



# Charon-SSP 3.1.21 AWS User's Guide



# Contents

---

About this Guide	3
Introduction to Charon-SSP	5
Virtual Hardware and Guest OS Supported by Charon-SSP AWS	6
Charon-SSP Product Variant Comparison	7
Setting up a Charon-SSP AWS Cloud Instance	10
Charon-SSP Manager Installation	20
Accessing the Charon-SSP AWS Instance	22
SSH Command-Line Access	23
SFTP File Transfer	25
Connecting with the Charon-SSP Manager	26
Additional Charon-SSP AWS Instance Configuration	28
Storage Management	29
Network Management	35
Configuring and Managing the System Using the Charon-SSP Manager	39
Starting the Charon Manager	40
Creating a Virtual Machine	45
Configuring a Virtual Machine	46
Hardware Family (Model) Configuration	47
CPU Configuration	48
DIT Configuration	50
Memory Configuration	53
Graphics Configuration	54
SCSI Storage Configuration	58
Serial Line Configuration	64
Audio Configuration	69
Ethernet Configuration	72
NVRAM Configuration	74
Log Configuration	75
Virtual Machine Context Menu	77
Host System Network Configuration	79
Miscellaneous Management Tasks	86
AWS Cloud Tools	89
Graphical Interface via X11 Server on Linux	92
Starting, Stopping, and Suspending the Emulated System	97
User Access to the Virtual SPARC System	100
AWS Networking and Charon-SSP	105
SSH VPN - Connecting Charon Host and Guest to Customer Network	110
Example of a More Complex Network Configuration	116
Data Transfer Options	121
Firewall and AWS Security Group Considerations	123
Upgrading Charon-SSP AWS	124
Charon-SSP Software Deinstallation	128
OpenBoot Console	129

# About this Guide

## Contents

- Intended Audience
- Document Structure
- Obtaining documentation
- Obtaining technical assistance
- Throughout the document(s) these conventions are followed
- The following definitions apply
- Related Documents

## Intended Audience

This user's guide is intended for anyone who needs to install, configure, or manage the Stromasys Charon-SSP processor/platform virtualization software. The content of this manual is targeted at general users (not just system managers and administrators). However, a general working knowledge of PC operating systems and their conventions is expected.

The user's guide covers the **Charon-SSP AWS EC2 AMI** distribution. This appliance package contains the full software set including the underlying Linux host operating system. Interactive host operating system access is limited to a subset of commands. If you need to install the non-cloud version of Charon-SSP please refer to the Charon-SSP User's Guides for these product variants.

## Document Structure

The document contains the following main sections:

- [Virtual Hardware and Guest OS Supported by Charon-SSP AWS](#): list of supported virtual hardware and supported guest operating systems.
- [Charon-SSP Product Variant Comparison](#): overview of differences between the different Charon-SSP products (conventional, Baremetal, AWS AMI).
- [Setting up a Charon-SSP AWS Cloud Instance](#): basic steps to create and launch a Charon-SSP AWS EC2 instance.
- [Charon-SSP Manager Installation](#): the Charon-SSP Manager must be installed on a local system to access and manage the Charon-SSP AWS EC2 host and the emulated systems running on it.
- [Accessing the Charon-SSP AWS Instance](#): explains how to use SSH, SFTP, and the Charon-SSP Manager to access the instance, and how to set the initial management password.
- [Additional Charon-SSP AWS Instance Configuration](#): steps to add additional storage and network interfaces.
- [Configuring and Managing the System Using the Charon-SSP Manager](#): shows how to configure and run emulated SPARC systems.
- [User Access to the Virtual SPARC System](#): shows the different methods of how a user can access the console of the emulated SPARC system and the guest operating system.
- [AWS Networking and Charon-SSP](#): overview of networking functionality that is specific to the AWS environment, and detailed steps to configure two specific network configurations.
  - [SSH VPN - Connecting Charon Host and Guest to Customer Network](#): describes how to create a VPN tunnel between a remote Linux system and the Charon-SSP AWS instance to enable communication between host, guest, and customer network.
  - [Example of a More Complex Network Configuration](#): example involving two Charon-SSP AWS instances with one acting as router and NAT gateway.
- [Data Transfer Options](#): information about how data can be transferred to/from the host and guest.
- [Firewall and AWS Security Group Considerations](#): information about ports used by the different applications.
- [Upgrading Charon-SSP AWS](#): ways to upgrade Charon-SSP AWS to a higher version.
- [OpenBoot Console](#): OpenBoot console command overview

## 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

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](mailto:support@stromasys.com).

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

For further information on purchases and the product best suited to your requirements, please contact your regional sales team:

Region	Email address	Phone	Address
Australasia-Pacific	<a href="mailto:apac.sales@stromasys.com">apac.sales@stromasys.com</a>	+852 3520 1030	Room 1113, 11/F, Leighton Centre 77 Leighton Road, Causeway Bay, Hong Kong, China
Americas	<a href="mailto:ams.sales@stromasys.com">ams.sales@stromasys.com</a>	+1 919 239 8450	2840 Plaza Place, Ste 450 Raleigh, NC 27612 U.S.A.
Europe, Middle-East and Africa	<a href="mailto:emea.sales@stromasys.com">emea.sales@stromasys.com</a>	+41 22 794 1070	Avenue Louis-Casai 84 5th Floor 1216 Cointrin Switzerland

## Throughout the document(s) these conventions are followed

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).
<b>User input</b>	Bold monospace type in interactive examples indicates typed user input.
<b>&lt;path&gt;</b>	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.
<i>dsk0</i>	Italic monospace type, in interactive examples, indicates typed context dependent user input.

## The following definitions apply

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-SSP V3.0.2 for Linux](#)
- [CHARON-SSP V2.0.5 for Linux](#)
- [CHARON-SSP V1.4.1 for Linux](#)
- [CHARON-SSP V1.0.36 for Linux](#)
- [CHARON-SSP V1.0.34 for Linux](#)
- [CHARON-SSP V1.0.26 for Linux](#)

## Introduction to Charon-SSP

In 1987, Sun Microsystems released the SPARC V7 processor, a 32-bit RISC processor. The SPARC V8 followed in 1990 – a revision of the original SPARC V7, with the most notable inclusion of hardware divide and multiply instructions. The SPARC V8 processors formed the basis for a number of servers and workstations such as the SPARCstation 5, 10 and 20. In 1993, the SPARC V8 was followed by the 64-bit SPARC V9 processor. This too became the basis for a number of servers and workstations, such as the Enterprise 250 and 450.

Due to hardware obsolescence and lack of spare or refurbished parts, software and systems developed for these older SPARC-based workstations and servers have become harder to maintain. To fill the continuous need for certain, end-of-life SPARC-based systems, Stromasys S.A. developed the Charon-SSP line of virtual machine products. The following products are software-based, virtual machine replacements for the specified native-hardware SPARC systems. A general overview of the emulated hardware families is shown below:

**Charon-SSP/4M** emulates the following SPARC hardware:

- **Sun-4m family (for example Sun SPARCstation 20):** Originally, a multiprocessor Sun-4 variant, based on the MBus processor module bus introduced in the SPARCServer 600MP series. The Sun-4m architecture later also encompassed non-MBus uniprocessor systems such as the SPARCstation 5, utilizing SPARC V8-architecture processors. Supported starting with SunOS 4.1.2 and by Solaris 2.1 to Solaris 9. SPARCServer 600MP support was dropped after Solaris 2.5.1.

**Charon-SSP/4U(+)** emulates the following SPARC hardware:

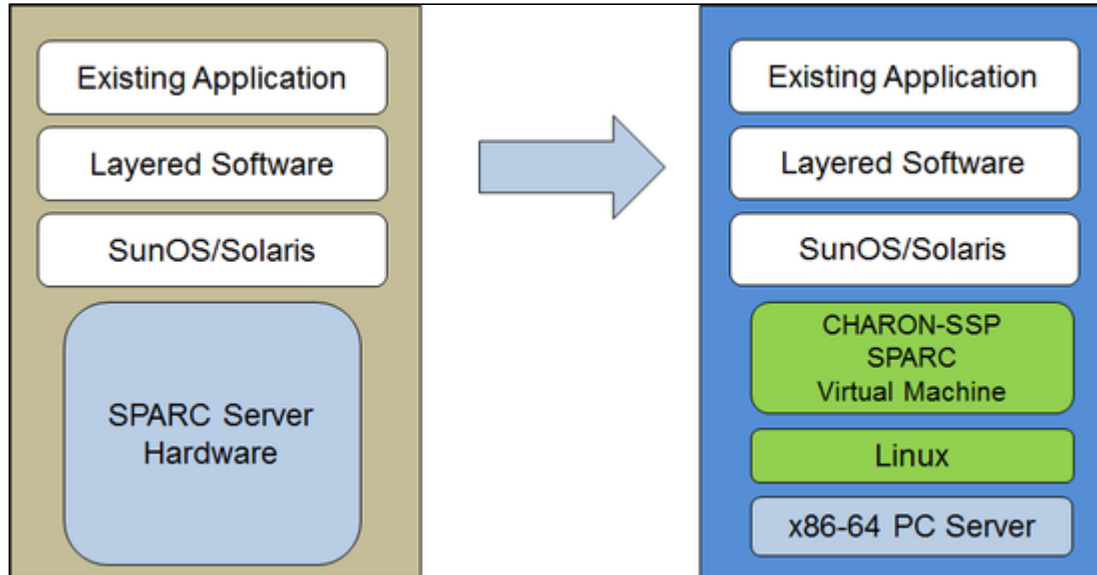
- **Sun-4u family (for example Sun Enterprise 450):** (U for UltraSPARC) – this variant introduced the 64-bit SPARC V9 processor architecture and UPA processor interconnect first used in the Sun Ultra series. Supported by 32-bit versions of Solaris starting from version 2.5.1. The first 64-bit Solaris release for Sun-4u was Solaris 7. UltraSPARC I support was dropped after Solaris 9. Solaris 10 supports Sun-4u implementations from UltraSPARC II to UltraSPARC IV.

**Charon-SSP/4V(+)** emulates the following SPARC hardware:

- **Sun-4v family (for example SPARC T2):** A variation on Sun-4u which includes hypervisor processor virtualization; introduced in the UltraSPARC T1 multicore processor. Selected hardware was supported by Solaris version 10 starting from release 3/05 HW2 (most models - including the hardware emulated by Charon-SSP - require newer versions of Solaris 10). Several Solaris 11 versions are also supported.

⚠ For up-to-date information about supported features and versions refer to [Virtual Hardware and Guest OS Supported by Charon-SSP AWS](#) and to the release notes of your product.

The image below shows the basic concept of migrating physical hardware to an emulator:



The Charon-SSP virtual machines allow users of Sun and Oracle SPARC-based computers to replace their native hardware in a way that requires little or no change to the original system configuration. This means you can continue to run your applications and data without the need to switch or port to another platform. The Charon-SSP software runs on commodity, Intel 64-bit systems ensuring the continued protection of your investment.

**Charon-SSP/4U+** supports the same virtual SPARC platforms as Charon-SSP/4U, and **Charon-SSP/4V+** the same as Charon-SSP/4V. However, the 4U+ and 4V+ versions take advantage of Intel's VT-x/EPT hardware assisted virtualization technology in modern Intel CPUs to offer end users better virtual CPU performance. Charon-SSP/4U+ and Charon-SSP/4V+ require Intel CPUs with VT-x/EPT capability and **must** be installed on a dedicated Intel-based host. Running these product variants in a VM is **not supported**.

**i** If you consider running Charon-SSP/4U+ or 4V+ in AWS to achieve better performance, please contact Stromasys or a Stromasys VAR.

**i** Unless otherwise mentioned, the terms Charon-SSP/4U and Charon-SSP/4V also include Charon-SSP/4U+ and Charon-SSP/4V+.

# Virtual Hardware and Guest OS Supported by Charon-SSP AWS

## Contents

- Supported Virtual Hardware
- Supported Guest Operating Systems

## Supported Virtual Hardware

The different families of Charon-SSP virtual machines support a number of different hardware devices. The table below describes the device features and maximum number supported by the different Charon-SSP virtual machine families.

Charon-SSP AWS Supported Virtual Hardware			
	Charon-SSP/4M	Charon-SSP/4U(+) <sup>(1)</sup>	Charon-SSP/4V(+) <sup>(1)</sup>
<b>SPARC V8 (32-bit)</b>	Y		
<b>SPARC V9 (64-bit)</b>		Y <sup>(2)</sup>	Y <sup>(4)</sup>
<b>Max. number of CPUs</b>	4	24	64
<b>Max. RAM</b>	64MB to 512MB	1GB to 128GB	1GB to 1024GB <sup>(5)</sup>
<b>Ethernet controllers</b>	2 (controller type le)	19 (controller types hme and qfe)	4 (controller types bge and qfe)
<b>SCSI controllers</b>	1	2	2
<b>SCSI target IDs</b>	7 <sup>(3)</sup>	30 <sup>(3)</sup>	30 <sup>(3)</sup>
<b>Serial ports</b>	2	2	2 + Vconsole
<b>Graphics controllers</b>	1 (CGTHREE or CGSIX)	1 (CGSIX or RAGE XL)	
<b>Audio controllers</b>	1 (DBRle)	1 (DBRle)	

<sup>(1)</sup> Charon-SSP/4U+ has same virtual hardware specification as Charon-SSP/4U, Charon-SSP/4V+ the same als Charon-SSP/4V.

<sup>(2)</sup> SPARC V9 is backward compatible. Hence, Charon-SSP/4U can also support V8 32-bit systems.

<sup>(3)</sup> Each SCSI target ID can have up to 8 LUNs. Therefore, the overall number of SCSI devices can be larger than the number of target IDs. The exact number depends on the emulated hardware, the guest operating system version, and the SCSI devices used.

<sup>(4)</sup> Charon-SSP/4V supports one LDom per instance. An LDom virtual disk image can be booted by Charon-SSP without modifications.

<sup>(5)</sup> Actual maximum values are different depending on guest OS: Solaris 10: 1TB, Solaris 11: 512GB.

## Supported Guest Operating Systems

The Charon-SSP/4M virtual machines support the following guest operating system releases:

- SunOS 4.1.3 - 4.1.4
- Solaris 2.3 to Solaris 9

The Charon-SSP/4U(+) virtual machines support the following guest operating system releases:

- Solaris 2.5.1 to Solaris 10

The Charon-SSP/4V(+) virtual machines support the following guest operating system releases:

- Solaris 10 (starting with update 4, 08/07) and Solaris 11.1 to Solaris 11.3

# Charon-SSP Product Variant Comparison

When looking at the Charon-SSP product features, one can look at the differences between the **different emulated models** (as shown in [Virtual Hardware and Guest OS Supported by Charon-SSP AWS](#)).

Another comparison is the comparison between the **different product variants**. This section provides an overview of important differences between the product variants.

## Contents

- [Product Variant Overview](#)
- [Product Variant Comparison](#)

## Product Variant Overview

The basic functionality of Charon-SSP in the different product variants is very similar. However, the product variants also have important differences.

Currently available Charon-SSP product variants:

- Conventional product
  - Individual RPM installation
  - Barebone installation
- Baremetal version
- Amazon AWS version

### Conventional product

This product variant exists in two flavors:

- The product is installed as individual RPM packages on a supported Linux distribution and version.
- The product is delivered as an ISO installation file that contains the operating system, the product RPMs and any additional software typically required (Barebone version).

The conventional product, especially when installed as individual RPM packages, offers the greatest flexibility for any customization and for integration into the customers' system management environments.

The Barebone version offers additional features, but is more restrictive in what changes a user can make to the operating system.

### Baremetal product

Charon-SSP Baremetal is a software appliance distributed as an ISO installation file. The host operating system is encapsulated and not visible to the user.

A customized GUI offers the necessary system management tools.

The Baremetal variant provides a fast and easy way to set up Charon-SSP if no major customization and integration requirements exist.

### Amazon AWS version

Charon-SSP AWS provides an AMI image on Amazon Marketplace that can be used to easily launch a Charon-SSP host containing all the necessary software as an AWS instance.

Licensing is set up automatically at launch and usage is billed through AWS.

The instance configuration (e.g., network configuration) can be adapted to the customer's requirement, but only a subset of a full Linux host operating system is accessible.



## Product Variant Comparison

The following table lists important differences between the Charon-SSP product variants (as opposed to the differences between the different emulated architectures):

Functionality differences	Conventional product		Baremetal	AWS image
	RPM	Barebone		
<b>General differences</b>				
User shell access to host OS		Y	N	Y (limited)
Linux operating system upgrades from distribution repositories	Y	Restricted (kernel version dependencies)	N	N (package installation is possible if agreed by Stromasys)
Special GUI for host management		N	Y	N
Special user accounts for Charon		N <sup>(4)</sup>	Y <sup>(5)</sup>	Y <sup>(6)</sup>
Licensing general	HL/SL/Network license			Cloud license server
Changes to number of host CPUs / amount of memory possible		Y (if software license is used, new license may be needed)		N (invalidates license, requires setup of new AWS instance)
Internet connection required		N		Y
Jumpstart		Y		N
Network interface sharing (not recommended)		Y		N
Configurable log path		Y	N	N
Additional tools	X11, iSCSI, NFS	X11, iSCSI, NFS, VNC	X11, iSCSI, NFS, VNC	X11
<b>Emulated HW differences</b>				
4U+ and 4V+ support	N	Y	Y	Y
PCI pass-through devices (Digi and GPIB)	N	Y <sup>(2)</sup>	Y <sup>(2)</sup>	N
Digi AccelePort emulation		Y <sup>(2)</sup>		N
Additional on-board serial lines		Y		N
USB devices		Y <sup>(2)</sup>		N
Parallel port		Y <sup>(3)</sup>		N
Floppy drive		Y <sup>(3)</sup>		N
Physical SCSI devices		disk, tape, CD-ROM, generic		disk
External serial console via TCP		Y		N
Physical serial ports		Y		Only via terminal server
<b>Host HW differences</b>				
Customer selectable hypervisor support		Y <sup>(1)</sup>		N <sup>(7)</sup>

## **Notes**

- (1) Not for Charon-SSPU/4U+/4V+ (require VT-x/EPT support); supported Hypervisors are listed in *Host System Requirements*.
- (2) Not on Charon-SSP/4M
- (3) Charon-SSP/4M only
- (4) A **charon** user is created on Barebone systems during installation. However, normal Linux account management is possible. Interactive command-line access not restricted.
- (5) User **charon** for GUI operation, SFTP and VNC access, and Charon Manager integrated SSH tunnel. User **sshuser** for setting up the general SSH VPN tunnel. No interactive command-line access.
- (6) User **charon** for GUI operation, SFTP access, and Charon Manager integrated SSH tunnel. User **sshuser** for setting up the general SSH VPN tunnel, and for interactive command-line access (limited command set); **root** access possible.
- (7) Normal AWS instance types run on shared hardware; "metal" instance types are required for Charon-SSP/4U+/4V+.

# Setting up a Charon-SSP AWS Cloud Instance

This chapter describes how to set up a basic Charon-SSP instance in AWS.

## Contents

- Prerequisites
  - General Prerequisites
  - Licensing
  - AWS Instance Type Prerequisites (Hardware Prerequisites)
- AWS Login and New Instance Launch
- New Instance Configuration

## Prerequisites

### General Prerequisites

To access and use Charon-SSP AWS, you need an Amazon AWS account.

Please note the following details about the different AWS environments:

With EC2-Classic, your instances run in a single, flat network that you share with other customers. With Amazon VPC, your instances run in a virtual private cloud (VPC) that is logically isolated from other AWS accounts. The EC2-Classic platform was introduced in the original release of Amazon EC2. If you created your AWS account after 4 December 2013, it does not support EC2-Classic, so you must launch your Amazon EC2 instances in a VPC. If your account does not support EC2-Classic, Amazon AWS will create a default VPC. By default, when you launch an instance, it is launched in your default VPC. Alternatively, you can create a non-default VPC and specify it when you launch an instance.


For more information about the specifics of each environment, please refer to the documentation on the Amazon web page: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-classic-platform.html>.

### Licensing

Charon-SSP AWS requires a license to run emulated SPARC systems. This license is created automatically upon first launch of the Charon-SSP AWS instance. Please note the following points:

- The Charon-SSP AWS instance requires Internet access (via public IP address or NAT) for the license mechanism to work.
- If you change the instance type after first launching the instance and thereby change important hardware characteristics (e.g., number of CPU cores, memory size), the license will be invalidated.
- If you need to change important hardware characteristics (e.g., number of CPU cores, memory size), you need to create a new instance. Therefore it is strongly recommended to store all relevant data on a separate EBS volume that can easily be detached from the old instance and attached to a new instance.
- Should access to the license be lost, there is a grace period of 24 hours. If license access is not restored within this period, the emulator will stop (if a guest system is running at the time, this is the equivalent of disconnecting the power without clean shutdown, i.e., it may lead to loss of data).

 You will be billed by Amazon for your use of the Charon-SSP AWS instance. Stromasys will not bill you directly.

 The user is responsible for any Solaris licensing obligations and has to provide the appropriate licenses.

## AWS Instance Type Prerequisites (Hardware Prerequisites)


By selecting an instance type in AWS, you select the virtual hardware that will be used for Charon-SSP AWS. Therefore, the selection of an instance type determines the hardware characteristics of the Charon-SSP virtual host hardware (e.g., how many CPU cores and how much memory your virtual Charon host system will have).

### Important information:

Please make sure to dimension your instance correctly from the beginning (check the minimum requirements below). The Charon-SSP license is created when the instance is first launched. Changing later to another instance type with a different number of CPU cores, different memory size, or other significant differences in hardware characteristics **will invalidate the license** and thus prevent Charon instances from starting.

### Minimum requirements for Charon-SSP:

- Minimum number of host system CPU cores:
  - At least one CPU core for the host operating system.
  - **For each emulated SPARC system:**
    - One CPU core for each emulated CPU of the instance.
    - At least one additional CPU core for I/O. If server JIT optimization is used, add an additional I/O CPU for improved translation speed.
- Minimum memory requirements:
  - At least 2GB of RAM for the host operating system.
  - **For each emulated SPARC system:**
    - The configured memory of the instance.
    - 2GB of RAM (6GB of RAM if server JIT is used) to allow for DIT optimization, emulator requirements, run-time buffers, SMP and graphics emulation.
- One or more network interfaces, depending on customer requirements. The network performance level of an instance type provides an indication of the data transfer rates to be expected from the AWS instance.
- Charon-SSP/4U+ and Charon-SSP/4V+ must run on hardware supporting VT-x. For this, you need to select an instance with the suffix **metal** in the name. If you consider running Charon-SSP/4U+ or 4V+ in AWS to achieve better performance, please contact Stromasys or a Stromasys VAR for additional information.

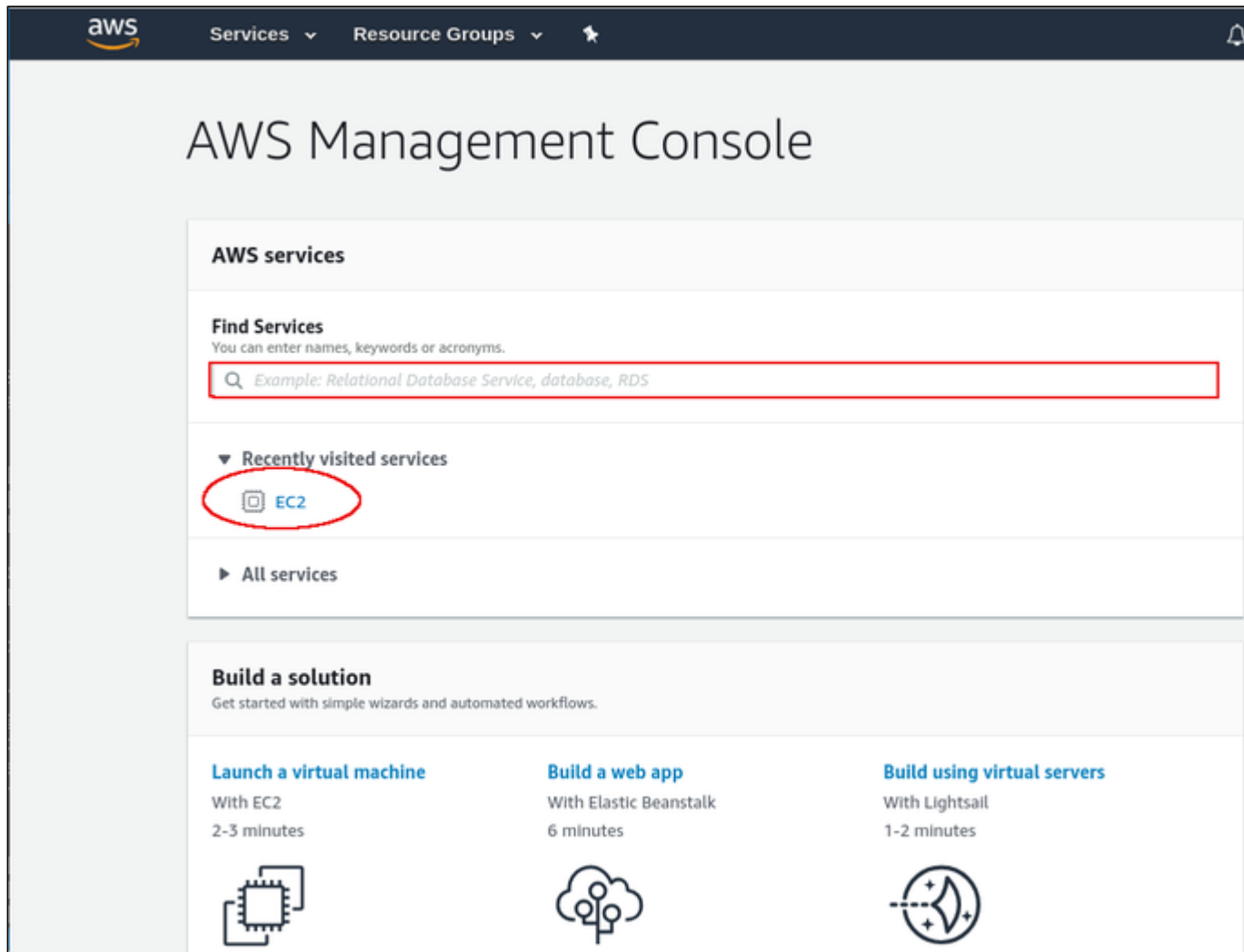
 Please note that the sizing guidelines above—in particular regarding number of host CPU cores and host memory—show the minimum requirements. Every use case has to be reviewed and the actual host sizing has to be adapted as necessary. For example, the number of I/O CPUs may have to be increased if the guest applications produce a high I/O load. Also take into consideration that a system with many emulated CPUs in general is also able to create a higher I/O load and thus the number of I/O CPUs may have to be raised.

## AWS Login and New Instance Launch

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications such as Charon-SSP.

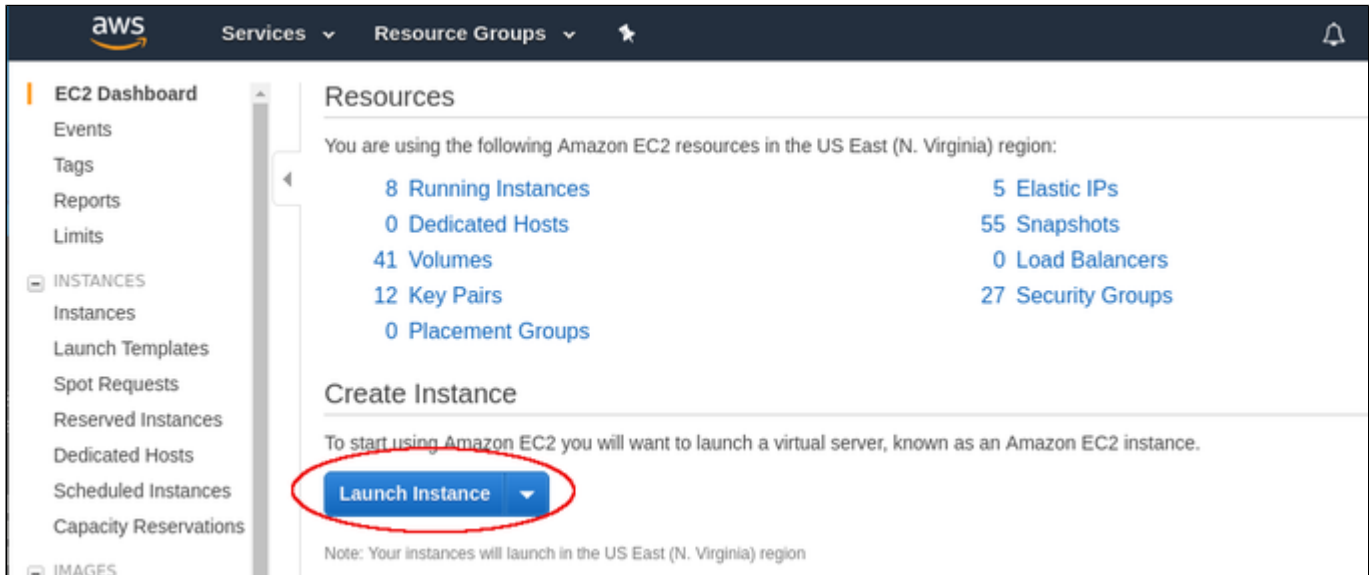
To start the creation of a new cloud instance using the Charon-SSP AMI, perform the following steps:

1. **Log in** to your AWS management console.
2. Find and select the **EC2 service**. You can use the search window or find it in the recently used services.



This will open the E2C dashboard.

3. On the EC2 dashboard click on the **Launch Instance** button.



The screenshot shows the AWS Management Console EC2 dashboard. The left-hand navigation pane includes sections for 'EC2 Dashboard' (Events, Tags, Reports, Limits), 'INSTANCES' (Instances, Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), and 'IMAGES'. The main content area is titled 'Resources' and lists EC2 resources in the US East (N. Virginia) region: 8 Running Instances, 0 Dedicated Hosts, 41 Volumes, 12 Key Pairs, 0 Placement Groups, 5 Elastic IPs, 55 Snapshots, 0 Load Balancers, and 27 Security Groups. Below this is the 'Create Instance' section, which contains the text 'To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.' and a blue 'Launch Instance' button with a dropdown arrow, which is circled in red. A note at the bottom states: 'Note: Your instances will launch in the US East (N. Virginia) region'.

This will initiate the instance creation process consisting of seven steps:

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Groups
7. Review, launch and select/create key-pair for access.

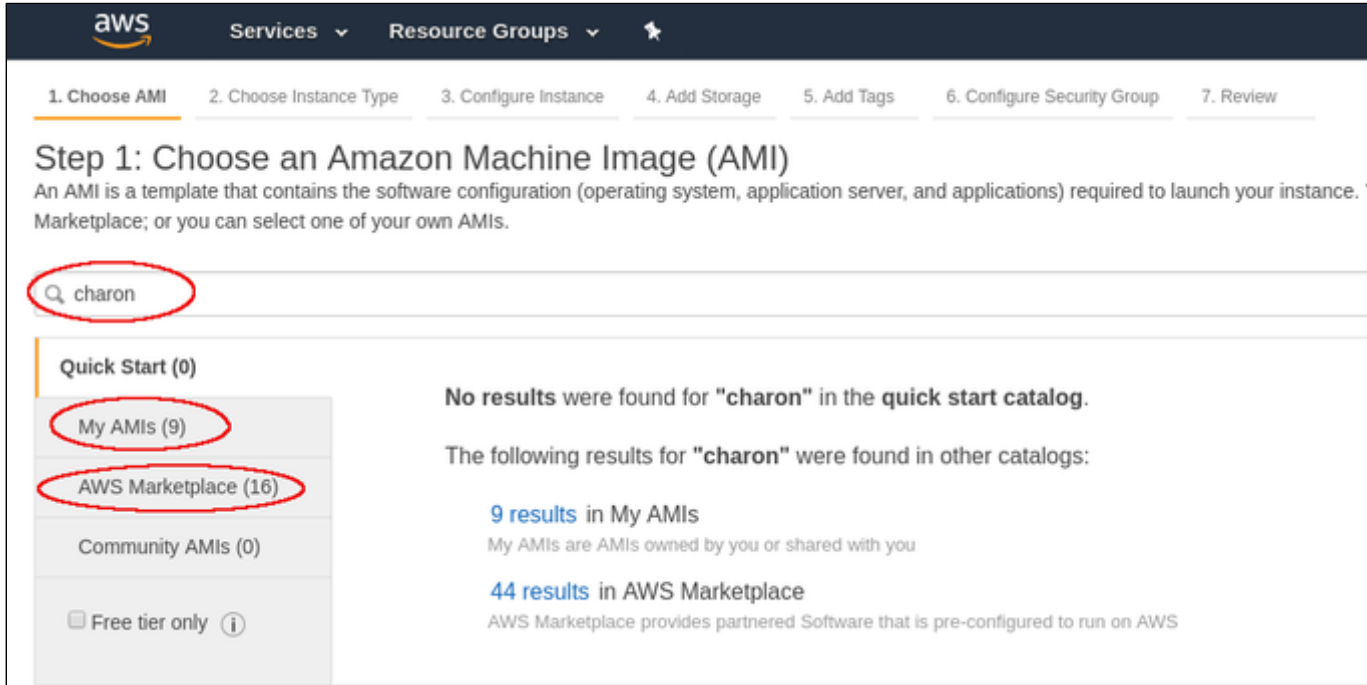
These steps are described in the next section.

## 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. Choose AMI:

Search for Charon products and select the desired Charon products from **Marketplace** or (depending on your environment) from My AMIs.



The screenshot shows the AWS console interface for selecting an Amazon Machine Image (AMI). At the top, there's a navigation bar with 'aws' logo, 'Services', and 'Resource Groups'. Below that, a progress bar indicates the current step: '1. Choose AMI', followed by '2. Choose Instance Type', '3. Configure Instance', '4. Add Storage', '5. Add Tags', '6. Configure Security Group', and '7. Review'. The main heading is 'Step 1: Choose an Amazon Machine Image (AMI)'. Below the heading, a brief explanation states: 'An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Marketplace; or you can select one of your own AMIs.' A search bar contains the text 'charon'. On the left, a sidebar lists categories: 'Quick Start (0)', 'My AMIs (9)', 'AWS Marketplace (16)', and 'Community AMIs (0)'. The main content area displays the search results: 'No results were found for "charon" in the quick start catalog.' and 'The following results for "charon" were found in other catalogs: 9 results in My AMIs' and '44 results in AWS Marketplace'. The 'My AMIs' section includes the text 'My AMIs are AMIs owned by you or shared with you'. The 'AWS Marketplace' section includes the text 'AWS Marketplace provides partnered Software that is pre-configured to run on AWS'.

Clicking on one of the categories above will display a list of images. Select the appropriate Charon-SSP AMI.

This will take you to the next step, the instance type selection.

## 2. 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-SSP product.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 2: Choose an Instance Type

<input type="radio"/>	Compute optimized	c5.large	2	4	EBS only	Yes	Up to 10 Gigabit	Yes
<input checked="" type="radio"/>	Compute optimized	c5.xlarge	4	8	EBS only	Yes	Up to 10 Gigabit	Yes
<input type="radio"/>	Compute optimized	c5.2xlarge	8	16	EBS only	Yes	Up to 10 Gigabit	Yes
<input type="radio"/>	Compute optimized	c5.4xlarge	16	32	EBS only	Yes	Up to 10 Gigabit	Yes
<input type="radio"/>	Compute optimized	c5.9xlarge	36	72	EBS only	Yes	10 Gigabit	Yes
<input type="radio"/>	Compute optimized	c5.12xlarge	48	96	EBS only	Yes	12 Gigabit	Yes
<input type="radio"/>	Compute optimized	c5.18xlarge	72	144	EBS only	Yes	25 Gigabit	Yes
<input type="radio"/>	Compute optimized	c5.24xlarge	96	192	EBS only	Yes	25 Gigabit	Yes
<input type="radio"/>	Compute optimized	c5.metal	96	192	EBS only	Yes	25 Gigabit	Yes
<input type="radio"/>	Compute optimized	c4.large	2	3.75	EBS only	Yes	Moderate	Yes
<input type="radio"/>	Compute optimized	c4.xlarge	4	7.5	EBS only	Yes	High	Yes
<input type="radio"/>	Compute optimized	c4.2xlarge	8	15	EBS only	Yes	High	Yes
<input type="radio"/>	Compute optimized	c4.4xlarge	16	30	EBS only	Yes	High	Yes
<input type="radio"/>	Compute optimized	c4.8xlarge	36	60	EBS only	Yes	10 Gigabit	Yes
<input type="radio"/>	FPGA instances	f1.2xlarge	8	122	1 x 470 (SSD)	Yes	Up to 10 Gigabit	Yes
<input type="radio"/>	FPGA instances	f1.4xlarge	16	244	1 x 940 (SSD)	Yes	Up to 10 Gigabit	Yes

Cancel Previous Review and Launch **Next: Configure Instance Details**

When done, continue by clicking on the **Next: Configure Instance** button.



### 3. Configure Instance:

In this section, you can set up the details of your instance configuration.

For example, you can select the VPC **subnet** your instance should be in and whether an interface should automatically be assigned a **public IP address**.

**i** Automatic assignment of a public IP address only works if there is only one network interface attached to the instance.

The screenshot shows the AWS Management Console interface for configuring an EC2 instance. The page is titled "Step 3: Configure Instance Details" and includes a progress bar at the top with steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance (active), 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. The main content area contains several configuration sections:

- Number of instances:** Set to 1, with a "Launch into Auto Scaling Group" link.
- Purchasing option:** "Request Spot instances" is unchecked.
- Network:** Set to "vpc".
- Subnet:** Set to "subnet-c691e89a | Default in us-east-1c" (4077 IP Addresses available).
- Auto-assign Public IP:** Set to "Use subnet setting (Enable)".
- Placement group:** "Add instance to placement group" is unchecked.
- Capacity Reservation:** Set to "Open".
- IAM role:** Set to "None".
- Shutdown behavior:** Set to "Stop".
- Enable termination protection:** Unchecked.
- Monitoring:** "Enable CloudWatch detailed monitoring" is unchecked.
- Tenancy:** Set to "Shared - Run a shared hardware instance".
- Elastic Inference:** "Add an Elastic Inference accelerator" is unchecked.
- T2/T3 Unlimited:** "Enable" is unchecked.

At the bottom right, there are four buttons: "Cancel", "Previous", "Review and Launch", and "Next: Add Storage". The "Next: Add Storage" button is circled in red.

Once you have selected all desired configuration options, click on **Next: Add storage** to continue.

#### 4. Add Storage:

The size of the root volume must be at least 20GB for the Charon-SSP host system to start. You can add more storage later to provide space for virtual disk containers and other storage requirements.

**!** It is recommended to create separate storage space (using AWS EBS volumes) for Charon application data (e.g., disk images). If required, such volumes can later easily be migrated to another instance (see [Storage Management](#)).

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0fa9b5b198575cd35	20	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Once you are done, again click on the **Next: Add tags** button.

#### 5. Add Tags:

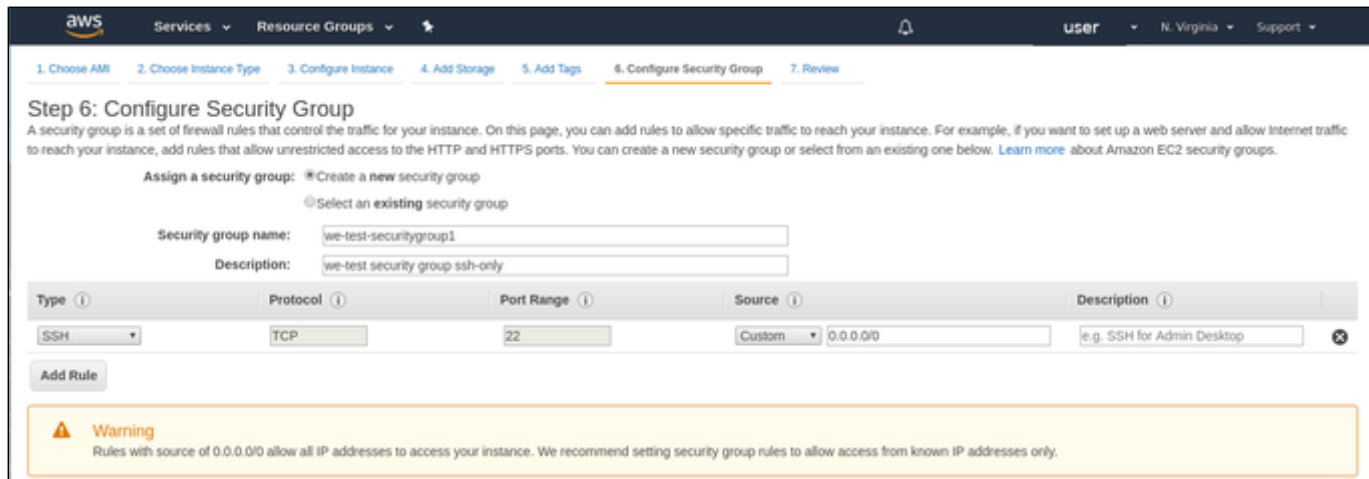
Tags allow you to add information to your instance, for example, an easily remembered name as shown in the example below:

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes
Name	we-test2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

After adding tags as required, continue to the next step (**Configure Security Groups**).

## 6. Configure Security Groups:

A security group is similar to a firewall. It defines which traffic is allowed to flow to and from the instance. For Charon-SSP you must at least enable SSH access to the system. This will allow you to access the management interface and to run Charon-SSP services via an SSH VPN tunnel. You can select an existing group or create a new one. If you create a new one, you can enter a name and an appropriate description. An example of a security group is shown below.



**Step 6: Configure Security Group**

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:  Create a new security group  Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

[Add Rule](#)

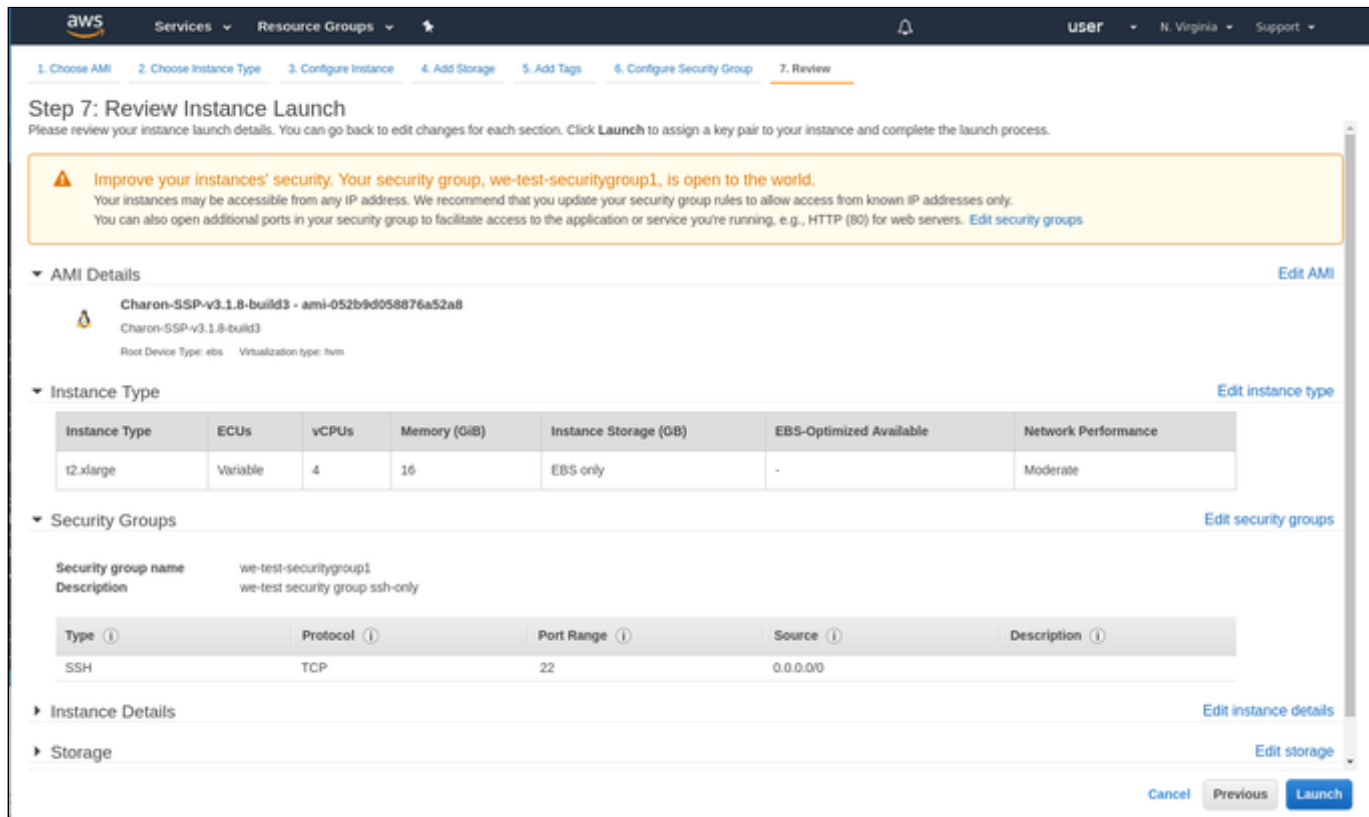
**Warning**  
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

The warning shown alerts the user to the fact that the source IP addresses are not restricted, i.e., any system is allowed to use SSH to access the instance. Restrict the source address range if possible. See also [Firewall and AWS Security Group Considerations](#).

Once you have set up your security group, continue to the next step (**Review and Launch**).

## 7. Review:

Here you can review the configuration of your instance and edit the individual sections if required. The image below shows a sample:



**Step 7: Review Instance Launch**

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**Warning**  
Improve your instances' security. Your security group, we-test-securitygroup1, is open to the world. Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

**AMI Details** [Edit AMI](#)

Charon-SSP-v3.1.8-build3 - ami-052b9d058876a52a8  
Charon-SSP-v3.1.8-build3  
Root Device Type: ebs Virtualization type: hvm

**Instance Type** [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.xlarge	Variable	4	16	EBS only	-	Moderate

**Security Groups** [Edit security groups](#)

Security group name: we-test-securitygroup1  
Description: we-test security group ssh-only

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	

**Instance Details** [Edit instance details](#)

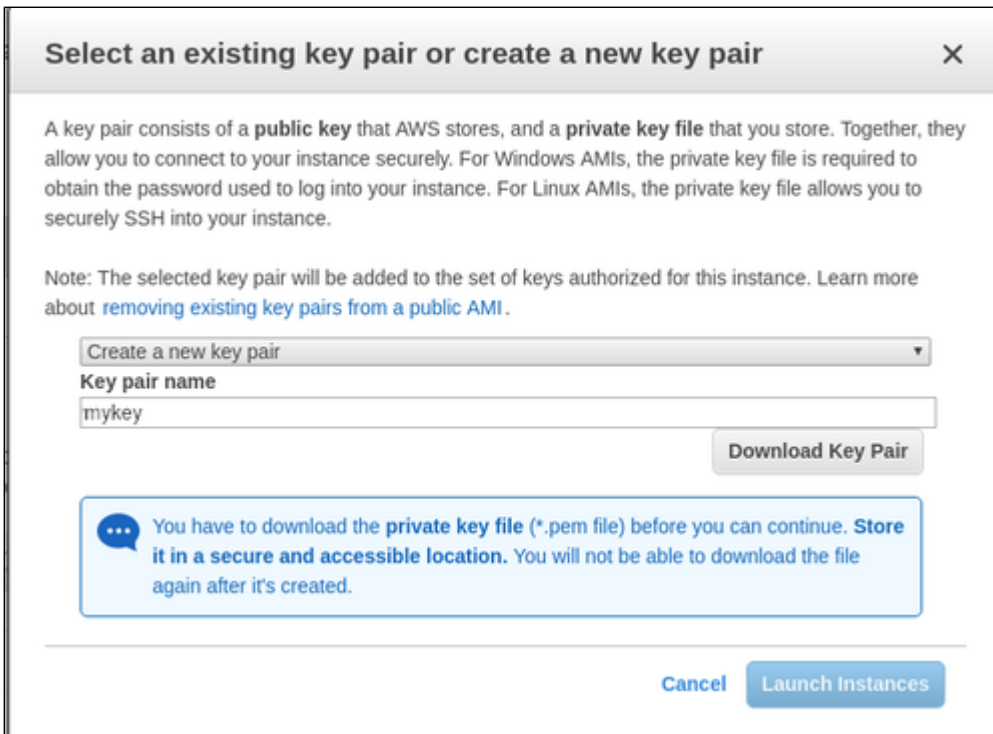
**Storage** [Edit storage](#)

[Cancel](#) [Previous](#) [Launch](#)

If you are satisfied with the settings, click on the **Launch** button to start your instance for the first time.

## 8. Launch and select/create key-pair for access:

When starting the instance for the first time, you will be shown a window asking you to create a new key-pair or to use an existing one. When creating a new key-pair, you must download the **private key** to your local system and store it in a safe place. It is required to access your instance. The **public key** is stored in the newly created Charon-SSP host system, in the **authorized\_keys** file of the **sshuser** and the **charon** user. The sample below shows the window when the creation of a new key-pair was selected:



You cannot start the instance without downloading the key. If you select to re-use an existing key-pair, you have to confirm that you are in possession of the private key before you can launch the instance.

### 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. After this, 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:



The following sections will show you how to access the instance and how to perform additional storage and network configurations.

**i** If you select your instance, the bottom of the screen will show a detailed description and status information of your instance.

# Charon-SSP Manager Installation

## Contents

- Overview
- Installation Packages and Installation Steps
  - Installation Packages
  - Installation Steps on Linux

## Overview

The Charon-SSP Manager is the main interface for managing the emulated SPARC systems running on a Charon-SSP AWS EC2 host. Therefore, the Charon-SSP Manager must be installed on every local system that will be used to manage the Charon instances running on the Charon-SSP cloud host.

Stromasys provides Charon-SSP Manager installation packages for the following Linux distributions and versions:

- Versions 7.x or higher of Oracle Linux (64 bit) version, Red Hat Enterprise Linux (64 bit), or CentOS (64 bit).
- Ubuntu 17 or higher (64 bit)

Support for Charon Manager on Microsoft Windows is planned for a later date.

## Installation Packages and Installation Steps

### Installation Packages

Installation packages are available in RPM or Debian package formats:

- RPM package: **charon-manager-ssp-*<version>*.rpm**
- Debian package: **charon-manager-ssp-*<version>*.deb**

#### Obtaining the installation packages:

The packages are included in the Charon-SSP AWS AMI. Once a new instance has been launched, you can download the Charon-SSP Manager packages from the running instance:

- Connect to the public IP address of the instance via SFTP using the private key assigned during launch and the user **charon**:  
`$ sftp -i <path-to-private-key> charon@<public-ip-of-aws-instance>`
- Download the required package:  
`sftp> get charon-manager-ssp-<version>.[rpm | deb]`

### Installation Steps on Linux

The following table describes the installation steps for Charon-SSP Manager:

Step	Details
1	Log-in to the local Linux system as the <b>root</b> user (denoted by the <b>#</b> prompt).
2	Copy the installation package to the local Linux system
3	Go to the directory where the package has been stored: <code># cd &lt;package-location&gt;</code>
4	<b>Install package:</b>
	For systems with RPM package management (Red Hat, CentOS): <code># yum install &lt;package-name&gt;</code>
	For systems with Debian package management (Debian, Ubuntu): <code># dpkg -i &lt;package-name&gt;</code>

**Example (RPM):**

```

# yum install /media/sf_vmshare/charon-manager-ssp-3.1.21.rpm
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Examining /media/sf_vmshare/charon-manager-ssp-3.1.21.rpm: charon-manager-ssp-3.1.21-1.x86_64
Marking /media/sf_vmshare/charon-manager-ssp-3.1.21.rpm to be installed
Resolving Dependencies
--> Running transaction check
--> Package charon-manager-ssp.x86_64 0:3.1.21-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
charon-manager-ssp x86_64 3.1.21-1 /charon-manager-ssp-3.1.21 4.2 M

Transaction Summary
=====
Install 1 Package

Total size: 4.2 M
Installed size: 4.2 M
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : charon-manager-ssp-3.1.21-1.x86_64 1/1
Verifying : charon-manager-ssp-3.1.21-1.x86_64 1/1

Installed:
charon-manager-ssp.x86_64 0:3.1.21-1

Complete!

```

# Accessing the Charon-SSP AWS Instance

## Contents

There are several ways to connect to your Charon-SSP AWS instance. Some of them are described below.

- SSH Command-Line Access
- SFTP File Transfer
- Connecting with the Charon-SSP Manager

## SSH Command-Line Access

### Contents

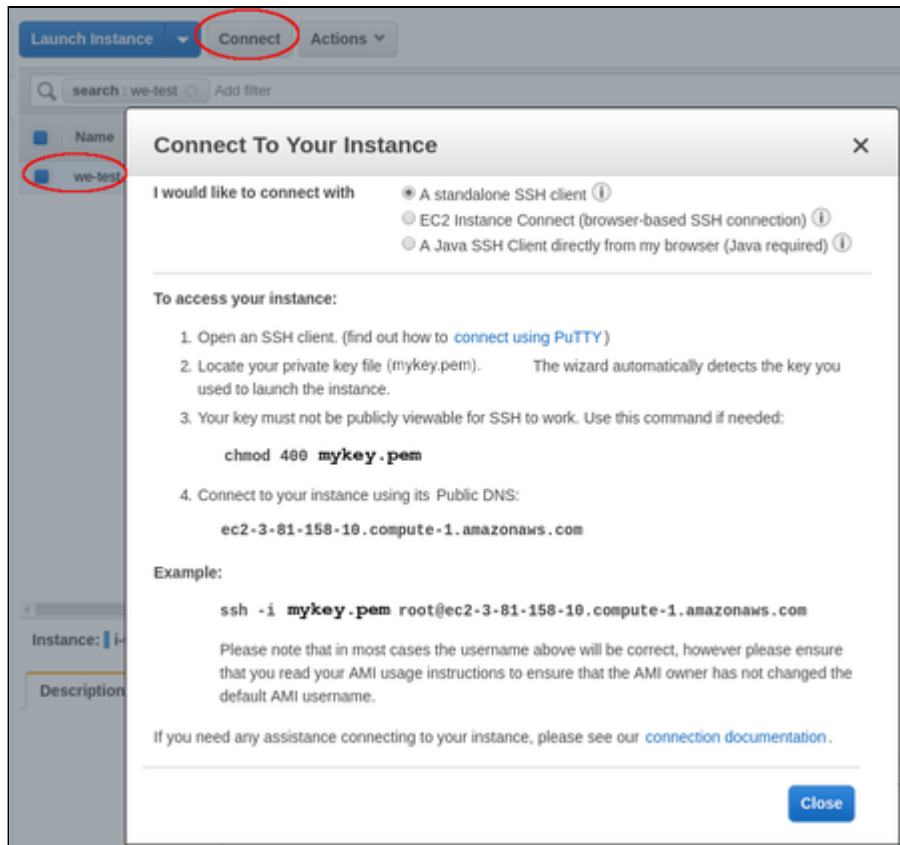
- General Information
- General Login Steps
- Setting the Management Password

### General Information

During the configuration of your instance you should have created a security group allowing at the minimum SSH access to the instance. If this has been done correctly, you can use SSH from the command-line or from a tool such as PuTTY to access the command-line of the user **sshuser** on the Charon-SSP instance. If you select your instance in the instance list and then click on **Connect**, you will see the instructions for connecting via SSH.

As shown in the image below, you will see in particular

- the name of the private key that must be used to connect to the instance, and
- the public DNS name of the instance.



⚠ The file permissions of the private key file must be set such that the file is only readable by the user as shown in the **chmod** example above.

## General Login Steps

To connect to the instance interactively, you must connect as the user **sshuser**. Use the following command:

```
$ ssh -o ServerAliveInterval=30 -i <path-to-your-private-key> sshuser@<AWS-public-IP-address>
```

The parameter `ServerAliveInterval` will protect the connection from timing out.

Below, you see sample output of a login:

```
$ ssh -o ServerAliveInterval=30 -i .ssh/mykey.pem sshuser@<public-ip-address>
Last login: Tue May 21 05:34:33 2019 from myhost.example.com
[sshuser@ip-172-31-38-252 ~]$ pwd
/home/sshuser
```

**!** Note that this account allows root access to a limited subset of commands (use `sudo -i`). In particular, commands that are required to create more complex network configurations are allowed.

## Setting the Management Password

**!** **Initial management password configuration:** before connecting to the Charon-SSW AWS host with the Charon Manager for the first time after the initial installation of your instance you must set the management password. This can either be done via the Charon Manager itself (see [Connecting with the Charon-SSP Manager](#)) or via the command line as shown below.

Steps to set the management password:

- Log in to the Charon host using SSH as show above.
- Become the root user (`sudo -i`).
- Change to the Charon Agent utilities directory (`cd /opt/charon-agent/ssp-agent/utils`).
- Run the charon-password script (`./charon-passwd`).
- Enter and confirm the new management password when prompted.

After this has been completed, you can connect to the host using the Charon Manager with the new management password.

Below, you see sample output of the steps:

```
$ ssh -i .ssh/mykey.pem sshuser@<public-ip-address>
[sshuser@ip-172-31-35-32 ~]$ sudo -i
[root@ip-172-31-35-32 ~]# cd /opt/charon-agent/ssp-agent/utils
[root@ip-172-31-35-32 utils]# ./charon-passwd
Enter new Charon password:
Retype new Charon password:
Password updated successfully.
Changing password for user charon.
passwd: all authentication tokens updated successfully.
Changing password for user sshuser.
passwd: all authentication tokens updated successfully.
sh: /home/charon/.vnc/passwd: No such file or directory
[root@ip-172-31-35-32 utils]#
```



## SFTP File Transfer

SFTP enables file transfers to and from the Charon-SSP AWS instance. The user for file transfers is the **charon** user. The security group must allow SSH access to allow SFTP access to the Charon-SSP AWS instance.

To connect to the instance as the user **charon**, use the following command:

```
$ sftp -i <path-to-your-private-key> charon@<AWS-public-IP-address>
```

Below you see sample output of a connection:

```
$ sftp -i .ssh/mykey.pem charon@<public-ip-address>
Connected to storage@3.81.64.139.
sftp> ls
media                ssp-snapshot
```

## Connecting with the Charon-SSP Manager

### Contents


- General Information
- Starting the Charon Manager and Login to Charon Host
  - Starting the Charon Manager
  - Entering Charon Manager Login Information and Connecting to Charon Host

### General Information

To manage Charon-SSP and the emulated SPARC systems, you must connect to the Charon-SSP AWS instance with the Charon-SSP Manager. The Charon-SSP Manager is the main interface to all important functions of the Charon-SSP software.

#### Prerequisites:

- The **Charon-SSP Manager** must be installed on your local system.
- **For access via the public IP address of the Charon host instance:**
  - The **Security Group** on your Charon host instance must at least allow SSH access. This allows the **built-in SSH tunneling** of the Charon-SSP Manager to work. Should you not use SSH tunneling, you must open up additional ports (see [Firewall and AWS Security Group Considerations](#)). However, if the connection runs over the Internet without a VPN, Stromasys recommends strongly to use SSH tunneling to protect your Charon-SSP cloud instance and any emulated systems running on it.
  - You must have the public IP address of the Charon-SSP AWS instance. To determine this address refer to the instance information displayed on the AWS EC2 management console.
  - To use the Charon Manager integrated SSH tunnel, you need the private SSH key of the key-pair associated with your instance.
- **For access via an SSH-based VPN:**
  - Active SSH-based VPN (see [SSH VPN - Connecting Charon Host and Guest to Customer Network](#))
  - Private IP address of the Charon-SSP host in the VPN

 **Initial management password configuration:** before connecting to the Charon-SSP AWS host with the Charon Manager for the first time after the initial installation you must set the management password. This can either be done via the command line (see [SSH Command-Line Access](#)) or via the Charon Manager as described below.

## Starting the Charon Manager and Login to Charon Host

### Starting the Charon Manager

**To start the Charon-SSP Manager** and to open the Charon Manager login window, log in on your Linux management system and use the following command:

```
$ /opt/charon-manager/ssp-manager/ssp-manager
```

The steps above will open the Charon Manager login window which has **two tabs**.

## Entering Charon Manager Login Information and Connecting to Charon Host

### Step 1: the Charon Manager **Login** tab

If the management password has not yet been set, perform the following steps:

- Enter the **public** IP address of your Charon-SSP host instance.
- Leave the password field empty.
- Enable the SSH tunnel configuration (select **ON**).
- Change to the SSH tab to fill in the required information there.

If the management password has already been set, perform the following steps:

- Enter the public IP address or the private VPN IP address of your Charon-SSP instance.
- Enter the Charon-SSP management password.
- Enable the SSH tunnel configuration for communication across a public network unless you use a secure VPN connection.
- If the SSH tunnel is enabled, change to the SSH tab to fill in the required information there.

### Step 2: the Charon Manager **SSH** tab

If you use the integrated SSH tunnel, perform the following steps:

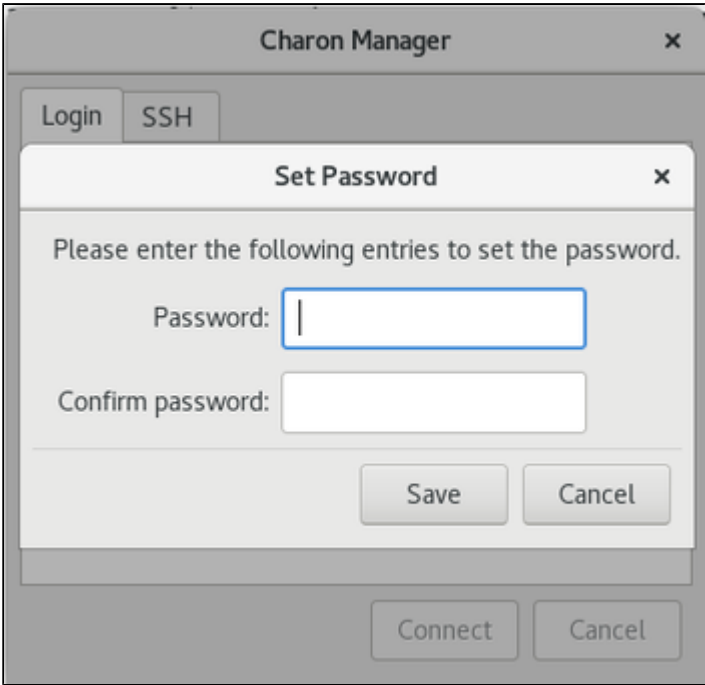
- Enter the Charon-SSP user (**charon** or **sshuser**).
- Enter the path to the private key file (click on the three dots to open a file browser),
- In rare cases, you may need to add the path to the public key on the local system.
- Enter the passphrase for the private key if required.
- Adjust the server port (default 22) if required.

**i** The public key of the AWS key-pair can be copied from the `.ssh/authorized_keys` file of the **sshuser** of the instance.

**Step 3: connecting to the Charon host system**

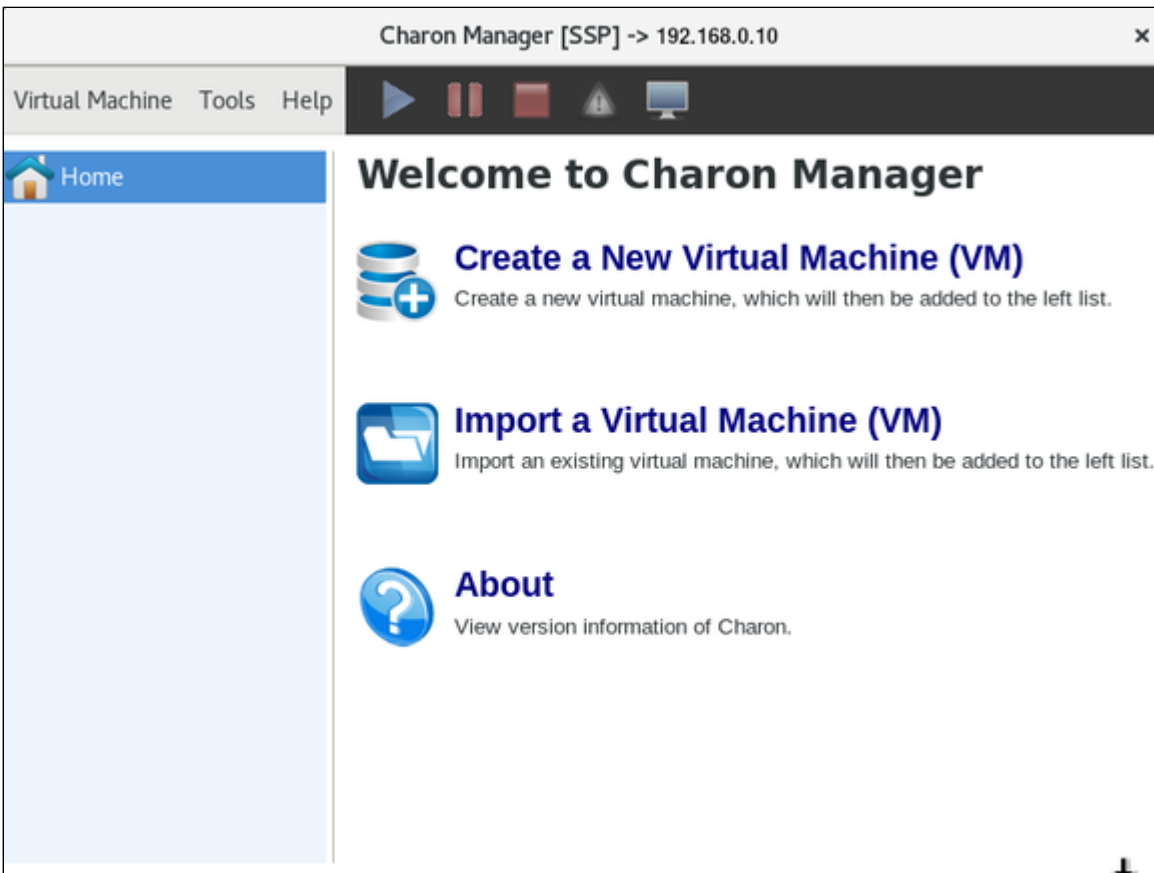
After entering all the required information, click on **Connect** to connect to the Charon-SSP AWS instance.

If the management password still needs to be set, you will receive a prompt to enter the new password:



- Enter the desired password and confirm it.
- Then click on **Save**.
- The login process continues.

After a connection has been successfully created, the Charon Manager welcome screen opens. Example of the Charon Manager welcome page:



## Additional Charon-SSP AWS Instance Configuration

This section describes some additional AWS configuration options that can be used with the Charon-SSP AWS instance.

### Contents

- Storage Management
- Network Management

### Storage Management

To add additional disk storage to your Charon-SSP AWS instance (for example, for storing virtual disk containers), perform the steps described below.

### Contents

- AWS Storage Environment
  - Creating a New Volume
  - Attaching an Existing Volume to an Instance
  - Detaching a Volume from an Instance
- Storage Manager of the Charon-SSP Manager
  - Mounting a Newly Attached Volume
  - Unmounting a Volume

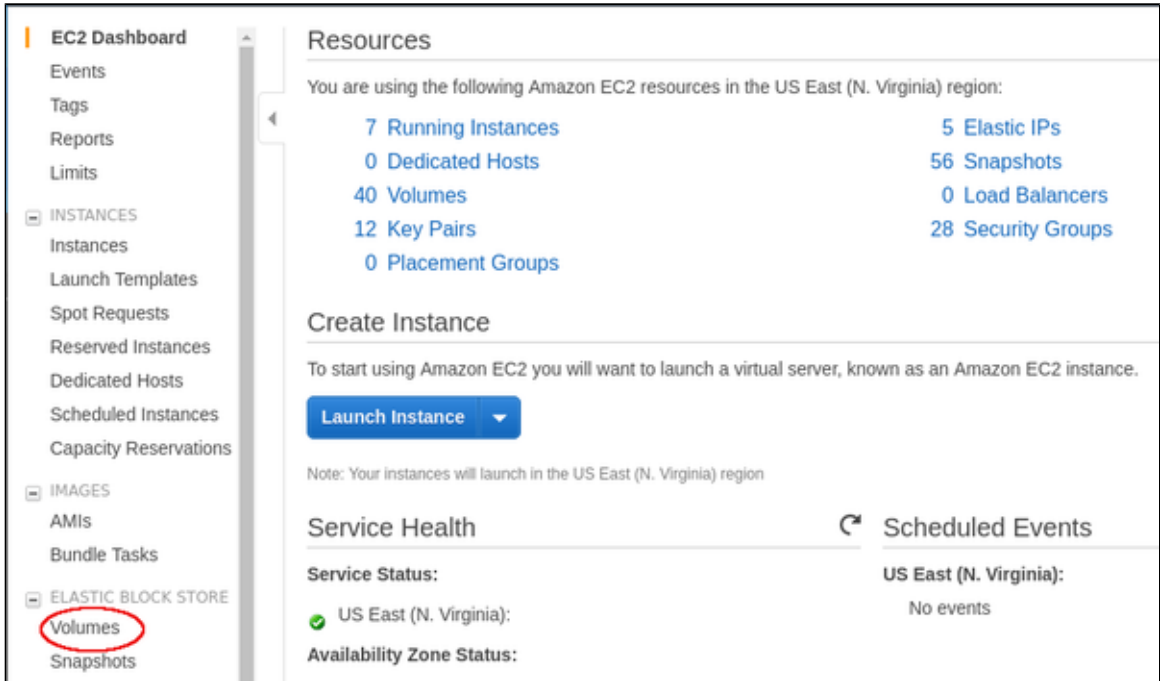
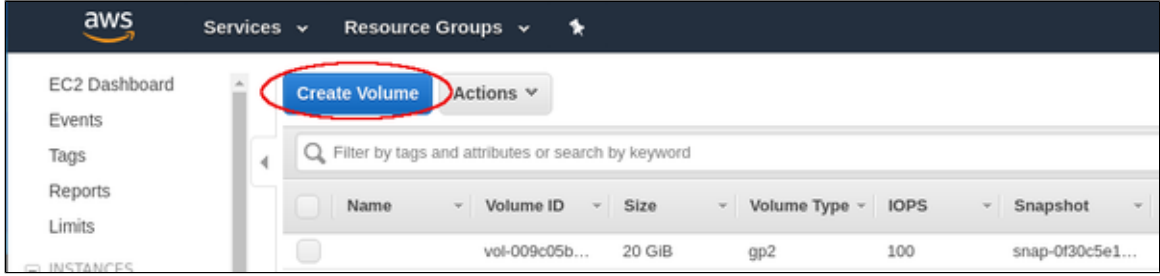
### AWS Storage Environment

In the AWS environment, you can, for example,

- create a new storage volume,
- attach an existing storage volume to your instance,
- detach a storage volume from your instance.

These steps are shown below.

## Creating a New Volume

Step	Details
<p><b>Open the <i>Volumes</i> configuration from the EC2 dashboard</b></p>	 <p>The screenshot shows the AWS EC2 dashboard. On the left, there is a navigation menu with categories: EC2 Dashboard, INSTANCES, IMAGES, and ELASTIC BLOCK STORE. Under ELASTIC BLOCK STORE, the 'Volumes' link is circled in red. The main content area shows 'Resources' for the US East (N. Virginia) region, including 7 Running Instances, 0 Dedicated Hosts, 40 Volumes, 12 Key Pairs, and 0 Placement Groups. Below this is a 'Create Instance' section with a 'Launch Instance' button. To the right, there is a 'Service Health' section showing 'US East (N. Virginia): No events'.</p> <p>This will open the volume overview screen.</p>
<p><b>Create a new volume</b></p>	 <p>The screenshot shows the AWS Volumes overview screen. At the top, there is a navigation bar with 'Services' and 'Resource Groups'. Below this, there is a 'Create Volume' button circled in red. The main content area shows a search bar and a table of volumes. The table has columns for Name, Volume ID, Size, Volume Type, IOPS, and Snapshot. One volume is listed with Volume ID 'vol-009c05b...' and Size '20 GiB'.</p> <p>This will open the volume creation window.</p>

## Select

- Volume type
- Volume size
- Availability zone ( ⚠ Must be the same as the Charon-SSP AWS instance!)
- Optionally, add a name tag.

Then click on **Create Volume**.

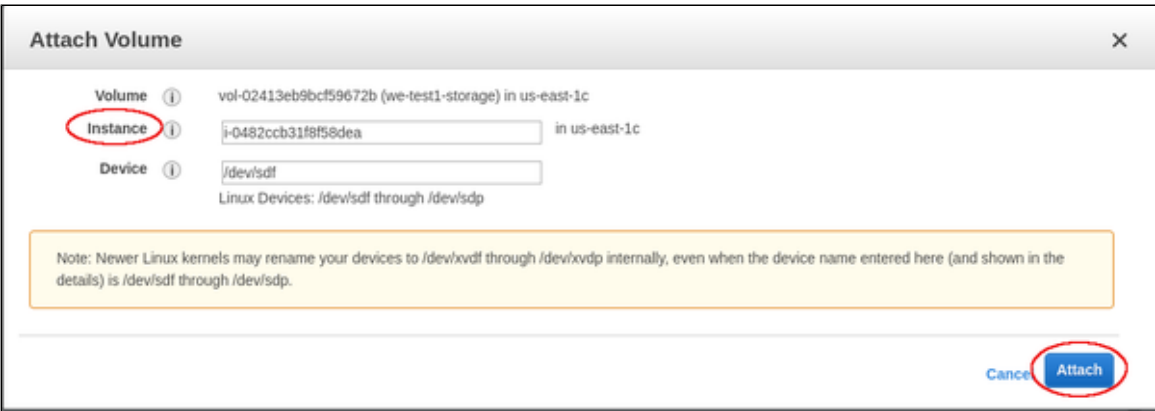
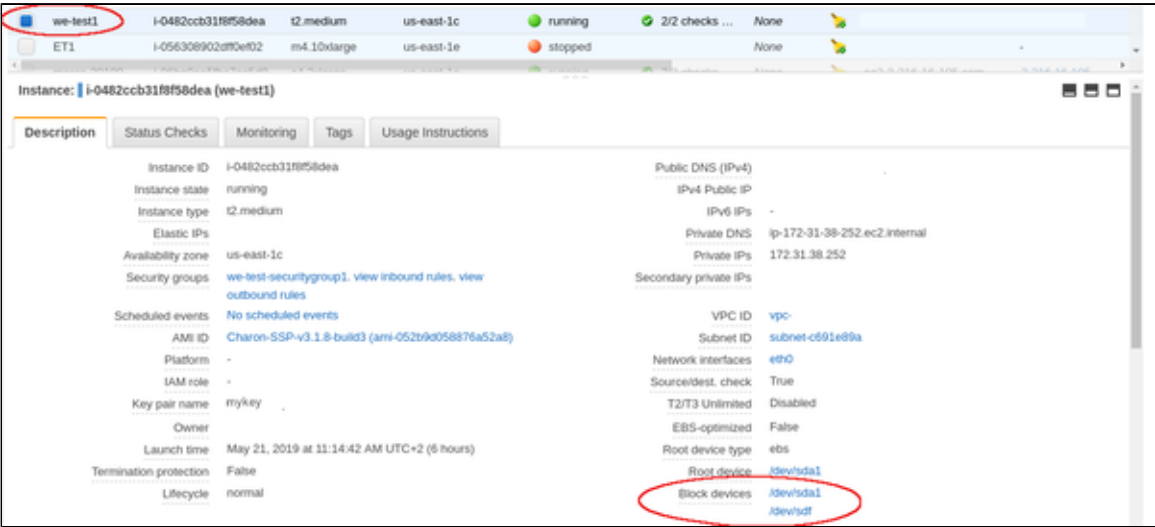
The screenshot displays the AWS 'Create Volume' interface. Key elements include:

- Volume Type:** General Purpose SSD (gp2)
- Size (GiB):** 100 (Min: 1 GiB, Max: 16384 GiB)
- IOPS:** 300 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)
- Availability Zone:** us-east-1c
- Throughput (MB/s):** Not applicable
- Snapshot ID:** Select a snapshot
- Encryption:**  Encrypt this volume
- Key/Value:** Key (127 characters maximum), Value (255 characters maximum)
- Name:** we-test-storage
- Add Tag:** 49 remaining (Up to 50 tags maximum)
- Buttons:** Cancel, **Create Volume**

You will receive a confirmation window. Close it to return to the volume overview screen where you should now see the new volume.

## Attaching an Existing Volume to an Instance

Once a volume has been created, you can attach it to your instance.

Step	Details
<b>Attach the volume to your instance</b>	<p>In the volumes overview screen, check that your volume has the state <b>available</b>. Right-click on it and select <b>Attach Volume</b>. This will open a small input screen.</p>  <p>Select the instance to which the volume is to be attached. Optionally, you can change the device name that will be presented to the Charon-SSP AWS instance.</p> <p>Click on <b>Attach</b> to confirm the configuration and to attach the volume to the instance.</p> <p>The status of the volume in the volume overview screen will change from <b>available</b> to <b>in-use</b>.</p>
<b>Verify success in the instance description</b>	<p>Go to the instance overview list and select your instance. In the description tab at the bottom, you will see that the instance now has two block devices.</p> 



## Detaching a Volume from an Instance

If the volume is the root device of the instance, you must stop the instance before detaching the volume.

If the volume is not the root device of the instance, **unmount** the volume in the Charon host system before detaching it (see Charon-SSP Manager section below).

Then detach the volume from your instance:

- Go to the Elastic Block Store volumes list.
- Select the volume to be detached.
- Use the menu **Actions > Detach volume**.

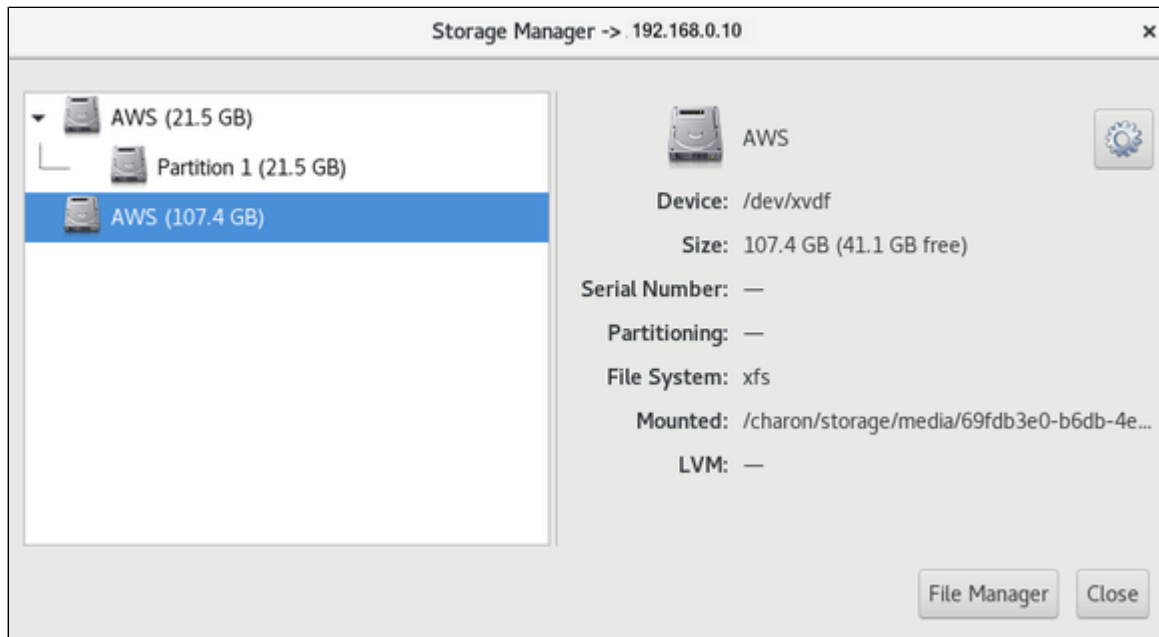
## Storage Manager of the Charon-SSP Manager

### Mounting a Newly Attached Volume

After the volume has been attached to the instance, it must be included in the Charon-SSP host system configuration. This is achieved via the Charon-SSP Manager.

1. Open the Charon-SSP Manager on your local system and connect to your AWS instance.
2. Select **Tools > AWS Cloud > Storage Manager**.
3. In the **Storage Manager** window, perform the following steps:
  - a. Select the new device.
  - b. Click on the cog-wheel symbol.
  - c. **Only if required**, select **Format Volume** to create a filesystem on the new device.
    - ⚠ This will delete all data on the volume.
  - d. Click on the cog-wheel symbol and select **Mount the Filesystem**.

This will mount the new volume under `/charon/storage/media/<UUID>/`. The following image shows a sample:



⚠ Note that the device on the host system is called `/dev/xvdf` (XEN virtual block device). This is equivalent to an `/dev/sdf` volume shown in AWS.

Once the filesystem has been mounted, the space is available to the Charon-SSP host system.

### Unmounting a Volume

To **unmount** a volume before perform the following steps:

- Stop all Charon instances that might use the volume that is about to be unmounted.
- In Charon Manager go to **Tools > AWS Cloud > Storage Manager**.
- Select the volume.
- Click on the cogwheel symbol and select **Unmount the Filesystem**.

## Network Management

To add an additional network interface to an instance or to remove an interface from your instance perform the steps described below.

**!** The steps below only provide a basic overview. The exact tasks required will vary depending on your network design.

### Contents

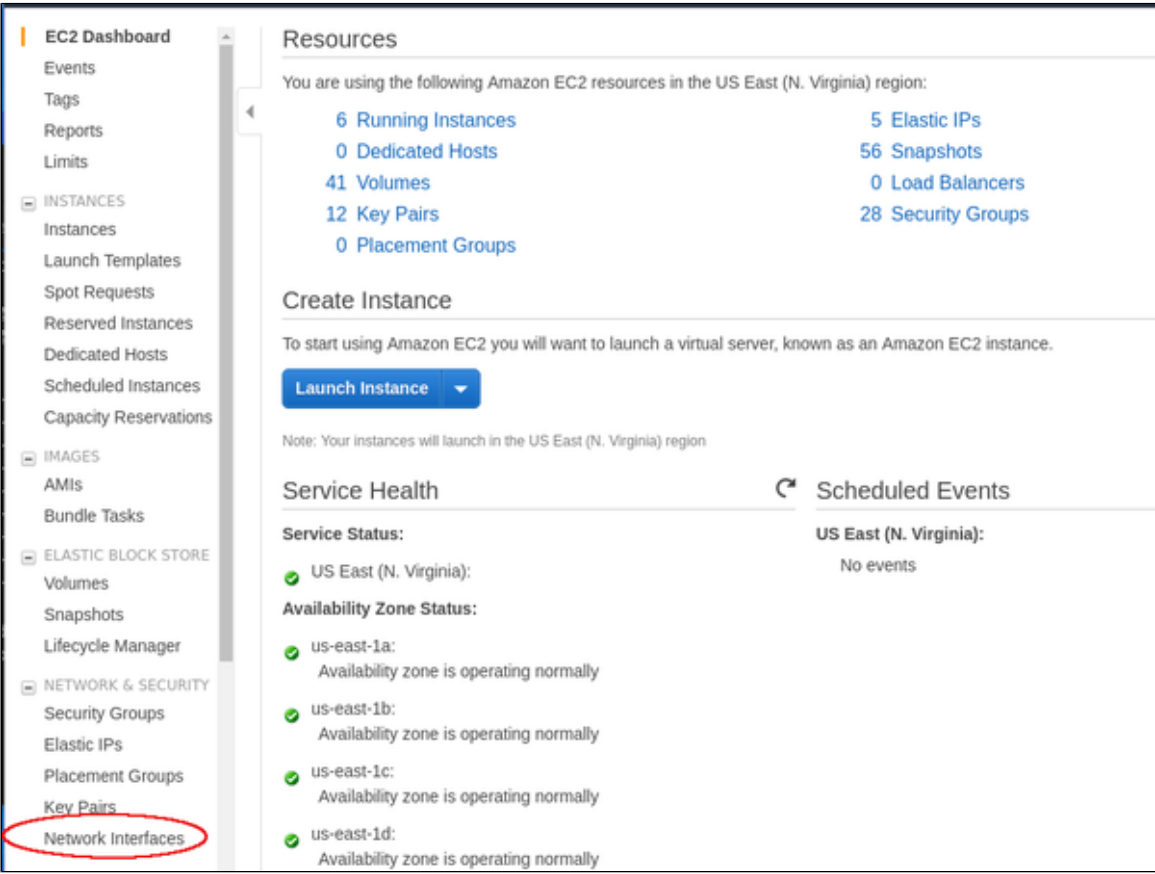
- Creating a New Network Interface
- Attaching the Interface to your Instance
- Assigning an Elastic IP Address to the Network Interface
- Detaching a Network Interface from an Instance

When an instance is created, a default Ethernet interface is attached to the system (eth0). You can create additional network interfaces and attach them to an instance.

**!** If an instance has only one Ethernet interface, a public IP address can be assigned to the interface automatically. However, this address will be removed by AWS if a second interface is added to the instance and the instance is stopped and restarted. Be careful not to lose connectivity to your instance when changing the network configuration.

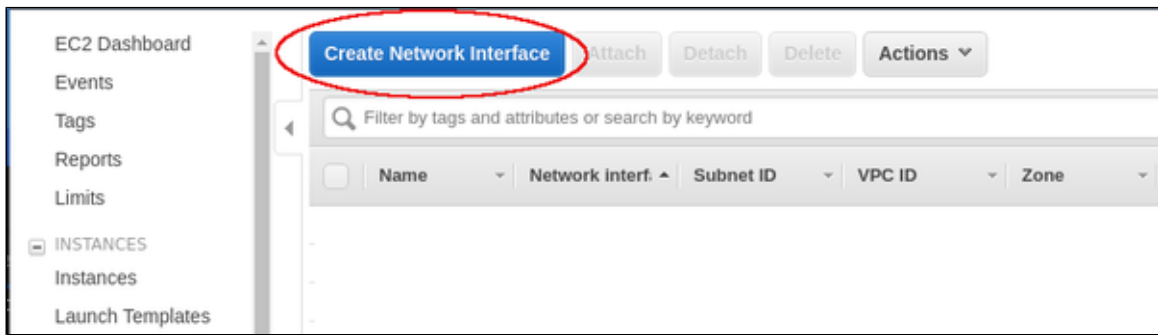
### Creating a New Network Interface

The following steps are required to create a new network interface that can later be added to an instance:

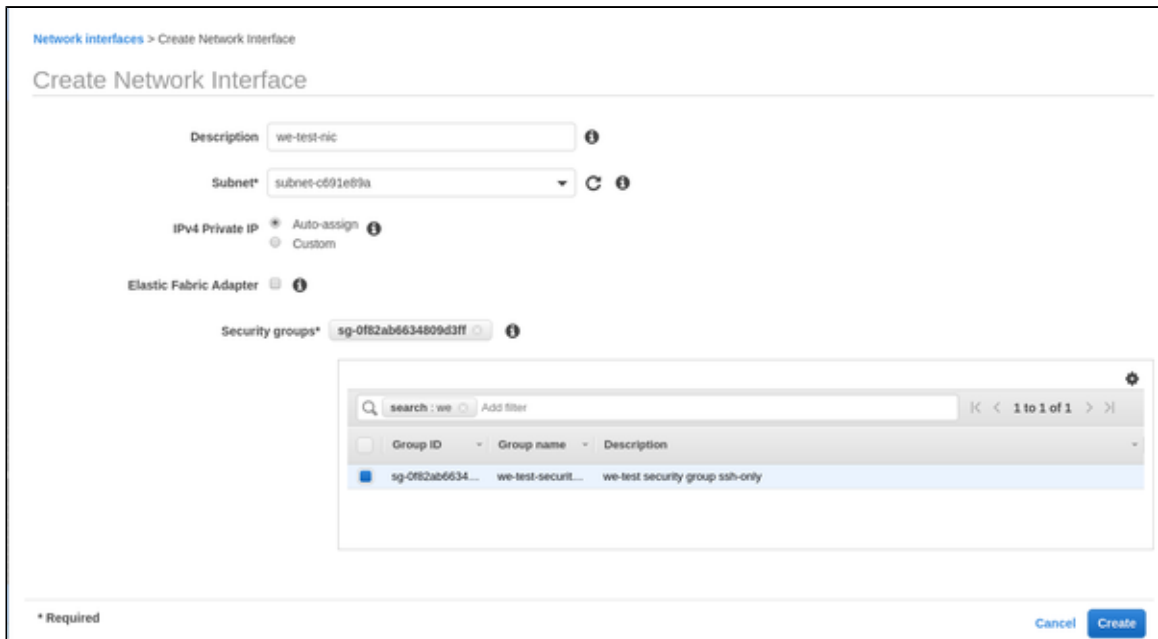
Step	Details
<p><b>Locate the Network Interfaces option on the EC2 dashboard and click on it.</b></p>	 <p>The screenshot displays the AWS Management Console interface. On the left, a navigation pane lists various services under the 'NETWORK &amp; SECURITY' category, with 'Network Interfaces' highlighted by a red circle. The main area shows a 'Resources' section with counts for EC2 resources in the US East (N. Virginia) region: 6 Running Instances, 0 Dedicated Hosts, 41 Volumes, 12 Key Pairs, 0 Placement Groups, 5 Elastic IPs, 56 Snapshots, 0 Load Balancers, and 28 Security Groups. Below this is a 'Create Instance' section with a 'Launch Instance' button and a note about the region. The 'Service Health' section shows that the US East (N. Virginia) region and its four availability zones (us-east-1a, us-east-1b, us-east-1c, us-east-1d) are all operating normally.</p>
	<p>Clicking on <b>Network Interfaces</b> opens the list of existing network interfaces.</p>

**Create a new interface.**

Click on **Create Network Interface** at the top of the interface list.



This opens the interface creation window.



On this screen,

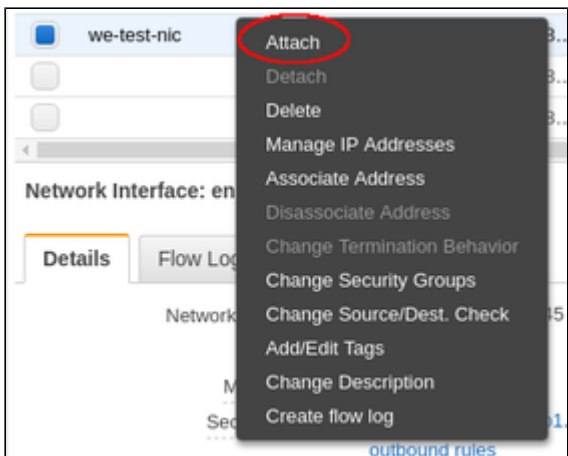


- enter a description,
- select the subnet the interface should be on (select the subnet to which your instance is to be connected),
- allow AWS to automatically assign a private IP address or set a custom one from the subnet IP range, and
- associate the interface with a security group (often the same as for the instance).

Click on **Create** when you are done. The new interface will appear in the overview list. There you can assign a name to the interface. Check that the interface is in state **available**.

## Attaching the Interface to your Instance

After creating a network interface, you have to assign it to the instance where it will be used.

- Stopping and restarting the instance after adding a second network interface will release any automatically assigned public IP address. If several interfaces are required where one or more are configured with a public address, use Elastic IP addresses.
- Additionally, adding a second network interface to a non-Amazon Linux EC2 instance causes traffic flow issues. This occurs in cases of asymmetric routing where traffic to the instance arrives at one network interface and leaves the instance through the other network interface. This is blocked by AWS because a mismatch between MAC address and IP address. Refer to the AWS documentation and [AWS Networking and Charon-SSP](#) for more information. Failure to use the proper steps, may make your instance unreachable!
- If your instance supports enhanced networking there may be naming inconsistencies when adding additional interfaces to a running instance. Please refer to the interface names section in [AWS Networking and Charon-SSP](#).
- The NetworkManager is disabled on Charon-SSP AWS. Therefore, `ifcfg`-files in `/etc/sysconfig/network-scripts` are required to define the IP configuration of an interface.

Step	Details	
<p><b>Locate your network interface in the interface list and right-click on it.</b></p>	<p>The right-click opens the context menu. Select <b>Attach</b>.</p> <p>This will open the window to enter the necessary instance information.</p>	
<p><b>Select your instance and confirm entry.</b></p>	<p>Select your instance from the drop-down list and click on <b>Attach</b>.</p> <p>The state of your interface will change from <b>available</b> to <b>in-use</b>.</p>	
<p><b>Verify that instance has second interface.</b></p>	<p>Select your instance in the instance list. The description tab in instance details should now display two network interfaces:</p>	

**i** You can also attach/detach existing interfaces from the instance overview screen. Select your instance and then **Actions > Networking > Attach or Detach network interface**.

## Assigning an Elastic IP Address to the Network Interface

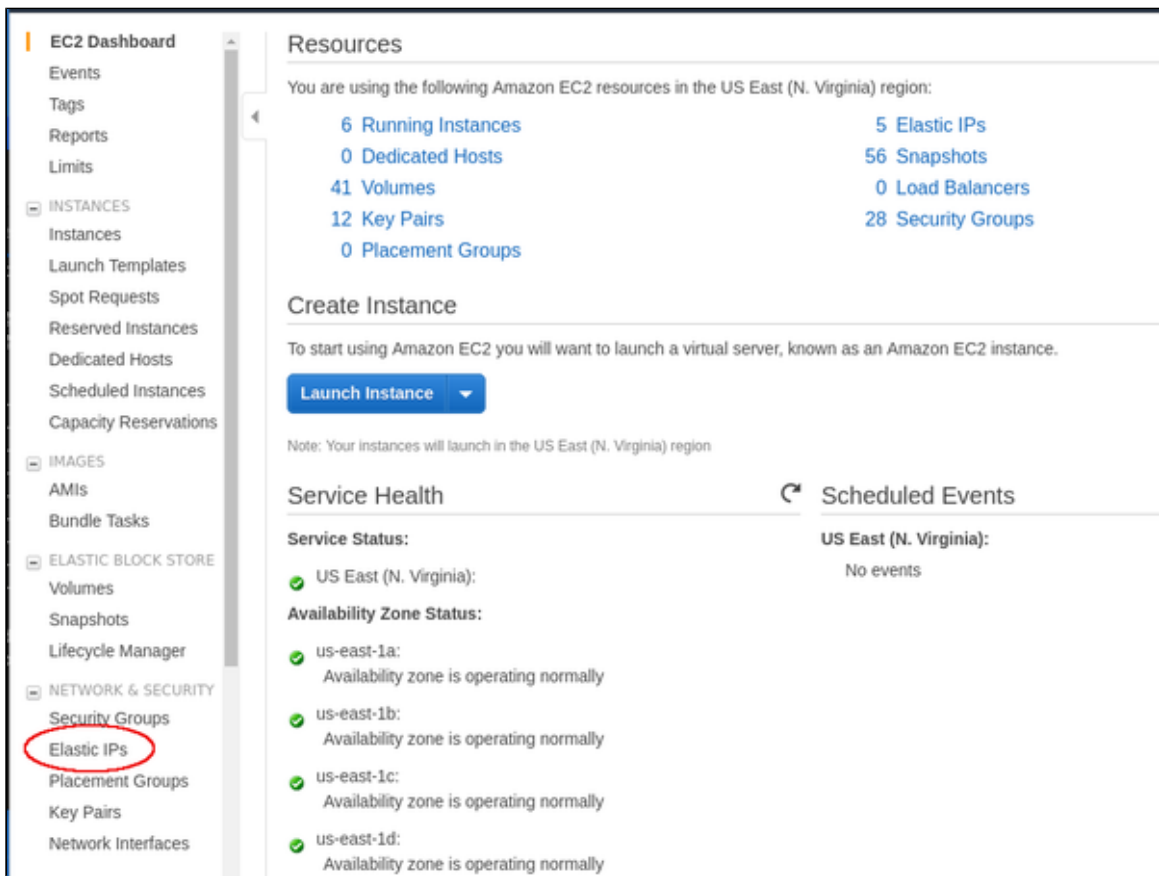
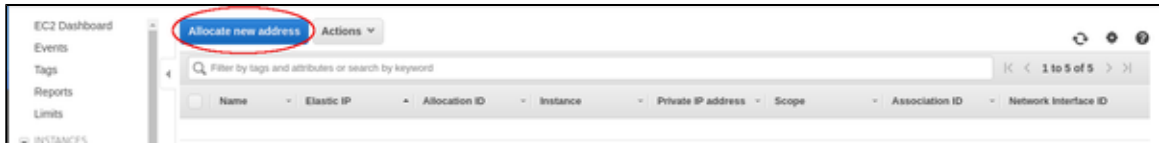
The public IP address assigned to your instance by default when it starts, is not persistent. You will receive a new address when the instance is stopped and started again.

An Elastic IP address is a persistent, public IPv4 address to be used for one of your network interfaces or instances. You can associate an Elastic IP address with any instance or network interface in your account.

**i** The advantage of associating the Elastic IP address with the network interface instead of directly with the instance is that you can move the network interface with its attributes easily from one instance to another.

**!** The initial automatically assigned public IP address will be removed as soon as you restart the instance after adding a network interface with an Elastic IP address to your instance. Do not restart your instance before you are sure you can reach it via the Elastic IP address. The automatically assigned public IP address will also be disabled if you assign an Elastic IP address to the primary Ethernet interface of the instance.

The table below describes the steps required to add an Elastic IP address to a network interface.

Step	Details
<p><b>Locate the Elastic IPs option on the EC2 dashboard and click on it.</b></p>	 <p>This will list the already created Elastic IP addresses.</p>
<p><b>Allocate a new address.</b></p>	<p>In the overview list, click on <b>Allocate new address</b> if you need to allocate a new address. It is also possible to assign an existing address to an interface. However, each address can only be used for one instance.</p>  <p>This will open the address allocation window.</p>

In the address allocation window, select the Amazon pool (or your own pool of public addresses), and click on **Allocate**.

The new address will be shown in the list.

### Associate the address with the network interface.

Right-click on the address and select **Associate**. A window to enter the required options opens.


In the window,

- select to associate the IP address with a network interface,
- select your network interface from the drop-down menu,
- connect the public address to the private address of the interface, and
- click on **Associate** to complete the step.

## Detaching a Network Interface from an Instance

You can detach a network interface from your instance in two ways:

1. Select your instance in the instance list and use the menu **Actions > Networking > Detach Network Interface**. Or,
2. Select your network interface in the network interface list and use the menu **Actions > Detach**.

 Take care that this step will not make your instance unreachable.

 The primary network interface cannot be detached.

# Configuring and Managing the System Using the Charon-SSP Manager

The Charon Manager is the GUI-based management interface for Charon-SSP AWS. With the Charon Manager, you can manage multiple virtual machines and virtual networks on the Charon-SSP AWS instance. The Charon manager runs on a remote system and accesses the Charon-SSP cloud instance across the SSH tunnel functionality integrated in the Manager or via a VPN.

**i** Unencrypted access is possible but should not be used across a public network.

The following sections describe how to use the Charon Manager for the different aspects of managing an emulated SPARC system on Charon-SSP AWS.

The main topics in this section are:

- Starting the Charon Manager
- Creating a Virtual Machine
- Configuring a Virtual Machine
  - Hardware Family (Model) Configuration
  - CPU Configuration
  - DIT Configuration
  - Memory Configuration
  - Graphics Configuration
  - SCSI Storage Configuration
  - Serial Line Configuration
  - Audio Configuration
  - Ethernet Configuration
  - NVRAM Configuration
  - Log Configuration
- Virtual Machine Context Menu
- Host System Network Configuration
- Miscellaneous Management Tasks
- AWS Cloud Tools
- Graphical Interface via X11 Server on Linux
- Starting, Stopping, and Suspending the Emulated System

## Starting the Charon Manager

### Contents

- Starting the Charon Manager
- General Information
- Starting the Charon Manager and Login to Charon Host
  - Starting the Charon Manager
  - Entering Charon Manager Login Information and Connecting to Charon Host
- Charon-SSP Manager for AWS Overview
  - Charon Manager on Linux
  - Other Options Provided by the Charon Manager




## Starting the Charon Manager

### General Information

To manage Charon-SSP and the emulated SPARC systems, you must connect to the Charon-SSP AWS instance with the Charon-SSP Manager. The Charon-SSP Manager is the main interface to all important functions of the Charon-SSP software.

#### Prerequisites:

- The **Charon-SSP Manager** must be installed on your local system.
- **For access via the public IP address of the Charon host instance:**
  - The **Security Group** on your Charon host instance must at least allow SSH access. This allows the **built-in SSH tunneling** of the Charon-SSP Manager to work. Should you not use SSH tunneling, you must open up additional ports (see [Firewall and AWS Security Group Considerations](#)). However, if the connection runs over the Internet without a VPN, Stomasys recommends strongly to use SSH tunneling to protect your Charon-SSP cloud instance and any emulated systems running on it.
  - You must have the public IP address of the Charon-SSP AWS instance. To determine this address refer to the instance information displayed on the AWS EC2 management console.
  - To use the Charon Manager integrated SSH tunnel, you need the private SSH key of the key-pair associated with your instance.
- **For access via an SSH-based VPN:**
  - Active SSH-based VPN (see [SSH VPN - Connecting Charon Host and Guest to Customer Network](#))
  - Private IP address of the Charon-SSP host in the VPN

 **Initial management password configuration:** before connecting to the Charon-SSP AWS host with the Charon Manager for the first time after the initial installation you must set the management password. This can either be done via the command line (see [SSH Command-Line Access](#)) or via the Charon Manager as described below.

## Starting the Charon Manager and Login to Charon Host

### Starting the Charon Manager

**To start the Charon-SSP Manager** and to open the Charon Manager login window, log in on your Linux management system and use the following command:

```
$ /opt/charon-manager/ssp-manager/ssp-manager
```

The steps above will open the Charon Manager login window which has **two tabs**.

## Entering Charon Manager Login Information and Connecting to Charon Host

### Step 1: the Charon Manager **Login** tab

If the management password has not yet been set, perform the following steps:

- Enter the **public** IP address of your Charon-SSP host instance.
- Leave the password field empty.
- Enable the SSH tunnel configuration (select **ON**).
- Change to the SSH tab to fill in the required information there.

If the management password has already been set, perform the following steps:

- Enter the public IP address or the private VPN IP address of your Charon-SSP instance.
- Enter the Charon-SSP management password.
- Enable the SSH tunnel configuration for communication across a public network unless you use a secure VPN connection.
- If the SSH tunnel is enabled, change to the SSH tab to fill in the required information there.

### Step 2: the Charon Manager **SSH** tab

If you use the integrated SSH tunnel, perform the following steps:

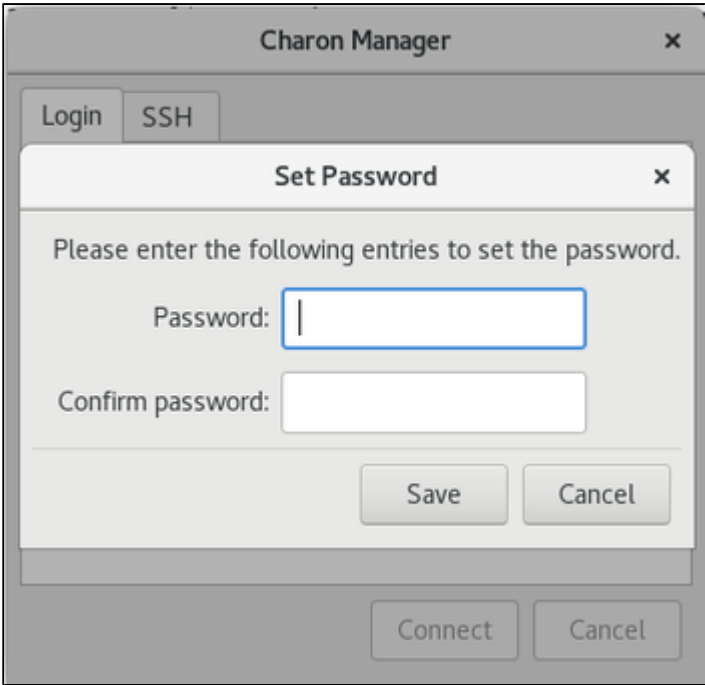
- Enter the Charon-SSP user (**charon** or **sshuser**).
- Enter the path to the private key file (click on the three dots to open a file browser),
- In rare cases, you may need to add the path to the public key on the local system.
- Enter the passphrase for the private key if required.
- Adjust the server port (default 22) if required.

**i** The public key of the AWS key-pair can be copied from the `.ssh/authorized_keys` file of the **sshuser** of the instance.

**Step 3: connecting to the Charon host system**

After entering all the required information, click on **Connect** to connect to the Charon-SSP AWS instance.

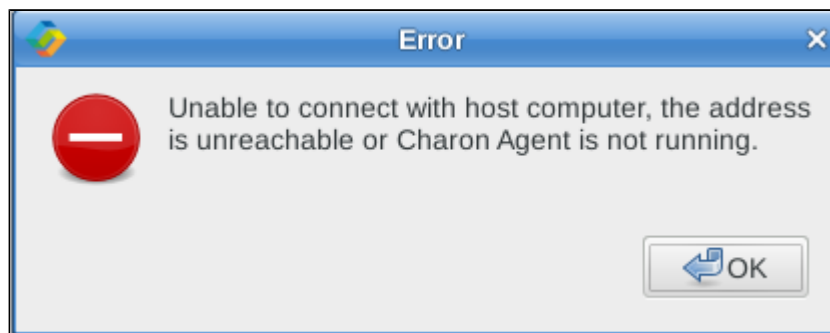
If the management password still needs to be set, you will receive a prompt to enter the new password:



- Enter the desired password and confirm it.
- Then click on **Save**.
- The login process continues.

**Problem if the agent on target system is not running or unreachable:**

If you receive an error similar to the one displayed in the screenshot below, verify that the host specified in the IP address field is correct and that access is not blocked by a firewall or security group. If the problem persists, connect to your instance via SSH, become the root user and restart the Charon Agent (`# /etc/init.d/charon-agentd-ssp restart`).

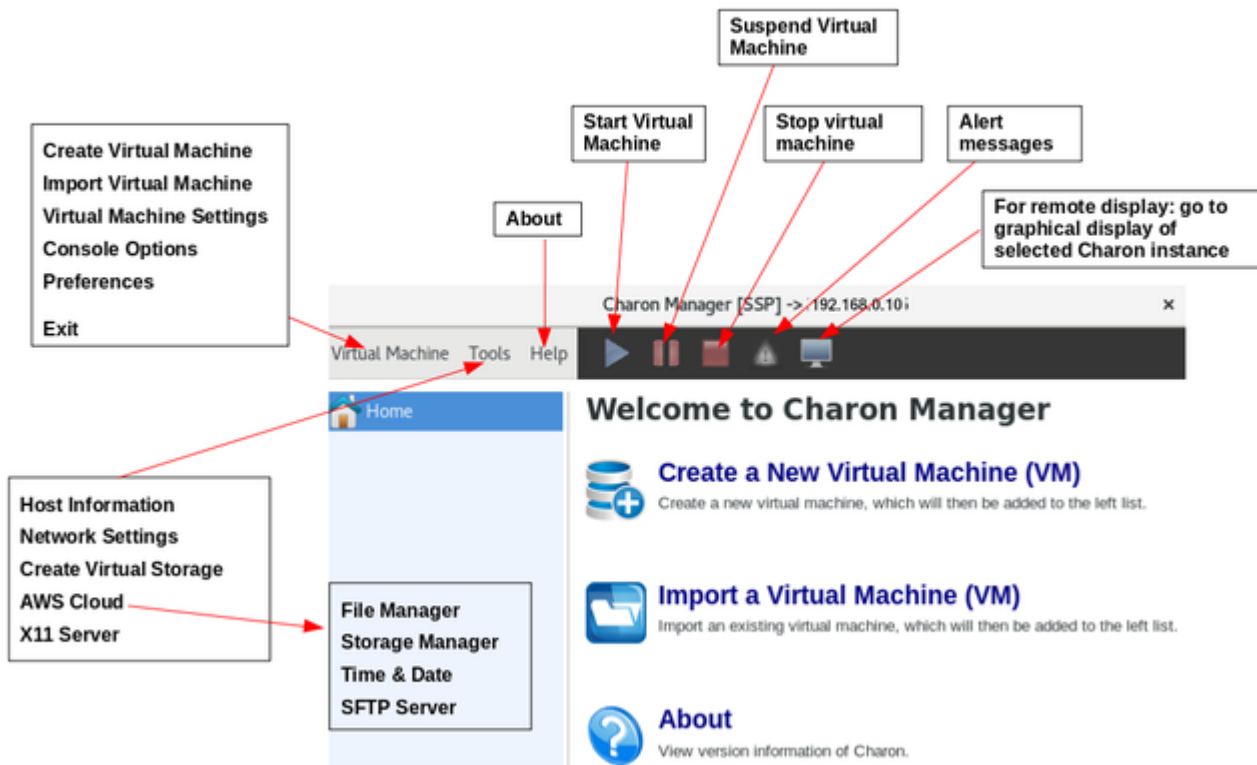


**i** Remember that the automatically assigned public IP address changes when the instance is restarted (this is different from a persistent Elastic IP address).

## Charon-SSP Manager for AWS Overview

### Charon Manager on Linux

The image below provides a first overview of the Charon Manager menus:



### Other Options Provided by the Charon Manager

The Charon-SSP Manager virtual machine pane on the left has some additional functions applicable to all product variants:

- Double-clicking on Home sorts the virtual machines alphabetically. Repeating the action toggles between ascending and descending sort order.
- The position of a virtual machine in the list can be changed by manual drag-and-drop.
- A right-click in the pane when no virtual machine is selected opens a context menu to create or import a virtual machine.

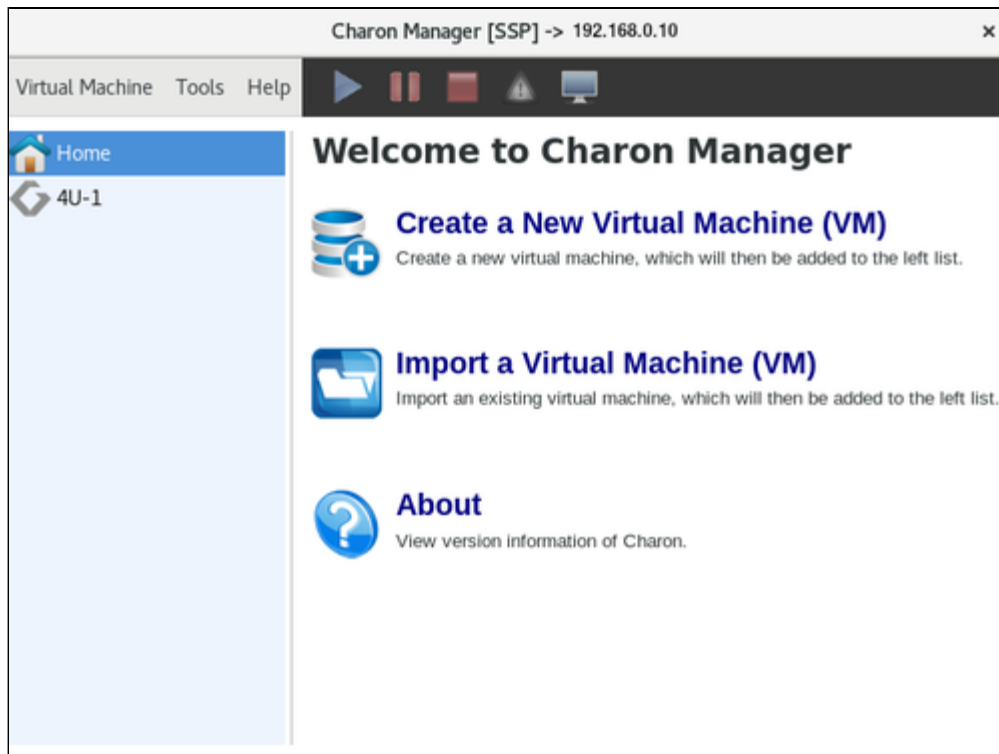
## Creating a Virtual Machine


The first step to running an emulated SPARC machine is to create the initial configuration using the following steps.

Steps to create a virtual machine	
Step	Description
1	Open the Charon Manager.
2	Open the <b>New Virtual Machine</b> window using either of the following methods: <ul style="list-style-type: none"> <li>From the opening screen titled <b>Welcome to Charon Manager</b>, click the <b>Create a New Virtual Machine</b> icon, or</li> <li>use the <b>Create</b> option in the <b>Virtual Machine</b> menu, or</li> <li>while <b>Home</b> is selected, right-click into an empty area in the virtual machine list pane and select the option to create a new virtual machine from the context menu.</li> </ul>
3	Select the appropriate <b>Hardware Model</b> by clicking the radio button labelled with the SPARC family that most closely matches the system to you wish to run. <ul style="list-style-type: none"> <li>The hardware family <b>SUN-4M</b> represents a SPARC V8 32-bit model.</li> <li>The hardware family <b>SUN-4U</b> represents a SPARC V9 64-bit model.</li> <li>The hardware family <b>SUN-4V</b> represents a SPARC V9 64-bit model with the 4V features.</li> </ul> <p>The configured model must be covered by your license.</p>
4	Enter a name for the virtual machine in the <b>Virtual machine name</b> field.
5	Click on <b>OK</b> .

The steps above create a basic new virtual machine configuration. The new virtual machine is listed the left-hand pane of the management interface showing the **Virtual machine name** you specified.

The screenshot below shows the management interface screen after the 4U-1 emulated system has been created:



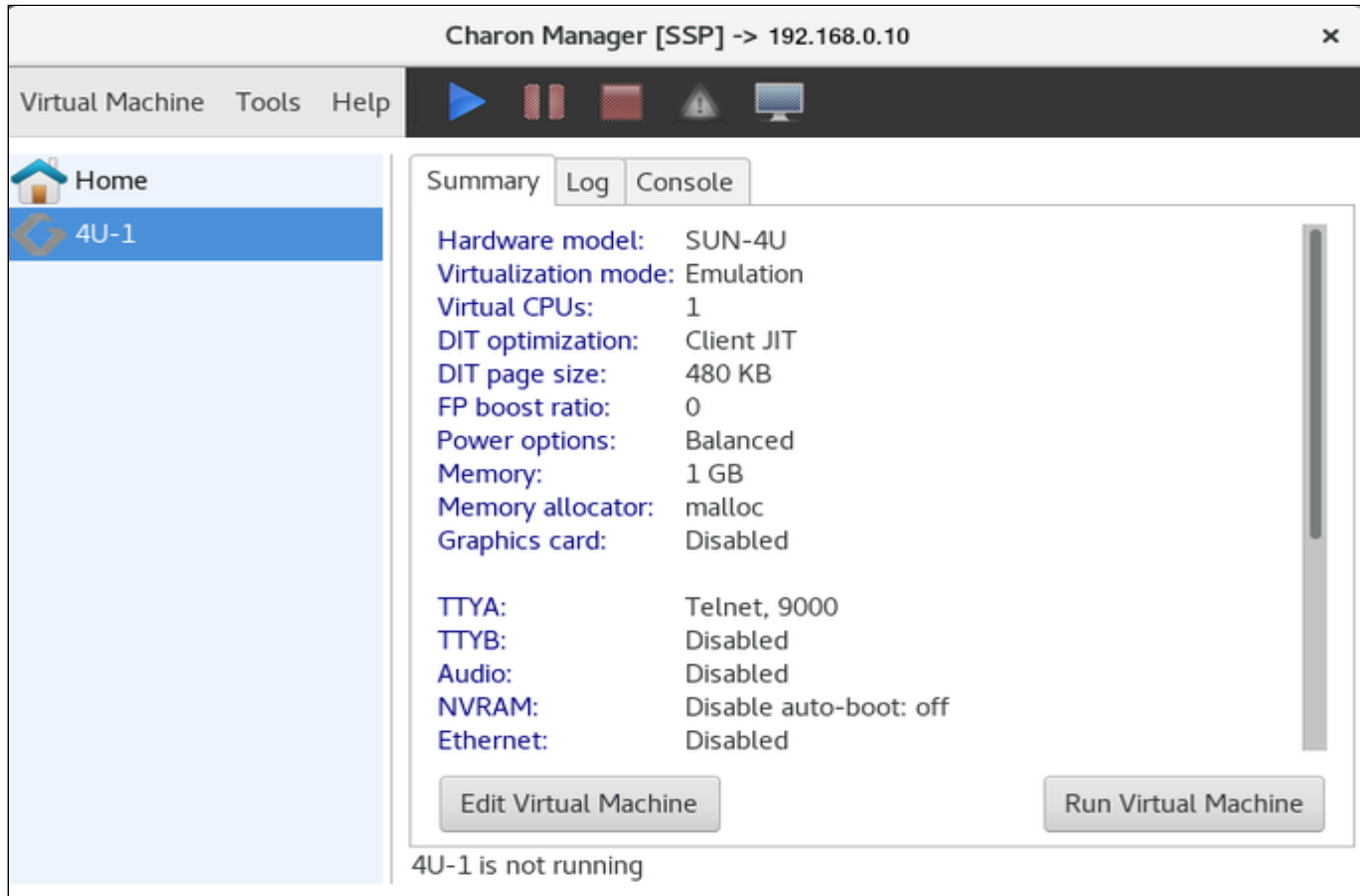
 The initial configuration of the virtual machine is only a configuration template. To complete the configuration, continue with the next section (*Configuring a Virtual Machine*).

## Configuring a Virtual Machine

To open the configuration window in Charon Manager, first **select** the name of the virtual machine in the left-hand pane of the Charon Manager. This shows the virtual machine overview page, including

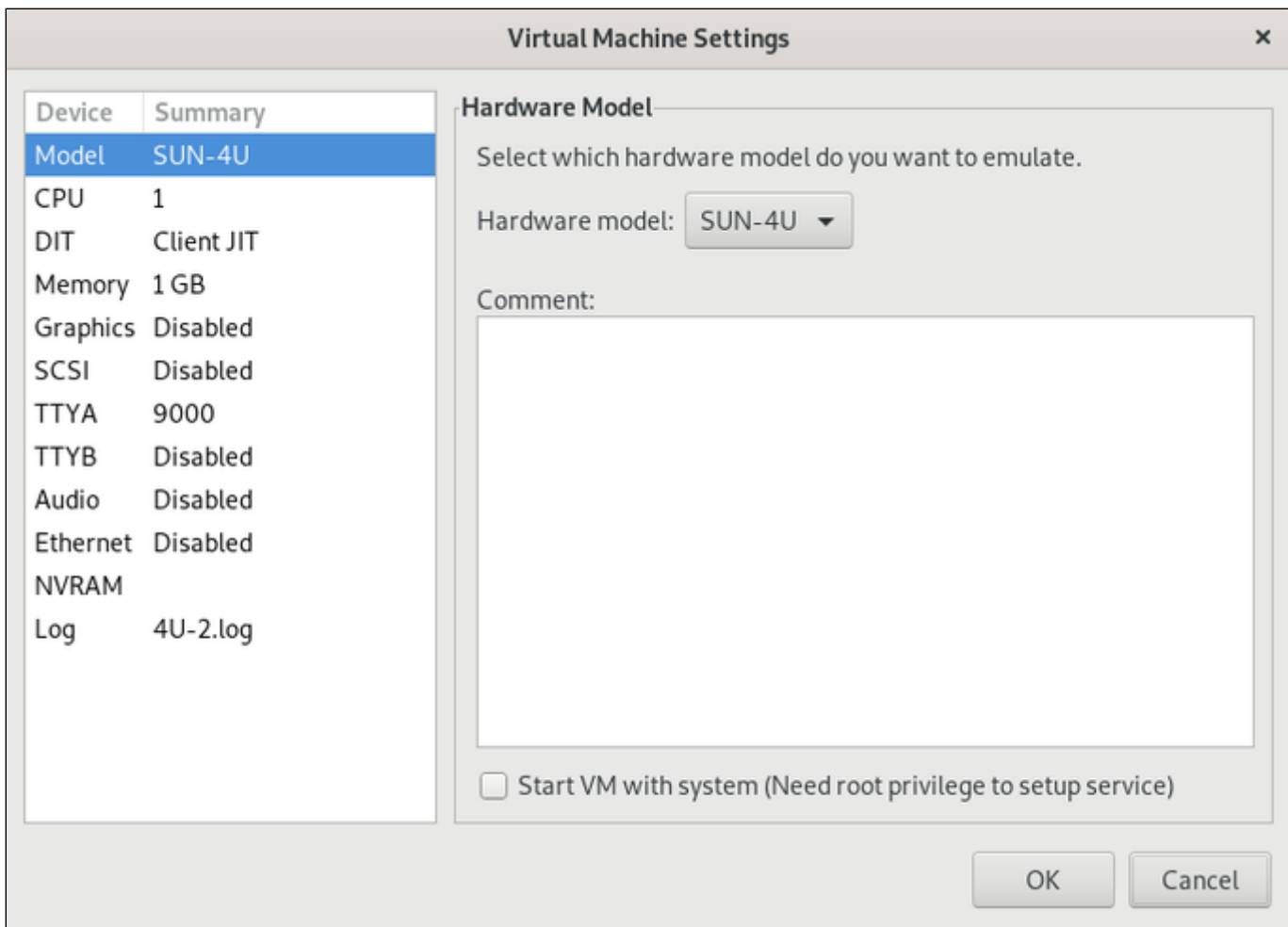
- a **summary** tab showing an overview of the current configuration of the virtual machine,
- the **log** file tab to display the log files of the virtual machine, and
- the **console** tab to display the built-in serial console of the emulated system.

The image below shows an example of the virtual machine overview page:



To continue with the configuration of the emulated system, **click** on the **Edit Virtual Machine** button or select **Virtual Machine Settings** from the emulated system context menu or the Virtual Machine menu. This opens the **Virtual Machine Settings** window for the virtual machine.

The example below shows the configuration window of a SUN-4U system:



The following sections describe the individual items of the **Virtual Machine Settings** window.

**!** For any changes to take effect, the virtual machine must be restarted. However, it is also recommended that before making any configuration changes the virtual machine be shut down correctly.

## Hardware Family (Model) Configuration

To **view** the configured virtual machine hardware family, select **Model** in the **Device** column on the left. This displays the current value in the field **Hardware Model** (see figure above).

The hardware families currently supported by **Charon-SSP/4M** are:

- Sun-4c and Sun-4m (an example would be the Sun SPARCstation 20)

The hardware family currently supported by **Charon-SSP/4U(+)** is:

- Sun-4u (an example would be the Sun Enterprise 450)

The hardware family currently supported by **Charon-SSP/4V(+)** is:

- Sun-4v (an example would be the SPARC T2)

The **Comments** field allows you to add additional optional information about the virtual machine.

The option **Start VM with system** integrates the startup of the Charon-SSP instance in the host system startup. With this option enabled, the virtual machine starts automatically when the system boots. The root privileges required for this setup are provided automatically if the configuration is made via Charon-SSP Manager and Agent.

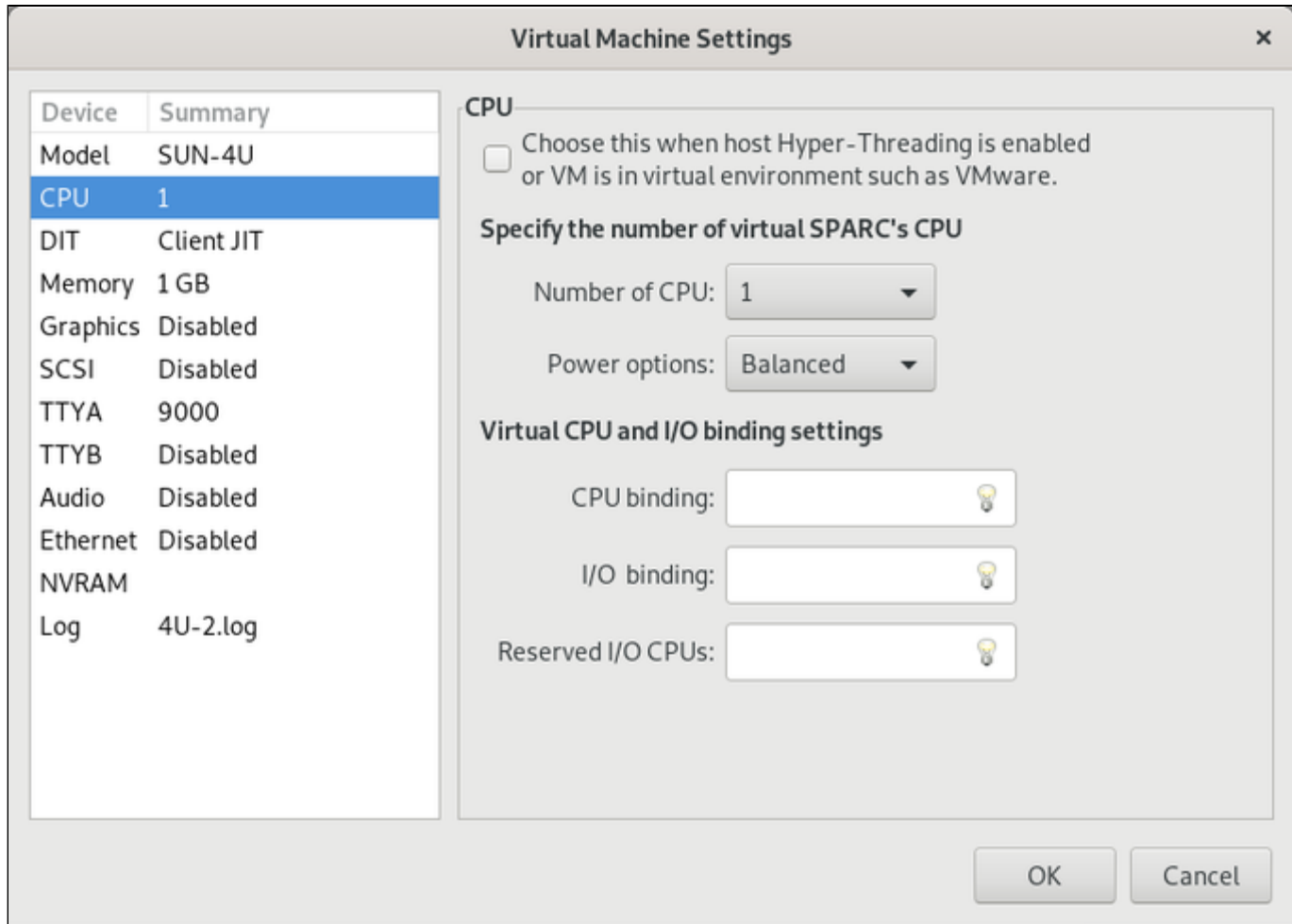
**!** If a VM is started automatically with the host system startup and stopped with host system shutdown, it is the responsibility of the user to **shut down the guest OS cleanly** before host system shutdown. Failing to do so may cause data corruption in the guest system.

**i** Unless otherwise mentioned, the terms Charon-SSP/4U and Charon-SSP/4V also include Charon-SSP/4U+ and Charon-SSP/4V+.

## CPU Configuration




To view or change the current virtual machine CPU configuration, select **CPU** in the left-hand pane of the Settings window.

This will open the CPU configuration screen. The image below shows the Charon-4U configuration screen as an example:





The following table lists each of the fields in the CPU configuration window and describes their use.

Virtual machine CPU configuration fields	
Field	Description
<b>Hyper-threading checkbox</b>	<p>Enable the Charon-SSP adaption for a hyper-threading host environment or for running the Charon host under a Hypervisor. With this mode enabled, Charon-SSP does not set a CPU core affinity on the host system, but relies on the scheduler of the host operating system instead. Power option <b>Power save</b> will allow idle guest system CPU threads to be rescheduled.</p> <p> If the Charon host runs in the AWS environment on a non-metal instance type, this option should be enabled.</p>
<b>Number of CPU</b>	<p>Configure the number of virtual SPARC CPUs. Supported number of CPUs:</p> <ul style="list-style-type: none"> <li>Charon-SSP/4M: 1 to 4 virtual SPARC CPUs</li> <li>Charon-SSP/4U(+): 1 to 24 virtual CPUs</li> <li>Charon-SSP/4V(+): 1 to 64 virtual CPUs</li> </ul>
<b>Power options</b>	<p>This option determines the host CPU behavior when the guest Solaris is in idle state.</p> <ul style="list-style-type: none"> <li><b>Performance</b> Choosing this option keeps the host CPU in a busy loop waiting for next Solaris activity. This option offers the best response time in Solaris but the host CPU usage is at 100% all the time.</li> <li><b>Balanced (default)</b> Choosing this option allows the host CPU to go into an idle state until the next Solaris activity. This option offers a good balance between Solaris response time and host CPU usage.</li> <li><b>Power save</b> The host CPU is in deep "sleep" mode when the guest Solaris is in idle state. With this option and hyper-threading mode set, an idle Solaris guest system CPU thread can be rescheduled.</li> </ul>
<b>CPU binding</b>	<p>Assign specific host CPUs to the processing of SPARC instructions. If configured, each virtual SPARC CPU must be assigned to exactly one specific host CPU for instruction processing.</p> <p>This field consists of a comma-separated list of CPU IDs (index starts from 0). If left blank, the virtual machine software will assign affinity itself starting with the highest CPU ID (<b>recommended</b>). Cannot be used with hyper-threading mode enabled. CPU cores assigned to emulated CPUs are never shared between instances.</p>
<b>I/O binding</b>	<p>Assign specific host CPUs to the processing of virtual machine I/O requests.</p> <p>This field consists of a comma-separated list of CPU IDs. If left blank, the virtual machine will assign I/O processing affinity itself starting from CPU ID 0 (<b>recommended</b>). CPUs listed here cannot be shared between instances.</p> <p> If there is an overlap with manually configured bindings in other instances or the automatically calculated I/O CPU allocations, the instance will not start with the message: "Wrong IO affinity setting: already allocated by another thread."</p>
<b>Reserved I/O CPUs</b>	<p>Reserve a number of CPUs on the host system for processing virtual machine I/O requests. Allocation will start from the lowest CPU ID. If neither <b>I/O binding</b> nor <b>Reserved I/O CPUs</b> is set, Charon will assign 1/3 (minimum 1; rounded down) of the number of host CPU cores to I/O processing starting from the lowest CPU ID (<b>recommended</b>). If there is an overlap between a manual configuration in one instance and the automatic calculation of I/O CPUs in other instances, overlapping I/O CPUs are shared between instances.</p> <p> If the number of I/O CPU cores (configured or calculated automatically) + the number of emulated CPUs is higher than the number of available host CPU cores, the following error is logged and the emulator does not start: "Wrong CPU affinity setting: no enough host CPUs."</p>

**Please note:**

- Manual I/O CPU bindings can be used to optimize I/O and DIT performance on a host system running multiple Charon-SSP instances, because it allocates dedicated I/O CPU cores to a system (no sharing).
- Manually configuring the number of reserved I/O CPUs can be used to adjust the CPU pool used for I/O operations. Overlapping CPU cores between several instances will be shared.
- Once any manual configuration is used, its influence on all concurrently active Charon-SSP instances must be considered in order to avoid performance degradation.

## DIT Configuration

### Contents

- Introduction
- Client JIT
- Server JIT

### Introduction

To view or change the current DIT configuration, select **DIT** in the left-hand pane of the Settings window.

There are three levels of DIT optimization:

- OFF: no DIT optimization.
- Client JIT or first level DIT: this is the equivalent of the DIT optimization available in older versions.
- Server JIT or second level DIT: this is a more aggressive optimization. It optimizes SPARC instructions at runtime, based on an MRU policy (Most Recently Used).

**Server JIT is not available on Charon-SSP/4M.** Client and server JIT are implemented in **two separate images** that will be configured automatically by Charon-SSP Manager depending on the configuration.

Comparison between the DIT configuration options:

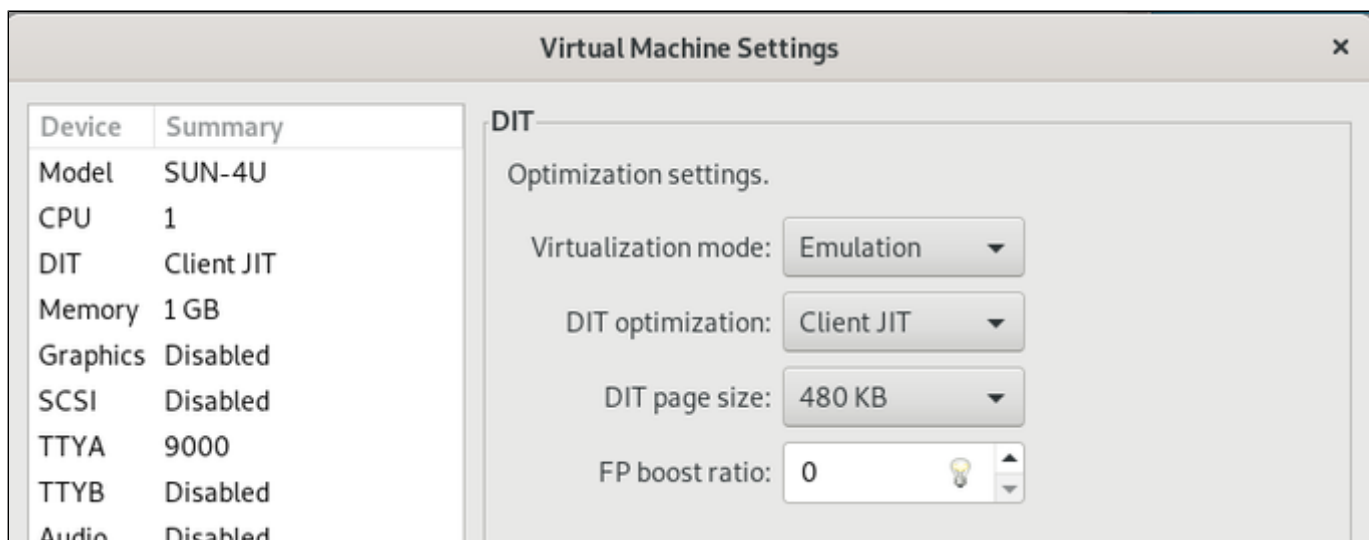
DIT Optimization	Translation speed	Command execution after Translation	Memory requirements
OFF	n/a	Slow	n/a
Client JIT	Fast	Faster	Approx. 2GB RAM
Server JIT	Slow	Fastest (depending on application)	Approx. 6GB RAM

Due to the slower and more resource-consuming translation in Server JIT mode, mostly long-running applications will benefit from Server JIT.

The following sections show the details of both modes.

### Client JIT

This section describes the configuration options available in Client JIT mode. The image below shows a sample of the Charon-4U configuration screen.



The following table lists each of the fields available in the DIT Client JIT mode and describes their use.

Virtual machine client JIT configuration fields	
Field	Description
<b>Virtualization Mode</b>	<p><b>Emulation</b> selects the Charon-SSP/4U or Charon-SSP/4V emulator, <b>Intel VT-x/EPT</b> informs the manager that the hardware prerequisites for running Charon-SSP/4U+ or Charon-SSP/4V+ are met.</p> <p>This option is inactive on Charon-SSP/4M or if Charon-SSP/4U+/4V+ is not installed.</p> <p>Attempting to run Charon-SSP/4U+/4V+ on insufficient hardware will cause the instance to exit with an error message. For example,  “MMU module insertion failed, please check if VT-X is enabled in BIOS”  or  “The host CPU doesn’t support Intel VT-x / EPT”).</p> <p>Check if your AWS instance runs on dedicated hardware if you encounter such errors.</p>
<b>DIT Optimization</b>	<p>This option controls the Dynamic Instruction Translation (DIT). DIT is a just in time compilation technology to dynamically optimize the SPARC instruction execution on x86-64 platforms. It can be set to <b>OFF</b>, <b>Client JIT</b>, or <b>Server JIT</b> (Charon-4U/4V only). The remainder of this table describes the <b>Client JIT</b> parameters.</p>
<b>DIT page size</b>	<p>This option controls the size of the translation buffer holding the translated binary code that results from the DIT optimization. It can be increased to a maximum of 2048KB. This parameter should only be changed if the log file indicates that the DIT optimization was disabled because the translation buffer size was too small. <b>This option is not available on Charon-SSP/4M.</b></p>
<b>FP boost ratio</b>	<p>Defines the level of floating-point optimization. The parameter can be set to a value from 0 to 100. The default is 0 (= no boost). If customer applications are likely to benefit from floating point optimization, the value can be increased. <b>This option is not available on Charon-SSP/4M</b></p>

## Server JIT

This section describes the configuration options available in Server JIT mode (**not available on Charon-SSP/4M**).

The example below shows a Charon-4U configuration screen:

Device	Summary
Model	SUN-4U
CPU	1
DIT	Server JIT
Memory	1 GB
Graphics	Disabled
SCSI	Disabled
TTYA	9000
TTYB	Disabled
Audio	Disabled
Ethernet	Disabled
NVRAM	
Log	4U-2.log

**DIT**

Optimization settings.

Virtualization mode: Emulation

DIT optimization: Server JIT

DIT page size: 480 KB

DIT code cache: 2048 MB

INT boost ratio: 100

FP boost ratio: 0

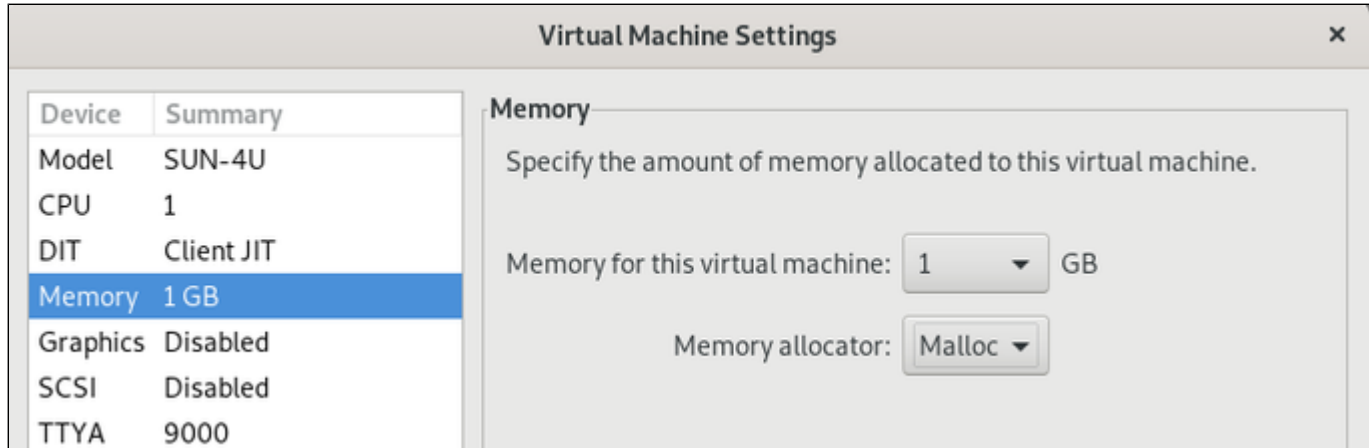
The following table lists each of the fields available in the DIT Server JIT mode and describes their use.

Virtual machine server JIT configuration fields	
Field	Description
<b>Virtualization Mode</b>	<p><b>Emulation</b> selects the Charon-SSP/4U or Charon-SSP/4V emulator, <b>Intel VT-x/EPT</b> informs the manager that the hardware prerequisites for running Charon-SSP/4U+ or Charon-SSP/4V+ are met.</p> <p>This option is inactive on Charon-SSP/4M or if Charon-SSP/4U+/4V+ is not installed.</p> <p>Attempting to run Charon-SSP/4U+/4V+ on insufficient hardware will cause the instance to exit with an error message. For example,  “MMU module insertion failed, please check if VT-X is enabled in BIOS”  or  “The host CPU doesn’t support Intel VT-x / EPT”).</p> <p>Check if your AWS instance runs on dedicated hardware if you encounter such errors.</p>
<b>DIT Optimization</b>	<p>This option controls the Dynamic Instruction Translation (DIT). DIT is a just in time compilation technology to dynamically optimize the SPARC instruction execution on x86-64 platforms. It can be set to <b>OFF</b>, <b>Client JIT</b>, or <b>Server JIT</b>. The remainder of this table describes the <b>Server JIT</b> parameters (<b>not available on Charon-SSP/4M</b>).</p>
<b>DIT page size</b>	<p>This option controls the size of the translation buffer holding the translated binary code that results from the DIT optimization. It can be increased to a maximum of 2048KB. This parameter should only be changed if the log file indicates that the DIT optimization was disabled because the translation buffer size was too small</p>
<b>DIT code cache</b>	<p>Size of cache between 1024MB and 8192MB in steps of 1024MB.</p>
<b>FP boost ratio</b>	<p>Defines the level of floating-point optimization. The parameter can be set to a value from 0 to 100. The default is 0 (= no boost). If customer applications are likely to benefit from floating point optimization, the value can be increased.</p>
<b>INT boost ratio</b>	<p>Defines the level of integer operation optimization. The parameter can be set to a value from 0 to 100. The default is 100 (= maximum boost). The higher the value the more resources are required. Hence high values are likely to provide most benefit if the guest system applications run for a long time.</p>

## Memory Configuration

To view or change the current memory configuration, select **Memory** in the left-hand pane of the Settings window.

The example below shows the Charon-SSP/4U configuration screen:



The following table lists each of the fields available in the memory configuration window and describes their use.

Virtual machine memory configuration fields	
Field	Description
<b>Memory for this virtual machine</b>	<p>Set the amount of RAM allocated to the virtual SPARC machine. Memory must be allocated in certain increments. The allocation rules for each virtual machine family are as follows:</p> <ul style="list-style-type: none"> <li>• SUN-4M: 64MB, 128MB, 256MB and 512MB</li> <li>• SUN-4U: 1 to 128GB in 1GB increments</li> <li>• SUN-4V: 1 to 1024GB in 1GB increments (not a drop-down list but manual entry). Actual limits are different depending on guest OS: Solaris 10: 1TB, Solaris 11: 512 GB. The GUI allows higher values, but this is for future use.</li> </ul>
<b>Memory allocator</b>	<p>This option specifies the memory allocation method used for the virtual machine. The default is <b>malloc</b>. It is appropriate for most cases. Please contact Stomasys if your environment has special memory requirements. Options:</p> <ul style="list-style-type: none"> <li>• Malloc: all virtual machine RAM is allocated from system heap.</li> <li>• Mmap: all virtual machine RAM is allocated from file backed virtual memory by memory mapping.</li> </ul>

## Graphics Configuration

**⚠ Not applicable to Charon-SSP/4V(+).**

### Contents

- Configuration Steps in Charon-SSP Manager
- Configuration Steps in Host and Guest Systems

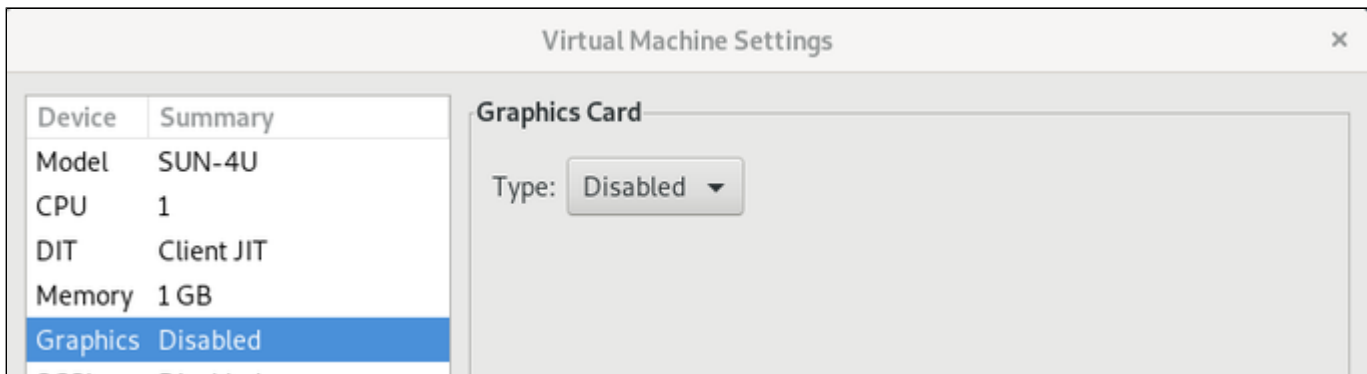
#### Please note:

- The graphical performance depends on many parameters, for example, the performance of host system, emulated system, and network. One important requirement is that the round-trip time of the network connection between display device and emulated Solaris system running on the AWS instance should not be more than 20ms. For every use case, a test is required to evaluate the suitability for the specific customer environment.
- If the integrated SSH tunnel of the Charon-SSP Manager is used, the ports used for mouse and keyboard events are redirected through the tunnel. The remote port is not redirected. Therefore, in such situations firewalls and security groups must allow the port. If a VPN connection is used to communicate with the Charon host and guest in the AWS environment, all connections can be routed through the VPN (see [SSH VPN - Connecting Charon Host and Guest to Customer Network](#)).

### Configuration Steps in Charon-SSP Manager

To view or change the current graphics emulation configuration, select **Graphics** in the left-hand pane of the Settings window.

This opens the graphics configuration window. As shown below, the graphics emulation is disabled by default:

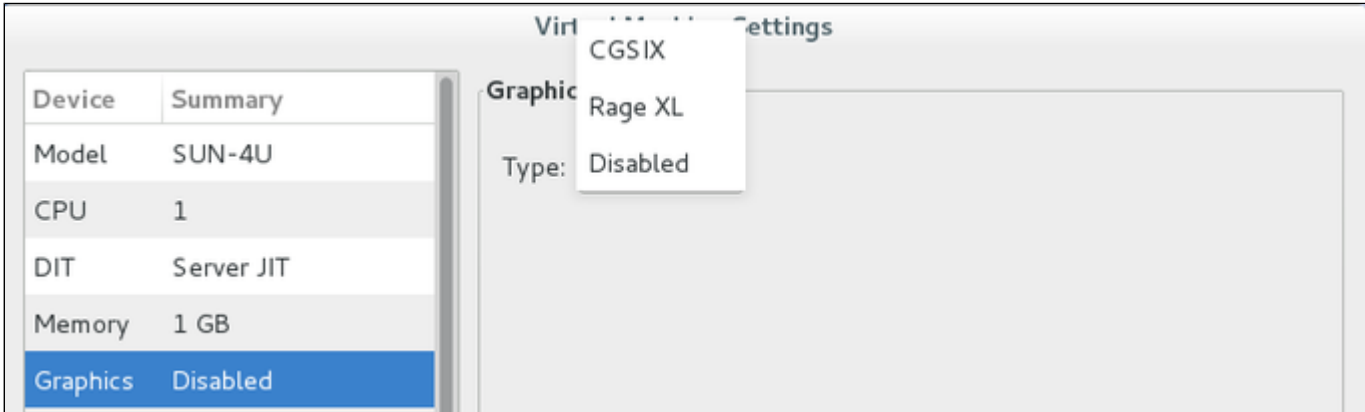


To enable it, select a graphics card type from the drop-down menu. Possible values are

- CGSIX or CGTHREE on Charon-SSP/4M
- CGSIX or Rage XL on Charon-SSP/4U(+)

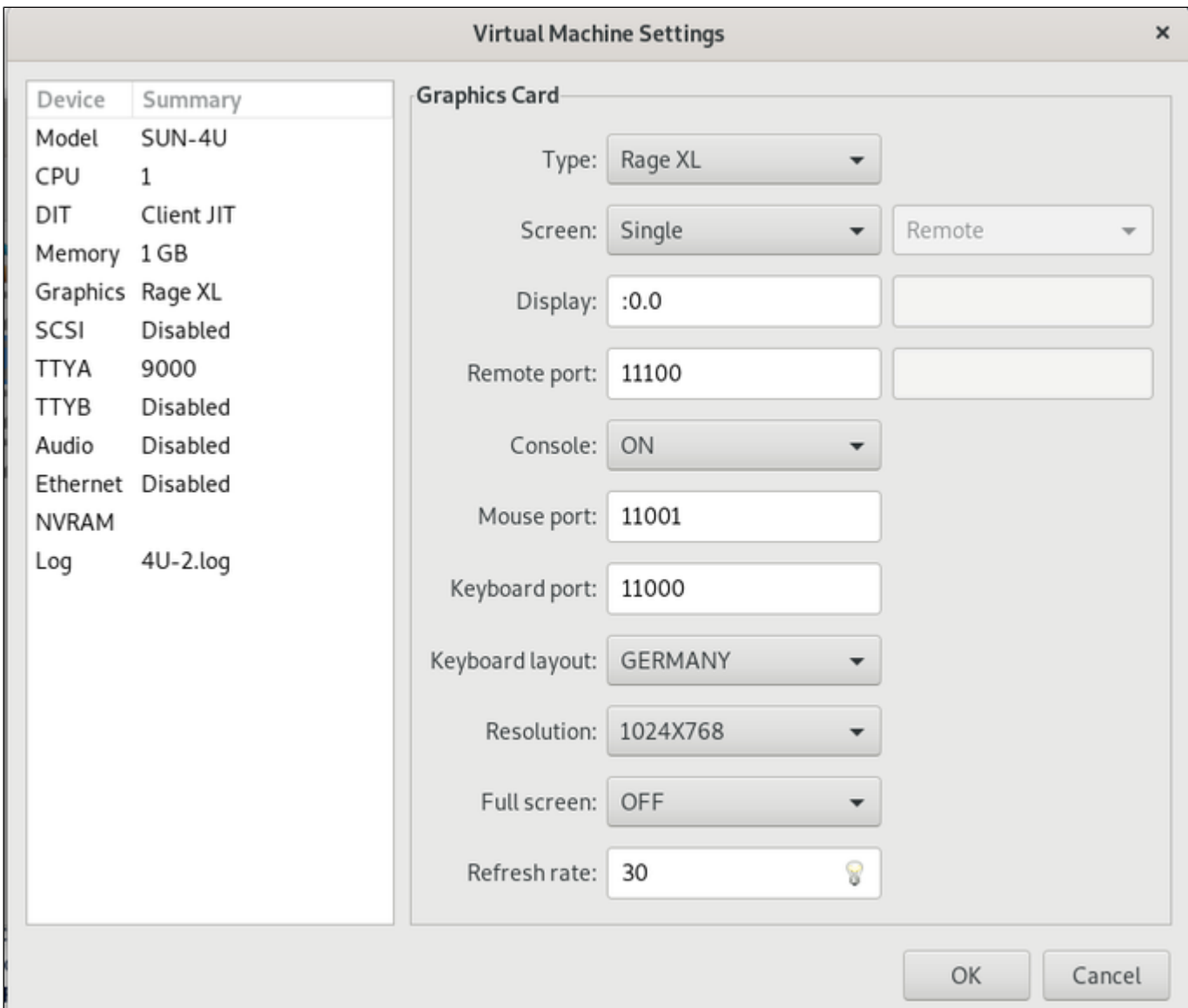
The CGTHREE adapter is a graphic adapter with frame buffer; the CGSIX adapter is a graphic adapter with frame buffer and 2D acceleration, the Rage XL is a graphic adapter with frame buffer, 8MB Memory, and 2D acceleration.

The following image shows the options on a SUN-4U system:



To start configuring a graphics device, select one of the supported graphics options.

This opens the configuration window as shown in the Charon-4U example below:



The following table describes the graphics configuration options:

Virtual machine graphics configuration fields	
Field	Description
<b>Type</b>	Selection of supported graphics options: <ul style="list-style-type: none"> <li>• <b>CGSIX</b> or <b>CGTHREE</b> on Charon-SSP/4M</li> <li>• <b>CGSIX</b> or <b>Rage XL</b> on Charon-SSP/4U(+)</li> <li>• <b>Disabled</b></li> </ul>
<b>Screen</b>	Number of screens: <ul style="list-style-type: none"> <li>• <b>Single</b>: use one screen</li> <li>• <b>Dual</b>: use two screens</li> </ul> Location for displaying the graphics output: <ul style="list-style-type: none"> <li>• <b>Local</b>: disabled for Charon-SSP AWS, only remote display is possible.</li> <li>• <b>Remote</b>: graphics output is displayed on a remote system</li> </ul>
<b>Display</b>	Defines the DISPLAY variable to be used by the graphics output. The default value is “:0.0” (display 0 screen 0). This value has to be set to match the display configuration on the system where the graphics output is to be displayed.  If a dual screen configuration is selected, two display variables can be defined.
<b>Remote port</b>	Defines the port(s) to which a Charon-SSP Manager on a remote system connects to display the graphics output of the guest system. The default value is 11100 for a single screen configuration. For a dual screen configuration, the default ports are 11100 and 11101. Only relevant for remote screen configurations. The ports must be unique on the host system.
<b>Console</b>	Defines whether the graphical device should act as the console of the guest system. <ul style="list-style-type: none"> <li>• <b>ON</b>: the graphics device is the system console of the guest system (default). In this case, the serial console window in Charon-SSP Manager is not available.</li> <li>• <b>OFF</b>: the serial console in Charon-SSP Manager or an external serial console is used.</li> </ul>
<b>Mouse port</b>	Port for transmitting mouse event data. Default 11001. The port must be unique on the host system.
<b>Keyboard port</b>	Port for transmitting keyboard event data. Default 11000. The port must be unique on the host system.
<b>Keyboard layout</b>	The appropriate keyboard layout can be selected from the drop-down menu.  The META key of the Solaris keyboard is mapped to the Windows key on the PC keyboard.
<b>Resolution</b>	The appropriate resolution can be selected from the drop-down menu.  CG3 supports 800 x 600, 1024 x 768 and 1152 x 900;  CG6 and Rage XL support 1024 x 768, 1152 x 900, 1280 x 1024, and 1600 x 1280.
<b>Full screen</b>	If set to <b>ON</b> , the emulated graphics device will start in full-screen mode. Best results are achieved if the resolution of the host system display matches the resolution of the emulated device. To toggle between full-screen and normal mode during operation use the key combination <b>CTRL+SHIFT+F</b> after clicking into the window to give it focus.
<b>Refresh rate</b>	The refresh rate for the graphical output can be set to a value between 20 and 100. Charon-SSP/4U(+) only.

#### Mouse and keyboard capture and release:

- When you click into the graphics device window, it will **capture mouse and keyboard**.
- To release mouse and keyboard press **LEFT-CTRL+ESC**.

Use the toggle key combination (**CTRL+SHIFT+F**) to switch between normal window mode and full-screen mode (first click into the graphics window to make sure it has focus).

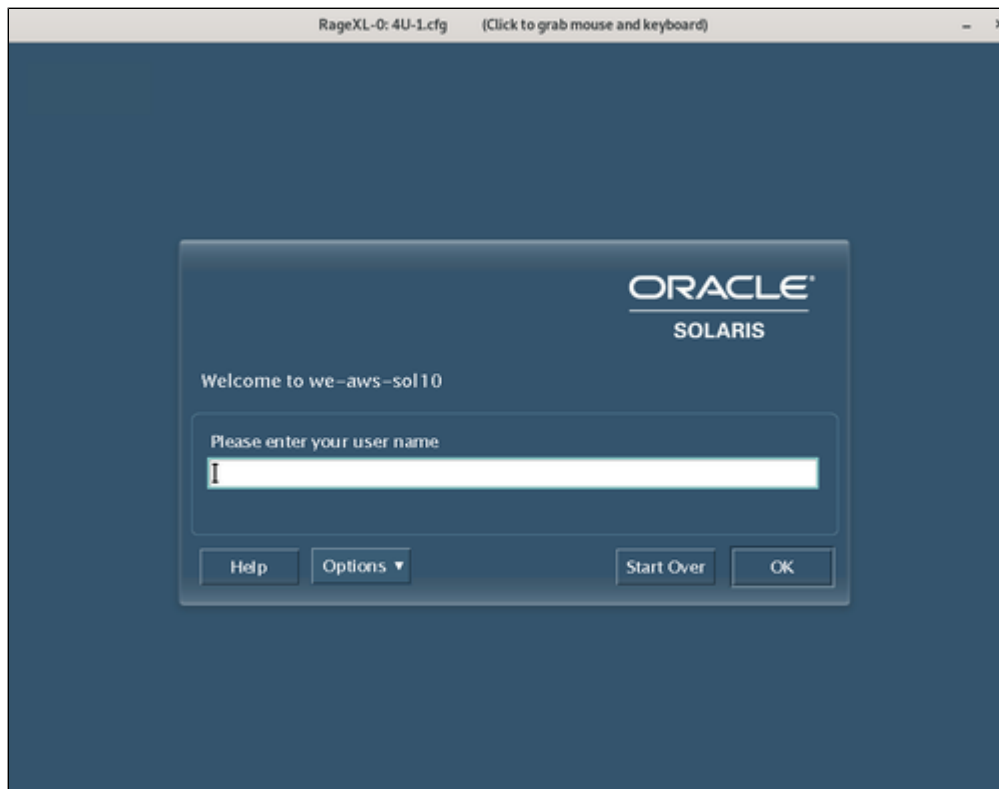


## Configuration Steps in Host and Guest Systems

In addition to configuring the graphics emulation in the Charon-SSP Manager, there are several prerequisites:

Virtual machine graphics configuration fields	
Field	Description
Host system	Ensure that the required ports for display, mouse, and keyboard events are not blocked by a firewall or the security group.
Solaris guest	<ul style="list-style-type: none"> <li>Ensure that the required drivers (SUNWcg6*, SUNWdfb*, SUNWm64*) are installed on the system. They are part of the standard system and are normally installed if the matching devices are found. Should they be missing, the packages can be installed or the drivers can be copied from the installation CD (must be same version and patch level as on Solaris guest). The names of the drivers are <i>cgsix</i>, <i>cgthree</i>, and <i>m64</i>.</li> <li>After configuring the graphical device or changing the configuration between single and dual screen configurations, reboot the system with the <b>boot &lt;device&gt; -r</b> option to create the correct device special files and the <i>/dev/fb*</i> links that point to these devices.</li> <li>If the Solaris graphical user interface is to be used on the device, ensure that <ul style="list-style-type: none"> <li><i>/usr/openwin/bin</i> is in the path of the user,</li> <li><b>dtlogin</b> is enabled at system start (e.g., on Solaris 2.6: <b>/usr/dt/bin/dtconfig -e</b>)</li> </ul> </li> <li>Ensure that the X-server starts on the correct fb device (default <i>/dev/fb</i>). Otherwise, it may fail with the message that the device does not exist. If such a problem occurs, perform the following steps: <ul style="list-style-type: none"> <li>Create the directory <b>/etc/dt/config</b>.</li> <li>Copy <b>/usr/dt/config/Xservers</b> into <b>/etc/dt/config</b>.</li> <li>Modify the X-server start line to contain the correct <i>/dev/fb*</i> line. You can find the existing framebuffer device links using <b>ls -l /dev/fb*</b>. Sample line in the Xservers file:  <b>:0 Local local_uid@console root /usr/openwin/bin/Xsun :0 -dev /dev/fb0 nobanner</b>            If you use a dual monitor configuration, you have to add a second <b>-dev</b> entry. Please note that on Solaris 10 the path for the Xserver is <i>/usr/X11/bin/Xserver</i>.</li> </ul> </li> </ul>

Once the configuration is correct, the graphical login screen will be shown when the guest system boots:



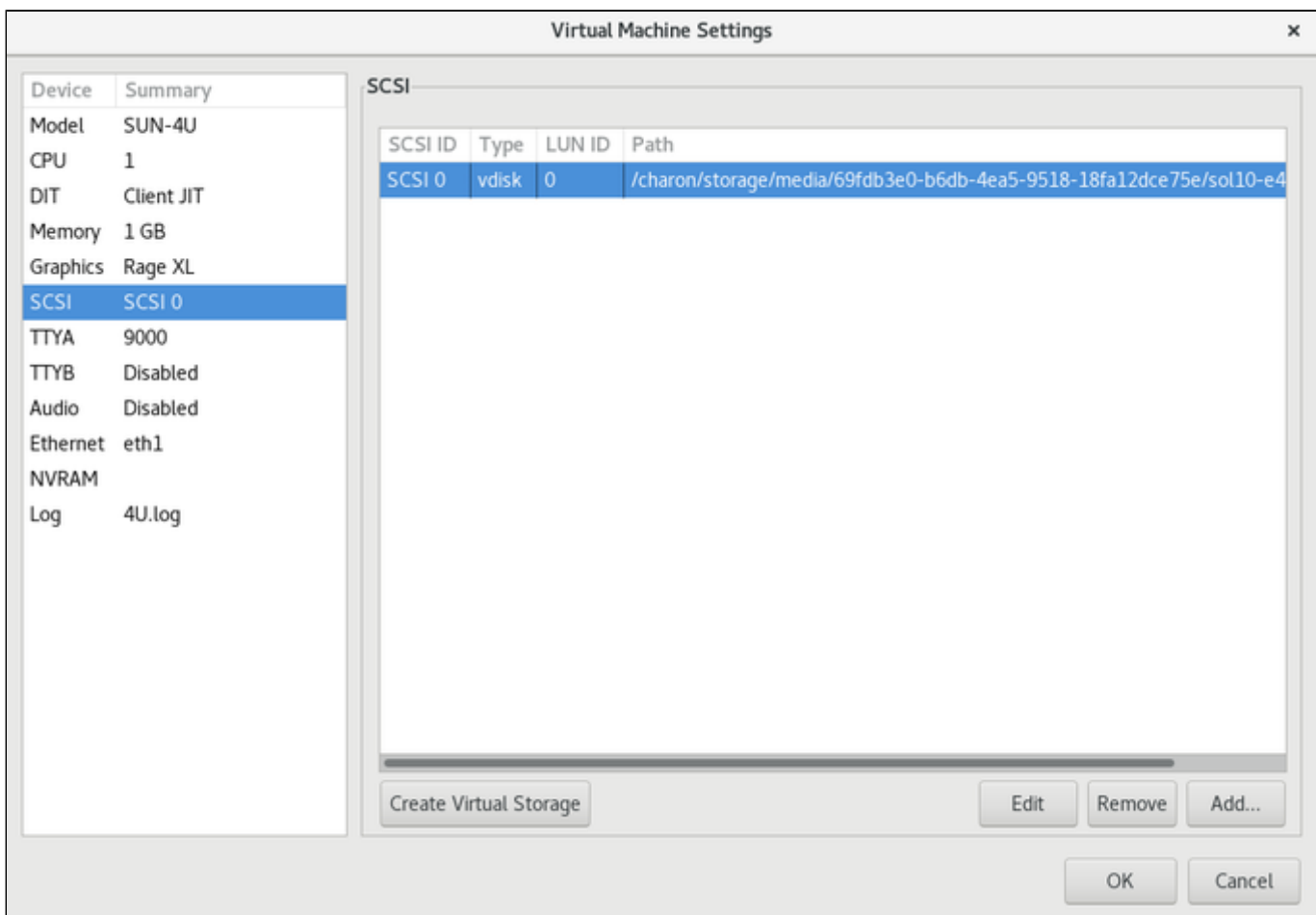
## SCSI Storage Configuration

### Contents

- SCSI Configuration Window Overview
- Creating a New Virtual Disk Container File
- Creating a New Virtual Tape Container File
- Adding or Editing a Virtual SCSI Device
  - Physical Disk Parameters on Charon-SSP
- Removing a Virtual Storage Device

### SCSI Configuration Window Overview

To view or change the current virtual machine SCSI configuration, select **SCSI** in the left-hand pane of the Settings window. This opens the SCSI configuration window similar to the one shown below.



From this window, you can create virtual disk and tape container files using the **Create Virtual Storage** button. You can also attach virtual storage devices (both physical devices and container files) to the virtual machine (**Add** button). When selecting an existing virtual storage device, you can edit or remove it (**Edit** and **Remove** buttons are only visible if a device is selected).

**i** The **Create Virtual Storage** option is also available in the **Tools** menu of the Charon-SSP Manager. The functions provided are identical to the functions provided via the **Virtual Machine Settings** window shown above.

## Creating a New Virtual Disk Container File

It is often convenient to use container files for virtual disk and tape devices. This section describes how to create disk container files.

To create a virtual disk container file, **click** on **Create Virtual Storage** in the SCSI device **Virtual Machine Settings** window. This displays the **Create Virtual Storage** dialog opened on the virtual disk tab as shown below.

To create a virtual disk container file, follow the instructions below:

1. Select the virtual disk type from the drop-down list **Virtual disk type**.
  - If you select a preconfigured **Virtual disk type** the **Block number** field is updated to match that model.
  - If you specify the type of **Custom**, enter the container file size as a number of 512-byte blocks at the field **Block number**. The size of the custom disk is shown in KB/KiB, MB/MiB, or GB/GiB depending on the configured number of blocks.
2. Specify a name for the virtual disk container file in the field **Virtual disk name**.
3. Select the location on the host filesystem for the container file by **clicking** on **the location** selection button and selecting the correct path. The default is different depending on the Charon-SSP product. For Charon-SSP AWS, it is recommended to store virtual disk and tape containers on a separate EBS volume.
4. Click on **Create** to create the virtual disk container file. Depending on the size of the container file, this may take some time.

The screenshot shows the 'Create Virtual Storage' dialog with the following fields and values:

- Virtual Disk type:** SUN1.05 1.1 GB
- Virtual disk name:** datadisk.vdisk
- Location:** storage
- Virtual disk geometry:**
  - Block number: 2054304
  - Block size: 512 Bytes
  - Disk size: 1.1 GB/1003 MiB
- Progress bar:** 0%
- Buttons:** Create, Close

**⚠** Before the disk can be used by the Solaris guest system, it must be added to the system configuration and formatted/labeled by the Solaris guest according to the customer specific requirements.

## Creating a New Virtual Tape Container File

To create a virtual tape container file, **click** the **Create Virtual Storage** button in the SCSI device **Virtual Machine Settings** window. This opens the **Create Virtual Storage** window. Select the **Virtual Tape** tab.

To create a virtual tape container, follow the instructions below:

1. Specify a name for the virtual tape container file in the field **Virtual tape name**.
2. Select the location on the host filesystem for the container file by **clicking** on the location selection button and selecting the correct path.
3. Specify a size for the virtual tape file in megabytes (MB) in the field **Tape size**. The vtape file will expand automatically if more space is needed while writing to the tape.
4. **Click** on **Create** to create the virtual tape container file. Depending on the size of the container file, this may take some time.

### Using a virtual tape:

Once a virtual tape device has been created, it can be added to the Charon-SSP configuration and used by the Solaris guest system. To simulate “**swapping a tape**” during guest system operation, the following steps are required:

1. **Guest system:** rewind tape if required, write content to it, and “eject” it:

```
# mt -f <device-name> rewind
# tar -cvf <device-name> <files-to-save>
# mt -f <device-name> offline
```

2. **Host system:** use **sftp** to rename/copy the original container file and to copy a new empty file with the same name in its place.

3. **Guest system:** display tape status (thereby loading the new file), rewind tape if required, and write content to it:

```
# mt -f <device-name> status

# mt -f <device-name> rewind

# tar -cvf <device-name> <more-files-to-save>
```

**i** Solaris tape device names have the format `/dev/rmt/<device>` where device can be a digit (e.g., `/dev/rmt/0`) or a combination of digits and certain letters (e.g., `/dev/mnt/0n` is the first drive set to no rewind).

Should the devices not exist after adding a virtual tape drive, boot the emulated SPARC guest system with the `-r` (reconfigure) parameter. Example:

```
boot disk0 -r.
```

## Adding or Editing a Virtual SCSI Device

To **add** a new virtual disk device, **click the Add button**.

To **modify** an existing virtual disk device, select it from the list of configured devices and **click the Edit button**. The **Edit** button appears when an existing virtual disk is selected.

In both cases, a window similar to the one below opens with the configuration parameters of the virtual SCSI device.

Charon-SSP does not place any restrictions on the bus and SCSI target ID used for emulated SCSI devices, e.g., a virtual CD-ROM. However,

- some versions of Solaris may expect the boot CD-ROM device to be on the external bus (if available) and SCSI ID 6, and
- the emulated Charon-SSP/4V console environment expects the CD-ROM device on the internal bus and SCSI ID 6.

If you encounter the problem that the CD-ROM is not found when trying to boot from it, verify its expected location in the OBP environment (using the `devalias` command).

The following table lists the fields in the **Add/Edit SCSI Device** configuration window and describes their use.

Add/Edit virtual SCSI device configuration fields											
Fields	Description										
<b>SCSI bus</b>	Specify either the <b>Primary SCSI Bus</b> or the <b>External SCSI Bus</b> .  Please note: Charon-SSP/4M has only one SCSI bus.										
<b>SCSI ID</b>	SCSI device target ID: <ul style="list-style-type: none"> <li>• <b>Charon-SSP/4M</b>: Acceptable values are a 3-bit narrow SCSI device IDs between 0 and 7.</li> <li>• <b>Charon-SSP/4U</b> and <b>Charon-SSP/4V</b>: Acceptable values are a 4-bit wide SCSI device IDs between 0 and 15.</li> </ul> <p>⚠ The SCSI target ID 7 is reserved for the SCSI host bus adapter. It cannot be used for a user-configurable SCSI device.</p>										
<b>LUN ID</b>	SCSI device LUN ID. A SCSI device is identified by a combination of bus, target ID (SCSI ID), and LUN ID. This parameter must be configured to match the storage device configuration. Valid IDs are 0 through 7. Default value is 0.  ⚠ The LUNs configured for one SCSI target ID must belong to the same virtual device type.										
<b>Removable</b>	Default: OFF. If enabled, the emulator will start even if the device/file does not exist on the host.										
<b>SCSI device type</b>	Drop-down list of configurable device types. Available device types: <ul style="list-style-type: none"> <li>• <b>Virtual Disk</b>: Virtual disk device backed by a container file.</li> <li>• <b>Virtual CDROM</b>: Virtual CD-ROM device backed by a container file.</li> <li>• <b>Virtual Tape</b>: Virtual tape device backed by a container file.</li> <li>• <b>Physical Disk</b>: Virtual disk device mapped to a physical disk or a physical disk partition on the host system.</li> </ul>										
<b>SCSI device path</b>	Click on the <b>path button</b> to specify the location of the virtual SCSI device. This will open a file browser. To sort the file browser display by name, click on the corresponding heading.  Select an appropriate device or file using the file browser, or type the correct name in the file name field.  <b>Note</b> : if you manually enter a device name instead of selecting a device from the file browser window, make sure that the file/device exists (relative to the path of the opened file browser) or is set to removable.  The list below shows sample device paths for each <b>SCSI device type</b> option: <table border="1" data-bbox="289 1150 813 1394"> <thead> <tr> <th>Device type</th> <th>Sample device path</th> </tr> </thead> <tbody> <tr> <td>Virtual Disk</td> <td>/usr/local/vm/lela/scsi0.vdisk</td> </tr> <tr> <td>Virtual CDROM</td> <td>/usr/local/share/iso/sunos_4.1.4.iso</td> </tr> <tr> <td>Virtual Tape</td> <td>/usr/local/vm/lela/scsi1.vtape</td> </tr> <tr> <td>Physical Disk</td> <td>/dev/sda</td> </tr> </tbody> </table>	Device type	Sample device path	Virtual Disk	/usr/local/vm/lela/scsi0.vdisk	Virtual CDROM	/usr/local/share/iso/sunos_4.1.4.iso	Virtual Tape	/usr/local/vm/lela/scsi1.vtape	Physical Disk	/dev/sda
Device type	Sample device path										
Virtual Disk	/usr/local/vm/lela/scsi0.vdisk										
Virtual CDROM	/usr/local/share/iso/sunos_4.1.4.iso										
Virtual Tape	/usr/local/vm/lela/scsi1.vtape										
Physical Disk	/dev/sda										

## Physical Disk Parameters on Charon-SSP

The Charon-SSP virtual machines offer additional options when adding physical disks as virtual SCSI devices. The windows for adding a new device and for editing an existing device contain the same fields. The configuration windows are different for Charon-SSP/4U/4V and Charon-SSP/4M because only Charon-SSP/4U and Charon-SSP/4V support a second SCSI bus. The two different configuration windows are shown below:

Add physical disk on Charon-SSP/4U/4V	Add physical disk on Charon-SSP/4M

The following table describes the additional parameters available for physical disks on Charon-SSP:

Additional physical disk parameters	
Field	Description
<b>Pass through</b>	<p>You can select <b>OFF</b> (default) or <b>ON</b>. SCSI pass-through is used to allow direct access to SCSI devices. Such devices can be locally or remotely connected SCSI storage devices (e.g., local disks, iSCSI connected disks, Fibre Channel disks, etc.) and other SCSI devices that support the SCSI command set. On the host side, this feature depends on the generic SCSI driver (SG) capabilities of the host operating system. The emulator does not depend on particular adapter types.</p> <p>This feature is useful, for example, for using shared disks in cluster environments (fencing / persistent reservations) and special SCSI peripherals, such as tape robots or SCSI-connected serial devices and scanners.</p>
<b>Serial Number</b>	<p>The serial number is a physical characteristic of hard disks and is used mainly to persistently and unambiguously identify iSCSI mapped disks (and possibly Fibre Channel disks), for which the device identification on the host (i.e. /dev/sdX) may change when the host system reboots. If the <b>Serial Number</b> field is enabled, the field <b>SCSI device path</b> is disabled.</p> <p>You can find the serial number using the <b>Storage Manager</b> that is started from the <b>Tools &gt; AWS Cloud</b> menu of the Charon-SSP Manager.</p>

## Removing a Virtual Storage Device

To remove a virtual storage device, **select** the device in the **Virtual Machine Settings** SCSI configuration window, then **click** the **Remove** button. The device is removed immediately from the configuration. The Charon-SSP Manager does not ask for confirmation.

**!** If the virtual SCSI storage device is attached to a container file, the file itself is not removed with the configuration.

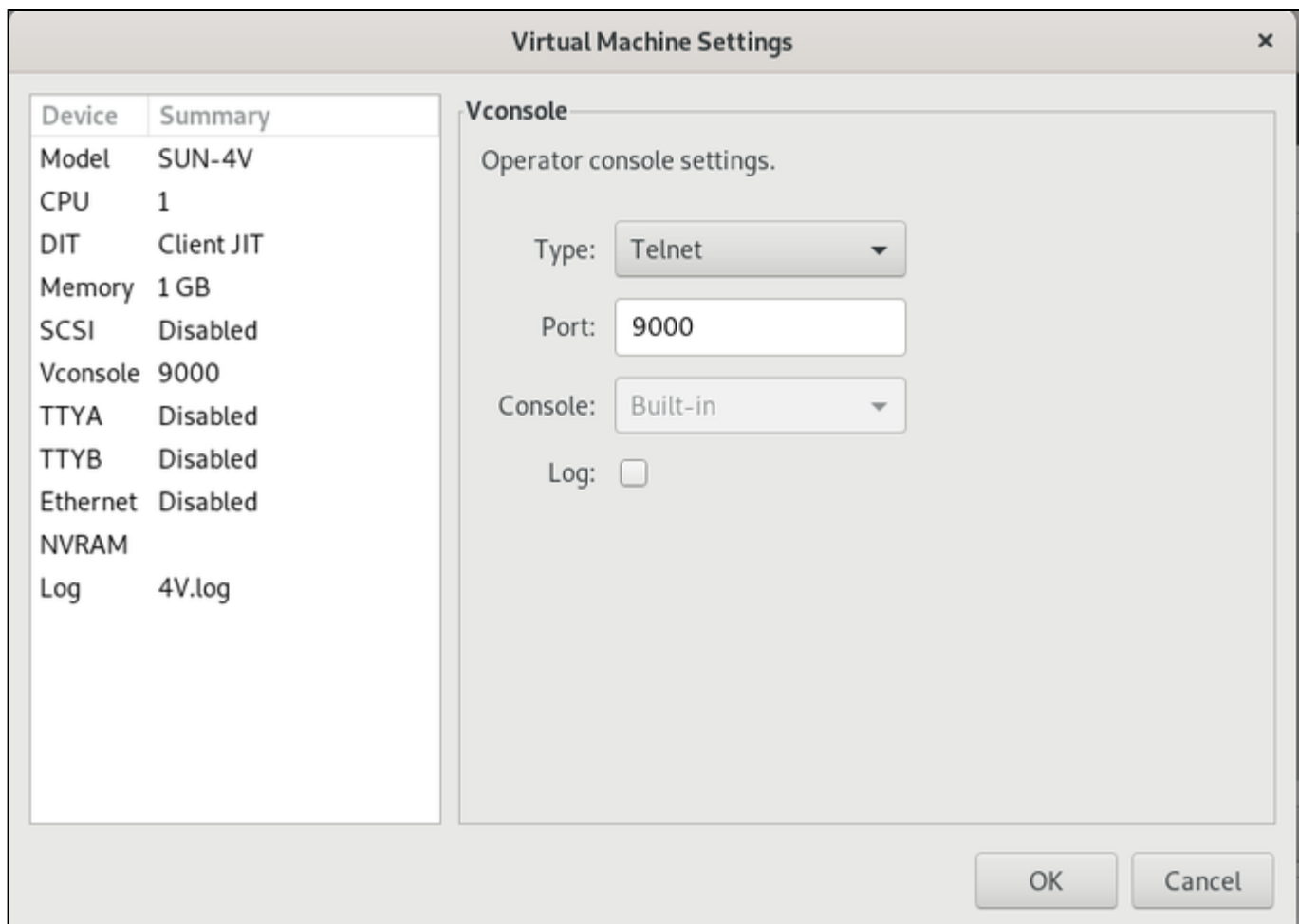
## Serial Line Configuration

### Contents

- Vconsole Configuration (Charon-SSP/4V only)
  - Vconsole Network Configuration
    - Network Vconsole Configuration Options and their Functions
  - Vconsole Physical Line Configuration
- TTYA Configuration
  - TTYA Physical Line Configuration
  - TTYA Network Configuration
    - TTYA Network Console Configuration Options and their Functions
- TTYB Configuration

### Vconsole Configuration (Charon-SSP/4V only)

The Vconsole represents the serial console device of a Charon-SSP/4V instance. To view or change the current virtual machine console configuration, select **Vconsole** in the left-hand pane of the Settings window. This opens the **Vconsole** configuration window, shown below.



The emulated terminal type can have one of four values as described below. Use the **Type** drop-down list to set the value.

- **TCP raw:** configure the console device as a network device (TCP socket) without any protocol enabled.
- **Telnet:** configure the console device as a network device (TCP socket) with the telnet protocol enabled.
- **Physical:** configure the console device as physical terminal directly attached to the host system. For an AWS instance, this could be a virtual serial line provided by a serial line server (terminal server).
- **Disabled:** disable the virtual console device entirely.



## Vconsole Network Configuration

When configuring a network console device, the user can select one of two modes:

- TCP raw (serial line without protocol), or
- Telnet (serial line with telnet protocol support).

## Network Vconsole Configuration Options and their Functions

### Port:

This option specifies the TCP/IP port to use when listening for incoming console client connections. A different port must be specified for each network console and serial port used on the same Charon-SSP host system.

 Using a port that is already in use results in error messages in the virtual machine log file similar to the following.

```
2019-08-27 09:54:03 ERROR SocketIO Failed to open socket server (port: 9000).
2019-08-27 09:54:03 ERROR Serial  Fail to initialize serial device.
2019-08-27 09:54:03 ERROR VM      Failed to initialize VCONSOLE
```

To access the console of a guest system across the network without, make sure the port configured for the console is permitted by any intermediate firewalls and/or security groups. If using the built-in console function of the Charon Manager in combination with the Charon Manager SSH tunnel, the console port is redirected through this SSH tunnel.

### Console:

For Charon-SSP AWS, the value of this parameter is fixed to **Built-in**. The built-in console is displayed and accessible from the **Console** tab in the Charon Manager. The console process listens on TCP port 9000 by default.

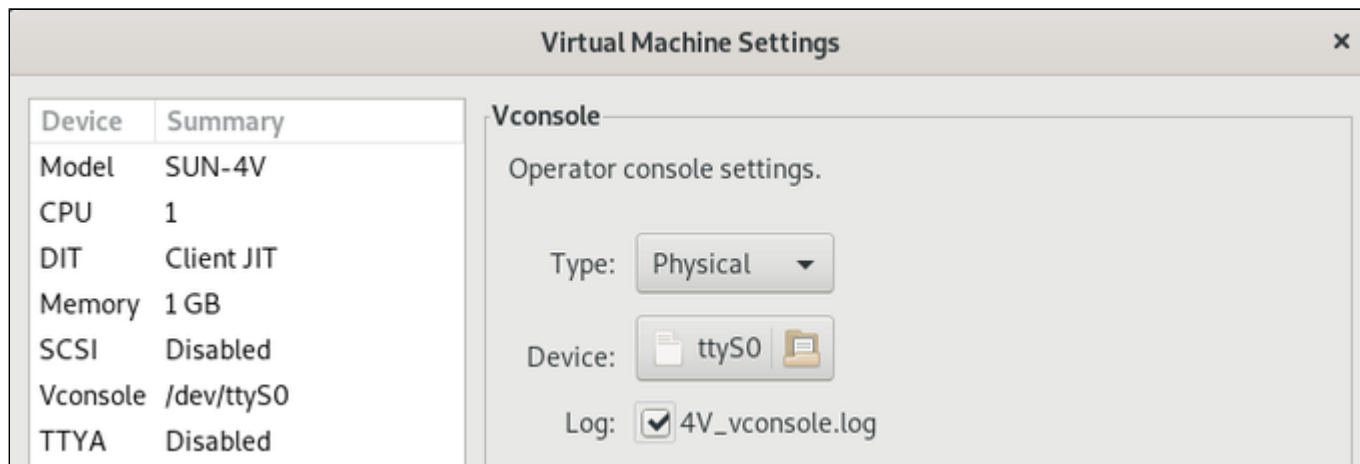
### Log:

When this box is checked, Charon-SSP writes a console log file.

**Please note:** the log file configured here is separate from the file the Charon-SSP Manager uses to cache the console output for the built-in serial console of the Charon-SSP Manager.

## Vconsole Physical Line Configuration

The image below shows the configuration window for a physical console device of a Charon-SSP/4V system.

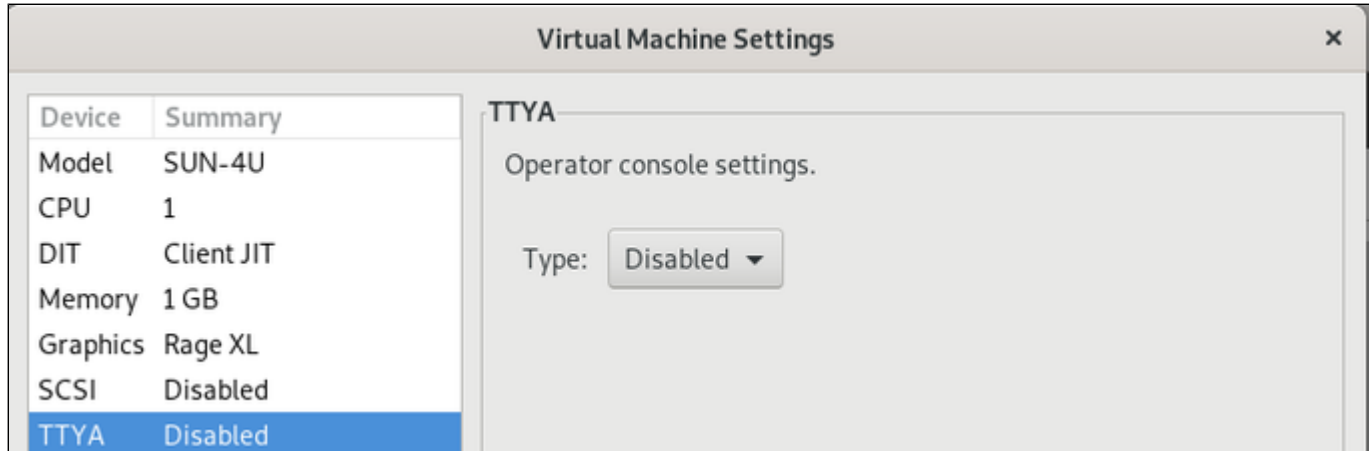


Physical serial console configuration options:

- **Device:** opens a file browser to let the user select from the directly attached serial ports available on the host system (*tty\** devices).
- **Log:** used to enable and disable the console log.

## TTYA Configuration

To view or change the current virtual machine console configuration, select **TTYA** in the **Device** column of the left-hand pane of the configuration window. This opens the **TTYA** configuration window, shown below. In this example, TTYA is disabled.



**i** On Charon-SSP/4U and Charon-SSP/4M, TTYA can be configured as the serial console or, if the graphical device is configured to be the system console, TTYA can be used as a normal serial line. On Charon-SSP/4V systems, it can only be used as a normal serial line.

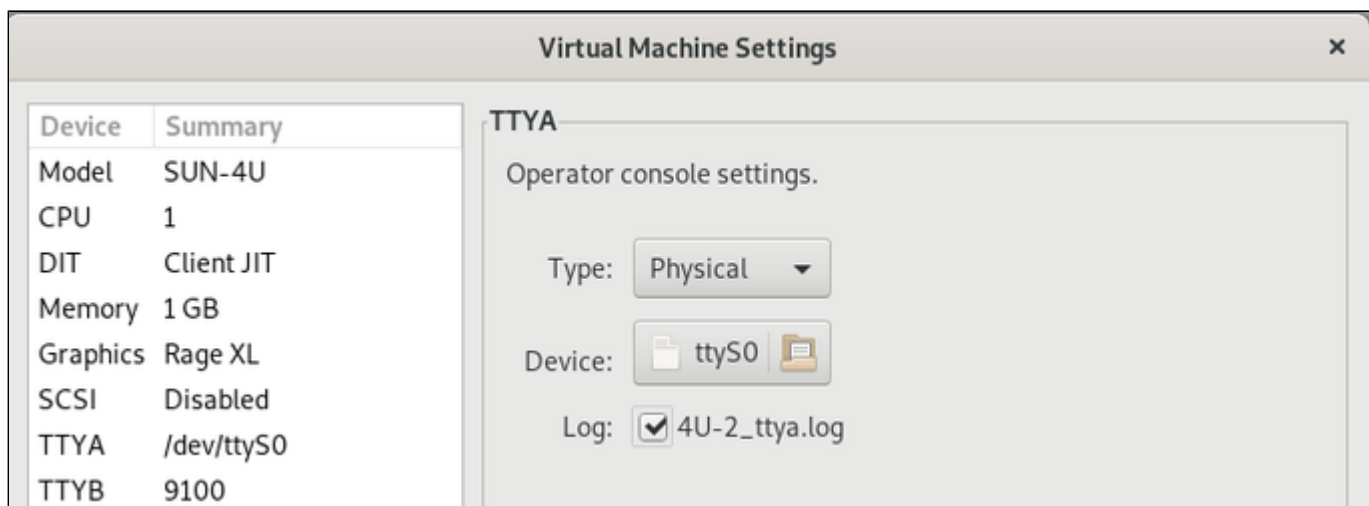
The emulated terminal type can have one of four values as described below. Use the **Type** drop-down list to set the value.

- **TCP raw**: configure the console device as a network device (TCP socket) without any protocol enabled.
- **Telnet**: configure the console device as a network device (TCP socket) with the telnet protocol enabled.
- **Physical**: configure the console device as physical terminal directly attached to the host system. For an AWS instance, this could be a virtual serial line provided by a serial line server (terminal server)
- **Disabled**: disable the virtual console device entirely.

The following sections describe the specific configuration details of physical and network consoles.

## TTYA Physical Line Configuration

The image below shows the configuration window for a physical console device of a Charon-SSP/4U system.



Physical serial console configuration options:

- **Device**: opens a file browser to let the user select from the directly attached serial ports available on the host system (*tty\** devices).
- **Log**: used to enable and disable the console log.

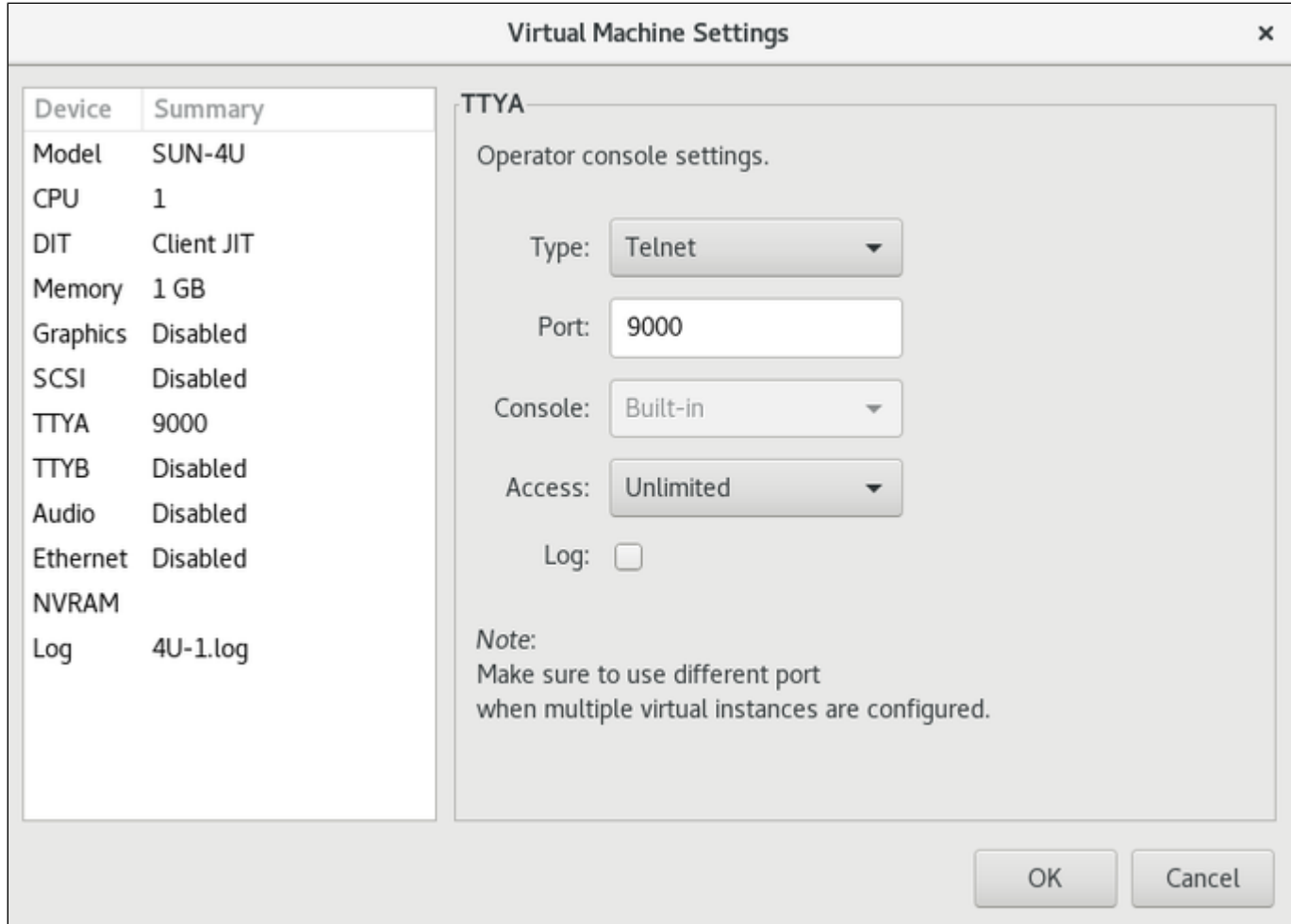
## TTYA Network Configuration

When configuring a network console device, the user can select one of two modes:

- **TCP raw** (serial line without protocol), or
- **Telnet** (serial line with telnet protocol support).

**Please note that TTYA cannot be used as the serial system console for Charon-SSP/4V systems. Such systems must use the Vconsole device.**

The image below shows the configuration window for a network console device with telnet protocol support on a Charon-SSP/4U system:



## TTYA Network Console Configuration Options and their Functions

### Port:

This option specifies the TCP/IP port to use when listening for incoming console client connections. A different port must be specified for each network console and serial port used on the same Charon-SSP host system.

 Using a port that is already in use results in the following error messages in the virtual machine log file.

```
2015-03-23 11:45:50 ERROR SocketIO Failed to open socket server!
2015-03-23 11:45:50 ERROR serial fail to init serial!
2015-03-23 11:45:50 ERROR vm Failed to initialize device:4
```

To access the console of a guest system across the network, make sure the port configured for the console is permitted by any intermediate firewalls and/or security groups. If using the built-in console function of the Charon Manager in combination with the Charon Manager SSH tunnel, the console port is redirected through this SSH tunnel.

### Console:

 Not applicable to Charon-SSP/4V. Please refer to the **Vconsole** section instead.

 Option is not visible if graphics device is configured with console enabled.

For Charon-SSP AWS, the value of this parameter is fixed to **Built-in**. The built-in console is displayed and accessible from the **Console** tab in the Charon Manager. The console process listens on TCP port 9000 by default.

### Access:

Possible values:

**Unlimited:** Connection to the console is possible via a remote network connection.

**Local only:** Connection to the console is only possible from the local host.

### Log:

When this box is checked, Charon-SSP writes a console log file.

**Please note:** the log file configured here is separate from the file the Charon-SSP Manager uses to cache the console output for the built-in serial console of the Charon-SSP Manager.

## TTYB Configuration

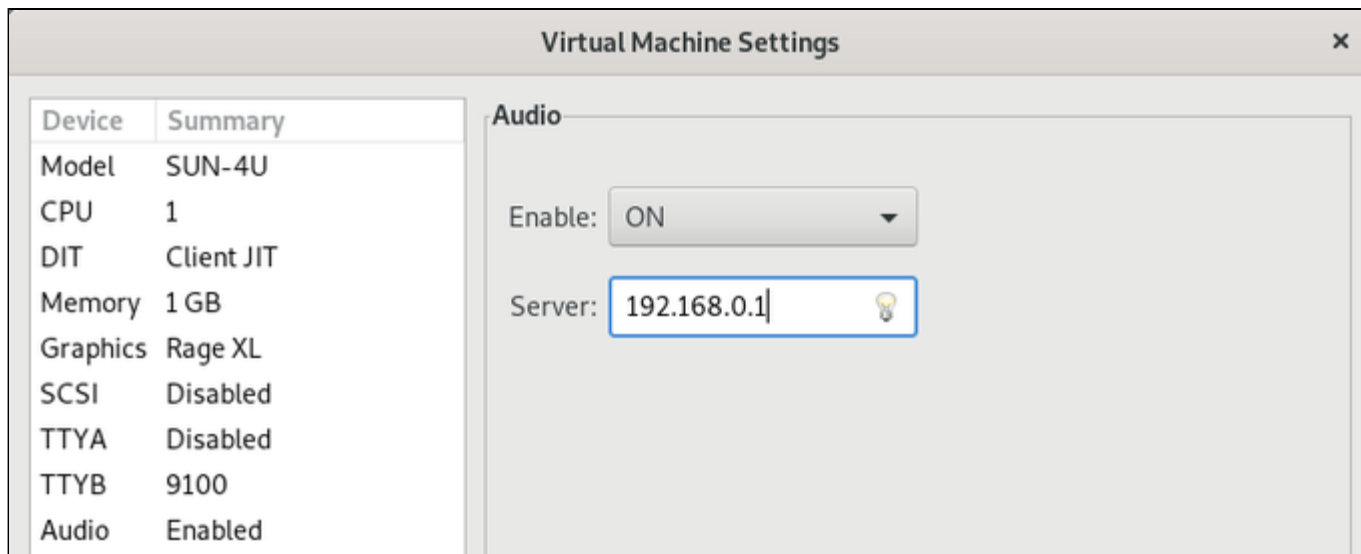
To view or change the virtual machine TTYB configuration, select **TTYB** in the **Device** column of the left-hand pane of the configuration window. The virtual TTYB serial device can be configured as both a physical or network connected device. The configuration of this device is very similar to TTYA. For further details related to configuring this device, consult the section [TTYA Configuration](#).

## Audio Configuration

⚠ Not applicable to Charon-SSP/4V.

⚠ The audio feature is supported across a VPN, but not across a NAT connection.

To enable, disable, or change an audio server for the emulated Solaris system, click on the option **Audio** in the left-hand pane of the Settings window:



The audio stream is mapped from the emulated DBRle device to a PulseAudio Server accessed remotely on TCP port 4713.

After enabling the functionality, you can set the **IP address** of the audio server in the **Server** field and click on **OK** to save the settings. The default is the local host system. The target IP address must point to a **directly reachable** audio server (no NAT). This is typically a system reachable via the VPN connection to the customer network.

⚠ **Currently, only PulseAudio on Linux is a supported audio server.**

The audio emulation emulates a DBRle SBUS adapter and supports the following features:


- CS4215 16-bit multimedia codec for mono and stereo audio playback and recording
- Audio data encoding: uLaw, aLaw, 8/16 bit linear
- Sample rates from 5513Hz to 48000Hz (Voice to DAT quality)
- Speakers volume, recording volume and MIC/speakers muting

**i** In addition to providing an emulated sound card to the Solaris guest operating system, the audio configuration also enables the **Keyboard Buzz** feature, i.e., it allows Solaris applications to create keyboard beeps.

**Prerequisites:**a) *Linux audio server*

On the Linux audio server, PulseAudio must be enabled for network access as shown below:

Configure PulseAudio on Linux for network access		
Step	Description	Command
1	Check if PulseAudio is installed on the system.	<pre># rpm -qa   grep -i pulseaudio</pre> <p>If the software is not installed, use</p> <pre># yum install pulseaudio pulseaudio-utils</pre> <p>to install it.</p>
2	Enable network access to the PulseAudio server.	<p>Edit the PulseAudio configuration file:</p> <p><i>If PulseAudio runs under the non-root account of the current desktop user (normal case, recommended):</i></p> <pre># vi /etc/pulse/default.pa</pre> <p><i>If PulseAudio is run as root user (system mode, not recommended, only useful in special cases – e.g., embedded use where no real local users exist):</i></p> <pre># vi /etc/pulse/system.pa</pre> <p>Add the following line if it does not already exist:</p> <pre>load-module module-native-protocol-tcp auth-anonymous=1</pre> <p>Save the file.</p>
3	Restart the PulseAudio server.	<p>If the <b>default.pa</b> file was modified, the following commands must be run as the non-root user under which PulseAudio was originally started.</p> <p>Stop the PulseAudio server:</p> <pre>\$ pulseaudio -k</pre> <p>Start the PulseAudio server:</p> <pre>\$ pulseaudio --start</pre> <p>If PulseAudio was started in system mode and the <b>system.pa</b> file was modified, the system-wide service must be restarted.</p> <p>Please note: if "<b>autospawn = yes</b>" is set in <b>/etc/pulse/client.conf</b>, the process will be restarted automatically after stopping it.</p>
4	Check if the server is listening on its port.	<pre># netstat -an   grep 4713</pre> <p><b>OR</b></p> <pre># netstat -a   grep -i pulse</pre>

 Make sure access to the PulseAudio server port is not blocked by a firewall or security group. However, access to the port should only be allowed as required in order to minimize security risks.

### b) Solaris

On Solaris, the audio driver is part of the standard Solaris installation kit. No additional driver should be needed.

However, after configuring the audio server (e.g., in Charon-SSP Manager),

- the Charon instance needs to be restarted, and
- the Solaris guest must be booted with the **reconfigure** option (**boot <device> -r**) to create the **/dev/audio** device.

Note: If the connection to the PulseAudio server is interrupted (e.g. configuration changes or network problems), the audio device in the guest stops working. Even if the connection is then restored, the audio device will not start working again until the emulator instance has been restarted.

 If only the Keyboard Buzz feature will be used, the Solaris guest system does not require an active sound card.

#### **Testing the audio functionality:**

After configuring the audio function and rebooting the Solaris guest system, use the command-line utilities **audioplay/audiorecord**, the GUI-based Java **media player**, **sdtaudio**, or **audiotool** depending on the Solaris version used. These tools allow you to record and play back audio.

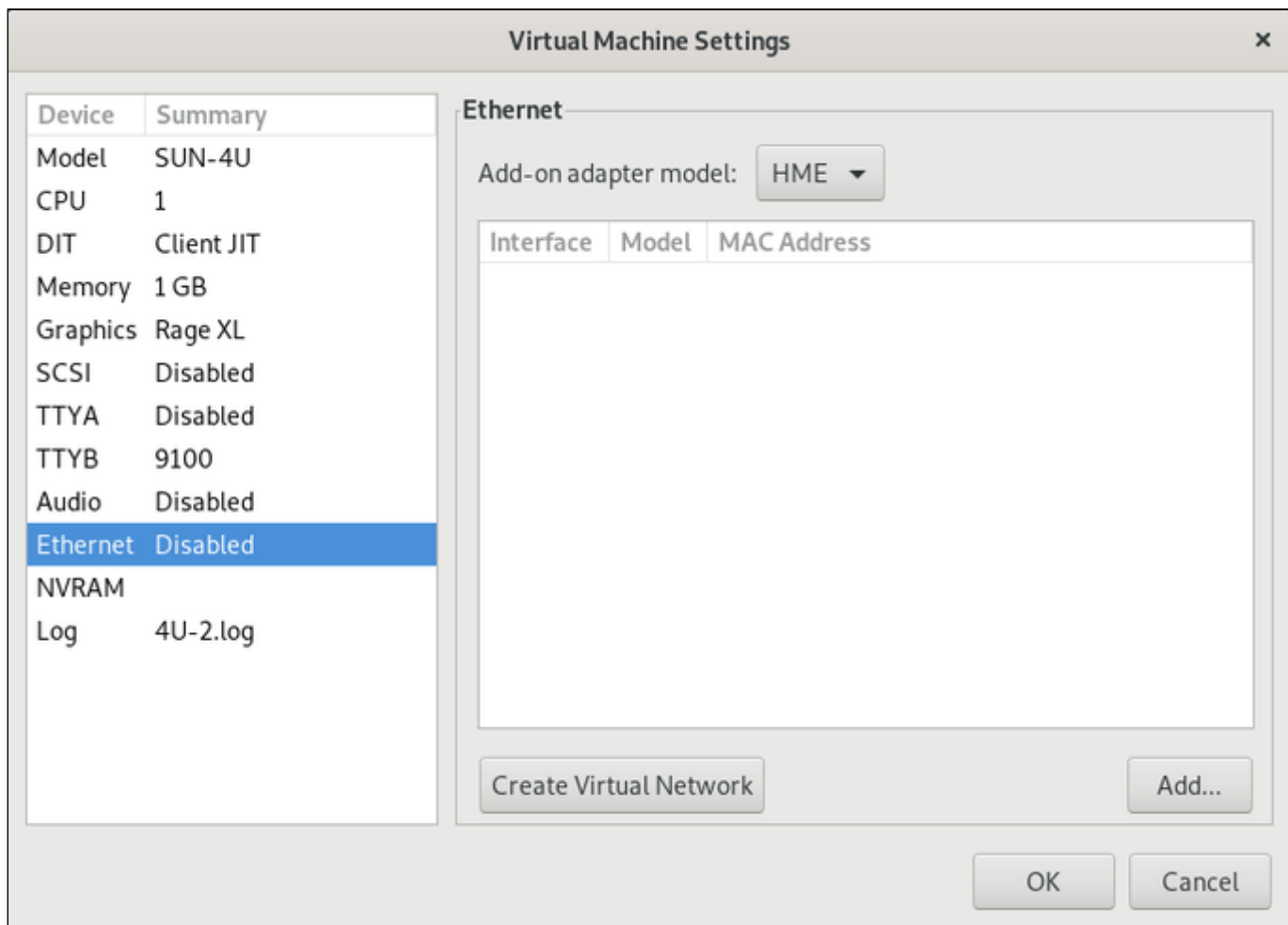
## Ethernet Configuration

### Contents

- Ethernet Configuration Window Overview
- Supported Adapter Models
  - QFE Ethernet Controller Information
- Adding and Editing an Ethernet Adapter

### Ethernet Configuration Window Overview

To view or change the virtual machine Ethernet configuration, select **Ethernet** in the left-hand pane of the Settings window. This will open the Ethernet configuration window. A sample Charon-SSP/4U window is shown below:



#### Charon-SSP Ethernet configuration screen functions:

- To **create** a virtual network, **click** the **Create Virtual Network** button. For further details on creating, changing and removing a virtual network, see the section *Host System Network Configuration*. Only internal bridges (without binding interface) can be used with Charon-SSP AWS.
- To **modify** an existing virtual Ethernet adapter, select it from the list of configured devices and **click** on **Edit** (the edit button becomes visible once an interface has been selected).
- To **remove** an existing virtual Ethernet adapter, select the adapter from the list of configured devices and **click** the **Remove** button (the remove button becomes visible once an interface has been selected).
- To **add** a new virtual Ethernet adapter, **click** the **Add** button.



## Supported Adapter Models

The different Charon-SSP variants support different adapter models: Supported adapter models:

- Charon-SSP/4U: **HME** and **QFE** (4-port Fast Ethernet)
- Charon-SSP/4M: **LE**
- Charon-SSP/4V: **BGE** and **QFE** (4-port Fast Ethernet)

**i** Please note that for **Charon-SSP/4U** the first configured Ethernet interface in the Charon-SSP Manager represents the SPARC on-board device and must be of type HME. Hence, it will always show model HME even if type QFE has been selected.

## QFE Ethernet Controller Information

### Prerequisites for QFE controllers on Solaris:

After newly configuring one or more QFE Ethernet ports, boot the guest system with the reconfigure flag (**boot <disk> -r**). To support the QFE controller, Solaris needs the *Sun Quad FastEthernet Adapter Driver* (SUNWqfed). This package is bundled with the Solaris operating environment starting with Solaris 2.6 Hardware: 5/98. For earlier versions of Solaris, the vendor provided a driver CD with the adapter. After installing the driver, the interfaces should become visible in the **ifconfig** output upon entering the command **ifconfig qfeX plumb**. X denotes the interface number. Use **prtconf** to identify the correct interface numbers. To assign an address to the qfeX interface, create a */etc/hostname.qfeX* file with the hostname for the interface and add the address for the hostname to */etc/hosts*.

Please note that on Solaris 11, different commands are required to configure the interface:

```
# ipadm create-ip netX
# ipadm create-addr -T static -a <ip-address>/<netmask> netX/v4
```

### QFE configuration notes:

To configure a QFE Ethernet ports:

- Select **QFE** in the **Add-on adapter model** drop-down menu.
- For Charon-SSP/4U: configure the on-board Ethernet device. This will be of type HME even if adapter model QFE is selected.
- Configure the desired number of emulated QFE ports (the number does not have to be a multiple of four).

**i** If the guest system does not use the HME controller, you can create a virtual network without an external interface and assign one of the bridge interfaces to the controller as a kind of dummy interface. Alternatively, you could assign the localhost interface (lo) to the unused emulated Ethernet device.

## Adding and Editing an Ethernet Adapter

After selecting to **add** or to **edit** an adapter, a window similar to the one below will open:

The following parameters can be configured:


### **Interface:**

Select the host attached Ethernet device to be connected to the virtual device. This field is a drop-down list of all the network adapters available on the host system. Important points:

- On a Charon-SSP AWS host, you can either use an internal bridge to create TAP interfaces that will then be used in the emulator configuration, or you can use a dedicated NIC. If a dedicated NIC is used,
  - the MAC address of the emulated interface must be set to the MAC address of the NIC connected to the Charon host, and
  - the IP address of the guest system must be set to the private IP address allocated to the NIC by AWS.
- It is permitted to assign the localhost interface (lo) to an emulated device (if only a dummy device is required in the guest).
- Some options are configurable, but will not work with Charon-SSP AWS. They are listed here only for completeness:
  - Using the same physical device for multiple emulated Ethernet devices of the same instance.
  - Sharing a NIC between emulator and host (i.e., both have their own IP address on the same interface).
  - Assigning the same physical interface to more than one Charon-SSP instance.

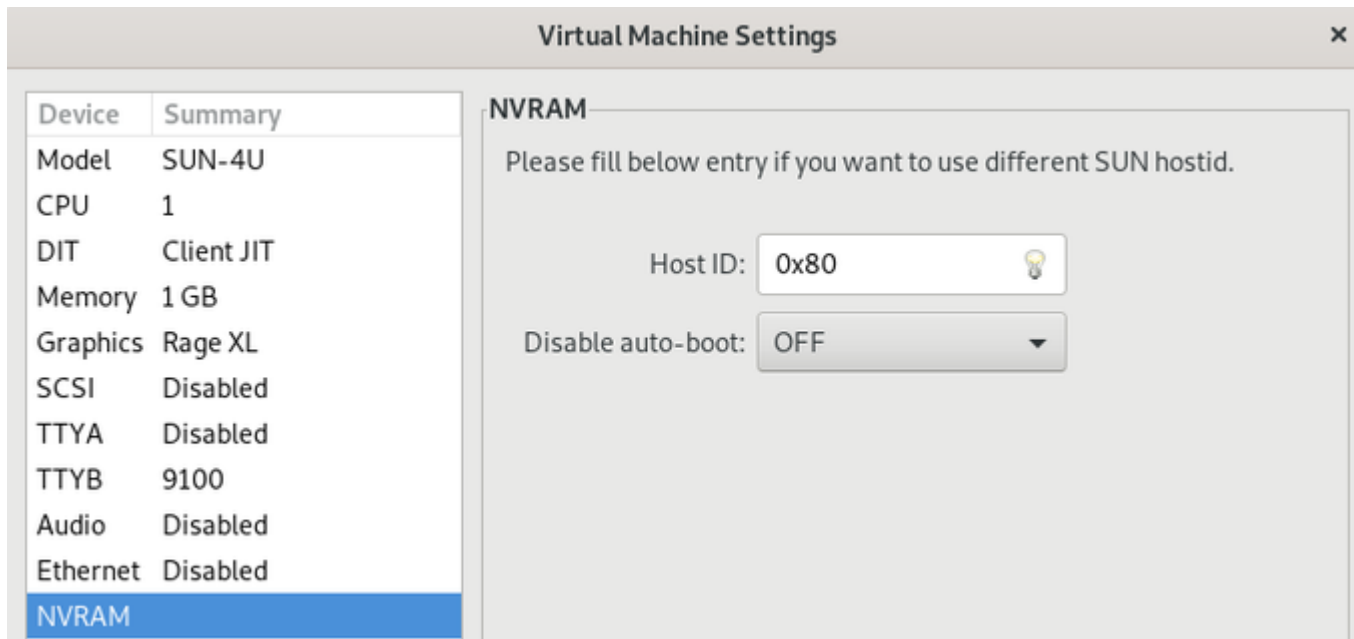
### **Set MAC Address:**

To force the MAC address of the virtual Ethernet device to a specific value, select the checkbox and enter the address in groups of two-character hexadecimal digits, separated by a colon, e.g. 08:00:2b:aa:bb:cc.

 This option can be useful in cases where licensing is tied to a network adapter MAC address. It is required if a dedicated NIC is used on the Charon-SSP AWS instance.

## NVRAM Configuration

To view or change the NVRAM configuration of the emulated system, select **NVRAM** in the left-hand pane of the **Virtual Machine Settings** Window:




The screenshot shows the 'Virtual Machine Settings' window. On the left, a table lists various devices and their summaries. The 'NVRAM' device is selected and highlighted in blue. The main area displays the NVRAM configuration options:

Device	Summary
Model	SUN-4U
CPU	1
DIT	Client JIT
Memory	1 GB
Graphics	Rage XL
SCSI	Disabled
TTYA	Disabled
TTYB	9100
Audio	Disabled
Ethernet	Disabled
<b>NVRAM</b>	

**NVRAM**

Please fill below entry if you want to use different SUN hostid.

Host ID:  

Disable auto-boot:

On this screen, two parameters can be configured:

- **Host ID:** This option can be useful in cases where licensing is tied to the host ID of the physical system.
- **Disable auto-boot:** Default: OFF. The automatic boot of the emulated system can be disabled.

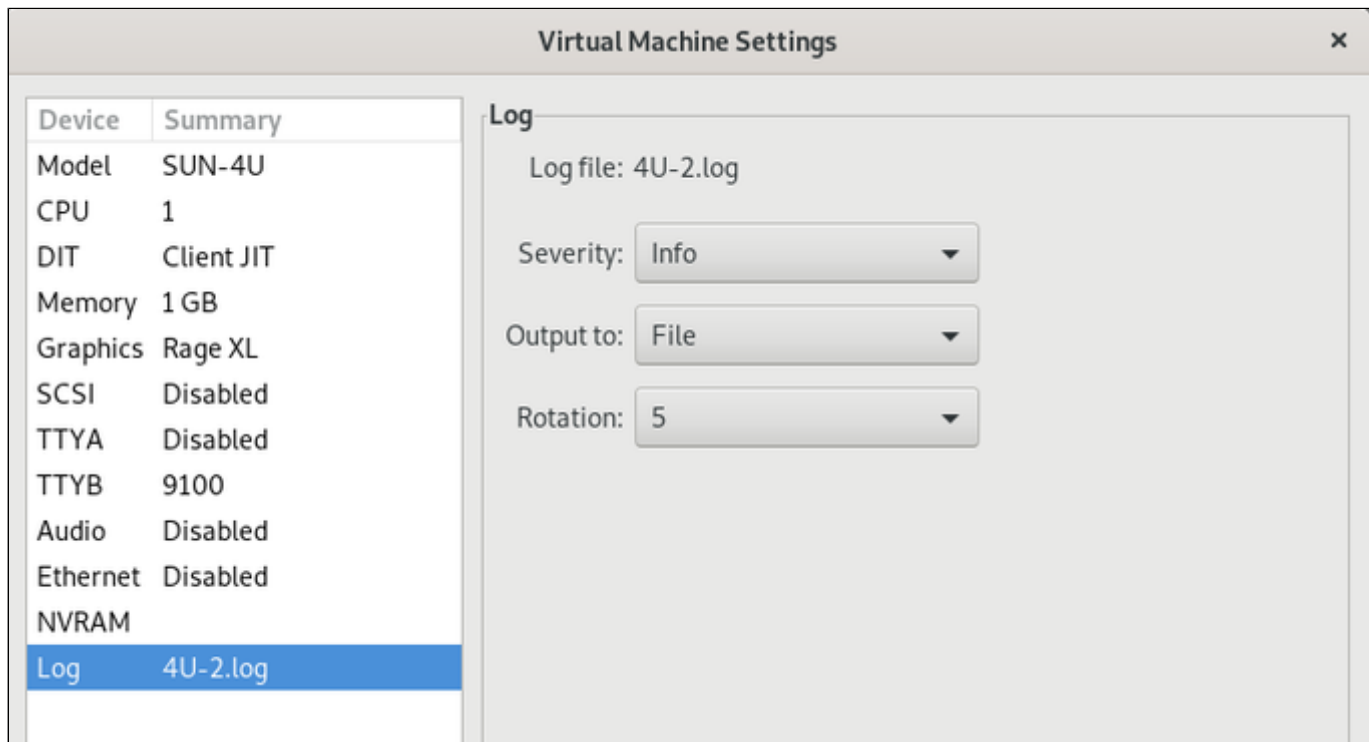
## Log Configuration

### Contents

- Log Configuration Parameters
- Viewing the Charon-SSP Log Files

### Log Configuration Parameters

To view or change the virtual machine logging configuration, select **Log** in the left-hand pane of the Settings window.



The overview below shows each of the fields in the log configuration window and describes their operation.

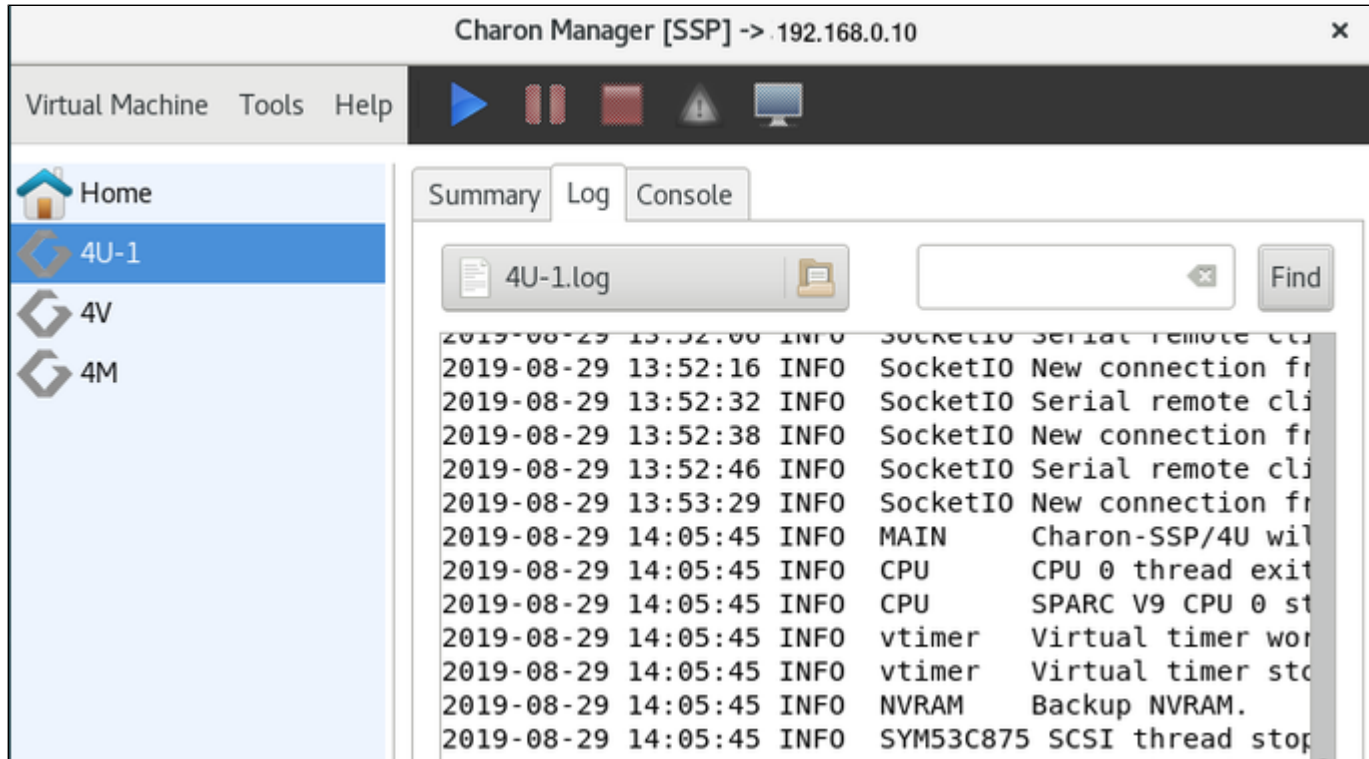
- **Log file:** display only field.
- **Severity:** set the minimum level of messages that should be reported. Legal values are **debug**, **info**, **warning**, **error** and **fatal**. The default is **info**.
- **Output to:** specifies the location to which virtual machine logging information should be written. The default is **file**.
  - **file:** write virtual machine logging information only to the file shown in the **Log file** parameter.
  - **console:** write virtual machine logging information only to the virtual machine console.
  - **all:** write virtual machine logging information to both the file show in **Log file** and the virtual machine console.
- **Rotation:** select the number of old versions of the log files to be saved. The Charon-SSP log files are rotated when the virtual machine starts and, during operation, based on the number of lines written to the log. Once the number of log lines reaches 800.000, the log is rotated.

## Viewing the Charon-SSP Log Files

Currently, Charon-SSP writes three types of instance specific log files:

- **Virtual machine log:** it documents the operation and potential problems of the Charon-SSP instance in question. For example, if no valid license is available, this is logged here.
- **Console log:** if configured, Charon-SSP keeps a console log for Vconsole, TTYA and TTYB.
- **Crash log:** should the Charon-SSP instance terminate unexpectedly, trace-backs and similar information are found in this log file. The contents help Stromasys engineering to identify and repair the problem.

The log files can be viewed using **Log** tab of the Charon-SSP Manager:



To select a log file, click on the file-browser button. This shows all available logs in the default (or configured log path) and lets you select a file. You can also select a different file path to display log files in other locations.

Entering text into the search field and pressing find will filter the log contents according to the search string.

## Virtual Machine Context Menu

### Contents

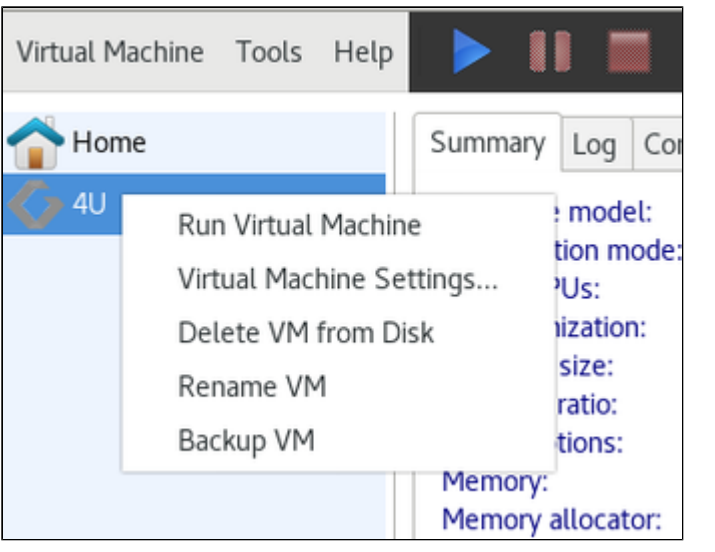
- Overview
- Virtual Machine Context Menu Entries
  - Run Virtual Machine
  - Virtual Machine Settings
  - Delete VM from Disk Menu
  - Rename VM
  - Backup VM

### Overview

Each configured virtual machine in the Charon-SSP Manager has a context menu that is opened by **clicking** on the virtual machine with the **right mouse button**.

The context menu has the following options:

- Run Virtual Machine
- Virtual Machine Settings
- Delete VM from Disk
- Rename VM
- Backup VM



These options are described in the following sections.

**Please note:** right-clicking into the virtual machine list pane when no virtual machine is selected opens an additional small context menu with options to create or import a virtual machine.

## Virtual Machine Context Menu Entries

### Run Virtual Machine

The option **Run the Virtual Machine** starts the virtual machine. The Charon-SSP icon next to the machine name changes from gray to multi-color to indicate a running instance. After starting the virtual machine, all options in the context menu apart from **Virtual Machine Settings** (and, in the case of a Baremetal system, **Backup VM**) are inactive until the virtual machine is stopped again.

The **Run the Virtual Machine** option is equivalent to

- clicking on **Run Virtual Machine** at the right-hand bottom of the virtual machine summary page, or
- clicking on the blue triangle at the top of the Charon-SSP Manager window.

This option is inactive while the virtual machine is running.

## Virtual Machine Settings

The option **Virtual Machine Settings** leads to the configuration options that are described in section [Configuring a Virtual Machine](#).

## Delete VM from Disk Menu

The complete removal of a virtual machine must be performed in several steps:

1. Shut down the guest operating system and stop the virtual machine if it is running. The menu option to delete a virtual machine is inactive while the virtual machine is running.
2. **Right-click** on the name of the virtual machine in the left-hand pane of Charon-SSP Manager.
3. The context menu opens. Select **Delete VM from Disk**. You will be prompted to confirm your choice.
4. Any configurations and log files related to the system are removed and do no longer exist. Associated virtual storage container files are not deleted.

## Rename VM

The option **Rename VM** allows you to rename your virtual machine. When you click on the option, you will be prompted for the new VM name. Enter the new name and confirm your input by clicking on OK.

The virtual machine appears in the Charon-SSP Manager with the new name. This action renames the configuration directory of the virtual machine and the associated configuration file. This option is inactive, while the virtual machine is running.

## Backup VM

Use this function to create a ZIP-file of the configuration file, log files and other VM information. When this option is selected, a window opens where storage location and ZIP-file name can be selected. The resulting backup can be copied to a remote system via SFTP (via the user **charon**).

**This function does not backup the virtual and physical disks used by the Charon-SSP instance.**

## Host System Network Configuration

### Contents

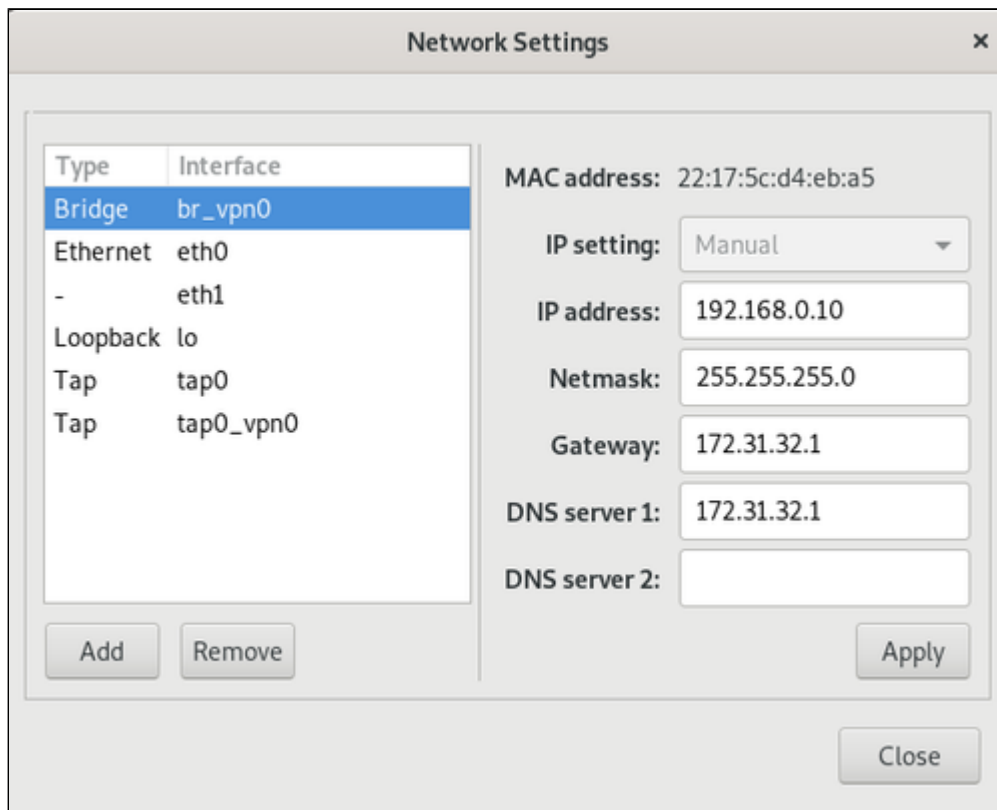
- Overview
- Managing Host System Network Interfaces
- Managing Virtual Networks
  - Creating a Virtual Network
  - Deleting a Virtual Network
  - Resizing a Virtual Network
- Managing VLAN Interfaces
  - Adding a VLAN Interface
  - Deleting a VLAN Interface

### Overview

Charon-SSP Manager provides features to configure the following host system network configuration aspects:

- Configuring host system network interface settings.
- Adding a virtual bridge, i.e., a collection of virtual network tap (TAP) devices that constitute a host-attached virtual LAN. A virtual bridge can be connected to the customer network or be internal to the host system.
- Adding VLAN interfaces to a parent Ethernet interface. This allows the host system to participate in the specified VLAN in the customer network.

To open the network settings window, click on **Tools > Network Settings**. This will open a window similar to the ones shown below:



Content of the network settings window:

- **Left-hand side:** list of available host system network interfaces (including bridge and VLAN interfaces created previously).
- **Right-hand side:** settings of the currently selected interface.
- **Apply** button: confirms any configuration changes made for the selected interface.
- **Add** button: opens a submenu where you can select to add a virtual bridge or a VLAN interface.
- **Remove** button: allows to remove the selected virtual bridge or VLAN interface.

Please refer to the next sections for a detailed description of the network configuration options.

## Managing Host System Network Interfaces

The AWS EC2 environment has specific characteristics that could conflict with interface configurations made via the Charon Manager. Please refer to the Amazon AWS documentation and the sections [Network Management](#) and [AWS Networking and Charon-SSP](#) to understand the networking behavior of an AWS instance before you change any interface settings via the Charon Manager. In particular, if you added a second interface to the system, do not apply any changes via the Network Settings until you created a configuration file for the second interface and are sure both interfaces are working correctly.

Open the network settings window as described above by clicking on **Tools > Network Settings**.

Type	Interface
Ethernet	eth0
Ethernet	eth1
Loopback	lo

MAC address: 0e:f5:f7:60:aa:34

IP setting: Automatic (DHCP)

IP address: 172.31.44.220

Netmask: 255.255.240.0

Gateway: 172.31.32.1

DNS server 1: 172.31.32.1

DNS server 2:

Using the network settings window, you can set up the existing host system network interfaces according to your requirements. The window also contains previously created bridge and VLAN interfaces.

First, **select the interface** that is to be configured.

After selecting an interface, you can then set the following **host system network interface parameters**:

- **IP setting:** specify the method used for the IPv4 addressing of the interface. Options are **Automatic (DHCP)**, **Manual**, and **None**.
- **IP address:** if manual addressing is selected, the host IP address can be added in this field. The field is inactive if DHCP or None is selected.
- **Netmask:** if manual addressing is selected, the netmask for the host IP address can be added in this field. The field is inactive if DHCP or None is selected.
- **Gateway:** if manual addressing is selected, the default gateway for the host can be added in this field. The field is inactive if DHCP or None is selected.
  - ⚠ Be careful not to select a default gateway not matching the AWS subnet structure. Doing so may cause you to permanently lose access to your instance.
  - ⚠ In some cases when several network interfaces are configured on the Charon host, a second routing table has to be created on the Charon host. This is not supported by the Charon Manager and must be configured from the command-line. See [AWS Networking and Charon-SSP](#) for more information.
- **DNS server 1** and **DNS server 2:** if manual addressing is selected, enter the IP address of one or two DNS name servers. Inactive if DHCP or None is selected.

The **Apply** button confirms any changes made and **Close** discards them.



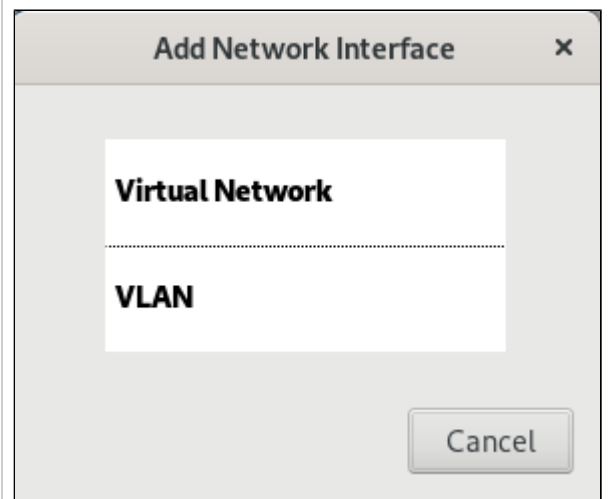
## Managing Virtual Networks

### Creating a Virtual Network

A virtual network can be used to create a virtual bridge on the host system with a number of virtual network interfaces attached to it. The virtual interfaces can be used to provide network interfaces for use by Charon-SSP instances. A virtual network can be connected to the external network using a so-called binding interface, or it can be internal to the host system.

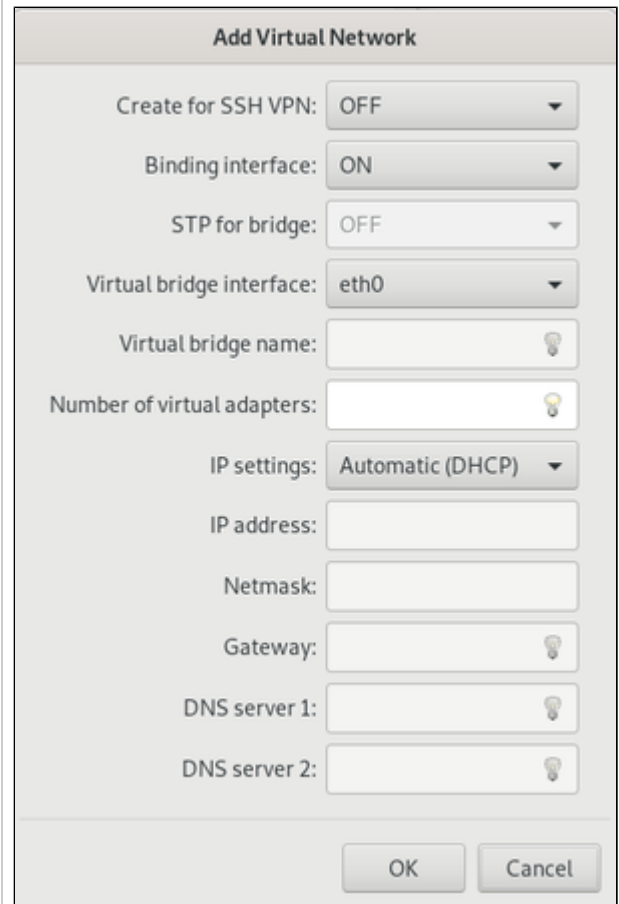
To create a new virtual network, open the network settings window via **Tools > Network Settings**. Then follow the steps shown below:

- Click on the **Add** button to open the submenu for selecting between virtual networks and VLANs.
- Select **Virtual Network**.




This will open the virtual network configuration window as shown here.

Configure the virtual bridge. The configuration settings are described below.



Virtual bridge (i.e., virtual network) configuration options:

Virtual network configuration options	
Field	Description
<b>Create for SSH VPN</b>	If set to <b>ON</b> , a special virtual network will be created to be used as the basis for creating an SSH VPN tunnel as described in <a href="#">SSH VPN - Connecting Charon Host and Guest to Customer Network</a> . This is the most relevant configuration mode for the Charon-SSP AWS product.
<b>Binding interface</b>	<p>If set to <b>ON</b>, a physical interface can be selected from the <b>Virtual bridge interface</b> drop-down menu, on which the bridge is configured. The bridge is connected to the host system LAN. <b>This option is listed for completeness. It is not suitable for Charon-SSP AWS.</b></p> <p>If set to <b>OFF</b>, a user-defined name can be entered in the <b>Virtual bridge name</b> field. This name will be used in naming the bridge and TAP interfaces instead of using the physical interface name. The bridge is internal to the host system.</p> <p>Always <b>OFF</b> if <b>Create for SSH VPN</b> is enabled.</p>
<b>STP for bridge</b>	Enable or disable the Spanning Tree Protocol on the virtual bridge. Always <b>OFF</b> if binding interface is set to <b>ON</b> or SSH VPN is enabled.
<b>Virtual bridge interface</b>	Drop-down menu to select a physical interface that will provide an external network connection to the bridge. Inactive if the binding interface is disabled and if SSH VPN is enabled.
<b>Virtual bridge name</b>	Used to set a user-defined bridge name if the binding interface is disabled. This name will be used in place of the physical interface name when creating the bridge and TAP interfaces. Inactive if the binding interface is enabled. Fixed name <b>vpnX</b> for SSH VPN configuration (X = 0, 1, ...).
<b>Number of virtual adapters</b>	Specify how many virtual adapters are needed.
<b>IP settings</b>	Specify the method used for addressing the interface used to connect the host to the external network. Options are <b>Automatic (DHCP)</b> , <b>Manual</b> , and <b>None</b> . If the binding interface is disabled, manual configuration is mandatory (to assign a configuration to the host-internal bridge interface).
<b>IP address</b>	If manual addressing is selected, the host IP address can be added in this field. The field is inactive if DHCP or None is selected.
<b>Netmask</b>	If manual addressing is selected, the netmask for the host IP address can be added in this field. The field is inactive if DHCP or None is selected.
<b>Gateway</b>	<p>If manual addressing is selected, the default gateway for the host can be added in this field. The field is inactive if SSH VPN configuration, DHCP or None is selected.</p> <p> Be careful not to select a default gateway not matching the AWS subnet structure. Doing so may cause you to permanently lose access to your instance. When you create a custom internal bridge, leave this field empty (the host default gateway will apply).</p>
<b>DNS server 1 and DNS server 2</b>	If manual addressing is selected, you can add the IP address of one or two DNS name servers. Inactive if SSH VPN configuration is selected.

The virtual network connected to a binding interface consists of

- a bridge device called `br_<physical interface>`, and
- a series of TAP devices named `tapX_<physical interface>`.

If the **binding interface is disabled**, the virtual network consists of

- a bridge called `br_<bridgename>`, and
- a series of `tapX_<bridgename>` TAP devices.

If SSH VPN is enabled, the first virtual network created consists of

- a bridge called `br_vpn0`,
- a `tap0` interface, and
- a series of `tapX_vpn0` interfaces

X is a number from 0 up to the number of virtual adapters (0 to configured number minus 1) specified in **Number of the virtual adapters**. These devices can then be configured for use as virtual Ethernet controllers.

## Deleting a Virtual Network

To delete a virtual network, follow the instructions listed below.


1. Follow the menu path **Tools > Network Settings** to open the network settings window.
2. Select the bridge you want to delete and click on the **Remove** button. This will open a confirmation window.
3. To delete **all** virtual network interfaces associated with the selected bridge, **click** on **YES**.

Following the instructions above will immediately delete all TAP devices and the bridge.

## Resizing a Virtual Network

To resize a virtual network, follow the instructions listed below:

1. Shut down any running guest operating systems and stop all virtual machines connected to the virtual network TAP devices.
2. Delete the current virtual network, using the instructions detailed in *Deleting a Virtual Network*.
3. Re-create the virtual network using the instructions detailed in *Creating a Virtual Network*. Make sure to specify the new virtual network size in the **Number of the virtual adapters** field.
4. Reconfigure the Ethernet configuration of the virtual machines. This step is only necessary if shrinking the virtual network and only if the virtual machines are configured for TAP devices that no longer exist.
5. Start the attached virtual machines.

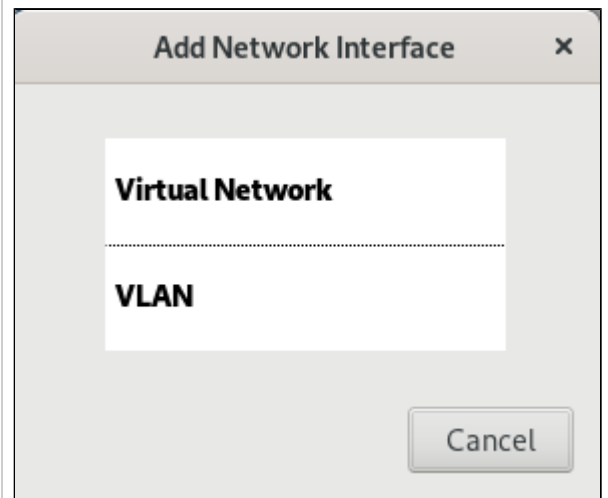
 Shrinking a virtual network may make it necessary to adjust a number of virtual machine configurations because the name of their virtual Ethernet interface has changed.

## Managing VLAN Interfaces

**i** This option is described for completeness. However, it is normally not relevant for the Charon-SSP AWS product.

### Adding a VLAN Interface

- Click on the **Add** button to open the submenu for selecting between virtual networks and VLANs.
- Select **VLAN**.



This will open the VLAN configuration window as shown here.

Configure the VLAN interface. The configuration settings are described below.

VLAN configuration options:

VLAN configuration options	
Field	Description
<b>Parent interface</b>	Select the host system Ethernet interface that will serve as the base interface for the LAN connection.
<b>VLAN ID</b>	Enter the VLAN number matching the customer's LAN configuration. Values: 2-4094. The interface name of the new interface has the format: <i>&lt;parent-interface&gt;.&lt;vlan-id&gt;</i>
<b>IP settings</b>	Specify the method used for addressing the interface used to connect the host to the external network. Options are <b>Automatic (DHCP)</b> , <b>Manual</b> , and <b>None</b> .
<b>IP address</b>	If manual addressing is selected, the host IP address can be added in this field. The field is inactive if DHCP or None is selected.
<b>Netmask</b>	If manual addressing is selected, the netmask for the host IP address can be added in this field. The field is inactive if DHCP or None is selected.
<b>Gateway</b>	If manual addressing is selected, the default gateway for the host can be added in this field. The field is inactive if DHCP or None is selected.
<b>DNS server 1 and DNS server 2</b>	If manual addressing is selected, you can add the IP address of one or two DNS name servers.

## Deleting a VLAN Interface

To delete a VLAN interface, follow the instructions listed below:

1. Follow the menu path **Tools > Network Settings** to open the network settings window.
2. Select the VLAN interface you want to delete and click on the **Remove** button. This will open a confirmation window.
3. To delete the VLAN interface, click on **YES**.

Following the instructions above will immediately delete the VLAN interface.

## Miscellaneous Management Tasks

### Contents

- Contents
- Gathering Host Information
- Adding an Existing Virtual Machine to Charon Manager
- Determining the Version of the Charon-SSP Manager
- Modifying the Charon-SSP Agent Preferences
- Setting Console Options

### Gathering Host Information

To view the details of the system hosting the Charon-SSP instance, follow the menu path **Tools > Host Information** to open a window similar to the one below.

This window provides details of the host system's hardware configuration and operating system version.

The screenshot shows a window titled "Host Information" with a close button (X) in the top right corner. The window contains the following information:

- System edition:** CentOS Linux release 7.3.1611 (Core)
- Processor:** Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz  
2 processors
- Memory (RAM):** 3618116 kB
- Ethernet:** Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)  
br\_vpn0 eth0 eth1 lo tap0 tap0\_vpn0
- Physical Memory (KB):**

Total	3618116 kB
Cached	632876 kB
Free	2611556 kB
- System:**

CPU Usage (%)	4.5
Physical Memory	3618116 kB
Processes	132
Up Time	15 day, 6 hour, 27 minute

A "Close" button is located at the bottom right of the window.

## Adding an Existing Virtual Machine to Charon Manager

To add an existing virtual machine to the Charon-SSP Manager, you have to use the **Import** function. This function is available in the **Virtual Machine** menu (when Home is selected), on the **Home** page of the Charon-SSP Manager, and in the context menu of the virtual machine pane when no Charon-SSP instance is selected.

The **Import** function lets you select an existing Charon-SSP virtual machine configuration and a name for the newly added system.

**!** The imported configuration may have to be adapted to the possibly different environment on the new host system. For example, the path to the virtual storage container files or the names of network devices may be different when compared to the previous environment.

## Determining the Version of the Charon-SSP Manager

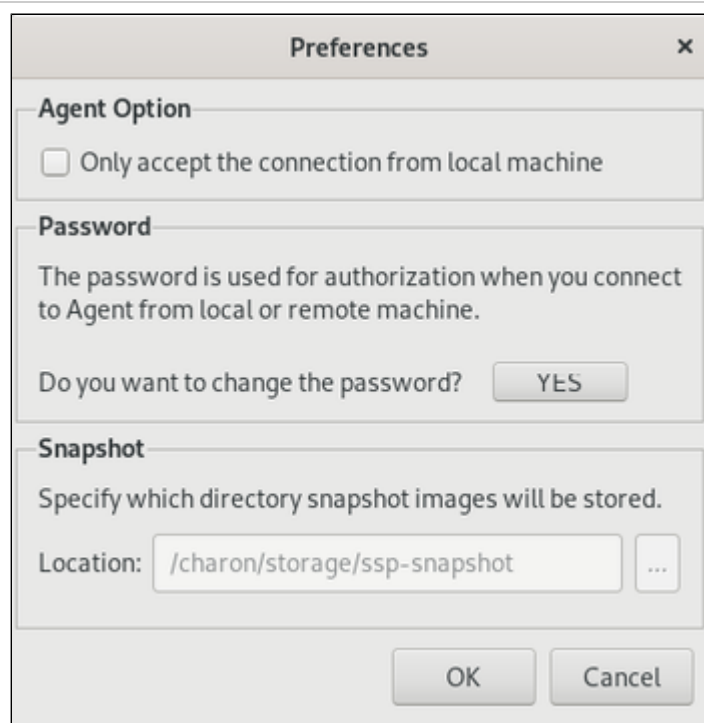
To display the version of Charon-SSP Manager currently running, select **Help > About** from the menu bar. This will open a window displaying the version of the software.

## Modifying the Charon-SSP Agent Preferences

To modify the preferences maintained by the Charon-SSP Agent software, follow the menu path **Virtual Machine > Preferences** to open a window similar to the one shown below.

The preferences window offers the following configuration options:

- To limit the access to the Charon-SSP agent to the local system, check the box under **Agent Option**.  
**!** This option **should not be used on Charon-SSP AWS** without a really good reason as it will cut off Charon Manager access to the cloud instance!
- The password to be used by the Charon-SSP Manager to connect to the current Charon-SSP Agent can be modified by **clicking** on the **YES** button next to **Do you want to change the password?** This will open a change-password dialog.
- The Snapshot parameter shows where currently the resulting files are stored if a Charon-SSP virtual machine is suspended. On Charon-SSP AWS, the location cannot be changed.

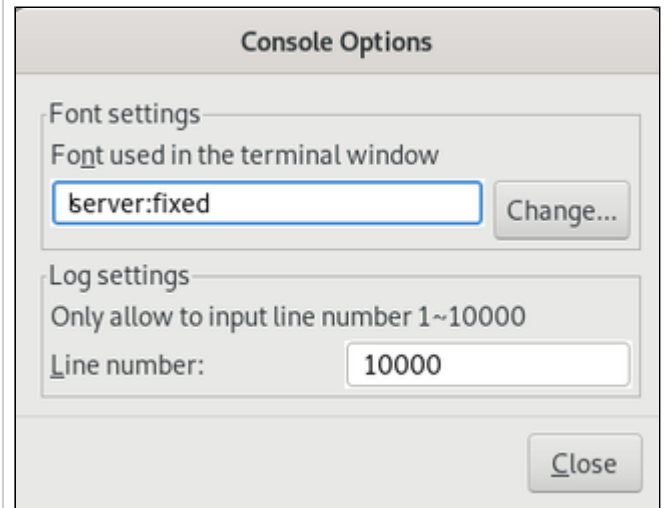


## Setting Console Options

The way the Charon-SSP Manager displays the built-in console can be influenced by using the console options configuration. To open the configuration window, select **Virtual Machine > Console Options**. This displays a window similar to the one shown below.

The configuration window contains two configuration options for the built-in console:

- **Font settings** allow selecting a different font to use for displaying the console output. Click on the **Change** button to select the desired font from a menu.
- **Log settings** allow selecting the number of lines cached for the console display area in the Charon-SSP Manager. When the virtual machine is stopped, the console display tab shows the cached lines of console output for this machine. Please note, this log setting is not related to defining a log file in the TTYA and TTYB configuration.



The screenshot shows a window titled "Console Options" with a light gray background. It is divided into two sections: "Font settings" and "Log settings".

**Font settings:** This section contains the text "Font used in the terminal window" followed by a text input field containing "server:fixed" and a "Change..." button to its right.

**Log settings:** This section contains the text "Only allow to input line number 1~10000" followed by a "Line number:" label and a text input field containing "10000".

At the bottom right of the window is a "Close" button.



## AWS Cloud Tools

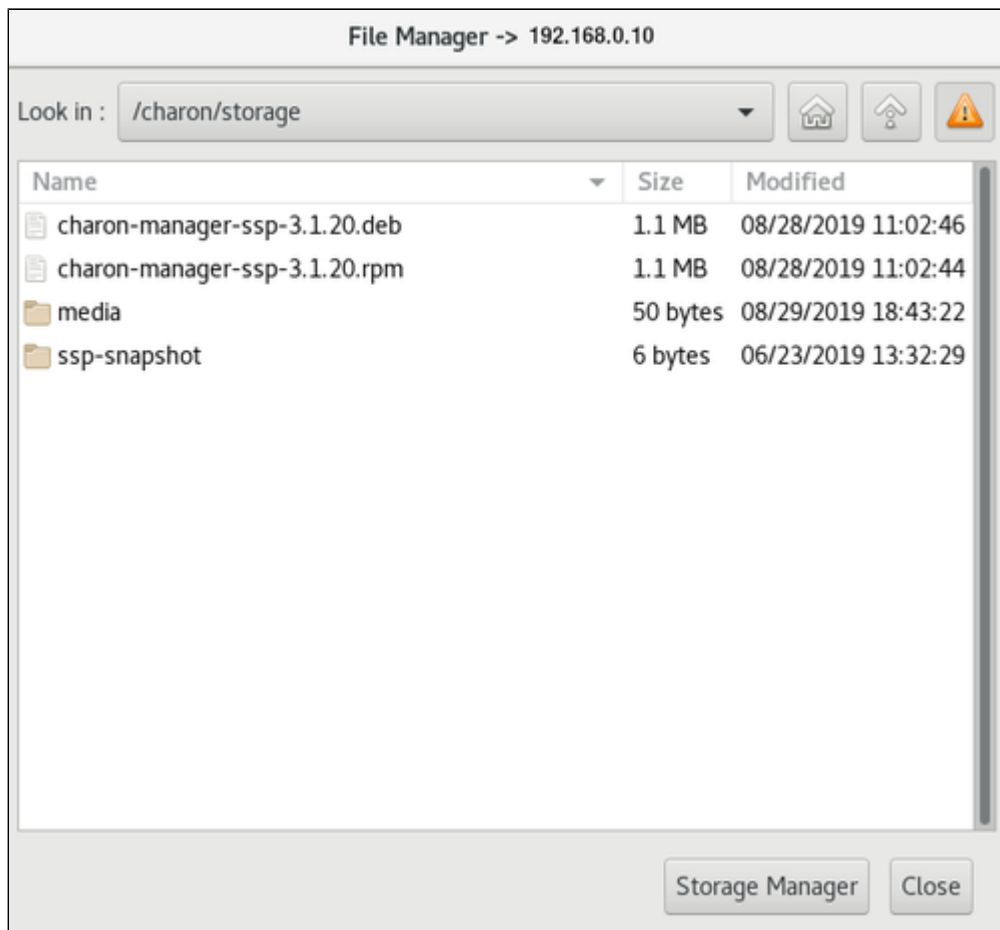
When **Charon-SSP Manager is connected to an Charon-SSP AWS instance**, the Charon-SSP Manager **Tools** menu shows additional tools in the **Tools > AWS Cloud** menu. These tools are described below.

### Contents

- File Manager
- Storage Manager
- Setting Time and Date
- SFTP Server

### File Manager

As the operating system tools of a Charon-SSP AWS instance are not fully accessible to the user, the file manager allows the user to manage files and directories in the data area of the Charon-SSP host system. The image below shows an example of a file manager window:



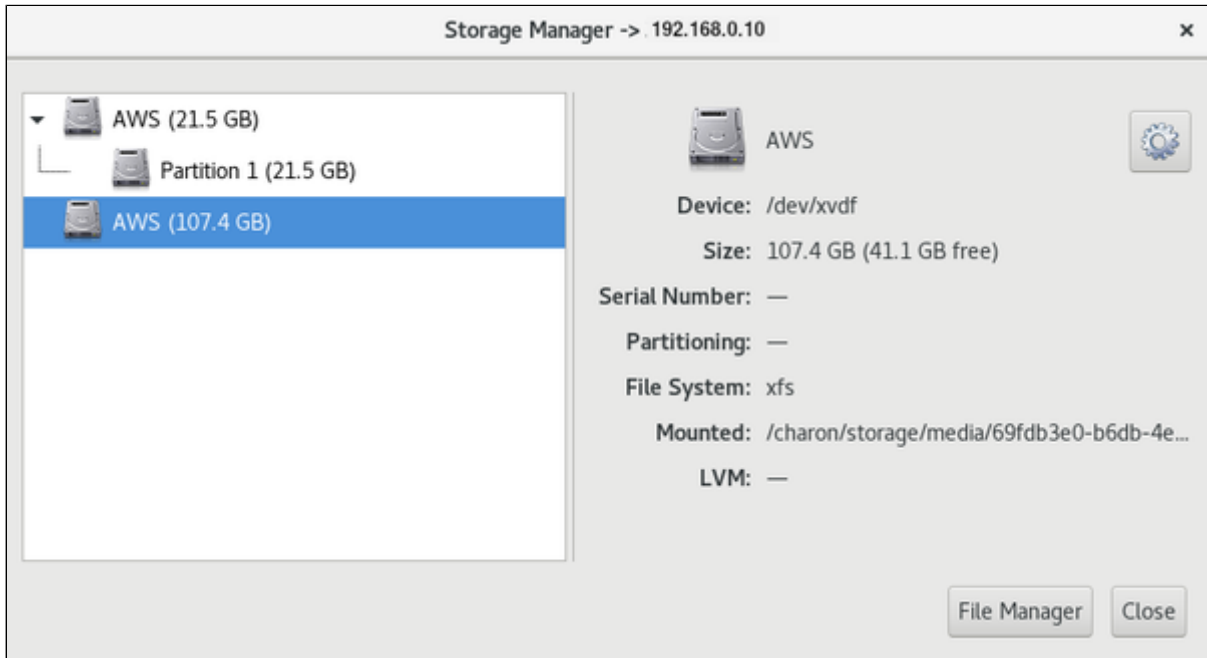
A right-click in the window opens a context menu that provides access to basic file management tasks:

- Create a new folder
- Cut, copy, and paste files and folders
- Delete files and folders
- Rename files and folders

The buttons at the bottom of the window allow closing the file manager or opening the storage manager. The triangle at the top right shows relevant alerts, if any such alerts exist.

## Storage Manager

As the operating system tools of a Charon-SSP AWS instance are not fully accessible to the user, the storage manager allows the user to manage storage devices connected to the Charon-SSP host system. The image below shows an example of a storage manager window:



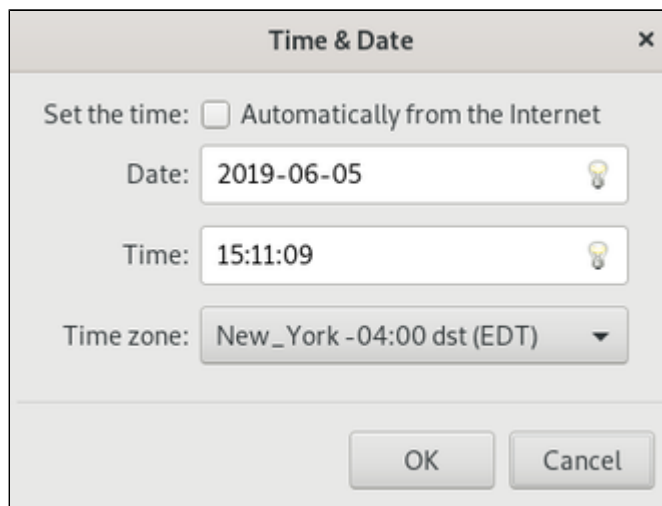
A right-click on a device (or clicking on the cog-wheel) will open a context menu enabling the following tasks:

- Mounting the selected volume
- Unmounting the selected volume
- Formatting the selected volume

Using the buttons at the bottom of the window, the storage manager can be closed, or the file manager can be opened.

## Setting Time and Date

The **Time & Date** option allows setting the time and date of the Charon-SSP AWS instance via Charon-SSP Manager. The following image shows the available options:

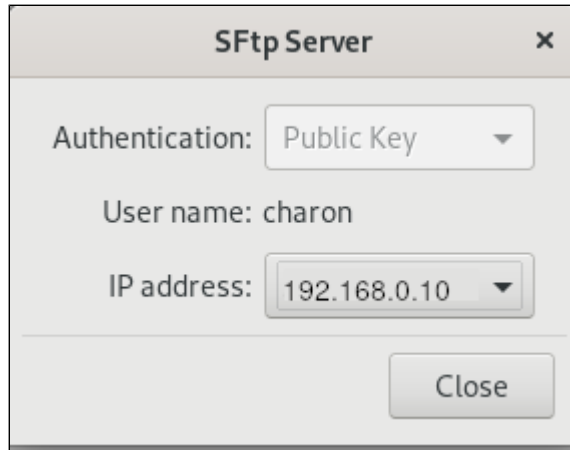


In this window you can

- enable NTP or manually set time and date, and
- configure the time zone.

## SFTP Server

This option allows to configure the login method and the host system address used for SFTP.



The image shows a configuration window titled "SFTP Server" with a close button (X) in the top right corner. The window contains three configuration fields: "Authentication" with a dropdown menu set to "Public Key", "User name" with the text "charon", and "IP address" with a dropdown menu set to "192.168.0.10". A "Close" button is located at the bottom right of the window.

In the configuration window, the following options can be selected (if supported by the host system):


- Authentication: Public Key or Password authentication (for Charon-SSP AWS, Public Key is the only option).
- IP Address: if the host system has several usable IP addresses, select the correct address to connect to via SFTP.

## Graphical Interface via X11 Server on Linux

### Contents

- Overview
- Prerequisites
  - Installing Xephyr on the Remote Linux Host
  - Firewall Considerations
- Enabling XDMCP
  - Enabling XDMCP on Solaris 2.5 to Solaris 9
  - Enabling XDMCP on Solaris 10
- Configuring and Starting the X11 Server in Charon-SSP Manager
  - Basic Configuration Steps and Start
  - X11 Server Configuration Parameters
  - Stopping the X11 Server
- Running the X11 Server on Other Operating Systems.

### Overview

 The X11 feature is supported across a VPN, but not across a NAT connection.

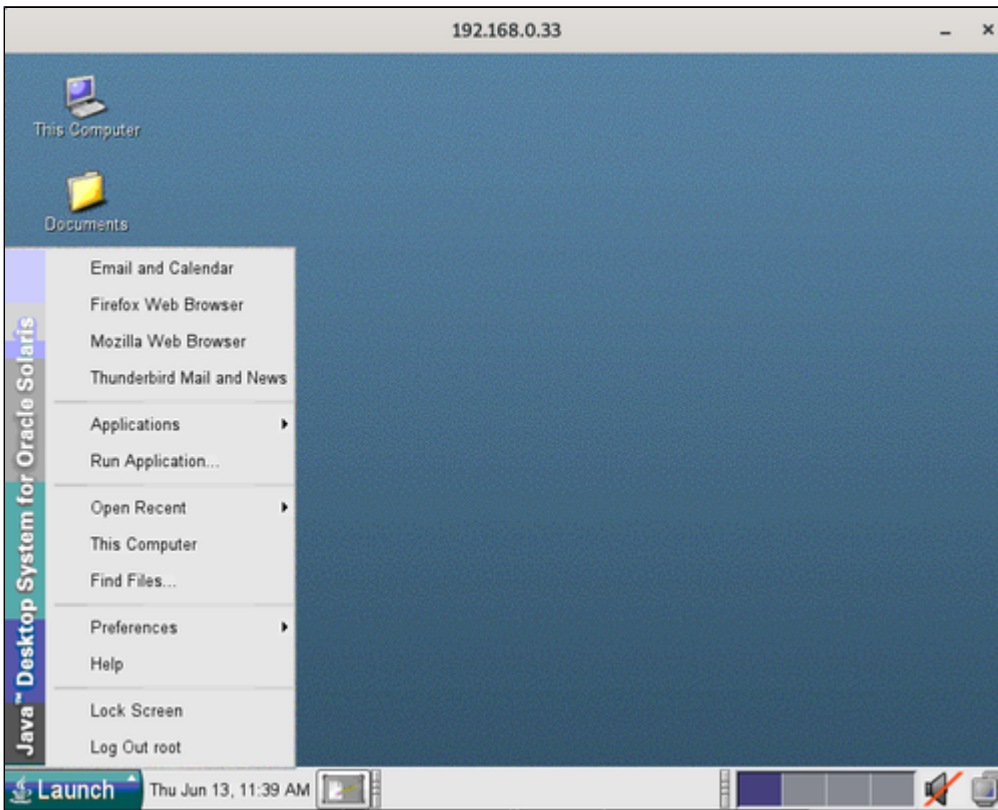
The Charon-SSP Manager can set up an X11 login session using Xephyr and the XDMCP protocol.

Xephyr is a nested X-server that can run within a normal Linux or Baremetal GUI-based user session. It supports the Solaris GUI (Java Desktop, Openwin, CDE, and Gnome) and can provide graphics 3D acceleration based on the OpenGL 1.4 specification.

Running an X-server to access the graphical Solaris interface, requires a network configuration that allows a TCP/IP connection between the system running the X-server and the Solaris Guest operating system (Stromasys recommends that both systems be in the same subnet).

If the X-server runs on a remote system, the remote system must have a working TCP/IP connection to the guest system running in the Charon-SSP instance.

The screenshot below shows an X-session from Charon-SSP Manager to a guest running Oracle Solaris 10.



**Please note:**

The graphical performance depends on many parameters, for example, the performance of the host system, the emulated system, and the network. One important requirement is that the round-trip time of the network connection between display device and emulated Solaris system running on the AWS instance should not be more than 20ms.

For every use case, a test is required to evaluate the suitability for the specific customer environment.

## Prerequisites

### Installing Xephyr on the Remote Linux Host

If it has not happened yet, Xephyr must be installed on the remote Linux system where the Charon-SSP Manager will be used to start the X-server.

Use the following command to install the software on a Linux system with RPM based packet management:

```
# yum install xorg-x11-server-Xephyr
```

Use the following commands to install the software on a Linux system with Debian package management:

```
# apt-get update
# apt-get install xserver-xephyr
```

## Firewall Considerations

The Xephyr nested X-server listens for connections on port range 6001-6100 depending on the X11 Server configuration in Charon-SSP Manager. The configured ports must be allowed if a firewall (e.g. iptables on Linux) is used. For a quick assessment, in case the X-server does not show the dtlogin screen, the following commands on the Linux system running Xephyr can be used to turn off the firewall **temporarily** (depending on what firewall is being used).

```
# systemctl stop firewalld or # service stop iptables
```

On the Charon-SSP AWS instance, verify that the security group assigned to the instance permits the required traffic. If the connection to the X-server runs across an SSH VPN tunnel, only SSH must be allowed to the AWS instance.

Ask your network system administrator to configure proper access to the required port range.

## Enabling XDMCP

Before using the X-server, XDMCP must be enabled on the guest system. The actions for enabling XDMCP are different depending on the version of Solaris installed on the guest. Follow the relevant sub-section below to configure XDMCP on your guest.

### Enabling XDMCP on Solaris 2.5 to Solaris 9

Use the following instructions to enable remote login over XDMCP up to Solaris 9:

1. Edit the file `/usr/dt/config/Xconfig`.

```
# vi /usr/dt/config/Xconfig
```

2. Locate the line `Dtlogin.requestPort: 0` and insert a comment character, '#', at the beginning of the line.

3. Save the configuration file and restart the X-server (if there is no `dtlogin` file in `/etc/init.d`, you have to run `/usr/dt/bin/dtconfig -e` first):

```
# /etc/init.d/dtlogin restart
```

### Enabling XDMCP on Solaris 10

Use the following commands to enable remote login over XDMCP on Solaris 10:

1. Allow access to XDMCP over the network:

```
# svccfg -s cde-login setprop 'dtlogin/args=""'
```

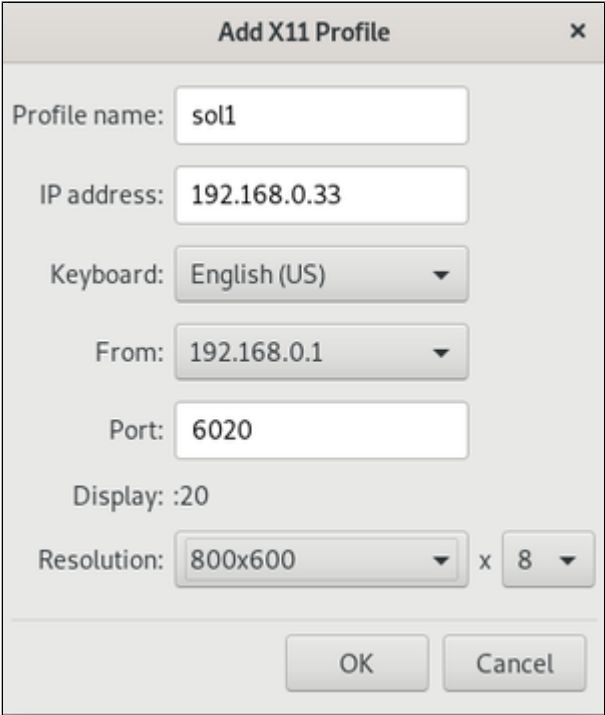
2. Restart CDE.

```
# svcadm restart cde-login
```

## Configuring and Starting the X11 Server in Charon-SSP Manager

### Basic Configuration Steps and Start

Once XDMCP has been enabled on the guest, use the following basic instructions to start the X-server display. The parameters are described in detail in the next section. You can add multiple profiles with different sets of parameters to the configuration of the Charon-SSP Manager.

Basic steps for configuring and starting the X11 server	
Step	Description
1	<p>Open the <b>X11 server Configuration</b> window from Charon-SSP Manager (menu path <b>Tools &gt; X11 Server</b>).</p> <p>Here you can <b>start/stop</b> already configured X11 servers and <b>add, modify</b> or <b>delete</b> them.</p> <p>To add a new server, click on <b>Add</b>. This opens the <b>Add X11 Profile</b> window as shown here:</p> 
2	<p>Configure the X11 server by completing the fields:</p> <ul style="list-style-type: none"> <li>• Enter a profile name</li> <li>• Enter the address or name of the guest in the field <b>IP address</b>.</li> <li>• Choose the keyboard layout preferred for this X-session.</li> <li>• Select the host IP address from which the X-server connects to guest Solaris.</li> <li>• Select the port to be used for the communication.</li> <li>• Select the X-session screen resolutions or <i>Full Screen</i> from the <b>Resolution</b> drop down box.</li> <li>• Click <b>OK</b> to save the configuration.</li> </ul>
3	<p><b>Click on Start</b> to start the selected X-server.</p>

An existing X-server definition can be modified by selecting it and then using the **Edit** button in the X11 overview window.

## X11 Server Configuration Parameters

The parameters of the X11 server configuration are explained in the following table:

X11 server configuration parameters	
Parameter	Description
<b>Profile name</b>	Name to identify a specific set of configuration parameters in the list of saved configurations.
<b>IP address</b>	IP address of the guest Solaris system. If you are using an SSH VPN tunnel to the AWS instance, enter the address the Solaris system has in the VPN network.  The X protocol is not encrypted. So it should not be run over a public network without proper protection by a VPN.
<b>Keyboard</b>	Select the required keyboard from the drop-down list. You can select from the layouts provided by the system on which the Charon-SSP manager runs.
<b>From</b>	If the system running the X-server has only one IP address, this parameter can be left at <b>default</b> . If there is more than one IP address configured on the X-Server host, select the address that is on the same subnet as the Solaris guest or at least reachable from Solaris. This parameter prevents older Solaris versions from choosing a random (potentially unreachable) address from multiple IP addresses available on the host running the X-Server.
<b>Port</b>	Values 6001 - 6100. The port number determines on which display the X-server is started. For example, port 6001 results in the X-server running on display ":1".
<b>Display</b>	Read-only field. Shows the display number based on the port number selected.
<b>Resolution</b>	This parameter can be adapted to specific requirements of applications with respect to the X-server capabilities ("VISUALS"). One example would be the 256-bit indexed color visual, which requires a display depth of 8 bits. It also allows users to set the X display to full screen mode.

## Stopping the X11 Server

To stop the X-server, follow the instructions below:

1. Open the **X11 Server Configuration** window from Charon-SSP Manager by following the menu path **Tools > X11 Server**. A window opens showing all configured X11 profiles.
2. Select the X-server you want to stop.
3. **Click the Stop** button to terminate the X-session.
4. If multiple sessions to the same host are open, it will be necessary to repeat these steps for each session.

An existing X-server definition can be modified by selecting it and then using the **Edit** button in the X11 overview window.

## Running the X11 Server on Other Operating Systems.

The mechanism described above is only valid for Linux operating systems on which the Charon-SSP Manager runs. On other systems, for example Microsoft Windows system, you can use alternative X-server applications. However, the steps are different from the ones used via the Charon-SSP Manager on Linux. As the first step, you must install an X-server. There are several commercial products. However, there are also free X-server packages, for example the X-server integrated in Cygwin, VcXsrv, or Xming. For example, the installer for Xming and more product information are available on <http://www.straightrunning.com/XmingNotes/>. Please also refer to the non-Cloud product documentation for additional information.



## Starting, Stopping, and Suspending the Emulated System

### Contents

- Starting the Emulated System
  - Interactive Start
  - Start with Host System Startup
- Stopping the Emulated System
- Suspending the Emulated System

### Starting the Emulated System

Once the emulated SPARC system has been configured, you can start the emulated system.

An emulated system can be started

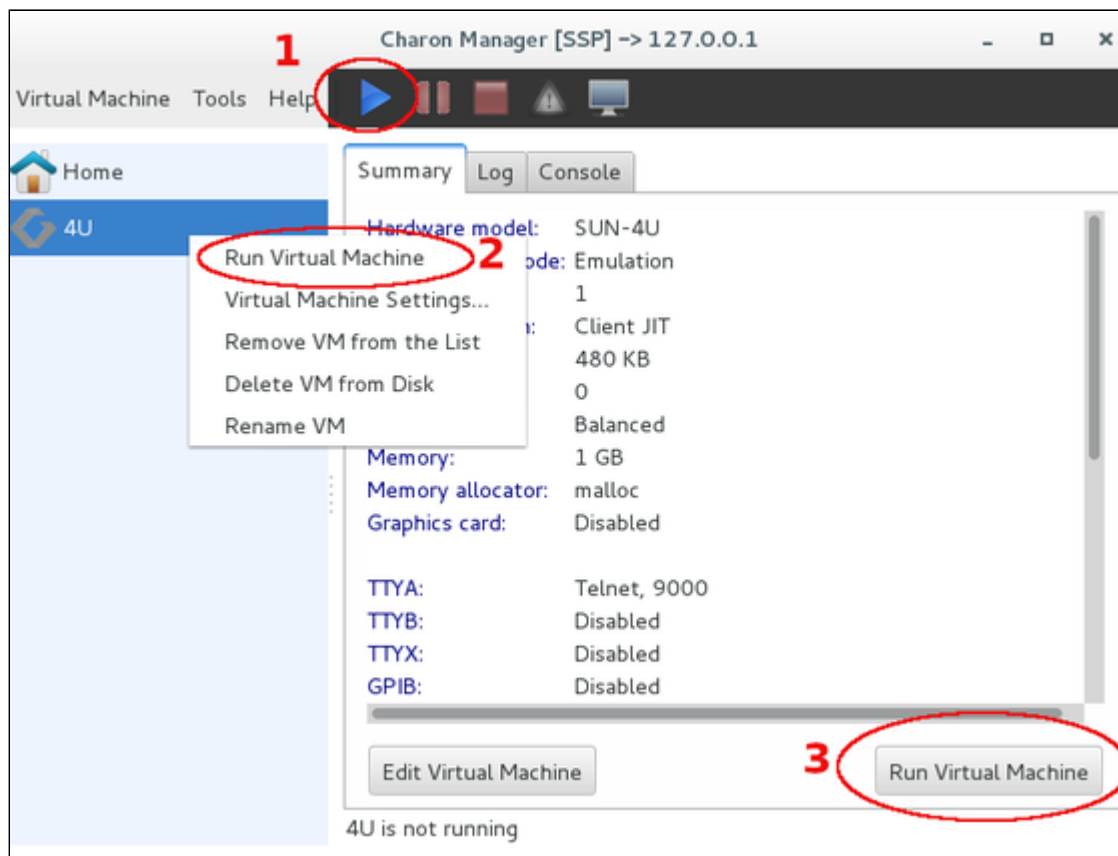
- interactively via the Charon Manager, or
- as a service during host system startup.

#### Interactive Start

An emulated system can be started interactively from the Charon Manager. There are **three different options** inside the Charon Manager to start an emulated system:

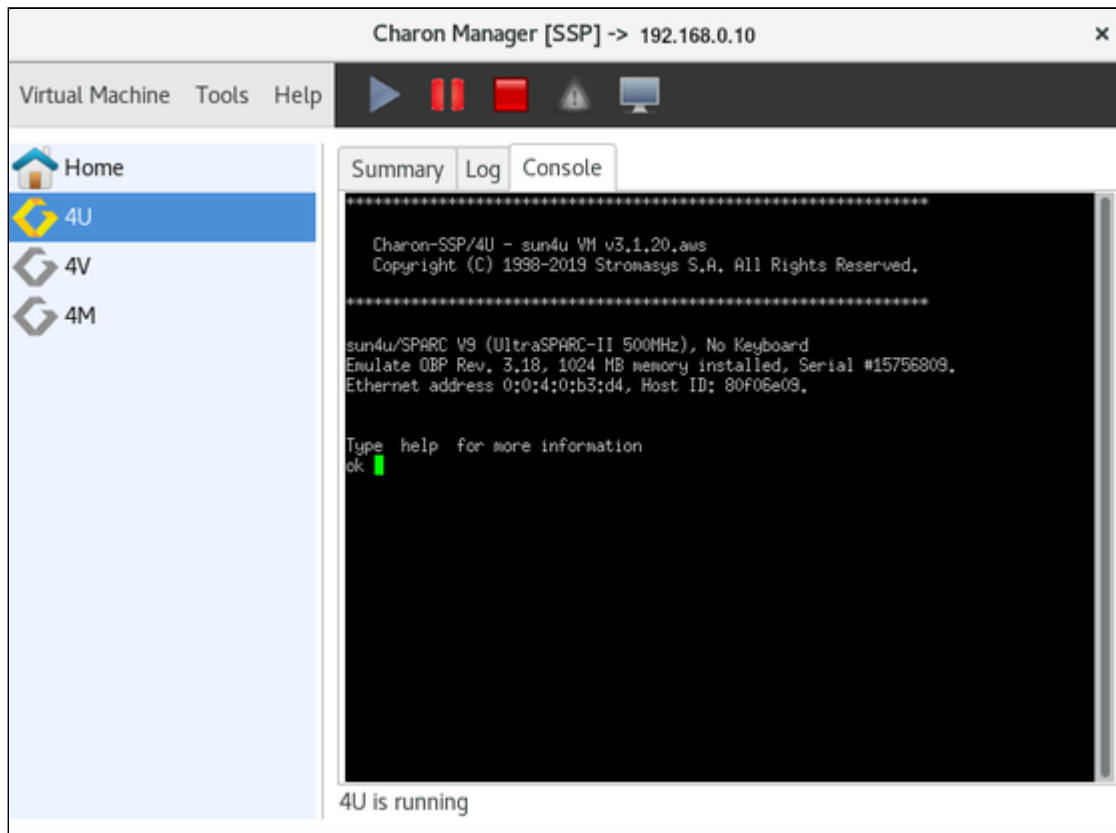
1. Click on the **blue triangle** at the top of the Charon Manager window, or
2. right-click on the virtual machine and select **Run Virtual Machine** from the context menu, or
3. select the virtual machine. Then select the Summary tab and click on the **Run Virtual Machine** button at the bottom of the summary page.

The image below shows the three options:

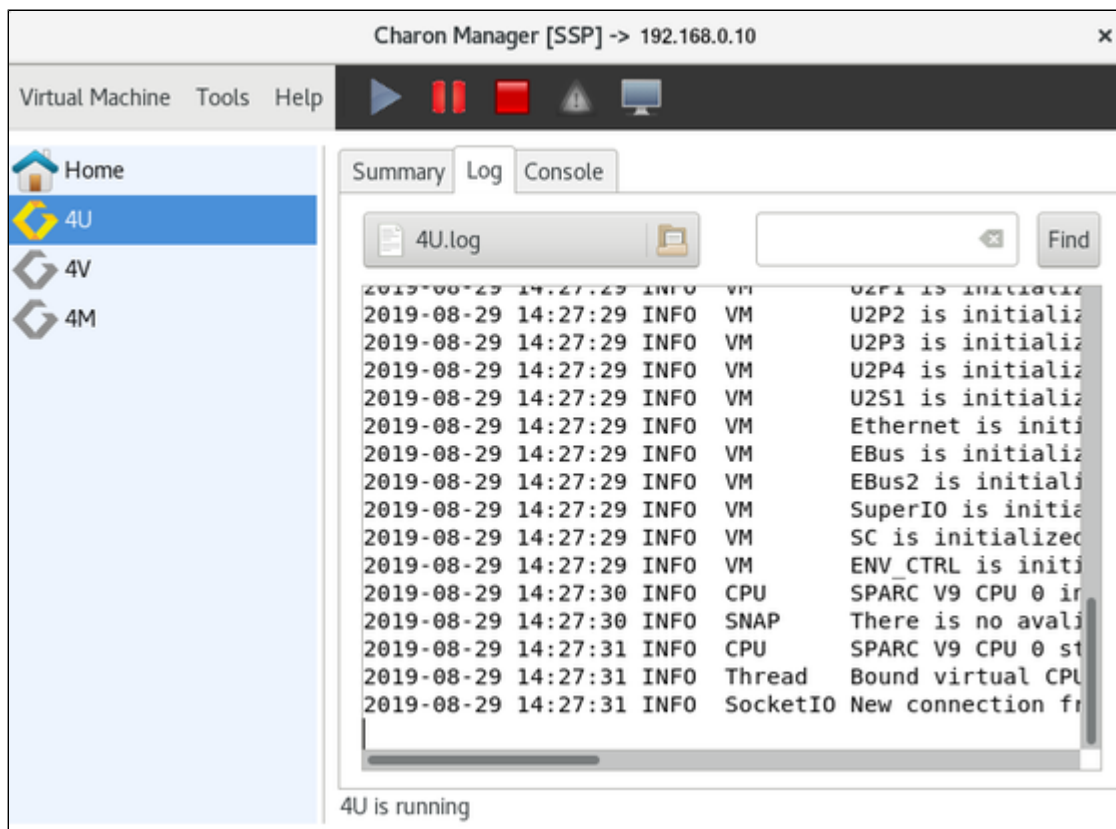


After the system has been started, the built-in console and the emulator log are displayed in Charon Manager.

The image below shows the console prompt of an emulated SPARC system:

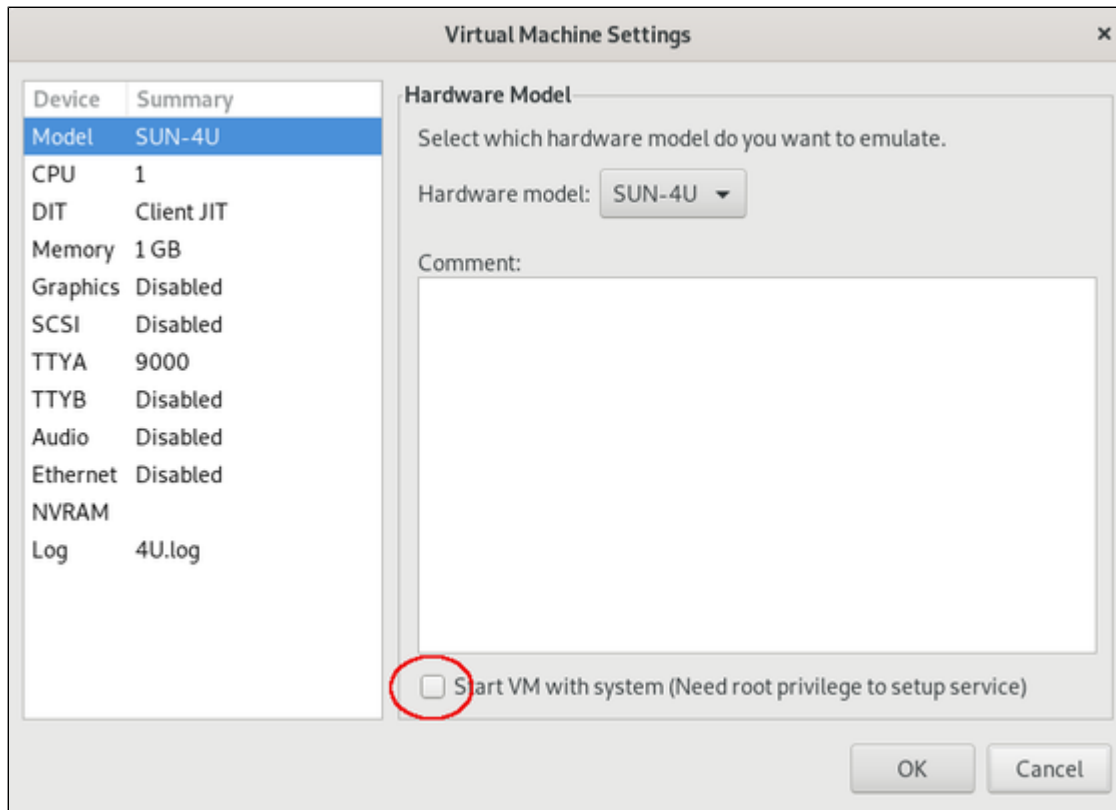


The **Log** tab allows the user to view the different log files produced by the emulator. The example below shows a view of the emulator log file:



## Start with Host System Startup

The model configuration screen in Charon Manager allows to automatically start the emulated system when the host system starts, as shown below.

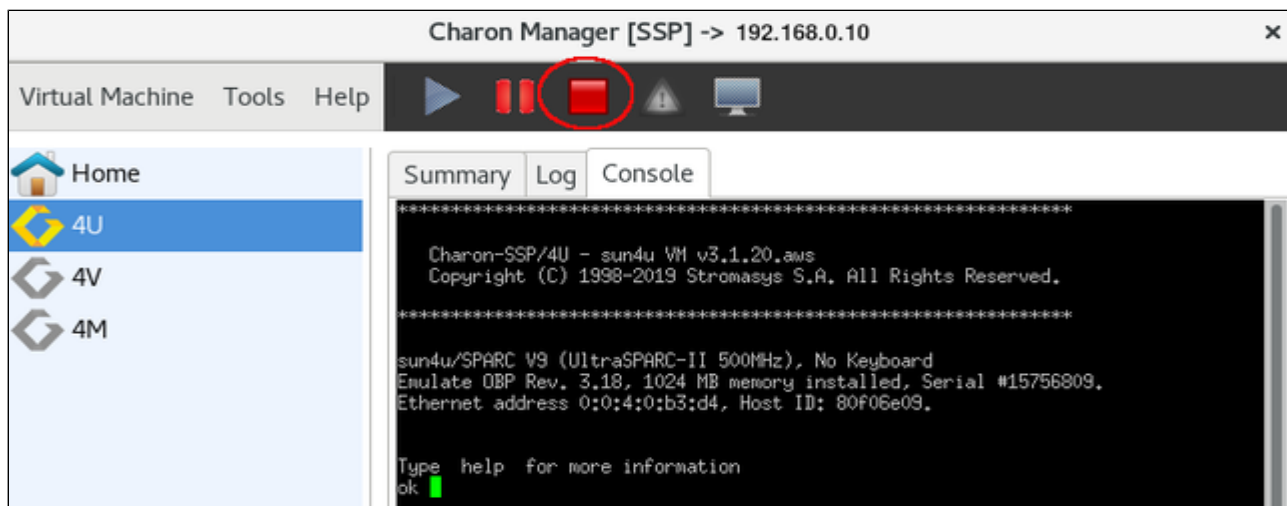


⚠ The guest operating system must be shut down cleanly or be suspended before stopping the emulator when the host system is shut down. Failing to do so may cause corruption of the guest operating system.

## Stopping the Emulated System

After shutting down the guest operating system cleanly, you can stop the emulator in Charon Manager:

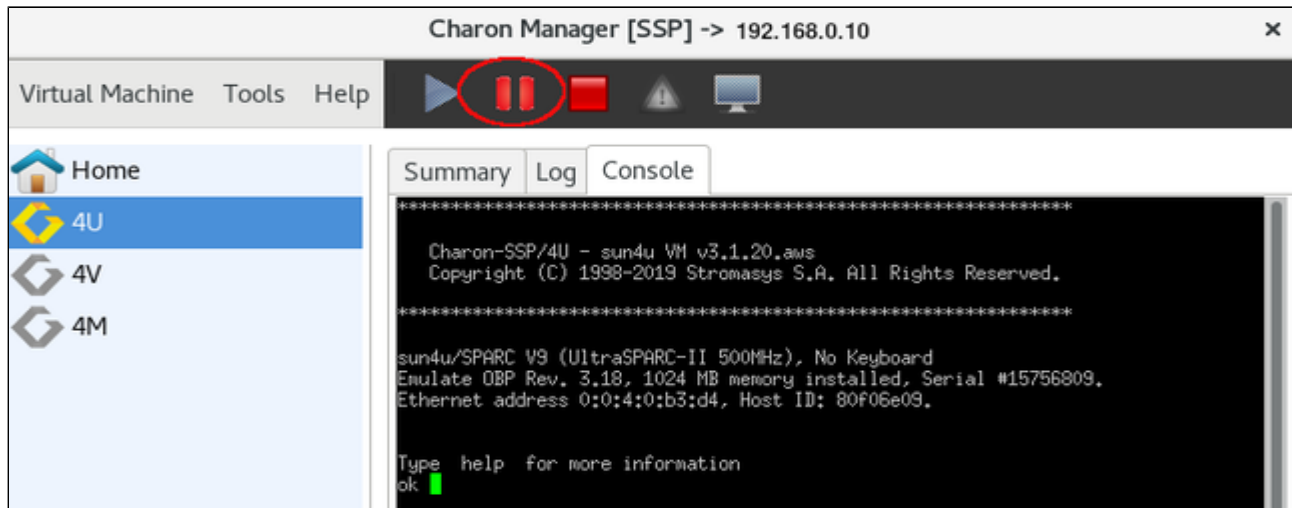
- Select the emulated system you want to stop.
- Click on the red square at the top of the window.



ℹ Alternatively, you can also stop the emulator by typing **poweroff** at the console prompt.

## Suspending the Emulated System

The emulated system can be suspended. This means that the memory content of the system is saved. Use the pause symbol at the top of the Charon Manager window to suspend the system as shown in the image below:



At the next start, the emulated system will start with the status it had when it was suspended.

The snapshot files are saved in the `/charon/storage/ssp-snapshot` directory.

# User Access to the Virtual SPARC System

## Contents

- Console Access
  - Physical Serial Console Access
  - Built-in Serial Console of the Charon Manager
  - Console Access via the Emulated Graphics Device (Charon-SSP/4M/4U(+)) only
- Other Interactive Access to the Virtual SPARC System

**Console access** to the virtual SPARC system is possible in different ways:

- Virtual serial port (appearing as a physical port to Linux)
- Built-in serial console displayed by the Charon Manager
- Graphical console (**not applicable to Charon-SSP/4V**)

There are also several **other methods for interactive access**:

- Telnet or SSH connection from a remote system via a terminal emulation program
- Graphical user interface via the emulated graphics device or a remote X-Display

## Console Access

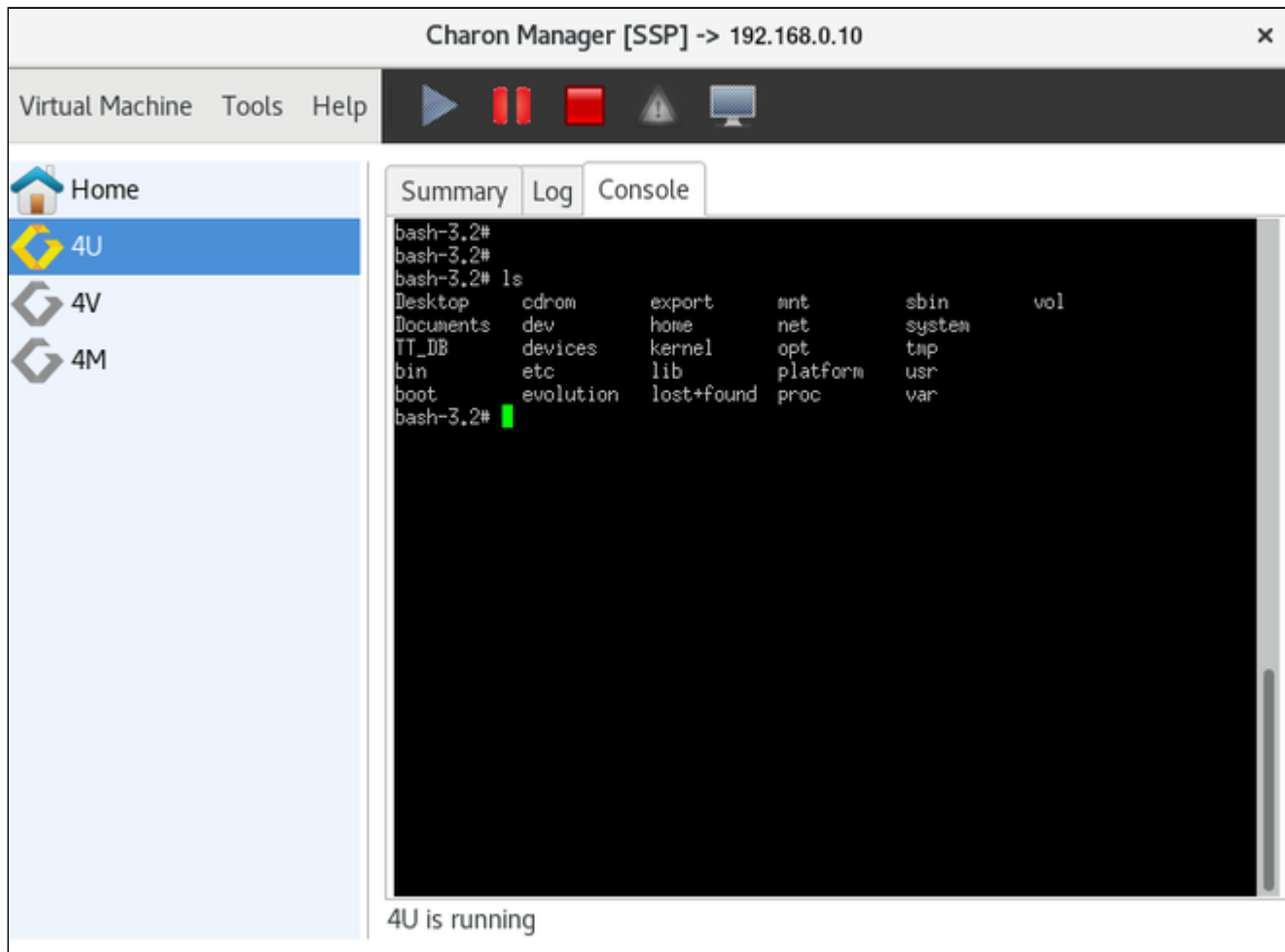
### Physical Serial Console Access

For physical console access, the virtual machine must be configured to attach the virtual serial port to a physical serial port on the host system. This configuration task is performed using the Charon-SSP Manager as shown in [Serial Line Configuration](#). Charon-SSP AWS cannot have hardware physical ports, so these ports must be emulated over a network connection.

Additional serial port configuration options, such as speed, parity, and stop-bits must be configured using the **ttymode** variable in the OpenBoot guest environment. For additional information regarding the configuration of the **ttya-mode** variable, see the [OpenBoot Console](#) in the appendix.

## Built-in Serial Console of the Charon Manager

From the Charon-SSP Manager you can access the **serial console** via the **Console** tab. The example below shows the console of a SUN-4U system.



To configure the serial console access for the Charon-SSP Manager, use a **TTYA** (4M and 4U) or a **Vconsole** (4V) configuration similar to the one below.

- The port **type** must be **TCP Raw** or **Telnet**.
- The **console** parameter must be set to **Built-in** (this is the only option in the Charon-SSP AWS product).
- The TCP **port** specified must not be used for another application or another emulated Charon-SSP serial port on the same host system.

**!** If the access to the Charon-SSP host system is via an SSH tunnel, no additional ports need to be opened in any intermediate firewall or AWS security groups. Otherwise, make sure the ports selected for the console connection are not blocked by such firewalls or security groups.

The example below shows the serial console configuration of a Charon-SSP/4U and a Charon-SSP/4V system.

### Charon-SSP/4U

Virtual Machine Settings ✕

Device	Summary
Model	SUN-4U
CPU	1
DIT	Client JIT
Memory	1 GB
Graphics	Disabled
SCSI	SCSI 0
TTYA	9000
TTYB	Disabled
Audio	Disabled
Ethernet	Disabled

#### TTYA

Operator console settings.

Type: Telnet ▾

Port:

Console: Built-in ▾

Access: Unlimited ▾

Log:

### Charon-SSP/4V

Virtual Machine Settings ✕

Device	Summary
Model	SUN-4V
CPU	1
DIT	Client JIT
Memory	1 GB
SCSI	Disabled
Vconsole	9000
TTYA	Disabled
TTYB	Disabled
Ethernet	Disabled
NVRAM	
Log	4V.log

#### Vconsole

Operator console settings.

Type: Telnet ▾

Port:

Console: Built-in ▾

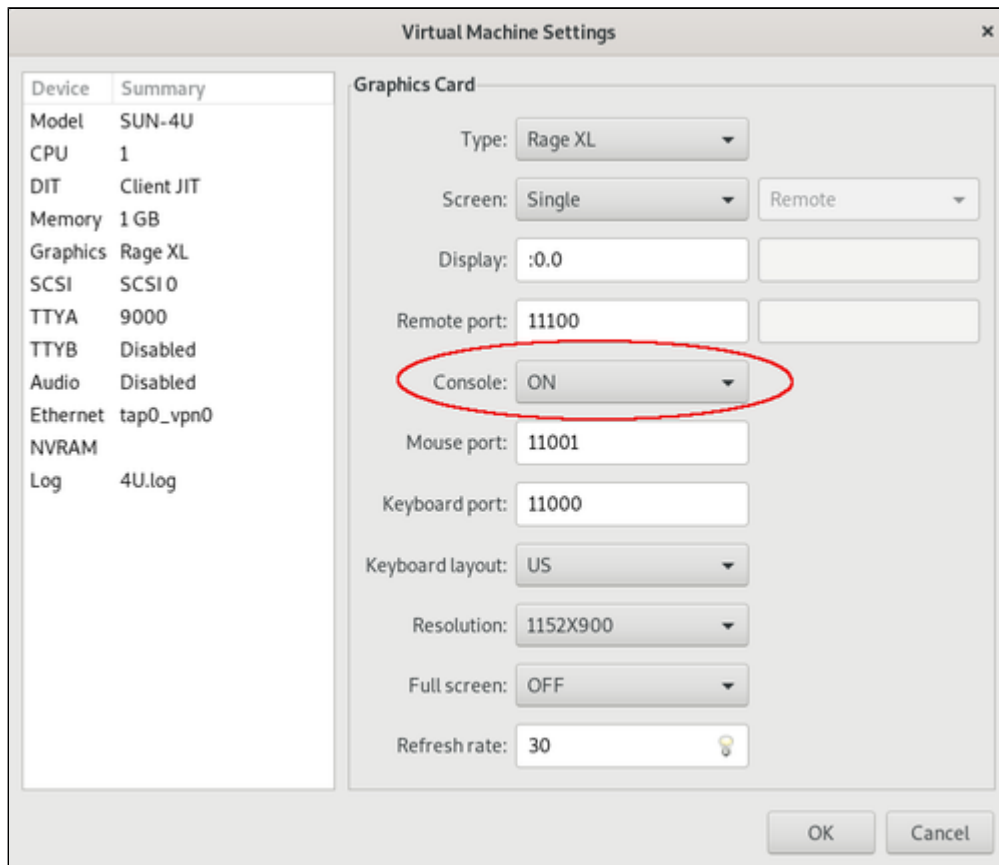
Log:  4V\_vconsole.log

**!** Only one connection to the console is possible at one time. If the Charon Manager is connected via the integrated SSH tunnel and you try to open a second connection to the console via a remote terminal emulation program, Charon Manager will terminate the second connection and re-establish the built-in console connection. If the Charon Manager is not running or not connected via the integrated SSH tunnel, a console connection can be established via a remote terminal program and the built-in console tab will be disconnected.

**i** If a second connection via Charon Manager is made to the same system, the current Charon-Manager connection is terminated.

## Console Access via the Emulated Graphics Device (Charon-SSP/4M/4U(+)) only

To enable the emulated graphics device as the console, set the **Console** configuration option to **ON**:



**!** Unless the connection is made across a VPN, the ports configured in the window above must be permitted by any intermediate firewall and/or AWS security group.

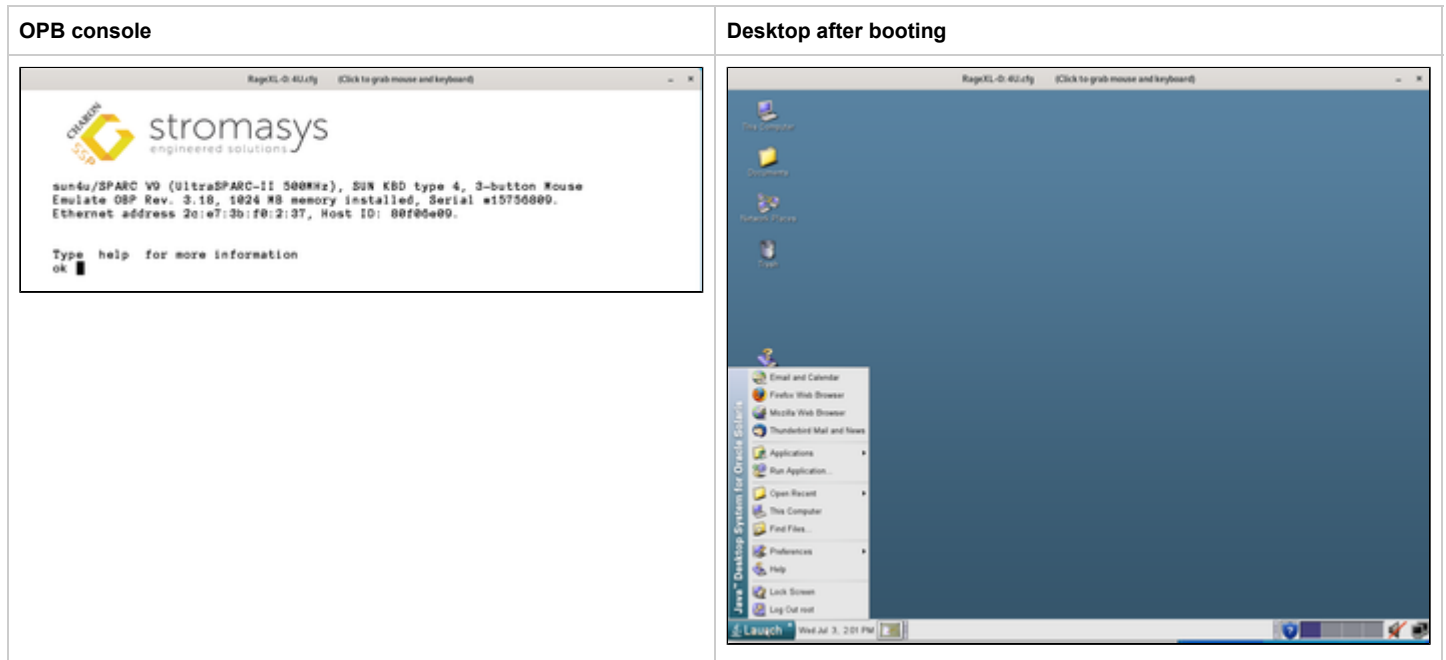
**i** When booting for the first time after adding the configuration, use boot command

```
boot <device> -r
```

to configure the graphics device correctly in the Solaris guest system.



After starting the emulated system, the graphics devices opens:



**Please note:**

The graphical performance depends on many parameters, for example, the performance of the host system, the emulated system, and the network. One important requirement is that the round-trip time of the network connection between display device and emulated Solaris system running on the AWS instance should be less or equal to 20ms.

For every use case, a test is required to evaluate the suitability for the specific customer environment.

## Other Interactive Access to the Virtual SPARC System

Once the Solaris guest has been booted, you can connect to it by several means. For example:

- Remote terminal emulation program using telnet.
- Remote terminal emulation program using SSH.
- Graphical desktop via a remote X-server configured in the Charon Manager (see [Graphical Interface via X11 Server on Linux](#)).

**i** Direct root login over the network must be allowed on Solaris if required.

- For SSH: set `PermitRootLogin yes` in `/etc/ssh/sshd_config` to allow interactive login with passwords.
- For general login: comment out the `CONSOLE=/dev/console` line in `/etc/default/login`.

# AWS Networking and Charon-SSP

## Contents

- General Information
  - Concepts
  - Address Assignment
  - Interface Names
  - Host to Guest Communication Considerations
  - External Communication Considerations
  - Using a Charon host system as a Router
  - Guest to Guest Layer 2 Communication Considerations
  - Asymmetric Routing Considerations
- Further Information

## General Information

This section provides some basic information about AWS networking that is likely to affect Charon-SSP when running in the cloud.

**i** NetworkManager is disabled on Charon-SSP AWS. Therefore, the interface configuration relies on **ifcfg**-files in **/etc/sysconfig/network-scripts**.

**!** The information is not comprehensive. Please refer to the Amazon AWS documentation for up-to-date and comprehensive information.

## Concepts

**VPC:** VPC stands for virtual private cloud. It is a virtual network associated with your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC. You can specify an IP address range for the VPC, add subnets, associate security groups, and configure route tables. In addition to the default VPC, an account can have non-default VPCs. The default VPC includes an Internet gateway.

**Subnet:** A subnet is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. There are public subnets that can be connected to the Internet and private subnets that have no direct Internet connection. Instances in a private subnet can reach the Internet via a NAT gateway.

**Instance:** An instance is a virtual machine that is launched into a VPC. It is associated with an image (e.g., Charon-SSW AMI) and a certain instance type representing the virtual hardware.

## Address Assignment

Each VPC is assigned a block of private IP addresses. This block can be split by the user to form several IP subnets. Routing between such subnets is automatically enabled.

When an E2C instance is launched into the default VPC and a public subnet, the **default** behavior is as follows:

- If the instance has only one network interface, it is automatically assigned a private IP address and a public IP address from the address range assigned to the public subnet. This network interface is the primary network interface. It is called eth0.
- If the instance has more than one network interface, it is automatically assigned a private IP address for each of the network interfaces - but no public IP address.

The default behavior can be modified, for example:

- Manually assigning a private IP address from the subnet range.
- Enabling or disabling the automatic assignment of a private IP address to deviate from the subnet setting.
- Manually assigning a public IP address from the AWS range or the customer range.


**i** Public IP addresses are not directly visible to the instance. The instance operating system always works with the private address. For external connections, the private address is mapped to the public IP address via NAT.

**Reserved addresses (important, if manual address assignment is used):**

The following address range is reserved to allow AWS to query meta-data about instance configuration: 169.254.0.0/16. This range is automatically configured on every network interface.


The following addresses are reserved in each subnet and cannot be used for E2C instances (shown in the example below for network 10.1.1.0/24):

- 10.1.1.0: the network address
- 10.1.1.1: reserved by AWS for the VPC router
- 10.1.1.2: reserved by AWS in any subnet; the second host address in the base VPC network range is the DNS server for the VPC.
- 10.1.1.3: reserved by AWS for future use
- 10.1.1.255: network broadcast address; AWS networks do not use broadcasts.

 An automatically assigned public IP address is released (and not re-assigned) by AWS for example if

- a second interface is added to the instance and the instance is then stopped and restarted,
- an Elastic IP is associated with the the instance,
- an Elastic IP address is associated with the primary interface of the instance.

See <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-instance-addressing.html> for details.

 An automatically assigned public IP address is not persistent. Every time a instance starts, it is assigned a new public IP address. If persistent public addresses are needed, use Elastic IP addresses.


## Interface Names

When looking at the instance from the AWS management console, the interface names are **eth0**, **eth1**, etc.


On instances **without support for enhanced networking** the Linux interface names are also **eth0**, **eth1**, etc.

However, on **instances with support for enhanced networking**, the names on the AWS level differ from those on the Linux level:

- The first (primary) interface is called **ensX** (where **X** is an integer denoting the interface number; example: ens5).
- When a second interface is added to a running instance, it maybe initially called **eth0**. However, the command `ethtool -i eth0` shows that the enhanced network driver (ena) will be used for this interface. This interface will change its name to **ensY** (where **Y** is X+1) after restarting the instance. This means that any configuration file created for this interface must use the final name of the interface instead of eth0. Otherwise, the instance may become unreachable after a restart because there is no valid interface configuration (NetworkManager is not enabled on Charon-SSP AWS, so a configuration file must exist to configure the interface properly).

 This numbering scheme may change in the future. Currently, it is based on the PCI slot on which the Ethernet controller is presented and which is incremented by one for each additional Ethernet interface added. On the Charon host system, the slot can be verified with the following command:

```
# lspci -vv | grep -A20 Ethernet
```

-  To avoid confusion before the instance can be restarted, the new interface can be renamed to its final name using the command `ip link set eth0 name ensY && ip link set ensY up`

## Host to Guest Communication Considerations

There are several ways a communication between the host operating system and the guest Solaris system can be implemented. For example:

### 1. Internal virtual bridge on the host system:

Such a bridge has several TAP interfaces. The host and the guest systems are connected to this bridge and can communicate directly to one another using L3 and L2 protocols. The bridge uses its own IP subnet that can be defined by the user. Setting up such a configuration is supported by the Charon Manager.

### 2. Communication via the AWS subnet LAN:

In this case, a second interface is added to the Charon host system. The second interface is then assigned to the emulated guest system. After the correct configuration, the host and guest can communicate across the AWS LAN using IP, L2 protocols or any protocols that require changing the MAC address to something different than the MAC address assigned to the second interface by AWS will not work.

To connect the guest system to the LAN, the following basic configuration steps must be performed:

- Add the additional interface to the Charon host system.
- Create a configuration file for the additional interface.
- Remove the private IP address assigned to the second interface by AWS from the Linux configuration (if it has been configured).
- Use Charon Manager to assign the interface to the emulated SPARC system.
- Use Charon Manager to set the MAC address of the emulated SPARC system to the same value as the one used on the host system Ethernet interface.
- On the Solaris system, configure the private IP address that was previously assigned to the second interface on Linux and configure the appropriate default route for the LAN.
- Additional steps may be required:
  - If the primary interface has an automatically assigned public IP address, this will be released when the instance is stopped and restarted. Hence the configuration must be changed to use a persistent Elastic IP address first to maintain reachability of the host system.

**i** The section [Example of a More Complex Network Configuration](#) provides some hints on how to configure the second interface in the different situations. Please refer to the AWS documentation for up-to-date comprehensive information.

**i** If Layer 2 communication between guests on different Charon hosts is required, a bridged tunnel solution must be set up between the two Charon host systems.

## External Communication Considerations

In addition to allowing SSH access to the host system for management purposes, it may be necessary to enable Internet communication to the host and guest system or connect host and guest to the customer's network.

### Recommended way to connect the Charon host and Solaris guest systems to the customer network:

To ensure data traffic between the Charon host and guest systems and the customer network is encrypted, it is strongly recommended to use a VPN connection. An example of a simple VPN connection based on an SSH tunnel is described in [SSH VPN - Connecting Charon Host and Guest to Customer Network](#). This connection is based on a bridge between Charon host and guest system and (via an encrypted SSH tunnel) the remote end-point in the customer network. The connection supports L3 and L2 protocols.

AWS also provides a VPN gateway instance that can be added to the customer VPC to connect the VPC to the customer network (for a charge).

### Recommended way to connect the Solaris guest system to the Internet:

The Internet connection can be implemented across the VPN to the customer network. In this case, the customer can allow the guest Solaris system to access the Internet exactly following the security policies defined by the customer.

### Access to the Internet from private VPC subnets or a Solaris guest system with only private IP addresses:

Access to the Internet for private VPC subnets is possible across a gateway instance providing VPN access to the customer network and allowing (NATted) Internet access via this path. Alternatively, a NAT gateway in the cloud can be used to map the private addresses to public addresses. The NAT gateway can be implemented on a Charon host system or it can be provided by AWS for a charge.

### Direct Solaris guest access to the Internet:

This not a recommended standard solution for security reasons. However, should it be required, two interfaces with public IP addresses can be assigned to the Charon host.

One of these interfaces is then dedicated to the guest system which uses the private interface address and the MAC address assigned to the Charon host by AWS (similar to point 2 in section *Host to Guest Communication Considerations* above).

## Using a Charon host system as a Router

If a Charon host system is to be used as a router (for example as shown in [Example of a More Complex Network Configuration](#) or to provide Internet connectivity to other Charon host and guest systems), it is not sufficient to configure Linux for IP forwarding.

The following settings have to be made on the Charon host instance via the AWS management console:

For each interface, the **source/destination check** has to be disabled. Unless this is configured correctly, traffic from and to an AWS instance will only be allowed if either source or destination address belongs to the instance. Transit traffic destined to be forwarded by the router, would be discarded.

## Guest to Guest Layer 2 Communication Considerations

Should L2 protocols be required between two guest systems on different host systems, a bridge/tunnel solution similar to the one described in [SSH VPN - Connecting Charon Host and Guest to Customer Network](#) must be set up between the two host systems to allow the L2 traffic to pass.


## Asymmetric Routing Considerations


This section applies to the case where several interfaces are configured on an instance and they all have IP addresses configured on the Linux level.

From the AWS documentation (<https://aws.amazon.com/premiumsupport/knowledge-center/ec2-ubuntu-secondary-network-interface/>):

"Adding a secondary network interface to a non-Amazon Linux EC2 instance causes traffic flow issues. These issues occur because both the primary and the secondary network interfaces are in the same subnet, and there is only one routing table with one gateway. Traffic that comes into the secondary network interface leaves the instance using the primary network interface. But this isn't allowed, because the secondary IP address doesn't belong to the MAC address of the primary network interface.

To make the secondary interface work, create a secondary network configuration file, configure the routing table, and then set up rules in the custom routing table policy database so that traffic for the secondary interface uses the new routing table."


 The above documentation only describes the required steps for Ubuntu. An earlier article for CentOS and Red Hat was removed from the AWS site. **So the information presented here may change in the near future.**

 Review the section about interface names if using an instance with enhanced networking enabled.

When adding a second IP interface (for example **eth1**) to the same subnet as the first on the Charon-SSP host, the routing problems described above can occur. To solve them, perform the following basic steps.

1. Create a configuration file (`/etc/sysconfig/network-scripts/ifcfg-<interface-name>`) for the second interface (if there is no configuration file for the primary interface, create it as well).
2. Set the correct interface for default route in `/etc/sysconfig/network` (example: `GATEWAYDEV=eth0`).
3. To prevent the cloud-init from resetting your custom network configurations, add the following lines to the `/etc/cloud/cloud.cfg` file:
 

```
network:
  ; config: disabled
```
4. Restart the network.
5. Create an additional routing table (use the command: `ip route add <path> dev <interface-name> table <table-id>`). There must be an entry for every IP address assigned to the second interface and any other route to be used.
6. Set rules in the Routing Policy Database (use the command: `ip rule add from <ip-address-of-second-interface> lookup <table-id>`)
7. Create a static route file (`/etc/sysconfig/network-scripts/route-<interface-name>`)
8. Create a static rule file (`/etc/sysconfig/network-scripts/rule-<interface-name>`)

 Please refer to the Linux man pages for **ip rule** and **ip route** for more information. The AWS example for Ubuntu may also provide helpful hints.

## Further Information

The following sections show sample network configurations:

- [SSH VPN - Connecting Charon Host and Guest to Customer Network](#)
- [Example of a More Complex Network Configuration](#)

## SSH VPN - Connecting Charon Host and Guest to Customer Network


### Contents

- Contents
- Overview
  - Prerequisites
- Setting up the VPN Tunnel
  - Steps on the Charon-SSP Host System
    - Creating a VPN Bridge
    - Assigning the Guest Ethernet Interface
  - Steps on the Remote Linux System
  - Steps on the Solaris Guest System
- Stopping the SSH Tunnel
- Routing to/from Solaris Guest

### Overview

If the connection between the Charon-SSP host system, including the configured Charon-SSP guest systems, and the rest of the customer's network runs over a public network as is the case for Charon-SSP AWS instances, it is necessary to secure the traffic against unauthorized access.

The example in this section describes how to configure a bridged SSH-based VPN tunnel between the Charon-SSP AWS host and a remote Linux system across a public network. Topologies that are more complicated will require other, more sophisticated, solutions.

 The customer is responsible for ensuring that any VPN solution meets the requirements of his or her company's security guidelines. The example in this chapter is only for illustrative purposes.


 The advantage of a bridged connection is that L2 protocols are also supported.

Once the sample configuration has been set up, it can be used for

- communication between host and guest system,
- communication between customer network and guest system.

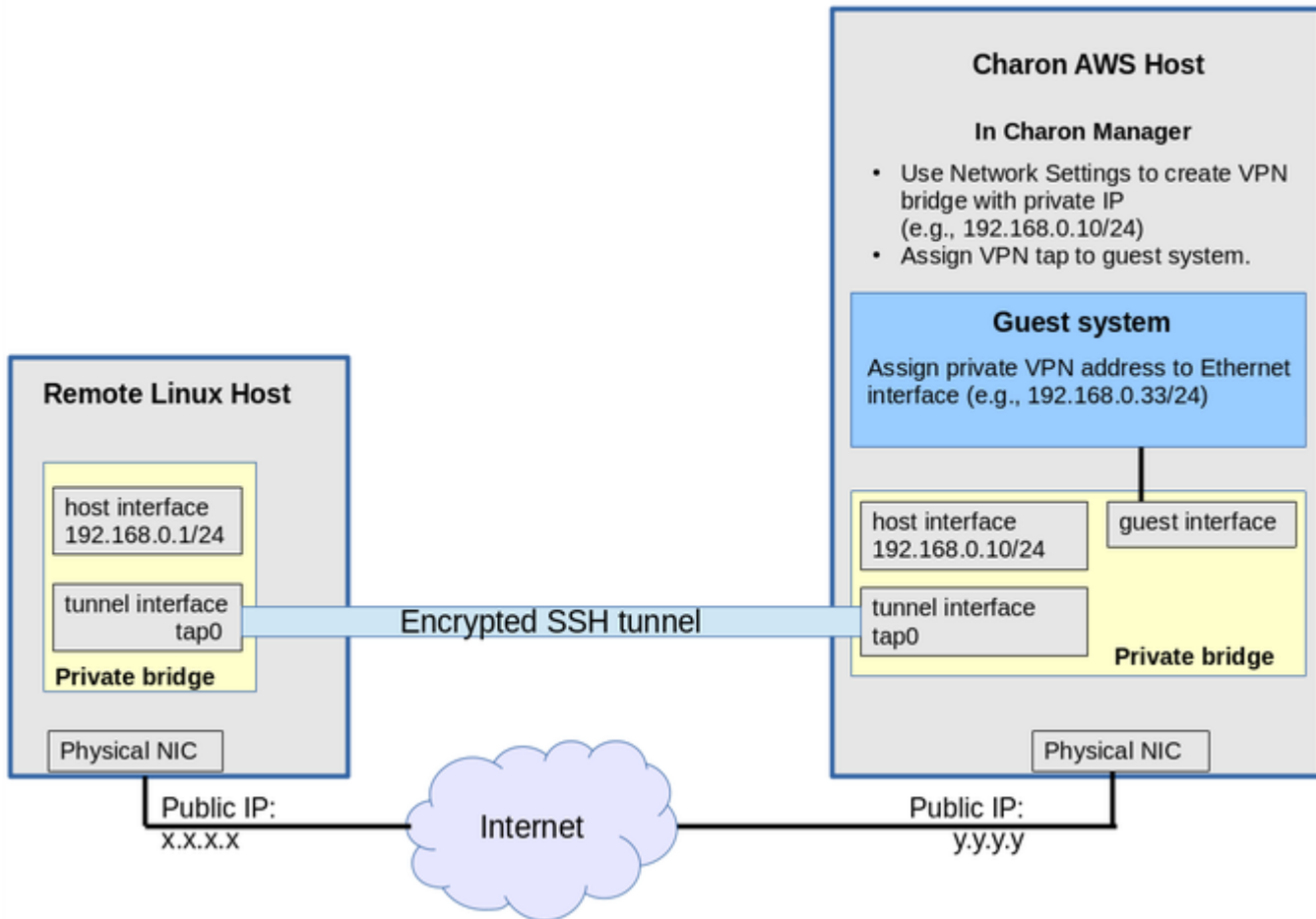
### Prerequisites

The example shows how to use the Charon Manager on the Charon-SSP AWS host and a set of commands on the remote Linux System to create an SSH VPN tunnel. For this configuration to work, the following prerequisites must be met:

- The remote Linux system must have access to the public IP address and the SSH port of the Charon-SSH AWS host.
- The private key necessary to access the instance must be available on the remote Linux system. The key-pair required to access the AWS instance is created via the AWS EC2 dashboard and associated with the instance when it is created.
  -  The private key can be downloaded only once when the key is created. If it is lost, access to the instance may be permanently lost.
- The *bridge-utils* and *autossh* packages must be installed on the remote Linux system.

## Setting up the VPN Tunnel

The image below shows a sample setup. This section describes how to configure this sample setup.



## Steps on the Charon-SSP Host System


### Creating a VPN Bridge

To configure the SSH VPN connection, you must setup a private VPN bridge (called a virtual network in the Charon context) using the Charon Manager. Use the following steps to perform this task:

1. Open the Charon-SSP Manager and log in to the Charon-SSP host.
2. In the Charon Manager, open the Network Settings window by clicking on **Tools > Network Settings**. This will open the **Network Settings** window.
3. Click on **Add** and then on **Virtual Network** to open the virtual network configuration window. This will open the **Add Virtual Network** configuration window as shown below.
4. Enter the required information as shown below:

Perform the following steps to configure a VPN bridge,

- Set **Create for SSH VPN** to **ON**.
- Enter the **Number of virtual adapters** (TAP interfaces) required. These interfaces will be assigned to the emulated SPARC systems as Ethernet interfaces.
- Configure the **IP address** for the bridge interface.
- Set the **Netmask**.

 This interface and the interface on the remote Linux system must be in the **same IP subnet**.

Click on **OK** to save your configuration.

### Add Virtual Network

Create for SSH VPN:	<input type="text" value="ON"/>
Binding interface:	<input type="text" value="OFF"/>
STP for bridge:	<input type="text" value="OFF"/>
Virtual bridge interface:	<input type="text"/>
Virtual bridge name:	<input type="text" value="vpn0"/>
Number of virtual adapters:	<input type="text" value="1"/>
IP settings:	<input type="text" value="Manual"/>
IP address:	<input type="text" value="192.168.0.10"/>
Netmask:	<input style="border: 2px solid blue;" type="text" value="255.255.255.0"/>
Gateway:	<input type="text"/>
DNS server 1:	<input type="text"/>
DNS server 2:	<input type="text"/>

To learn more about the virtual network configuration options, refer to section [Host System Network Configuration](#).

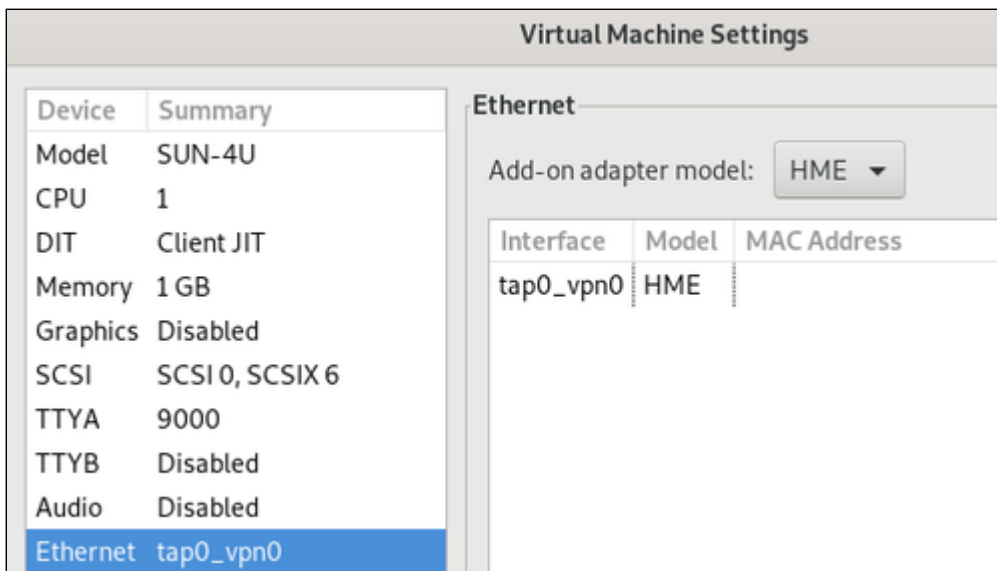
## Assigning the Guest Ethernet Interface

One of the TAP interfaces created in the step above, must be assigned to the Solaris guest system to add it to the LAN that will be tunneled across SSH to the remote Linux system.

Perform the following steps:

1. Open the Charon-SSP Manager and log in to the Charon-SSP host.
2. In the Charon Manager, select the guest system and then the **Ethernet** configuration category on the left. Assign one of the created TAP interfaces to the guest (see example below).





Click on **OK** to save the configuration change.

**i** If the emulated instance is currently running, the guest must be shut down and the emulated instance must be restarted for the change to become active.

## Steps on the Remote Linux System

**!** The steps on the Charon-SSP host must be performed first.

As the user **root** on the remote Linux system, perform the following steps to set up the VPN tunnel according to the overview image above:

Action	Command
Create TAP interface	<code># ip tuntap add dev tap0 mod tap</code>
Enable TAP interface	<code># ip link set tap0 up</code>
Create bridge	<code># ip link add name br_vpn0 type bridge</code>
Enable bridge interface	<code># ip link set br_vpn0 up</code>
Define IP address for bridge	<code># ip addr add 192.168.0.1/24 dev br_vpn0</code>
Add TAP interface to bridge	<code># ip link set tap0 master br_vpn0</code>
<b>Start the SSH tunnel</b>  <b>autossh</b> is a program to start a copy of <b>ssh</b> and monitor it, restarting it as necessary should it die or stop passing traffic.  Once started, you can move the program to the background.	<code># autossh -M 9876 -o ServerAliveInterval=60 -o Tunnel=ethernet \</code> <code>-w 0:0 -t -i &lt;path-to-private-key&gt; -NCT sshuser@&lt;public-AWS-instance-IP&gt;</code>  <b>-M</b> defines the monitoring port autossh uses to monitor the connection <b>-o</b> sets SSH options (bridged tunnel and keepalive) <b>-i</b> denotes the path to the private key matching the public key copied to the host system. <b>-w</b> denotes the number of the local and remote tunnel interfaces for tunnel device forwarding (e.g., the 0 in interface tap0). <b>-N</b> denotes that no remote command should be executed <b>-T</b> disables pseudo-terminal allocation <b>-C</b> requests data compression

### Possible additional steps:

- Enable IP forwarding on the remote Linux system if it is to act as a router between the tunnel connection and other systems in the customer network:  
`# /sbin/sysctl -w net.ipv4.ip_forward=1`  
 (to make permanent: add the setting to /etc/sysctl.conf)
- Add static or dynamic routes to distribute the tunnel subnet to other systems in the customer network that need to communicate with the Solaris guest system across the VPN..
- Adapt the firewall on the remote Linux system as required to allow the VPN traffic to pass.

## Steps on the Solaris Guest System

Set the IP address on the Ethernet interface to an address within the VPN subnet. To follow the example above, you would set the address to 192.168.0.33/24. To permanently change the IP address on the Solaris system, change the address in `/etc/hosts` for the hostname specified in `/etc/<interfacename>.hostname`.  
 On Solaris 11, use the commands `ipadm create-ip netX` and `ipadm create-addr -T static -a <ip-address>/<netmask> netX/v4`.

## Stopping the SSH Tunnel

To stop the SSH tunnel, perform the following steps on the remote Linux system:

Action	Command
Terminate the autossh process	<code># kill -9 &lt;autossh-pid&gt;</code>
Terminate remaining SSH tunnel connections	<code># kill -9 &lt;tunnel-ssh-pid&gt;</code>
Delete the bridge	<code># ip link delete br_vpn0</code>
Delete the TAP interface	<code># ip link delete tap0</code>

## Routing to/from Solaris Guest

After following the description above, the Solaris guest system can be reached from the systems that are also connected to the virtual bridge (in the example: remote Linux system and host system). To enable the Solaris guest system to **communicate with other systems** in the customer network (or the Internet) over the VPN connection, perform the following steps:

- Add the VPN address of the remote Linux system as the default gateway for the Solaris guest system.
- Propagate the IP network used for the SSH VPN within the customer network, as required.
- Enable IP forwarding on the remote Linux system and allow forwarded packages through the firewall.

The screenshot below illustrates the Solaris guest system behavior (after the VPN network has been made known within the customer LAN and the remote Linux host has been set up as a router):

- The interface address shows that the Solaris system is in the 192.168.0.0/24 network using the `ifconfig` command.
- The `netstat -rn` command shows the routing table without a default route.
- The ping to an IP address outside the SSH VPN fails.
- The `route add default <gateway>` command adds the remote Linux host as the default gateway.
- The `netstat -rn` command now shows the default route.
- The ping to an IP address outside the SSH VPN succeeds.

```
bash-3.2# ifconfig hme0
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 192.168.0.33 netmask ffffffff broadcast 192.168.0.255
      ether d4:2:7c:c1:d2:59
bash-3.2#
bash-3.2# netstat -rn

Routing Table: IPv4
  Destination          Gateway              Flags  Ref    Use  Interface
-----
192.168.0.0           192.168.0.33        U        1      1  hme0
224.0.0.0             192.168.0.33        U        1      0  hme0
127.0.0.1             127.0.0.1           UH       4     136  lo0
bash-3.2#
bash-3.2# ping 192.168.2.80
no answer from 192.168.2.80
bash-3.2#
bash-3.2# route add default 192.168.0.1
add net default: gateway 192.168.0.1
bash-3.2#
bash-3.2# netstat -rn

Routing Table: IPv4
  Destination          Gateway              Flags  Ref    Use  Interface
-----
default               192.168.0.1         UG       1      0
192.168.0.0           192.168.0.33        U        1      1  hme0
224.0.0.0             192.168.0.33        U        1      0  hme0
127.0.0.1             127.0.0.1           UH       4     136  lo0
bash-3.2#
bash-3.2#
bash-3.2# ping 192.168.2.80
192.168.2.80 is alive
bash-3.2#
```

To make the entry permanent


- on Solaris 10: use the `route -p` command (stores routes in `/etc/inet/static_routes`).
- on older Solaris versions: add the address of the default gateway to `/etc/defaultrouter`.


## Example of a More Complex Network Configuration

### Contents

- Introduction to Example
  - Gateway Configuration
    - Activate the Second Network Interface with Elastic IP
    - Make the Client Instance Reachable from the Gateway
    - Create an SSH VPN between Gateway and Customer Network
    - Enable Routing on the Gateway
    - Adapt Security Group on Gateway
    - Set up NAT on the Gateway
  - Client Configuration
    - Change Default Route
    - Adapt Security Group on Client
    - Add a Non-Default DNS Server
    - Add Second Network Interface for Solaris Guest
  - Solaris Guest Configuration
  - Remote Linux Configuration
  - Result

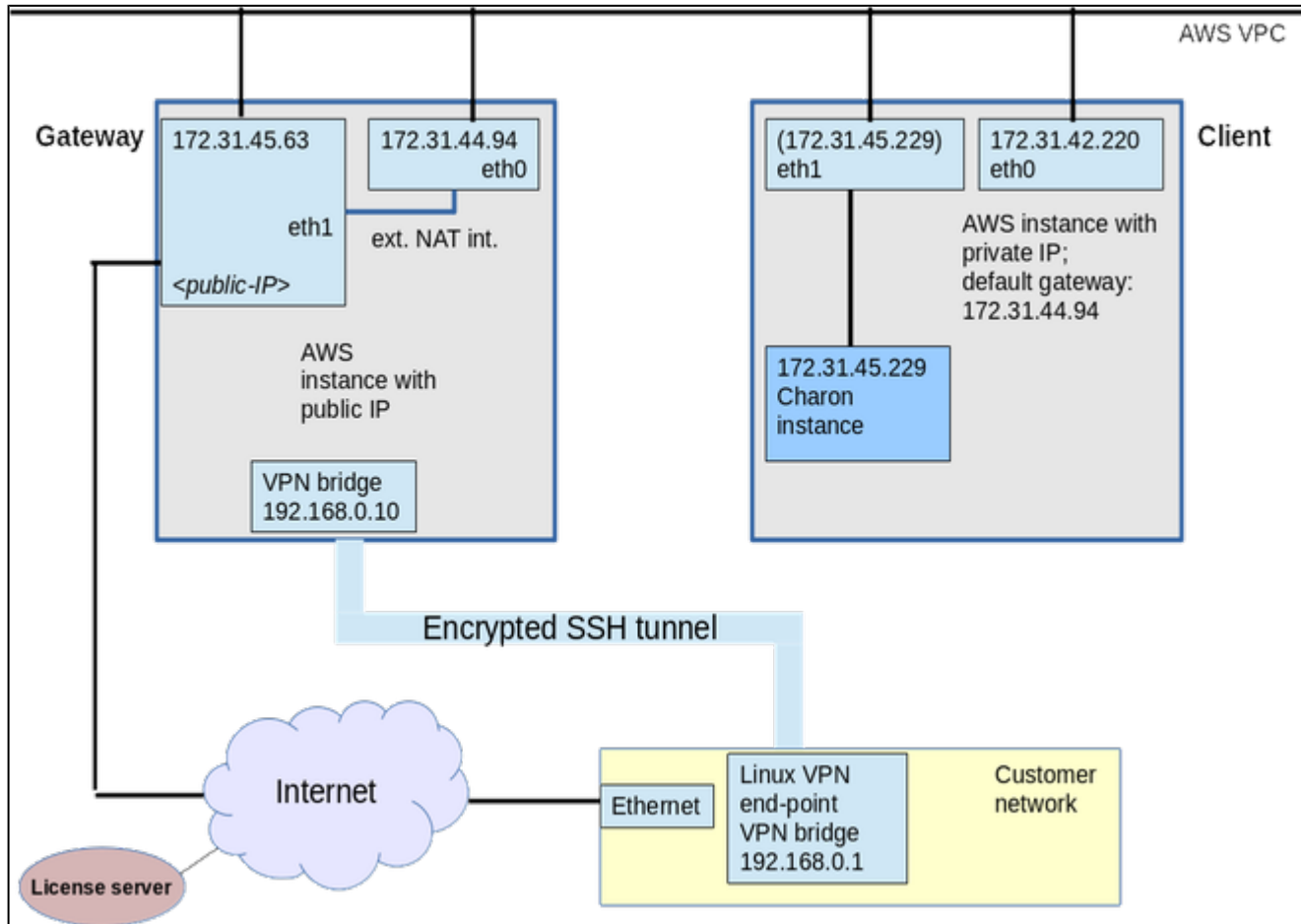
### Introduction to Example

 The customer is responsible for ensuring that any VPN solution and any access to the Internet meets the requirements of his or her company's security guidelines. The example in this chapter is only for illustrative purposes. Please refer to the AWS documentation for up-to-date information.

 Throughout this example, the physical interface names used on the Charon host are **eth0** and **eth1**. If your instance supports enhanced networking, the interface names are in the format **ensX**. If this is the case, please review the section about interface names in [AWS Networking and Charon-SSP](#) before you create any interface configuration files.

This example is less a blueprint for implementation than an illustration of certain features of AWS networking. AWS offers prepackaged NAT and VPN gateways that can be used by customers for a fee. Such prepackaged solutions should be evaluated in addition to what is shown in this document.

This section describes a sample configuration as depicted below:



In this example, there are two Charon-SSP AWS instances:

- Each instance has two Ethernet interfaces.
- Both are connected to the same subnet. This enables IP communication between host and guest systems (no L2 protocols).
- Only one instance (called the *gateway* in this section) has a public IP address on interface **eth1**.
- The gateway offers a VPN connection to the customer network and a NAT service for Internet access to instances that only have private IP addresses (called *clients* in this section). The NAT service can be used by the Solaris guest. For the Charon-SSP AWS host with only private IP addresses the NAT connection implements the Internet access required for reaching the license servers.
- On the instance with only private IP addresses (called the *client* in this section, one interface is dedicated to the host system, the other to a Solaris guest system).
- The Solaris guest system can reach the customer network via the VPN and the Internet via the NAT service offered by the gateway.

The steps to implement the sample configuration are described below. It is assumed that you have launched two basic Charon-SSP AWS instances, one with an auto-assigned public IP address (the gateway) and one with a private IP address only (the client). At the beginning of, each instance should only have one Ethernet interface and they should be in the same subnet. The security group assigned should at least allow SSH to enable initial access. Allowing also ICMP makes testing easier.

**i** All IP addresses used in the description below refer to the sample configuration shown in the diagram.

**i** The client has IP connectivity only. For Layer 2 protocols, a bridge with VPN tunnel would have to be configured between client and gateway.

**!** The configuration steps are based on the AWS environment at the time of writing. Changes in the AWS environment are outside the control of Stromasys.

## Gateway Configuration

### Activate the Second Network Interface with Elastic IP

1. Create and attach a new interface with an Elastic IP address to your instance (see [Network Management](#)).
2. Log into the gateway via SSH (see [Accessing the Charon-SSP AWS Instance](#)) using the auto-assigned public IP address.
3. Become the root user (enter `sudo -i`).
4. Create an interface configuration file for the second interface:
 

```
# cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth1
```
5. Edit this file, remove the hardware address line, and change the name of the interface to eth1.
6. Restart the network:
 

```
# systemctl restart network
```
7. Your SSH session will be disconnected.

Expected result:

- After restarting the instance, the automatically assigned public IP address will be released by AWS. Until then, it can be used if a second routing table and routing rules are created to ensure correct traffic flow for each interface.
- Gateway instance is reachable via Elastic IP address on eth1.

**i** The default gateway is automatically changed to eth1 with the Elastic IP address.

**i** In a VPC environment, auto-assigned private IP addresses are persistent (according to the AWS documentation at the time of writing).

### Make the Client Instance Reachable from the Gateway

1. Copy the private key required by the client to the gateway using SFTP (see [Accessing the Charon-SSP AWS Instance](#)).
2. As the `root` user on the gateway copy the private key to `~sshuser/.ssh/`.
3. Make the `sshuser` the owner of the `.ssh` directory and content:
 

```
# chown -R sshuser:sshuser ~/.sshuser/.ssh/
```
4. Set permissions on the keyfile:
 

```
# chmod 400 ~sshuser/.ssh/<keyfile>
```
5. As `sshuser` log in to the client from the gateway via SSH (see [Accessing the Charon-SSP AWS Instance](#)).

### Create an SSH VPN between Gateway and Customer Network

To create this VPN, follow the steps in [SSH VPN - Connecting Charon Host and Guest to Customer Network](#).

### Enable Routing on the Gateway

Since the gateway will have to forward packages from the client to the customer network and/or the Internet, IP forwarding must be enabled. In addition, AWS specific source/destination IP address checks must be disabled.

1. Enable IP forwarding:
 

```
# /sbin/sysctl -w net.ipv4.ip_forward=1
```

 (to make permanent: add the setting to `/etc/sysctl.conf`)
2. Disable source/destination checking on all interfaces of the gateway instance:
  - a. In the AWS EC2 dashboard, select **Network Interfaces** on the left.
  - b. Select the interface representing eth0 of the gateway instance.
  - c. Select **Change Source/Dest. Check** under **Actions**.
  - d. Disable the check.
  - e. Repeat steps b to d for eth1.
3. Allow forwarding through firewall:
 

```
# firewall-cmd --permanent --direct --add-rule ipv4 filter FORWARD 0 -i br_vpn0 -o eth0 -j ACCEPT
# firewall-cmd --permanent --direct --add-rule ipv4 filter FORWARD 0 -o br_vpn0 -i eth0 -j ACCEPT
```

### Adapt Security Group on Gateway

Change the security group content such that **all traffic from the client is allowed**.

1. In the AWS EC2 dashboard, select **Security Groups** on the left.
2. Select the security group used for the gateway instance.
3. Select **Edit Inbound Rules** in **Actions**.
4. Adapt the rules as required.
5. If needed, repeat for outbound rules.

## Set up NAT on the Gateway

To create a basic NAT configuration on the gateway instance, use the predefined zones **internal** and **external** of **firewalld**. Firewalld performs address translation between these two zones.

1. Make sure the firewall is enabled.
2. Move eth0 (with private IP address only) to the internal zone:  

```
# firewall-cmd --change-interface=eth0 --zone=internal --permanent
```
3. Move eth1 (with public IP address) to the external zone:  

```
# firewall-cmd --change-interface=eth1 --zone=external --permanent
```
4. Add DNS to the internal zone as an allowed service and add the web-cache port (required for license operation):  

```
# firewall-cmd --zone=internal --add-service=dns --permanent  
# firewall-cmd --zone=internal --add-port=8080/tcp --permanent
```
5. Reload the firewall:  

```
# systemctl restart firewalld
```

## Client Configuration

### Change Default Route

The client should use the internal interface of the gateway as its default gateway.

1. Delete the original default route:  

```
# ip route delete default via 172.31.32.1 dev eth0
```
2. Add the new default route:  

```
# ip route add default via 172.31.44.94 dev eth0
```
3. Make the route permanent by using the Network Settings option of Charon Manager and making the interface configuration static (manual configuration). Take care not to make your instance unreachable by entering incorrect data.

### Adapt Security Group on Client

Change the security group content such that **all required traffic from the customer network and other sources is allowed**.

1. In the AWS EC2 dashboard, select **Security Groups** on the left.
2. Select the security group used for the gateway instance.
3. Select **Edit Inbound Rules** in **Actions**.
4. Adapt the rules as required.
5. If needed, repeat for outbound rules.

### Add a Non-Default DNS Server

The name resolution for the client does not seem to work with the default AWS name server (it works with the gateway instance). Configure a non-default name server (either a public name server on the Internet or a name server in the customer network) by using the Network Settings option of the Charon Manager (add a non-default DNS server to the static interface configuration).

### Add Second Network Interface for Solaris Guest

The Solaris guest system should have a dedicated network interface. To achieve this, perform the following steps:

1. Create a new network interface in the same subnet with only a private IP address and attach it to the running client instance (see Network Management). Make a note of the private IP address assigned to the interface.
2. Create an interface configuration file for the second interface (the file for the first one should exist in Charon-SSP AWS):  

```
# cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth1
```
3. Edit this file to fit the characteristics of **eth1**. The private IP address used for this interface will be assigned to the Solaris guest. Therefore, configure the Linux Interface without IP address, similar to the following example:

```
BOOTPROTO=none
DEVICE=eth1
NAME=eth1
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
```

4. Make sure the default interface stays on eth0 by adding the following line to **/etc/sysconfig/network**:  

```
GATEWAYDEV=eth0
```
5. Prevent the cloud setup from changing your network configuration by adding the following lines to **/etc/cloud/cloud.cfg**:  

```
network:
  ; config: disabled
```
6. Restart the network:  

```
# systemctl restart network
```

Expected result:

1. The system should still be reachable via **eth0**.
2. Interface **eth1** should be up with out having an IP address configured.



## Solaris Guest Configuration

Using the steps below, the Solaris guest system is configured to use the second NIC. The IP addresses used follow the example shown in the image above.

1. Configure a guest Solaris system with **eth1** as its interface. Set the MAC address to the same value as used by the host interface.
2. Boot Solaris and configure the IP address assigned to interface eth1 for the Solaris Ethernet interface:
 

```
# ifconfig hme0 172.31.45.229/20 (Solaris 10 example)
or
# ifconfig hme0 172.31.45.229 netmask 255.255.255.0 (Solaris 2.6 example)
```

 Make permanent by editing **/etc/hosts** and set the new address for the systems hostname. Then edit **/etc/netmask** and add the netmask for the subnet-network.
3. Add default route on Solaris:
 

```
# route add default 172.31.44.94
```

 Make permanent by editing **/etc/defaultrouter** and add the address of the gateway.
4. Add DNS server to Solaris
  - a. Edit **/etc/resolv.conf** and add a nameserver line for the DNS server.
  - b. Make sure, DNS is used for hostname translation: # `cp /etc/nsswitch.dns /etc/nsswitch.conf` or edit **nsswitch.conf** to use files **dns** for the hostname resolution.

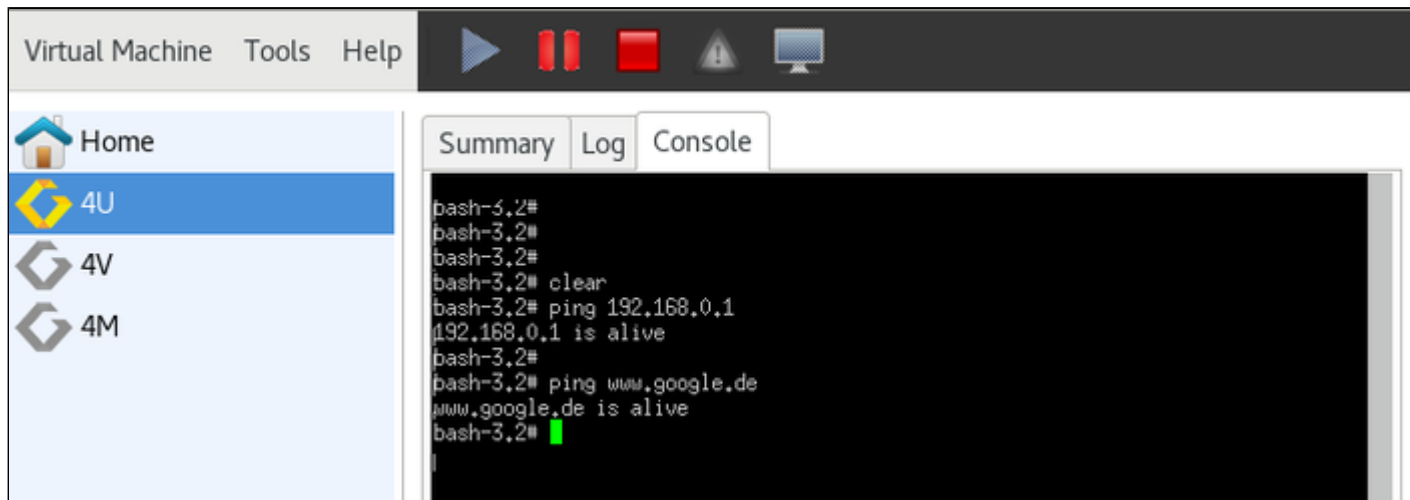
## Remote Linux Configuration

In addition to setting up the SSH VPN tunnel as described in [SSH VPN - Connecting Charon Host and Guest to Customer Network](#), you must set the route to the VPC subnet on the remote Linux system in order to be able to communicate with the client system there.

- Example: # `ip route add 172.31.32.0/20 via 192.168.0.10 dev br_vpn0`

## Result

The Solaris guest can reach the customer network via VPN and the Internet via NAT.



# Data Transfer Options

## Contents

- SFTP File Transfer
  - SFTP to/from Charon Host
  - SFTP to/from Solaris Guest
- Data Migration from Physical to Emulated System
  - Direct Data Transfer over the Network
  - Using UFSdump Backup Archives as VTape Container Files
  - Other Data Transfer Options

## SFTP File Transfer

### SFTP to/from Charon Host

SFTP enables file transfers to and from the Charon-SSP AWS instance. The user for file transfers is the **charon** user. The security group must allow SSH access to allow SFTP access to the Charon-SSP AWS instance.

To connect to the instance as the user **charon**, use the following command:

```
$ sftp -i <path-to-your-private-key> charon@<AWS-public-IP-address>
```

Below you see sample output of a connection:

```
$ sftp -i .ssh/mykey.pem charon@<public-ip-address>
Connected to storage@3.81.64.139.
sftp> ls
media                ssp-snapshot
```

This method can be used, for example,

- to copy ISO files to the the Charon AWS instance,
- to copy vdisk and vtape files to the Charon AWS instance,
- to copy backups taken of emulated SPARC systems from the Charon AWS instance.

### SFTP to/from Solaris Guest

Once the Solaris guest is reachable from the host system or from the customer network, SFTP can also be used to transfer data to/from the Solaris guest system.

#### SFTP availability on Solaris:

SFTP is part of the SSH software on Solaris. To use SSH/SFTP on older versions of Solaris (e.g. Solaris 2.6), this software must be obtained from a provider of public domain Solaris packages, such as [unixpackages.com](http://unixpackages.com). They often require a small fee. On more modern Solaris versions (e.g. Solaris 10), packages are available on the Solaris installation media that provide this functionality.

## Data Migration from Physical to Emulated System

When migrating a physical system to an emulated system, all the data of the physical system must be transferred to the emulated system. The sections below provide some hints about how this could be performed. **However, the section does not describe a recommended migration path.** Migrating a system depends very much on the specific customer environment and the best path has to be defined on a case-by-case basis.

**i** For consulting services supporting the migration (subject to a charge), please contact your Stromasys representative or VAR. You can find the contact information on the Stromasys [web page](#).

### Direct Data Transfer over the Network

A direct data transfer between the Solaris system on the real hardware and the Solaris system on the emulated hardware is often the easiest way to migrate data from a physical system to an emulated system.

Once the emulated system can be reached from the customer network (see [SSH VPN - Connecting Charon Host and Guest to Customer Network](#)), technically this is also possible when the emulated system is in the AWS cloud. **However, whether it is a feasible solution depends on the network throughput and stability across the VPN.**

### Using UFSdump Backup Archives as VTAPE Container Files

To transfer backup archives to the Charon host system, perform the following steps:

- Create ufsdump archives of the filesystems on the original system that are to be migrated.  
(# `ufsdump 0f <archive-filename> <disk-partition>`)
- Use SFTP to copy the archive files to the Charon host system across a VPN connection or directly to the public IP address of the host system.
- Use the Charon-SSP Manager File Manager (**Tools > AWS Cloud > File Manager**) to rename the files to `<archive-name>.vtape`.
- Add the files to the Charon-SSP virtual machine configuration of the emulated system as virtual tape drives.
- Use `ufsrestore` on Solaris to restore the files to the correct filesystems.

### Other Data 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.

For large data transfers, Amazon offers a special service, [AWS Snowball](#). This service may be helpful if large amounts of data need to be transferred from the physical system to the Charon-SSP AWS instance. This service is independent of the Stromasys offering.

## Firewall and AWS Security Group Considerations

This section provides an overview of the firewall and/or AWS security group requirements when running Charon-SSP.

**!** The ports used by Charon-SSP can be different depending on the applications running on the host system and on the guest Solaris system. They will also depend on the configured Charon-SSP features. The information in this section is informational only and can never be totally complete.

**Please note:** if an SSH VPN tunnel is created to access the Charon-SSP host and guest systems, only the SSH port must be accessible. All other applications can run through the encrypted tunnel.

The following table provides an overview of the most frequently used network ports in a Charon-SSP installation. They must be taken into account when configuring firewalls and AWS security groups allowing access to the Charon-SSP installation.

Component	Port(s)	Purpose	Applicable to AWS version
<b>SSH, SFTP, SSH tunneling</b>	22 (TCP)	SSH access; required for <ul style="list-style-type: none"> <li>access to the Charon-SSP host command-line,</li> <li>connecting to the Charon-SSP host using the Charon Manager's built-in SSH feature,</li> <li>SFTP file transfer, and</li> <li>SSH VPN tunnels.</li> </ul>	Y
<b>Charon-SSP Agent</b>	9091 (TCP and UDP)	Communication with Charon-SSP Manager and Charon-SSP Director	Y
	9101 (UDP)	Communication with Charon-SSP Director	Y
<b>Graphics emulation</b>	default: 11001 (TCP)	Mouse event data (port must be unique on host system)	Y
	default: 11000 (TCP)	Keyboard event data (port must be unique on host system)	Y
	default: 11100 (TCP), 11101 (TCP)	Remote screen emulation for single (one port) or dual (two ports) screen (default ports can be changed; must be unique on host system)	Y
<b>Telnet or TCP raw mode serial ports/serial console</b>	default: 9000 (TCP)	Port to access emulated serial console or other emulated serial port via TCP. Port must be unique for each emulated port on host system.	Y
<b>Xephyr X-server</b>	6001-6100 (TCP); port specified in X11 server configuration	Determines the X DISPLAY number. For example: 6100 indicates DISPLAY :100. Must be unique on host system.	Y
	7100 (TCP)	Font-server port	Y
	177 (TCP and UDP)	XDMCP server	Y
<b>NFS server</b>	111 (TCP and UDP)	RPC portmapper	
	ports assigned by portmapper	use <code># rpcinfo -p</code> to determine ports used (conventional product only)	
	static port assignments	For example: setting <code>RPCMOUNTDOPTS="-p port"</code> in <code>/etc/sysconfig/nfs</code> will add "-p port" to the <code>rpc.mount</code> command (conventional product only).	
<b>VNC server on host system</b>	5901-5910 (TCP)	Actual port depends on VNC server configuration. Allow a remote client to access the VNC server on the host system.	
<b>License manager, license server</b>	1947 (TCP and UDP)	Access to web-based Sentinel ACC GUI, identification of remote network licenses served by license servers, using remote network licenses.	
	8080 (TCP)	Access to cloud license server.	Y
<b>License client</b>	30000 to 65535 (UDP)	Incoming answers from license servers if broadcast search is used.	
<b>PulseAudio server</b>	4713 (TCP)	Emulated audio device	Y
<b>iSCSI target</b>	3260 (TCP and UDP)	Required for the initiator to access the target.	

# Upgrading Charon-SSP AWS

## Contents

- Software Package Upgrade
  - Charon-SSP AWS Installation Packages
  - Upgrading Charon Packages on the Charon-SSP AWS Instance
  - Upgrading Charon Manager on the Management Linux System
    - Standard Linux Package Update
    - Automatic Update upon First Connection to Updated Charon Host System
- Creating a New Instance with New Charon-SSP AWS AMI Version

## Software Package Upgrade

If required, Stomasys or your VAR will provide you with installation packages to update your Charon-SSP AWS instance.


In this case, perform the following steps:

1. Use **SFTP** to copy the RPM packages to your Charon AWS instance (see [Data Transfer Options](#)).
2. Connect to the host system using Charon Manager (see [Accessing the Charon-SSP AWS Instance](#))
3. Shutdown the guest systems running on the host and stop the emulator.
4. Use SSH to connect to your Charon AWS instance (see [Accessing the Charon-SSP AWS Instance](#)).
5. Install the new RPM packages **as described below**.
6. Install the new **Charon-SSP Manager** package on your local Linux system **as described below**.
7. Connect to the host system using Charon Manager.
8. Start the emulator and boot the guest systems.

## Charon-SSP AWS Installation Packages

The Charon-SSP packages are provided as RPM packages and (only Charon Manager) as Debian packages. You can download the required package(s) from Stomasys or receive them on a read-only medium, such as a CD-ROM. If you do not have the software package(s), please contact either Stomasys or your Value-Added Reseller (VAR) for further help.

The following table shows installation package names. The update of these packages is shown in the following sections.

Product part	Package names
Charon-SSP emulator software for the different SPARC architectures. 4U+ and 4V+ contain both the 4U and 4U+ and 4V and 4V+ versions respectively.	charon-ssp-4v+-<version>.aws-1.x86_64.rpm charon-ssp-4m-<version>.aws-1.x86_64.rpm charon-ssp-4u+-<version>.aws-1.x86_64.rpm
Charon-SSP Agent (required for the Charon Manager connection)	charon-agent-ssp-<version>.x86_64.rpm
Charon-SSP Manager	charon-manager-ssp-<version>.rpm
 This package must be installed on the local Linux system!	charon-manager-ssp-<version>.deb

## Upgrading Charon Packages on the Charon-SSP AWS Instance

After copying the packages to the Charon-SSP AWS instance, install the packages as described below. For details, please refer to the relevant man-pages on Linux.

Step	Details
1	Cleanly shut down any running Solaris guest system and stop the emulator.
2	Log in to the Charon-SSP AWS via SSH to the <b>sshuser</b> and become the <b>root</b> user ( <code>sudo -i</code> ).
3	Go to the directory where the packages have been stored: <code># cd &lt;package-location&gt;</code>
4	<b>Run upgrade command:</b> For systems with RPM package management (Red Hat, CentOS): <code># yum update &lt;package-name&gt;</code>

## Upgrading Charon Manager on the Management Linux System

There are two ways to upgrade the Charon Manager on the management Linux system.

### Standard Linux Package Update

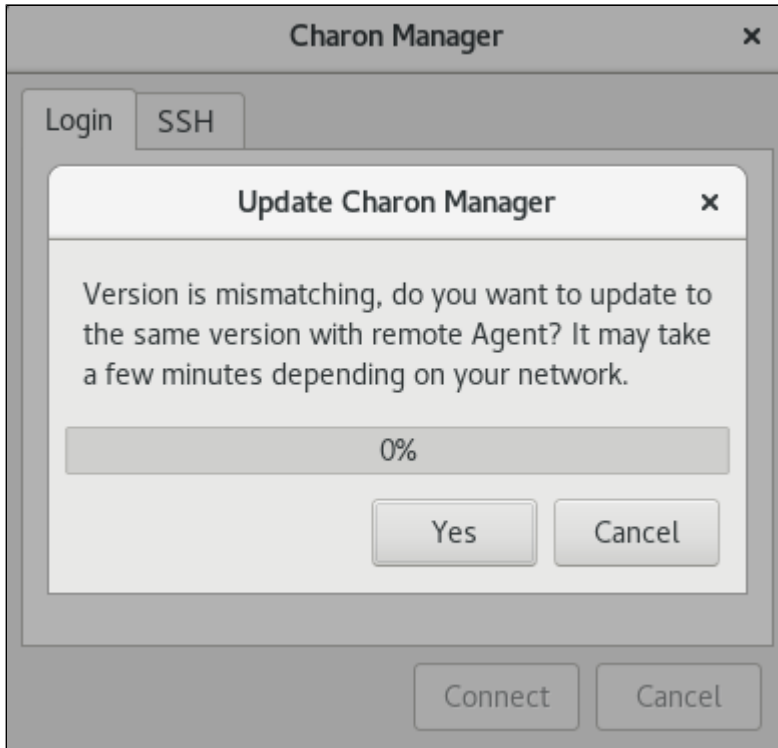
Step	Details
1	Log in as the <b>root</b> user.
2	Go to the directory where the packages have been stored: <code># cd &lt;package-location&gt;</code>
3	<b>Run upgrade command:</b> For systems with RPM package management (Red Hat, CentOS): <code># yum update &lt;package-name&gt;</code> For systems with Debian package management (Debian, Ubuntu): <code># dpkg -i &lt;package-name&gt;</code>

## Automatic Update upon First Connection to Updated Charon Host System

**i** New feature in Charon-SSP 4.0.x and its pre-release versions.

If you connect to a Charon host system from a management system that uses an older version of the Charon Manager than the one used on the host system, you will be offered an automatic update to the version of the Charon host system. If the target host system is a newly installed Charon system, you will be prompted to set the management password before you get to the upgrade window (see [Connecting with the Charon-SSP Manager](#) for more information).

The following image shows a sample of the upgrade offer:



Confirm the upgrade by clicking on **Yes**. This will initiate the download and installation of the new version. If you work as a non-privileged user, you will be prompted for your password. To decline the automatic update, click on **Cancel**.

**i** This mechanism also works for downgrades (i.e., if the Charon Manager version is newer than the version of the Charon host system it connects to, it can be automatically downgraded to the version of the target system).

**i** Some future major changes in the Charon Manager and/or Agent may prevent the automatic upgrade. In such cases, follow the manual package installation steps.

## Creating a New Instance with New Charon-SSP AWS AMI Version

One option to upgrade Charon-SSP AWS to a newer version is the creation of a new instance based on a new version of the Charon-SSP AWS AMI.

This may be applicable if

- all important Charon instance files (e.g., vdisks and ISO images) are on a separate EBS storage volume that can easily be moved to a new instance,
- the overall configuration of the Charon-SSP 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-SSP 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.
- Backup emulated SPARC disks and configuration data on the old instance to the separate EBS volume.
- If applicable, backup important system configuration files to the separate EBS volume.
- Stop the old instance.
- Copy the Charon Manager kit from the new instance and install it on your local Linux system, or connect to the new instance via Charon Manager to initiate the automatic Charon Manager update.
- Move the EBS volume(s) and (if applicable) network interfaces with elastic IP addresses to the new instance.
- Import the virtual SPARC 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 the AWS documentation for AWS specific details.**



# Charon-SSP Software Deinstallation

## Contents

- Deinstalling the Charon Manager
- Terminating the Charon-SSP AWS instance

## Deinstalling the Charon Manager


Perform the following steps to deinstall the Charon Manager from your Linux management system:

Step	Details
1	Log in as the <b>root</b> user.
2	<p><b>Run the deinstallation command:</b></p> <p>For systems with RPM package management (Red Hat, CentOS):  <code># yum erase &lt;package-name&gt;</code></p> <p>For systems with Debian package management (Debian, Ubuntu):  <code># dpkg -r &lt;package-name&gt;</code></p>

## Terminating the Charon-SSP AWS instance

To permanently remove your Charon-SSP AWS instance, select your instance from the instance list and select **Actions > Instance state > Terminate**.

This will stop the instance and remove it. Unless your data (configuration files, vdisk containers, etc.) was stored on a separate EBS volume, it will also be removed.

 Make sure you backup any data you wish to retain before terminating an instance.

# OpenBoot Console

## Contents

- OpenBoot Console Overview
- OpenBoot Console Command Reference
  - banner
  - boot
  - devalias
  - help
  - history
  - nvalias
  - nvunalias
  - printenv
  - probe-scsi
  - quit or poweroff
  - reset
  - setenv
  - show-devs

## OpenBoot Console Overview

The Charon-SSP SPARC virtual machines use a subset of the Sun OpenBoot console found on native Sun workstations and servers. The figure below shows the initial console screen at boot on a virtual SPARCstation 20.

```

SMCC SPARCstation 20 Emulator by Stromasys

CPU_#0      TI, TMS390Z50(3.x)      0Mb External cache

CPU_#1      ***** NOT installed *****
CPU_#1      ***** NOT installed *****
CPU_#1      ***** NOT installed *****

>>>> Power On Self Test (POST) is running .... <<<<<

SPARCstation 20 (1 X 390Z50), No Keyboard
Emulate OBP Rev. 2.25, 64 MB memory installed, Serial #12648430.
Ethernet address 2:c:29:4a:d3:29, Host ID: 72c0ffee.

Type help for more information

Can not load boot block!
ok

```

## OpenBoot Console Command Reference

The following sections describe the currently supported console commands.

### banner

Display power-on banner.

#### Syntax

```
banner
```

#### Description

Use this command to display the power-on banner.

#### Example

The following example demonstrates the output of the banner command on Charon-SSP configured as a SPARCstation 20.

```
ok banner

SPARCstation 20 (1 X 390Z50), No Keyboard
Emulate OBP Rev. 2.25, 64 MB memory installed, Serial #12648430.
Ethernet address 2:c:29:4a:d3:29, Host ID: 72c0ffee.
```

### boot

Load operating system.

#### Syntax

```
boot [ device-alias ] [ boot-args ]
```

#### Description

This command boots the specified *device-alias* passing any optional *boot-args* to the kernel. The *boot-args* must be recognized as valid by the Solaris kernel used. Booting from a ZFS disk is supported starting with Charon-SSP version 1.4.1 if the Solaris version supports this feature.

Starting from Charon-SSP version 2.0.5 there is a special boot argument when booting from CD-ROM:

```
boot cdrom -slow=<sec>
```

This parameter should only be used if there are problems when booting from ISO files resulting in a BAD TRAP error. This seems to happen quite frequently with Solaris 7 installation ISOs.

For more information about device aliases, see the **devalias** command.

#### Example

The following example demonstrates the output of the boot command on Charon-SSP configured as a SPARCstation 20 and booting SunOS 4.1.4 from CD-ROM.

```
ok boot cdrom

Boot device: /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@6,0:d   File and args: -v
Boot Release 4.1.4 (sun4m) #2: Fri Oct 14 11:07:52 PDT 1994
Copyright (c) 1983-1990, Sun Microsystems, Inc.
Boot: Romvec version 3.
root on /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@6,0:d fstype 4.2
Boot: vmunix
.Size: 868352.....
.....+2319136+75288 bytes
Statistics:
SuperSPARC: PAC ENABLED
SunOS Release 4.1.4 (MUNIX) #2: Fri Oct 14 11:09:07 PDT 1994
Copyright (c) 1983-1993, Sun Microsystems, Inc.
```

## devalias

Display device aliases.

### Syntax

```
devalias
```

### Description

These commands display the current device aliases. This shows the link between the aliases, such as `cdrom` and the devices shown in the device tree, listed by `show-devs`.

### Example

The following example demonstrates the output of the `devalias` command.

```
ok devalias
ttyb          /obio/zs@0,100000:b
ttya          /obio/zs@0,100000:a
keyboard!     /obio/zs@0,0:forcemode
keyboard      /obio/zs@0,0
floppy        /obio/SUNW,fdtwo
scsi          /iommu/sbus/espdma@f,400000/esp@f,800000
net-aiui      /iommu/sbus/ledma@f,400010:aiui/le@f,c00000
net-tpe       /iommu/sbus/ledma@f,400010:tpe/le@f,c00000
net           /iommu/sbus/ledma@f,400010/le@f,c00000
disk          /iommu/sbus/espdma@f,400000/esp@f,800000/sd@3,0
cdrom         /iommu/sbus/espdma@f,400000/esp@f,800000/sd@6,0:d
tape          /iommu/sbus/espdma@f,400000/esp@f,800000/st@4,0
tape1         /iommu/sbus/espdma@f,400000/esp@f,800000/st@5,0
tape0         /iommu/sbus/espdma@f,400000/esp@f,800000/st@4,0
disk3         /iommu/sbus/espdma@f,400000/esp@f,800000/sd@3,0
disk2         /iommu/sbus/espdma@f,400000/esp@f,800000/sd@2,0
disk1         /iommu/sbus/espdma@f,400000/esp@f,800000/sd@1,0
disk0         /iommu/sbus/espdma@f,400000/esp@f,800000/sd@0,0
```

## help

Display OpenBoot console help.

### Syntax

```
help [ command ]
```

### Description

Use this command to display the list of commands supported by the OpenBoot console. For brief help on individual commands specify the **command** parameter.

### Example

```
ok help
Following commands are supported by this version:

boot devalias nvalias nvunalias
printenv setenv probe-scsi show-devs
reset banner history help

Enter 'help command-name' for more help

Examples: help setenv
```

## history

Display console command history.

### Syntax

```
history
```

### Description

This command displays a list of all commands previously entered at the OpenBoot Console.

### Example

The following example demonstrates the output of the history command.

```
ok history
 1 printenv
 2 help
 3 help devalias
 4 help history
 5 help probe-scsi
 6 probe-scsi
 7 show-devs
 8 banner
```

## nvalias

Stores devalias values in nvramrc.

### Syntax

```
nvalias <alias> <device-path>
```

### Description

Stores the device aliases in in NVRAMRC. The alias persists until the **nvunalias** or **set-defaults** command is executed.

### Example

The following example demonstrates the use of the **nvalias** command to create and store a device alias named *disk3* that represents a SCSI disk with a target ID of 3 on a SPARCstation 10 system

```
ok nvalias disk3 /pci@1f,0/pci@1,1/ide@3/disk@3,0
```

## nvunalias

Removes a device alias from NVRAMRC.

### Syntax

```
nvunalias <alias>
```

### Description

Deletes the corresponding alias from NVRAMRC.

## printenv

Display environment variables.

### Syntax

```
printenv
```

### Description

Use this command to print the current and default values of OpenBoot console variables.

**Example**

The following examples illustrate the output of the **printenv** command on Charon-SSP/4U.

```
ok printenv
Variable Name Value Default Value
auto-boot? false true
local-mac-address? true true
output-device ttya screen
input-device ttya keyboard
boot-file -v
boot-device /pci@1f,4000/scsi@3/disk@1,0:a disk net
ttya-mode 9600,8,n,1,- 9600,8,n,1,-
ttyb-mode 9600,8,n,1,- 9600,8,n,1,-
diag-file -v
diag-device net disk net
diag-switch? true true
```

## probe-scsi

Scan SCSI bus for attached devices.

**Syntax**

```
probe-scsi
```

**Description**

This command scans the SCSI bus to locate attached devices.

**Example**

The following example demonstrates the output of the **probe-scsi** command on system with a single virtual CD-ROM.

```
ok probe-scsi
Target 0
Unit 0 Disk virtual Scsidrom (c)SRI0200
```

## quit or poweroff

Turn off virtual machine.

**Syntax**

```
quit | poweroff
```

**Description**

Use this command to shut down the virtual machine.

## reset

Restart the system.

### Syntax

```
reset
```

### Description

This command restarts the SPARC virtual machine.

## setenv

Set console environment variables.

### Syntax

```
setenv variable value
```

```
setenv variable --
```

### Description

This command sets a console configuration variable to a specific value. The current and default values of the variables are shown by the **printenv** command. To restore a variable to its default value, specify '--' in place of the value. For a complete list of possible variable names and their descriptions, see the list below.

- **auto-boot?** - If true, boots automatically after power on or reset.
- **local-mac-address?** - If true, the MAC address of the network card is used instead of the system MAC address.
- **output-device** - Output device used at power-on
- **input-device** - Input device used at power-on.
- **boot-device** - Space delimited list of devices to define boot attempt sequence.
- **boot-file** - A string of arguments to be passed to the boot loader (e.g. -a or -v).
- **ttya-mode** - Serial line configuration for ttya
- **ttyb-mode** - Serial line configuration for ttyb
- **diag-file** - Diagnostic mode boot arguments.
- **diag-device** - Diagnostic startup source device.
- **diag-switch?** - Indicates if system should run in diagnostics mode.

**i** Changes to environment variables are stored in NVRAM and are permanent. However, they only take effect after executing the **reset** command.

### Example

The following example illustrates the use of the **setenv** command.

```
ok setenv auto-boot? true
auto-boot? = true
```



## show-devs

Display device tree.

### Syntax

```
show-devs
```

### Description

This command displays the tree of devices visible from the console.

### Example

The following example demonstrates the output of the **show-devs** command.

```
ok show-devs

/TI,TMS390Z50@f,f8ffffffc
/SUNW,sx@f,80000000
/eccmemctl@f,0
/virtual-memory@0,0
/memory@0,0
/obio
/iommu@f,e0000000
/openprom
/aliases
/options
/packages
/obio/power@0,a01000
/obio/auxio@0,800000
/obio/SUNW,fdt@0,700000
/obio/interrupt@0,400000
/obio/counter@0,300000
/obio/eeprom@0,200000
/obio/zs@0,0
/obio/zs@0,100000
/iommu@f,e0000000/sbus@f,e0001000
/iommu@f,e0000000/sbus@f,e0001000/SUNW,bpp@f,4800000
/iommu@f,e0000000/sbus@f,e0001000/ledma@f,400010
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000

/iommu@f,e0000000/sbus@f,e0001000/ledma@f,400010/le@f,c00000
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/st
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd
/packages/obp-tftp
/packages/deblocker
/packages/disk-label
```