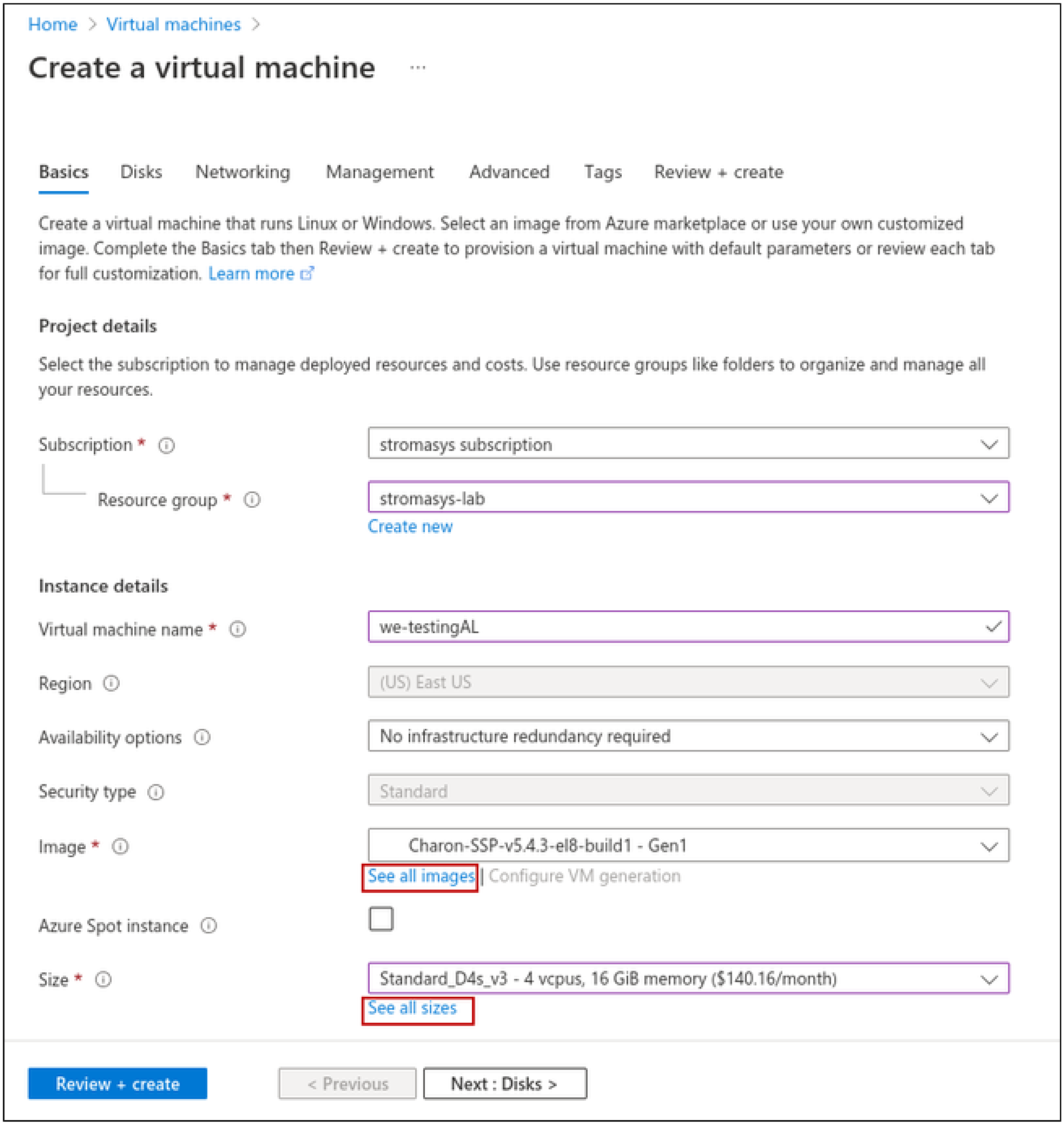
For a basic setup, select **Azure virtual machine** from the drop down list opened by the **Create** link. This opens the **Basics** tab of the **Create a Virtual Machine** window.

Step 3: Enter your data on the **Basics** tab. Mandatory data are, for example:

- Your subscription
- Existing resource group (or click on **Create new**)
- Virtual machine name (cannot be changed after launching the instance)
- Region for the virtual machine
- The Azure image from which to launch your instance. Click on **See all images** to select the correct image. If installing a prepackaged marketplace Charon image, select the matching image. If you plan to install Charon using RPM packages, use a Linux version supported by your Charon emulator product.
- Size of your VM (click on **See all sizes** to see a list of available sizes)

Basics tab **upper part** sample:

Select the image from which to launch your instance and the correct size of your instance (please review the sizing requirements above). Enter the other information in accordance to your environment.



Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	stromasys subscription
Resource group *	stromasys-lab Create new

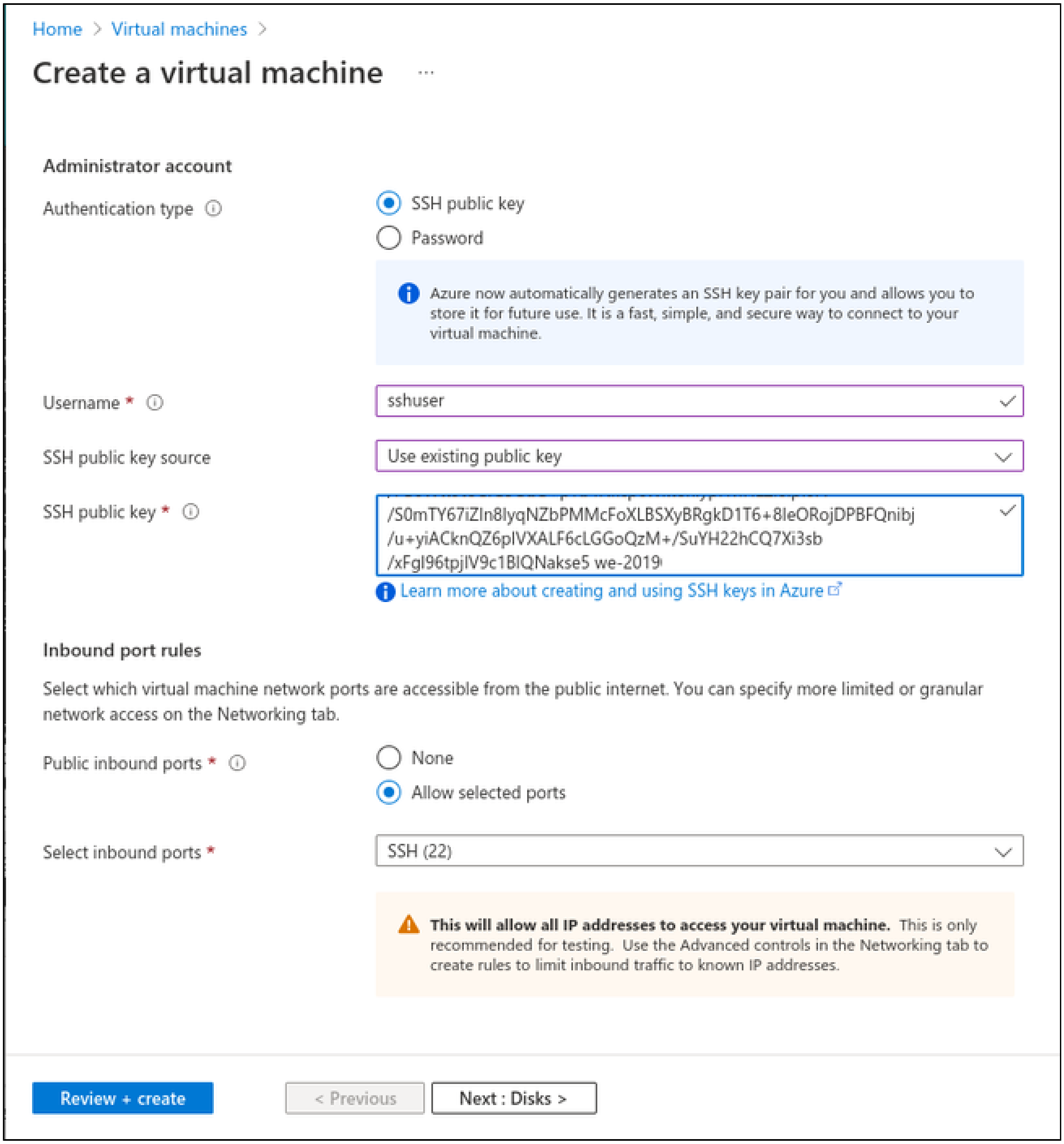
Instance details

Virtual machine name *	we-testingAL
Region	(US) East US
Availability options	No infrastructure redundancy required
Security type	Standard
Image *	Charon-SSP-v5.4.3-el8-build1 - Gen1 See all images | Configure VM generation
Azure Spot instance	□
Size *	Standard_D4s_v3 - 4 vcpus, 16 GiB memory ($140.16/month) See all sizes

Basics tab **lower part** sample:

- Enter the user **sshuser** as the administrative user.
- Select **SSH public key** authentication. You can then use **one of the following** steps to install your SSH public key.
  - Let Azure create a new key-pair for you.
  - Use the public key from a key-pair on your computer. As shown in the example below, you will have to past your public key into the field provided.
  - Use a key-pair previously created on Azure.
- The default allowed inbound port will allow SSH connections without limiting the source IP range. Some images may also have preconfigured access rules that cannot be changed during the launch of the instance. In either case, remember to adapt the rules to your requirements after creating the instance or in the Networking tab (advanced) during the creation of the instance.

Please note: if your management system supports it, for RHEL 9.x, Rocky Linux 9.x, and Oracle Linux 9.x use SSH key types ECDSA or ED25519. This will allow connecting to these Charon host Linux systems using an SSH tunnel without the default crypto-policy settings on the Charon host having to be changed for less secure settings. This is, for example, important for the Charon-SSP Manager. See also: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/security_hardening/using-the-system-wide-cryptographic-policies_security-hardening.



Create a virtual machine ···

Administrator account

Authentication type ⓘ ● SSH public key ○ Password

Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Username * ⓘ sshuser ✓

SSH public key source: Use existing public key

SSH public key * ⓘ
/S0mTY67iZIn8lyqNZbPMMcFoXLBSXyBRgkD1T6+8leORojDPBFQnibj
/u+yiACknQZ6pIVXALF6cLGGoQzM+/SuYH22hCQ7Xi3sb
/xFgl96tpjlV9c1BIQNakse5 we-2019

Learn more about creating and using SSH keys in Azure

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * ⓘ ○ None ● Allow selected ports

Select inbound ports * SSH (22)

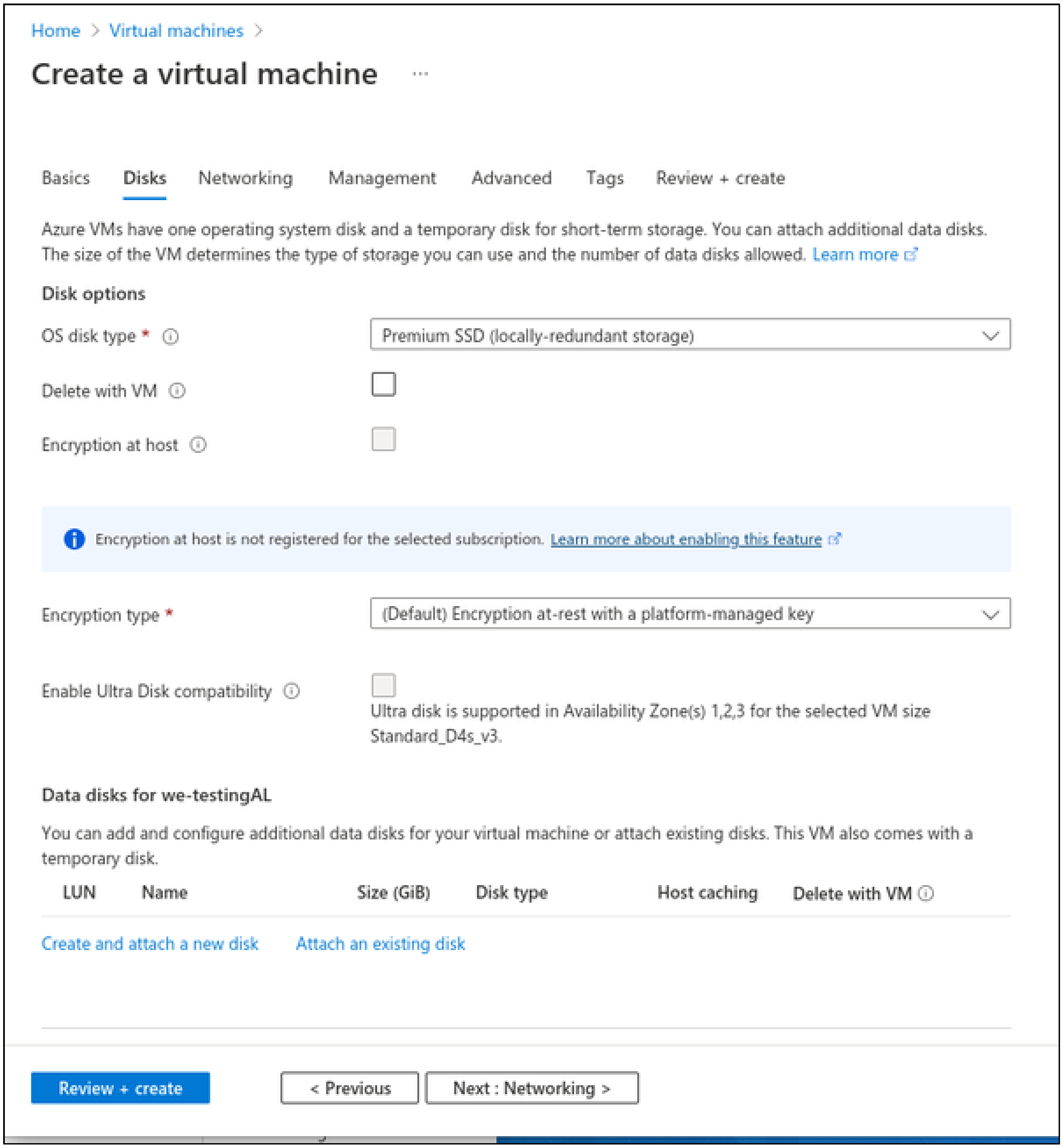
This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Click on **Next: Disks**. This will open the **Disks** tab of the VM creation window.

Step 4: Define the disks for your VM.

Please note: By default, Azure VMs have one operating system disk and a temporary disk for short-term storage (mounted on /mnt/resource and not persistent). The recommended minimum system disk size is 30GB. You can attach existing additional data disks, or create new disks and attach them.

Disks tab sample:



Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. Learn more

Disk options

OS disk type * : Premium SSD (locally-redundant storage)

Delete with VM □

Encryption at host □

Encryption at host is not registered for the selected subscription. Learn more about enabling this feature

Encryption type * : (Default) Encryption at-rest with a platform-managed key

Enable Ultra Disk compatibility □
Ultra disk is supported in Availability Zone(s) 1,2,3 for the selected VM size Standard_D4s_v3.

Data disks for we-testingAL

You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.

LUN	Name	Size (GiB)	Disk type	Host caching	Delete with VM

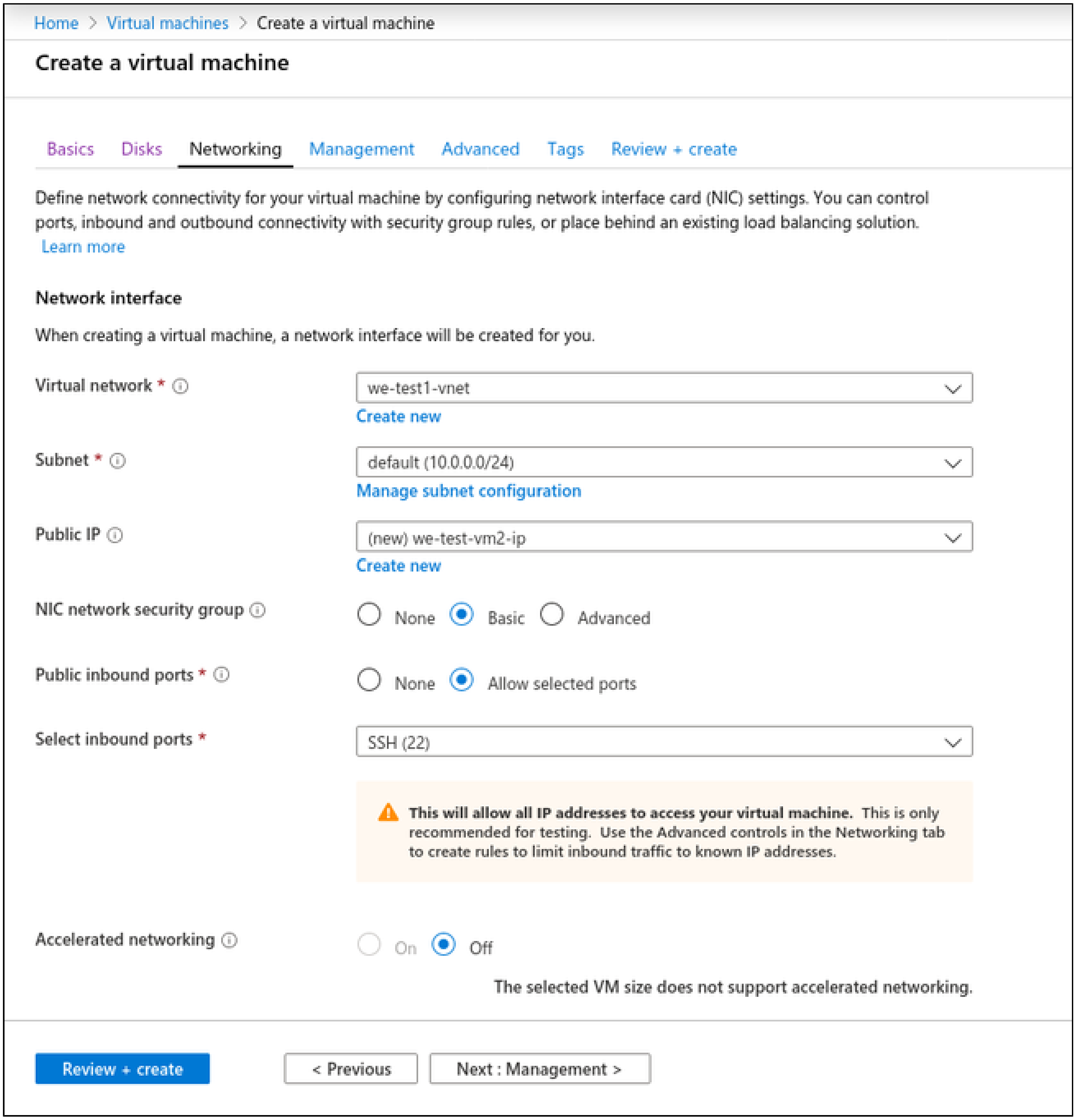
Click on **Next: Networking**. This will open the **Networking** tab of the VM creation window.

Step 5: Enter the necessary information in the **Networking** tab.

On this tab, you can define the network configuration of your VM:

- Virtual Network (existing or new)
- Subnet (default or other subnet)
- Whether a public IP should be assigned or not (note that if you use an image requiring a public, Stromasys-operated license server, this server must be accessed via a public IP address from the Azure range)
- Basic, advanced, or preconfigured security settings (which ports are open for access to the VM).

Networking tab sample:



Create a virtual machine

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution.

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network * : we-test1-vnet

Subnet * : default (10.0.0.0/24)

Public IP : (new) we-test-vm2-ip

NIC network security group : ○ None ● Basic ○ Advanced

Public inbound ports * : ○ None ● Allow selected ports

Select inbound ports * : SSH (22)

⚠ **This will allow all IP addresses to access your virtual machine.** This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Accelerated networking : ○ On ● Off

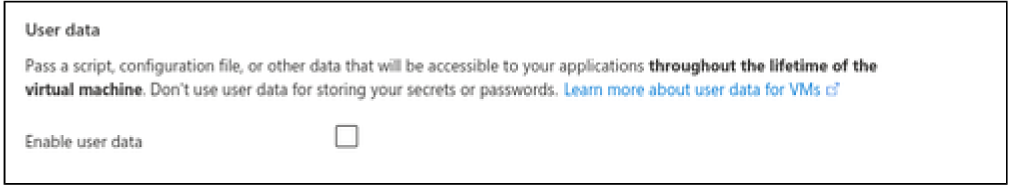
The selected VM size does not support accelerated networking.

Optionally, you can proceed to the **Management**, **Advanced**, and **Tags** tabs to configure additional details of your VM. However, for a basic test, this is not required. Click on **Review + Create** to proceed to the review screen.

Step 6: additional configuration for **AutoVE** setup.

If the instance is launched from a Charon AL marketplace image and is planned to use AutoVE licensing (instead of the public license servers), you must add the corresponding information to the instance configuration **before** the first launch of the instance:
The AutoVE license server information is entered as instance **User Data**. In the initial instance configuration window, go to the **Advanced** section.

- Open it and scroll down to the **User Data** section.
- Enable the **User Data** by ticking the checkbox.

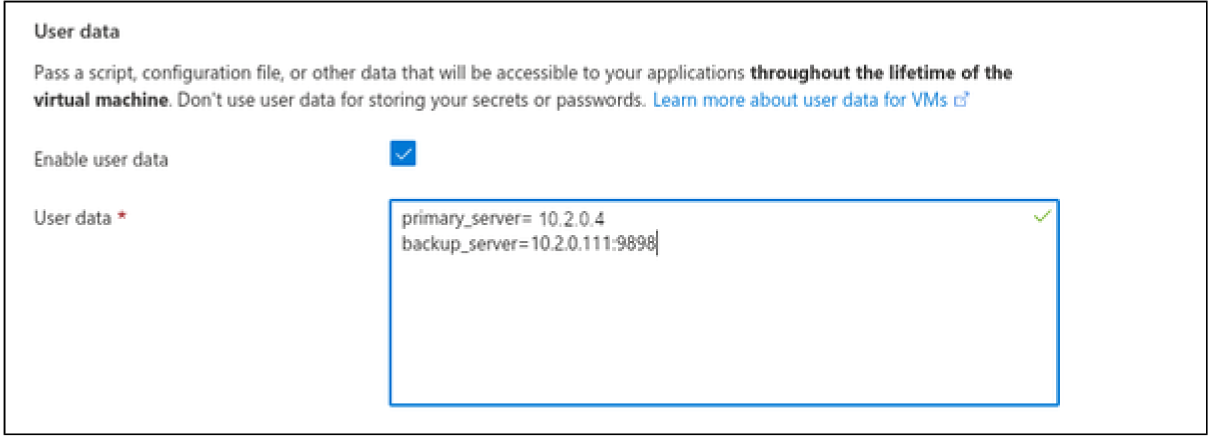


User data

Pass a script, configuration file, or other data that will be accessible to your applications **throughout the lifetime of the virtual machine**. Don't use user data for storing your secrets or passwords. Learn more about user data for VMs

Enable user data □

- Then enter the correct primary and backup AutoVE servers as shown in the example below:



User data

Pass a script, configuration file, or other data that will be accessible to your applications **throughout the lifetime of the virtual machine**. Don't use user data for storing your secrets or passwords. Learn more about user data for VMs

Enable user data ☑

User data *

```
primary_server= 10.2.0.4
backup_server=10.2.0.111:9898
```

Valid User Data configuration options:

- **`primary_server=`**`<`*`ip-address`*`>[:<`*`port`*`>]`
- **`backup_server=`**`<`*`ip-address`*`>[:<`*`port`*`>]`

where

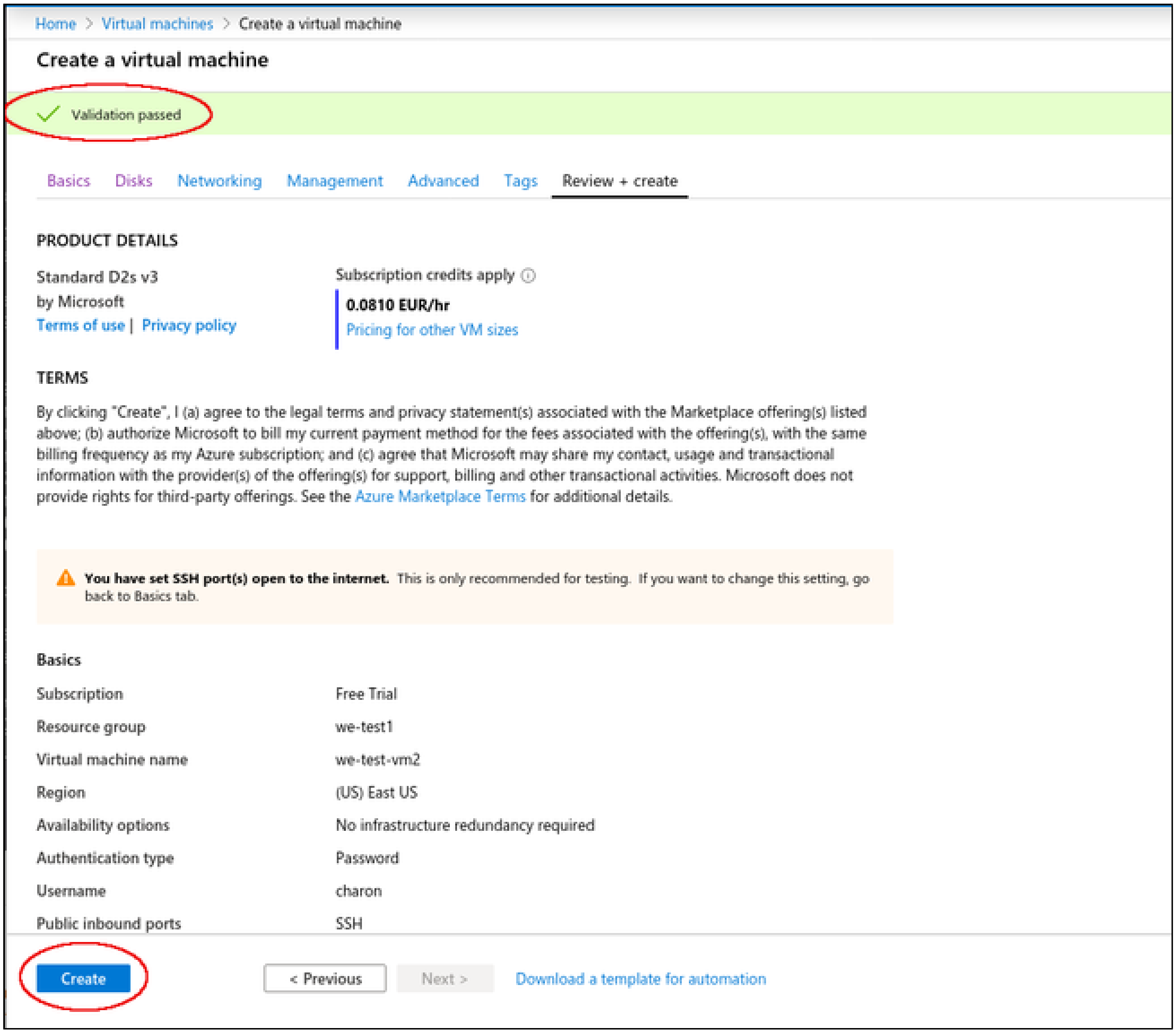
- *<ip-address>* stands for the IP address of the primary and the backup server as applicable, and
- *<port>* stands for a non-default TCP port used to communicate with the license server (default: TCP/8083).

Please note: at least one license server must be configured at initial launch to enable AutoVE mode. **Otherwise, the instance will bind to one of the public license servers operated by Stromasys.**

Step 7: Check the data on the Review + Create screen and create VM.

Verify that the checks passed successfully and click on **Create** to create the VM.

Sample **Review+Create** screen:



Create a virtual machine

✓ Validation passed

PRODUCT DETAILS

Standard D2s v3
by Microsoft
Terms of use | Privacy policy

Subscription credits apply ⓘ
0.0810 EUR/hr
Pricing for other VM sizes

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the Azure Marketplace Terms for additional details.

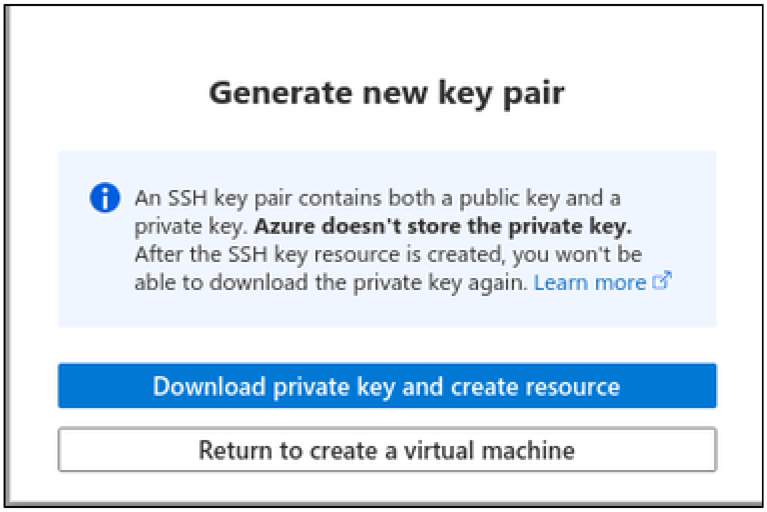
⚠ **You have set SSH port(s) open to the internet.** This is only recommended for testing. If you want to change this setting, go back to Basics tab.

Basics

Subscription	Free Trial
Resource group	we-test1
Virtual machine name	we-test-vm2
Region	(US) East US
Availability options	No infrastructure redundancy required
Authentication type	Password
Username	charon
Public inbound ports	SSH

If key-pair was newly created, download private key:

If you chose to let Azure create a new SSH key-pair, you will be asked to download the private key after clicking on the Create **button**, this step is very important as this is the only opportunity to download the private key, which is required to access your VM. The image below shows a sample of this prompt:



Generate new key pair

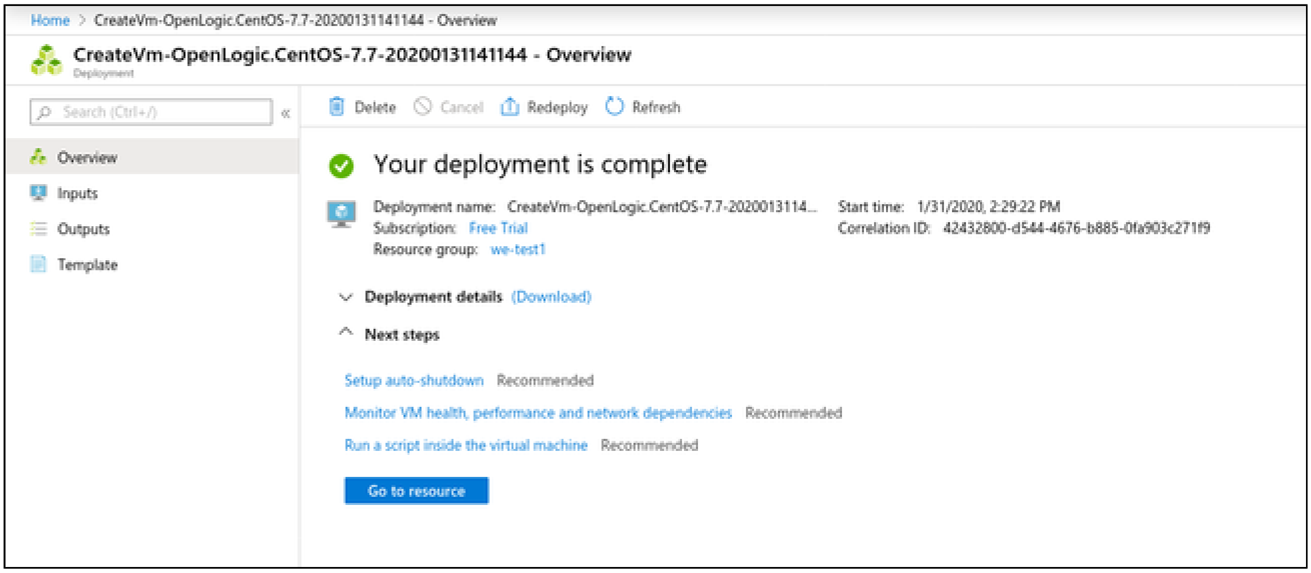
An SSH key pair contains both a public key and a private key. **Azure doesn't store the private key.** After the SSH key resource is created, you won't be able to download the private key again. Learn more

Download private key and create resource
Return to create a virtual machine

The Deployment page:

Create will take you to the **Deployment** page (possibly after downloading the private SSH key) where the current status of the deployment is displayed. Once the VM has been fully deployed, the **Deployment Complete** screen will be displayed.

Sample **Deployment Complete** screen:



Home > CreateVm-OpenLogic.CentOS-7.7-20200131141144 - Overview

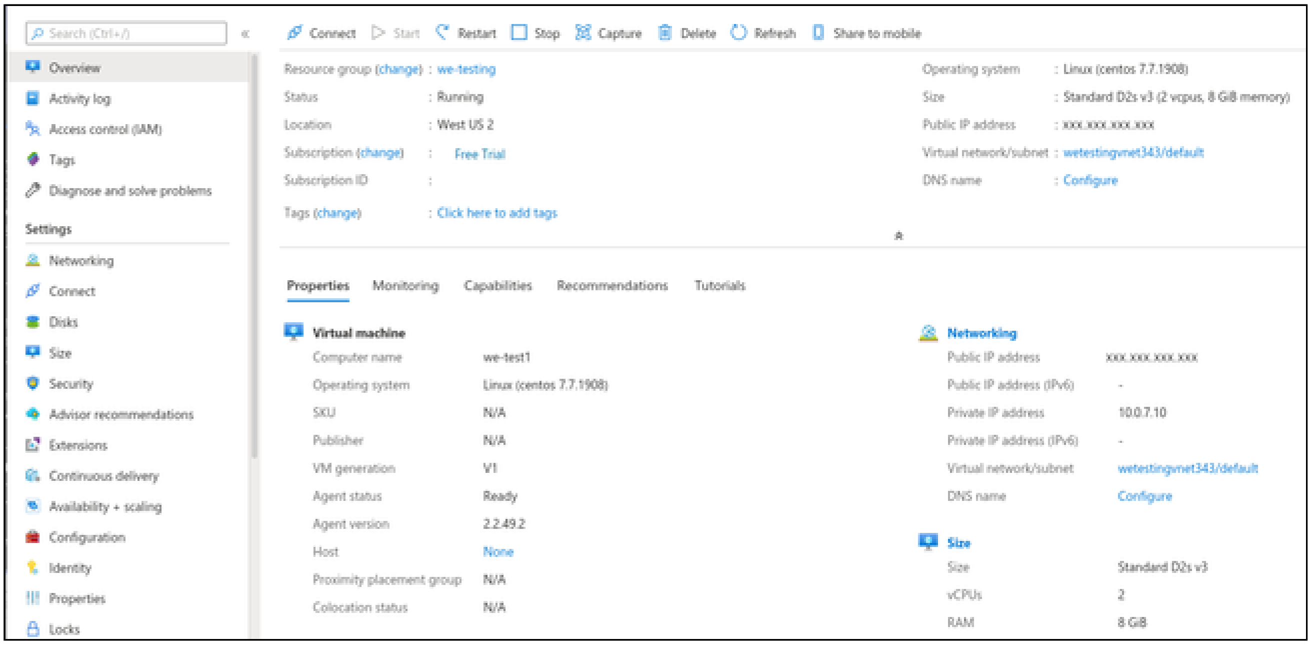
CreateVm-OpenLogic.CentOS-7.7-20200131141144 - Overview
Deployment

Your deployment is complete

Deployment name: CreateVm-OpenLogic.CentOS-7.7-2020013114...
Subscription: Free Trial
Resource group: we-test1
Start time: 1/31/2020, 2:29:22 PM
Correlation ID: 42432800-d544-4676-b885-0fa903c271f9

Search (Ctrl+/)
Overview
Inputs
Outputs
Template
Delete
Cancel
Redeploy
Refresh
Deployment details (Download)
Next steps
Setup auto-shutdown Recommended
Monitor VM health, performance and network dependencies Recommended
Run a script inside the virtual machine Recommended
Go to resource

Click on **Go to resource** to get to the details page of the newly created VM. The image below shows a sample of a detail page:



Resource group (change) : we-testing
Status : Running
Location : West US 2
Subscription (change) : Free Trial
Subscription ID :
Tags (change) : Click here to add tags

Operating system : Linux (centos 7.7.1908)
Size : Standard D2s v3 (2 vcpus, 8 GiB memory)
Public IP address : xxx.xxx.xxx.xxx
Virtual network/subnet : wetestingvnet343/default
DNS name : Configure

Virtual machine

Computer name	we-test1
Operating system	Linux (centos 7.7.1908)
SKU	N/A
Publisher	N/A
VM generation	V1
Agent status	Ready
Agent version	2.2.49.2
Host	None
Proximity placement group	N/A
Colocation status	N/A

Networking

Public IP address	xxx.xxx.xxx.xxx
Public IP address (IPv6)	-
Private IP address	10.0.7.10
Private IP address (IPv6)	-
Virtual network/subnet	wetestingvnet343/default
DNS name	Configure

Size

Size	Standard D2s v3
vCPUs	2
RAM	8 GiB

Initial Access to the Instance

Once you have access to the instance, you can create the access you require for your applications. This section just shows the basic steps for initial access to the instance.

SSH Interactive Access

To connect to the instance interactively, you must connect as the management user of your instance. Use the following command:

```
$ ssh -o ServerAliveInterval=30 -i <path-to-your-private-key> <management-user-name>@<cloudhost-IP-address>
```

The parameter `ServerAliveInterval` will protect the connection from timing out.

Please note:

- Depending on the type of connection, you will have to use either the public IP address of the cloud system or its address in a customer-specific VPN.
- The **private key** used must correspond to the public key installed in the *authorized_keys* file of the cloud instance management user. This is usually done during initial cloud instance launch.
- The management user account normally allows sudo access to privileged commands (use **sudo -i**).
- If the instance was created using a Stromasys-provided AL or VE marketplace image, the management user for **interactive login** is the user **sshuser**.

File Transfer with SFTP

SFTP enables file transfers to and from the cloud instance. Use the management user of your instance. The security rules must allow SSH access to allow SFTP access to the cloud instance.

Please note: Depending on the type of connection, you will have to use either the public IP address of the cloud system or its address in a customer-specific VPN.

To connect to the instance, use the following command:

```
$ sftp -i <path-to-your-private-key> <management-user-name>@<cloudhost-IP-address>
```

Please note:

- Depending on the type of connection, you will have to use either the public IP address of the cloud system or its address in a customer-specific VPN.
- The **private key** used must correspond to the public key installed in the *authorized_keys* file of the cloud instance management user. This is usually done during initial cloud instance launch.
- If the instance was created using a Stromasys-provided AL or VE marketplace image, the management user for **file transfer** is the user **charon**.
- If the user **charon** is used to transfer files, the home directory for the file transfer will be */charon/storage.*

Setting up a Linux Instance in GCP

Contents

General Prerequisites

As this description shows the basic setup of a Linux instance in the GCP cloud, it does not list specific prerequisites. However, depending on the use case, the following prerequisites should be considered:

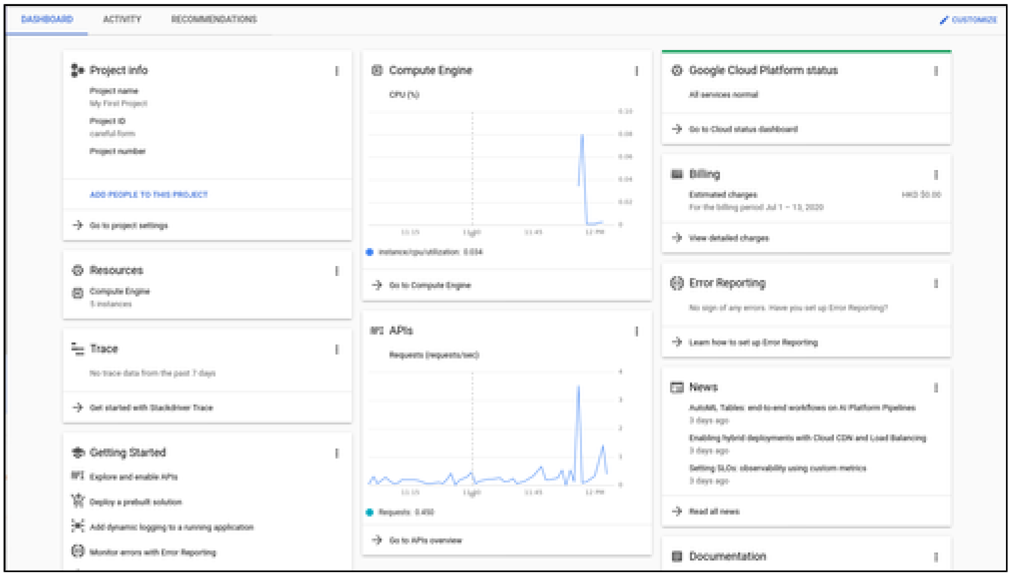
- To set up a Linux instance in the GCP cloud, you need an GCP account.
- Secondly, **prerequisites will be different depending on the planned use of the instance**:
  - Option 1: the instance is to be used as a **Charon emulator host system**:
    - Refer to the hardware and software prerequisite sections of the User's Guide and/or Getting Started guide of your Charon product to determine the exact hardware and software prerequisites that must be fulfilled by the Linux instance. The **image** you use to launch your instance and the **instance type** you chose determine the software and hardware of your cloud instance.
    - A Charon product **license** is required to run emulated legacy systems. Contact your Stromasys representative or Stromasys VAR for details.
  - Option 2: the instance is to be used as a dedicated **VE license server**:
    - Refer to the VE License Server Guide for detailed prerequisites.
- Certain legacy operating systems that can run in the emulated systems provided by Charon emulator products require a license of the original vendor of the operating system. The user is responsible for any licensing obligations related to the legacy operating system and has to provide the appropriate licenses.

GCP Login and New Instance Launch

Logging in to GCP

To log in perform the following steps:

- Go to https://console.cloud.google.com. You will see the login screen.
- Enter your login credentials.
- Upon successful login, a Google cloud dashboard screen will be displayed similar to the example below:

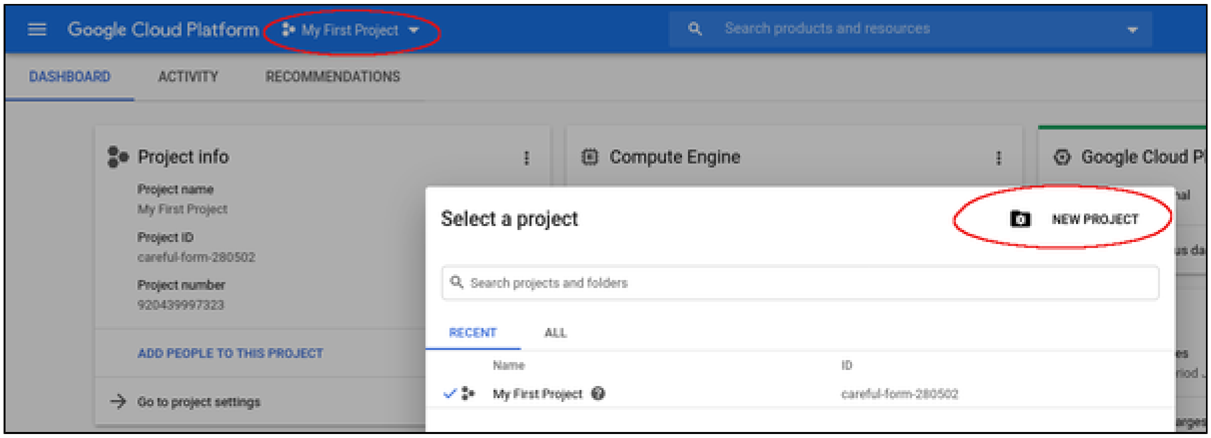

DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE
Project info
Project name
My First Project
Project ID
careful-form
Project number
ADD PEOPLE TO THIS PROJECT
Go to project settings
Resources
Compute Engine
5 instances
Trace
No trace data from the past 7 days
Get started with Stackdriver Trace
Getting Started
Explore and enable APIs
Deploy a prebuilt solution
Add dynamic logging to a running application
Monitor errors with Error Reporting
Compute Engine
CPU (%)
instance/cpu/utilization: 0.034
Go to Compute Engine
APIs
Requests (requests/sec)
Requests: 0.450
Go to APIs overview
Google Cloud Platform status
All services normal
Go to Cloud status dashboard
Billing
Estimated charges
For the billing period Jul 1 – 13, 2020
HKD $0.00
View detailed charges
Error Reporting
No sign of any errors. Have you set up Error Reporting?
Learn how to set up Error Reporting
News
AutoML Tables: end-to-end workflows on AI Platform Pipelines
3 days ago
Enabling hybrid deployments with Cloud CDN and Load Balancing
3 days ago
Setting SLOs: observability using custom metrics
3 days ago
Read all news
Documentation

Preparation

Select or Create Project

A project organizes all your Google Cloud resources. To organize all resources for a certain application purpose, you can group them in their own project. So before you start creating resources, select or create the appropriate project.

To select or create a project, select the project list from the top of the Google cloud console window, as shown below:



Either select the correct project or create a new one by clicking on the **NEW PROJECT** button.

Create VPCs and Subnets for Instance

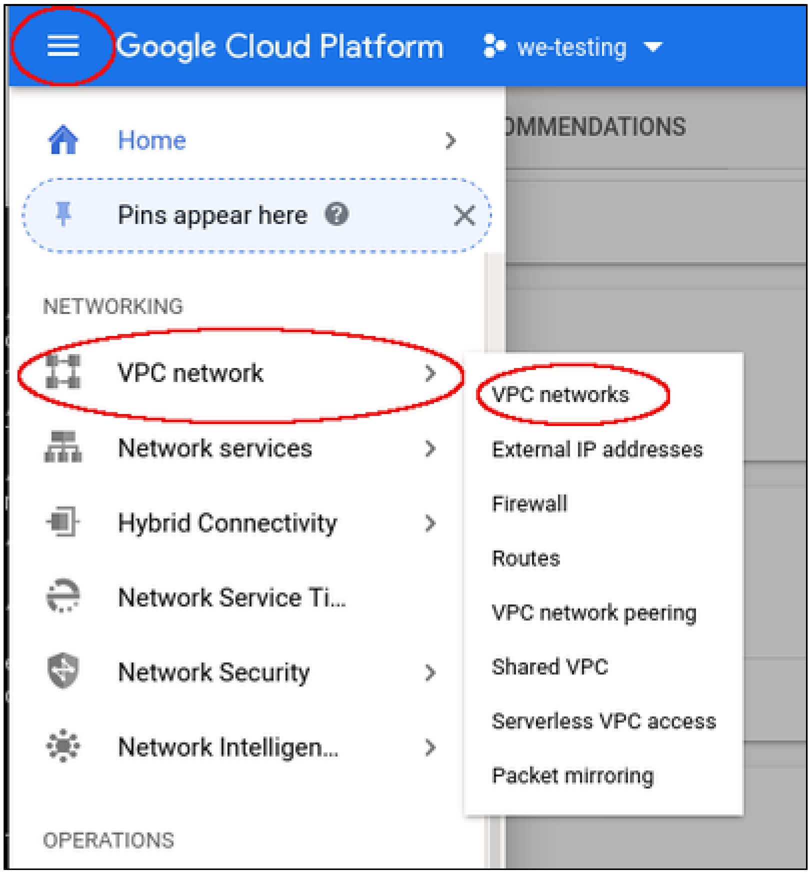
Important rules for Google cloud instances with respect to network interfaces:

- Interfaces can only be added during instance creation.
- Each network interface configured in a single instance must be attached to a different VPC network.
- The additional VPC networks that the multiple interfaces will attach to must exist before an instance is created. See Using VPC Networks for instructions on creating additional VPC networks.
- You cannot delete a network interface without deleting the instance.
- IP forwarding can only be enabled when the instance is created.
- A VPC network has a default transmission unit (MTU) of `1460` bytes for Linux images and Windows Server images. During the creation of a VPC you can set the MTU to a different value (e.g., 1500). In your instance (especially, if it does not rely on DHCP), set the MTU to the same value as configured for the VPC to avoid the increased latency and packet overhead caused by fragmentation, or even connectivity problems. For an MTU size of 1460, client applications that communicate with GCP instances over UDP must have a maximum payload of 1432 bytes to avoid fragmentation.
  In particular, **ensure that the MTU used on any Linux interface dedicated to the emulator is not smaller than the MTU used by the legacy guest system**. Failing to do so will cause network problems. For more information refer to the section *Interface MTU Considerations* in this guide.

Therefore the required VPCs and subnets must exist before the instance is created.

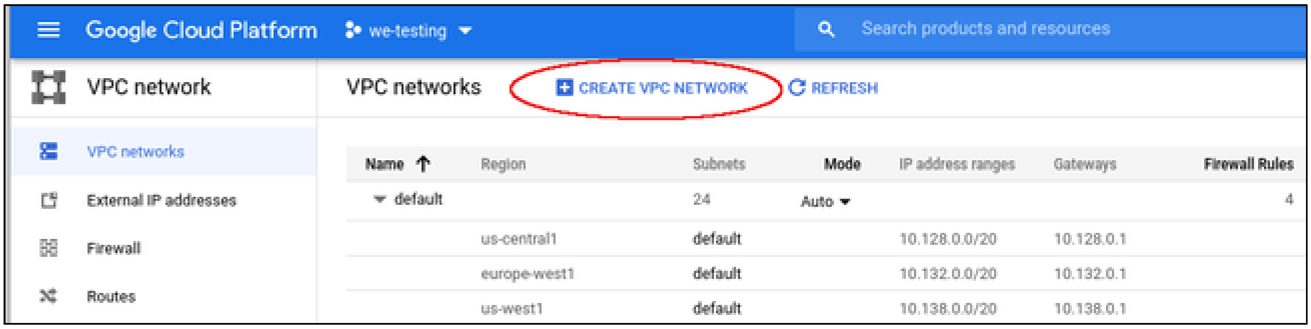
To create additional VPCs (if required), perform the following steps.

Step 1: Open the VPC network section by clicking on the Navigation menu, then selecting VPC network, and clicking on VPC networks - as illustrated below.



This will open the VPC overview page with the already existing VPCs. If all required VPCs and subnets already exist, continue with creating the new VM instance. Otherwise, continue with step 2.

Step 2: If you need to create a new VPC, click on **CREATE VPC NETWORK** at the top of the VPC overview list.



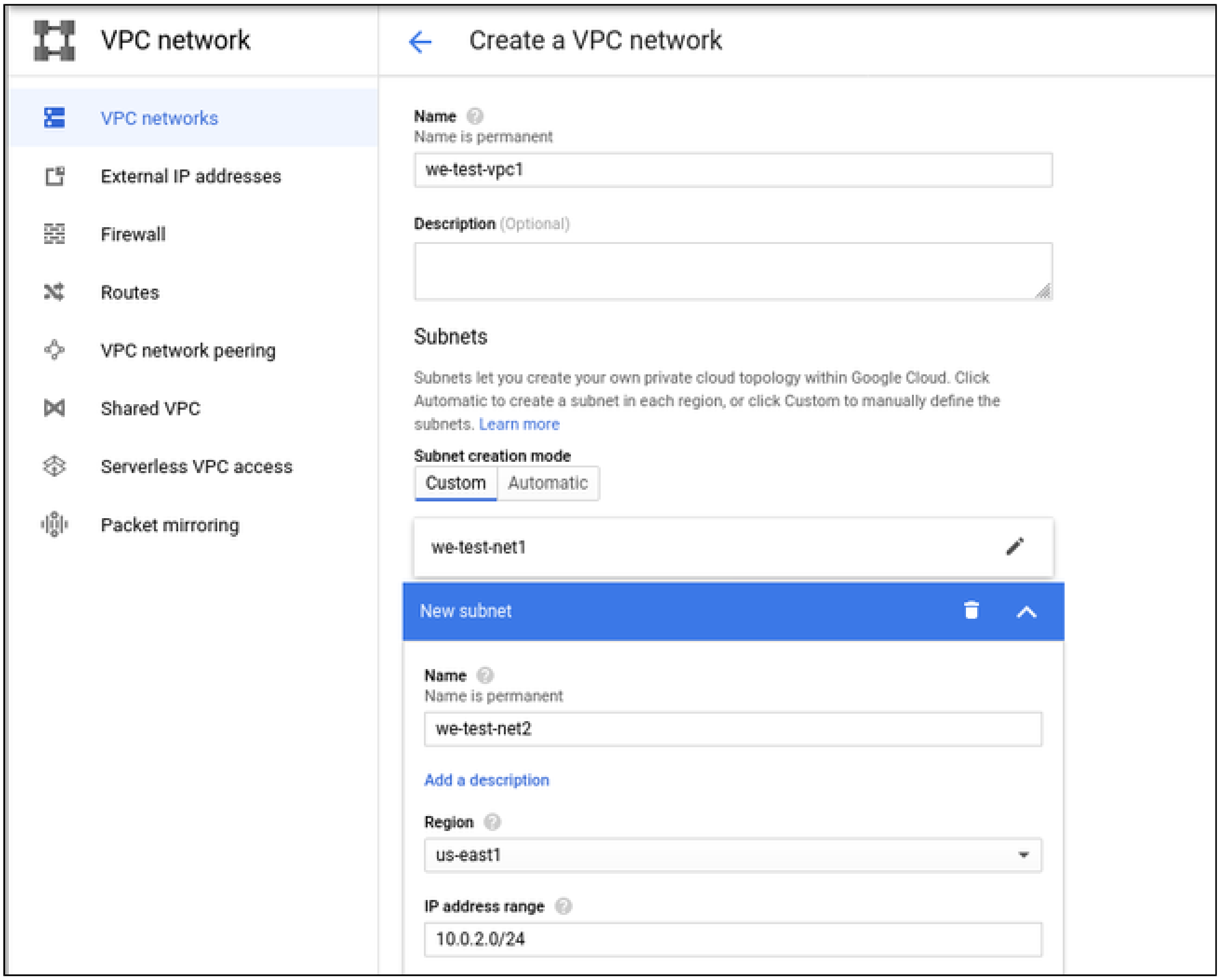
Name	Region	Subnets	Mode	IP address ranges	Gateways	Firewall Rules
default		24	Auto			4
	us-central1	default		10.128.0.0/20	10.128.0.1	
	europe-west1	default		10.132.0.0/20	10.132.0.1	
	us-west1	default		10.138.0.0/20	10.138.0.1	

This opens the VPC configuration window.

Step 3: Create VPC and subnets.

In the VPC configuration window, enter

- the VPC name,
- the subnet name, region and address, and
- optionally, an **alternative MTU size** (at the bottom of the window). The default MTU is 1460 bytes. If you want to dedicate an interface in this VPC to the emulator, this may cause problems as the default MTU size of the legacy guest systems is usually 1500 bytes. **The interface dedicated to the emulator must not have an MTU smaller than the MTU used by the legacy guest system**.



Name
Name is permanent
we-test-vpc1

Description (Optional)

Subnets

Subnets let you create your own private cloud topology within Google Cloud. Click Automatic to create a subnet in each region, or click Custom to manually define the subnets. Learn more

Subnet creation mode: Custom

we-test-net1

New subnet

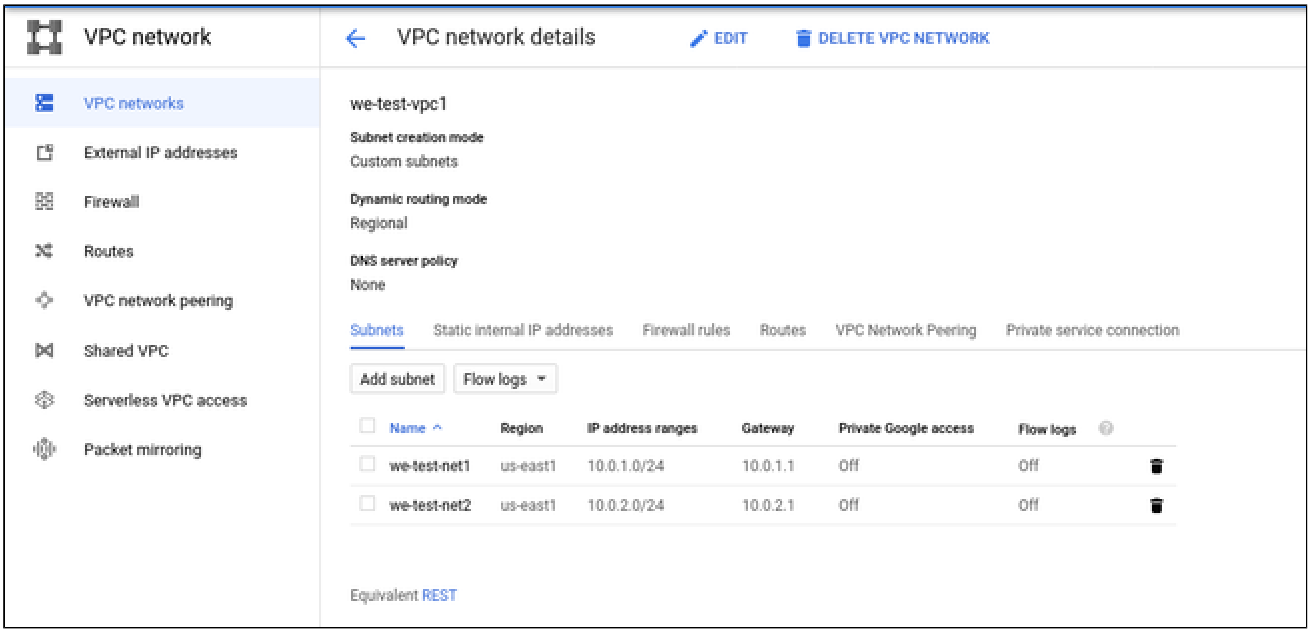
Name
Name is permanent
we-test-net2

Region
us-east1

IP address range
10.0.2.0/24

Click on **Create** at the bottom of the window to create the VPC.

The new VPC should appear in the VPC overview list. Selecting the VPC in the overview list will open the detail information window. Example:



we-test-vpc1

Subnet creation mode
Custom subnets

Dynamic routing mode
Regional

DNS server policy
None

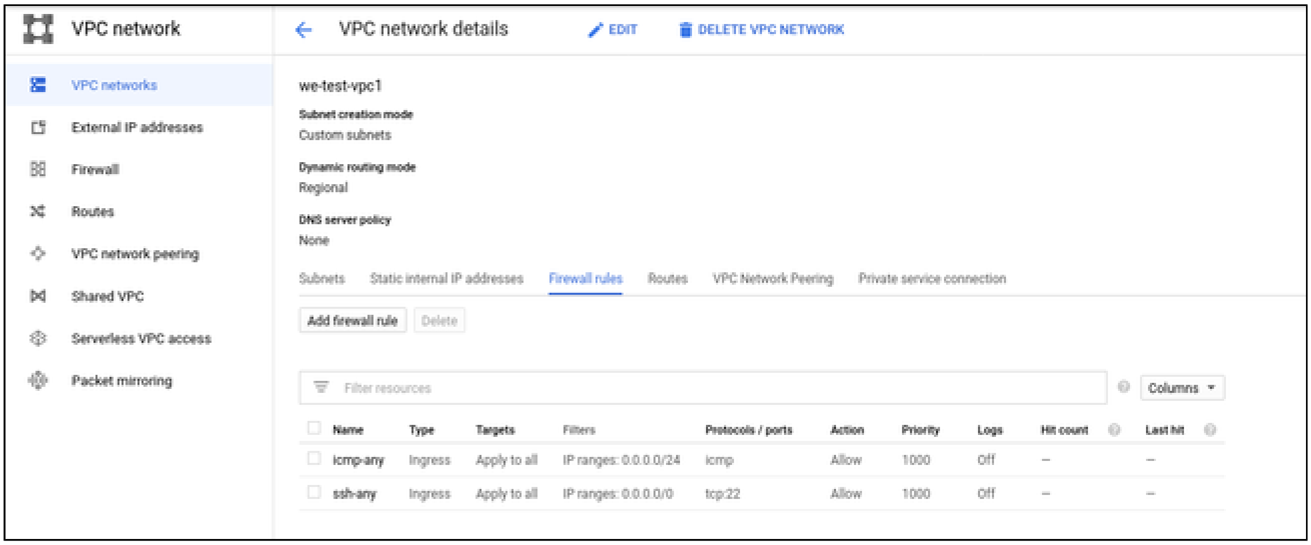
	Name	Region	IP address ranges	Gateway	Private Google access	Flow logs
□	we-test-net1	us-east1	10.0.1.0/24	10.0.1.1	Off	Off
□	we-test-net2	us-east1	10.0.2.0/24	10.0.2.1	Off	Off

Equivalent REST

Step 4: Create firewall rules for the VPC.

With the detail information open, click on Firewall. This will allow you to define the required firewall rules for the VPC.

An example of a small set of firewall rules that allow incoming SSH and ICMP is shown below:



we-test-vpc1

Subnet creation mode
Custom subnets

Dynamic routing mode
Regional

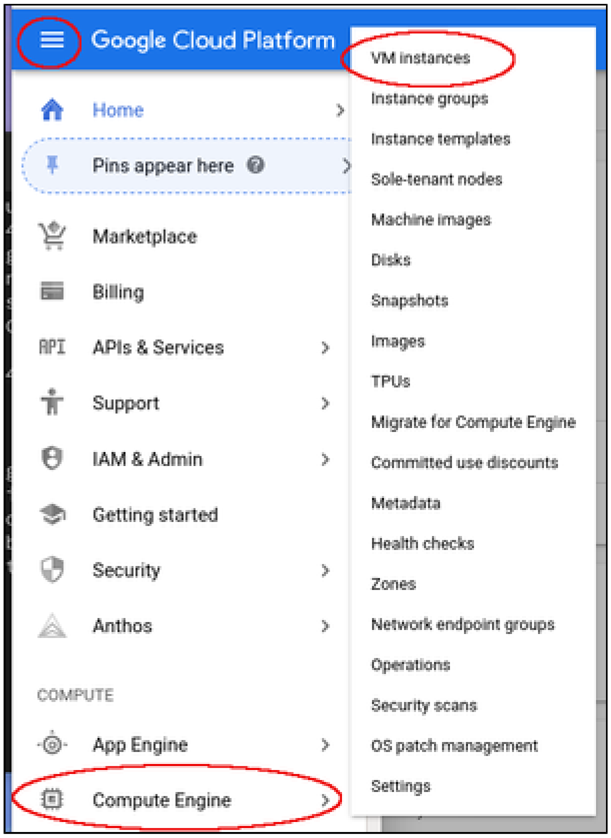
DNS server policy
None

	Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Logs	Hit count	Last hit
□	icmp-any	Ingress	Apply to all	IP ranges: 0.0.0.0/24	icmp	Allow	1000	Off	–	–
□	ssh-any	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	1000	Off	–	–

Creating a New VM Instance

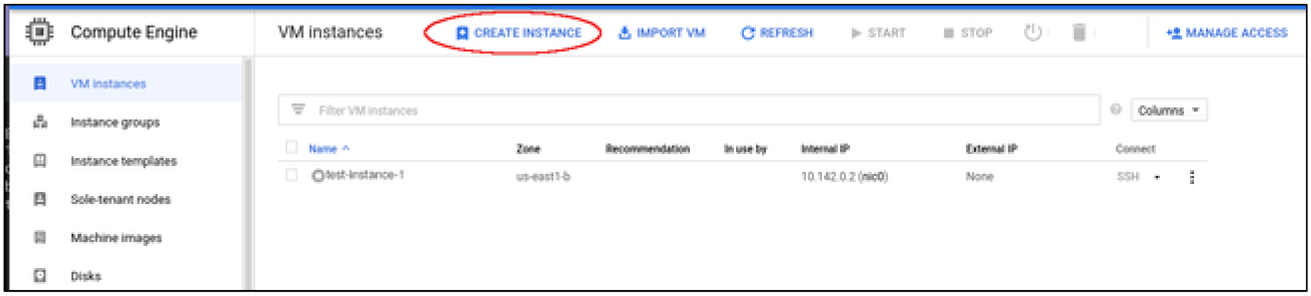
Step 1: Go to the VM instance overview page.

Open the Navigation menu, click on Compute Engine and then on VM Instances as illustrated below:



This will open the list of already existing VM instances.

Step 2: Click on **CREATE INSTANCE** at the top of the overview list.



Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
test-instance-1	us-east1-b			10.142.0.2 (nic0)	None	SSH

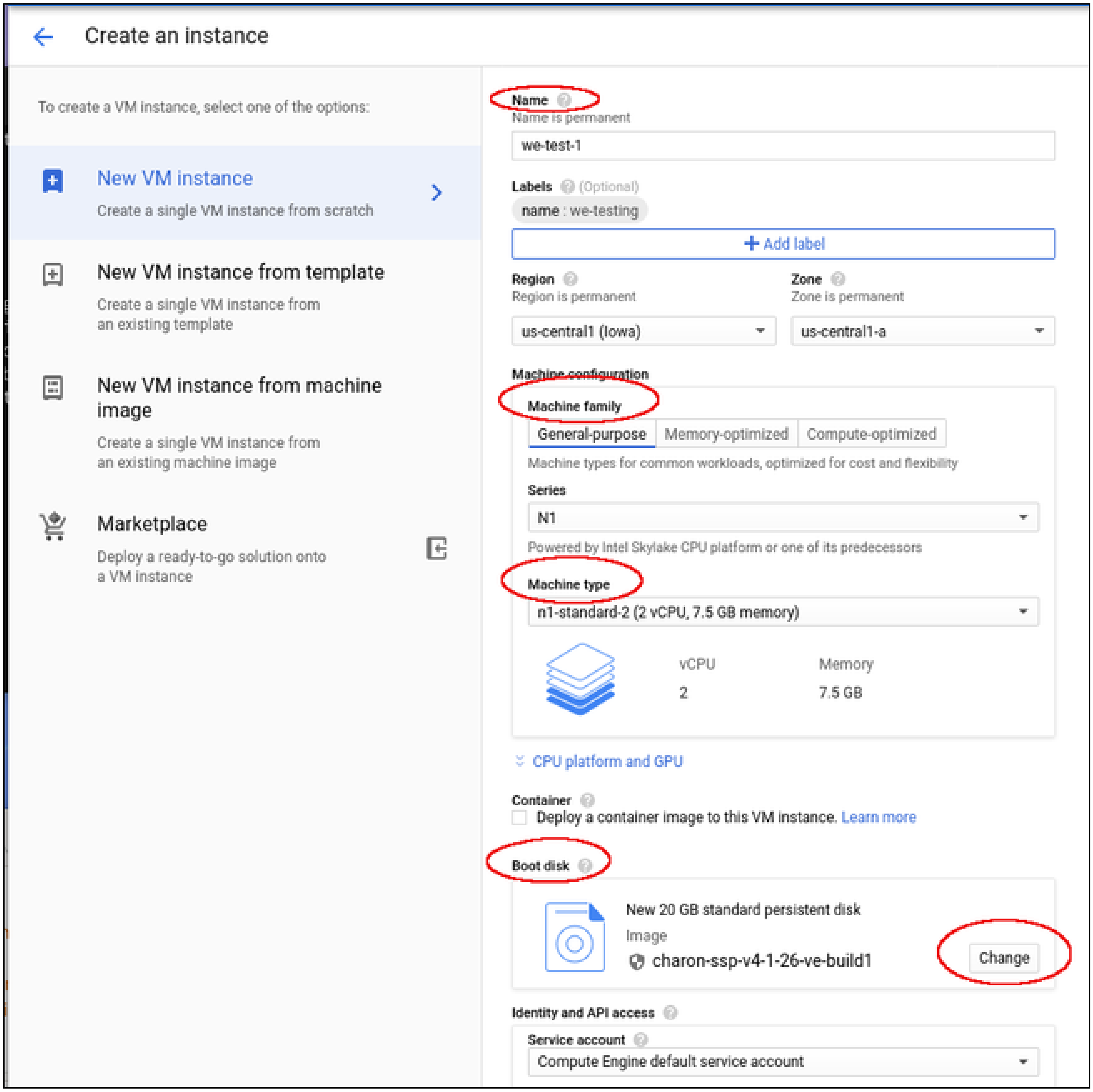
This will open the VM creation window as shown below.

Step 3: Configure the basic information of your new VM instance.

In the main configuration window set the following information at a minimum:

- **Name** of the instance (permanent setting)
- Correct **Machine family** and **Machine type** to match the requirements of the Charon products installed on the instance.
- **Boot disk** type and size, and the image to use as the operating system (recommended minimum system disk size: 30GB). To change the image for, press the **Change** button and select the correct image. If installing a prepackaged marketplace Charon image, select the matching image. If you plan to install your Charon product using RPM packages, use a Linux version supported for your product.

The following image illustrates the basic settings:



Create an instance

To create a VM instance, select one of the options:

New VM instance
Create a single VM instance from scratch

New VM instance from template
Create a single VM instance from an existing template

New VM instance from machine image
Create a single VM instance from an existing machine image

Marketplace
Deploy a ready-to-go solution onto a VM instance

Name
Name is permanent
we-test-1

Labels (Optional)
name : we-testing
+ Add label

Region
Region is permanent
us-central1 (Iowa)

Zone
Zone is permanent
us-central1-a

Machine configuration

Machine family
General-purpose | Memory-optimized | Compute-optimized
Machine types for common workloads, optimized for cost and flexibility

Series
N1
Powered by Intel Skylake CPU platform or one of its predecessors

Machine type
n1-standard-2 (2 vCPU, 7.5 GB memory)

vCPU	Memory
2	7.5 GB

CPU platform and GPU

Container
□ Deploy a container image to this VM instance. Learn more

Boot disk
New 20 GB standard persistent disk
Image
charon-ssp-v4-1-26-ve-build1
Change

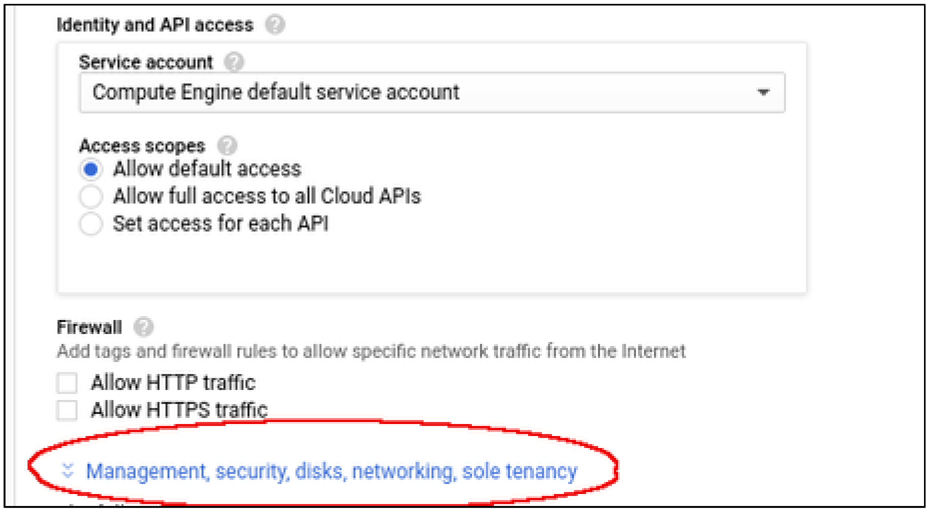
Identity and API access
Service account
Compute Engine default service account

Additional points to note:

- The **CPU platform and GPU** section provides the option to define the vCPU to core ration. That is, you can modify the settings such that each CPU visible to the host operating system corresponds to one CPU core of the GCP instance.
- In the **Identity and API access** section by default a service account (**Compute Engine default service account**) and the **Default access** scope are assigned to the instance. If this instance is to be used as a VE license server, **do not modify these settings** unless you are confident that you can provide equivalent permissions in a custom configuration. The VE license server will not function correctly without these permissions.

Step 4: Add you SSH key for remote access to the cloud instance.

Open the advanced settings at the bottom of the VM creation window by clicking on **Management, security, disks,...** at the bottom of the page.



Identity and API access

Service account

Compute Engine default service account

Access scopes

● Allow default access

○ Allow full access to all Cloud APIs

○ Set access for each API

Firewall

Add tags and firewall rules to allow specific network traffic from the Internet

□ Allow HTTP traffic

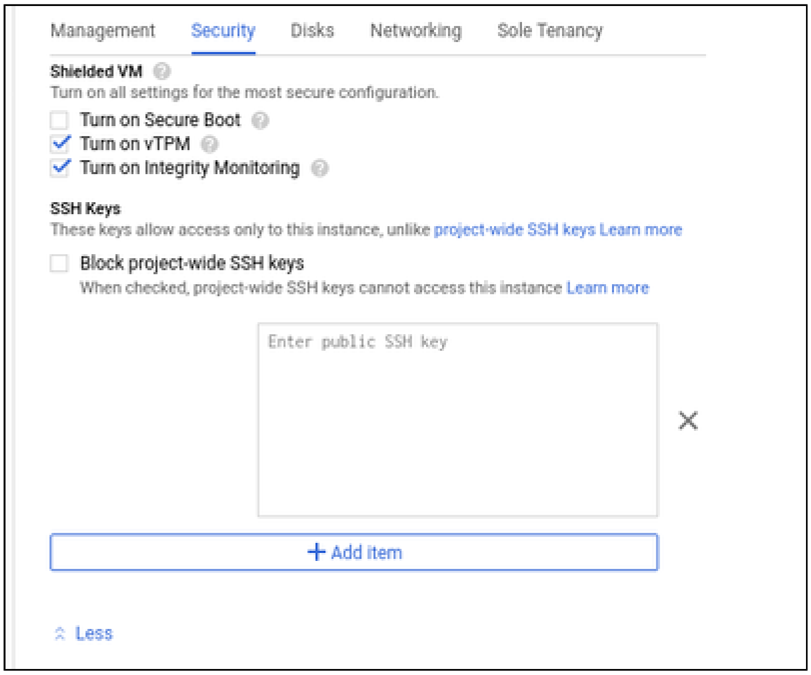
□ Allow HTTPS traffic

The advanced settings allow you to create and add disks and network interfaces during the creation of a VM.

Please note: network interfaces can only be added during the creation of a VM instance.

The advanced settings also allow you to add your public SSH key for accessing the VM once started. To do this,

- select the tab **Security** in the advanced settings section,
- paste your **public key** into the field provided (the username extracted from the key will be displayed).
- **Please note**: if your management system supports it, for RHEL 9.x, Rocky Linux 9.x, and Oracle Linux 9.x use SSH key types ECDSA or ED25519. This will allow connecting to these Charon host Linux systems using an SSH tunnel without the default crypto-policy settings on the Charon host having to be changed for less secure settings. This is, for example, important for the Charon-SSP Manager. See also: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/security_hardening/using-the-system-wide-cryptographic-policies_security-hardening.



Shielded VM

Turn on all settings for the most secure configuration.

□ Turn on Secure Boot

☑ Turn on vTPM

☑ Turn on Integrity Monitoring

SSH Keys

These keys allow access only to this instance, unlike project-wide SSH keys Learn more

□ Block project-wide SSH keys

When checked, project-wide SSH keys cannot access this instance Learn more

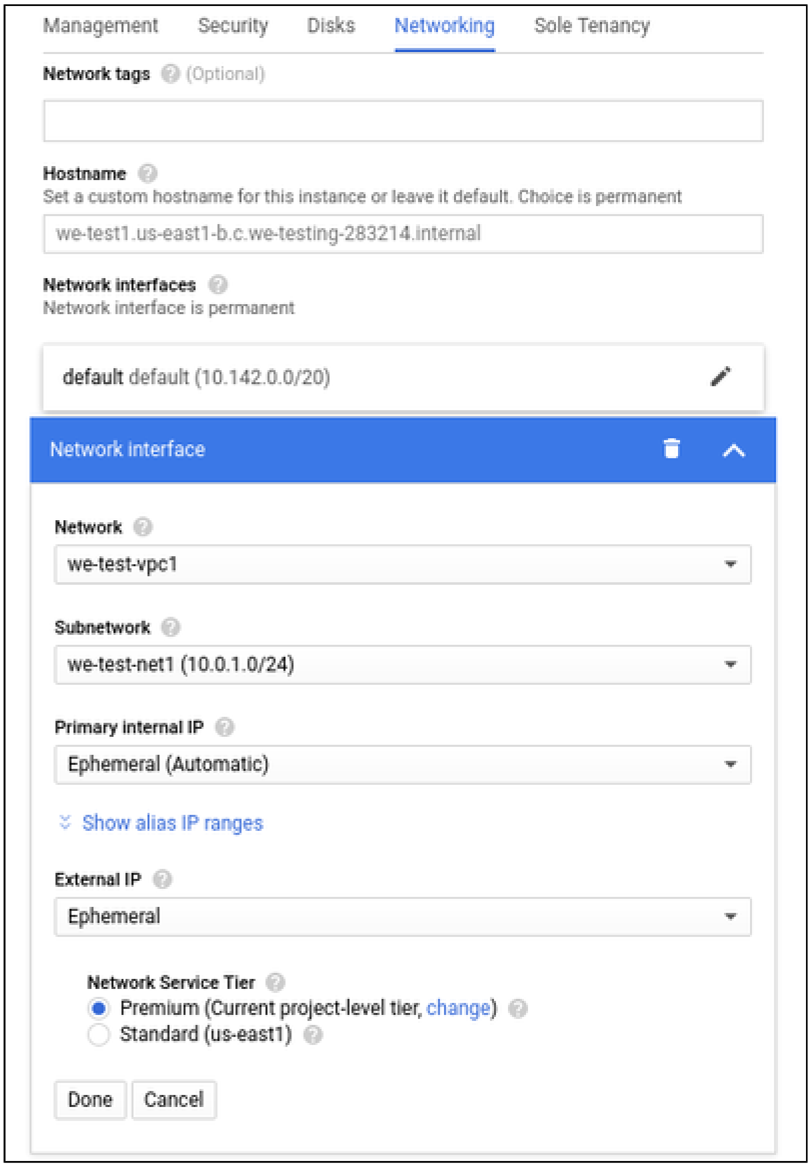
Enter public SSH key

You can collapse the section again by clicking on **Less**.

Step 5: Optionally, configure additional NICs and/or IP forwarding

To add an **additional network interface**, perform the following steps:

- Open the advanced settings at the bottom of the VM creation window by clicking on **Management, security, disks,...** at the bottom of the page.
- Select Networking from the advanced settings section.
- Click on **Add network interface**.
- Select the correct subnet.
- Set the information about internal and external IP address (static or ephemeral) as required.



Network tags (Optional)

Hostname
Set a custom hostname for this instance or leave it default. Choice is permanent
we-test1.us-east1-b.c.we-testing-283214.internal

Network interfaces
Network interface is permanent

default default (10.142.0.0/20)

Network: we-test-vpc1

Subnetwork: we-test-net1 (10.0.1.0/24)

Primary internal IP: Ephemeral (Automatic)

External IP: Ephemeral

Network Service Tier
● Premium (Current project-level tier, change)
○ Standard (us-east1)

After adding all the required information, click on **Done**.

To enable **IP forwarding,** perform the following steps:

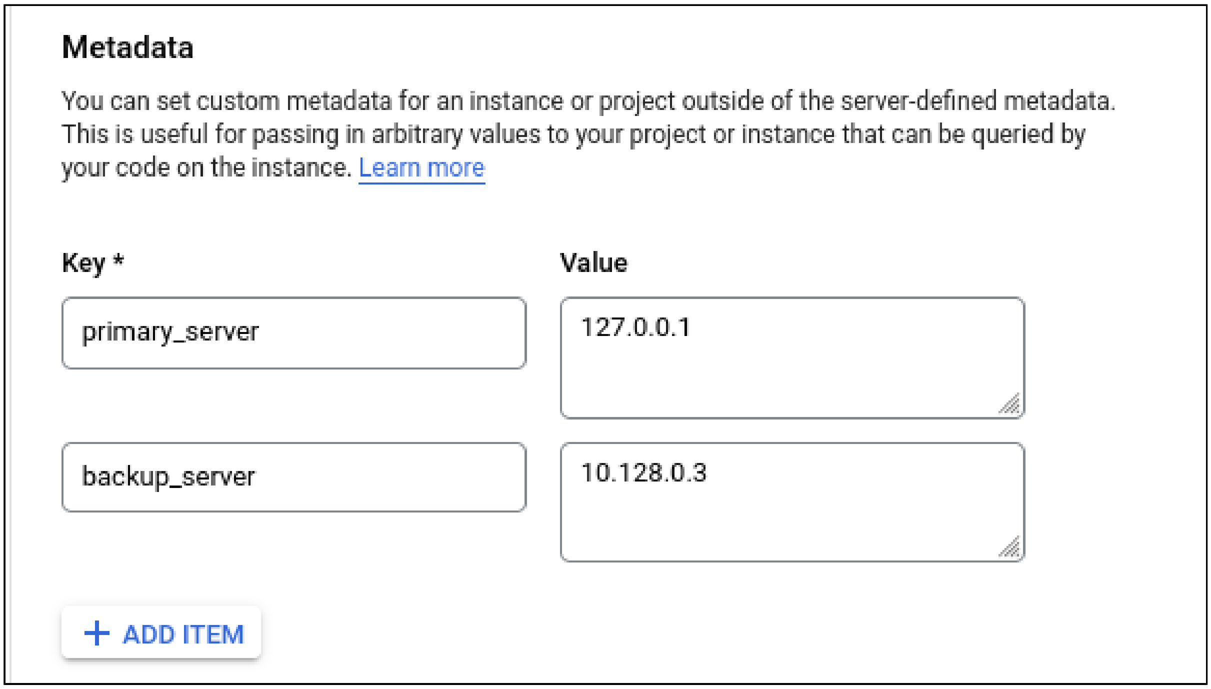
- Open the advanced settings at the bottom of the VM creation window by clicking on **Management, security, disks,...** at the bottom of the page.
- Select Networking from the advanced settings section.
- Select the edit option for the default NIC.
- Enable IP forwarding
- Click on **Done**.

Please note: you have to set up a firewall manually when you add additional network interfaces. See Network Management and the GCP documentation for more detail.

Step 6: additional configuration for **AutoVE** setup.

If the instance is launched from a Charon AL marketplace image and is planned to use AutoVE licensing (instead of the public license servers), you must add the corresponding information to the instance configuration **before** the first launch of the instance:

The AutoVE license server information is entered as **Custom Metadata**. In the initial instance configuration window, go to the bottom where the **NETWORKING, DISKS, SECURITY, MANAGEMENT...** configuration section is located. Open it and select the **Management** section. Add the Custom Metadata as shown in the example below:



Metadata

You can set custom metadata for an instance or project outside of the server-defined metadata. This is useful for passing in arbitrary values to your project or instance that can be queried by your code on the instance. Learn more

Key *	Value
primary_server	127.0.0.1
backup_server	10.128.0.3

Valid User Data configuration options:

- `primary_server <ip-address>[:<port>]`
- `backup_server  <ip-address>[:<port>]`

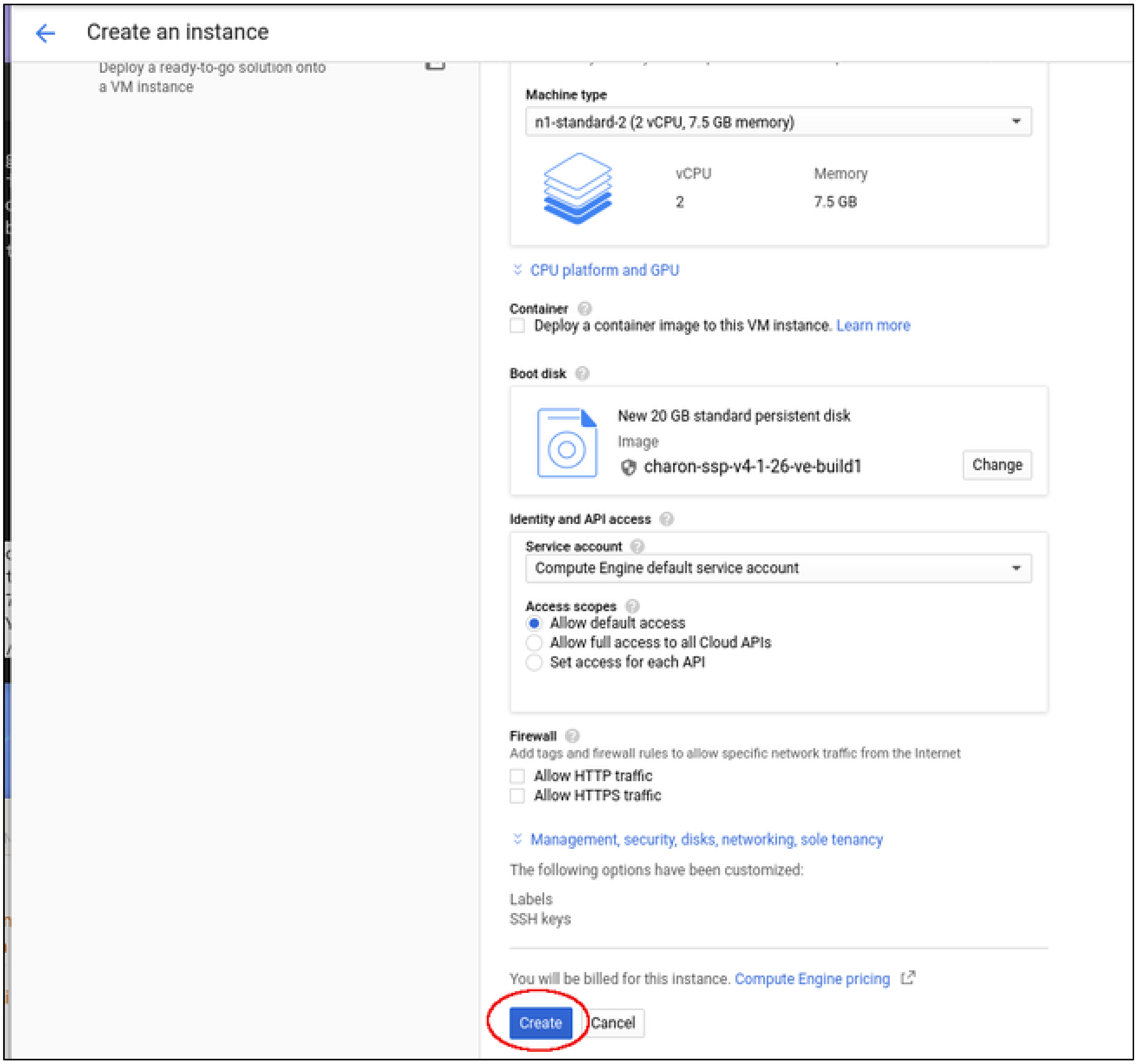
where

- *<ip-address>* stands for the IP address of the primary and the backup server as applicable, and
- *<port>* stands for a non-default TCP port used to communicate with the license server (default: TCP/8083).

Please note: at least one license server must be configured at initial launch to enable AutoVE mode. **Otherwise, the instance will bind to one of the public license servers operated by Stromasys.**

Step 7: Create the VM.

Once you filled in all the required data, create the VM by pressing the **Create** button at the bottom of the page:



Create an instance

Deploy a ready-to-go solution onto a VM instance

Machine type

n1-standard-2 (2 vCPU, 7.5 GB memory)

vCPU 2

Memory 7.5 GB

CPU platform and GPU

Container

□ Deploy a container image to this VM instance. Learn more

Boot disk

New 20 GB standard persistent disk

Image

charon-ssp-v4-1-26-ve-build1

Change

Identity and API access

Service account

Compute Engine default service account

Access scopes

● Allow default access

○ Allow full access to all Cloud APIs

○ Set access for each API

Firewall

Add tags and firewall rules to allow specific network traffic from the Internet

□ Allow HTTP traffic

□ Allow HTTPS traffic

Management, security, disks, networking, sole tenancy

The following options have been customized:

Labels

SSH keys

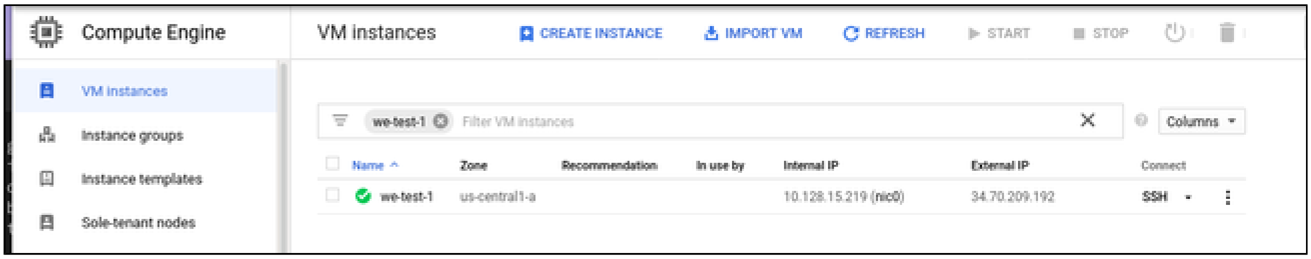
You will be billed for this instance. Compute Engine pricing

Create Cancel

This will create the VM, start it and show it in the VM instances list.

<u>Step 8</u><u>: Verify the settings of the newly created cloud instance.</u>

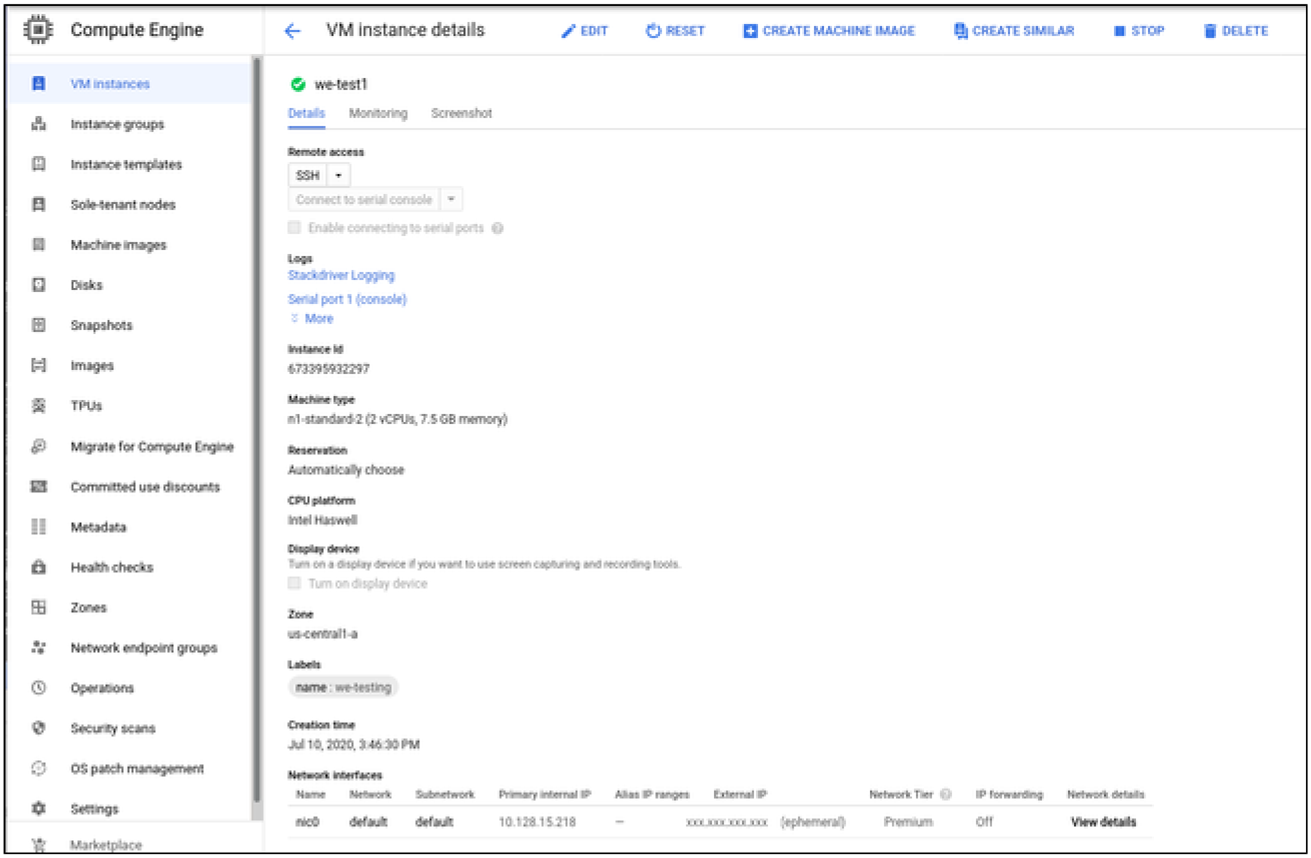
After successful creation, the new instance will be shown in the VM instances list:



we-test-1

	Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
☐	we-test-1	us-central1-a			10.128.15.219 (nic0)	34.70.209.192	SSH

By clicking on it, you will see the details of the cloud instance, as shown in the example below:



we-test1

Remote access
SSH
Connect to serial console
☐ Enable connecting to serial ports

Logs
Stackdriver Logging
Serial port 1 (console)
More

Instance id
673395932297

Machine type
n1-standard-2 (2 vCPUs, 7.5 GB memory)

Reservation
Automatically choose

CPU platform
Intel Haswell

Display device
Turn on a display device if you want to use screen capturing and recording tools.
☐ Turn on display device

Zone
us-central1-a

Labels
name : we-testing

Creation time
Jul 10, 2020, 3:46:30 PM

Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	default	default	10.128.15.218	–	xxx.xxx.xxx.xxx (ephemeral)	Premium	Off	View details

Initial Access to the Instance

Once you have access to the instance, you can create the access you require for your applications. This section just shows the basic steps for initial access to the instance.

SSH Interactive Access

To connect to the instance interactively, you must connect as the management user of your instance. Use the following command:

```
$ ssh -o ServerAliveInterval=30 -i <path-to-your-private-key> <management-user-name>@<cloudhost-IP-address>
```

The parameter `ServerAliveInterval` will protect the connection from timing out.

Please note:

- Depending on the type of connection, you will have to use either the public IP address of the cloud system or its address in a customer-specific VPN.
- The **private key** used must correspond to the public key installed in the *authorized_keys* file of the cloud instance management user. This is usually done during initial cloud instance launch.
- The management user account normally allows sudo access to privileged commands (use **sudo -i**).
- If the instance was created using a Stromasys-provided AL or VE marketplace image, the management user for **interactive login** is the user **sshuser**.

File Transfer with SFTP

SFTP enables file transfers to and from the cloud instance. Use the management user of your instance. The security rules must allow SSH access to allow SFTP access to the cloud instance.

Please note: Depending on the type of connection, you will have to use either the public IP address of the cloud system or its address in a customer-specific VPN.

To connect to the instance, use the following command:

```
$ sftp -i <path-to-your-private-key> <management-user-name>@<cloudhost-IP-address>
```

Please note:

- Depending on the type of connection, you will have to use either the public IP address of the cloud system or its address in a customer-specific VPN.
- The **private key** used must correspond to the public key installed in the *authorized_keys* file of the cloud instance management user. This is usually done during initial cloud instance launch.
- If the instance was created using a Stromasys-provided AL or VE marketplace image, the management user for **file transfer** is the user **charon**.
- If the user **charon** is used to transfer files, the home directory for the file transfer will be */charon/storage.*

Setting up a Linux Instance in OCI

Contents

General Prerequisites

As this description shows the basic setup of a Linux instance in OCI, it does not list specific prerequisites. However, depending on the use case, the following prerequisites should be considered:

- To set up a Linux instance in OCI, you need an OCI account.
- Secondly, **prerequisites will be different depending on the planned use of the instance**:
  - Option 1: the instance is to be used as a **Charon emulator host system**:
    - Refer to the hardware and software prerequisite sections of the User's Guide and/or Getting Started guide of your Charon product to determine the exact hardware and software prerequisites that must be fulfilled by the Linux instance. The **image** you use to launch your instance and the **instance type** you chose determine the software and hardware of your cloud instance.
    - A Charon product **license** is required to run emulated legacy systems. Contact your Stromasys representative or Stromasys VAR for details.
  - Option 2: the instance is to be used as a dedicated **VE license server**:
    - Refer to the VE License Server Guide for detailed prerequisites.
- Certain legacy operating systems that can run in the emulated systems provided by Charon emulator products require a license of the original vendor of the operating system. The user is responsible for any licensing obligations related to the legacy operating system and has to provide the appropriate licenses.

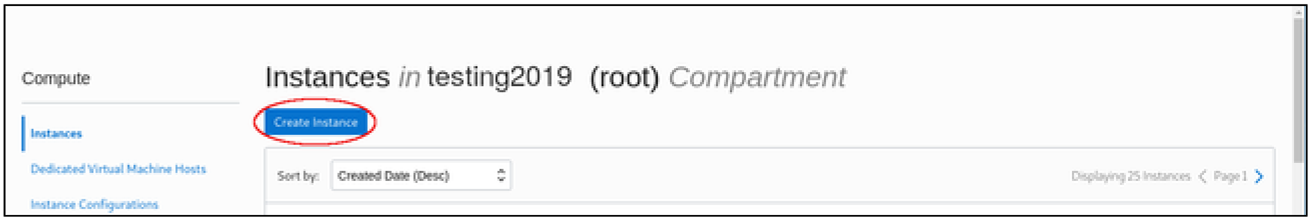
OCI New Instance Launch

Please note: This section only shows a very basic example. Please refer to the Oracle Cloud documentation for more detailed information.

To start the creation of a new cloud instance, perform the following steps:

Step 1: log in to your Oracle Cloud environment.

Step 2: go to the instance list in the compute section and click on **Create Instance**.

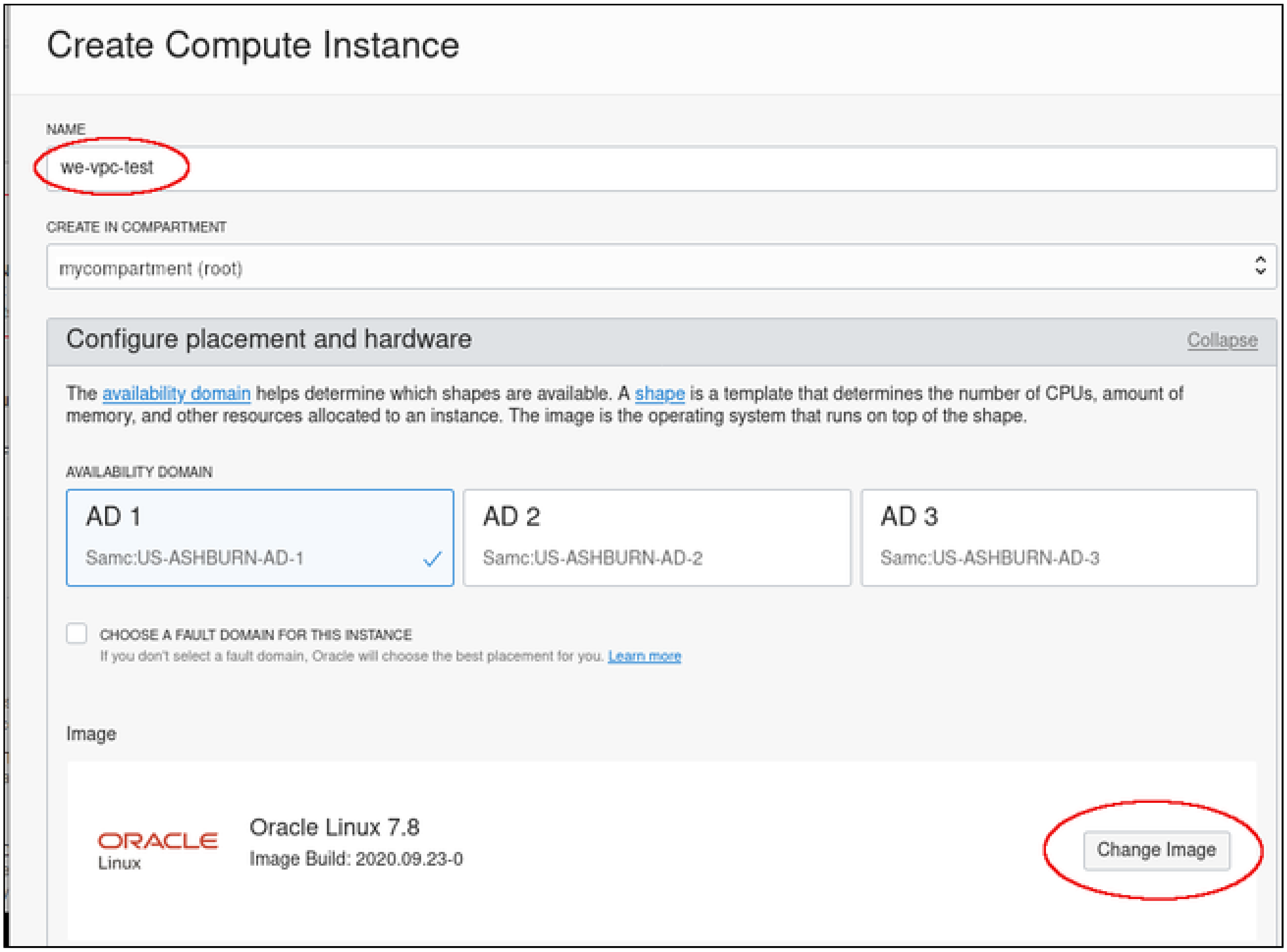


Instances *in* testing2019 (root) *Compartment*

Sort by: Created Date (Desc)

This opens the **Create Compute Instance** window.

Step 3: on the first part of **Create Compute Instance** window, name your instance and select the correct image for it. If installing a prepackaged marketplace Charon image, this image must be used. If you plan to install Charon using RPM packages, use a Linux version supported for your Charon product version.



Create Compute Instance

NAME

we-vpc-test

CREATE IN COMPARTMENT

mycompartment (root)

Configure placement and hardware

The availability domain helps determine which shapes are available. A shape is a template that determines the number of CPUs, amount of memory, and other resources allocated to an instance. The image is the operating system that runs on top of the shape.

AVAILABILITY DOMAIN

AD 1 Samc:US-ASHBURN-AD-1 ✓

AD 2 Samc:US-ASHBURN-AD-2

AD 3 Samc:US-ASHBURN-AD-3

□ CHOOSE A FAULT DOMAIN FOR THIS INSTANCE

If you don't select a fault domain, Oracle will choose the best placement for you. Learn more

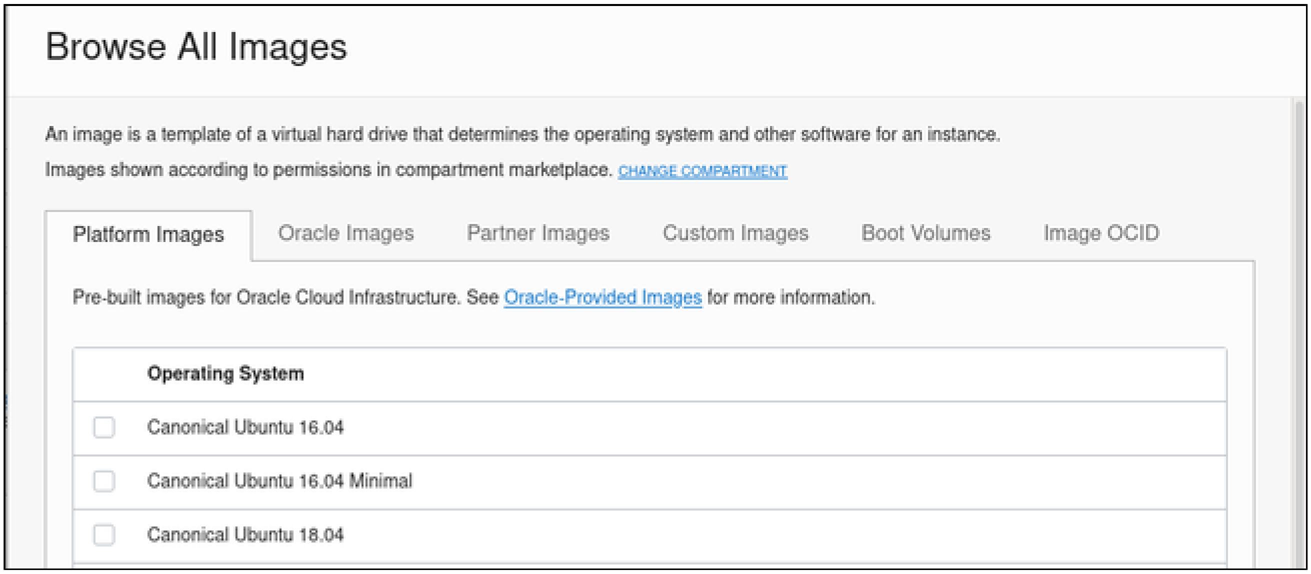
Image

Oracle Linux 7.8

Image Build: 2020.09.23-0

To select the correct image, select **Change Image**. This will allow you to browse the different available categories for the image from which to launch your instance.

The image below shows an example of the image selection screen:



Browse All Images

An image is a template of a virtual hard drive that determines the operating system and other software for an instance.

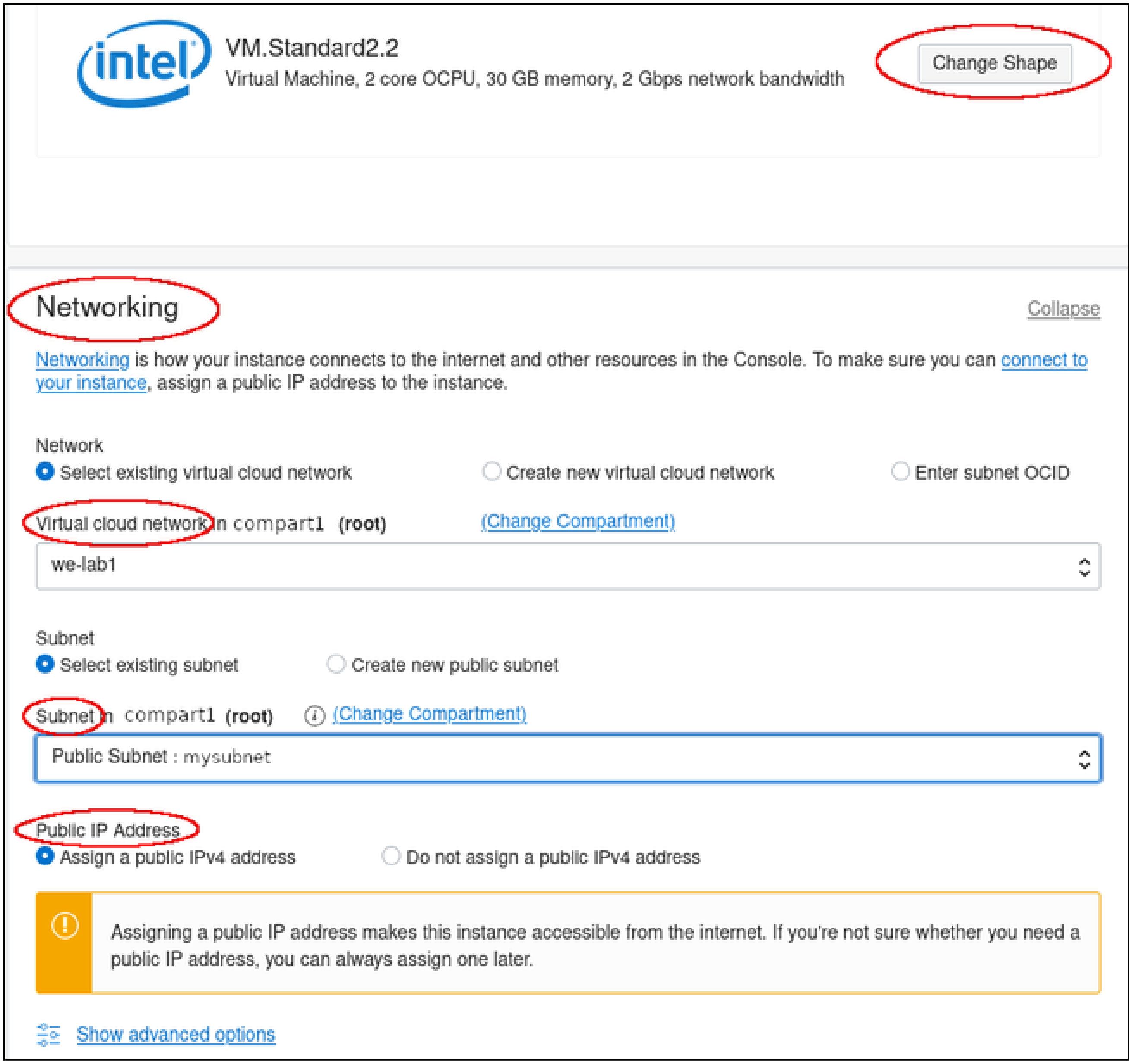
Images shown according to permissions in compartment marketplace. CHANGE COMPARTMENT

Pre-built images for Oracle Cloud Infrastructure. See Oracle-Provided Images for more information.

	Operating System
□	Canonical Ubuntu 16.04
□	Canonical Ubuntu 16.04 Minimal
□	Canonical Ubuntu 18.04

Optionally, change the compartment. Select the correct image and confirm your selection by clicking on **Select Image** at the bottom of the page. This will take you back to the **Create Compute Instance** window.

Step 4: in the middle part of the **Create Compute Instance** window, select the appropriate **shape** (i.e., the virtual Charon host hardware), the **subnet** membership of the instance and whether to assign a **public IP address**. If required, you can also create a new virtual cloud network or a new subnet here.



VM.Standard2.2

Virtual Machine, 2 core OCPU, 30 GB memory, 2 Gbps network bandwidth

Networking

Networking is how your instance connects to the internet and other resources in the Console. To make sure you can connect to your instance, assign a public IP address to the instance.

Network

● Select existing virtual cloud network ○ Create new virtual cloud network ○ Enter subnet OCID

Virtual cloud network in compart1 (root) (Change Compartment)

we-lab1

Subnet

● Select existing subnet ○ Create new public subnet

Subnet in compart1 (root) (Change Compartment)

Public Subnet : mysubnet

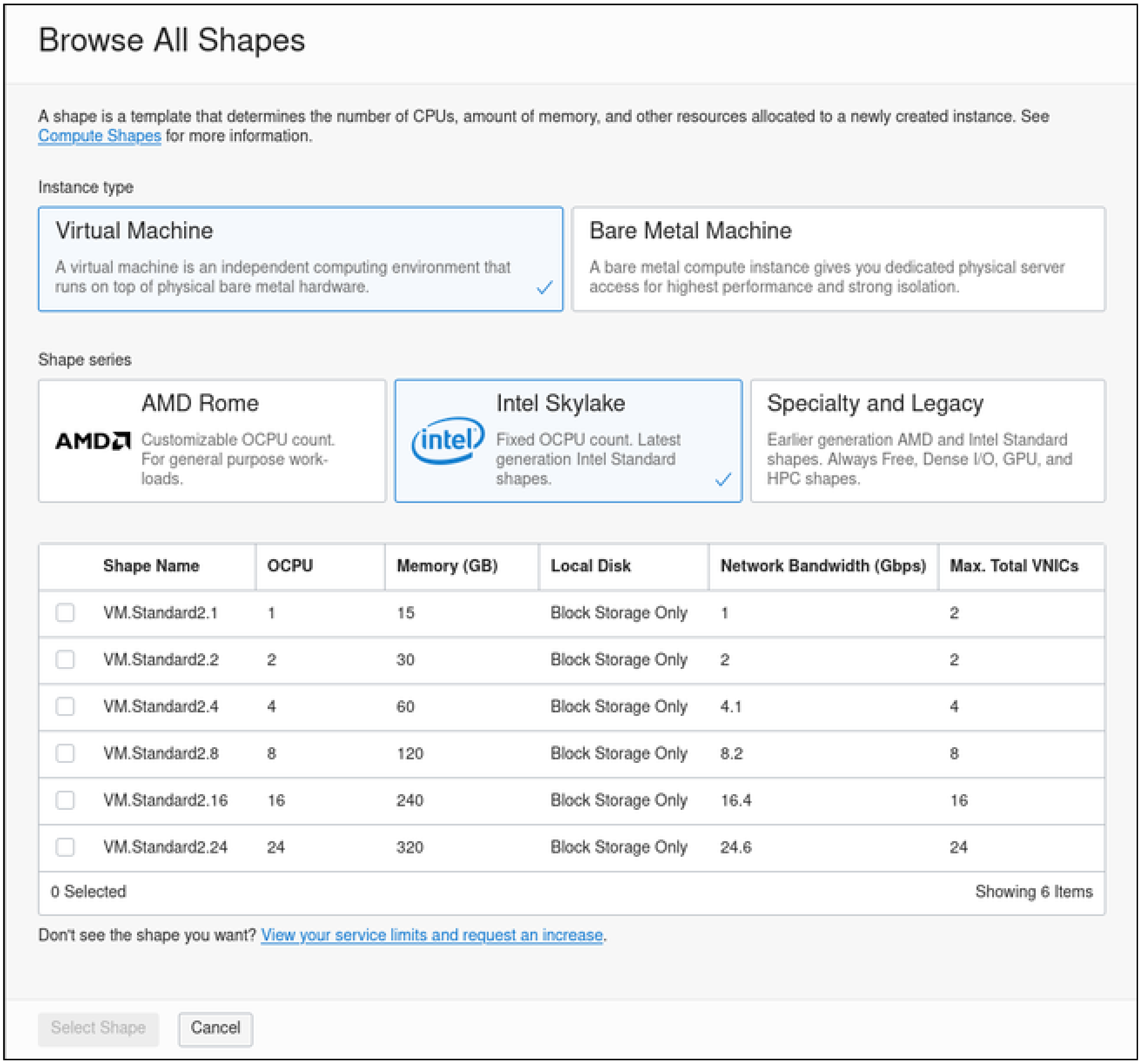
Public IP Address

● Assign a public IPv4 address ○ Do not assign a public IPv4 address

Assigning a public IP address makes this instance accessible from the internet. If you're not sure whether you need a public IP address, you can always assign one later.

To select an appropriate shape conforming to the hardware requirements of the emulated SPARC system, click on **Change Shape**.

This will open a window where you can select the correct system type. For flexible shapes you will have to configure the required number of OCPUs.



Browse All Shapes

A shape is a template that determines the number of CPUs, amount of memory, and other resources allocated to a newly created instance. See Compute Shapes for more information.

Instance type

- Virtual Machine ✓ — A virtual machine is an independent computing environment that runs on top of physical bare metal hardware.
- Bare Metal Machine — A bare metal compute instance gives you dedicated physical server access for highest performance and strong isolation.

Shape series

- AMD Rome — Customizable OCPU count. For general purpose work-loads.
- Intel Skylake ✓ — Fixed OCPU count. Latest generation Intel Standard shapes.
- Specialty and Legacy — Earlier generation AMD and Intel Standard shapes. Always Free, Dense I/O, GPU, and HPC shapes.

	Shape Name	OCPU	Memory (GB)	Local Disk	Network Bandwidth (Gbps)	Max. Total VNICs
□	VM.Standard2.1	1	15	Block Storage Only	1	2
□	VM.Standard2.2	2	30	Block Storage Only	2	2
□	VM.Standard2.4	4	60	Block Storage Only	4.1	4
□	VM.Standard2.8	8	120	Block Storage Only	8.2	8
□	VM.Standard2.16	16	240	Block Storage Only	16.4	16
□	VM.Standard2.24	24	320	Block Storage Only	24.6	24

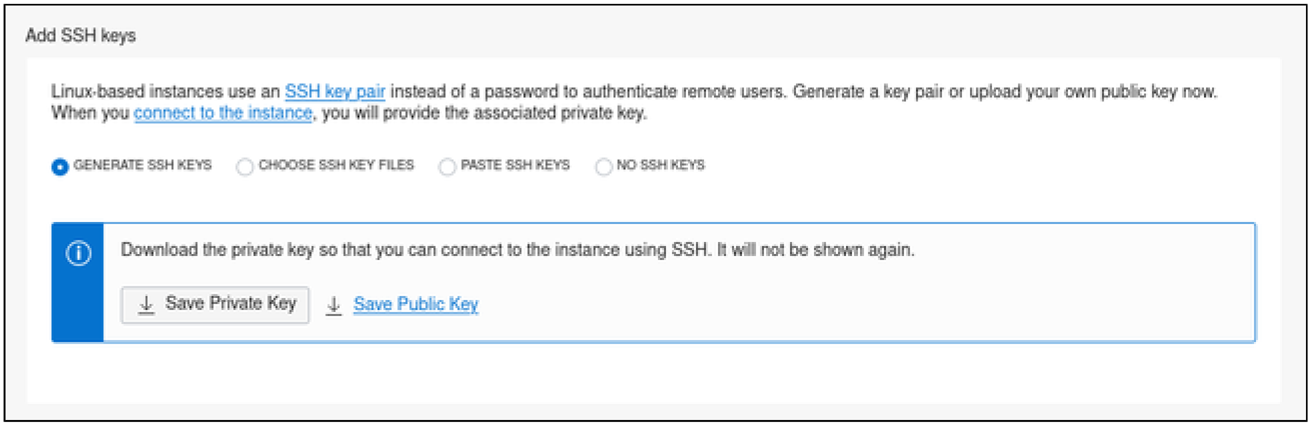
0 Selected Showing 6 Items

Don't see the shape you want? View your service limits and request an increase.

Select the appropriate shape and confirm your selection by clicking on **Select Shape** at the bottom of the page. This will take you back to the **Create Compute Instance** window.

Step 5: near the bottom of the **Create Compute Instance** window create a new SSH key-pair or upload the public SSH key of an existing key-pair that you will use to access your instance. If you create a new key-pair, you must download the private key and store it in a save place for later use. Optionally, you can also download the public key.

Please note: if your management system supports it, for RHEL 9.x, Rocky Linux 9.x, and Oracle Linux 9.x use SSH key types ECDSA or ED25519. This will allow connecting to these Charon host Linux systems using an SSH tunnel without the default crypto-policy settings on the Charon host having to be changed for less secure settings. This is, for example, important for the Charon-SSP Manager. See also: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/security_hardening/using-the-system-wide-cryptographic-policies_security-hardening.


Add SSH keys

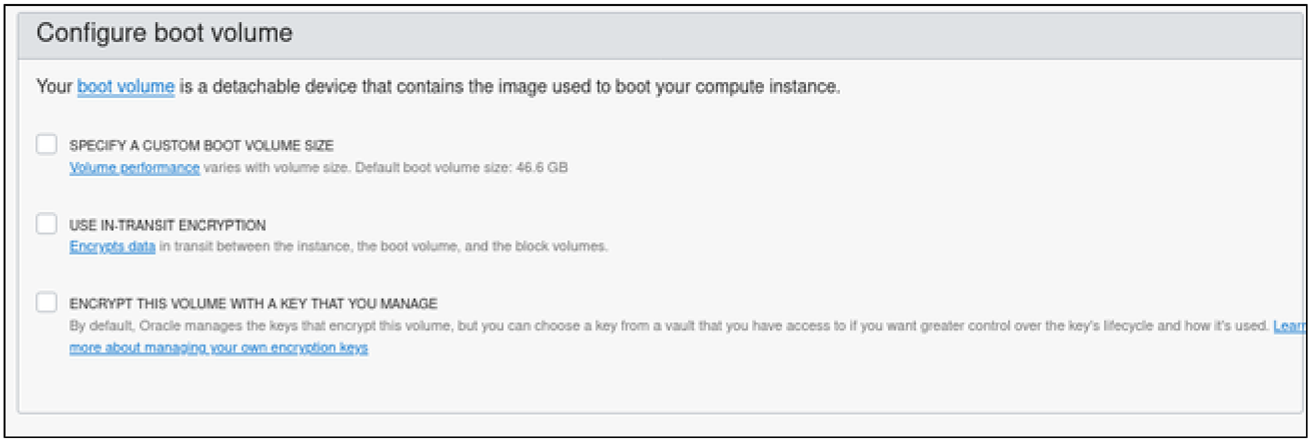
Linux-based instances use an SSH key pair instead of a password to authenticate remote users. Generate a key pair or upload your own public key now. When you connect to the instance, you will provide the associated private key.

● GENERATE SSH KEYS ○ CHOOSE SSH KEY FILES ○ PASTE SSH KEYS ○ NO SSH KEYS

Download the private key so that you can connect to the instance using SSH. It will not be shown again.

Step 6: optionally define non-default parameters (including the size) for the boot volume.

The boot volume section allows you to configure the boot volume of your instance with additional non-default parameters. For example, you can configure disk encryption parameters and a non-default system disk size (recommended minimum system disk size: 30GB).


Configure boot volume

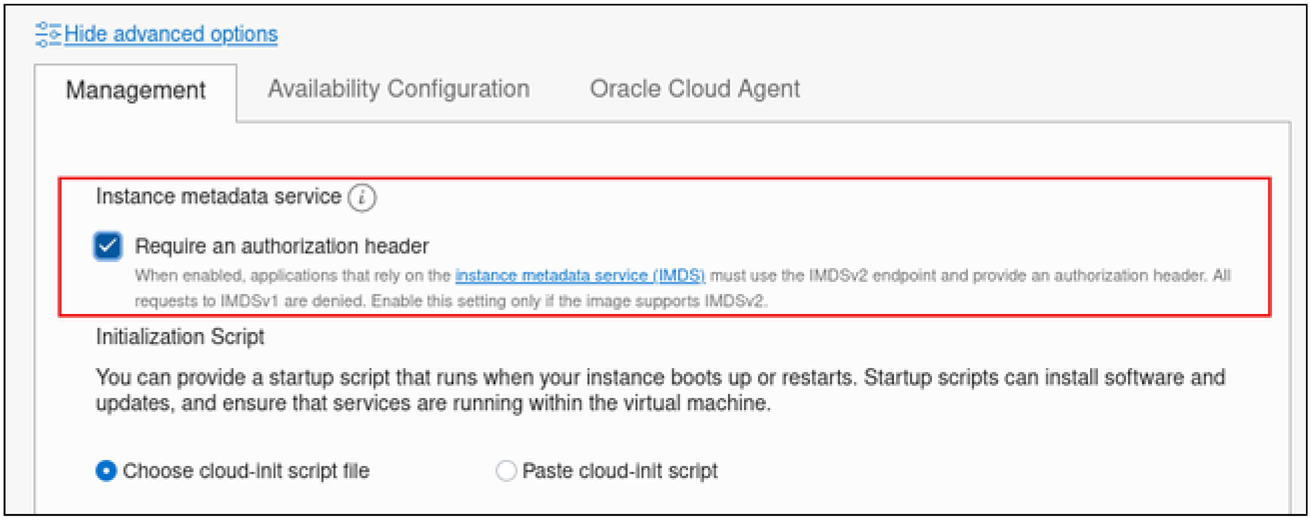
Your boot volume is a detachable device that contains the image used to boot your compute instance.

□ SPECIFY A CUSTOM BOOT VOLUME SIZE
Volume performance varies with volume size. Default boot volume size: 46.6 GB

□ USE IN-TRANSIT ENCRYPTION
Encrypts data in transit between the instance, the boot volume, and the block volumes.

□ ENCRYPT THIS VOLUME WITH A KEY THAT YOU MANAGE
By default, Oracle manages the keys that encrypt this volume, but you can choose a key from a vault that you have access to if you want greater control over the key's lifecycle and how it's used. Learn more about managing your own encryption keys

Step 7: support an IMDSv2 authorization header for applications relying on the IMDS service to improve security. For this, open the additional options by clicking on **Show Advanced Options** at the bottom of the instance creation page, select the **Management** tab, and activate the authorization header, as shown below:



Instance metadata service

☑ Require an authorization header

When enabled, applications that rely on the instance metadata service (IMDS) must use the IMDSv2 endpoint and provide an authorization header. All requests to IMDSv1 are denied. Enable this setting only if the image supports IMDSv2.

Initialization Script

You can provide a startup script that runs when your instance boots up or restarts. Startup scripts can install software and updates, and ensure that services are running within the virtual machine.

● Choose cloud-init script file ○ Paste cloud-init script

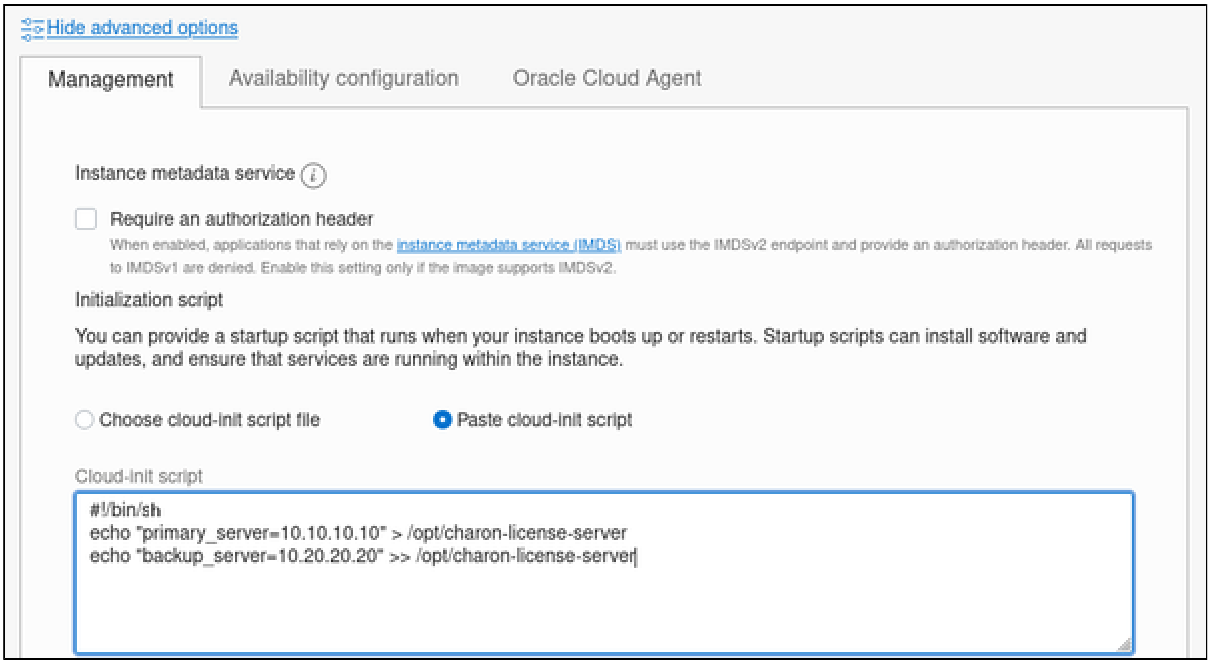
For Charon-SSP marketplace images, this is supported starting with Charon-SSP marketplace images version 4.2.2 and VE license server 1.0.33. On existing instances, this parameter can be changed, by editing the instance metadata service settings for the instance (go to **Instance Details** and click on **Edit** in the line **Instance Metadata Service**).
Only change the configuration to IMDSv2 if the image you launched the instance from supports it. Otherwise, you may not be able to connect to your instance. **Please note**: at the time of writing, the official CentOS 7 image on OCI did not support the new feature. If you create an instance to be used as a host for a manual VE license server or Charon VE installation, verify the capabilities of the image used before you enable the new IMDSv2 feature.

Step 8: Additional configuration for **AutoVE** setup.

If the instance is launched from a Charon AL marketplace image and is planned to use AutoVE licensing (instead of the public license servers), you must add the corresponding information to the instance configuration **before** the first launch of the instance:

For OCI, you have to enter a cloud-init script at instance configuration in the **Advanced Options** section.

The following image shows an example:



Instance metadata service

□ Require an authorization header

When enabled, applications that rely on the instance metadata service (IMDS) must use the IMDSv2 endpoint and provide an authorization header. All requests to IMDSv1 are denied. Enable this setting only if the image supports IMDSv2.

Initialization script

You can provide a startup script that runs when your instance boots up or restarts. Startup scripts can install software and updates, and ensure that services are running within the instance.

○ Choose cloud-init script file ● Paste cloud-init script

Cloud-init script

```
#!/bin/sh
echo "primary_server=10.10.10.10" > /opt/charon-license-server
echo "backup_server=10.20.20.20" >> /opt/charon-license-server
```

Valid User Data configuration options:

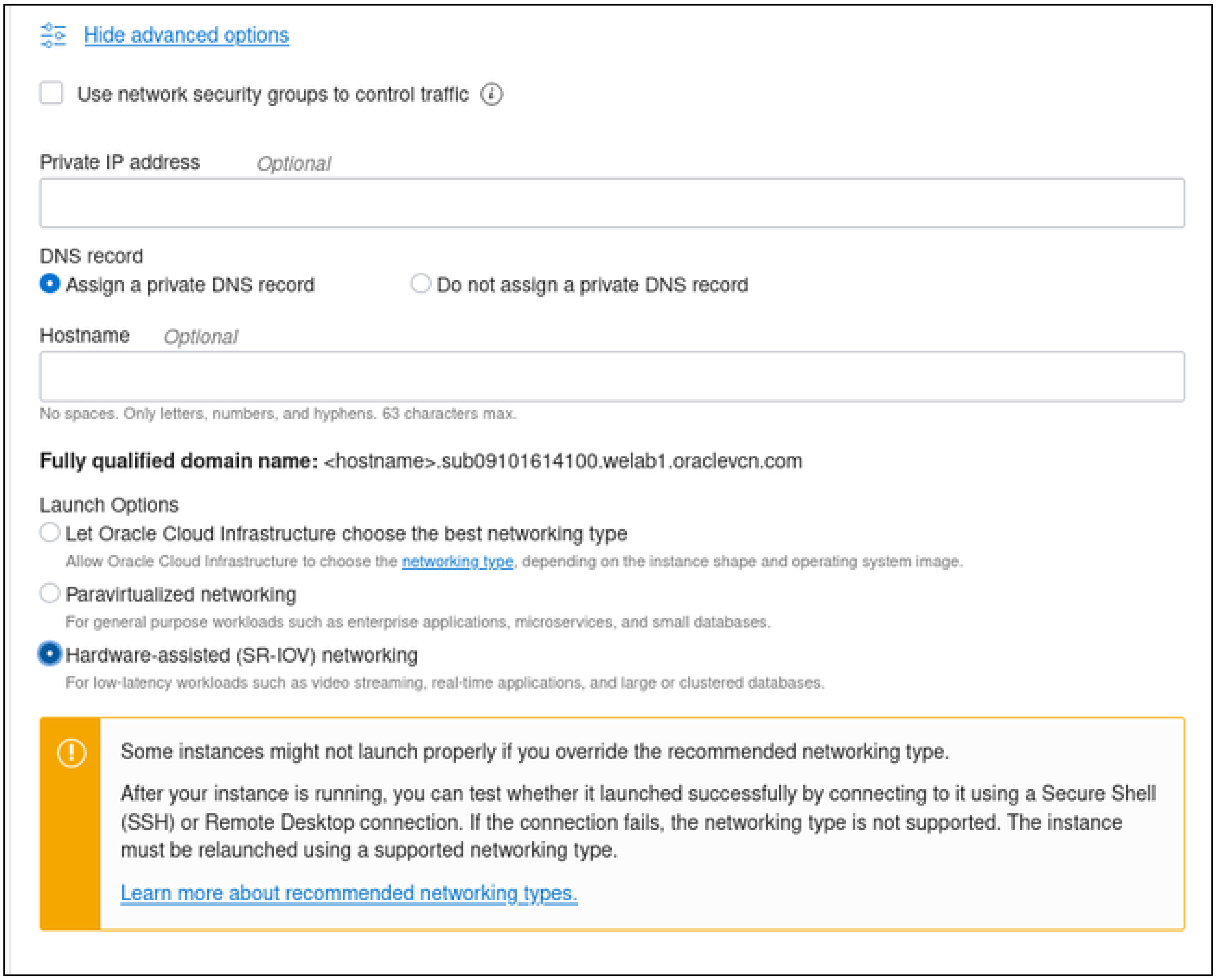
- `primary_server=<ip-address>[:<port>]`
- `backup_server=<ip-address>[:<port>]`

where

- *<ip-address>* stands for the IP address of the primary and the backup server as applicable, and
- *<port>* stands for a non-default TCP port used to communicate with the license server (default: TCP/8083).

Please note: at least one license server must be configured at initial launch to enable AutoVE mode. **Otherwise, the instance will bind to one of the public license servers operated by Stromasys.**

Step 9: The networking type selection may be required to allow offloading parameters to be disabled on an Ethernet interface dedicated to the emulator. For the Charon emulators, offloading parameters on the Ethernet interfaces it uses must be disabled. This is required for proper functionality and good performance of the emulator. To allow this configuration to be correctly reflected in the underlying cloud instance NICs for Charon-SSP versions before 4.1.32, the correct networking type (HARDWARE ASSISTED (SR-IOV) NETWORKING) must be chosen for the instance. For other emulator products, this is required if a dedicated interface is used by the emulator and there are problems with disabling offloading parameters. For this, open the additional options section by clicking on **Show Advanced Options** at the bottom of the network configuration section as shown below:



Hide advanced options

□ Use network security groups to control traffic

Private IP address *Optional*

DNS record

● Assign a private DNS record ○ Do not assign a private DNS record

Hostname *Optional*

No spaces. Only letters, numbers, and hyphens. 63 characters max.

Fully qualified domain name: <hostname>.sub09101614100.welab1.oraclevcn.com

Launch Options

○ Let Oracle Cloud Infrastructure choose the best networking type

Allow Oracle Cloud Infrastructure to choose the networking type, depending on the instance shape and operating system image.

○ Paravirtualized networking

For general purpose workloads such as enterprise applications, microservices, and small databases.

● Hardware-assisted (SR-IOV) networking

For low-latency workloads such as video streaming, real-time applications, and large or clustered databases.

Some instances might not launch properly if you override the recommended networking type.

After your instance is running, you can test whether it launched successfully by connecting to it using a Secure Shell (SSH) or Remote Desktop connection. If the connection fails, the networking type is not supported. The instance must be relaunched using a supported networking type.

Learn more about recommended networking types.

On this tab select **HARDWARE ASSISTED (SR-IOV) NETWORKING** (after creation, the instance will display the NIC Attachment Type VFIO). Please observe the warning displayed: **not all shapes support this type properly**.

Step 10: Click on **Create** at the bottom of the page to create your instance.

Step 11: verify your instance is running.

Your instance should now be visible in the list of compute instances.

Please note: It is recommended to create separate storage space (disk volumes) for Charon application data (e.g., disk images). If required, such volumes can later easily be migrated to another instance (see *Storage Management*).

Initial Access to the Instance

Once you have access to the instance, you can create the access you require for your applications. This section just shows the basic steps for initial access to the instance.

SSH Interactive Access

To connect to the instance interactively, you must connect as the management user of your instance. Use the following command:

```
$ ssh -o ServerAliveInterval=30 -i <path-to-your-private-key> <management-user-name>@<cloudhost-IP-address>
```

The parameter `ServerAliveInterval` will protect the connection from timing out.

Please note:

- Depending on the type of connection, you will have to use either the public IP address of the cloud system or its address in a customer-specific VPN.
- The **private key** used must correspond to the public key installed in the *authorized_keys* file of the cloud instance management user. This is usually done during initial cloud instance launch.
- The management user account normally allows sudo access to privileged commands (use **sudo -i**).
- If the instance was created using a Stromasys-provided AL or VE marketplace image, the management user for **interactive login** is the user **sshuser**.

File Transfer with SFTP

SFTP enables file transfers to and from the cloud instance. Use the management user of your instance. The security rules must allow SSH access to allow SFTP access to the cloud instance.

Please note: Depending on the type of connection, you will have to use either the public IP address of the cloud system or its address in a customer-specific VPN.

To connect to the instance, use the following command:

```
$ sftp -i <path-to-your-private-key> <management-user-name>@<cloudhost-IP-address>
```

Please note:

- Depending on the type of connection, you will have to use either the public IP address of the cloud system or its address in a customer-specific VPN.
- The **private key** used must correspond to the public key installed in the *authorized_keys* file of the cloud instance management user. This is usually done during initial cloud instance launch.
- If the instance was created using a Stromasys-provided AL or VE marketplace image, the management user for **file transfer** is the user **charon**.
- If the user **charon** is used to transfer files, the home directory for the file transfer will be */charon/storage.*