Charon-SSP Version 5 for OCI Cloud Getting Started Guide

Contents

About this Guide

Contents

Intended Audience

This Getting Started guide is intended for anyone who needs to install, configure, or manage the Stromasys Charon-SSP processor/platform emulation software in the OCI cloud. Its main focus is on installations that use the prepackaged Charon-SSP images available on the OCI Cloud marketplace. However, it may also be helpful when creating a Linux server in the cloud for a conventional RPM installation of Charon-SSP products. A general working knowledge of PC operating systems and their conventions is expected.

This guide describes the **cloud-specific aspects** of Charon-SSP for OCI cloud. It is supplemented by the general *Charon-SSP User's Guide* for your Charon-SSP product version (see CHARON-SSP for Linux) and the *VE License Server User's Guide* in Licensing Documentation.

For additional information about this product, please contact Stromasys at the regional offices listed below in *Obtaining Technical Assistance or General Product Information*, send an email to **Team.Support.OCI@Stromasys.com**, or contact your Stromasys VAR.

Product Overview

Stromasys provides Charon-SSP for on-premises installations and cloud environments. For both environments, there are several different options. **The availability of cloud-specific options may differ depending on the cloud environment**. Please contact your Stromasys representative for availability details. The **typical cloud-specific options** are:

1. **Prepackaged** images provided on the cloud-specific marketplace:

   a. Cloud-specific Charon-SSP AL (Automatic Licensing) image using a public, Stromasys-operated, cloud-specific license server for metered licensing (license created automatically at first instance launch). Starting with Charon-SSP version 5.3.8, Charon-SSP AL also supports AutoVE where the instance can be configured at first launch to automatically connect to a customer-operated license server running in AutoVE mode. The minimum VE license server version for AutoVE mode is 1.1.21.

   b. Specific to AWS: the original Charon-SSP AL marketplace image was replaced by an AWS service providing the same features as the original AL image. However, the underlying Linux host system is Amazon Linux 2023, and licensing is based on public AutoVE servers, i.e., different public servers (operated by Stromasys) are used than in the original AL image. Please contact your Stromasys representative should you require a private, customer-operated AutoVE license server.

   c. Cloud-specific Charon-SSP VE (Virtual Environment) image using a customer-operated, private VE license server in the cloud (license must be obtained from Stromasys).

2. Installation of **Charon-SSP for Virtual Environments (VE)** on a Linux server in the cloud using **RPM packages** provided by Stromasys, and utilizing a customer-operated, private VE license server in the cloud (license must be obtained from Stromasys).

Overview of the relevant documentation for Charon-SSP for cloud environments:

- This **Getting Started Guide** covers basic **cloud-specific aspects** when installing a Charon-SSP product in the cloud. The main focus is on the prepackaged images provided on the cloud-specific marketplaces. However, it can also serve as an introduction to general cloud-specific aspects when installing the individual Charon-SSP RPM packages on a server in the cloud.
- The general *Charon-SSP User's Guides* (see CHARON-SSP for Linux) cover **features, configuration, and management of the Charon-SSP products**.

- The *VE License Server User's Guide* in the Licensing Documentation section of the Stromasys Product Documentation site covers features, installation, and management of the **VE (Virtual Environment) license server** and the **VE licenses**.
- The **Release Notes** of your product provide important information regarding known problems and possible workarounds.

Charon-SSP overall product overview:

The following table provides an overview of **Charon-SSP packaging in cloud and non-cloud environments**, the associated licensing, and the applicable product documentation:

Basic Packaging	Packaging Variants		Installation Location 5)	Licensing	Documentation
Prepackaged Versions including underlying host OS and Charon-SSP software	Cloud marketplace image	Automatic Licensing (AL) AWS SSP Amazon Linux (AL)	Cloud only	Public, Stromasys-operated license server 1) or Customer-operated VE license server in AutoVE mode	• Cloud-specific Getting Started Guide • General Charon-SSP User's Guide • VE License Server Guide (for AutoVE)
		Virtual Environment (VE)		Private, customer-operated VE license server 2)	• Cloud-specific Getting Started Guide • General Charon-SSP User's Guide • VE License Server Guide
	Baremetal ISO		On-premises (physical HW or VM)	HASP license (local or via license server)	• General Charon-SSP User's Guide
Individual RPMs	VE-enabled emulator		Cloud or on-premises (physical HW or VMware VM) 3)	Private, customer-operated VE license server	• General Charon-SSP User's Guide • VE License Server Guide
	Non VE-enabled emulator		On-premises (physical HW or VM)	HASP license (local or via license server)	• General Charon-SSP User's Guide
	Marketplace RPM packages 4)		Cloud only	Public, Stromasys-operated license server 1) or Customer-operated VE license server in AutoVE mode	• Cloud-specific Getting Started Guide • General Charon-SSP User's Guide • VE License Server Guide (for AutoVE)

1) Use of public license server requires Internet access of Charon host.

2) Typically the VE license server is installed in the same cloud environment - either on the Charon host itself or a separate license server system.

3) For the VMware case, there are some restrictions:

- The Charon emulator must be on the same VM as the VE license server, **or**
- on a VM running on the same ESXi host as the VE license server (ESXi binding), **or**
- on a VM running on an ESXi host managed by the same vCenter Server as the ESXi host on which the VE license server VM runs (vCenter binding).

4) Only for updating the Charon-SSP emulator software on hosts based on Charon-SSP cloud-specific marketplace images for Automatic Licensing (AL).

5) Non VE-enabled packages can also be installed in cloud environments. However, this is not recommended as HASP licenses are not suitable for clouds.

Please contact your Stromasys representative for availability details of cloud-specific Charon-SSP options in the OCI cloud environment.

Document Structure

- Introduction to Charon-SSP: overview of emulator concepts.
- Charon-SSP Cloud - Virtual HW and Guest OS Support: overview of the emulated hardware and guest operating systems supported by the virtual SPARC systems provided by the emulator.
- Setting up an OCI Instance for Charon-SSP: basic steps to create and launch a cloud instance to be used as a Charon-SSP host system.
- Installing the Charon Manager: steps to install the main management tool for the cloud-based Charon-SSP host instance.
- Installing the VE License Server Software: steps to install the VE license server package if VE licenses are to be used.
- Accessing the Charon-SSP Cloud Instance: explains how to use SSH, SFTP, and the Charon-SSP Manager to access the cloud-based Charon host instance for management and file transfer, and how to set the initial management password.
- Additional Charon-SSP OCI Instance Configuration: adding additional storage and network interfaces; cloud-specific networking aspects.

Please note:

- Cloud providers may change their management GUI without prior warning. Hence, the screenshots in this document may not always reflect the latest GUI appearance of the cloud provider. However, they will still provide an illustration of the described configuration steps.
- In general, the sample outputs in this document may show different versions than the one documented in this manual, but they are still representative of what a user will see.

Obtaining Documentation

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

Obtaining Technical Assistance or General Product Information

Obtaining Technical Assistance

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

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

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

Obtaining General Product Information

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

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

Region	Phone	Address
Australasia-Pacific	+852 3520 1030	Room 1113, 11/F, Leighton Centre 77 Leighton Road, Causeway Bay, Hong Kong, China
Americas	+1 919 239 8450	Stromasys LLC 871 Marlborough Ave, suite 100, Riverside CA 92507 USA
Europe, Middle-East and Africa	+41 22 794 1070	Avenue Louis-Casai 84 2nd Floor 1216 Cointrin Switzerland

Conventions

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

Definitions

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

Related Documents

- Charon-SSP User's Guides and Release Notes
- *VE License Server User's Guide* in Licensing Documentation

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 SPARC emulator 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 (represented by the 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 (represented by the 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 (represented by the SPARC T2 and T4)**: this variation added hypervisor processor virtualization to the Sun-4u; introduced in the Ultra SPARC 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.

Please note: For up-to-date information about supported features and guest OS versions refer to the section *Virtual Hardware and Guest OS Supported by Charon-SSP*. Unless otherwise mentioned, the terms Charon-SSP/4U and Charon-SSP/4V also include Charon-SSP/4U+ and Charon-SSP/4V+.

The image below shows the basic concept of migrating physical hardware to an emulator:
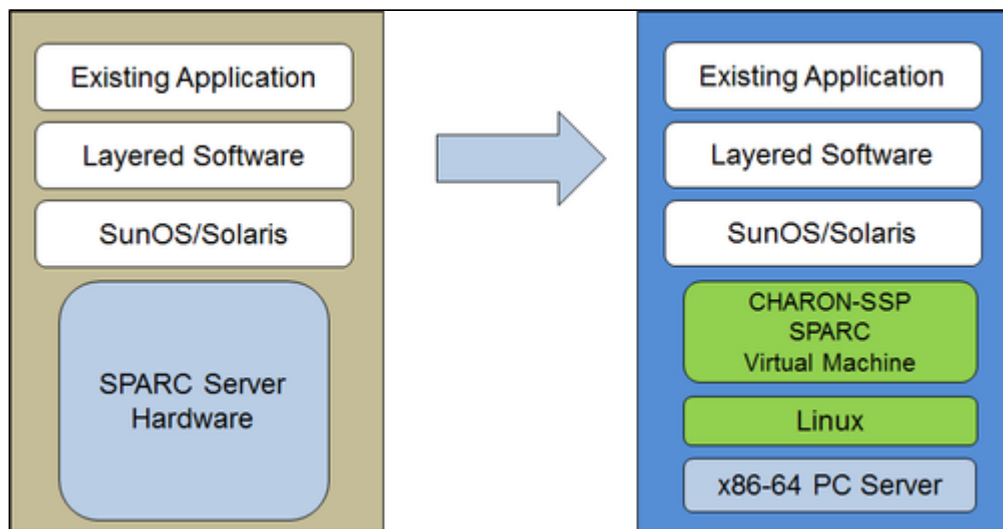
The Charon-SSP virtual machines allow users of Sun and Oracle SPARC-based computers to replace their native hardware in a way that requires little or no change to the original system configuration. This means you can continue to run your applications and data without the need to switch or port to another 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 VTx/EPT and AMD's AMD-v/NPT hardware assisted virtualization technology in modern CPUs to offer better virtual CPU performance. Charon-SSP/4U+ and Charon-SSP/4V+ require CPUs with VT-x/EPT or AMD-v/NPT support and **must** be installed on a dedicated host system. Running these product variants in a VM (e.g., on VMware) is **not supported**.

Please note: if you plan to run Charon-SSP/4U+ or 4V+ in a cloud environment, please contact Stromasys or a Stromasys VAR to discuss your requirements.

Charon-SSP Cloud - Virtual HW and Guest OS Support

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 supported virtual hardware in cloud-specific products		
	Charon-SSP/4M	**Charon-SSP/4U(+)** [1]	**Charon-SSP/4V(+)** [1]
SPARC V8 (32-bit)	Y		
SPARC V9 (64-bit)		Y [2]	Y [4]
Max. number of CPUs	4	24	64 / 128 [9]
Max. RAM	64MB to 512MB	1GB to 128GB	1GB to 1024GB [5]
Ethernet controllers	2 controller type le	19 controller types hme, qfe, bge [8]	8 controller types bge and qfe
SCSI controllers	1	2	2
SCSI target IDs	7 [3]	30 [3]	30 [3]
VDS storage devices			120 [7]
Serial ports	2	2	2 + Vconsole
Graphics controllers	1 (CGTHREE or CGSIX [6])	1 (CGSIX or RAGE XL)	
Audio controllers	1 (DBRIe)	1 (DBRIe)	

[1] Charon-SSP/4U+ has the same virtual hardware specification as Charon-SSP/4U, Charon-SSP/4V+ the same as Charon-SSP/4V. 4U+ and 4V+ are only supported on physical (baremetal) hardware with Intel VTx/EPT or AMD-v/NPT support running a Linux kernel supported by Stromasys. Please refer to the Charon-SSP User's guide for more details.

[2] SPARC V9 is backward compatible. Hence, Charon-SSP/4U can also support V8 32-bit systems.

[3] 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 and driver versions, and the SCSI devices used.

[4] Charon-SSP/4V supports one LDom per instance. An LDom virtual disk image can be booted by Charon-SSP directly (system device configuration may have to be adjusted for differences in emulated hardware).

[5] Before Charon-SSP version 5.6.1, the actual maximum values were different depending on guest OS: Solaris 10: 1TB, Solaris 11: 512GB.

[6] CGSIX emulation is not supported for SunOS 4.x guest systems.

[7] Guest systems starting with Solaris 10-U6 can use VDS and/or SCSI storage devices.

[8] The BGE controller for Charon-SSP/4U is supported starting with versions 5.2.5 and 5.3.4.

[9] Support for 128 emulated CPUs will be added to Charon-SSP/4V in version 5.5.4. Earlier versions of Charon-SSP/4V support 64 emulated CPUs.

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 11.4

For Charon-SSP/4V(+), please note:

- For the emulated SPARC T4, supported Solaris 10 versions are:
  Oracle Solaris 10 1/13, Oracle Solaris 10 8/11, and Solaris 10 9/10, or Solaris 10 10/09 with Oracle Solaris 10 8/11 patch set.
- The emulated SPARC T4 model is a prerequisite for running Solaris 11.4 in the emulator.
- Solaris kernel zones are not supported.

Setting up an OCI Instance for Charon-SSP

This chapter describes how to set up a basic cloud instance for Charon-SSP.

Contents

Charon Licensing for Charon Emulators in the OCI Cloud

Contents

General Information

This chapter describes the basic licensing options available for Charon-SSP instances in cloud environments. Please note that even though other solutions may be possible the ones listed here represent the recommended solutions for Charon-SSP cloud-based solutions.

Charon-SSP requires a license to run emulated SPARC systems. For a typical cloud-based installation, there are different Charon-SSP product variants with different licensing models. As **availability may differ depending on cloud environment**, please contact your Stromasys representative for any questions about product availability and licensing options.

1. The cloud-specific, prepackaged **Charon-SSP AL** (Automatic Licensing) marketplace image offers pay-as-you-go billing for Charon-SSP. Please contact Stromasys Sales should you be interested in this option. It utilizes either

- a **public, Stromasys-operated cloud-specific license server**, or
- a **private customer-operated VE license server** operating in **AutoVE** mode.

2. **Charon-SSP VE** (Virtual Environment) utilizing a **customer-operated, private VE license server** in a supported cloud environment. The license server, in this case, operates in **general VE mode.** Charon-SSP VE is available as a

- prepackaged marketplace image on some cloud platforms, and
- in RPM package format for a conventional installation.

These licensing options are briefly described below.

Please note: the user is responsible for any **Solaris** licensing obligations and has to provide the appropriate licenses.

Charon-SSP Automatic Licensing Overview

When a cloud instance is launched from a Charon-SSP AL marketplace image, this instance requires a license to run emulated SPARC systems. This license is created automatically upon first launch of the Charon-SSP AL instance.

Please note the following points:

- You will be billed by the cloud provider for your use of the Charon-SSP AL instance. Stromasys will not bill you directly. **Please contact Stromasys Sales should you be interested in this option**.
- Charon-SSP AL instances require Internet access (via their own public IP address or NAT) for the license mechanism to work. If NAT is used, the gateway must be an instance in the same cloud (i.e., the source IP address must be from the address range of the same cloud provider in which the Charon host instance runs). The public, Stromasys-operated license servers must be reachable on port 8080 (SSP before version 5.5.5) or port 8081 (SSP version 5.5.5 or higher). Also, a DNS service must be reachable to resolve the host names of the license servers, or corresponding entries in /etc /hosts must exist. The license server details are provided below.
- If you change the instance type after first launching the instance and thereby change the number of CPU cores (or if the number of CPU cores is changed by any other method), **the license will be invalidated**.
- Some licensing problems or other requirements (e.g., additional CPU cores needed) may make it necessary to move the emulator to a new instance. Therefore, it is strongly recommended to store all relevant emulator data on a separate disk volume that can easily be detached from the old instance and attached to a new instance.
- If you need to set up a new Charon host instance with AL licensing, you must create it via launching a new instance from the appropriate marketplace image and (as necessary) copying/moving the emulator data and configuration to the new instance. Cloning an instance with AL licensing will result in a system with an invalid license.
- 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).

At the time of writing, the public Stromasys-operated license servers in OCI were

- **cloud-oci1.stromasys.com** and
- **cloud-oci2.stromasys.com**
- TCP port 8080 for SSP version before 5.5.5 (old certificates)
- TCP port 8081 for SSP version 5.5.5 or higher (new certificates)

Charon-SSP VE Licensing Overview

Charon VE License Characteristics

The main characteristics of VE licenses are the following:

- Software licenses only.
- Installed on the Charon host or on a separate license server.
- Require the Charon VE license server software (the RPM package is included in the prepackaged, cloud-specific marketplace Charon-SSP VE image).
- Require matching Charon emulator software (preinstalled on the prepackaged, cloud-specific marketplace Charon-SSP VE image).
- Different modes of operation:
  - For **general VE mode**, the customer is billed by Stromasys depending on the number and type of the emulated systems allowed by the installed license(s). The license server software itself is free of charge.
  - **AutoVE mode** is an extension of automatic licensing and introduces metered billing (by the cloud-provider) for VE licenses in cloud environment. It defines how many Charon host instances can be run based on the respective license. The number of emulated systems on each host instance is limited by the host resources, not the license. The emulator host instance must be based on a suitable Charon automatic licensing marketplace image, and license server and emulator host must be in the same cloud. The new AWS marketplace image is based on this mode using public, Stromasys-operated license servers. If you intend to use private AutoVE license servers (no Internet access required), please contact your Stromasys representative.
  - The license server for both modes is managed by the customer.

If supported by the cloud provider, the VE license server instance can be moved to a different subnet, as long as the original instance can be moved. It is also possible to backup and restore (to the same instance) the license server data. However, the following actions will **invalidate the license**:

- Changing the number of CPU cores of the license server system.
- Copying the license server data to a different instance.
- Seriously damaging the root filesystem of the license server system.
- Re-installing the license server system.
- Copying the virtual machine on which the license server runs, cloning a virtual machine, or recovering a backup into a new virtual machine.

Please note: Charon-SSP 5.5.5 and VE license server 2.1.3 introduce new certificates and the option to use user-defined certificates. Please review the Virtual Environment (VE) License Server Documentation for details and to avoid possible compatibility problems between old and new versions.

Charon VE License Server Communication Requirements

For proper operation, the system on which the license server runs has the following communication requirements:

Communication with the cloud infrastructure:

For proper functionality, the OCI cloud instance on which the license server runs must be able to communicate with the cloud infrastructure:

- The metadata server of the cloud environment (169.254.169.254)

It must also be able to communicate with the client systems using the license. The following ports are used for this communication **by default**:

- **TCP/8083**: must be permitted from the client to the license server to enable the use of the license by the client.
- **TCP/8084**: must be permitted by the license server for any system that should access the web interface to display license information.

In current VE license server versions, the default ports can be changed in the file **/opt/license-server/config.ini**. Please refer to the VE license server guide for details.

Basic License Installation Steps Before an Emulator Can be Started

If there is no VE license server running already, decide on which cloud instance it should run and install the VE License Server package on the selected system. The VE License Server RPM package is included in the prepackaged Charon-SSP VE marketplace images. Alternatively, Stromasys will provide a download location. See *Installing the VE License Server Software*.

- If you don't already have a license, contact your Stromasys representative to procure an appropriate license.
- Log in on your Charon VE License Server instance.
- Create a C2V file and send it to the email address Stromasys will provide to you.
- Install the V2C file you will receive from Stromasys.
- Configure the emulator instance(s) to use the license server.

Please refer to the Licensing Documentation for more information.

Charon-SSP Cloud Instance Prerequisites

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

Please note:

- If you use a Charon-SSP marketplace image to launch your instance, all Linux host operating system requirements are fulfilled.
- If you use a generic Linux marketplace image to launch your instance, please refer to the Charon-SSP for Linux User's Guide for the software requirements that must be fulfilled.

General Notes

The minimum hardware requirements are described below. To learn about the **default settings and how to use the Charon-SSP configuration options to determine the resource allocation**, refer to the different configuration sections of the general *Charon-SSP User's Guide* of your Charon-SSP version (see CHARON-SSP for Linux), in particular, the *CPU Configuration* section.

Please note the following points regarding the sizing guidelines:

- The sizing guidelines below—in particular regarding number of host CPU cores and host memory—show the **minimum requirements**.
  Every deployment situation must be reviewed and the actual host sizing has to be adapted as necessary. For example, the number of CPU cores available for I/O must be increased if the guest applications produce a high I/O load. Also, a system with many emulated CPUs is typically able to create a higher I/O load and thus the number of CPU cores available for I/O may have to be increased. In a hyper-threading environment, for best performance, the number of CPU cores (i.e., real/physical CPUs) should be sufficient to fulfill CPU requirements of the active emulators, thus avoiding high-workload threads sharing one physical CPU core.
- The CPU core allocation for emulated CPUs and CPU cores for I/O processing is determined by the configuration. See *CPU Configuration* in the general *Charon-SSP User's Guide* for more information about this and the default allocation of CPU cores for I/O processing.

Important general information:

- To facilitate a fast transfer of emulator data from one cloud instance to another, it is strongly recommended to store all relevant emulator data on a separate disk volume that can easily be detached from the old instance and attached to a new instance.
- Please make sure to dimension your instance correctly from the beginning (check the minimum requirements below). The Charon-SSP license for **Charon-SSP AL** is created when the instance is first launched. Changing later to another instance size/type and thereby changing the number of CPU cores will invalidate the license and thus prevent Charon instances from starting (new instance required). If planning to use the **Charon-SSP AL instance in AutoVE mode**, be sure to include the AutoVE server information **before first launch,** otherwise the instance will register with the public license servers (registration only occurs at first launch).
- The license for **Charon-SSP VE** is created based on the fingerprint taken on the license server. If the license server is run directly on the emulator host and the emulator host later requires, for example, a change in the number of CPU cores, the license will be invalidated (new license and possibly new instance required). Hence, it is recommended to run the VE license server on a dedicated instance.

Instance Prerequisites

General CPU requirements: Charon-SSP requires modern x86-64 architecture processors. This could be Intel Servers based on Haswell v3 processors or later, or Desktop Core I7 (CPU frequency at least 3.0 GHz). AMD processors of the same or higher performance are also supported.

Minimum requirements for Charon-SSP:

- Minimum number of host system CPU cores:
  - At least one CPU core for the host operating system, plus:
  - **For each emulated SPARC system:**
    - One CPU core for each emulated CPU of the instance, plus:
    - At least one additional CPU core for I/O processing (at least two, if server JIT optimization is used). See the *CPU Configuration* section mentioned above for configuration options. By default, Charon will assign 1/3 (min. 1; rounded down) of the number of CPUs visible to the Charon host to I/O processing.

- Minimum memory requirements:
  - 4GB or more of RAM for the Linux host operating system. The actual requirements may be higher and will depend on the requirements of the non-emulator services running on the Linux host. The previous recommendation of at least 2GB of RAM for the Linux host will still be valid for many systems, but the increasing requirements of the Linux operating system and applications have led to the updated recommendation for new installations. Plus:
  - **For each emulated SPARC system:**
    - The configured memory of the emulated instance, plus:
    - 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.
- If hyper-threading is enabled on modern x86-64 CPUs, two threads can run on one physical CPU core providing two logical CPUs to the host operating system. If possible, disable hyper-threading on the Charon-SSP host. However, this is frequently not possible in VMware and cloud environments, or it is unclear whether hyper-threading is used or not. The Charon-SSP hyper-threading option enables Charon-SSP to adapt to such environments. See the *CPU Configuration* section in your general *Charon-SSP User's Guide* mentioned above for detailed configuration information. **P lease note**: for best performance, Charon-SSP threads should not share a physical CPU core – enough physical cores should be available on the host system to satisfy the requirements of the configured emulator(s).
- One or more network interfaces, depending on customer requirements.
- Charon-SSP/4U+ and Charon-SSP/4V+ must run on **physical** hardware supporting Intel VT-x/EPT or AMD-v/NPT (baremetal instances) and therefore **cannot run in all cloud environments**. Please check your cloud provider's documentation for the availability of such hardware. In addition, note the following points:
  - Charon-SSP/4U+ and Charon-SSP/4V+ are only available when using a Linux kernel supported by Stromasys. Please refer to the general Charon-SSP user's guide for details (see CHARON-SSP for Linux).
  - Please contact Stromasys or your Stromasys VAR if you need this type of emulated SPARC hardware to discuss your requirements in detail.

Creating and Configuring an OCI Cloud Instance for Charon

Contents

- General Prerequisites
- OCI New Instance Launch

General Prerequisites

As this description shows the basic setup of a Linux instance in OCI, it does not list specific prerequisites. However, depending on the use case, the following prerequisites should be considered:

- To set up a Linux instance in OCI, you need an OCI account.
- Secondly, **prerequisites will be different depending on the planned use of the instance**:
  - Option 1: the instance is to be used as a **Charon emulator host system**:
    - Refer to the hardware and software prerequisite sections of the User's Guide and/or Getting Started guide of your Charon product to determine the exact hardware and software prerequisites that must be fulfilled by the Linux instance. The **image** you use to launch your instance and the **instance type** you chose determine the software and hardware of your cloud instance.
    - A Charon product **license** is required to run emulated legacy systems. Contact your Stromasys representative or Stromasys VAR for details.
  - Option 2: the instance is to be used as a dedicated **VE license server**:
    - Refer to the VE License Server Guide for detailed prerequisites.
- Certain legacy operating systems that can run in the emulated systems provided by Charon emulator products require a license of the original vendor of the operating system. The user is responsible for any licensing obligations related to the legacy operating system and has to provide the appropriate licenses.
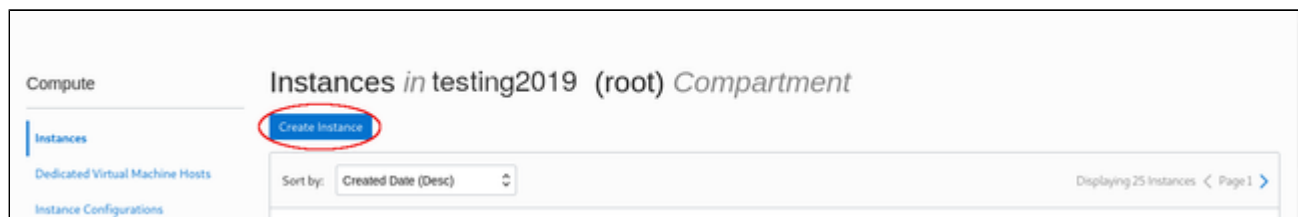
OCI New Instance Launch

Please note: This section only shows a very basic example. Please refer to the Oracle Cloud documentation for more detailed information.

To start the creation of a new cloud instance, perform the following steps:

Step 1: log in to your Oracle Cloud environment.

Step 2: go to the instance list in the compute section and click on **Create Instance**.

This opens the **Create Compute Instance** window.

Step 3: on the first part of **Create Compute Instance** window, name your instance and select the correct image for it. If installing a prepackaged marketplace Charon image, this image must be used. If you plan to install Charon using RPM packages, use a Linux version supported for your Charon product version.

To select the correct image, select **Change Image**. This will allow you to browse the different available categories for the image from which to launch your instance.

The image below shows an example of the image selection screen:

Optionally, change the compartment. Select the correct image and confirm your selection by clicking on **Select Image** at the bottom of the page. This will take you back to the **Create Compute Instance** window.

Step 4: in the middle part of the **Create Compute Instance** window, select the appropriate **shape** (i.e., the virtual Charon host hardware), the **subnet** membership of the instance and whether to assign a **public IP address**. If required, you can also create a new virtual cloud network or a new subnet here.

To select an appropriate shape conforming to the hardware requirements of the emulated SPARC system, click on **Change Shape**.

This will open a window where you can select the correct system type. For flexible shapes you will have to configure the required number of OCPUs.

Select the appropriate shape and confirm your selection by clicking on **Select Shape** at the bottom of the page. This will take you back to the **Create Compute Instance** window.

Step 5: near the bottom of the **Create Compute Instance** window create a new SSH key-pair or upload the public SSH key of an existing key-pair that you will use to access your instance. If you create a new key-pair, you must download the private key and store it in a save place for later use. Optionally, you can also download the public key.

Please note: if your management system supports it, for RHEL 9.x, Rocky Linux 9.x, and Oracle Linux 9.x use SSH key types ECDSA or ED25519. This will allow connecting to these Charon host Linux systems using an SSH tunnel without the default crypto-policy settings on the Charon host having to be changed for less secure settings. This is, for example, important for the Charon-SSP Manager. See also: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/security_hardening/using-the-system-wide-cryptographic-policies_security-hardening.

Step 6: optionally define non-default parameters (including the size) for the boot volume.

The boot volume section allows you to configure the boot volume of your instance with additional non-default parameters. For example, you can configure disk encryption parameters and a non-default system disk size (recommended minimum system disk size: 30GB).

Step 7: support an IMDSv2 authorization header for applications relying on the IMDS service to improve security. For this, open the additional options by clicking on **Show Advanced Options** at the bottom of the instance creation page, select the **Management** tab, and activate the authorization header, as shown below:

For Charon-SSP marketplace images, this is supported starting with Charon-SSP marketplace images version 4.2.2 and VE license server 1.0.33. On existing instances, this parameter can be changed, by editing the instance metadata service settings for the instance (go to **Instance Details** and click on **Edit** in the line **Instance Metadata Service**).
Only change the configuration to IMDSv2 if the image you launched the instance from supports it. Otherwise, you may not be able to connect to your instance. **Please note**: at the time of writing, the official CentOS 7 image on OCI did not support the new feature. If you create an instance to be used as a host for a manual VE license server or Charon VE installation, verify the capabilities of the image used before you enable the new IMDSv2 feature.

Step 8: Additional configuration for **AutoVE** setup.

If the instance is launched from a Charon AL marketplace image and is planned to use AutoVE licensing (instead of the public license servers), you must add the corresponding information to the instance configuration **before** the first launch of the instance:

For OCI, you have to enter a cloud-init script at instance configuration in the **Advanced Options** section.

The following image shows an example:
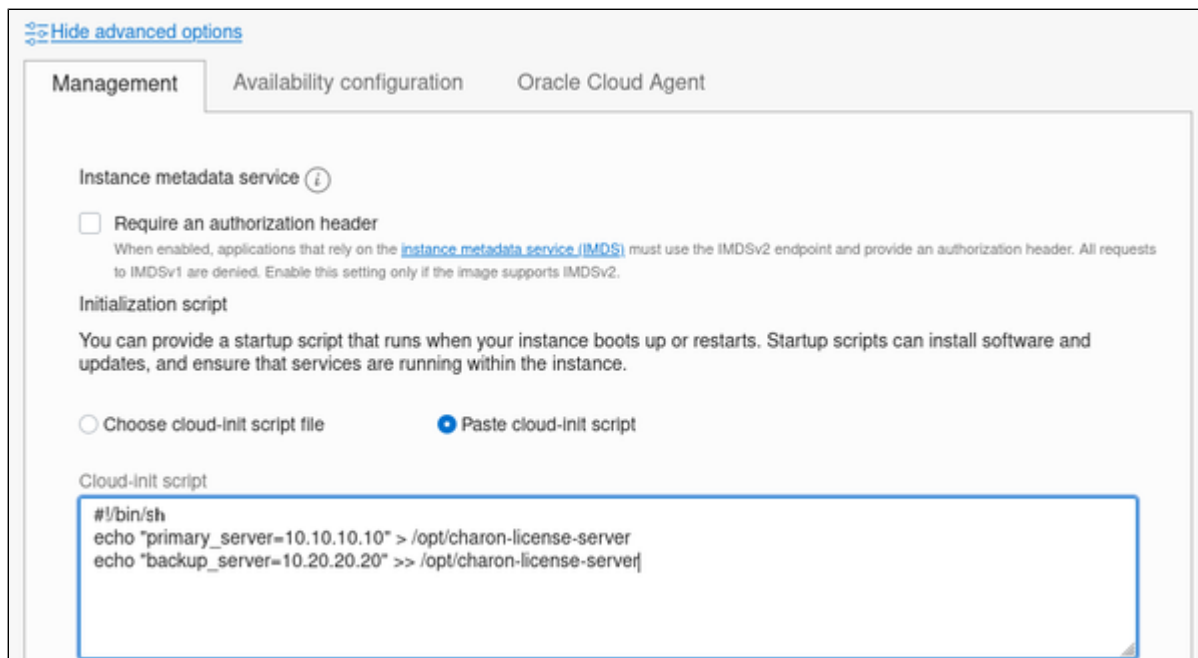
Valid User Data configuration options:

- `primary_server=<ip-address>[:<port>]`
- `backup_server=<ip-address>[:<port>]`

where

- *<ip-address>* stands for the IP address of the primary and the backup server as applicable, and
- *<port>* stands for a non-default TCP port used to communicate with the license server (default: TCP/8083).

Please note: at least one license server must be configured at initial launch to enable AutoVE mode. **Otherwise, the instance will bind to one of the public license servers operated by Stromasys.**

Step 9: The networking type selection may be required to allow offloading parameters to be disabled on an Ethernet interface dedicated to the emulator. For the Charon emulators, offloading parameters on the Ethernet interfaces it uses must be disabled. This is required for proper functionality and good performance of the emulator. To allow this configuration to be correctly reflected in the underlying cloud instance NICs for Charon-SSP versions before 4.1.32, the correct networking type (HARDWARE ASSISTED (SR-IOV) NETWORKING) must be chosen for the instance. For other emulator products, this is required if a dedicated interface is used by the emulator and there are problems with disabling offloading parameters. For this, open the additional options section by clicking on **Show Advanced Options** at the bottom of the network configuration section as shown below:

On this tab select **HARDWARE ASSISTED (SR-IOV) NETWORKING** (after creation, the instance will display the NIC Attachment Type VFIO). Please observe the warning displayed: **not all shapes support this type properly**.

Step 10: Click on **Create** at the bottom of the page to create your instance.

Step 11: verify your instance is running.

Your instance should now be visible in the list of compute instances.

Please note: It is recommended to create separate storage space (disk volumes) for Charon application data (e.g., disk images). If required, such volumes can later easily be migrated to another instance (see *Storage Management*).

Installing the VE License Server Software

Please note:

- The license server kit is included in pre-packaged Charon emulator marketplace images. Alternatively, Stromasys will provide you with a download location.
- Charon emulators that are to use Virtual Environments (VE) licenses require at least one VE license server on the Charon host system itself or an a separate license server. However, it is recommended to run the VE license server on a **dedicated system** to avoid license invalidation caused by changes to the system which are more likely to occur on a system used for other purposes as well, for example, to run a Charon emulator. It is also recommended to install a backup license server to ensure continued operation in case of a failure or invalidation of the primary license.
- For more information, and if you installed your host system using a different base image and the VE License Server RPM packages, please refer to the *VE License Server user's guide* in the Licensing Documentation section of the Stromasys documentation site.
- Additional information for Amazon Linux 2023: by default, Amazon Linux only installs a minimal **gnupg** package. This is not sufficient to unpack the VE license server kits archive. An attempt to unpack the archive will result in the error *gpg: uncompressing failed: Unknown compression algorithm*. To swap the minimal for the full package, use the following command:
  # dnf swap gnupg2-minimal gnupg2-full

In the description below, the placeholders used have the following meaning:

- *<mykey>* is the private key of the key-pair you associated with your cloud instance
  (for an on-premises VMware installation or an installation on a physical system where logging in with username/password is allowed, this is not needed).
- *<user>* is the management user associated with your license server instance. For an instance installed from a Stromasys-provided Charon AL or VE emulator marketplace image, use user *charon* for SFTP and user *sshuser* for interactive login.
- *<linux-ip>* is the ip address of your license server system.

Please note: if an instance was installed from a prepackaged Charon emulator marketplace image, the installation package is already stored in */charon /storage*. Please check, if there are newer versions available that would be preferable for your environment.

Perform the following steps to install the VE License Server software:

1. **Copy the license server software package to the license server host (if needed)**:
   a. For example, use **sftp** to connect to the VE license server system.
      # sftp -i ~/.ssh/*<mykey>* *<user>*@*<linux-ip>*
   b. Copy the software package to the license server system using the following SFTP command:
      > put *<local-path-to-license-server-package>*

2. **Use ssh to log in on the license server host**.
   # ssh-i ~/.ssh/*<mykey>* *<user>*@*<linux-ip>*

3. **As a privileged user (root) go to the directory where you stored the installation package and install the package**:
   a. Become the root user: **# sudo -i**
   b. Go to the package location: **# cd** *<path-to-package-directory>*
      If you used SFTP to copy the package to an instance installed from a prepackaged Charon marketplace image, the home directory of the *charon* user and the default location for file transfers is */charon/storage*.
   c. For VE license server 2.2.4 and above, unpack the archive and agree to the end-user license agreement:
      i. **# sh ./license-server-***<version>***.rpm.sh**
         This will display the EULA. After agreeing to it, for version 2.2.4, the RPM installation package will be unpacked in the current directory. For version 2.2.5 and later, the EULA and the RPM package will be unpacked in a subdirectory (*license-server-<version>.rpm*) of the current working directory.
   d. Install the package:
      i. Go to the directory in which the RPM package is located.
      ii. Linux 7.x: **# yum install license-server*.rpm**
      iii. Linux 8.x and 9.x: **# dnf install license-server*.rpm**

Sample installation:

```
# cd /charon/storage

# dnf install license-server-2.0.1.rpm
Last metadata expiration check: 0:19:36 ago on Di 03 Mai 2022 13:20:02 CEST.
Dependencies resolved.
================================================================================
 Package              Architecture  Version         Repository        Size
================================================================================
Installing:
 license-server       x86_64        2.0.1-1         @commandline      53 M

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

Total size: 53 M
Installed size: 85 M
Is this ok [y/N]: y
Downloading Packages:
     <lines removed>
Running transaction
  Preparing        :                                            1/1
  Running scriptlet: license-server-2.0.1-1.x86_64              1/1
  Installing       : license-server-2.0.1-1.x86_64             1/1
  Running scriptlet: license-server-2.0.1-1.x86_64             1/1
Created symlink /etc/systemd/system/multi-user.target.wants/licensed.service  /etc/systemd/system/licensed.service.

  Verifying        : license-server-2.0.1-1.x86_64             1/1

Installed:
  license-server-2.0.1-1.x86_64

Complete!
```

Installing the Charon Manager

Contents

Overview

The Charon-SSP Manager is the main interface for managing the emulated SPARC systems running on a Charon-SSP cloud host. Therefore, the Charon-SSP Manager must be installed on every system that will be used to manage the Charon instances running on the Charon-SSP cloud host. Configuring and managing Charon-SSP instances from the command-line is also possible, but outside the scope of this Getting Started Guide. Please refer to the general Charon-SSP User's Guide for information about using the command-line.

Typically, for the management of a remote Charon host, the Charon Manager is installed on a system on customer premises, and uses an encrypted connection to manage the Charon host in the cloud. The Charon Manager can also be installed on the Charon host itself and be accessed via X11-Forwarding across an SSH connection. The latter currently requires additional package installation (via standard or local repository) on the Charon host.

Stromasys provides Charon-SSP Manager installation packages for the following operating systems:

- **Linux distributions and versions**:
    - Oracle Linux, Red Hat Enterprise Linux, and CentOS: 7.x or higher (64-bit versions only). Please note that as of 1 January 2022 CentOS 8 is EOL. For new deployments, it is recommended to use a non-EOL alternative. For existing installations, the possible negative impacts of staying with an EOL host operating system should be carefully evaluated.
    - Rocky Linux version 8.x (64-bit) or higher
    - Ubuntu 17 or higher (64-bit)
- **Microsoft Windows:** versions 7, 8, 10, and (starting with Charon-SSP 5.6.1) version 11

Restriction: the Charon-SSP Manager is not supported on Linux hosts using Wayland when they run in a VMware instance with 3D-graphics. The Manager will show erratic behavior in such cases.

Installation Packages

Installation packages are available in RPM or Debian package formats for Linux and as a ZIP-file for Microsoft Windows.

Please note: starting with SSP version 5.6.1, the RPM packages are distributed in an self-extracting archive. The archive required for the Charon-SSP Manager is **charon-gui-**<*version*>**.sh**. It also contains the Charon-SSP Agent which must be installed on the Charon host system to be managed by the Charon Manager. The archive must be unpacked on a Linux system (even if you need the kit for Microsoft Windows).

Use the following command to unpack the RPM packages:

- Go to the directory containing the self-extracting archive.
- Run the script: **# sh charon-gui-**<*version*>**.sh**
- Read the end-user agreement and accept it.
- The RPM packages will be extracted in a subdirectory (*charon-gui-<version>*) of your current working directory.

Names of the Charon-Manager installation packages:

- RPM package: **charon-manager-ssp-**<*version*>**.rpm**
- Ubuntu package: **charon-manager-ssp-**<*version*>**.deb**
- Microsoft Windows package: **charon-manager-ssp-**<*version*>**.zip**

<u>There are different ways to obtain the Charon-SSP Manager installation packages. They are briefly described below:</u>

a) For installation on a management system on customer premises if using a prepackaged cloud marketplace image:

The packages are included in the Charon-SSP cloud-specific image (in */charon/storage*). Once a new instance has been launched, you can download the Charon-SSP Manager archive 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-cloud-instance>*
- Download the required package:
  sftp> get charon-gui-*<version>***.sh**

b) For installation on a Charon host where a conventional RPM installation was performed: Stromasys will provide you with a download link. The Charon Manager packages are also included in the Charon agent RPM and available in */opt/charon-agent/ssp-agent/bin/* once the agent has been installed.

Charon-Manager Installation on Linux

Prerequisites

The Charon Manager can be installed on the Charon host itself or on a remote management system. For the Charon Manager to work, the **Charon Agent must have been installed on the Charon host system**. The Charon Manager communicates with the Agent to configure and manage the emulator instances.

When the Charon Manager is installed on a Linux host with a graphical user environment, the prerequisites are often already fulfilled. However, when installing the Charon Manager on the Charon-SSP host in the cloud or on a Linux server without graphics (for example, to display it via a remote X11-connection) instead of on a local management system, **additional packages** may have to be installed that normally are already available in a workstation environment.

In particular, the Charon-SSP Manager requires the following packages:

- libX11
- xorg-x11-server-utils
- gtk2
- xorg-x11-xauth (only required for X11-Forwarding)

If you install the Charon Manager with the **yum** or **dnf** command, these packages (with the exception of xorg-x11-xauth) and any dependencies that these packages themselves may have, are resolved automatically if a package repository is available. The xorg-x11-xauth package must be installed separately (also with yum). If your server does not have access to the standard operating system repositories, refer to this document for instructions on setting up a local repositories.

Please note:

- The exact list of additionally required packages depends on what is already installed on the server.
- To install dependencies on Ubuntu, please refer to your Linux documentation.

Installation Steps on Linux

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

Step	Description	
1	Installation on a Linux management system on customer premises (typical installation): • Log in to the Linux management system as the **root** user (denoted by the **#** prompt). • Copy the installation package to your local Linux management system (from one of the sources described above).	Installation on the Charon-SSP host system in the cloud (non-typical installation): • Log in and become the root user on the Charon host using the following commands: **$ ssh -i** *<path-to-private-key>* **sshuser@**<i><cloud-instance-ip></i> **# sudo -i** • **Please note:** if the Charon host was not installed using a prepackaged marketplace image, the username may be different and the installation package will have to be copied to the Charon host in a separate step.
2	Go to the directory where the package has been stored: **# cd** *<package-location>*	
3	**Unpacking the shell archive**: • Run the script: **# sh charon-gui-**<i><version></i>**.sh** • Read the end-user agreement and accept it. • The RPM packages will be extracted in a subdirectory (*charon-gui-<version>*) of your current working directory	
4	**Installing the package:**	
	Assuming you are in the subdirectory containing the RPM file, use the following commands for supported Linux systems with RPM package management: • Linux 7.x: **# yum install** *<filename-of-package>* • Linux 8.x and higher: **# dnf install**<i><filename-of-package></i> (For an installation on the cloud host system, check if xorg-x11-xauth is already installed if X11-Forwarding is planned.)	
	For systems with Debian package management (Ubuntu): **# dpkg -i** *<filename-of-package>*	

Installation Steps on Microsoft Windows

The Charon-SSP Manager for Windows software is shipped as a zipped archive package which is contained in the **charon-gui-<*version*>.sh** archive. After unpacking the archive on a Linux system, copy the ZIP file to your Microsoft Windows system and use the following instructions to complete the installation.

1. **Right-click** on the zip archive charon-manager-ssp-{version}.zip and select **Extract All**.
2. A window titled **Extract Compressed (Zipped) Folders** opens. In this window:
   a. Click on the **Show extracted files when complete** checkbox.
   b. Click on the **Extract** button.
3. A new Windows Explorer window opens showing the extracted packages.
4. **Double-click** on the **setup.exe** executable to begin the installation.
5. If you are presented with an **Open File - Security Warning** window, click on the **Run** button.
6. You should now see the Charon-SSP Manager Setup Wizard. To proceed with the installation, click on the **Next** button. If the Windows Installer reports that Charon-SSP Manager for Windows is already installed, you must deinstall the currently installed software before you can install a different version. Normally, several versions can coexist.
7. To accept the default installation options, simply click on **Next** without modifying any options. Alternatively, the following installation options can be adjusted:
   a. Click on **Browse** to select an alternative installation target.
   b. Click the appropriate radio button, **Everyone** or **Just for Me**, to specify system-wide or private installation respectively (the system-wide installation will prompt for the administrator password if you are not using the administrator account).
   c. To determine the approximate disk usage after the installation, click on the **Disk Cost** button.
   d. Once all options have been set, click on **Next**.
8. Proceed with the installation by clicking on **Next**.
9. Once the installation has completed, click on **Close** to exit the SSP-Manager Setup Wizard.
10. The installation process creates:
    a. A Charon Manager icon on the desktop
    b. A Charon Manager entry in the Start menu (folder Stromasys)

Accessing the Charon-SSP Cloud Instance

OCI Security Lists and Network Security Groups

Access to an OCI cloud instance can be controlled by

- an external firewall,
- the operating system firewall of the instance,
- security list of the subnet to which the instance belongs, and
- VNIC-specific Network Security Groups.

Security Lists

Security lists form the original type of virtual firewall offered by the Oracle cloud network service.

A security list acts as a virtual firewall for an instance. It has ingress and egress rules that specify the types of traffic allowed in and out. Security lists are defined **at the subnet level**. Therefore, all VNICs in a given subnet are subject to the same set of security lists.

You can associate multiple security lists with a subnet. Each list can have multiple rules. Traffic is allowed if **any rule in any of the lists** allows the traffic. Traffic is also allowed if it is the response traffic of a permitted tracked connection.

If you don't specify one or more other security lists during the creation of a subnet, a default security list will be associated with it.

Please see the relevant Oracle documentation for more information and configuration details.

Network Security Groups

Network Security Groups (or NSGs) form another type of virtual firewall. Unlike a security list, an NSG does not apply to all VNICs in a subnet, but is assigned to specific VNICs connected to the subnet. This allows a more granular access control. By default, no NSG is assigned to a VNIC.

Please see the relevant Oracle documentation for more information and configuration detail.

Please note: Traffic is allowed if **any rule in any of the relevant lists and groups** allows the traffic. Traffic is also allowed if it is the response traffic of a permitted tracked connection.

Connecting to the Cloud Instance

With the default subnet security list, and without custom Network Security Groups installed, you can, for example, use SSH from the command-line or from a tool such as PuTTY to access the command-line of the user **sshuser** (for Charon-SSP prepackaged marketplace images) or your custom user (for RPM installations) on the Charon-SSP instance. If you select your instance in the instance list and then click on the name, you will see details about your instance including its public IP address as shown below.
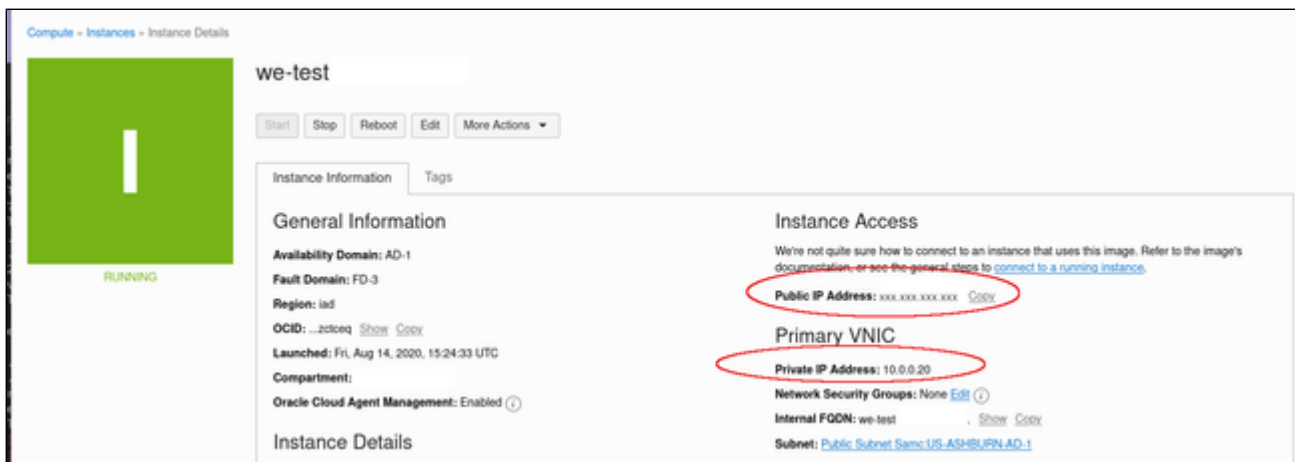
To connect to the instance, you need the private key corresponding to the public key uploaded during the launch of the instance.

Please note:

- The file permissions of the private key file must be set such that the file is only readable by the user (e.g., `# chmod 400 <private-key-file>`).
- PuTTY uses a different key file format. It comes with tools to convert between its own `.ppk` format and the format of OpenSSH used by the default Linux tools.

There are several ways to connect to your Charon-SSP cloud instance using this basic SSH protocol access. Some of them are described in the following sections below:

- SSH Command-Line Access
- SFTP File Transfer
- Connecting with the Charon 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 the necessary security rules 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 Charon-SSP instance.

Please note: 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 in the previous section.

General Login Steps

To connect interactively to an instance installed from a prepackaged Charon-SSP marketplace image, you must connect as the user **sshuser** (for a conventional RPM installation, use the configured user). To connect as the **sshuser**, use the following command:

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

The parameter `ServerAliveInterval` protects the connection from timing out.

Please note: Depending on the type of connection, you will have to use either the public IP address of the Charon host system in the cloud or its address in a customer-specific VPN.

Below, you see sample output of a login (using a private IP address in a customer-specific VPN):

```
$ ssh -o ServerAliveInterval=30 -i .ssh/mykey.pem sshuser@172.31.38.252
Last login: Tue May 21 05:34:33 2019 from myhost.example.com
[sshuser@ip-172-31-38-252 ~]$ pwd
/home/sshuser
```

Please note: This account allows root access (use `sudo -i`).

Setting the Management Password

Information about the initial management password configuration:
Before connecting to the Charon-SSP host instance in the cloud 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.

Please note: The steps described here can also be used to reset a forgotten Charon management password.

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 (exact output may vary depending on product and host system version):

```
$ ssh -i .ssh/mykey.pem  sshuser@172.31.38.252
[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.
[root@ip-172-31-35-32 utils]#
```

SFTP File Transfer

SFTP enables file transfers to and from the Charon-SSP host instance in the cloud. The user for file transfers is the **charon** user if the instance was installed from a prepackaged Charon-SSP marketplace image (for a conventional RPM installation, use the configured user). The security rules must allow SSH access to allow SFTP access to the Charon-SSP cloud instance.

Please note: Depending on the type of connection, you will have to use either the public IP address of the Charon host system in the cloud or its address in a customer-specific VPN.

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

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

Below you see sample output of a connection (using a private IP address in a customer-specific VPN):

```
$ sftp -i ~/.ssh/mykey.pem charon@10.1.1.50
Connected to charon@10.1.1.50.
sftp> ls
charon-manager-ssp-3.1.27.deb           charon-manager-ssp-3.1.27.rpm
media                                   ssp-snapshot
sftp>
```

Connecting with the Charon Manager

Contents

General Information

To use the management GUI for Charon-SSP and the emulated SPARC systems, you must connect to the Charon-SSP cloud instance with the Charon-SSP Manager. The Charon-SSP Manager is the main interface to all important functions of the Charon-SSP software. Managing Charon-SSP via the command-line is possible but outside the scope of this document (please refer to the user's guide of the conventional product for more information).

Notes:

- Typically, **Charon-SSP Manager** is installed either on the Charon host itself (if this system has a graphical interface) on a management system on customer premises. **This is the use-case described in this section**. Other configurations are possible. For example, the Charon Manager could be installed on a non-graphical Charon host in the cloud or in a VMware environment and be displayed on a remote system using X11-Forwarding via an SSH connection.
- **For accessing a Charon host instance in a cloud across the Internet using its public IP address**:
  - The **security configuration** on your Charon host instance must at least allow SSH access. This allows the **built-in SSH tunneling** of the Charon-SSP Manger to work. Should you not use SSH tunneling, you must open up additional ports. However, if the connection runs over the Internet without a general VPN, Stromasys strongly recommends to use SSH tunneling to protect your Charon-SSP cloud instance and any emulated systems running on it.
  - You must know the public IP address of the Charon-SSP host instance in the cloud. To determine this address, refer to the instance information displayed on the cloud 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 a Charon host instance in a cloud via an SSH-based VPN or another VPN solution**:
  - Active SSH-based VPN (see *SSH VPN - Connecting Charon Host and Guest to Customer Network* in the Charon-SSP User's Guide) or other active VPN solution
  - Private IP address of the Charon-SSP host in the VPN

Information about the initial management password configuration:
Before connecting to a Charon-SSP 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 itself as described below.

Starting the Charon Manager and Login to Charon Host

Starting the Charon Manager

To start the Charon-SSP Manager on Linux and to open the Charon Manager login window, use the following command:

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

To start the Charon-SSP Manager on Microsoft Windows, click on the Desktop icon or use the entry in the Start menu.

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 IP address of your Charon-SSP host instance in the **IP address** field.
- Leave the **Password** field empty.
- For cloud instances enable the SSH tunnel configuration (select **ON**).
  Set to **OFF** if connected to *localhost*. The SSH tunnel can generally be used if key-based SSH login is enabled on the target system.
- Change to the SSH tab to fill in the required information if the SSH tunnel has been enabled.

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

- Enter the IP address of your Charon-SSP instance in the **IP address** field.
- Enter the Charon-SSP management password.
- Enable the SSH tunnel configuration for communication across a public network unless you use a secure VPN connection (key-based SSH login required).
- 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
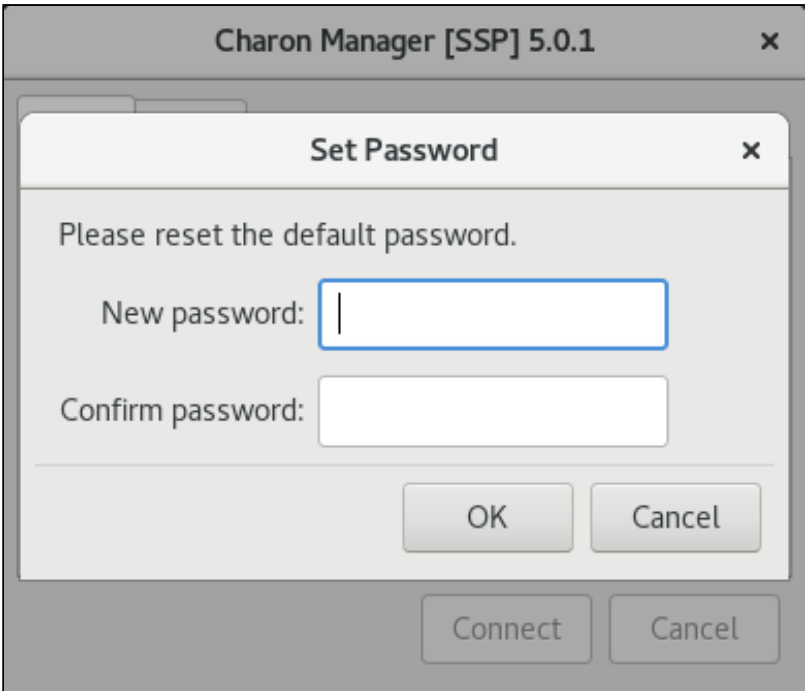
If you use the integrated SSH tunnel, perform the following steps:

- Enter the Charon-SSP user in the **Username** field. For prepackaged images, use **charon** or **sshuser**; for RPM installations use the user for whom the correct public key has been installed.
- Enter the path to the private key file (click on the three dots next to the **Private key** field to open a file browser). You typically associated your cloud instance with this key-pair during instance creation.
- Enter the passphrase for the private key if required.
- Adjust the server port (default 22) if required.

Please note: the public key of the key-pair must be in the `.ssh/authorized_keys` file of the user entered above (**sshuser** and **charon** for prepackaged images).
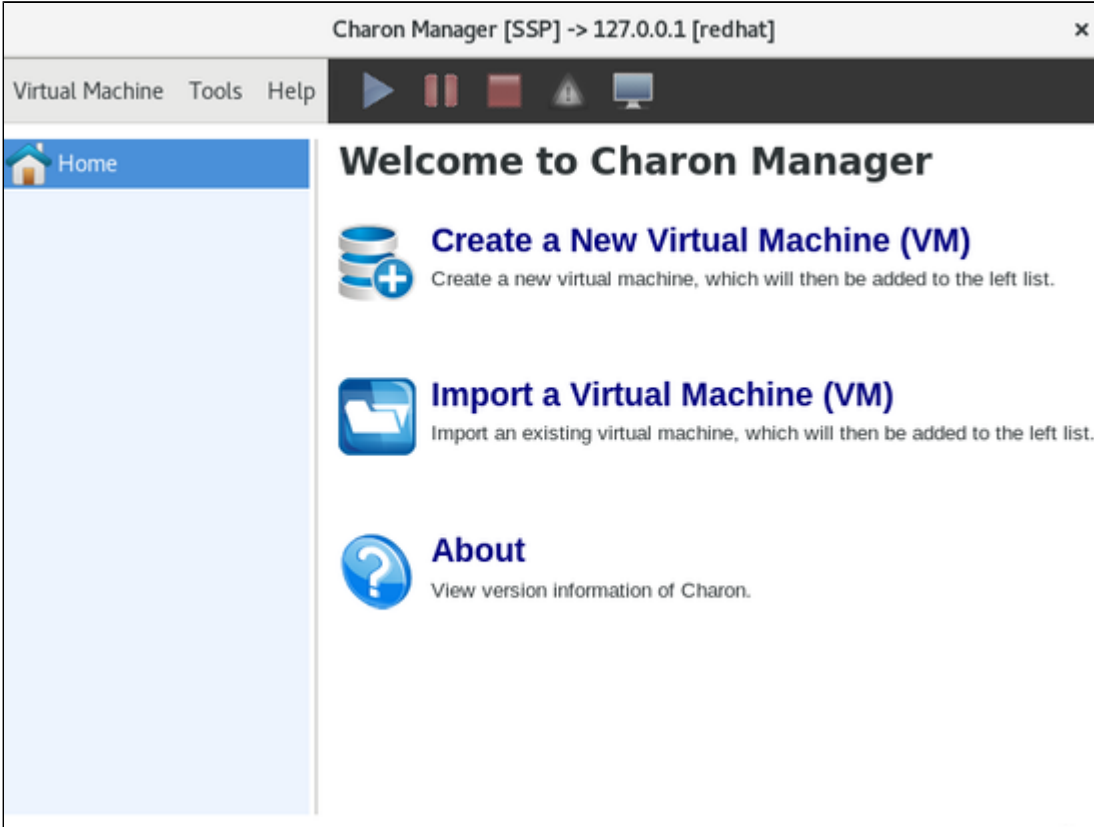
Step 3: connecting to the Charon host system

After entering all the required information, click on **Connect** to connect to the Charon-SSP instance. **If the management password still needs to be set,** you will receive a prompt to enter the new password:

- Enter the desired password in the **New password** field and confirm it in the **Confirm password** field. This management password is then valid for all subsequent logins by the same or a different user until it is changed again. It is not removed if Charon-SSP is reinstalled. Note that older versions of the product will not prompt for the password at first login but will use a default password (**stromasys**). If you need to reset a forgotten management password, please refer to the Charon-SSP user's guide.
- Then click on **OK**.
- The login process continues.

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

Please note: the **title bar** of this screen indicates the managed system type in square brackets (conventional Red Hat installation in the example).

Additional Charon-SSP OCI Instance Configuration

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

Contents

- Storage Management
- Charon Cloud Networking Information

Storage Management

Contents

OCI Storage Environment

In the OCI environment, you can, for example,
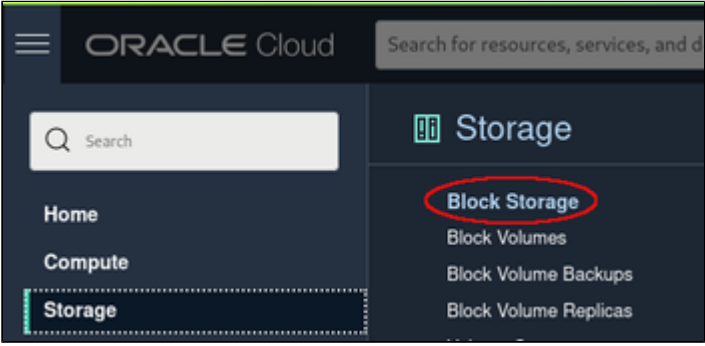
- create a new storage volume (a block volume),
- attach an existing storage volume to your instance,
- detach a storage volume from your instance.

These steps are shown below.

For more details, please refer to the Oracle Cloud documentation.

Creating a New Volume

Step	Details
Open the *Block Storage* configuration from the *Core Infrastructure* menu section.	This will open the block volume list.
Click on *Create Block Volume* to create a new virtual disk.	This will open the block volume creation screen.

Enter the details of the new volume.	Add at least
	- name, - compartment, - availability zone, and - size and performance. Adapt the other settings as required. When done, click on **Create Block Volume**. If you chose to view the volume details after the volume has been created, the detail screen will be displayed.
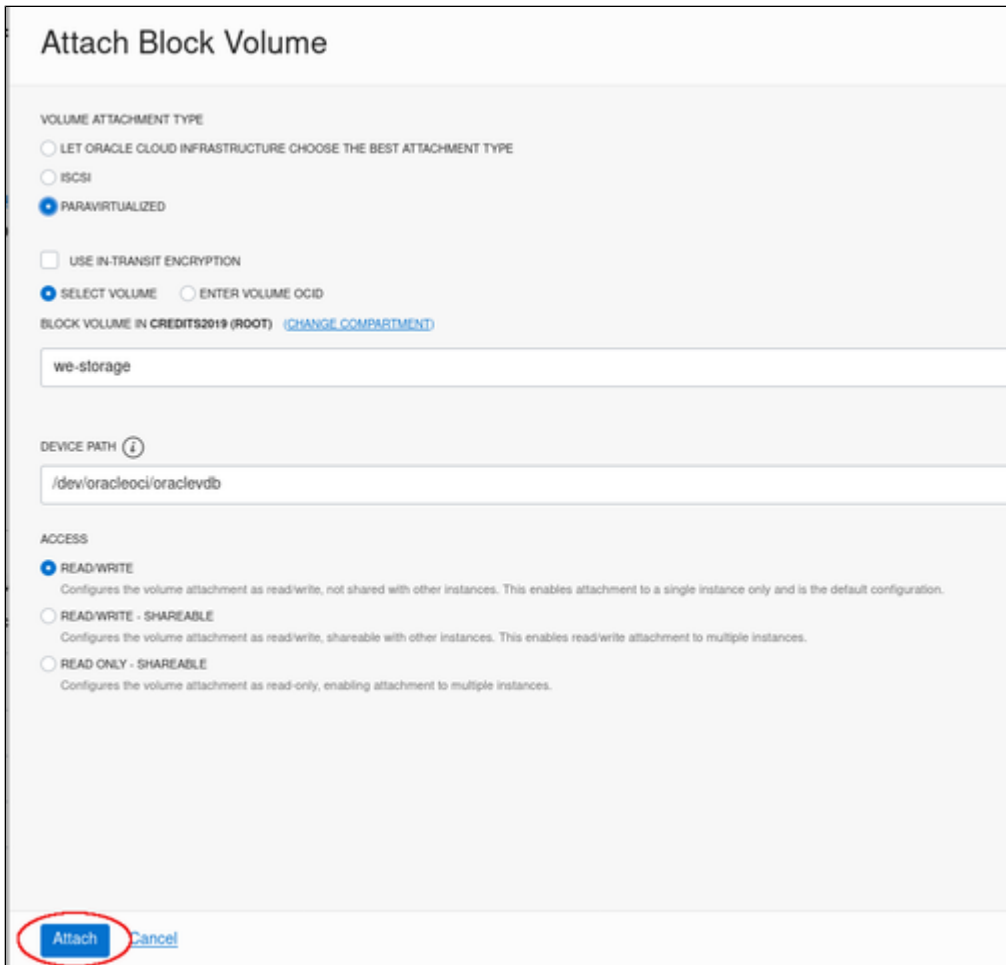
Return to the instance list to attach the new volume to your instance.

Attaching an Existing Volume to an Instance

After returning to the instance list (via **Compute > Instances**), select your instance to display the instance details. Then perform the following steps to attach the block volume to your instance:

Step	Details
Select *Attached Block Volumes* from the *Resources* section on the left.	 In the example above, no volumes have been attached so far. Click on **Attach Block Volume** to select the volume that is to be attached.
Select the parameters for the volume and attach it to the instance.	Select which block volume to attach, the parameters for attaching the device (paravirtualized, read/write), the compartment, and the device path presented to the Charon-SSP host. Once all parameters have been set, click on **Attach**.

Verify that the device has been attached.	The instance details list now shows the attached volume.
	Attached Block Volumes
	Attach Block Volume
	Name ▲ · Status · Volume Type · Device Path · Type · Access · Size · AD · Created
	we-small-disk · ● Attached · Block Volume · /dev/oracleoci/oraclevdb · paravirtualized · Read/Write · 50 GB · AD-1 · Tue, May 12, 2020, 17:27:44 UTC
	Showing 1 Item ‹ Page 1 ›

Detaching a Volume from an Instance

First, **unmount** the volume in the Charon host system (see Charon-SSP Manager section below).

Then detach the volume from your instance:

- Go to the Attached Block Volumes list in the instance details display.
- Click on the **three dots** next to the volume to be detached and select **Detach**.

Steps on the Charon-SSP Host System

Mounting a Newly Attached Volume Using the Storage Manager (AL Images Only)

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

1. Open the Charon-SSP Manager on your local system and connect to your Charon cloud instance.
2. Select **Tools > Oracle 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.
      <u>Please note</u>: 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:
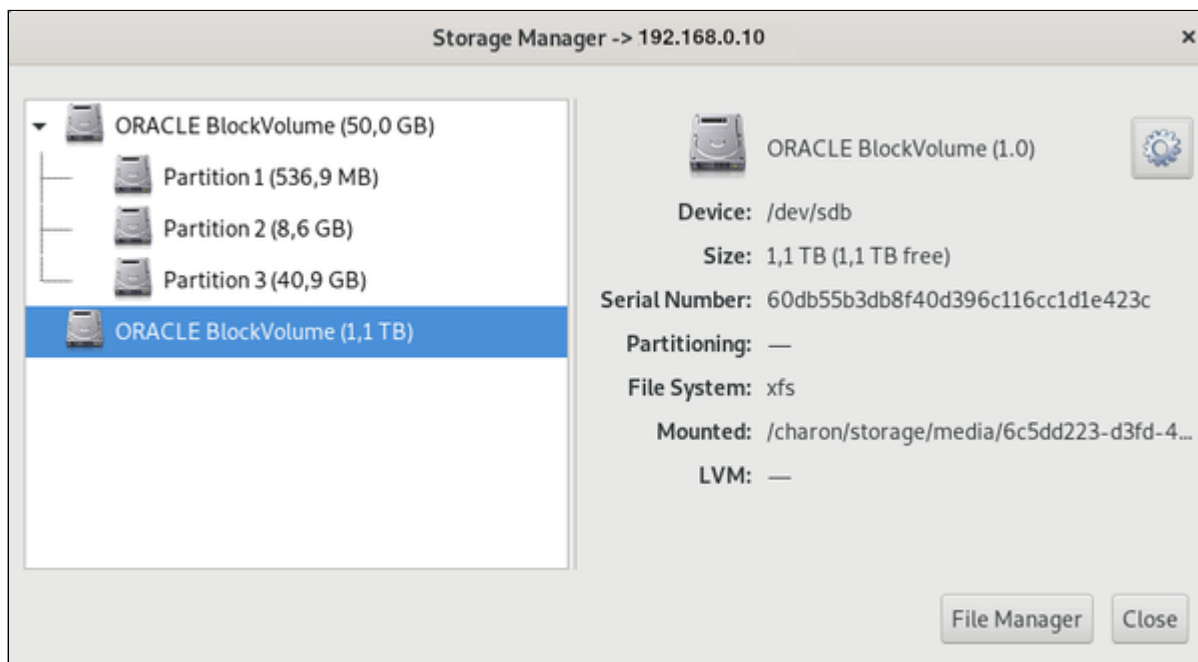
Once the filesystem has been mounted, the space is available to the Charon-SSP host system. After the first mount via the Storage Manager, the filesystem will be automatically mounted after a restart of the Charon host instance.

Mounting a Newly Attached Volume Manually

This is an example of how to mount (and if necessary partition) an additional disk on a Charon host system. Please refer to the Linux manual pages for details.

The general tasks on the Charon host system require to identify the disk, add a file system to it (if this has not been done before), and mount the disk on a suitable mount-point.

Please note: the different cloud environments may offer the disk volumes to the Linux instance using different names. However, the basic steps will be the same as in the examples below.

Step 1: Identify new disk

After logging in on the system, you can identify the new disk using the **lsblk** command:

```
# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   20G  0 disk
sda1   8:1    0  200M  0 part /boot/efi
sda2   8:2    0 19,8G  0 part /
sdb      8:16   0  200G  0 disk
```

In the example above, the new disk is **/dev/sdb**. The output shows no mount-point, i.e., the disk is not mounted yet. It also does not have any partitions.

Please note:

- A disk without partitions can also have a filesystem and data on it. Hence be sure that the disk really does not have any important data on it before you partition it.
- If a system has many disks, it is helpful to run the **lsblk** command before the new disk is added. This makes it easy to identify the new disk in the output after it has been added.

Step 2: Partition disk (fdisk or parted) - **only if required**

Please note: This step is only meant for new disks or to re-partition an existing disk. **It will destroy all data on an existing disk.**

Please refer to the manual pages (`$ man parted` and `$ man fdisk`) of your Linux distribution for details on the disk-partitioning commands. If the whole disk is used for one filesystem, it is not strictly required to create a partition. The decision of which disk layout is required depends on the customer requirements is the responsibility of the user.

After creating one partition on disk with fdisk (**# fdisk /dev/sdb**), the **lsblk** output shows the new partition:

```
# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   30G  0 disk
sda1   8:1    0  500M  0 part /boot
sda2   8:2    0 29,5G  0 part /
sdb      8:32   0   64G  0 disk
sdb1   8:33   0   64G  0 part
```

Step3: Create a filesystem on the new partition(s)

Use the **mkfs** command to create a new filesystem. The selection of a filesystem depends on customer requirements. For example, to create an XFS filesystem, use

```
# mkfs.xfs /dev/sdb1
```

Please refer to the documentation of your Linux distribution for details about the **mkfs** command.

Step 4: Create a mount-point and mount the new filesystem

The following example shows how to create a mount-point and mount the file system.
To keep the example consistent with the sample outputs above, **/dev/sdb1** is used in this example. However, as the */dev/sdX* device names are not guaranteed to be persistent across reboots, it is strongly recommended to use names from the /dev/disk/by-* hierarchy (for example by-uuid) for permanent, production use.

```
# mkdir /space
# mount /dev/sdb1 /space
```

The **df** command shows the mounted filesystem:

```
# df
Filesystem      1K-blocks     Used Available Use% Mounted on
devtmpfs         4065684        0   4065684   0% /dev
tmpfs            4077556       16   4077540   1% /dev/shm
tmpfs            4077556     9224   4068332   1% /run
tmpfs            4077556        0   4077556   0% /sys/fs/cgroup
/dev/sda2       30929148  1677416  29251732   6% /
/dev/sda1         508580    65512    443068  13% /boot
tmpfs             815512        0    815512   0% /run/user/1000
/dev/sdb1       65923628    53272  62498580   1% /space
```

Step 5: Mount the disk automatically at system boot

To mount the disk automatically when the system boots, you must add it to the file **/etc/fstab**.

Please note: The device naming /dev/sdXN (e.g., /dev/sdb1) is not guaranteed to be persistent across reboots. Hence, it is advisable to use a persistent name from the **/dev/disk/by*** hierarchy (for example, the UUID).

You can use the **ls** or the **blkid** command to identify the UUID. Examples:

```
$ ls -l /dev/disk/by-uuid/
total 0
lrwxrwxrwx. 1 root root 10 2020-08-14 21:14 0c523909-fb78-48cb-9dc8-e7a08197a673 -> ../../dm-4
lrwxrwxrwx. 1 root root 10 2020-08-14 21:14 31fa8e8c-a6c0-45f7-9892-da13ba81e0e5 -> ../../sdb1

$ blkid |grep sdb1
/dev/sdb1: UUID="31fa8e8c-a6c0-45f7-9892-da13ba81e0e5" BLOCK_SIZE="4096" TYPE="xfs" PARTUUID="db62deaa-f25f-43d4-b958-
700c1c13d844"
```

To add the device to **/etc/fstab** perform the following steps:

1. As the root user, open the file **/etc/fstab** with a text editor.
2. Add the mount command to the file. **Please note**: The following is for illustration only. The exact options depend on your requirements.
   Sample fstab entry:
   UUID=31fa8e8c-a6c0-45f7-9892-da13ba81e0e5 /space **xfs defaults 1 2**
3. Save the file.
4. Test if the automatic mount works correctly.

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.
- On host systems based on AL images:
  - In Charon Manager go to Tools > Oracle Cloud > Storage Manager.
  - Select the volume.
  - Click on the cogwheel symbol and select **Unmount** the Filesystem.
- On other systems:
  - Use the command **# umount** *<device-path>* or **# umount** *<mount-point>*
  - To make this permanent, remove the corresponding entry in **/etc/fstab**.

Charon Cloud Networking Information

Contents

General Information

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

Linux Versions and NetworkManager

There are significant differences regarding the NetworkManager in the different Linux versions (RHEL 7, 8, 9 and derivatives). This section provides an overview of some important differences.

There are **two basic network configuration systems** in the relevant Linux systems:

- The network service with the network configuration based on **ifcfg-files** in */etc/sysconfig/network-scripts*. This requires the **network-scripts** package.
- The **NetworkManager** with its own configuration file syntax. Persistent configuration files are stored in */etc/NetworkManager/system-connections*. The NetworkManager has a plugin (ifcfg-rh) to handle ifcfg-files. This plugin **does not support** all configuration options of the network-scripts system (e.g., tunnel and tap interfaces are not supported).

Linux 7.x:

- The **network-scripts** and **NetworkManager** methods coexist. It is possible to disable the NetworkManager completely or only for certain interfaces (parameter **NM_CONTROLLED=no** in the ifcfg-file).
- The default NetworkManager plugin is the ifcfg-rh plugin.
- Interfaces managed by the Charon-SSP Manager must have an ifcfg-file and be removed from NetworkManager control (unmanaged interfaces).

Linux 8.x:

- The **network-scripts** package is deprecated. It is not installed by default, but available in the Linux package repositories.
- The default NetworkManager plugin configuration is **ifcfg-rh, keyfile**. The *keyfile* plugin is responsible for handling the native NetworkManager configuration file syntax.
- If virtual bridge configurations including TAP interfaces are configured using **ifcfg-files**, the **network-scripts** package is required. Otherwise, the TAP interfaces cannot be activated (missing support in the ifcfg-rh plugin). Alternatively, such interfaces can be configured as native NetworkManager connections.
- There is an **ifup** command which by default points to nm-ifup. Once the network-scripts package is installed, it points to the ifup command contained in this package.
- The loopback interface (lo) cannot be managed by the NetworkManager.
- Interfaces managed by the Charon-SSP Manager must be under NetworkManager control (managed interfaces).

Linux 9.x:

- The **network-scripts** package is no longer available in the Linux package repositories.
- The default Networkmanager plugin configuration is **keyfile, ifcfg-rh**.
- Existing ifcfg-files can still be read and written, but only if supported by the ifcfg-rh plugin.
- A new **nmcli** command option (`nmcli connection migrate <con-name>`) helps with the conversion of ifcfg-configuration files to native NetworkManager connection profiles. However, this command only works for connections supported by the ifcfg-rh plugin. This means, for example, that TAP interfaces that were previously configured via ifcfg-files must now be recreated using nmcli commands or another NetworkManager configuration tool. Before using the migration command, take a **backup copy** of the content of /etc/sysconfig/network-scripts.
- By default, there is no ifup command. If it is needed, the NetworkManager variant of the command can be installed (**NetworkManager-initscripts-updown**).
- The loopback interface (lo) cannot be managed by the NetworkManager in versions before 9.2.
- Interfaces managed by the Charon-SSP Manager must be under NetworkManager control (managed interfaces).

Additional information about the ifcfg-rh plugin:

The **ifcfg-rh plugin** is used by the NetworkManager to read/write the traditional **ifcfg-files** in */etc/sysconfig/network-scripts*. Each NetworkManager connection corresponds to one ifcfg-file. The plugin does not support all the connection types supported by the original **network-scripts** package. The plugin currently supports Ethernet, Wi-Fi, InfiniBand, VLAN, Bond, Bridge, and Team connections. This means that, for example, TYPE=Tap is not supported and cannot be handled by the NetworkManager in the ifcfg-file format. In Linux 7.x and Linux 8.x, the network-scripts package can be used to support the ifcfg-file format. In Linux 9.x, this package is no longer available. Thus, unsupported connection types must be manually recreated.

Interface MTU Considerations

When configuring a dedicated network interface for an emulator, **ensure that the MTU of the Charon host interface used is not smaller than the MTU used by the legacy guest operating system**. Failing to do so will cause network problems. For further information, please refer to the chapter *Interface MTU Considerations* in this guide.

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. For Charon-SSP, setting up such a configuration is supported by the Charon-SSP Manager (leave the default gateway field empty for the bridge interface). Several hosts configured with guest systems and such an internal bridge can communicate across the cloud-internal LAN and the host systems can route the private IP subnets of the bridges between themselves. L2 protocols are not possible if routing across the cloud LAN is used.

2. **Communication via the cloud-internal 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 configuring the interface correctly, the host and guest can communicate across the cloud-internal 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 the cloud provider 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 for the additional interface. This can be done via configuration files on Linux 7.x. For Linux 8.x **nmcli** commands, the **nmtui** utility, or (for SSP) the Charon Manager can be used.
   - Make a note of the private IP address assigned to the second interface by the cloud provider, and remove it from the Linux configuration (if it has been configured).
   - Assign the interface to the emulated system. This can be done by modifying the emulator configuration file or by using the Charon-SSP Manager.
   - Set the MAC address of the emulated system to the same value as the one used on the host system Ethernet interface. For Charon-SSP, this configuration can be implemented using the Charon Manager.
   - On the guest operating 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.

Please note:

- The section *Dedicated NIC for Guest System* provides some hints on how to configure the second interface in the different situations. Please refer to your cloud-provider's documentation for up-to-date comprehensive information.
- 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.

Please note: Charon hosts based on Charon-SSP AL (Automatic Licensing) marketplace images and using the public license servers always need either direct Internet access or Internet access via NAT from a NAT gateway in the same cloud as the Charon host to access the license server.

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.

Cloud providers usually also provides a VPN gateway instance that can be added to the customer cloud network to connect the cloud network to the customer network (for a charge).

Recommended way to connect the 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 subnets or guest systems with only private IP addresses:

Access to the Internet for subnets with only private IP addresses 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, a dedicated customer-operated gateway, or it can often be provided by the cloud provider for a charge.

Please note: a Charon-SSP AL host system that use the public license servers always needs either direct Internet access or Internet access via NAT from a NAT gateway in the same cloud as the Charon host to access the public license server.

Direct guest system 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 the cloud provider (see also *Dedicated NIC for Guest System*).

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**.

When you add a secondary NIC to a Linux instance, a new interface (that is, an Ethernet device) is added to the instance and automatically recognized by the OS. Depending on the cloud-provider, DHCP may not be active for the secondary VNIC, and you must configure the interface with a static IP address and add any routes that are relevant for the new interface.

Connectivity problems caused by asymmetric routing arise if traffic arrives through one interface and, when the service replies, the reply packets (with the incoming interface's IP address as the source address) go out the other interface. Policy-based routing is required to ensure that packets are sent out via the interface configured with the same IP address that is used as the source IP address in the packet, and to find the correct default gateway (if needed).

Please note:

- The steps below show a simple non-persistent example (can be made persistent if run at every system startup). Please refer to your Linux documentation for details.
- The actual steps depend on your configuration and may vary slightly depending on the specific cloud environment and Linux version. Please always refer to your cloud provider's documentation.

Assumptions:

- The Linux instance has a **primary Ethernet interface** (eth0) with address **10.2.0.8/24** and a public IP address (**PubIP0**).
- The Linux instance has a **secondary Ethernet interface** (eth1) with address **10.2.0.9/24** which also has a public IP address (**PubIP1**).
- All firewalls on the operating system and cloud level are set to allow ICMP traffic to both interfaces.

Problem description:

- A ping from an external host to the public IP address *PubIP0* of the primary Ethernet interface works.
- A ping from an external host to the public IP address *PubIP1* of the secondary Ethernet interface fails.
- A network trace on the cloud host shows that the ICMP packets to *PubIP1* arrive at the cloud instance on eth1 as expected, but there is no answer.

When adding a second IP interface on the Charon-SSP host, the routing problems described above can occur. They can be solved by creating a second routing table and adding a routing policy as shown in the following **example which uses the data provided in the assumptions above**:

```
ip route add 10.2.0.9/32 dev eth1 table 99
ip route add default via 10.2.0.1 dev eth1 table 99
ip route add 10.2.0.0/24 dev eth1 table 99
ip rule add from 10.2.0.9 lookup 99
```

The example has the following effect:

- It creates a non-default routing table (table ID 99) and adds the routes required for the secondary interface to this table. In particular, any primary or alias IP address assigned to the interface must be added.
- It then defines a routing policy that any traffic with the source address of the secondary Ethernet interface must use the non-default routing table. This forces traffic sent to the IP address of the secondary Ethernet interface to also leave the system via this interface.

You can verify the configuration using the commands:

- `ip route show table 99`
- `ip rule show all`

Once you found a configuration solving your problem, you can make the configuration permanent by adding it to a startup script.

Please refer to the Linux man pages for **ip rule** and **ip route** for more information.

Additional information for Charon-SSP marketplace images: the home directory of the **sshuser** contains a script named **active_sec_network.sh**. This script is **only an example** that illustrates how to create a **systemd** service to activate necessary routes and rules during system boot (instead of using steps 7 and 8 above). Do not use this script without carefully adapting it to your requirements - failing to do so, may make your system unreachable.

Cloud Instance and IP Forwarding

If a Charon cloud instance is to forward IP packages between its interfaces (act as a router), in addition to configuring IP forwarding on Linux (`/sbin /sysctl -w net.ipv4.ip_forward=1`), an additional configuration step is required in the configuration of the cloud instance. This configuration has different names in the different cloud environments.

- Source/Destination checking on AWS and OCI must be disabled for all relevant interfaces of the instance.
- IP forwarding on Azure must be enabled for all relevant interfaces of the instance.
- IP forwarding on GCP must be enabled for an instance **when it is created**.
- IP spoofing must be enabled for all relevant interfaces of the instance in the IBM cloud.

Without this configuration, the cloud providers block packets that do not contain the IP address of the cloud instance interface in either the source or destination field.

Interface Configuration Basics

This section shows some basic approaches on how to configure the network interfaces on a Charon host for use by the guest system.That is, the interface should be **activated at boot, but without an IP address**. The IP address assigned by the Cloud provider can then be used by the guest system.

It is by no means a complete documentation but should provide a starting point. Further information can be found in the documentation of your Charon Linux host and the documentation of your cloud provider. Please refer to them for any additional information beyond the basic examples below.

The examples show possible configuration steps on

- Linux systems with file-based network configuration (mostly Linux 7.x), and
- Linux system with NetworkManager-based network configuration (mostly Linux 8.x and higher)

Please note: the **interface names** used in the following section are for illustrative purposes only. Please familiarize yourself with the interface naming conventions used in your cloud environment.

Expected result of the example:

1. The system should still be reachable via **eth0**.
2. Interface **eth1** should be up without having an IP address configured.

Basic File-based Interface Configuration without NetworkManager

This configuration applies to systems with a file-based network configuration where the NetworkManager is either not active, or where network interfaces should be excluded from NetworkManager control (e.g., to be managed by the Charon Manager). The NetworkManager is disabled by default in older Charon-SSP marketplace images that are based on Centos 7.

Please note:

- The sample configuration assumes a CentOS 7 system and that the interface is configured outside the control of the NetworkManager.
- Should the NetworkManager be active, the plugins **ifcfg-rh** and **ibft** must be enabled in section **main** of the NetworkManager configuration file */etc /NetworkManager/NetworkManager.conf*.
- After the initial creation of the ifcfg-file, the interface can be managed by the Charon-SSP Manager.

To make the second interface usable for the Charon guest system, perform the following steps:

1. Add a second interface to your instance as described in the cloud-specific Getting Started guide and your cloud provider's documentation.

2. Log into the instance and become the root user (use: `sudo -i`)

3. Identify the names of the two Ethernet interfaces:
   `# ip link show`

4. Create an interface configuration file for the second interface (the file for the first one should exist). Example (use correct interface name for your configuration):
   `# cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth1`

5. Edit this file to match the characteristics of **eth1** (use correct interface name for your configuration). 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 example below.

   Please note:
   On some cloud platforms, the automatic cloud-specific configuration prevents the entries in the **ifcfg**-file to take effect (for example on GCP). Please refer to your cloud-provider's documentation and the *Network Management* section in the *Getting Started Guide* of your version for additional information.

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

6. Restart the network:
   `# systemctl restart network`
   Please note: Should there be an error when executing this command, kill the DHCP client process and retry the command.

Basic Interface Configuration with NetworkManager

This configuration applies to systems where the NetworkManager is active and network interfaces are under NetworkManager control. The NetworkManager is enabled by default in newer Charon-SSP marketplace images that are based on Rocky Linux 8.x.

Please note:

- The **interface names** used in the following section are for illustrative purposes only. Please familiarize yourself with the interface naming conventions used in your cloud environment.
- The sample configuration assumes a Rocky Linux 8.x system and that the interfaces are under the control of the NetworkManager.
- On some cloud platforms, the automatic cloud-specific configuration prevents the operating system configuration to take effect (for example on GCP). Please refer to your cloud-provider's documentation and the *Network Management* section in the *Getting Started Guide* of your version for additional information.

In such environments, you have different options to configure network interfaces for use by the guest system. The main options are the following:

1. On a Charon-SSP system, use the Charon Manager Network Settings utility
2. On a Linux system with a graphical user interface, use the provided graphical network management tools. This is typically not available in cloud environments.
3. On a Linux system without a graphical user interface, use the **nmtui** utility or **nmcli** commands.
4. Manually create and modify ifcfg-files in /etc/sysconfig/network-scripts.

The following sections show samples for options 1 and 3.

Charon-SSP Manager Network Settings

The Charon-SSP Manager provides basic network configuration options.

- To access them, start the Charon Manager and open the menu option:
  Tools > Network Settings
- To configure a host system for use by the emulator perform the following steps:
  - Select the correct interface.
  - In the **IP setting** field select **None**.
  - Click on **Apply**.

Using the *nmtui* Utility

The **nmtui** utility provides a method to configure the network settings for the NetworkManager in a text-based environment without having to know the *nmcli* commands. It is provided via the **NetworkManager-tui** package.

The following **basic example** shows how to remove the IP address from the interface and how to reactivate the interface afterwards.

- Start the tool as the root user: `# nmtui`
- Use the up/down and left/right arrows to navigate.
- Select **Edit a connection** and press **RETURN**.
- Select the interface you want to configure.
- Select **Edit** on the right side and press **RETURN**.
- To make the interface come up without an IP address at boot, set the IP configuration to disabled (pressing **RETURN** on the value field will open a menu), and enable the automatic connection.
- Select **OK** and press **RETURN**.
- Navigate back to the main screen.
- Select Activate a connection and press **RETURN**
- **Select the interface** you want to reactivate and select **Deactivate** on the right. Press RETURN.
- Repeat the steps for the **Activate** option to reactivate the interface.
- Navigate back to the main screen and end the session.

Using *nmcli* Commands

To configure the interface dedicated to the emulator such that it receives no IP address but is activated at start, you could use command similar to the following:

1. Identify the NetworkManager connection to configure. The interface may have been automatically activated by the NetworkManager. In the example, it is "Wired connection 1" on device eth1.

```
# nmcli conn show
NAME                UUID                                    TYPE      DEVICE
System eth0         5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03   ethernet  eth0
Wired connection 1  027a1c2b-3397-37fb-a6e2-f2e02eb59992   ethernet  eth1
```

If there is no connection for the interface yet, check if the device is visible using the command `nmcli dev status` or `ip link show`.

2. For an existing connection:

  a) Configure an appropriate name for the connection if required:

```
# nmcli conn mod "Wired connection 1" con-name eth1
```

  b) Set the IP configuration such that no IP address is assigned:

```
# nmcli conn mod eth1 ipv4.method manual ipv4.address 0.0.0.0
```

  c) Configure automatic interface activation at boot:

```
# nmcli conn mod eth1 connection.autoconnect yes
```

3. **If no connection for the second interface exists**:

  Add a new connection (with automatic interface activation, without IP address):

```
# nmcli conn add con-name eth1 type ethernet ifname eth1 autoconnect yes ipv4.method manual ipv4.addresses 0.0.0.0
```

4. (Re-)Activate the connection:

```
# nmcli con up eth1
```

Further Information

The following sections provide additional information:

- Network Interface Management (adding additional NICs to the Charon host instance)
- Interface MTU considerations
- Configuration example: SSH VPN tunnel to connect Charon host and guest to remote systems or customer networks
- Configuration example: dedicated NIC for the Solaris guest system

Network Interface Management

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

Please note: The steps below only provide a basic overview. The exact tasks required will vary depending on your network design. Please refer to the Oracle cloud documentation for details. In particular, adding additional VNICs to a Linux system running on a baremetal OCI instance requires additional steps to enable the operating system to recognize the additional interface. See https://docs.cloud.oracle.com/iaas/Content/Network/Tasks /managingVNICs.htm for details.

Contents

- Adding a New Network Interface to an Instance
- Removing a VNIC from an Instance
- Address Assignment Information

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

Please note: for Charon emulators to run properly, certain offloading parameters on the Ethernet interfaces they use must be disabled (see the respective product user's guides). In some cases, the configuration on the Charon host level is not correctly reflected in the underlying OCI cloud instance NICs if the default networking type is used. The correct behavior can often be facilitated by selecting the networking type HARDWARE ASSISTED (SR-IOV) NETWORKING when creating the instance. This setting is required for Charon-SSP versions before 4.1.32 (see *Setting up an OCI Instance for Charon-SSP* in Charon-SSP for OCI Cloud Getting Started Guide of your version).

Adding a New Network Interface to an Instance

The following steps are required to create a new network interface and to add it to an instance:

Step	Details
Open the list of compute instances and select your instance.	Click on **Compute > Instances**. Click on the instance name to open the instance details screen.
Select *Attached VNICs* **in the** *Resources* **section on the left.**	The screen shows the already attached VNICs. Click on **Create VNIC** to continue. This will open the VNIC creation window.

Add the necessary data to configure the VNIC.	In this window, add for example • the name of the VNIC, • the virtual cloud network and the subnet to which the VNIC is connected (can be created via **Compute > Networking > Virtual Cloud Networks**), • the private IP address of the VNIC (within the CIDR block of the virtual cloud network), • whether a public IP address should be assigned to the VNIC. Once you added all the desired information, click on **Create VNIC**.
Verify that the new interface has been attached to the instance.	The instance details section should now show the newly attached VNIC. Example:

Removing a VNIC from an Instance

To remove a VNIC from an instance, select the instance in the instance list and display the **Attached VNICs** in the **Resources** section.

Then click on the *three dots* next to the VNIC and select **Delete**.

Please note: the primary VNIC cannot be deleted.

Address Assignment Information

Each VCN is assigned a block of private IP addresses. This block can be split by the user to form several IP subnets. Routing within one VCN works automatically.

When an OCI instance is launched into a subnet or if a new vNIC is being added,

- it is automatically assigned a private IP address from the address range assigned to the subnet (unless the user configured an address from the subnet range manually in the instance configuration),
- the user can choose whether to assign a public IP address if the subnet is a public subnet.

Please note:

- 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.
- Both, private and public IP addresses for a vNIC can be displayed via the vNIC detail view in the OCI GUI (**Instance details > Attached VNICs > *vNIC name***) .
- On the **Linux level**,
    - the private IP address assigned to the **primary vNIC** is configured **automatically via DHCP**.
    - the private IP address assigned to any **secondary vNIC** must be **configured manually** (DHCP is not possible).

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

The following address range is reserved, for example, to allow OCI 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 instance VNICs (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 OCI for the default router
- 10.1.1.255: network broadcast address.

Other special addresses:

- 169.254.0.0/16: Reserved for OCI use.

Public IP addresses:

There are two types of public IP addresses (only available in public subnets):

- **Ephemeral addresses**:
    - maximum one per VNIC,
    - assigned by Oracle,
    - persistent during the lifetime of the associated private IP address,
    - can only be associated with the primary private IP address of a VNIC,
    - a user can only delete it but not associate it with a different private IP address.
- **Reserved addresses**:
    - maximum 32 per VNIC,
    - created and assigned by the user,
    - can be re-assigned to a different private IP,
    - can be associated with primary and secondary private IP addresses,
    - exists until the user deletes it.

SSH VPN - Connecting Charon Host and Guest to Customer Network

Contents

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 instances hosted in a cloud, 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 host and a remote Linux system across a public network. Topologies that are more complicated will require other, more sophisticated, solutions.

Please note:

- 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.

The tunnel in this example has two endpoints:

- **The remote Linux system**: in this example, this system could be in the customer on-premises network and use the tunnel configuration to connect across the Internet to a Charon-SSP host system in the cloud. If in conformance with the customer security policies, the configuration could be expanded to make this Linux system the router between the customer network and the Charon-SSP host system (optionally including guest systems) in the cloud.
- **The Charon-SSP host system**: in this example, the Charon host system could be in a public cloud and require a connection to other customer devices across the Internet.

Prerequisites

The example shows how to use the Charon Manager on the Charon-SSP 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 host instance in the cloud.
- The private key necessary to access the instance must be available on the remote Linux system. The key-pair required to access the cloud instance is typically associated with the instance when it is created.
  <u>Please note</u>: If the key-pair is not created automatically during the launch of the instance, you can create it using a command similar to the following:
  Sample 1: **# ssh-keygen -t rsa -b 4096 -f ~/.ssh/***<keyname>* **-q**
  Sample 2: **# ssh-keygen -t ecdsa -f ~/.ssh/***<keyname>* **-q**
  The resulting key-pair can then be associated with instance during instance creation or later by adding it to the *authorized_keys* file of the correct user, and then be used to create an encrypted SSH connection.
  Please note: if your management system supports it, for RHEL 9.x, Rocky Linux 9.x, and Oracle Linux 9.x use SSH key types ECDSA or ED25519. This will allow connecting to these Linux systems using an SSH tunnel without the default crypto-policy settings on the Charon host having to be changed for less secure settings. See also: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/security_hardening/using-the-system-wide-cryptographic-policies_security-hardening.
- If the Linux host system runs RHEL, CentOS, or Oracle Linux 7.x the **bridge-utils** package must be installed, if the Charon Manager configuration options (vs. manual configuration) are to be used.

- The **autossh** package must be installed on the remote Linux system.
- The Charon host must allow SSH tunnels. This is preconfigured on Charon-SSP marketplace images. On conventional RPM installations, make sure that **PermitTunnel** is set to **yes** in **/etc/ssh/sshd_config**. If the root user is to be used for the tunnel creation, a key-based login should be set up for this user (**PermitRootLogin** set to **without-password**), Restart the SSH daemon after changes to the file (**# systemctl restart sshd**).

Setting up the VPN Tunnel

The image below shows a sample setup. This section describes how to configure this sample setup.
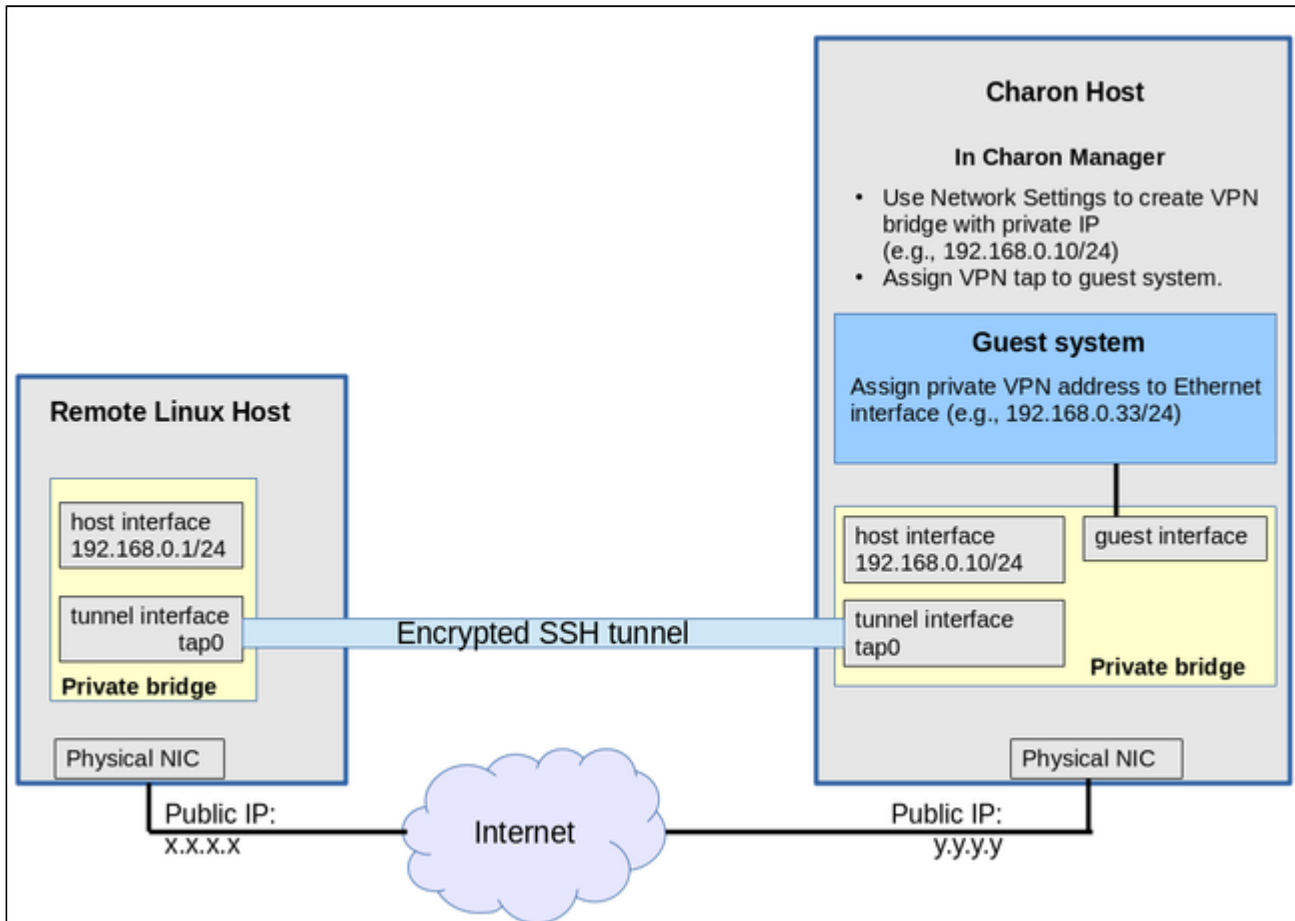
Steps on the Charon-SSP Host System

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**.

Please note: 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:	ON
Binding interface:	OFF
STP for bridge:	OFF
Virtual bridge interface:	
Virtual bridge name:	vpn0
Number of virtual adapters:	1
IP settings:	Manual
IP address:	192.168.0.10
Netmask:	255.255.255.0
Gateway:	
DNS server 1:	
DNS server 2:	

OK   Cancel

To learn more about the virtual network configuration options, refer to section *Host System Network Configuration* in the general Charon-SSP User's Guide.

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.

Please note: 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 effective.

Steps on the Remote Linux System

Please note: 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 (the ip commands are not persistent across reboots; they should be put into a script once the configuration is working):

Action	Command
Create TAP interface	`# ip tuntap add dev tap0 mod tap`
Enable TAP interface	`# ip link set tap0 up`
Create bridge	`# ip link add name br_vpn0 type bridge`
Enable bridge interface	`# ip link set br_vpn0 up`
Define IP address for bridge	`# ip addr add 192.168.0.1/24 dev br_vpn0`
Add TAP interface to bridge	`# ip link set tap0 master br_vpn0`

Start the SSH tunnel
 	
autossh is a program to start a copy of **ssh** and monitor it, restarting it as necessary should it die or stop passing traffic.
 	
Once started, you can move the program to the background.
	`# autossh -M 9876 -o ServerAliveInterval=60 -o Tunnel=ethernet \`
	`   -w 0:0 -t -i <path-to-private-key> -NCT sshuser@<public-cloudinstance-IP>`
	
	-M defines the monitoring port autossh uses to monitor the connection
	-o sets SSH options (bridged tunnel and keepalive)
	-i denotes the path to the private key matching the public key copied to the host system.
	-w denotes the number of the local and remote tunnel interfaces for tunnel device forwarding (e.g., the 0 in interface tap0).
	-N denotes that no remote command should be executed
	-T disables pseudo-terminal allocation
	-C requests data compression
	
	Value for parameter user: On instances based on prepackaged marketplace images use **sshuser**, on other systems use the **root** user or another user for whom you installed the public key.

<u>Possible additional steps</u>:

To turn the remote Linux system into a router for other systems in the customer network, perform the following steps:

1. Enable IP Forwarding using the command:
   `# sysctl -w net.ipv4.ip_forward=1`
   To make permanent, add `net.ipv4.ip_forward=1` to the file **/etc/sysctl.conf**.
2. If the Linux firewall is enabled (firewalld assumed), allow the forwarding of packets through the firewall. Basic example:
   `# firewall-cmd --permanent --direct --add-rule ipv4 filter FORWARD 0 -i <tunnel-bridge-interface> -o <NIC-to-LAN> -j ACCEPT`
   `# firewall-cmd --permanent --direct --add-rule ipv4 filter FORWARD 0 -o <NIC-to-LAN> -i <tunnel-bridge-interface> -j ACCEPT`
3. 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..

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 net`*X* and `ipadm create-addr -T static -a` *<ip-address>/<netmask>* `net`*X*`/v4`.

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 ffffff00 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*.

Stopping the SSH Tunnel

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

Action	Command
Terminate the autossh process	`# kill -9 <autossh-pid>`
Terminate remaining SSH tunnel connections	`# kill -9 <tunnel-ssh-pid>`
Delete the bridge	`# ip link delete br_vpn0`
Delete the TAP interface	`# ip link delete tap0`

Dedicated NIC for Guest System

Contents

Providing a dedicated NIC for guest operating systems is the standard method in non-cloud environments. However, this configuration poses some challenges in cloud environments where MAC address / IP address combinations are fixed parameters set by the cloud provider.

This section will provide some information about how to configure such a setup in a cloud environment. **It is not specific to one cloud provider. Hence, the descriptions may refer to different cloud providers if appropriate.**

Basic Concept

The following images illustrates the basic concept when working with a dedicated network interface for the guest operating system. There are, of course, many variations depending on the specific environment.

Scenario: host and guest system have a dedicated NIC. The NIC used by the Charon host has a private and a public IP address, the NIC used by the guest system a private IP address and optionally a public IP address. The Internet and VPN gateways are only used for illustration and are not part of this example.

Charon host instance with two NICs
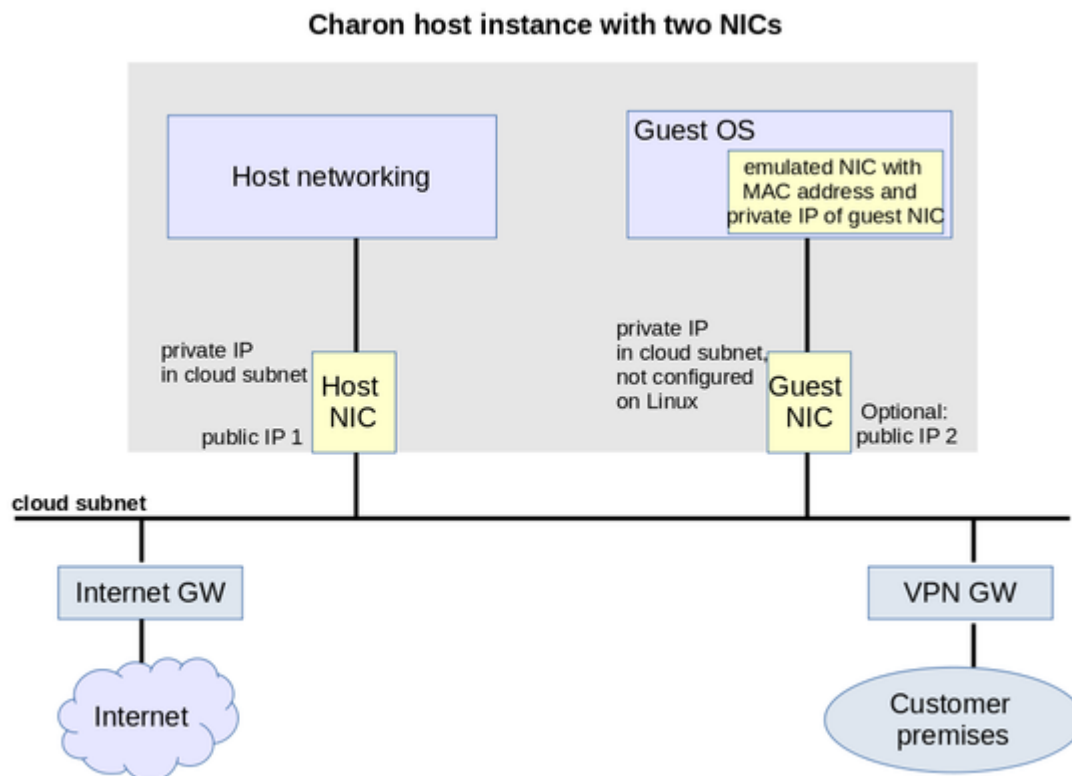
Please note: If the NIC dedicated to the guest OS does not have a public IP address, the guest system may still be able to access the Internet via the customer network reachable across a VPN gateway. This will depend on the customer specific network configuration. This type of connection is the recommended way to provided external network access to the guest system as the VPN ensures that traffic across a public network is encrypted.

The basic steps to implement the above configuration are as follows:

- Create a cloud instance in which the Charon host system runs.
- Add two NICs to the Charon host system. One for the Charon host and one for the guest system.
- Configure the appropriate access rules for instance and NICs.
- One NIC is dedicated to the Charon host, one to the guest system. Configure a private and public IP address for the NIC used by the Charon host. Configure a private IP address for the NIC used by the guest system (and optionally a public IP address - not recommended).
- On the Charon host, remove the private IP address from the NIC dedicated to the guest system if it was automatically configured and ensure that the interface will be enabled when the system starts.
- Assign the appropriate NIC to the guest system.
- Configure the guest system MAC address to be the same as the one of the NIC selected for the guest.
- After booting the guest system, configure the private IP originally assigned to the guest NIC by the cloud provider as the IP address of the guest Ethernet interface.
- Set the default route of the guest system to the default gateway or VPN gateway of the LAN.

Depending on firewall rules and cloud-specific security settings, the guest system should then be able to communicate with the following systems:

- The host system.
- The other systems in cloud-internal network (e.g. other guest and host systems).
- The customer internal network via a previously configured VPN gateway.
- Directly with the Internet if a public IP address was configured for the interface (not recommended).

The additional sections in this chapter show the basic configuration steps for the above example.

Please note:

- In this scenario any direct traffic between host and guest system (if configured with a public IP address) and external systems reachable via the Internet gateway is not encrypted by default. If this traffic runs across a public network, it is exposed to being monitored and even modified by third parties. The user is responsible for ensuring data protection conforming to the user's company security rules. It is strongly recommended to use encrypted VPN connections for any sensitive traffic.
- Guest operating systems are often old and no longer maintained by the original vendor. This means they are more easily compromised by attacks from the Internet. Therefore, direct Internet access for the guest system is not recommended.
- The actual configuration steps vary depending on the cloud environment used. The sample configuration below will have to be adapted to the specific environment.

Configuration Examples

Important information:

- The example assumes that the host operating system is a RHEL version 7 or 8 compatible Linux system. If you use a different host operating system version, you must adapt the example accordingly.
- **As explained for AWS**, remember that any automatically assigned public IP addresses will be removed by the cloud provider once the instance is restarted with a second NIC. Hence, on AWS Elastic IP addresses must be used.
- **For Google cloud**, note the following:
  - The default is that all interfaces are configured with IP addresses automatically by GCP services on the Linux host. Please refer to the *Network Management* section in the respective *Getting Started* guide for information on how to disable this automatic configuration.
  - Some base images used to create a Charon host instance may be configured to use /32 netmasks for additional interfaces, and only ARP requests for the default gateway are answered by Google. This can cause communication problems between Solaris and other instances on the same subnet (ARP requests are not answered). The workaround is to use static ARP entries on Solaris. Please refer to the *Getting Started* guide for more information. Current images provided by Stromasys use /24 netmasks, so this point does not apply to them.
- The interface names used in this example (eth0 and eth1) may be different on your system. Please verify the names on your system and refer your cloud provider's documentation for more detail. **Make sure you use the correct names**!
- The example uses only a private address for the dedicated interface. If a public address is required, the basic steps for making the interface available to the guest system are the same.

Step 1: Prepare Second NIC on Host for Use by the Solaris Guest

The host system interface configuration must ensure that the private IP address allocated to the new interface by the cloud provider is not configured on the Linux Ethernet interface. This address will be used by the guest system.

The configuration depends on whether the network configuration on the Linux host is file-based (typically Linux 7.x) or NetworkManager-based (typically Linux 8.x). Examples for both are shown below. Typically, the NetworkManager is disabled on Charon-SSP marketplace images based on Linux 7.x, and enabled on Charon-SSP marketplace images based on Linux 8.x.

Expected result of the examples in this section:

1. The system should still be reachable via **eth0**.
2. Interface **eth1** should be up without having an IP address configured.

File-based Configuration Examples

This configuration applies to systems with a file-based network configuration where the NetworkManager is either not active, or where network interfaces should be excluded from NetworkManager control (e.g., to be managed by the Charon Manager). The NetworkManager is disabled by default in older Charon-SSP marketplace images that are based on Centos 7.

Please note:

- The sample configuration assumes a CentOS 7 system and that the interface is configured outside the control of the NetworkManager.
- Should the NetworkManager be active, the plugins **ifcfg-rh** and **ibft** must be enabled in section **main** of the NetworkManager configuration file */etc/NetworkManager/NetworkManager.conf*.
- After the initial creation of the ifcfg-file, the interface can be managed by the Charon-SSP Manager.

To make the second interface usable for the Charon guest system, perform the following steps:

1. Add a second interface to your instance as described in the cloud-specific Getting Started guide and your cloud provider's documentation.

2. Log into the instance and become the root user (use: `sudo -i`)

3. Identify the names of the two Ethernet interfaces:
   `# ip link show`

4. Create an interface configuration file for the second interface (the file for the first one should exist). Example (use correct interface name for your configuration):
   `# cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth1`

5. Edit this file to match the characteristics of **eth1** (use correct interface name for your configuration). 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 example below.

   Please note:
   On some cloud platforms, the automatic cloud-specific configuration prevents the entries in the **ifcfg**-file to take effect (for example on GCP). Please refer to your cloud-provider's documentation and the *Network Management* section in the *Getting Started Guide* of your version for additional information.

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

6. Restart the network:
   `# systemctl restart network`
   Please note: Should there be an error when executing this command, kill the DHCP client process and retry the command.

NetworkManager-based Configuration Examples

The following sections show two examples:

- Configuration using **nmcli** commands
- Configuration using the Charon-SSP Manager

Using nmcli Commands

To configure the interface dedicated to the emulator such that it receives no IP address but is activated at start, you could use command similar to the following:

1. Identify the NetworkManager connection to configure. The interface may have been automatically activated by the NetworkManager. In the example, it is "Wired connection 1" on device eth1.

```
# nmcli conn show
NAME                UUID                                    TYPE      DEVICE
System eth0         5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03   ethernet  eth0
Wired connection 1  027a1c2b-3397-37fb-a6e2-f2e02eb59992   ethernet  eth1
```

If there is no connection for the interface yet, check if the device is visible using the command **nmcli dev status** or **ip link show**.

2. For an existing connection:

    a) Configure an appropriate name for the connection if required:

```
# nmcli conn mod "Wired connection 1" con-name eth1
```

    b) Set the IP configuration such that no IP address is assigned:

```
# nmcli conn mod eth1 ipv4.method manual ipv4.address 0.0.0.0
```

    c) Configure automatic interface activation at boot:

```
# nmcli conn mod eth1 connection.autoconnect yes
```

3. **If no connection for the second interface exists**:

    Add a new connection (with automatic interface activation, without IP address):

```
# nmcli conn add con-name eth1 type ethernet ifname eth1 autoconnect yes ipv4.method manual ipv4.addresses 0.0.0.0
```

4. (Re-)Activate the connection:

```
# nmcli con up eth1
```

Using the Charon Manager

The Charon-SSP Manager provides basic network configuration options.

- To access them, start the Charon Manager and open the menu option:
  Tools > Network Settings
- To configure a host system for use by the emulator perform the following steps:

  - Select the correct interface.
  - In the **IP setting** field select **None**.
  - Click on **Apply**.

Step 2: Add the Dedicated NIC to the Emulator Configuration

- Start the Charon Manager and open the configuration window for the emulated system.
- Configure the emulated system with the dedicated Ethernet interface as its interface.
- Set the MAC address to the same value as used by the host interface (the value assigned by your cloud provider).
- Save your configuration.

Step 3: Configure the Solaris Guest to Use the Private Cloud IP Address

Using the steps below, the Solaris guest system is configured to use the second NIC configured on the host system (please refer to your Solaris documentation for configuration details).

1. Boot Solaris and configure the IP address assigned to the dedicated guest NIC for the Solaris Ethernet interface as shown in the examples below:
   # ifconfig *<interface-name> <private-guest-nic-ip>/<netmask* **up** (Solaris 10 example)
   or
   # ifconfig *<interface-name> <private-guest-nic-ip>* **netmask***<mask>* **up** (Solaris 2.6 example)
   or
   # ipadm create-ip net*X* and **ipadm create-addr -T static -a** *<private-guest-nic-ip>/<netmask>* **net***X***/v4** (Solaris 11 example)
   For Solaris versions before version 11, 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.

2. Add default route on Solaris:
   # route add default *<default-gateway-of-cloud-lan> <metric>*
   Make permanent by editing **/etc/defaultrouter** and add the address of the gateway (use route -p for newer Solaris versions).

3. Add DNS server to Solaris (if needed)

   a. Edit **/etc/resolv.conf** and add a nameserver line for the DNS server.
   b. Make sure, DNS is used for hostname translation: ensure that **/etc/nsswitch.conf** is configured to allow **dns** (in addition to **files**) for the hostname resolution.

   For Solaris 11, please refer to the Oracle Solaris documentation.

Expected result (depending on security rules and firewalls):

1. The guest system should be able to communicate with the host system across the cloud LAN using the private IP addresses.
2. The guest system should be able to communicate directly with the Internet if the dedicated NIC has a public IP address (not recommended).

Please note: Do not forget that traffic transmitted across the Internet by the guest system is not encrypted by default. Take appropriate measures to protect your data. It is strongly recommended to protect the Solaris guest system by an appropriate firewall and security group configuration. If possible, any communication across the Internet should be encrypted (e.g., by using a VPN).

Interface MTU Considerations

Contents

- Interface MTU Introduction
- Fragmentation
  - Possible Problems with IPv4 Fragmentation
    - Performance Problems
    - Connectivity Problems
- Implications for Charon Emulators and the Legacy Guest Systems

Interface MTU Introduction

The MTU (Maximum Transfer Unit) is the maximum packet size that can be transmitted across an interface. The MTU of an interface depends on the type of interface. This section only applies to Ethernet interfaces. The MTU value includes the IP and additional (e.g., TCP) protocol headers.

TCP/IP example for an Ethernet interface for which the default MTU is 1500:

```
1460 byte TCP application payload (Transport
layer)

+ 20 byte TCP header (Transport layer)

+ 20 byte IPv4 header (Routing layer)

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

= 1500 byte MTU (Ethernet payload)

+ 14 byte Ethernet frame header (datalink layer)

+ 4 byte Frame Check Sequence (FCS)

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

= 1518 byte Ethernet frame
```

The maximum size of the payload data varies depending on the transport layer protocol used. It can also be smaller as in the example above, for example, if a VPN is used which requires its own protocol informatin within the packet. In case of TCP/IP, the actual maximum payload size is also called MSS (Maximum Segment Size).

Fragmentation

The IP protocol includes the option to split up data packets into smaller packets (fragments) that are too big for the MTU of an interface. The recipient of the packet is responsible for reassembling the packets.

Possible Problems with IPv4 Fragmentation

The need to fragment IP packets can cause several problems. Some common problems are described below.

Performance Problems

Excessive fragmentation and reassembly causes additional network load due to more packets being transmitted. It also causes additional CPU load on systems that are required to fragment and reassemble packets. In addition, the loss of one fragment causes the retransmission of the whole packet. So in case of network instability, additional load is added to an already unstable network connection.

Connectivity Problems

If there are intermediate systems between two communication partners that have smaller MTUs than the two communication partners, data transmission may fail, for example in the following cases:

- If the intermediate systems do not allow fragmentation or block packet fragments.
- If the intermediate systems block the ICMP messages necessary for Path MTU Discovery.

For a TCP connections, the two communication partners come to an agreement regarding the MSS to be used based on their own interface MTU. If any intermediate systems have a smaller MTU but either the two communication partners don't know this (failed Path MTU Discovery) and the intermediate systems cannot fragment the data packets, any packet that is too large will be dropped.

For protocols without the ability to negotiate the payload size or cannot handle fragmented traffic, data will be lost as soon as a packet is larger than the smallest MTU on the path.

Example with two ICMP data sizes across a DSL connection that is limited to a MTU of 1492. The IP + ICMP headers are 28 bytes in total. In the example, fragmentation is disabled (`-M do`). The source system has an MTU of 1500 bytes.

```
$ ping -c 1 -s $((1493-28)) -M do www.stromasys.com
PING stromasys.com (192.124.249.190) 1465(1493) bytes of data.

--- stromasys.com ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms


$ ping -c 1 -s $((1492-28)) -M do www.stromasys.com
PING stromasys.com (192.124.249.190) 1464(1492) bytes of data.
1472 bytes from cloudproxy10190.sucuri.net (192.124.249.190): icmp_seq=1 ttl=58 time=21.6 ms
```

The first attempt with an attempted size of 1493 bytes fails, the second with a packet size of 1492 succeeds.

Implications for Charon Emulators and the Legacy Guest Systems

When dedicating a Charon host NIC to the emulator for use by the legacy guest system, **ensure that the MTU of the dedicated NIC is not smaller than the MTU interface visible in the guest system**.

Typically, legacy guest systems, such as Solaris, HP-UX, Tru64, and OpenVMS have a default MTU of 1500 bytes.

Physical Charon hosts usually also have a default MTU of 1500 bytes.

However, in cloud environments, the default MTU of Ethernet interfaces configured for a Charon host can vary. The following table provides an overview:

Platform	Default MTU size	Comment
Legacy OS	1500	
AWS	9001	Jumbo frames are not supported by Charon emulators (Charon-SSP has implemented a workaround). Therefore, all end-systems involved in the communication with the legacy guest system should use an MTU no larger than 1500.
Azure	1500	
GCP	1460	A different MTU size can be defined when creating the VPC to be used for the instance.
OCI	9000	Jumbo frames are not supported by Charon emulators (Charon-SSP has implemented a workaround). Therefore, all end-systems involved in the communication with the legacy guest system should use an MTU no larger than 1500.
IBM	1500	

Caveat: ensuring that the Charon host MTU is not smaller than the guest system MTU prevents problems arising at the emulator host. However, it cannot prevent problems arising from smaller MTU sizes along the complete communication path if fragmentation does not work properly. Therefore, it is recommended to identify the actual MTU, for example, of the path across a VPN between cloud and on-premises networks and to either ensure that fragmentation and path discovery work correctly, or adapt the MTU of the legacy guest system accordingly.

Next Steps

Once you have set up your Charon-SSP instance in the cloud, please proceed to the general *Charon-SSP User's Guide* for your Charon-SSP version (see CHARON-SSP for Linux) and the *VE License Server User's Guide* in Licensing Documentation for more information about configuring and managing Charon-SSP.