



Charon-SSP 4.2 for Google Cloud Getting Started Guide

Current product version: 4.2.7

Contents

- About this Guide 3
- Introduction to Charon-SSP 6
- Virtual Hardware and Guest OS Supported by Charon-SSP for Cloud 7
- Setting up a GCP Cloud Instance for Charon-SSP 8
- Installing the VE License Server Software 22
- Installing the Charon-SSP Manager 23
- Accessing the Charon-SSP GCP Instance 27
 - SSH Command-Line Access 29
 - SFTP File Transfer 31
 - Connecting with the Charon-SSP Manager 32
- Additional Charon-SSP GCP Instance Configuration 34
 - Storage Management 35
 - Network Management 42
 - Charon-SSP Cloud Networking 51
 - SSH VPN - Connecting Charon Host and Guest to Customer Network 54
 - Dedicated NIC for Guest System 61
- Next Steps 64

About this Guide

Contents

- [Intended Audience](#)
- [Product Overview](#)
- [Document Structure](#)
- [Obtaining Documentation](#)
- [Obtaining Technical Assistance or General Product Information](#)
 - [Obtaining Technical Assistance](#)
 - [Obtaining General Product Information](#)
- [Conventions](#)
- [Definitions](#)
- [Related Documents](#)

Intended Audience

This Getting Started guide is intended for anyone who needs to install, configure, or manage the Stromasys Charon-SSP processor/platform virtualization software in the Google cloud (GCP). Its main focus is on installations that use the prepackaged Charon-SSP images available on the GCP 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 GCP. It is supplemented by the general [Charon-SSP User's Guide](#) and the [VE License Server User's Guide](#).

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.GCP@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. **Depending on the cloud environment, the availability of cloud-specific options may differ.** The **typical cloud-specific options** are:

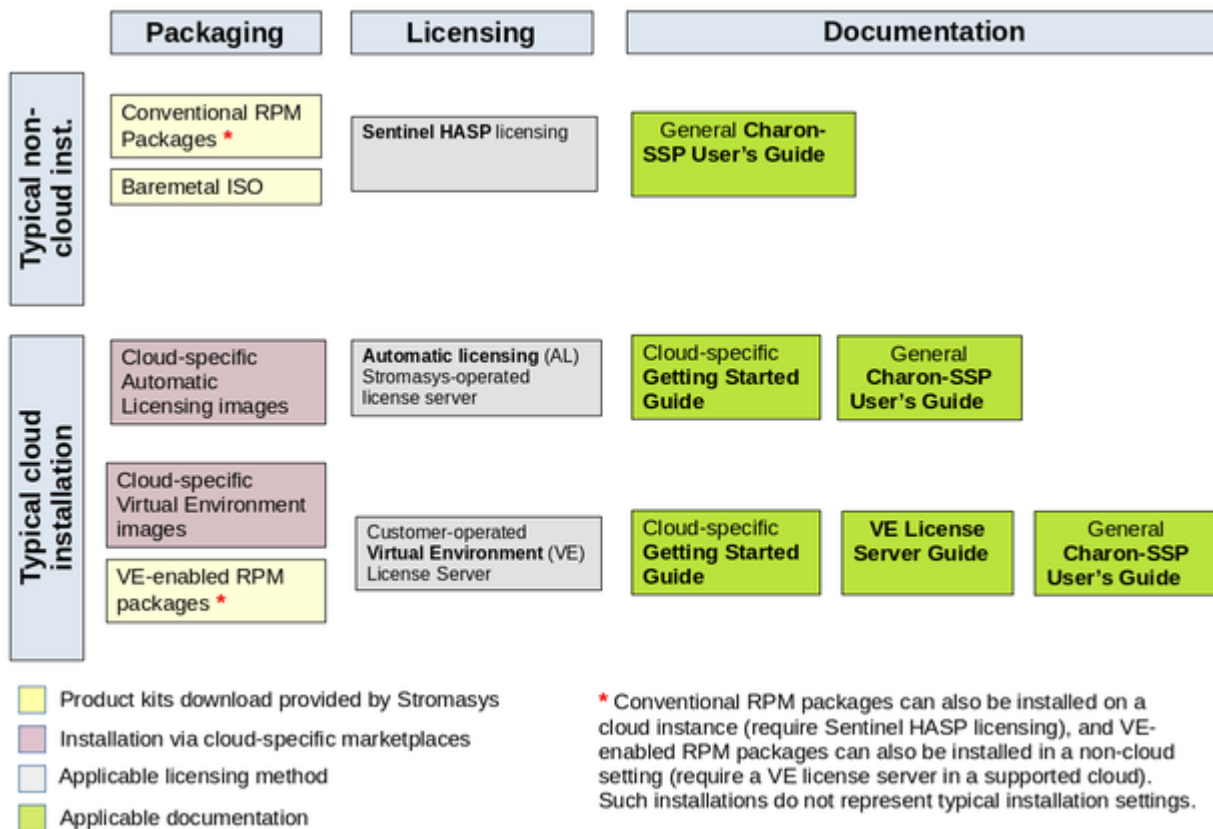
1. **Prepackaged** images provided on the cloud-specific marketplace:
 - a. Cloud-specific Charon-SSP AL (Automatic Licensing) image using a public, Stromasys-operated, cloud-specific license server (license created automatically at first instance launch).
 - b. Cloud-specific Charon-SSP VE (Virtual 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 Guide](#) covers **features, configuration, and management of the Charon-SSP products**.
- The [VE License Server User's Guide](#) 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 image provides an overview of **Charon-SSP packaging in cloud and non-cloud environments**, the associated licensing, and the applicable product documentation:



Availability of cloud-specific Charon-SSP options in the Google cloud environment (as planned at the time of writing):

- On the GCP marketplace: cloud-specific Charon-SSP VE (Virtual Environment) image utilizing a customer-operated, private VE license server.
- Installation of Charon-SSP for Virtual Environments (VE) on a Linux server in the cloud using RPM packages provided by Stromasys.

Document Structure

The document contains the following main sections:

- [Introduction to Charon-SSP](#): overview of emulator concepts.
- [Virtual Hardware and Guest OS Supported by Charon-SSP for Cloud](#): list of supported virtual hardware and supported guest operating systems.
- [Setting up a GCP Cloud 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-SSP 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 GCP 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 GCP Instance Configuration](#): steps to add additional storage and network interfaces; introduction to 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 Stomasys 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 Stomasys, please visit <http://www.stomasys.com/support/> for up-to-date support telephone numbers and business hours. Alternatively, the support center is available via email at support@stomasys.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 Stomasys [Product Documentation and Knowledge Base](#) and on the [Stomasys web site](#) you can contact the Stomasys team using <https://www.stomasys.com/contact/>, or by sending an email to info@stomasys.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	2840 Plaza Place, Ste 450 Raleigh, NC 27612 U.S.A.
Europe, Middle-East and Africa	+41 22 794 1070	Avenue Louis-Casai 84 5th 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.
<i>disk0</i>	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 Guide and Release Notes](#)
- [VE License Server User's Guide](#)

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:

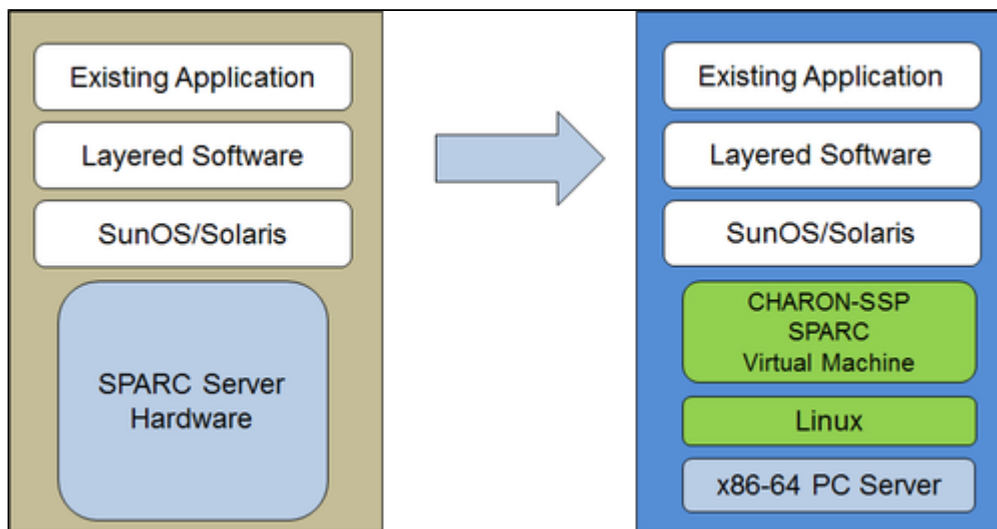
- **Sun-4u family (represented by the Sun Enterprise 450):** (U for [UltraSPARC](#)) – this variant introduced the [64-bit SPARC V9](#) processor architecture and UPA processor interconnect first used in the [Sun Ultra series](#). Supported by 32-bit versions of Solaris starting from version 2.5.1. The 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):** A variation on Sun-4u which includes hypervisor processor virtualization; introduced in the UltraSPARC T1 multicore processor. Selected hardware was supported by Solaris version 10 starting from release 3/05 HW2 (most models - including the hardware emulated by Charon-SSP - require newer versions of Solaris 10). Several Solaris 11 versions are also supported.

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 VT-x/EPT hardware assisted virtualization technology in modern Intel CPUs to offer end users better virtual CPU performance. Charon-SSP/4U+ and Charon-SSP/4V+ require Intel CPUs with VT-x/EPT capability (currently only experimental AMD support) and **must** be installed on a dedicated host system. Running these product variants in a VM 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.

Virtual Hardware and Guest OS Supported by Charon-SSP for Cloud

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
Max. RAM	64MB to 512MB	1GB to 128GB	1GB to 1024GB ⁽⁵⁾
Ethernet controllers	2 (controller type le)	19 (controller types hme and qfe)	4 (controller types bge and qfe)
SCSI controllers	1	2	2
SCSI target IDs	7 ⁽³⁾	30 ⁽³⁾	30 ⁽³⁾
Serial ports	2	2	2 + Vconsole
Graphics controllers	1 (CGTHREE or CGSIX ⁽⁶⁾)	1 (CGSIX or RAGE XL)	
Audio controllers	1 (DBRle)	1 (DBRle)	

⁽¹⁾ Charon-SSP/4U+ has the same virtual hardware specification as Charon-SSP/4U, Charon-SSP/4V+ the same as Charon-SSP/4V. Charon-SSP/4U+ and Charon-SSP/4V+ are only supported on physical Intel hardware (VT-x support) and with Linux kernels provided by Stromasys. AMD support for Charon-SSP/4U+ and Charon-SSP/4V+ is currently only experimental.

⁽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 without modifications.

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

⁽⁶⁾ CGSIX emulation is not supported for SunOS 4.x guest systems.

Supported Guest Operating Systems

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

- SunOS 4.1.3 - 4.1.4
- Solaris 2.3 to Solaris 9

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

- Solaris 2.5.1 to Solaris 10

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

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

Setting up a GCP Cloud Instance for Charon-SSP

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

Contents

- Prerequisites
 - General Prerequisites
 - Licensing
 - Charon-SSP Automatic Licensing Overview
 - Charon-SSP VE Licensing Overview
 - Charon-SSP VE license characteristics
 - Charon-SSP VE License Server Communication Requirements
 - Basic License Installation Steps Before an Emulator Can be Started
 - GCP Machine Type Prerequisites (Hardware Prerequisites)
- GCP Login and New Instance Launch
 - Logging in to GCP
- Preparation
 - Select or Create Project
 - Create VPCs and Subnets for Instance
- Creating a New VM Instance

Prerequisites

General Prerequisites

To install and configure Charon-SSP in the Google cloud, you need an account on the Google cloud platform.

Licensing

Charon-SSP requires a license to run emulated SPARC systems. For a typical cloud-based installation, there are two different Charon-SSP product variants with two different licensing models (**availability may differ depending on cloud environment**):

1. The cloud-specific, prepackaged Charon-SSP AL (Automatic Licensing) image utilizing a public, Stromasys-operated cloud-specific license server.
2. Charon-SSP VE (Virtual Environment) utilizing a customer-operated, private VE license server in a supported cloud environment. Charon-SSP VE is available as a prepackaged image on some cloud platforms, and in RPM package format for a conventional installation.

Both licensing options are briefly described below. Please contact your Stromasys representative for any questions about product availability and licensing options.

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

Charon-SSP Automatic Licensing Overview

Not available on Google cloud. This section is for information only.

Charon-SSP AL images with automatic licensing use a specialized Charon-SSP environment. They require a license to run emulated SPARC systems. This license is created automatically upon first launch of the Charon-SSP instance. Please note the following points:

- The Charon-SSP instance requires Internet access (via 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-environment (the source address must be from the address range of the same cloud provider in which the Charon-SSP host instance runs). The public, Stromasys-operated license servers must be reachable on port 8080. 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 will be provided by Stromasys for platforms supporting the Charon-SSP AL images.
- 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 data volume that can easily be detached from the old instance and attached to a new instance.
- Should access to the license be lost, there is a grace period of 24 hours. If license access is not restored within this period, the emulator will stop (if a guest system is running at the time, this is the equivalent of disconnecting the power without clean shutdown, i.e., it may lead to loss of data).

Charon-SSP VE Licensing Overview

This licensing option is applicable to prepackaged Charon-SSP VE images on cloud marketplaces and to VE-capable Charon-SSP emulator software installed from RPM packages.

Charon-SSP VE license characteristics

The main characteristics of VE licenses are the following:

- Software licenses only.
- Installed on Charon-SSP host or separate license server.
- Require the Charon-SSP VE license server software (RPM package included in the prepackaged, cloud-specific marketplace Charon-SSP VE image).
- Require matching Charon-SSP emulator software (preinstalled on the prepackaged, cloud-specific marketplace Charon-SSP VE image).

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.

Charon-SSP VE License Server Communication Requirements

For proper functionality, the system 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**)
- The host **www.googleapis.com**

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

- **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.

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-SSP 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 [VE License Server User's Guide](#) for more information.

GCP Machine Type Prerequisites (Hardware Prerequisites)

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

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.

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). 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 required).

General CPU requirements: Charon-SSP requires modern x86-64 architecture processors with a recommended CPU frequency of at least 3.0GHz.

Minimum requirements for Charon-SSP:

- Minimum number of host system CPU cores:
 - At least one CPU core for the host operating system.
 - **For each emulated SPARC system:**
 - One CPU core for each emulated CPU of the instance.
 - At least one additional CPU core for I/O processing (at least two, if server JIT optimization is used). See the *CPU Configuration* section mentioned above for default allocation and configuration options
- Minimum memory requirements:
 - At least 2GB of RAM for the host operating system.
 - **For each emulated SPARC system:**
 - The configured memory of the emulated instance.
 - 2GB of RAM (6GB of RAM if server JIT is used) to allow for DIT optimization, emulator requirements, run-time buffers, SMP and graphics emulation.
- If hyper-threading cannot be disabled on the Charon-SSP host, configure the hyper-threading option in the Charon-SSP Manager. See the *CPU Configuration* section mentioned above for additional configuration information.
- One or more network interfaces, depending on customer requirements.
- Charon-SSP/4U+ and Charon-SSP/4V+ must run on **physical** Intel hardware supporting VT-x/EPT (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:
 - The support of these product variants on AMD processors (AMD-v/NPT required) is currently experimental.
 - Charon-SSP/4U+ and Charon-SSP/4V+ are only available when using the Linux kernels provided by Stromasys.
 - Please contact Stromasys or your Stromasys VAR if you need this type of emulated SPARC hardware to discuss your requirements in detail.

Please note:

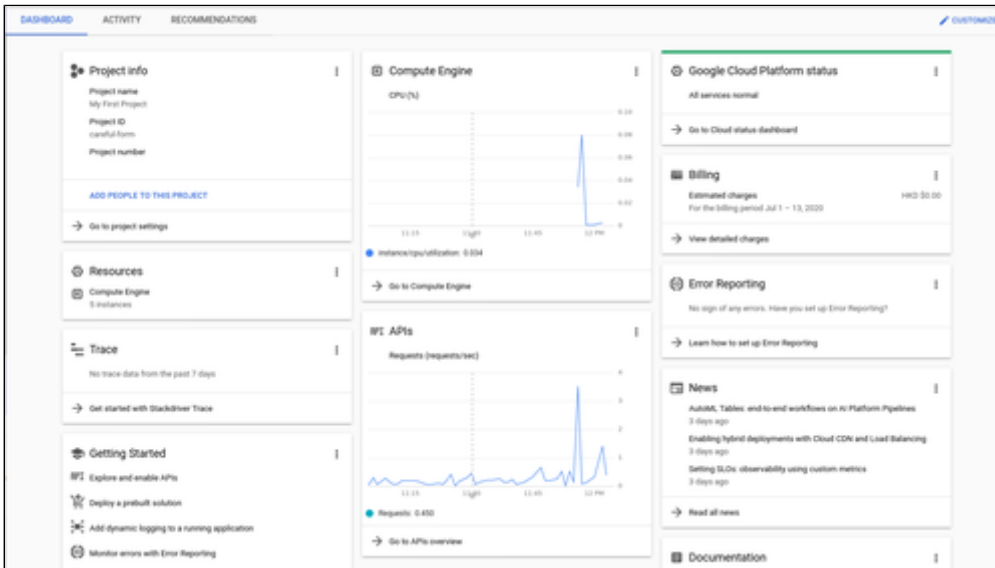
- The sizing guidelines above—in particular regarding number of host CPU cores and host memory—show the **minimum requirements**. Every use case has to be reviewed and the actual host sizing has to be adapted as necessary. For example, the number of I/O CPUs may have to be increased if the guest applications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

GCP Login and New Instance Launch

Logging in to GCP

To log in perform the following steps:

- Go to <https://console.cloud.google.com>. You will see the login screen.
- Enter your login credentials.
- Upon successful login, a Google cloud dashboard screen will be displayed similar to the example below:

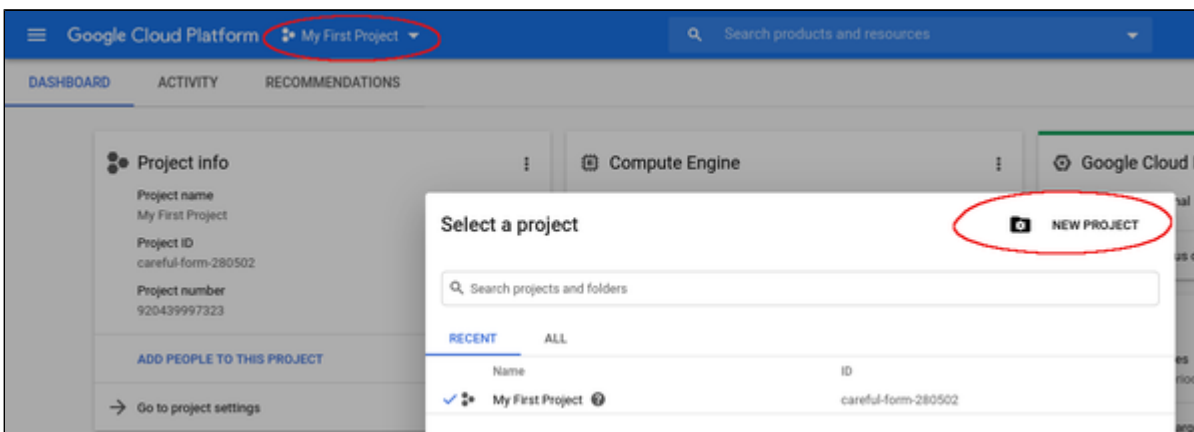


Preparation

Select or Create Project

A project organizes all your Google Cloud resources. To organize all resources for a certain application purpose, you can group them in their own project. So before you start creating resources, select or create the appropriate project.

To select or create a project, select the project list from the top of the Google cloud console window, as shown below:



Either select the correct project or create a new one by clicking on the **NEW PROJECT** button.

Create VPCs and Subnets for Instance

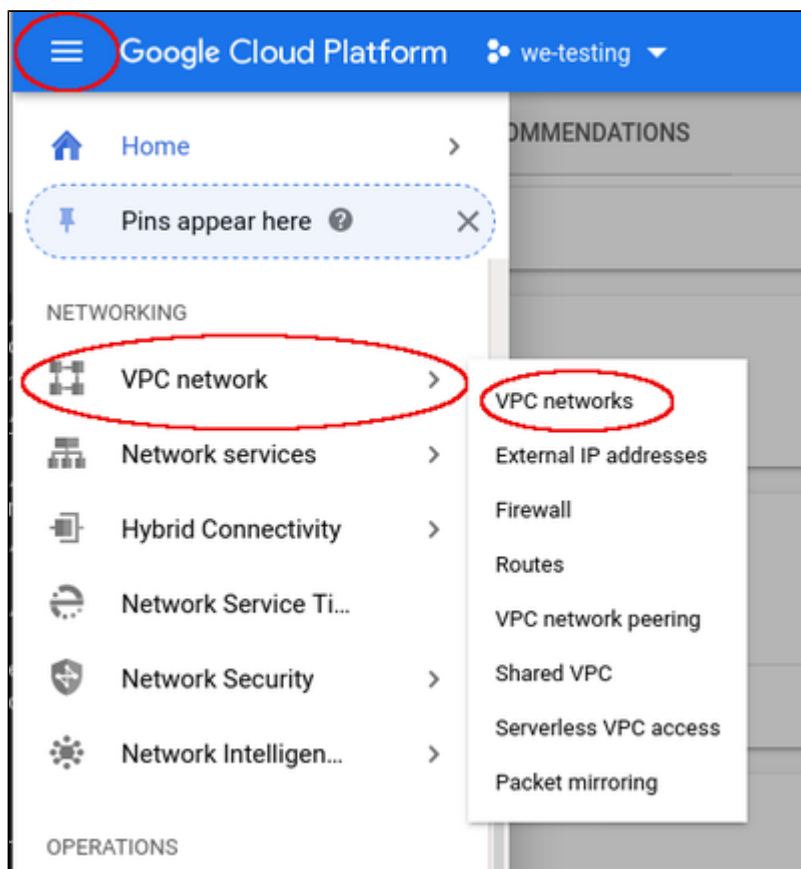
Important rules for Google cloud instances with respect to network interfaces:

- Interfaces can only be added during instance creation.
- Each network interface configured in a single instance must be attached to a different VPC network.
- The additional VPC networks that the multiple interfaces will attach to must exist before an instance is created. See [Using VPC Networks](#) for instructions on creating additional VPC networks.
- You cannot delete a network interface without deleting the instance.
- IP forwarding can only be enabled when the instance is created.
- The VPC network has a maximum transmission unit (MTU) of 1460 bytes for Linux images and Windows Server images. Operating system images provided by Compute Engine are already configured with the appropriate MTU. For custom images, set the MTU to 1460 for custom Linux images and Windows Server images to avoid the increased latency and packet overhead caused by fragmentation.

Therefore the required VPCs and subnets must exist before the instance is created.

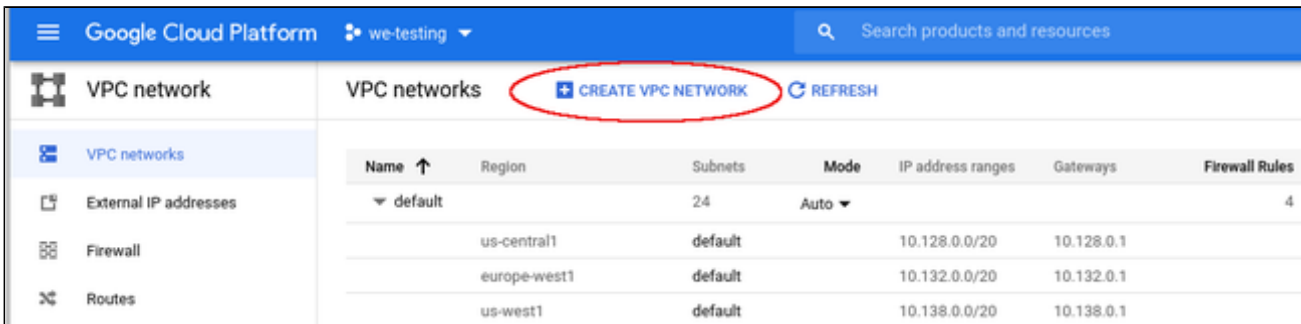
To create additional VPCs (if required), perform the following steps.

Step 1: Open the VPC network section by clicking on the Navigation menu, then selecting VPC network, and clicking on VPC networks - as illustrated below.



This will open the VPC overview page with the already existing VPCs. If all required VPCs and subnets already exist, continue with creating the new VM instance. Otherwise, continue with step 2.

Step 2: If you need to create a new VPC, click on **CREATE VPC NETWORK** at the top of the VPC overview list.



The screenshot shows the Google Cloud Platform interface for VPC networks. The top navigation bar includes 'Google Cloud Platform', the project name 'we-testing', and a search bar. The main content area is titled 'VPC networks' and features a '+ CREATE VPC NETWORK' button circled in red, along with a 'REFRESH' button. Below this is a table listing VPC networks:

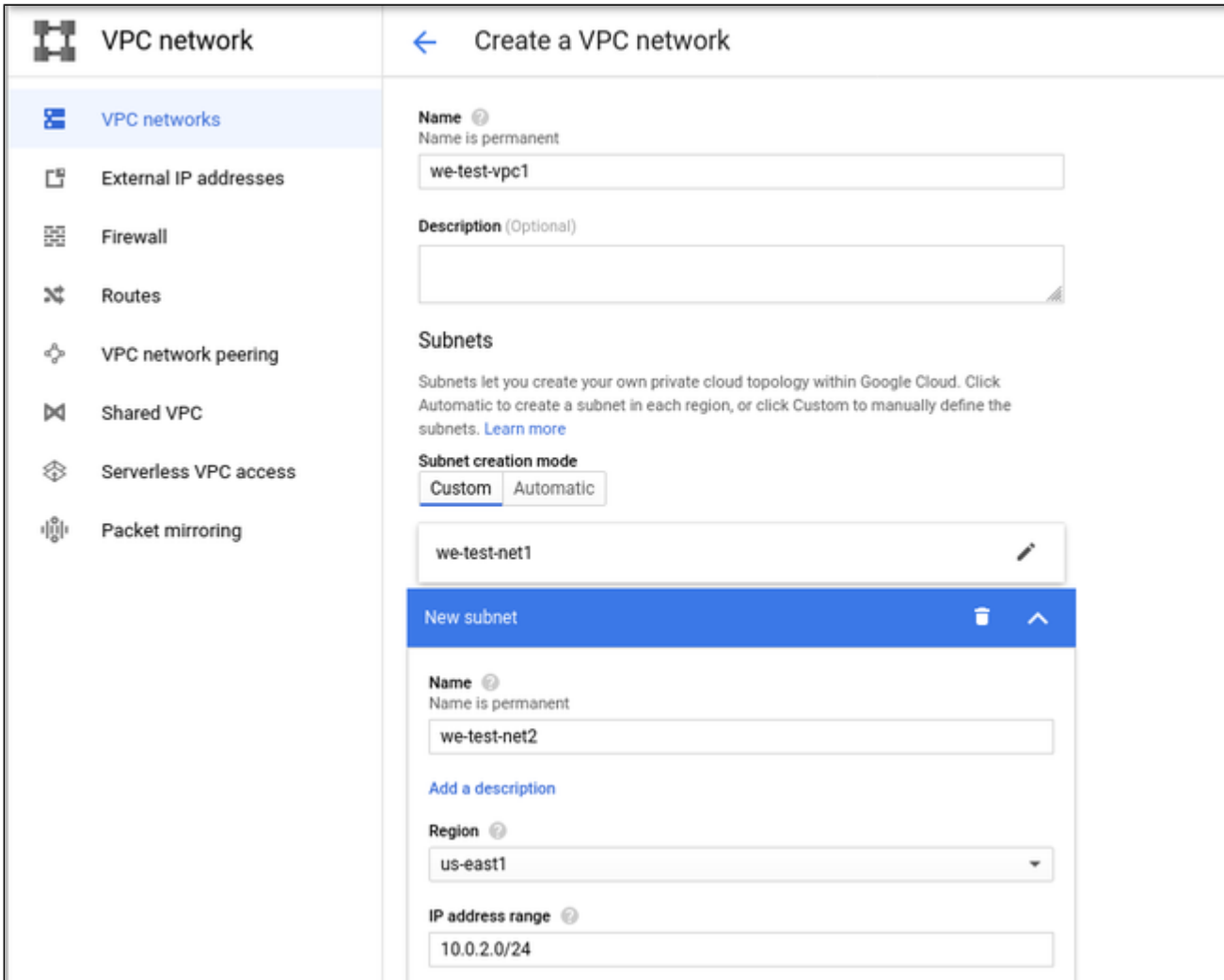
Name ↑	Region	Subnets	Mode	IP address ranges	Gateways	Firewall Rules
▼ default		24	Auto ▼			4
	us-central1	default		10.128.0.0/20	10.128.0.1	
	europa-west1	default		10.132.0.0/20	10.132.0.1	
	us-west1	default		10.138.0.0/20	10.138.0.1	

This opens the VPC configuration window.

Step 3: Create VPC and subnets.

In the VPC configuration window, enter

- the VPC name, and
- the subnet name, region and address.



The screenshot shows the 'Create a VPC network' configuration window. The left sidebar contains navigation options: VPC networks, External IP addresses, Firewall, Routes, VPC network peering, Shared VPC, Serverless VPC access, and Packet mirroring. The main configuration area includes:

- Name:** we-test-vpc1 (Note: Name is permanent)
- Description (Optional):** (Empty text area)
- Subnets:** Subnets let you create your own private cloud topology within Google Cloud. Click Automatic to create a subnet in each region, or click Custom to manually define the subnets. [Learn more](#)
- Subnet creation mode:** Custom (selected) / Automatic
- Subnet Name:** we-test-net1
- New subnet configuration:**
 - Name:** we-test-net2 (Note: Name is permanent)
 - Add a description:** (Link)
 - Region:** us-east1
 - IP address range:** 10.0.2.0/24

Click on **Create** at the bottom of the window to create the VPC.

The new VPC should appear in the VPC overview list. Selecting the VPC in the overview list will open the detail information window. Example:

The screenshot shows the 'VPC network details' page for a VPC named 'we-test-vpc1'. The left sidebar contains navigation options: VPC networks, External IP addresses, Firewall, Routes, VPC network peering, Shared VPC, Serverless VPC access, and Packet mirroring. The main content area shows the VPC name and configuration details: Subnet creation mode (Custom subnets), Dynamic routing mode (Regional), and DNS server policy (None). Below this, there are tabs for Subnets, Static internal IP addresses, Firewall rules, Routes, VPC Network Peering, and Private service connection. The 'Subnets' tab is active, showing a table of subnets:

<input type="checkbox"/>	Name ^	Region	IP address ranges	Gateway	Private Google access	Flow logs	
<input type="checkbox"/>	we-test-net1	us-east1	10.0.1.0/24	10.0.1.1	Off	Off	
<input type="checkbox"/>	we-test-net2	us-east1	10.0.2.0/24	10.0.2.1	Off	Off	

At the bottom, there is a link for 'Equivalent REST'.

Step 4: Create firewall rules for the VPC.

With the detail information open, click on Firewall. This will allow you to define the required firewall rules for the VPC.

An example of a small set of firewall rules that allow incoming SSH and ICMP is shown below:

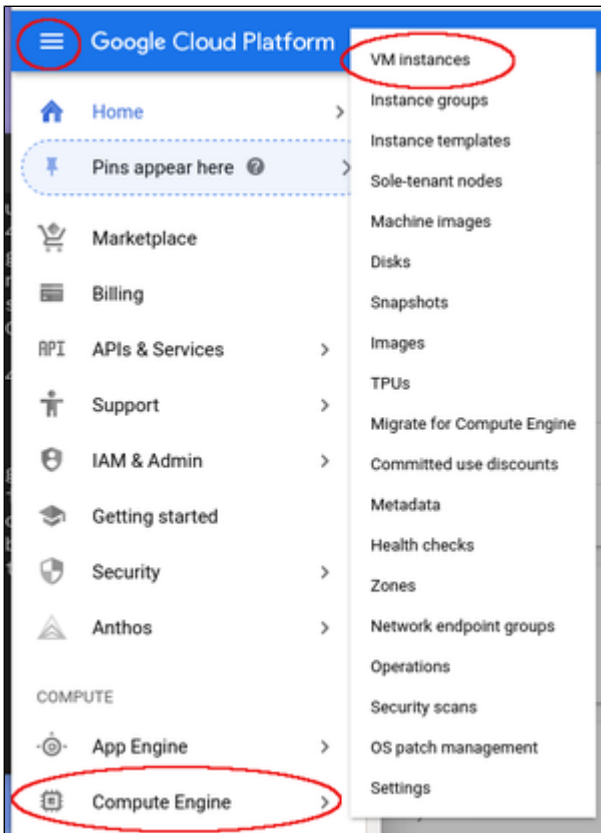
The screenshot shows the 'VPC network details' page for 'we-test-vpc1' with the 'Firewall rules' tab selected. The configuration details (Subnet creation mode, Dynamic routing mode, DNS server policy) are the same as in the previous screenshot. The 'Firewall rules' tab shows a table of firewall rules:

<input type="checkbox"/>	Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Logs	Hit count	Last hit
<input type="checkbox"/>	icmp-any	Ingress	Apply to all	IP ranges: 0.0.0.0/24	icmp	Allow	1000	Off	--	--
<input type="checkbox"/>	ssh-any	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	1000	Off	--	--

Creating a New VM Instance

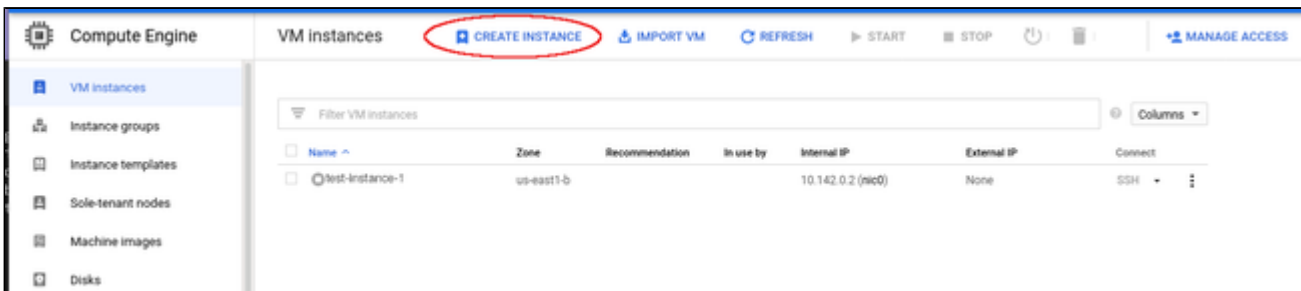
Step 1: Go to the VM instance overview page.

Open the Navigation menu, click on Compute Engine and then on VM Instances as illustrated below:



This will open the list of already existing VM instances.

Step 2: Click on **CREATE INSTANCE** at the top of the overview list.



This will open the VM creation window as shown below.

Step 3: Configure the basic information of your new VM instance.

In the main configuration window set the following information at a minimum:

- **Name** of the instance (permanent setting)
- Correct **Machine family** and **Machine type** to match the Charon-SSP host and guest requirements
- **Boot disk** type and size, and the image to use as the operating system (recommended minimum system disk size: 30GB). To change the image for Charon-SSP, press the **Change** button and select the correct image. If installing a prepackaged marketplace Charon-SSP image, this image must be used. If you plan to install Charon-SSP using RPM packages, use a Linux version supported for Charon-SSP.

The following image illustrates the basic settings:

Create an instance

To create a VM instance, select one of the options:

- New VM instance** (Create a single VM instance from scratch)
- New VM instance from template** (Create a single VM instance from an existing template)
- New VM instance from machine image** (Create a single VM instance from an existing machine image)
- Marketplace** (Deploy a ready-to-go solution onto a VM instance)

Name (Name is permanent)
we-test-1

Labels (Optional)
name : we-testing
[+ Add label](#)

Region (Region is permanent): us-central1 (Iowa)
Zone (Zone is permanent): us-central1-a

Machine configuration

Machine family
General-purpose | Memory-optimized | Compute-optimized
Machine types for common workloads, optimized for cost and flexibility

Series
N1
Powered by Intel Skylake CPU platform or one of its predecessors

Machine type
n1-standard-2 (2 vCPU, 7.5 GB memory)

	vCPU	Memory
	2	7.5 GB

CPU platform and GPU

Container
 Deploy a container image to this VM instance. [Learn more](#)

Boot disk
New 20 GB standard persistent disk
Image: charon-ssp-v4-1-26-ve-build1 [Change](#)

Identity and API access
Service account: Compute Engine default service account

Step 4: Add your SSH key for remote access to the cloud instance.

Open the advanced settings at the bottom of the VM creation window by clicking on **Management, security, disks,...** at the bottom of the page:

Identity and API access ?

Service account ?

Compute Engine default service account

Access scopes ?

Allow default access

Allow full access to all Cloud APIs

Set access for each API

Firewall ?

Add tags and firewall rules to allow specific network traffic from the Internet

Allow HTTP traffic

Allow HTTPS traffic

Management, security, disks, networking, sole tenancy

The advanced settings allow you to create and add disks and network interfaces during the creation of a VM.

Please note: network interfaces can only be added during the creation of a VM instance.

The advanced settings also allow you to add your public SSH key for accessing the VM once started. To do this,

- select the tab **Security** in the advanced settings section,
- paste your **public key** into the field provided (the username extracted from the key will be displayed).

Management **Security** Disks Networking Sole Tenancy

Shielded VM ?

Turn on all settings for the most secure configuration.

Turn on Secure Boot ?

Turn on vTPM ?

Turn on Integrity Monitoring ?

SSH Keys

These keys allow access only to this instance, unlike [project-wide SSH keys](#) [Learn more](#)

Block project-wide SSH keys

When checked, project-wide SSH keys cannot access this instance [Learn more](#)

Enter public SSH key

+ Add item

Less

You can collapse the section again by clicking on **Less**.

Step 5: Optionally, configure additional NICs and/or IP forwarding

To add an **additional network interface**, perform the following steps:

- Open the advanced settings at the bottom of the VM creation window by clicking on **Management, security, disks,...** at the bottom of the page.
- Select Networking from the advanced settings section.
- Click on **Add network interface**.
- Select the correct subnet.
- Set the information about internal and external IP address (static or ephemeral) as required.

The screenshot shows the 'Networking' tab in the GCP VM creation interface. At the top, there are tabs for 'Management', 'Security', 'Disks', 'Networking' (selected), and 'Sole Tenancy'. Below these, there are sections for 'Network tags (Optional)', 'Hostname', and 'Network interfaces'. A modal dialog titled 'Network interface' is open, showing the following configuration:

- Network:** we-test-vpc1
- Subnetwork:** we-test-net1 (10.0.1.0/24)
- Primary internal IP:** Ephemeral (Automatic)
- External IP:** Ephemeral
- Network Service Tier:** Premium (Current project-level tier, change) (selected), Standard (us-east1)

Buttons for 'Done' and 'Cancel' are visible at the bottom of the dialog.

After adding all the required information, click on **Done**.

To enable **IP forwarding**, perform the following steps:

- Open the advanced settings at the bottom of the VM creation window by clicking on **Management, security, disks,...** at the bottom of the page.
- Select Networking from the advanced settings section.
- Select the edit option for the default NIC.
- Enable IP forwarding
- Click on **Done**.

Please note: you have to set up a firewall manually when you add additional network interfaces. See [Network Management](#) and the GCP documentation for more detail.

Step 6: Create the VM.

Once you filled in all the required data, create the VM by pressing the **Create** button at the bottom of the page:

Create an instance

Deploy a ready-to-go solution onto a VM instance

Machine type

n1-standard-2 (2 vCPU, 7.5 GB memory)

	vCPU	Memory
	2	7.5 GB

⌵ CPU platform and GPU

Container ?

Deploy a container image to this VM instance. [Learn more](#)

Boot disk ?

New 20 GB standard persistent disk
Image
charon-ssp-v4-1-26-ve-build1 [Change](#)

Identity and API access ?

Service account ?

Compute Engine default service account

Access scopes ?

Allow default access
 Allow full access to all Cloud APIs
 Set access for each API

Firewall ?

Add tags and firewall rules to allow specific network traffic from the Internet

Allow HTTP traffic
 Allow HTTPS traffic

⌵ Management, security, disks, networking, sole tenancy

The following options have been customized:

Labels
SSH keys

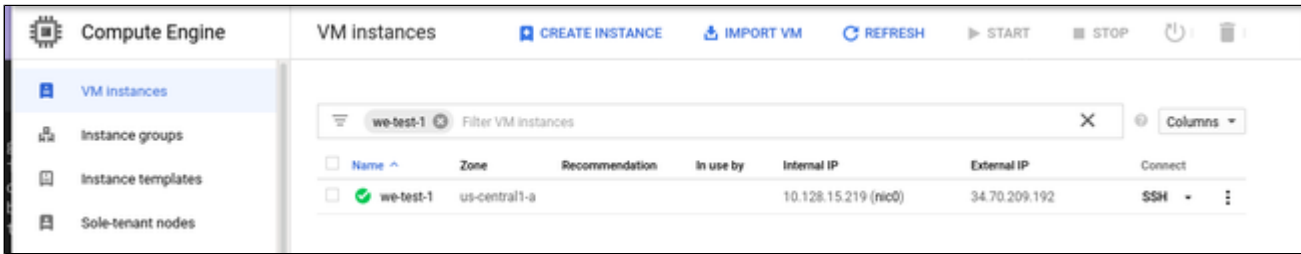
You will be billed for this instance. [Compute Engine pricing](#) [↗](#)

Create Cancel

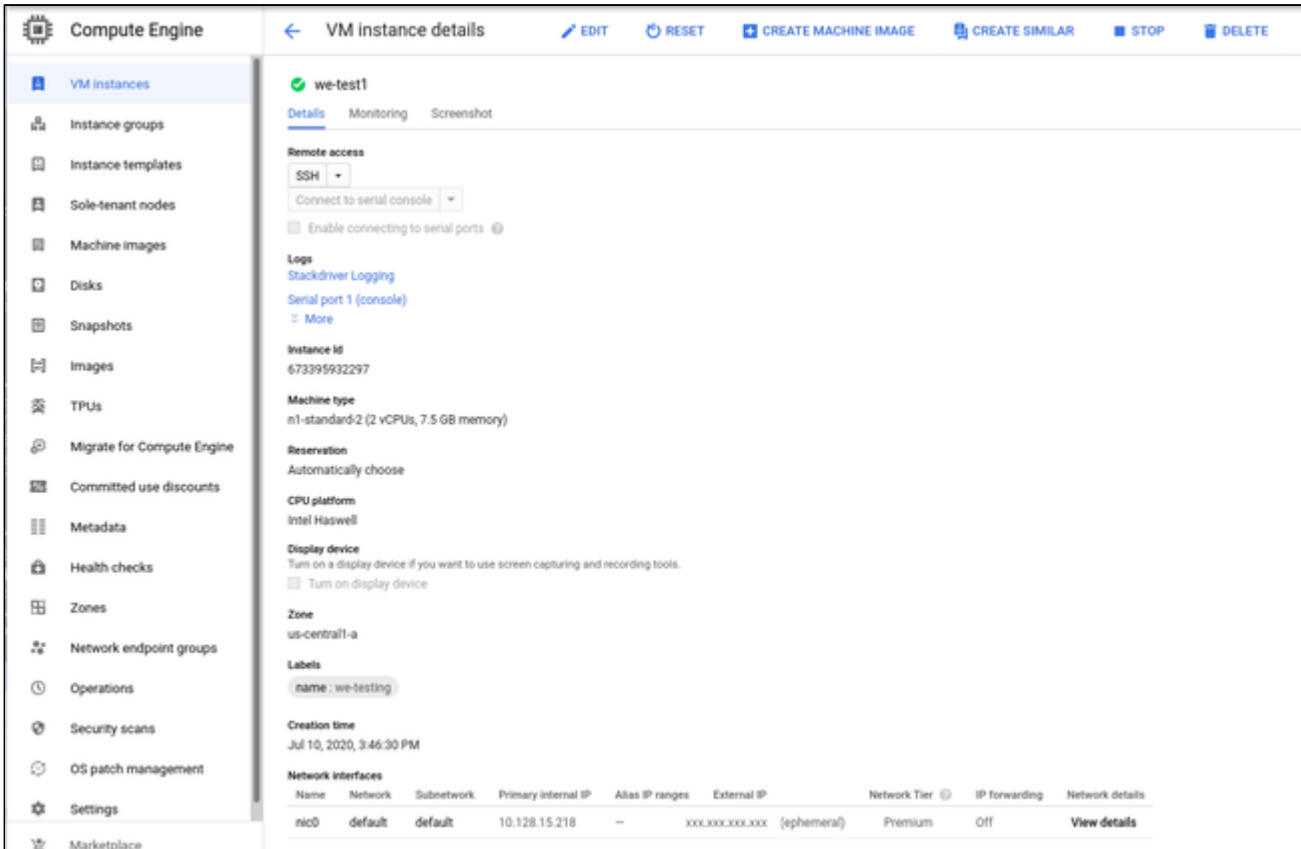
This will create the VM, start it and show it in the VM instances list.

Step 7: Verify the settings of the newly created cloud instance.

After successful creation, the new instance will be shown in the VM instances list:



By clicking on it, you will see the details of the cloud instance, as shown in the example below:



Installing the VE License Server Software

Please note: this section is only applicable to Charon-SSP VE cloud images where the included Charon-SSP VE License Server kit is to be installed on the Charon host system itself. For more information, and if you installed your host system using the Charon-SSP VE RPM packages, please refer to the [Charon-SSP VE License Server Guide](#).

Charon-SSP for Virtual Environments (VE) requires at least one VE license server on the Charon host system itself or an a separate license server. The VE license server must run in a supported cloud environment. The software package is included in the Charon-SSP VE cloud image. The installation of the VE License Server package on the Charon host system is described below:

Step	Description
1	Log into your cloud instance and become the root user on the Charon host in the cloud using the following commands: <pre>\$ ssh -i <path-to-private-key> sshuser@<ip-address-of-cloud-instance> \$ sudo -i</pre>
2	Go to the directory where the package has been stored (this location is applicable to the prepackaged Charon-SSP VE marketplace image): <pre># cd /charon/storage/</pre>
3	Install the package: <pre># yum install license-server-<version>.rpm</pre>

Sample installation:

```
# cd /charon/storage/

# yum install license-server-1.0.35.rpm
Loaded plugins: fastestmirror
Examining license-server-1.0.35.rpm: license-server-1.0.35-1.x86_64
Marking license-server-1.0.35.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package license-server.x86_64 0:1.0.35-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package            Arch      Version      Repository      Size
=====
Installing:
license-server     x86_64    1.0.35-1     /license-server-1.0.35    79 M

Transaction Summary
=====
Install 1 Package

Total size: 79 M
Installed size: 79 M
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : license-server-1.0.35-1.x86_64                1/1
Created symlink from /etc/systemd/system/multi-user.target.wants/licensed.service to /etc/systemd/system/licensed.service.
  Verifying  : license-server-1.0.35-1.x86_64                1/1

Installed:
  license-server.x86_64 0:1.0.35-1

Complete!
```

Installing the Charon-SSP Manager

Contents

- [Overview](#)
- [Installation Packages](#)
- [Charon-Manager Installation on Linux](#)
 - [Prerequisites](#)
 - [Installation Steps on Linux](#)
- [Installation Steps on Microsoft Windows](#)

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, the Charon Manager is installed on a system on customer premises and used via 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)
 - Ubuntu 17 or higher (64-bit)
- **Microsoft Windows:** versions 7, 8, and 10

Installation Packages

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

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

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

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

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

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

b) For installation on the Charon host in the cloud if using a prepackaged marketplace image: the packages are located in the `/charon/storage/` directory.

c) For installation on a Charon host in the cloud where a conventional RPM installation was performed: the packages can be downloaded from a Stromasys server.

Charon-Manager Installation on Linux

Prerequisites

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** 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	<p>Installation on a Linux management system on customer premises (typical installation):</p> <ul style="list-style-type: none"> • 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 (as described above). <p>Installation on the Charon-SSP host system in the cloud (optional):</p> <ul style="list-style-type: none"> • Log in and become the root user on the Charon host in the cloud using the following commands: <code>\$ ssh -i <path-to-private-key> sshuser@<ip-address-of-cloud-instance></code> <code># sudo -i</code> • 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	<p>Go to the directory where the package has been stored:</p> <pre># cd <package-location></pre>
3	<p>Installing the package:</p> <p>For systems with RPM package management (Red Hat, CentOS, Oracle Linux):</p> <pre># yum install <filename-of-package></pre> <p>(For an installation on the cloud host system, check if xorg-x11-xauth is already installed if X11-Forwarding is planned.)</p> <p>For systems with Debian package management (Ubuntu):</p> <pre># dpkg -i <filename-of-package></pre>

Example (RPM installation with yum command recursively resolving package dependencies):

```

# yum install charon-manager-ssp*.rpm
Loaded plugins: fastestmirror, langpacks
Examining charon-manager-ssp-4.2.5.rpm: charon-manager-ssp-4.2.5-1.x86_64
Marking charon-manager-ssp-4.2.5.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package charon-manager-ssp.x86_64 0:4.2.5-1 will be installed

<lines removed>

Dependencies Resolved

=====
Package                Arch    Version      Repository      Size
=====
Installing:
 charon-manager-ssp     x86_64  4.2.5-1      /charon-manager-ssp-4.2.5  5.8 M
Installing for dependencies:

<lines removed>

Transaction Summary
=====
Install 1 Package (+42 Dependent packages)

Total size: 14 M
Total download size: 9.5 M
Installed size: 37 M
Is this ok [y/d/N]: y
Downloading packages:

< lines removed >

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction

< lines removed >

Installed:
 charon-manager-ssp.x86_64 0:4.2.5-1

Dependency Installed:
 atk.x86_64 0:2.28.1-1.e17
 cairo.x86_64 0:1.15.12-4.e17
 dejavu-fonts-common.noarch 0:2.33-6.e17
 dejavu-sans-fonts.noarch 0:2.33-6.e17

<lines removed>

 xorg-x11-server-utils.x86_64 0:7.7-20.e17

Complete!

```

Installation Steps on Microsoft Windows

The Charon-SSP Manager for Windows software is shipped as a zipped archive package. To complete the installation, use the following instructions.

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 GCP Instance

GCP Security Overview

Access to an GCP cloud instance can be controlled by

- an external firewall,
- the operating system firewall of the instance,
- GCP-specific firewall settings.

In addition to allowing SSH access, the different firewall levels must be configured to permit at least access to any required license servers.

GCP Firewall Rules

In addition to firewall rules created by the customer, there are other rules that can affect incoming or outgoing traffic:

- Certain IP protocols, such as GRE, are not allowed within a VPC network. For more information, see [always blocked traffic](#).
- Communication between a VM instance and its corresponding metadata server (169.254.169.254). Is always allowed.
- Every network has two implied firewall rules that permit outgoing connections and block incoming connections. Firewall rules that you create can override these implied rules.
- The default network is pre-populated with firewall rules that can be deleted or modified.

VPC firewall rule characteristics:

- Each rule is either for incoming or outgoing traffic. It can allow or deny traffic.
- Only IPv4 traffic is supported.
- Firewall rules are stateful (return traffic for an established connection is allowed).
- If TCP traffic is fragmented, a rule is only applied to the first fragment of a packet.

Connecting to the Cloud Instance

During the configuration of your instance you should have created a security group allowing at the minimum SSH access to the instance. If this has been done correctly, you can, 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.

You will need the following:

- Access to the private key associated with the public key you uploaded during the configuration of the instance.
- The public IP address of the instance.
- If you did not create the instance from a Charon-SSP marketplace image, you will also need the username created during instance launch (based on the uploaded SSH key).

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. GCP also offers additional ways of connecting to your instance (e.g., in a browser window). Please refer to the Google cloud documentation for more information about these methods.

- [SSH Command-Line Access](#)
- [SFTP File Transfer](#)
- [Connecting with the Charon-SSP Manager](#)

SSH Command-Line Access

Contents

- [General Information](#)
- [General Login Steps](#)
- [Setting the Management Password](#)

General Information

During the configuration of your instance you should have created 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-SSP Manager

Contents

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

General Information

To 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).

Prerequisites:

- Typically, **Charon-SSP Manager** is installed on a 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 the Charon host itself and be displayed on a remote system using X11-Forwarding via an SSH connection.
- **For access via the public IP address of the Charon host instance:**
 - 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 Manager 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 have 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 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 the Charon-SSP host in the cloud with the Charon Manager for the first time after the initial installation you must set the management password. This can either be done via the command line (see *SSH Command-Line Access*) or via the Charon Manager as described below.

Starting the Charon Manager and Login to Charon Host

Starting the Charon Manager

To start the **Charon-SSP Manager 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 public IP address or the private VPN IP address of your Charon-SSP host instance in the **IP address** field.
- Leave the **Password** field empty.
- Enable the SSH tunnel configuration (select **ON**) unless connected to localhost.
- Change to the SSH tab to fill in the required information there.

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

- Enter the public IP address or the private VPN IP address of your Charon-SSP instance 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.
- 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

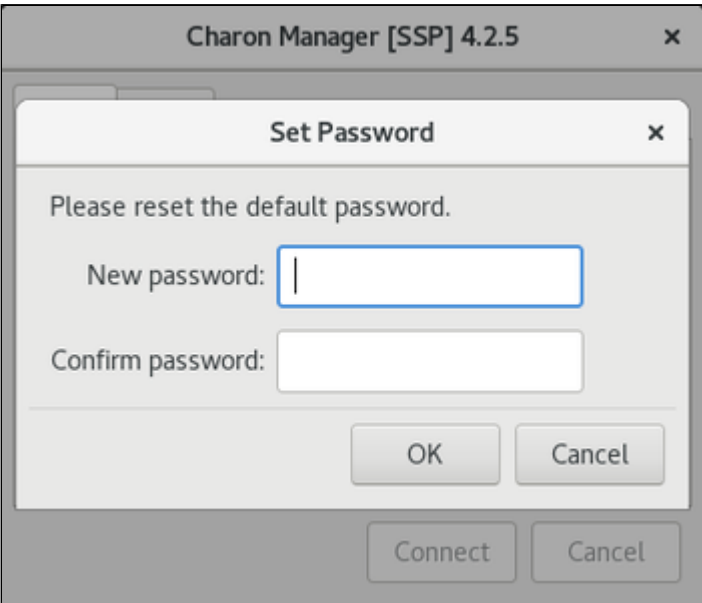
If you use the integrated SSH tunnel, perform the following steps:

- 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).

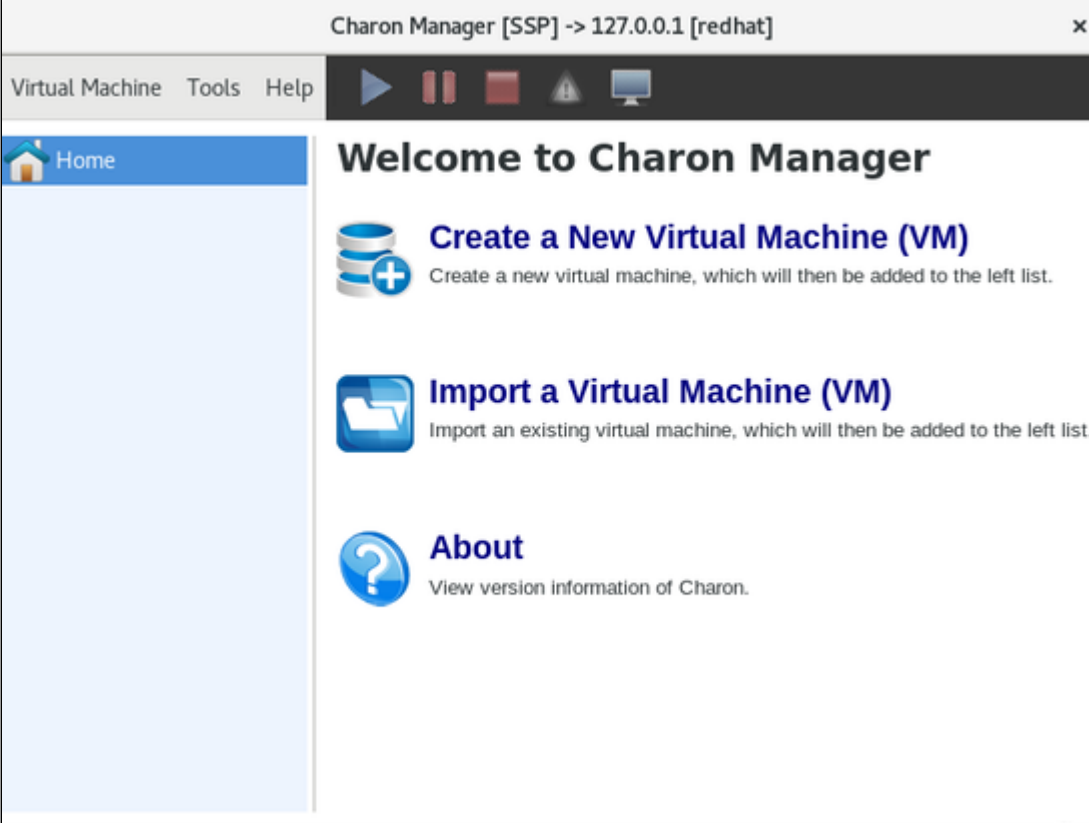
Step 3: connecting to the Charon host system

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.
- 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 GCP Instance Configuration

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

Contents

- [Storage Management](#)
- [Network Management](#)
- [Charon-SSP Cloud Networking](#)

Storage Management

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

Contents

- [Steps in the GCP Storage Environment](#)
 - [Creating a New Volume](#)
 - [Attaching an Existing Volume to an Instance](#)
 - [Detaching a Volume from an Instance](#)
- [Steps on the Charon-SSP Host System](#)
 - [Mounting a Newly Attached Volume Using the Storage Manager \(AL images only\)](#)
 - [Mounting a Newly Attached Volume Manually](#)
 - [Unmounting a Volume](#)

Steps in the GCP Storage Environment

In the GCP environment, you can, for example,

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

These steps are shown below.

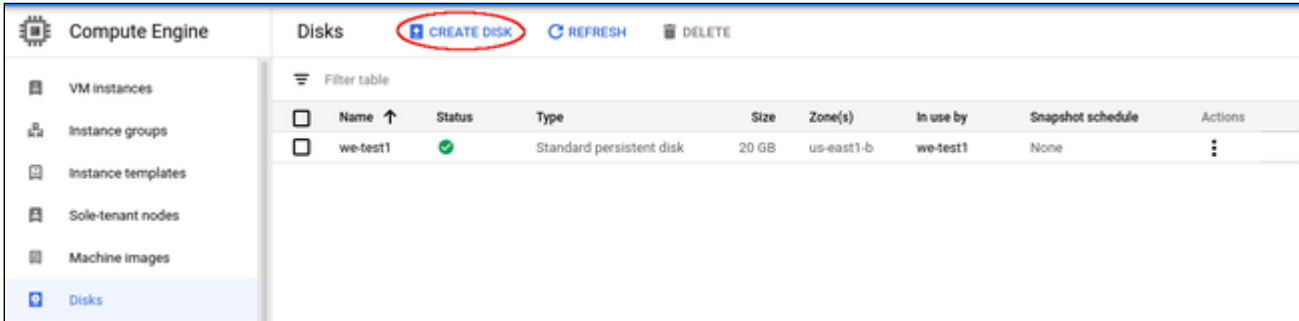
For more details, please refer to the GCP documentation

Creating a New Volume

Please note: You can also create a new disk from the Edit VM function. The present example shows how to create a new disk in the disk section.

Step 1: open the disk creation window.

As shown in the example below, select **Compute Engine > Disks** to get to the disk overview page. On this page click on **Create Disk** to open the disk creation window.



Step 2: enter the disk information in the disk creation window.

Once the disk creation window has been opened, add the required information and select the disk size and characteristics appropriate for your project. The image below shows an example:

Compute Engine

- VM instances
- Instance groups
- Instance templates
- Sole-tenant nodes
- Machine images
- Disks
- Snapshots
- Images
- TPUs
- Migrate for Compute Engine
- Committed use discounts
- Metadata
- Health checks
- Zones
- Network endpoint groups
- Operations
- Security scans
- OS patch management
- Settings
- Marketplace

← Create a disk

Name ?
Name is permanent

Description (Optional)

Type ?

Standard persistent disk

Replicate this disk within region ?

Region ? **Zone** ?
Region is permanent Zone is permanent

us-east1 (South Carolina)

us-east1-b

Snapshot schedule
Use snapshot schedules to automate disk backups. [Scheduled snapshots](#) ↗

No schedule

Source type ?

Blank disk

Image

Snapshot

Size (GB) ?

Estimated performance ?

Operation type	Read	Write
Sustained random IOPS limit	375.00	750.00
Sustained throughput limit (MB/s)	60.00	60.00

Encryption
Data is encrypted automatically. Select an encryption key management solution.

Google-managed key
No configuration required

Customer-managed key
Manage via Google Cloud Key Management Service

Customer-supplied key
Manage outside of Google Cloud

Labels ? (Optional)

+ Add label

When done with entering the information, click on Create at the bottom of the screen.

The new disk will be listed on the disk overview page:

Compute Engine	Disks	CREATE DISK	REFRESH	DELETE						
	Filter table									
	<input type="checkbox"/>	Name ↑	Status	Type	Size	Zone(s)	In use by	Snapshot schedule	Actions	
	<input type="checkbox"/>	we-disk1	✔	Standard persistent disk	200 GB	us-east1-b		None	⋮	
	<input type="checkbox"/>	we-test1	✔	Standard persistent disk	20 GB	us-east1-b	we-test1	None	⋮	

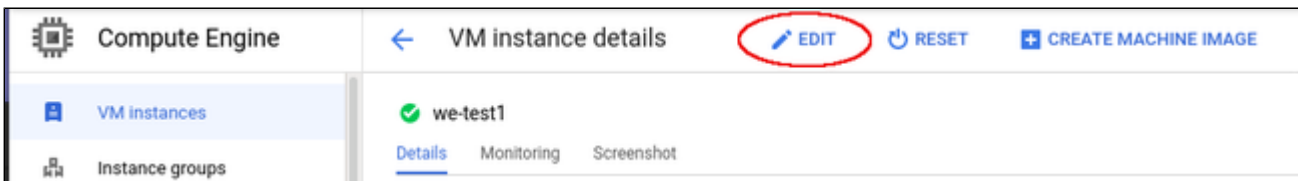
Attaching an Existing Volume to an Instance

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

Step 1: open the editor for your instance.

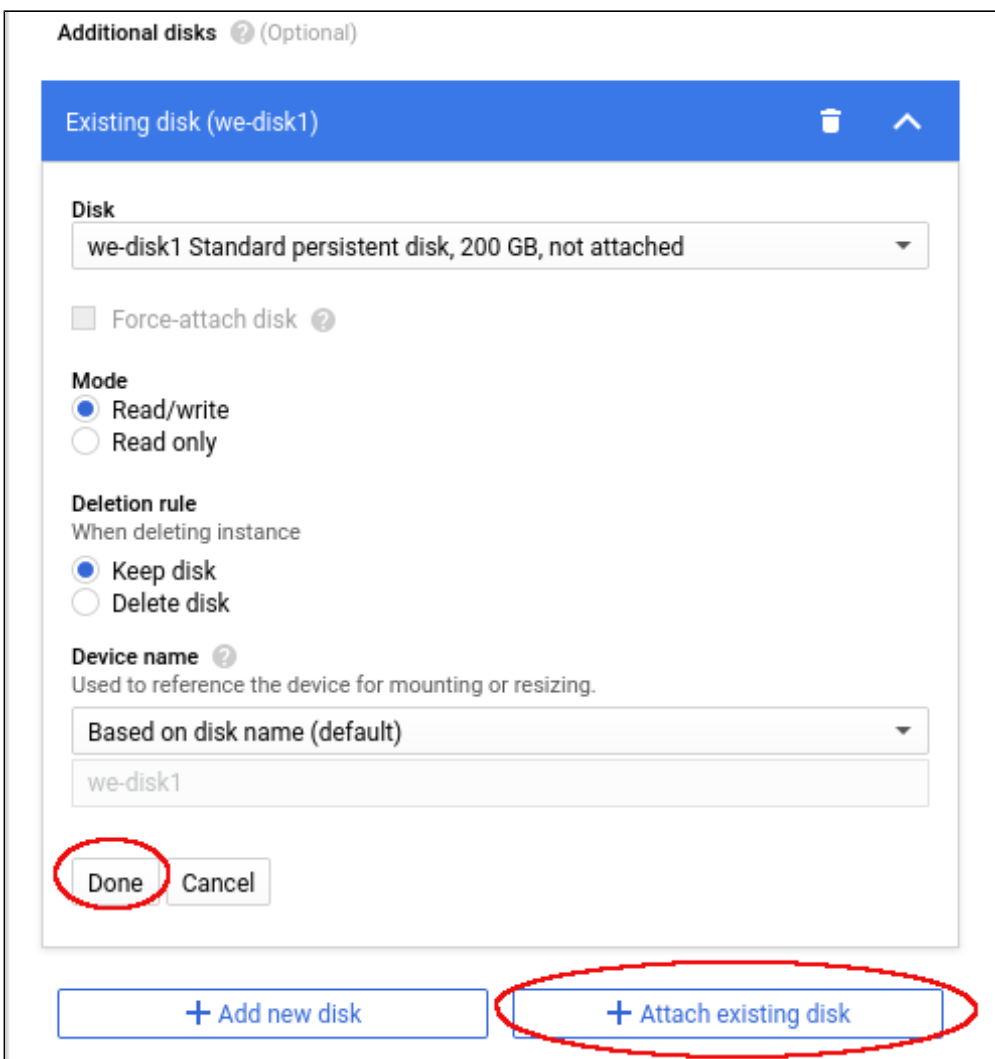
To edit your instance,

- select it to open the details page,
- then click on **Edit** at the top of the details page.



Step 2: add disk to configuration.

In the editor window, scroll down to the disk section.



Here you can either create and add a new disk or, as shown in the image above, attach and existing disk by clicking on **Attach existing disk**.

- Select your disk in the drop-down menu.
- Adjust the other parameters as needed.
- Click on **Done** to complete the configuration.

To save the changes, click on **Save** at the bottom of the editor window.

Detaching a Volume from an Instance

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

Then detach the volume from your instance:

- Open the editor for instance.
- Go to the disk section.
- Select the **wastebasket** symbol next to the disk.
- Save the changes by clicking on **Save** at the bottom of the editor page.

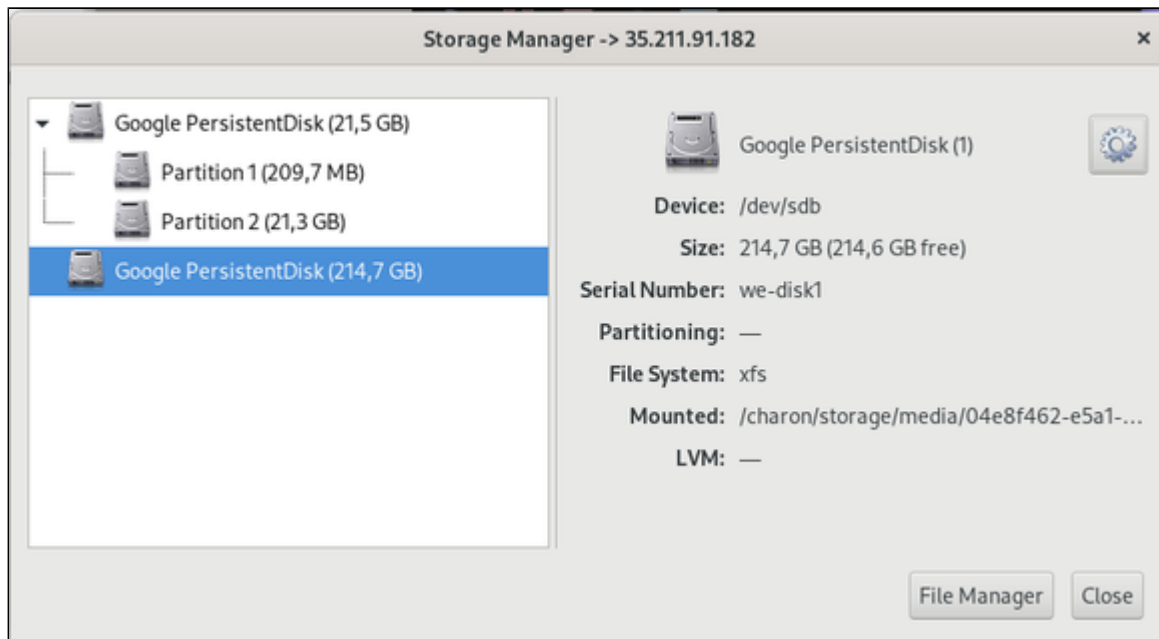
Steps on the Charon-SSP Host System

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 > Google 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.
Please note: 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.

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. Please note that this is just a basic example. As the `/dev/sdX` device names are not guaranteed to be persistent, it is better to use names from the `/dev/disk/by-*` hierarchy (for example `by-uuid`) for permanent 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 > Google 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`.

Network 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 GCP documentation for details.

Contents

- [General Information](#)
- [Create VPCs and Subnets for Instance](#)
- [Adding Additional NICs to an Instance](#)
- [Assigning a Static IP Address to a Network Interface](#)
- [Detaching a Network Interface from an Instance](#)
- [Address Assignment Information](#)
 - [General information](#)
 - [Address Ranges](#)
- [Interface Configuration on Linux](#)
- [Additional GCP-specific Information](#)
 - [IP Interface Netmask](#)
 - [Routing between VPCs](#)
 - [Network Interface MTU](#)

When an instance is created, a default Ethernet interface is attached to the system. This default network interface is mandatory. During the creation of the instance, you can add additional network interfaces.

General Information

The rules for Google cloud instances with respect to network interfaces are strict:

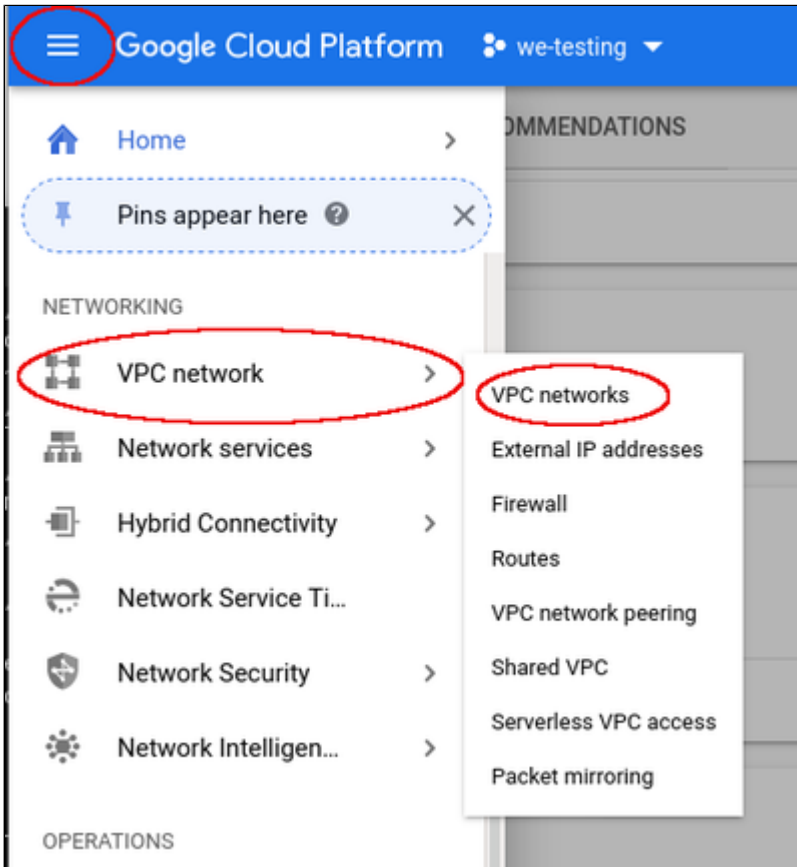
- Interfaces can only be added during instance creation.
- Each network interface configured in a single instance must be attached to a different VPC network.
- The additional VPC networks that the multiple interfaces will attach to must exist before an instance is created. See [Using VPC Networks](#) for instructions on creating additional VPC networks.
- You cannot delete a network interface without deleting the instance.

Therefore the required VPCs and subnets must exist before the instance is created.

To create additional VPCs (if required), perform the steps below.

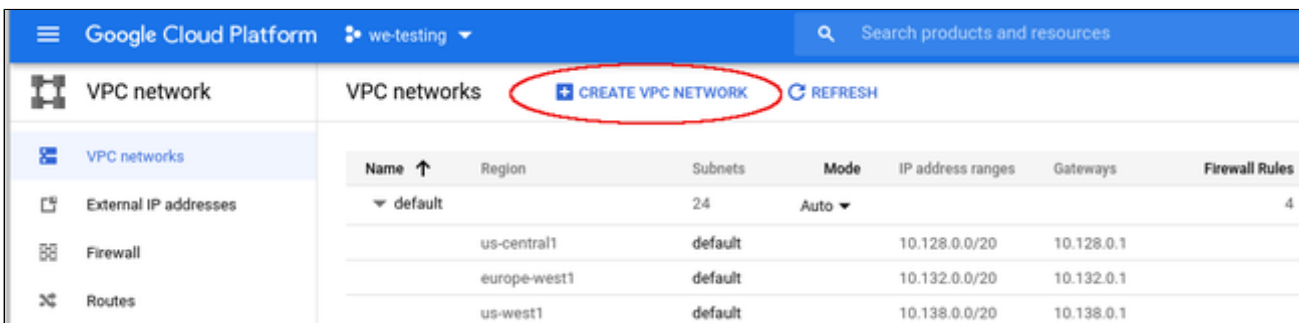
Create VPCs and Subnets for Instance

Step 1: Open the VPC network section by clicking on the Navigation menu, then selecting VPC network, and clicking on VPC networks - as illustrated below.



This will open the VPC overview page with the already existing VPCs. If all required VPCs and subnets already exist, continue with creating the new VM instance. Otherwise, continue with step 2.

Step 2: If you need to create a new VPC, click on CREATE VPC NETWORK at the top of the VPC overview list.

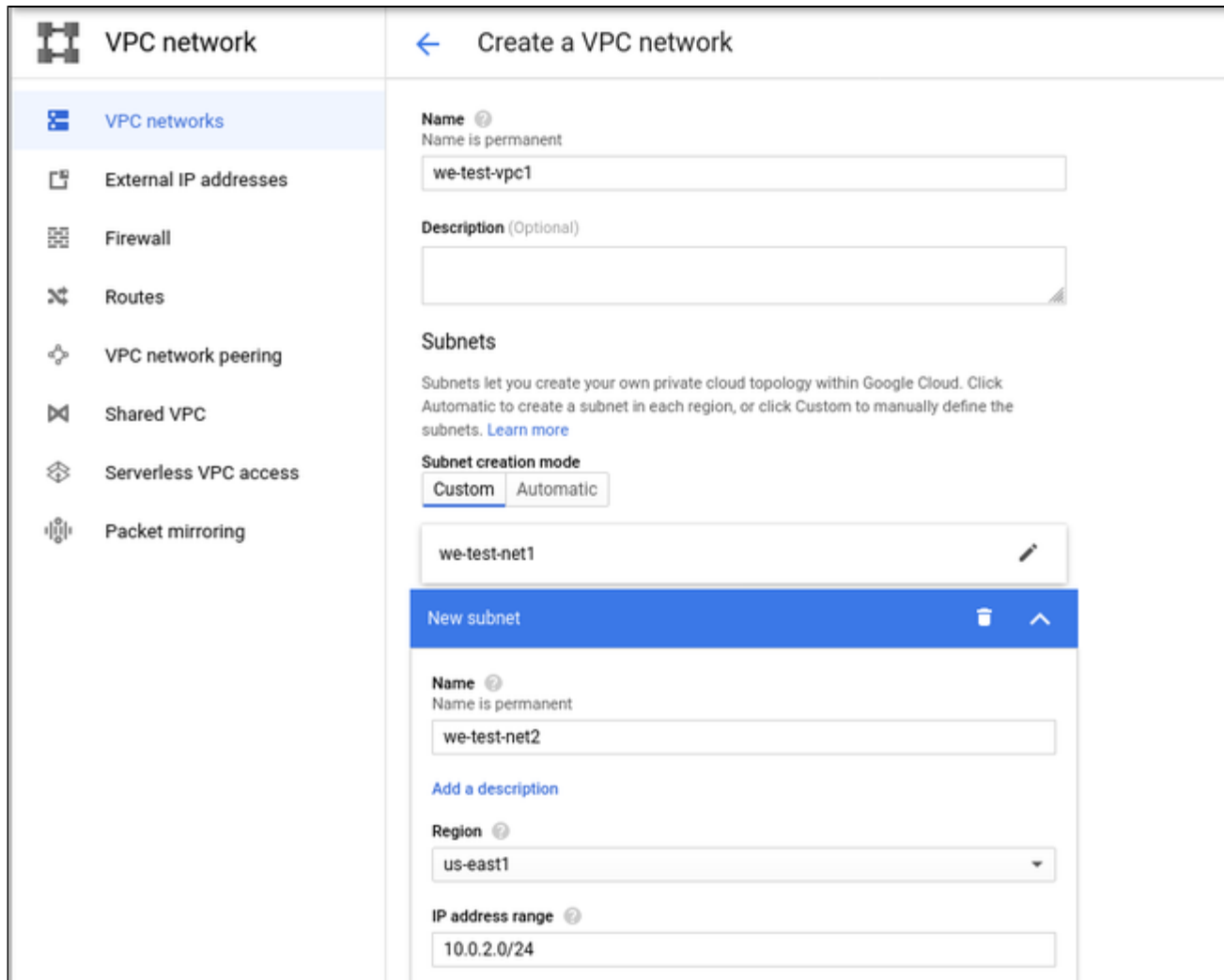


This opens the VPC configuration window.

Step 3: Create VPC and subnets.

In the VPC configuration window, enter

- the VPC name, and
- the subnet name, region and address.



VPC network

Create a VPC network

Name ⓘ
Name is permanent

we-test-vpc1

Description (Optional)

Subnets

Subnets let you create your own private cloud topology within Google Cloud. Click Automatic to create a subnet in each region, or click Custom to manually define the subnets. [Learn more](#)

Subnet creation mode

Custom Automatic

we-test-net1

New subnet

Name ⓘ
Name is permanent

we-test-net2

[Add a description](#)

Region ⓘ

us-east1

IP address range ⓘ

10.0.2.0/24

Click on **Create** at the bottom of the window to create the VPC.

The new VPC should appear in the VPC overview list. Selecting the VPC in the overview list will open the detail information window. Example:

The screenshot shows the 'VPC network details' page for a VPC named 'we-test-vpc1'. The left sidebar contains navigation options: VPC networks, External IP addresses, Firewall, Routes, VPC network peering, Shared VPC, Serverless VPC access, and Packet mirroring. The main content area shows the VPC name and configuration: Subnet creation mode (Custom subnets), Dynamic routing mode (Regional), and DNS server policy (None). Below this are tabs for Subnets, Static internal IP addresses, Firewall rules, Routes, VPC Network Peering, and Private service connection. The 'Subnets' tab is active, showing a table of subnets:

<input type="checkbox"/>	Name ^	Region	IP address ranges	Gateway	Private Google access	Flow logs	
<input type="checkbox"/>	we-test-net1	us-east1	10.0.1.0/24	10.0.1.1	Off	Off	
<input type="checkbox"/>	we-test-net2	us-east1	10.0.2.0/24	10.0.2.1	Off	Off	

At the bottom, there is a link for 'Equivalent REST'.

Step 4: Create firewall rules for the VPC.

With the detail information open, click on Firewall. This will allow you to define the required firewall rules for the VPC.

An example of a small set of firewall rules that allow incoming SSH and ICMP is shown below:

The screenshot shows the 'VPC network details' page for 'we-test-vpc1' with the 'Firewall rules' tab selected. The configuration details (Subnet creation mode, Dynamic routing mode, DNS server policy) are the same as in the previous screenshot. The 'Firewall rules' tab shows a table of rules:

<input type="checkbox"/>	Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Logs	Hit count	Last hit
<input type="checkbox"/>	icmp-any	Ingress	Apply to all	IP ranges: 0.0.0.0/24	icmp	Allow	1000	Off	--	--
<input type="checkbox"/>	ssh-any	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	1000	Off	--	--

Adding Additional NICs to an Instance

Additional NICs are added **during instance creation**. Perform the following steps in the instance creation window:

- Open the advanced settings at the bottom of the VM creation window by clicking on **Management, security, disks,...** at the bottom of the page.
- Select Networking from the advanced settings section.
- Click on **Add network interface**.
- Select the correct subnet (created before).
- Set the information about internal and external IP address (static or ephemeral) as required.

Management Security Disks **Networking** Sole Tenancy

Network tags [?] (Optional)

Hostname [?]
Set a custom hostname for this instance or leave it default. Choice is permanent
we-test1.us-east1-b.c.we-testing-283214.internal

Network interfaces [?]
Network interface is permanent

default default (10.142.0.0/20)

Network Interface

Network [?]
we-test-vpc1

Subnetwork [?]
we-test-net1 (10.0.1.0/24)

Primary internal IP [?]
Ephemeral (Automatic)

Show alias IP ranges

External IP [?]
Ephemeral

Network Service Tier [?]
 Premium (Current project-level tier, [change](#)) [?]
 Standard (us-east1) [?]

Done Cancel

After adding all the required information, click on **Done**.

The second interface is now visible in the details page of the VM instance:

OS patch management		Network interfaces								
	Settings	Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier [?]	IP forwarding	Network details
		nic0	default	default	10.142.0.2	–	35.196.76.164 (ephemeral)	Premium	Off	View details
		nic1	we-test-vpc1	we-test-net1	10.0.1.2	–	104.196.35.212 (ephemeral)	Premium		View details
	Marketplace									

Assigning a Static IP Address to a Network Interface

During the creation of a VM instance, when you add the default and optional additional NICs, you can determine if the IP addresses assigned to a NIC are static (persistent across restarts) or ephemeral (non-persistent across restarts). The process to add a static IP requires reserving the IP address. The public IP address may also have to be created first.

If you choose to add a **static private IP** address to an interface, you will get the following window to reserve a static private IP address:

Reserve static internal IP address

Reserve IP address 10.0.1.2

Name ⓘ
Name is permanent

Description (Optional)

[CANCEL](#) [RESERVE](#)

If you choose to add a **static public IP** address to an interface, you will get the following window to create (if needed) and reserve an address:

Reserve a new static IP address

Name ⓘ
Name is permanent

Description (Optional)

Network Service Tier ⓘ

Premium (Current project-level tier, [change](#)) ⓘ

Standard ⓘ

Region
us-east1

ⓘ Standard tier uses the same region as your VM instance

[CANCEL](#) [RESERVE](#)

You can also manage external IP addresses from the VPC network management section (**Navigation menu > VPC network > External IP addresses**):

Name	External Address	Region	Type	Version	In use by
--	35.211.32.252	us-east1	Ephemeral	IPv4	VM instance we-test1 (Zone us-east1-b)
--	35.211.91.182	us-east1	Ephemeral	IPv4	VM instance we-test1 (Zone us-east1-b)

Detaching a Network Interface from an Instance

You cannot delete a network interface without deleting the instance it is attached to. So if you do not need a network anymore, but do not want to delete the instance, you can only disable it from the operating system level.

Address Assignment Information

General information

Each VM instance interface can have one primary internal IP address, one or more secondary IP addresses, and one external IP address.

Addresses can be static (persistent) or ephemeral (on-persistent):

- Ephemeral external IP addresses:
 - For VM instances, the ephemeral external IP address is also released if you stop the instance. After you restart the instance, it is assigned a new ephemeral external IP address.
- Static external IP addresses:
 - Static external IP address can be reserved and thereby assigned a project indefinitely until they are explicitly released. You can reserve a new static external IP address or promote an existing ephemeral external IP address to a static external IP address.
- Ephemeral internal IP addresses:
 - Ephemeral internal IP addresses remain attached to VM instances until the instance is deleted.
- Static internal IP addresses:
 - For VM instances, static internal IP addresses remain attached to stopped instances until they are removed.

Address Ranges

When creating a VPC and its subnets, subnet address ranges are assigned to these subnets. There are some restriction regarding permitted address ranges:

Restricted address ranges:

Restricted ranges include Google public IP addresses and commonly reserved RFC ranges, as described below. These ranges cannot be used for subnet ranges.

- Public IP addresses for Google APIs and services, including Google Cloud netblocks: You can find a link to these IP addresses in this [Google FAQ](#).
- 199.36.153.4/30 and 199.36.153.8/30: private Google access-specific virtual IP addresses
- 0.0.0.0/8: Current (local) network RFC 1122
- 127.0.0.0/8: Local host RFC 1122
- 169.254.0.0/16: Link-local RFC 3927
- 224.0.0.0/4: Multicast RFC 5771
- 255.255.255.255/32: Limited broadcast destination address RFC 8190 and RFC 919

Reserved subnet addresses:

Every subnet has four reserved IP addresses in its primary IP range. There are no reserved IP addresses in the secondary IP ranges.

- Network: first address in the primary IP range for the subnet 10.1.2.0 in 10.1.2.0/24
- Default gateway: Second address in the primary IP range for the subnet 10.1.2.1 in 10.1.2.0/24
- Second-to-last address: second-to-last address in the primary IP range for the subnet that is reserved by Google Cloud for potential future use 10.1.2.254 in 10.1.2.0/24
- Broadcast: last address in the primary IP range for the subnet 10.1.2.255 in 10.1.2.0/24

Please note:

- The default gateway does not respond to ping.
- The default gateway does not decrement TTL headers (used for traceroute).
- Only IPv4 unicast traffic is supported.

Interface Configuration on Linux

By default, Google cloud tools installed on the Linux instance automatically start the attached network interfaces and configure them using DHCP.

Should this be undesirable, for example, because a NIC is to be dedicated to the Solaris guest system, this automatic configuration can be suppressed by disabling the setup in the file `/etc/default/instance_configs.cfg`.

Important information:

- Currently, the Charon-SSP marketplace images are based on CentOS 7.
- NetworkManager is disabled by default in these images.
- If you disable the automatic interface setup as shown above, you **must make sure that the correct ifcfg-files for every interface exist in /etc/sysconfig/network-config**. Failure to do so, can make your instance unreachable after the next network restart.
- If you use a RHEL/CentOS 8 image as the base image for your Charon host, the interface must be controlled by the NetworkManager. You can set up the appropriate configuration by editing the interface configuration files or using `nmcli` commands.

To disable automatic interface configuration by the cloud tools, edit the file and set the parameter **setup** to **false** as shown in the example below:

```
# vi /etc/default/instance_configs.cfg
[NetworkInterfaces]
dhclient_script = /sbin/google-dhclient-script
dhcp_command =
ip_forwarding = true
setup = false
```

After restarting the network (`systemctl restart network`), the configuration as defined in the ifcfg-files should be set for the interfaces. On RHEL /CentOS 8 systems restart the NetworkManager instead (`systemctl restart NetworkManager`).

Additional GCP-specific Information

IP Interface Netmask

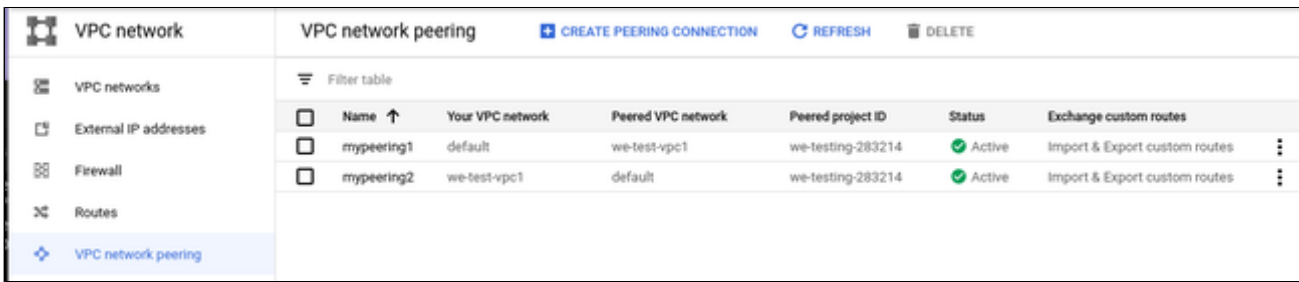
Please note: The latest images provided by Stromasys use a /24 netmask for additional NICs. Therefore, the following information no longer applies to instances created with these images.

However, other base images used to create an instance, may use a netmask of /32 for additional NICs on the VM instance. This means that only ARP requests for the default gateway are answered by the Google metadata server. In such cases, when providing a **dedicated NIC to the Solaris guest system**, that is, the internal IP address of the interface is not configured on the Linux level, but on the Solaris level, please note the following points:

- The **netmask** on Solaris has to be set to a value that includes the default gateway (e.g., /24). Otherwise, Solaris will return an error when setting the default gateway (network unreachable).
- If Solaris should communicate with systems on the **same subnet**, it needs a static ARP entry for these systems (`arp -s <target-ip> <target-mac>`). This is because the ARP requests sent by Solaris for the MAC addresses of these systems will not be answered by the Google metadata server and they will not be forwarded to the target system.

Routing between VPCs

If a VM instance has more than one NIC, each NIC must be in a different VPC. Routing between VPCs is not enabled by default. It has to be enabled through a mutual VPC peering configuration as shown in the sample below:



VPC network		VPC network peering				
		CREATE PEERING CONNECTION REFRESH DELETE				
		Filter table				
	Name ↑	Your VPC network	Peered VPC network	Peered project ID	Status	Exchange custom routes
<input type="checkbox"/>	mypeering1	default	we-test-vpc1	we-testing-283214	Active	Import & Export custom routes
<input type="checkbox"/>	mypeering2	we-test-vpc1	default	we-testing-283214	Active	Import & Export custom routes

The example shows one rule for each routing direction between the two VPCs.

If this is not enabled, host and guest system can only communicate via the external IP addresses, not via the internal IP addresses.

Network Interface MTU

The VPC network has a maximum transmission unit (MTU) of 1460 bytes. Interfaces should be configured to this value to avoid the increased latency and packet overhead caused by fragmentation. Client applications that communicate with GCP instances over UDP should have a maximum payload of 1432 bytes to avoid fragmentation.

Charon-SSP Cloud Networking

Contents

- [General Information](#)
 - [Host to Guest Communication Considerations](#)
 - [External Communication Considerations](#)
 - [Guest to Guest Layer 2 Communication Considerations](#)
 - [Asymmetric Routing Considerations](#)
 - [Cloud Instance and IP Forwarding](#)
- [Further Information](#)

General Information

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

Please note:

- NetworkManager is disabled on Charon-SSP cloud-specific marketplace images (CentOS 7). Therefore, the interface configuration relies on **ifcfg**-files in **/etc/sysconfig/network-scripts**. For conventional RPM installations the basic network configuration environment used depends on the Linux image chosen to launch the instance. It is recommended to create the **ifcfg**-files for additional interfaces manually before using the Charon Manager to manage these interfaces.
- If the Charon host instance runs RHEL/CentOS 8.x, the network interfaces must be managed by the NetworkManager. The configuration can be set up via configuration files or **nmcli** commands. The Charon Manager ignores any interface that is not under NetworkManager control (unmanaged interfaces in the output of **nmcli device**).
- If the information in this chapter is not sufficient, please refer to the other Charon-SSP documentation provided by Stromasys and the documentation provided by your cloud provider for up-to-date and comprehensive information.

Host to Guest Communication Considerations

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

1. Internal virtual bridge on the host system:

Such a bridge has several TAP interfaces. The host and the guest systems are connected to this bridge and can communicate directly to one another using L3 and L2 protocols. The bridge uses its own IP subnet that can be defined by the user. Setting up such a configuration is supported by the Charon-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 no longer possible if the 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 the correct configuration, 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 file for the additional interface.
- Remove the private IP address assigned to the second interface by the cloud provider from the Linux configuration (if it has been configured).
- Use the Charon Manager to assign the interface to the emulated SPARC system.
- Use the Charon Manager to set the MAC address of the emulated SPARC system to the same value as the one used on the host system Ethernet interface.
- On the Solaris system, configure the private IP address that was previously assigned to the second interface on Linux and configure the appropriate default route for the LAN.

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 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 endpoint 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 Solaris guest system to the Internet:

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

Access to the Internet from subnets or a Solaris guest system 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 or it can be often be provided by the cloud provider for a charge.

Please note: a Charon-SSP AL host system 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 Solaris guest access to the Internet:

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

One of these interfaces is then dedicated to the guest system which uses the private interface address and the MAC address assigned to the Charon host by 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 assume a RHEL or CentOS 7 system (as provided by the Charon-SSP marketplace images) where interfaces are managed outside the control of the NetworkManager. A RHEL or CentOS 8 system must use the NetworkManager to manage network interfaces. Please refer to your Linux documentation for details.
- The actual steps may vary slightly depending on the specific cloud environment. Please always refer to your cloud provider's documentation.

When adding a second IP interface on the Charon-SSP host, the routing problems described above can occur. To solve them, perform the following basic steps:

1. Create a configuration file (`/etc/sysconfig/network-scripts/ifcfg-<interface-name>`) for the second interface (if there is no configuration file for the primary interface, create it as well).
2. Set the correct interface for default route in `/etc/sysconfig/network` (example: `GATEWAYDEV=eth0`).
3. Restart the network.
4. Create an additional routing table (use the command: `ip route add <path> dev <interface-name> table <table-id>`). There must be an entry for every IP address assigned to the second interface and any other route to be used via this interface. The parameter `<table-id>` is a numeric value or the string `default` (for the main routing table).
5. Set rules in the Routing Policy Database (use the command: `ip rule add from <ip-address-of-second-interface> lookup <table-id>`).
6. If required, remove conflicting routes from the main routing table and add routing rules for the main routing table.
7. Create a static route file (`/etc/sysconfig/network-scripts/route-<interface-name>`).
8. Create a static rule file (`/etc/sysconfig/network-scripts/rule-<interface-name>`).

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

Additional information: 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**.

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.

Further Information

The following sections show sample network configurations:

- SSH VPN tunnel to connect Charon host and guest to remote systems or customer networks
- Dedicated NIC for the Solaris guest system

SSH VPN - Connecting Charon Host and Guest to Customer Network

Contents

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

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.

Please note: 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:

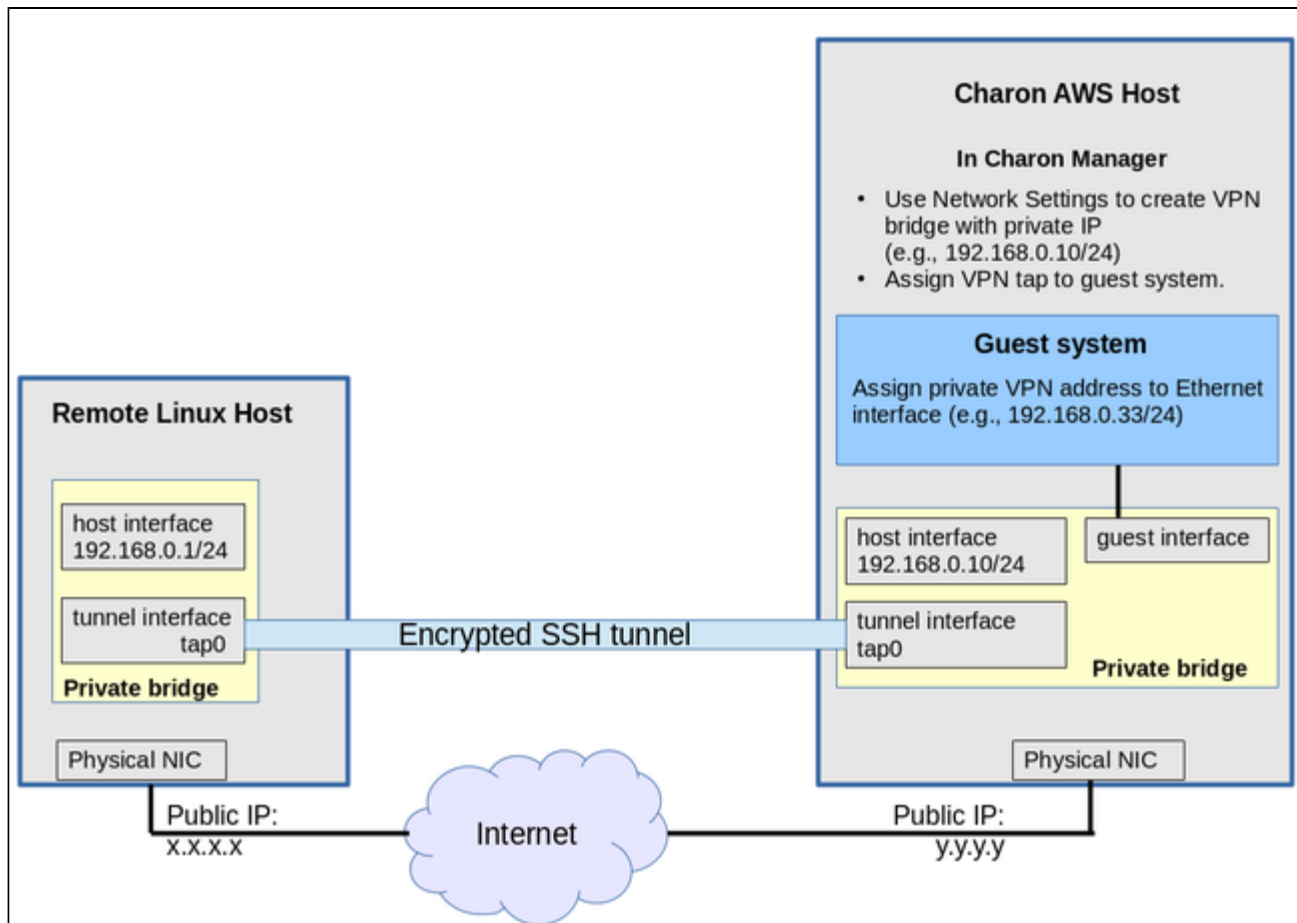
```
# ssh-keygen -t rsa -b 4096 -f ~/.ssh/<keyname> -q
```

The resulting key-pair can then be associated with instance during instance creation and used to create an encrypted SSH connection.
- The *bridge-utils* package must be installed on the Charon host, and 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:

- Open the Charon-SSP Manager and log in to the Charon-SSP host.
- In the Charon Manager, open the Network Settings window by clicking on **Tools > Network Settings**. This will open the **Network Settings** window.
- 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:	<input type="text" value="ON"/>
Binding interface:	<input type="text" value="OFF"/>
STP for bridge:	<input type="text" value="OFF"/>
Virtual bridge interface:	<input type="text"/>
Virtual bridge name:	<input type="text" value="vpn0"/>
Number of virtual adapters:	<input type="text" value="1"/>
IP settings:	<input type="text" value="Manual"/>
IP address:	<input type="text" value="192.168.0.10"/>
Netmask:	<input style="border: 2px solid blue;" type="text" value="255.255.255.0"/>
Gateway:	<input type="text"/>
DNS server 1:	<input type="text"/>
DNS server 2:	<input type="text"/>

To learn more about the virtual network configuration options, refer to section *Host System Network Configuration* 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).

The screenshot shows the 'Virtual Machine Settings' window. On the left, a 'Device Summary' table lists various hardware components. The 'Ethernet tap0_vpn0' entry is highlighted in blue. On the right, the 'Ethernet' configuration section is active, showing 'Add-on adapter model: HME' with a dropdown arrow. Below this, a table lists the assigned interface:

Interface	Model	MAC Address
tap0_vpn0	HME	

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	<code># ip tuntap add dev tap0 mod tap</code>
Enable TAP interface	<code># ip link set tap0 up</code>
Create bridge	<code># ip link add name br_vpn0 type bridge</code>
Enable bridge interface	<code># ip link set br_vpn0 up</code>
Define IP address for bridge	<code># ip addr add 192.168.0.1/24 dev br_vpn0</code>
Add TAP interface to bridge	<code># ip link set tap0 master br_vpn0</code>
<p>Start the SSH tunnel</p> <p>autossh is a program to start a copy of ssh and monitor it, restarting it as necessary should it die or stop passing traffic.</p> <p>Once started, you can move the program to the background.</p>	<pre># autossh -M 9876 -o ServerAliveInterval=60 -o Tunnel=ethernet \ -w 0:0 -t -i <path-to-private-key> -NCT sshuser@<public-cloudinstance-IP></pre> <p>-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</p> <p>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.</p>

Possible additional steps:

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/<interface-name>.hostname`.

On Solaris 11, use the commands `ipadm create-ip netX` and `ipadm create-addr -T static -a <ip-address>/<netmask> netX/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 ffffffff broadcast 192.168.0.255
      ether d4:2:7c:c1:d2:59
bash-3.2#
bash-3.2# netstat -rn

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

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

To make the entry permanent

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

Stopping the SSH Tunnel

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

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

Dedicated NIC for Guest System

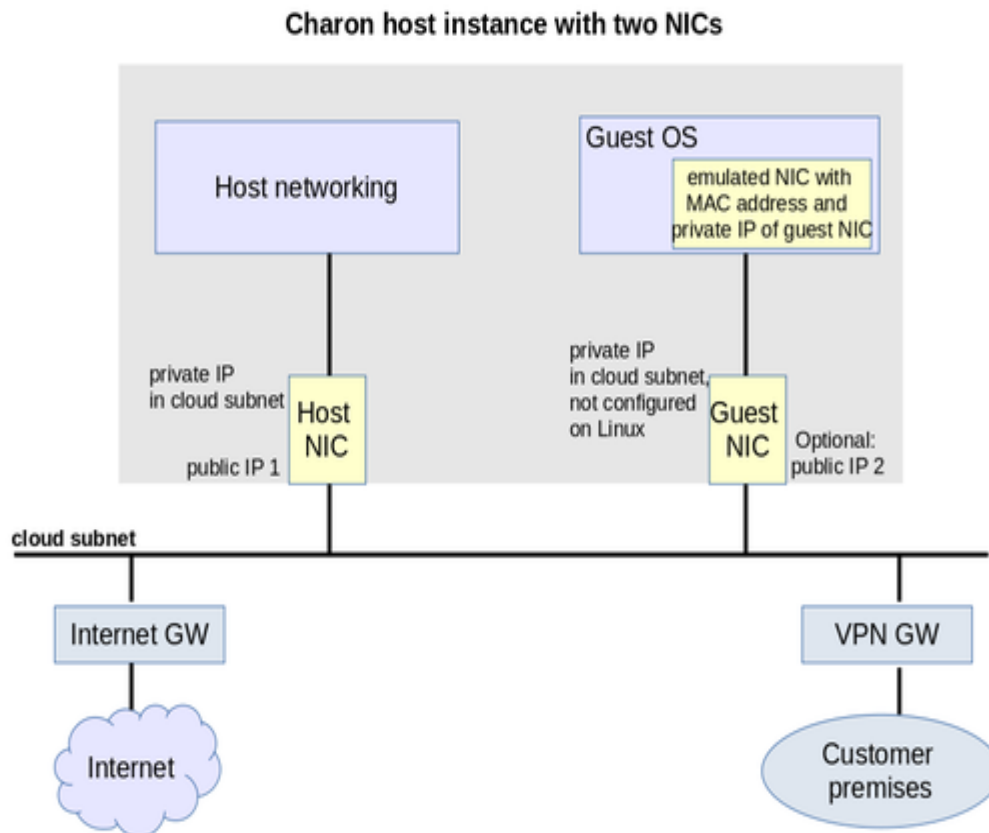
Providing a dedicated NIC for guest operating systems is the standard method in non-cloud environments. However, this configuration poses some challenges in cloud environments where MAC address / IP address combinations are fixed parameters set by the cloud provider.

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.



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 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 Example

Important information:

- **The example assumes that a Charon-SSP cloud-specific marketplace image is used.** This means in particular:
 - The host system is a CentOS 7 system.
 - NetworkManager is disabled and the ifcfg-files in /etc/sysconfig/network-scripts are used to set up the configuration.
- If you use a different host operating system version, you must adapt the example accordingly.
- If you use a **RHEL/CentOS 8 system**, you must use NetworkManager to configure the interface. A similar procedure as the one described here can be used, but the **interfaces must be under NetworkManager control** and instead of restarting the network, you must restart the NetworkManager after editing the ifcfg-files. Alternatively, you can use nmcli commands to configure the connection. Please refer to your Linux documentation and manual pages for further information.
- **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. The latest 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.
- If you use the Charon Manager for the interface configuration (steps 4 and 5 of the example), use **None** as the interface configuration. Charon Manager will also activate the changes (step 6 in the manual example below).

Step 1: configure a second network interface on the Charon host system for use by the Solaris guest system.

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

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 CentOS 7 system and that the interface is configured outside the control of the NetworkManager.

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:

- a) On Charon-SSP instances based on cloud-specific marketplace images (**CentOS 7**), the NetworkManager is normally disabled. However, if the NetworkManager is enabled on such systems, the line `NM_CONTROLLED=no` prevents the NetworkManager from changing the configuration of the interface. If using a **RHEL/CentOS 8** host system, the `NM_CONTROLLED` statement **must be removed or set to yes**.
- b) 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          (see note a above)
```

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.

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.

Please note: Make sure to use the correct interface names in use on your instance.

Step 2: add the dedicated Ethernet interface 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 interface on the Solaris guest system to use the private IP assigned to the second NIC by your cloud provider.

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 netX** and **ipadm create-addr -T static -a <private-guest-nic-ip>/<netmask> netX/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
 - 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).

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](#) for more information about configuring and managing Charon-SSP.