Stromasys Charon-SSP Solaris Emulator

Azure Setup Guide

By Jonathon J. Frost
Microsoft Principal Program Manager

Jim Dugan
Microsoft Principal Program Manager

Reviewed by Waltraud Erber and Tri Trong Trinh

Stromasys

June 2020

Contents

Authored by Jonathon Frost and Jim Dugan, Azure Global Engineering. Edited by Nanette Ray. Reviewed by Waltraud Erber and Tri Trong Trinh, Stromasys.

1 Introduction

This guide walks through the steps to set up and install Stromasys Charon-SSP for Solaris Emulator on Microsoft Azure. Charon-SSP is a member of the Charon cross-platform hardware virtualization product family. It creates a virtual replica of Sun-4m, Sun-4u, or Sun-4v SPARC family members on a standard x86-64 computer system running Linux on top of physical hardware or a hypervisor.

Through Charon-SSP, you can continue to use applications that run on end-of-life SPARCstation or SPARCserver without changes. Running applications in an emulator on Azure has several benefits, such as lower operational costs and energy consumption. In addition, you can run multiple application instances on a single x86-64 standard host or an existing virtualization infrastructure, giving you the added advantages of consolidation while easing management and maintenance of legacy systems.

Charon-SSP provides the following virtualized SPARC models. This guide covers the models with a checkmark—Charon-SSP/4M, Charon-SSP/4U, and Charon-SSP/4V.

Table 1. Charon-SSP virtualized SPARC models covered in this guide

Model	In this guide	Description
Charon-SSP/4M	✔	Based on SPARC-V8 32-bit processor specification. MBUS for processor/memory interconnection, and SBUS for IO peripherals.
Charon-SSP/4U	✔	Based on SPARC-V9 64-bit processor specification. UPA bus for processor/memory interconnection and PCI bus for IO peripherals.
Charon-SSP/4U+		Same as /4U. Uses Intel VT-x / EPT to offload SPARC MMU operations to hardware. Must run on a bare-metal Intel host.
Charon-SSP/4V	✔	Based on SPARC-V9 64-bit processor specification and Sun-4v hypervisor architecture. Each instance supports one LDom.
Charon-SSP/4V+		Same as /4V. Uses Intel VT-x / EPT to offload SPARC MMU operations to hardware. Must run on a bare-metal Intel host.

Stromasys is a Microsoft partner, and we worked closely together in creating this guide. It describes how to set up a Linux Azure Virtual Machine (VM) to install and run the Charon-SSP Solaris emulator, install Solaris 10 in the emulated environment, configure the networking, use Azure Files storage for virtual tape backup, and set up XDMCP for Solaris graphical desktop access.

1.1 Charon-SSP architectural overview

This guide implements a relatively simple setup of two Azure VMs—one running Stromasys Charon-SSP and the other, a nested VM running an emulation of Solaris 10. An optional hop server is used to connect to the Solaris VM in Azure over XDMCP, a remote desktop protocol. Figure 1 provides a high-level look at the implementation, which uses the IP addresses shown in Table 1 for the network interface cards (NICs).

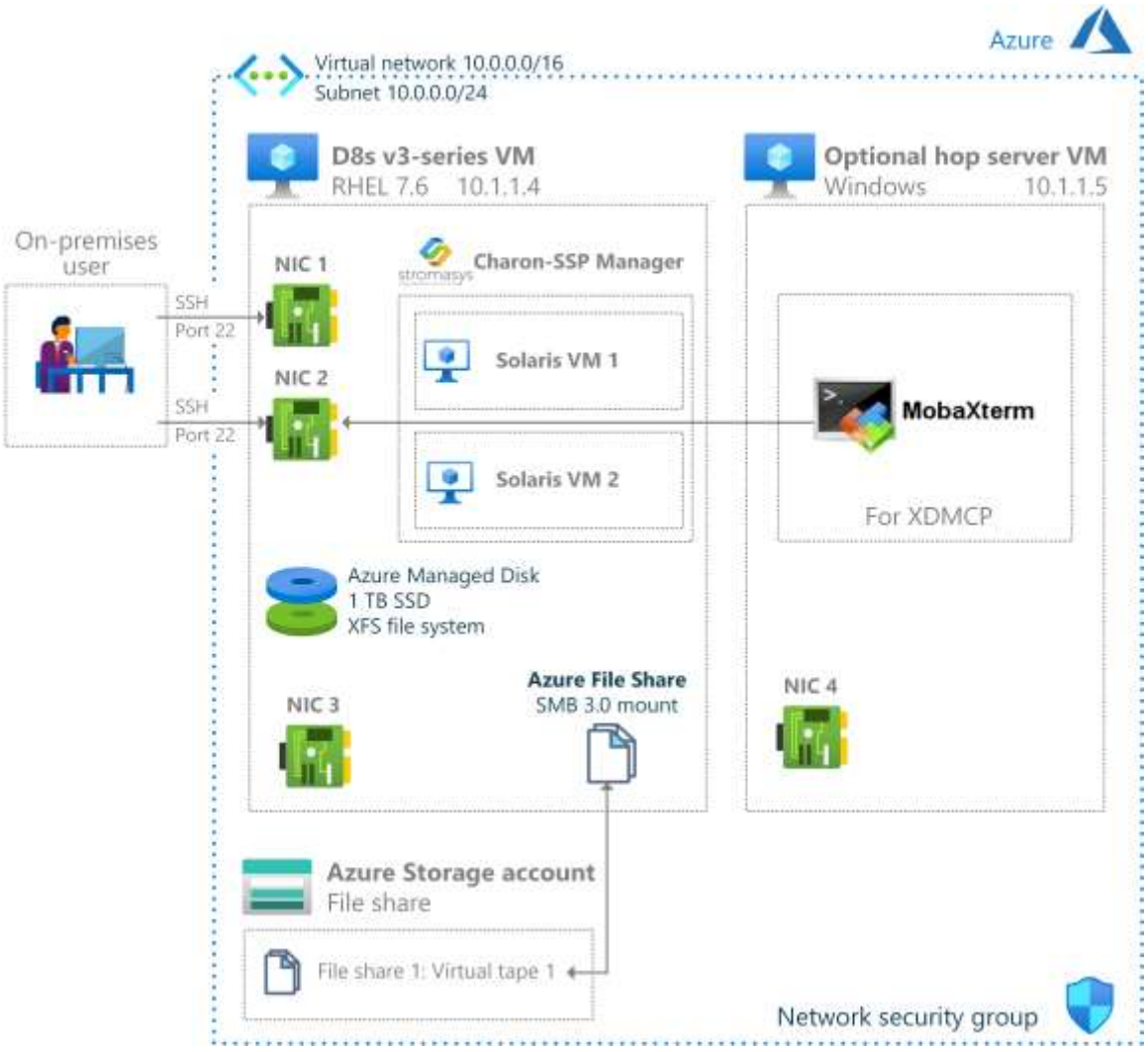


Figure 1. Architecture overview of a single-host VM running Charon Manager and Agent.

Table 2. The public and private IP addresses for the NICs

Network interface	Private IP	Public IP	Network interface	Private IP	Public IP
NIC1	10.1.1.4	51.X.X.57	NIC3	10.1.1.26	51.X.X.192
NIC2	10.1.1.25	51.X.X.83	NIC4	10.1.1.5	51.X.X.67

1.2 Charon-SSP in a scale out scenario

In a large production environment, Charon-SSP is typically distributed across VMs in a scale-out configuration. Although this type of setup is beyond the scope of this guide, it is important to keep it in mind if you support enterprise-scale deployments. Figure 2 shows a typical scale-out scenario. The numbered annotations refer to the following:

1. Charon-SSP Director is used to manage multiple server hosts, each running one or more child Solaris VMs. This setup provides a single place of management as you scale out your farm of host VMs and their Solaris child VMs.
2. Charon-SSP Agent runs on Linux distributions on VMs. This component runs the child Solaris VMs and emulates the SPARC processor architecture.
3. Solaris VMs are based on the SPARC processor architecture to support low friction lift-and-shift of the on-premises workloads running on SPARC Solaris machines to Azure.
4. Each child Solaris VM has its own NIC with a dedicated private IP address. You can easily set up an Azure public IP address on the same network interface.
5. The Solaris VM images can reside on the solid-state drive (SSD) managed disk of the host VM. For even higher IOPS, consider Ultra SSD disks.
6. An Azure storage account file share is optionally mounted on the Linux VM. You can then map the Charon-SSP Virtual Tape feature to a locally mounted device that is backed by the file share. This setup provides a low-cost way to archive tapes for regulatory or other reasons.
7. The management VM running Charon-SSP Director and Charon-SSP Manager can run Windows or Linux with a graphic user interface such as GNOME.
8. Users can use Secure Shell Host (SSH) to connect directly to the Solaris VMs, which have dedicated network interfaces and IP addresses. XDMCP provides desktop access to the Solaris VMs though a hop server running a client such as MobaXterm. This adds a layer of security as XDMCP is not an encrypted protocol. All the network traffic can go over the private Azure Virtual Network.

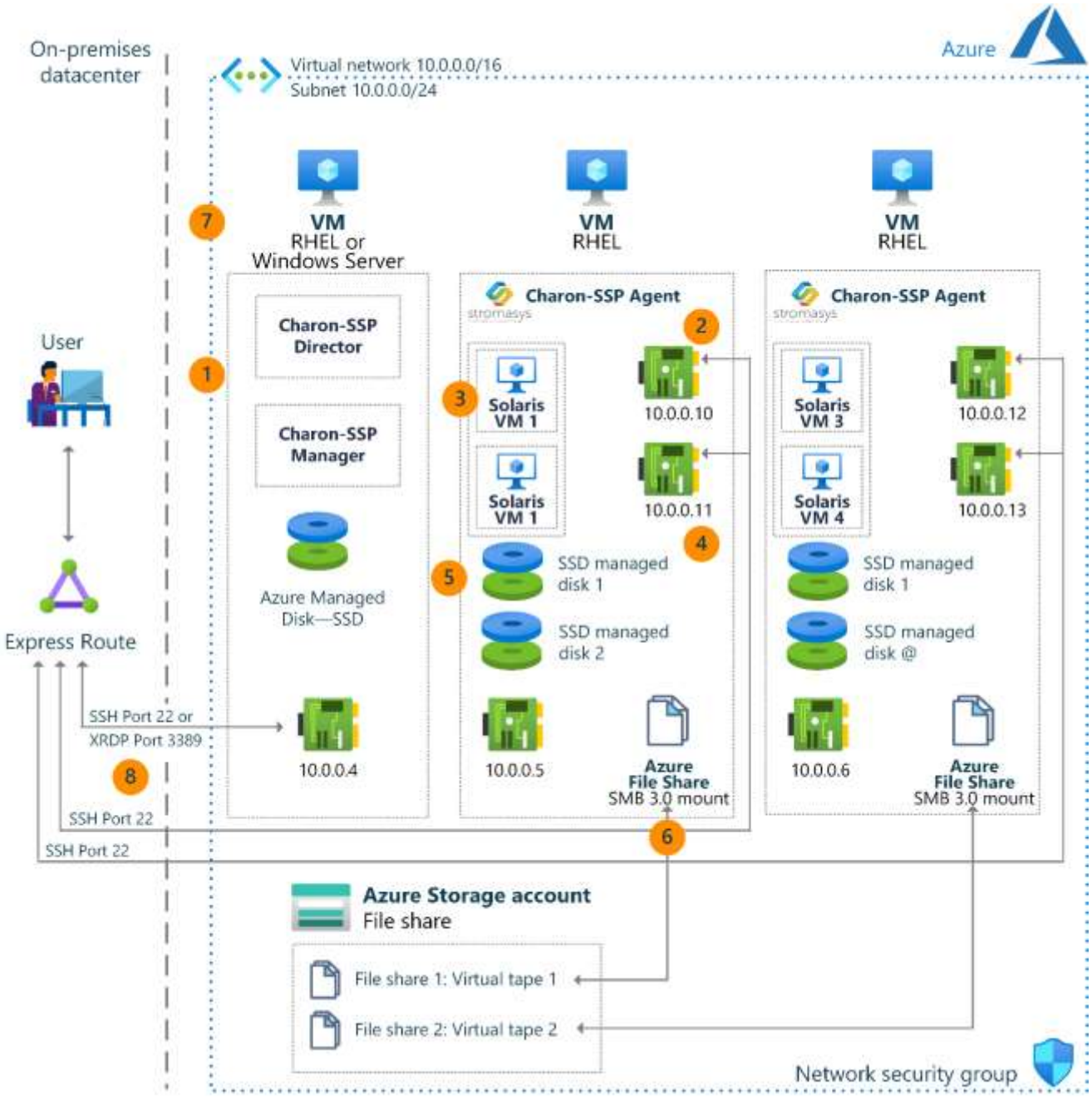


Figure 2. Architectural overview of a scale-out scenario using Charon-SSP Director and multiple Agents.

> **Note:** For the most up-to-date information about Charon-SSP, see the Stromasys Charon-SSP documentation.

2 Set up the Linux VM

This section shows you how to get started in the Azure portal and set up a VM capable of running Charon-SSP. For best performance, we recommend a compute-optimized F-Series Azure VM. For the emulated hosts, it's recommended you use one CPU core for each emulated Solaris CPU instance plus one additional CPU core for I/O.

If server just-in-time (JIT) optimization is used, add another I/O CPU core to improve the translation speed. For example, compare the following figures:

Table 3. SPARC Hardware to be emulated

SPARC	CPU	RAM
SPARC V240	2	4 GB

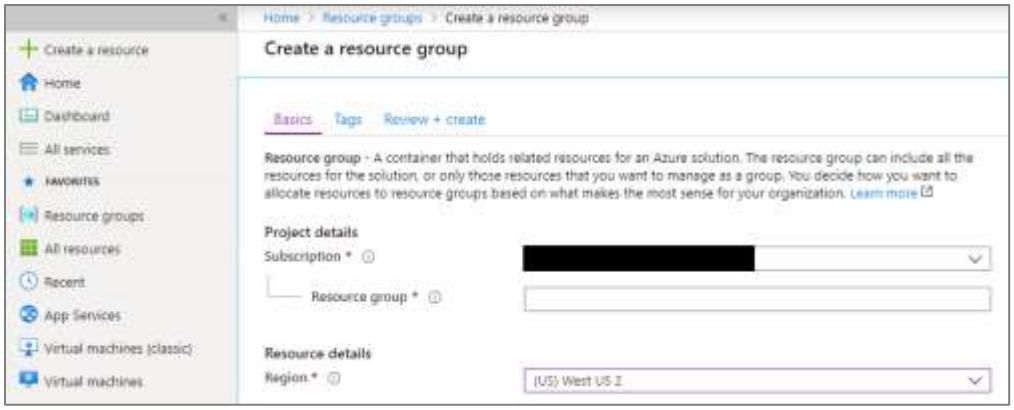
Table 4. VM Series we would select—the Azure series minimum

Instance	CPU	RAM
F-Series Linux Instance	4	8 GB

2.1 Create an Azure resource group

A resource group is a container that holds related resources for an Azure solution.

1. On your main laptop web browser, go to the Azure portal (portal.azure.com).
2. Click **Resource groups**, then click **Add**.
3. Under **Create a resource group**, select an Azure subscription, and give the resource group a name.



4. For **Region**, choose the Azure location where you plan to deploy your artifacts.
5. Click **Review + create**.

2.2 Provision Red Hat Enterprise Linux VM

This example uses Red Hat Enterprise Linux 7.6, but other operating systems (OSs) are supported. For a complete list of supported OSs, see the Stromasys Chron-SSP documentation.

> **Note:** If you use Centos 7.7, you must run the following extra command before installing Charon-SSP:
>
> `sudo yum install gtk2 -y`

1. In the portal, on the screen for the newly created resource group, click **Add**.
2. In the search box, search for **red hat**.

3. In the search results, choose the Red Hat Enterprise Linux 7.6 template shown, then click **Create**.

4. Under **Create a virtual machine**, on the **Basics** tab, use the following settings:

   - **Subscription:** Choose the subscription in which you created your new resource group.
   - **Resource Group:** Choose the resource group you just created.
   - **Virtual Machine Name:** Type a unique name for this VM to use as the host name. For example, this guide uses **jfrost-rhel-stromasys1**.
   - **Region:** Choose the geographic location where you want this VM deployed in Azure.
   - **Availability Options:** *Optional*. For now, use **No infrastructure redundancy required**.
   - **Image:** Choose Red Hat Enterprise Linux 7.6.
   - **Azure Spot Instance:** Choose **No**.
   - **Size:** Choose a VM with at least 4 CPU cores and includes Premium disk support (**Yes**). Choose enough RAM to support the guest Solaris Emulated VMs, such as 8 GB of RAM. For example, DS13-4_v2 is one option.

> **Note:** For the most up to date hardware sizing requirements for Charon-SSP, please see the Stromasys Charon-SSP documentation.

   - **Authentication Type:** Choose **Password** (unless your organization security policies require a key-pair authentication type).
   - **Username:** Choose an easy to remember username, such as **stromadmin**.
   - **Password:** Choose a password that meets your security requirements.
   - **Public Inbound Ports:** Choose **None** for now. You will set up the network security group (NSG) rules later.)

5. Click **Next: Disks** to go to the **Disks** tab.

6. For **OS disk type**, choose **Premium SSD**.

7. Under **Advanced**, choose the following settings:

   - **Use managed disks:** Yes
   - **Use ephemeral OS disk:** No

8. Click **Create and attach a new disk**. You need to create at least one managed SSD disk to attach to the VM where you will place the ISO for Solaris and create the Solaris emulation virtual disks. These disks exceed 36 GB in size.

9. Under **Create a new disk**, select the following options:

   - **Name:** Choose a unique name for this managed disk.
   - **Source type:** Choose **None** (empty disk).
   - **Size:** Choose at least 128 GB of size.

10. Click **OK**.

11. Under **Create a virtual machine**, choose the following settings:

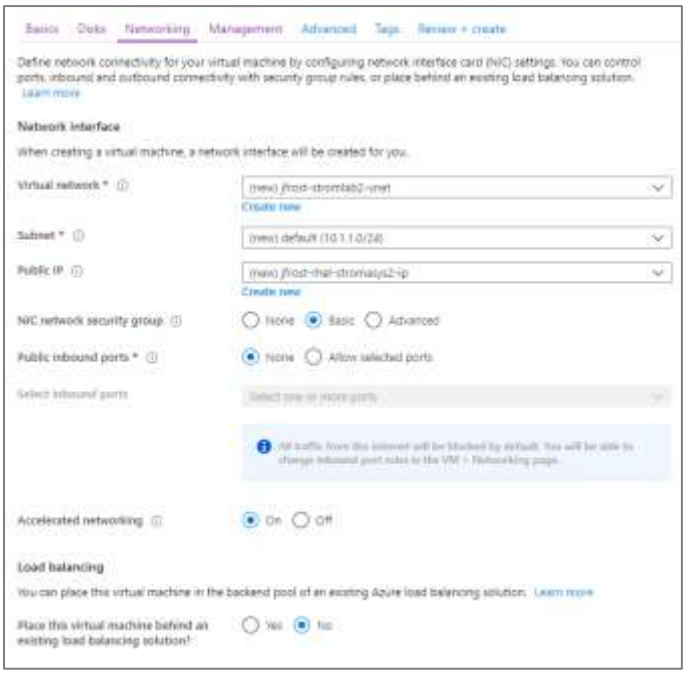
    - **Data disks / LUN:** 0
    - **Data disks / Host caching:** None

    > **Note:** The **None** setting for host caching is optimal for VM disk performance.

12. Click **Next: Networking** to go to the **Networking** tab. Choose the following settings:

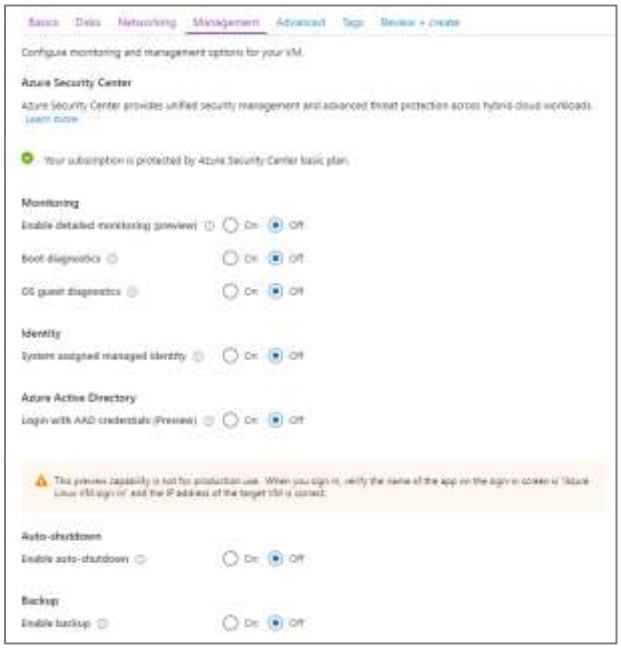
    - **Virtual network:** Either allow the wizard to create a new virtual network, or select an existing one if you have one you want to use. This guide assumes the wizard is used to create a new virtual network.
    - **Subnet:** Either allow the wizard to create a new subnet, or select an existing one if you have one you want to use. This guide assumes the wizard is used to create a new subnet named **default**.

- **Public IP:** If your organization policies allow public IPs, allow the wizard to create a new one.
- **NIC network security group:** Basic.
- **Public inbound ports:** None. (You set up the inbound ports in the NSG later.)
- **Accelerated Networking:** On
- **Load balancing:** No
- **Place this virtual machine behind an existing load balancing solution:** No

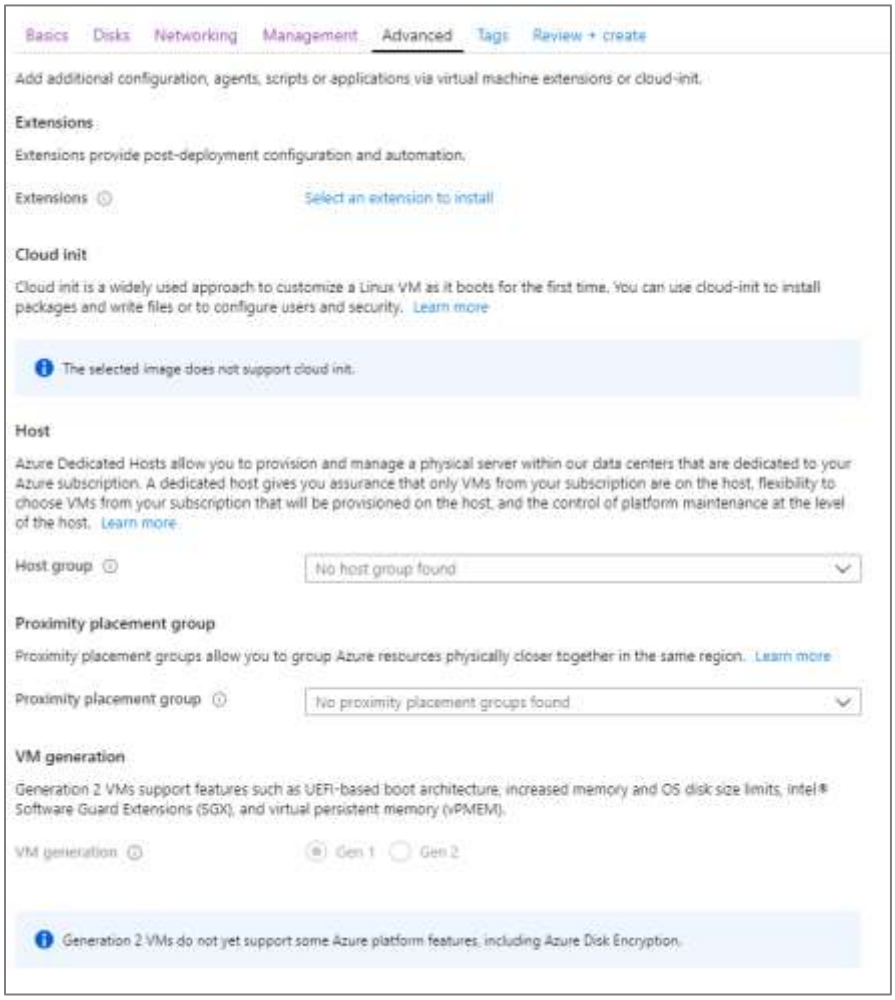


13. Click **Next: Management** to go to the **Management** tab, and use the following settings:

- **Enable detailed monitoring:** Off
- **Boot diagnostics:** Off
- **OS guest diagnostics:** Off
- **System assigned managed identity:** Off
- **Login with AAD credentials:** Off
- **Enable auto-shutdown:** Off
- **Enable backup:** Off



14. Click **Next: Advanced** to go to the **Advanced** tab. Use the default settings.



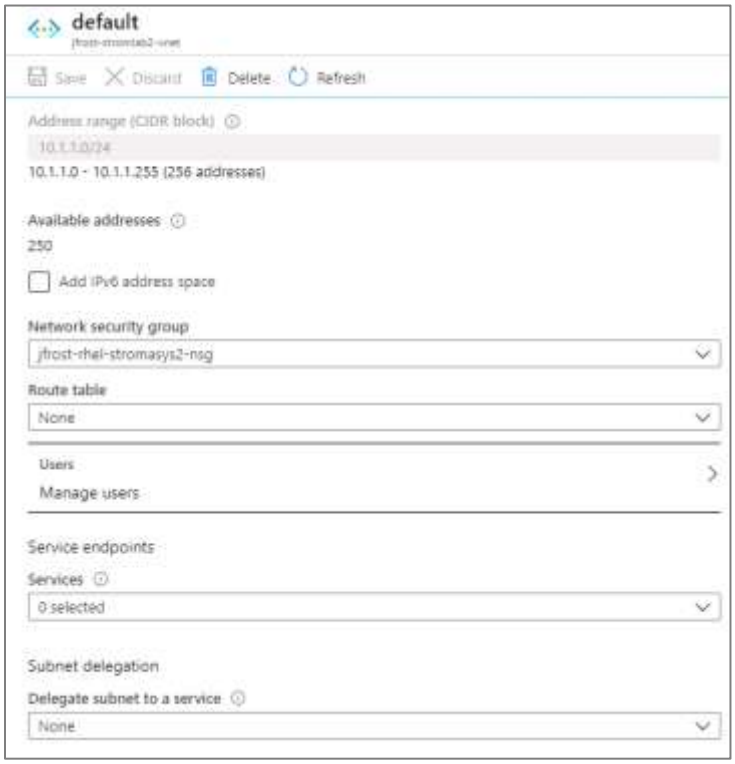
15. Click **Review + create**, then click **Create** to kick off the VM creation process.

2.3 Configure the inbound Linux VM ports

After you create the VM, you need to configure the inbound port settings so you can access the VM over SSH on port 22 using an SSH client.

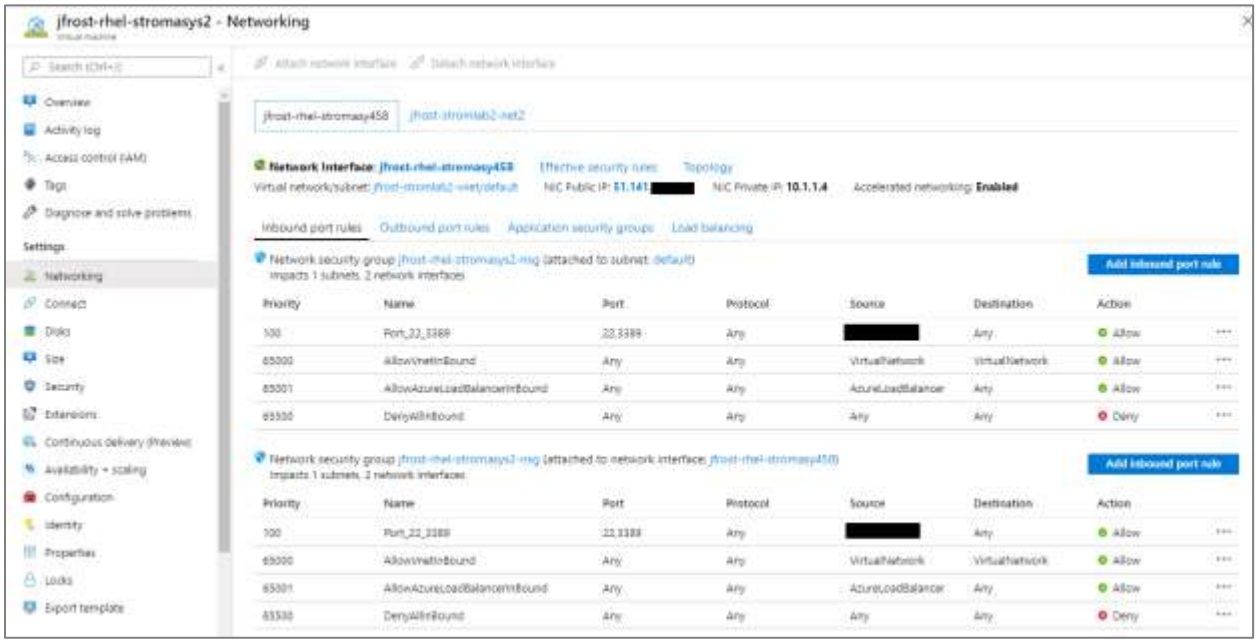
1. Go to the VM **Overview** and make a note of the **Public IP address**. You'll need this in a later step to access the VM.
2. On the portal menu, click **Networking**.
3. Click the name of the **Virtual network/subnet**.
4. On the portal menu, click **Subnets**.
5. Click the subnet associated with the VM, such as **default**.

6. For **Network security group**, choose the NSG you created earlier. The name is something like **{VM name}-nsg**.



7. Click **Save**, and the go back to the VM **Overview**.

8. Click **Networking**. The NSG appears twice because it is now associated with both the subnet and the network interface. This is normal.



9. Click one of the **Add inbound port rule button** buttons (doesn't matter which one).
10. Use the following settings for the inbound port rule:
    - **Source:** IP Addresses
    - **Source IP addresses/CIDR ranges:** Enter your internet-facing public IP address from where you are trying to connect to the VM from. To find your internet-facing public IP address, go to What is my IP address? or another website that identifies IP addresses.

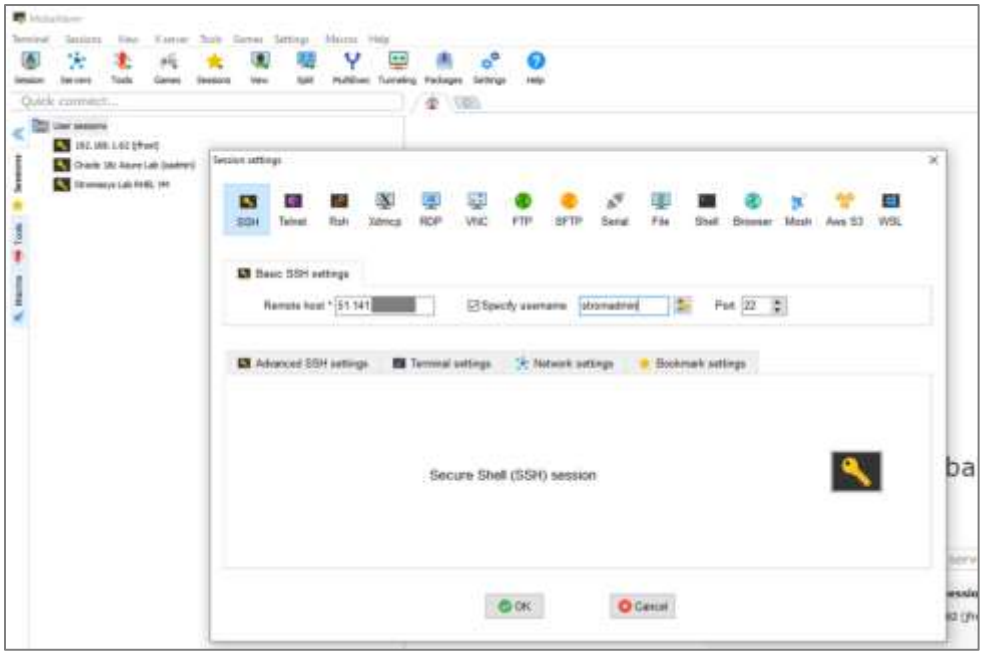
> **Note:** Make a note of your IP v4 address and enter that IP in the **Source IP addresses/CIDR ranges** box.

    - **Source port ranges:** *
    - **Destination:** Any
    - **Destination port ranges:** 22,3389
    - **Protocol:** Any
    - **Action:** Allow
    - **Priority:** 100
    - **Name:** Port_22_3389
11. Click **Add**.

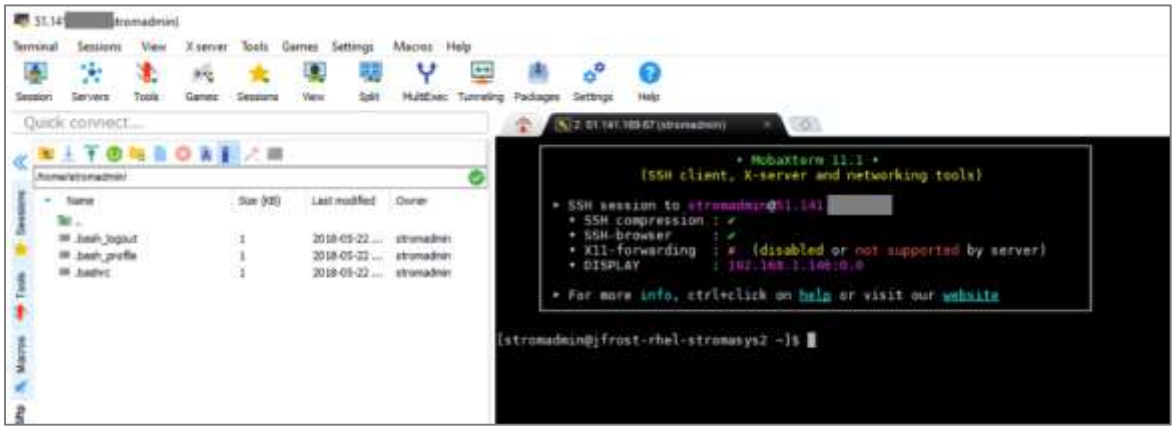
2.4 Connect to the Linux VM

After creating the rules, you can use SSH to connect to the VM. The following steps assume that MobaXterm is used, but you can use any SSH tool you like.

1. Open MobaXterm, create a new session, and select **SSH**.
2. For **Remote host**, use the VM's public IP address.
3. Check **Specify username** and use the VM administrator username.
4. Leave **Port** set to **22**, then click **OK**.



5. Enter the password you chose for the VM. If prompted to save it, choose **Yes** if you want. When connected, the bash shell prompt for the VM is displayed.



2.5 Format and mount managed disk on the Linux VM

The next step is to format and mount the managed disk you created earlier. Linux supports several different file systems for provisioning on a disk, but you can only provision one. To change the file system later, you would have to format the disk completely, so it's important to make this decision before continuing.

To optimize Linux for most workloads, XFS is a good choice. It's optimized for both large and small files and is a robust and mature 64-bit file system on Linux.

1. At the command prompt, type the **lsblk** command to see which device the managed disk is associated with. If this is the first and only managed disk, the name is typically **sdc** as shown in the last line:

```
[stromadmin@host ~]$ lsblk
NAME      MAJ:MIN   RM   SIZE   RO   TYPE   MOUNTPOINT
fd0       2:0       1    4K     0    disk
sda       8:0       0    32G    0    disk
├─sda1    8:1       0    500M   0    part   /boot
└─sda2    8:2       0    31.5G  0    part   /
sdb       8:16      0    512G   0    disk
└─sdb1    8:17      0    512G   0    part   /mnt/resource
sdc       8:48      0    1T     0    disk
```

2. Run the **sudo fdisk -l /dev/sdc** command to see the size details of the disk and to ensure that you're targeting the correct one:

```
[stromadmin@host ~]$ sudo fdisk -l /dev/sdc

Disk /dev/sdc: 1099.5 GB, 1099511627776 bytes, 2147483648 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
```

3. Choose an option to format the disk. Linux offers several choices, but **fdisk** and **parted** are typical. For disks larger than 2 TB, you must use **parted**. This guide assumes a device named **sdc** that is a 1 TB Premium SSD managed disk.

   For example, to run **parted** from the Linux bash shell, enter the following on the command line, pressing **Enter** after each one:

```
sudo parted /dev/sdc
mklabel gpt
unit TB
mkpart primary 0.00TB 1.00TB
print
quit
```

The output looks like this onscreen:

```
[stromadmin@host ~]$ sudo parted /dev/sdc

GNU Parted 3.1
Using /dev/sdc
Welcome to GNU Parted! Type 'help' to view a list of commands.

(parted) mklabel gpt
(parted) unit TB
(parted) mkpart primary 0.00TB 1.00TB
(parted) print

Model: Msft Virtual Disk (scsi)
Disk /dev/sdc: 1.10TB
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name     Flags
 1      0.00TB  1.10TB  1.10TB               primary

(parted) quit
Information: You may need to update /etc/fstab.
```

4. After the disk is formatted, create a file system on the disk. For example, if creating an XFS file system, use the **sudo mkfs.xfs /dev/sdc1** command in the Linux bash shell. This command creates a partition ***1*** on the **sdc** device called **sdc1**:

```
sudo mkfs.xfs /dev/sdc1
```

If successful, output like this appears:

```
meta-data=/dev/sdc1              isize=512    agcount=4, agsize=67108736 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=268434944, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=131071, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

5. Use the following commands at the bash shell prompt to create a folder to use as the mount point for the disk. For example, these commands create a folder named **datadrive1** and places it on the root **/** location with the appropriate permissions:

```
sudo mkdir /datadrive1
sudo chown stromadmin: /datadrive1
sudo chmod u+w /datadrive1
```

6. Mount the disk to the new folder mount point. For mounting the managed disk to the Linux operating system, use the **nobarrier** flag to disable barriers on the managed disk. This flag optimizes disk throughput.

   When barriers are enabled, the disk incurs a substantial penalty for ensuring ordering of storage and writing system metadata. However, you do not need barriers, because the writes to disks backed by Azure Premium storage are durable for these cache settings.

   The following command mounts a partition/device named **sdc1** to the folder on the root named **datadrive1** and turn off barriers:

```
sudo mount /dev/sdc1 /datadrive1 -o nobarrier
```

7. When the mounting is finished, at the bash shell prompt, use the **mkdir** command as follows to create the subfolders needed in later steps:

```
mkdir /datadrive1/downloads
mkdir /datadrive1/installs
mkdir /datadrive1/tools
mkdir /datadrive1/vm
```

8. Follow the steps below to update the system **fstab** file and enable the disk to mount automatically in case the Linux VM is rebooted. Set the **nobarrier** flag on the **fstab** file to enable the managed disk with to mount properly if rebooted. Note that the flag you use to turn off barriers differs depending on the file system used as the following tables shows.

File system	To disable barriers:	To enable barriers:
reiserFS	barrier=none	barrier=flush
ext3/ext4	barrier=0	barrier=1
XFS	nobarrier	barrier

   a. At the bash shell prompt, run the **sudo -i blkid** command to get the UUID GUID for sdc1. In the results shown, this is the number on the fourth line that start with **450388f1-**. This step ensures that the drive maps to the correct mount point. The ID is permanent, but a drive name can change after a reboot.

```
[stromadmin@host ~]$ sudo -i blkid
/dev/sdb1: UUID="0213ebb4-5206-400c-9432-8eaeab5ca1ac" TYPE="ext4"
/dev/sda1: UUID="fc265fd2-cb1e-42ce-8c37-523aa9a2b597" TYPE="xfs"
/dev/sda2: UUID="693c6905-46b0-4851-8235-fde6c1c5631d" TYPE="xfs"
/dev/sdc1: UUID="450388f1-3704-4ba6-89df-476fbec4c177" TYPE="xfs"
PARTLABEL="primary" PARTUUID="f481e8cd-9abc-4ef8-96fd-8abb109b385a"
```

b. At the bash shell prompt, use the **sudo vi /etc/fstab** command to edit the fstab file:

```
sudo vi /etc/fstab
```

For more information about using vi, a text editor for UNIX and Linux systems, use one of the many available command references, such as *A beginner's guide to editing text files with vi* and *How to use the vi editor*.

c. In the vi text editor, add a new line in the fstab file using the UUID GUID from the previous step and the **barrier**, **xfs**, and other attributes as shown. It will look something like this:

```
UUID=01299622-a943-45d9-8bc9-c941c87e0da9   /datadrive1  xfs
defaults,nofail,nobarrier   1   2
```

d. Look over your fstab file. It should now look something like this:

```
#
# /etc/fstab
# Created by anaconda on Sat Oct 26 00:47:20 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=693c6905-46b0-4851-8235-fde6c1c5631d /            xfs   defaults
0 0
UUID=fc265fd2-cb1e-42ce-8c37-523aa9a2b597 /boot        xfs   defaults
0 0
UUID=01299622-a943-45d9-8bc9-c941c87e0da9 /datadrive1  xfs
defaults,nofail,nobarrier  1 2
```

9. To verify the disk was mounted correctly and the allocated space is recognized, run the **df -h** command. It displays all disks and their available space—including the newly added 1 TB disk on /datadrive1 as shown in the last line:

```
[stromadmin@host ~]$ df -h

Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2        32G  1.5G   31G   5% /
devtmpfs         28G     0   28G   0% /dev
tmpfs            28G     0   28G   0% /dev/shm
tmpfs            28G  9.1M   28G   1% /run
tmpfs            28G     0   28G   0% /sys/fs/cgroup
/dev/sda1       497M  101M  396M  21% /boot
/dev/sdb1       111G  2.1G  103G   2% /mnt/resource
tmpfs           5.6G     0  5.6G   0% /run/user/1000
/dev/sdc1       1.0T   33M  1.0T   1% /datadrive1
```

2.6 Set up GNOME and RDP

The next step is to set up GNOME, a graphical desktop environment for Linux. This procedure also installs xRDP, a component that provides remote desktop services. You remotely connect over port 3389 to the Linux VM to interact with it using the graphical desktop environment.

A few other Red Hat Package Manager (RPM) components are needed. The following steps install TigerVNC, an implementation of VNC (Virtual Network Computing), and enables you to launch and interact with graphical applications on remote machines. EPEL (Extra Packages for Enterprise Linux) is also installed as part of this procedure.

This setup makes it much easier to interact with the Charon-SSP Manager and other tools.

> **Note:** The guide uses Linux as the host of the Charon-SSP. However, Charon-SSP 4.2.X and later enable you to run the Charon Manager on Windows.

1. Make sure you have a Red Hat Developer Network Account. Some of the **yum** installation commands require a registered version of Linux. You can get an account at no charge from Red Hat.
2. From the bash shell prompt, use the subscription manager to register this Linux instance using the following command:

   ```
   sudo subscription-manager register --username {username} --password {password} --auto-attach
   ```

   You should see output like this:

```
Registering to: subscription.rhsm.redhat.com:443/subscription
The system has been registered with ID: d632e45c-7188-432e-a576-6880102fe53c
The registered system name is: jfrost-rhel-stromasys2
Installed Product Current Status:
Product Name: Red Hat Software Collections (for RHEL Server)
Status:       Subscribed

Product Name: dotNET on RHEL (for RHEL Server)
Status:       Subscribed

Product Name: Red Hat Enterprise Linux Server
Status:       Subscribed

Product Name: Red Hat Enterprise Linux Server - Extended Update Support
Status:       Subscribed


WARNING

The yum/dnf plugins: /etc/yum/pluginconf.d/subscription-manager.conf were
automatically enabled for the benefit of Red Hat Subscription Management. If not
desired, use "subscription-manager config --rhsm.auto_enable_yum_plugins=0" to
block this behavior.
```

3. To install the EPEL RPMs that you need, run the following command:

```
sudo rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-
7.noarch.rpm
```

4. When the installation is complete, run the following command to install the Nux Dextop component:

```
sudo rpm -Uvh http://li.nux.ro/download/nux/dextop/el7/x86_64/nux-dextop-release-
0-1.el7.nux.noarch.rpm
```

5. Run the following command to install xRDP and TigerVNC Server:

```
sudo yum -y install xrdp tigervnc-server
```

When the installation is finished, the “Complete! message appears.

6. To start the xRDP service, use the following command:

```
sudo systemctl start xrdp.service
```

7. To test that the service is running, use the following command:

```
sudo netstat -antup | grep xrdp
```

> **Note:** If **netstat** is not available in your Linux distribution, you can use the following command instead: `ss -tnlp sport eq 3389`

8. In the **netstat** results shown, note that the service is listening on port 3389:

```
tcp        0      0 127.0.0.1:3350          0.0.0.0:*               LISTEN      4737/xrdp-
sesman
tcp        0      0 0.0.0.0:3389            0.0.0.0:*               LISTEN      4738/xrdp
```

9. To open the Linux firewall for port 3389 and reload it, at the prompt, use the following two commands:

```
sudo firewall-cmd --permanent --zone=public --add-port=3389/tcp

sudo firewall-cmd --reload
```

10. To set the xRDP service to start automatically when the VM restarts, use the following command:

```
chkconfig xrdp on
```

11. To install the GNOME desktop, use the following command:

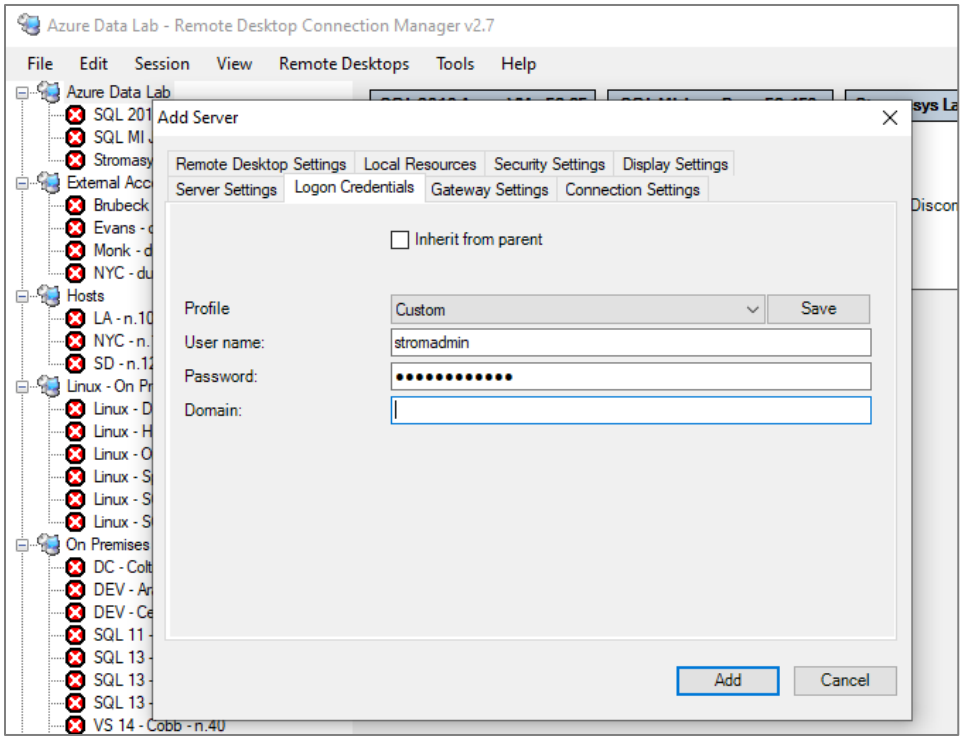
```
sudo yum -y groupinstall 'Server with GUI'
```

    When the installation is finished, the “Complete!” message appears.

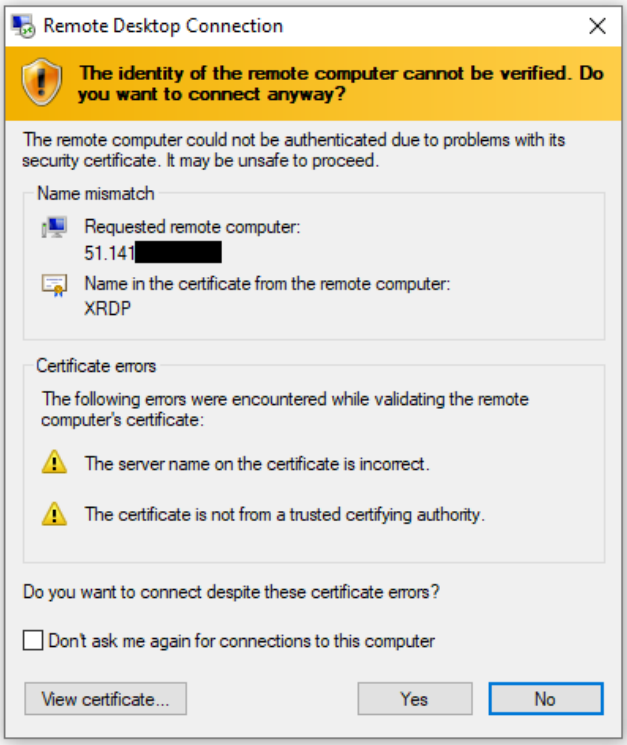
2.7 Set up RDP and connect to the Linux VM

For these steps, you can use Microsoft Remote Desktop Connection or another tool. This guide shows the steps for Remote Desktop Connection Manager 2.7.

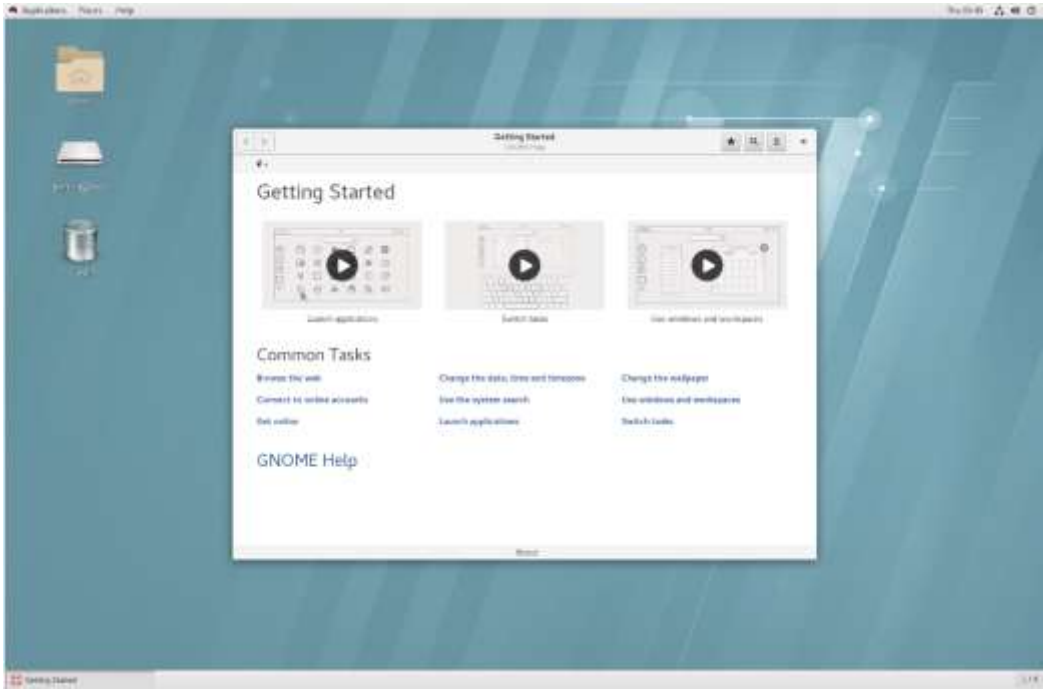
1. From a Windows computer, open Remote Desktop Connection Manager and click **Edit** > **Add Server**.
2. For **Server Name**, enter the public IP address of the Linux VM from the earlier step.
3. For **Display Name**, enter a helpful name to identify the VM.
4. Click the **Logon Credentials** tab and ensure **Inherit from parent** is unchecked.



5. For **Profile**, choose **Custom**.
6. For **User name**, enter *stromadmin* (or the username you chose earlier), then enter the Linux Admin password you chose earlier.
7. Clear the **Domain** box to make it blank.
8. Click **Add**.
9. In the list on the left, right-click the new server you just added and choose **Connect Server**. If the Remote Desktop Connection box appears, prompting you about certificate errors, click **Yes** to continue.



You should now be remotely connected to your Linux VM and see a GNOME desktop environment similar to this:



3 Set up Charon-SSP

After creating a Linux VM, you can install Charon-SSP, begin to configure it, and test it for the first time. You'll need a license for Charon-SSP, or you can get a trial license from Stromasys.

3.1 Download the Charon-SSP RPMs

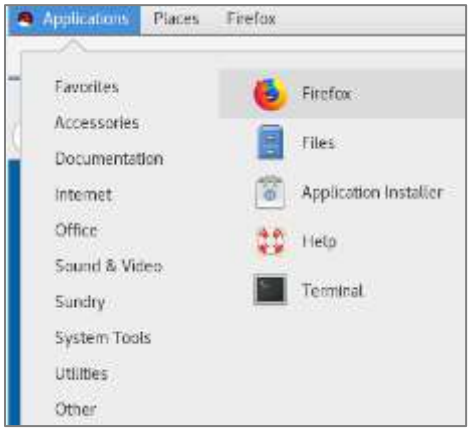
The Charon-SSP emulator consists of several RPM components that you must download and install. Before continuing, make sure to go to the Stromasys website for the latest software releases. Stromasys can also assist with providing the necessary trial license key.

The Charon-SSP for Linux suite of products consists of the following parts:

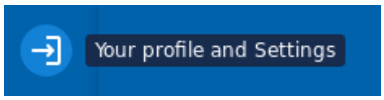
Component	Description
Kit for aksusbd	Sentinel runtime environment required for licensing the software. **Note**: In 32-bit environments, you must install the glibc.i686 component first.
Charon-SSP/4U (+)	64-bit SPARC V9 Sun-4u architecture **Note:** This emulator is not supported when running in a virtualized environment at the time this guide was written, and so you can't run it in Azure.
Charon-SSP/4V (+)	64-bit SPARC V9 Sun-4v architecture **Note:** This emulator is not supported when running in a virtualized environment at the time this guide was written, and so you can't run it in Azure.
Charon-SSP/4M	32-bit SPARC V8 Sun-4m architecture
Charon-SSP Manager	GUI-based virtual machine manager (local and remote)
Charon-SSP Director	GUI-based manager for distributed host systems running multiple virtual machines
Charon-SSP Agent	Bridge for communication between the Charon-SSP virtual machine and the Charon-SSP Manager. It enables the Charon-SSP Director to discover Charon-SSP hosts automatically. It can be configured to start of Charon-SSP virtual machines automatically at system boot.

> **Note:** The following steps require an account for access to the portal for Stromasys Authorized Partners.

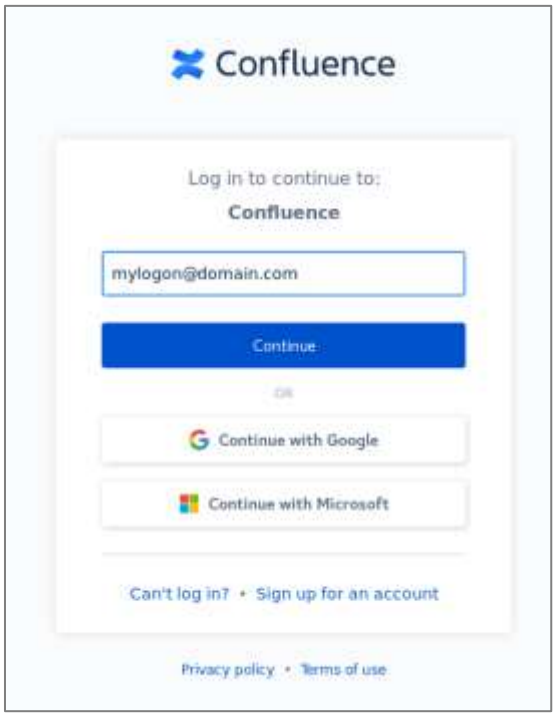
1. From your laptop, use RDP to connect to the Azure Linux VM you created in an earlier step.
2. Open the Firefox web browser in your Linux VM in the GNOME desktop environment:



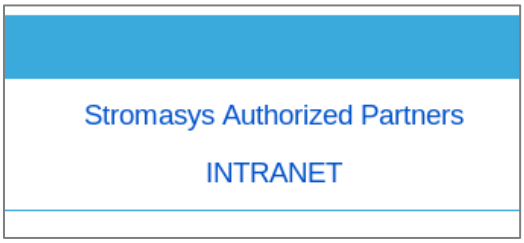
3. Go to the Stromasys portal (stromasys.atlassian.net) and click the button in the lower left:



4. Sign in using your Stromasys Partner Portal credentials.



5. On the Portal home page, click the **Stromasys Authorized Partners** link.



6. Under **Contents**, click **Download links**.
7. On the next page, next to the label **Charon-SSP(+)**, click **V4.0.4** in the **Download link** column of the table.
8. On the next page, click **InstallationKit**.
9. On the next page, click **Linux**.
10. On the next page, click **rpm**.
11. On the next page, click the link for each of the following files to download them to the **/datadrive1/downloads/** folder. To make downloading easier, you can change the Firefox settings to point all downloads to the **/datadrive1/downloads/** folder.

    aksusbd-7.63-1.i386.rpm
    charon-agent-ssp-4.0.4-x86_64.rpm
    charon-director-ssp-4.0.4.rpm
    charon-manager-ssp-4.0.4.rpm
    charon-ssp-4m-4.0.4-x86_64.rpm
    charon-ssp-4u-4.0.4-x86_64.rpm
    charon-ssp-4v-4.0.4-x86_64.rpm

3.2 Install the Charon-SSP RPMs

1. Open MobaXTerm and use SSH to connect to the Linux VM.
2. At the bash shell prompt (`[stromadmin@host downloads]$`), change the directory:

```
cd /datadrive1/downloads
```

3. Run the following commands to install each of the RPMs:

```
sudo yum -y install glibc.i686
sudo yum -y install SDL SDL-devel
sudo yum -y install aksusbd-7.63-1.i386.rpm
sudo yum -y install charon-ssp-4m-4.0.4-x86_64.rpm
sudo yum -y install charon-ssp-4u-4.0.4-x86_64.rpm
sudo yum -y install charon-ssp-4v-4.0.4-x86_64.rpm
sudo yum -y install bridge-utils
sudo yum -y install epel-release
sudo yum -y install autossh
sudo yum -y install charon-manager-ssp-4.0.4.rpm
sudo yum -y install charon-director-ssp-4.0.4.rpm
sudo yum -y install charon-agent-ssp-4.0.4-x86_64.rpm
sudo yum -y install xorg-x11-server-Xephyr
```

After running each **yum** command, the "Complete!" message at the end of the output shows that the installation was successful, like this:

```
Examining charon-agent-ssp-4.0.4-x86_64.rpm: charon-agent-ssp-4.0.4-1.x86_64
Marking charon-agent-ssp-4.0.4-x86_64.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package charon-agent-ssp.x86_64 0:4.0.4-1 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
================================================================================
=====
Package             Arch        Version      Repository
Size
================================================================================
=====
Installing:
 charon-agent-ssp   x86_64     4.0.4-1      /charon-agent-ssp-4.0.4-x86_64
6.2 M
Transaction Summary
================================================================================
=====
Install  1 Package
Total size: 6.2 M
Installed size: 6.2 M
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : charon-agent-ssp-4.0.4-1.x86_64
1/1
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/ssp-
agentd.service to /etc/systemd/system/ssp-agentd.service.
  Verifying  : charon-agent-ssp-4.0.4-1.x86_64
1/1
Installed:
  charon-agent-ssp.x86_64 0:4.0.4-1
Complete!
```

4. To set up the PATH environment variable for your bash shell profiles, do the following:

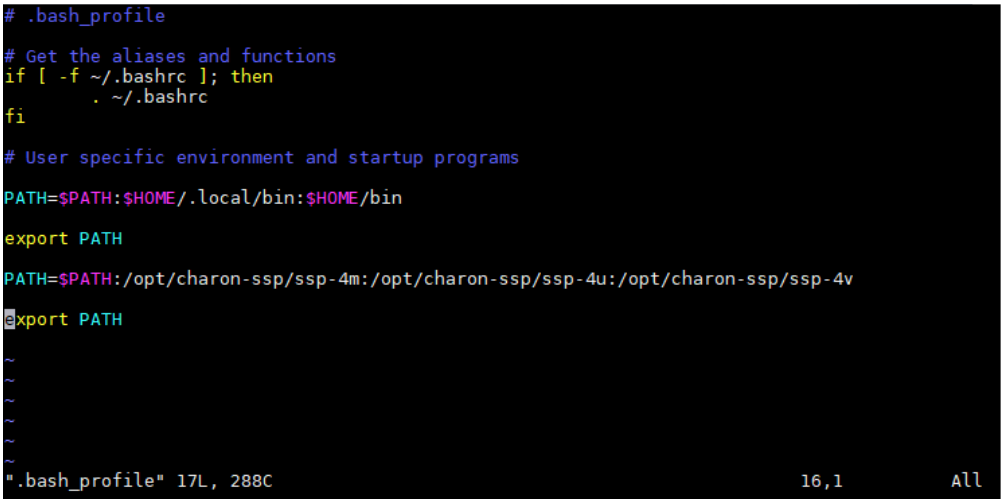
   a. As the stromadmin user, use the **cd ~** command to change to your home directory, then use vi to edit the hidden .bash_profile file:

   ```
   cd ~
   vi .bash_profile
   ```

   > **Note:** This step sets the PATH variables per user. If you want to set the variable systemwide, see the Stromasys Chron-SSP User's Guide.

   b. In vi, add the following lines at the end of the file:

   ```
   PATH=$PATH:/opt/charon-ssp/ssp-4m:/opt/charon-ssp/ssp-4u: /opt/charon-ssp/ssp-
   4v


   export PATH
   ```

   c. Save your changes and exit vi (the **wq** command).

5. To verify the changes, run the following command to see the full .bash_profile file contents:

   ```
   cat .bash_profile
   ```

   You should see contents similar to this:

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/.local/bin:$HOME/bin
export PATH
PATH=$PATH:/opt/charon-ssp/ssp-4m:/opt/charon-ssp/ssp-4u:/opt/charon-ssp/ssp-4v
export PATH
```

6. Repeat the setup of the PATH environment variable as the root user as follows:

   a. If necessary, set the root password as shown. By default, it isn't set in this RHEL 7.6 distribution.

   ```
   [stromadmin@host ~]$ sudo -s
   [sudo] password for stromadmin: ******
   [root@host ~]$ passwd root
   Changing password for user root.
   New password: ******
   Retype new password: ******
   passwd: all authentication tokens updated successfully.
   ```

   b. Use vi to edit the .bash_profile file as the root user:

   ```
   [stromadmin@host /]$ su
   Password: ******
   [root@host /]$ cd ~
   [root@host ~]$ vi .bash_profile
   ```

   c. Use vi to add the following lines at the end of the file:

   ```
   PATH=$PATH:/opt/charon-ssp/ssp-4m:/opt/charon-ssp/ssp-4u: /opt/charon-ssp/ssp-
   4v

   export PATH
   ```

   d. Verify the changes by running the **cat** command again:

   ```
   [root@host ~]$ cat .bash_profile
   ```

   e. Source the file to load up the environment:

   ```
   [root@host ~]$ source .bash_profile
   ```

7. To further verify that the PATH environmental variable has been activated, run the **echo $PATH** command to test it. You should see results like this:

```
[root@host ~]$ echo $PATH
```

```
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/stromadmin/.local/bin:/ho
me/stromadmin/bin:/opt/charon-ssp/ssp-4m:/opt/charon-ssp/ssp-4u:/opt/charon-
ssp/ssp-4v
```

3.3 Complete the Charon-SSP installation

After installing Charon-SSP, you can set up the desktop icons and configure the Sentinel License Manager for this deployment.

3.3.1 Set up desktop icons for Charon-SSP Manager and Charon-SSP Director

1. Run the **su** command to change to the root account and enter the password:

```
[stromadmin@host ~]$ su
Password: ******
```

2. To create a file named charon-ssp-manager.desktop in the **/usr/local/share/applications** folder and open it in vi, use the following command at the prompt (`[root@host ~]$`):

```
vi /usr/local/share/applications/charon-ssp-manager.desktop
```

3. Add the following contents to the file, then save your changes and exit vi:

```
[Desktop Entry]
Version=4.0.4
Name=Charon-SSP Manager
Exec=/opt/charon-manager/ssp-manager/ssp-manager
Icon=/opt/charon-manager/ssp-manager/resource/charon.png
Terminal=false
Type=Application
StartupNotify=true
Categories=System;
```

4. At the prompt, set permissions on the new file using the following two commands:

```
chmod 0644 /usr/local/share/applications/charon-ssp-manager.desktop

chown root:root /usr/local/share/applications/charon-ssp-manager.desktop
```

5. Repeat the process to create the desktop icon for Charon-SSP Director by running the following command at the prompt (`[root@host ~]$`) to create the charon-ssp-director.desktop file in the /usr/local/share/applications folder, and open it in vi:

```
vi /usr/local/share/applications/charon-ssp-director.desktop
```

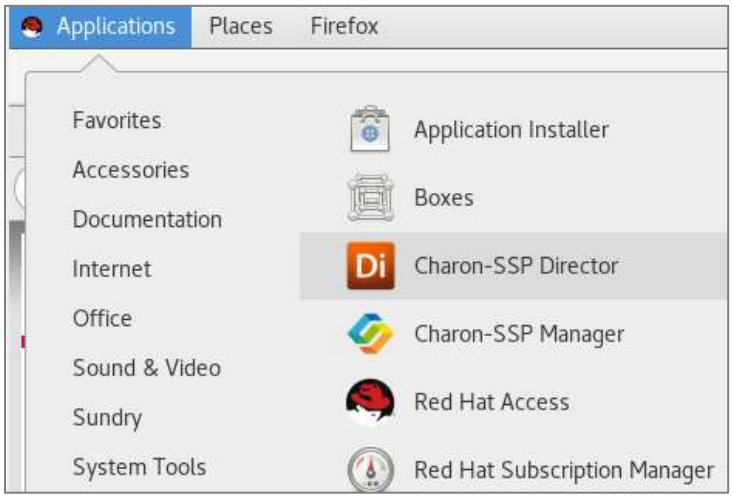
6. Add the following contents to the file, then save your changes and quit vi:

```
[Desktop Entry]
Version=4.0.4
Name=Charon-SSP Director
Exec=/opt/charon-director/ssp-director/ssp-director
Icon=/opt/charon-director/ssp-director/resource/director.png
Terminal=false
Type=Application
StartupNotify=true
Categories=System;
```

7. At the prompt (`[root@host ~]$`), set permissions on the new file using the following two commands:

```
chmod 0644 /usr/local/share/applications/charon-ssp-director.desktop

chown root:root /usr/local/share/applications/charon-ssp-director.desktop
```

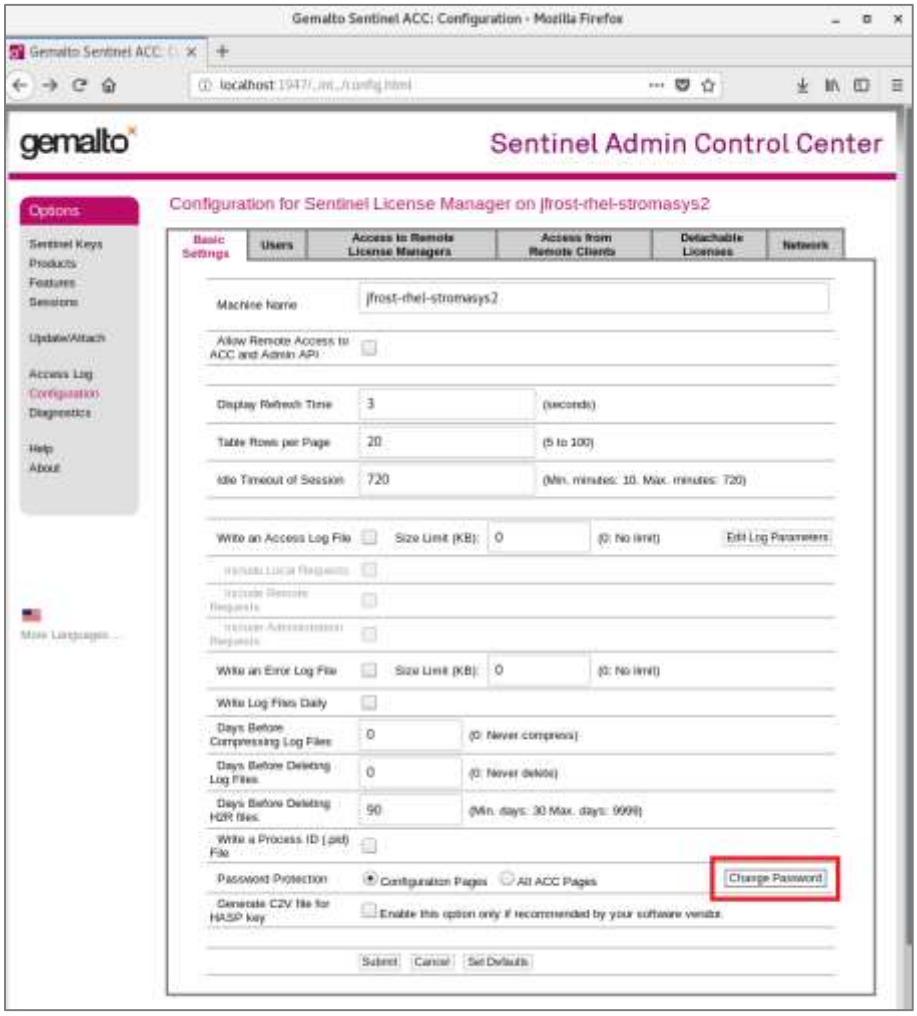
8. Use RDP to connect to the Linux VM. The Charon-SSP Manager and Charon-SSP Director icons now appear in the System Tools category:



3.3.2 Set up Sentinel HASP License Manager

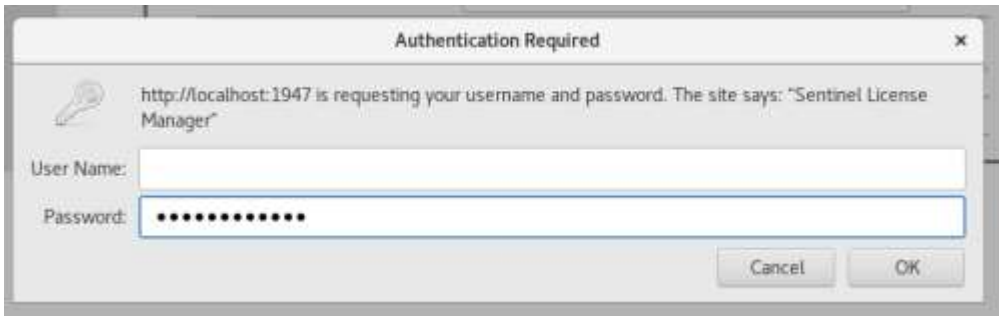
1. Open the Firefox web browser and navigate to http://localhost:1947
2. Under **Options**, click **Configuration** and choose the **Basic Settings** tab.

3. Click the **Change Password** button.



4. On the **Change Password** screen, leave the **Current Admin Password** blank. (By default there is no password.)

5. Create a new admin password, confirm it, and click **Submit**. If prompted for a username and password when you connect to the HASP GUI from a remote system, type the password and leave **User Name** blank:



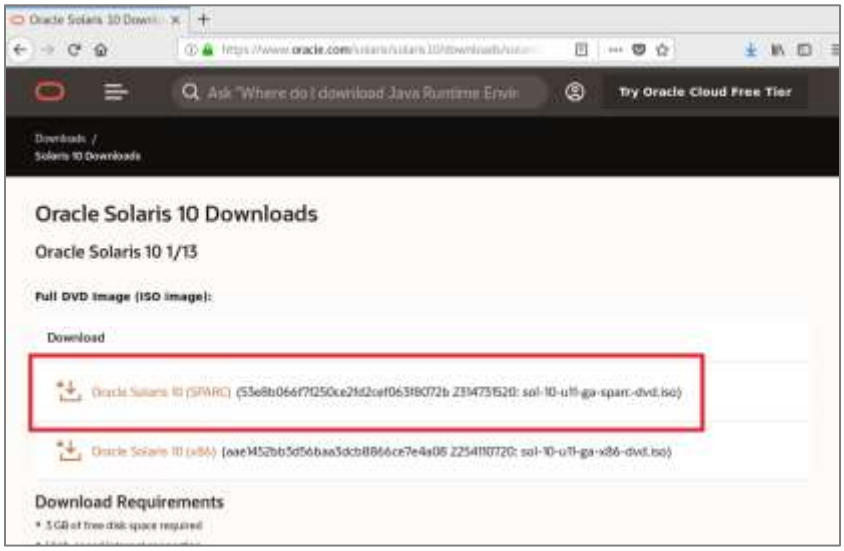
6. On the **Basic Settings** tab, under **Password Protection**, select **All ACC Pages** and click **Submit** to save this change.
7. If you want, enable remote access to the Sentinel HASP GUI. To do this, on the **Basic Settings** tab, check **Allow Remote Access to ACC** and click **Submit**.
8. Adjust the file protections for the Sentinel HASP configuration file, use the **su** command to sign on as root, then use the **chmod** command as shown:

```
[stromadmin@host ~]$ su
Password: ******
[root@host ~]$ chmod 0700 /etc/hasplm
[root@host ~]$ chmod 0600 /etc/hasplm/*
```

3.4 Download Solaris 10

To download the file in this step, you must have an Oracle developer account. You can create one at no charge on the Oracle Developers portal (developer.oracle.com).

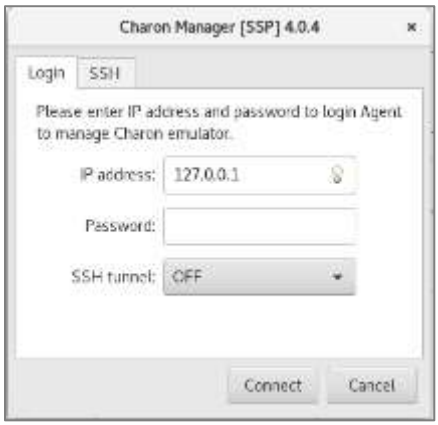
1. In the GNOME desktop, open the Firefox web browser and go to Oracle Solaris 10 Downloads.



2. When prompted, sign in.
3. Save the ISO file to the /datadrive1/downloads/ folder.

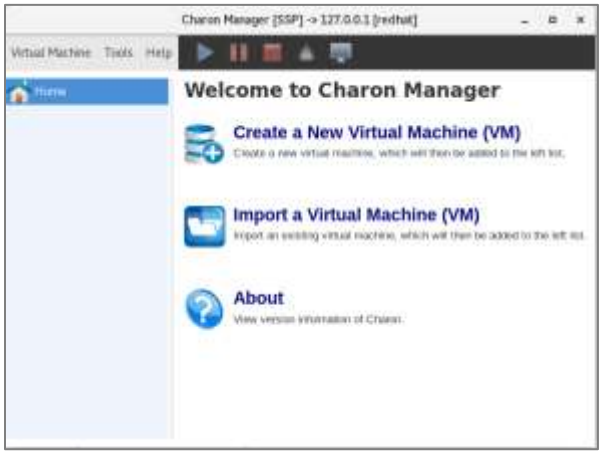
3.5 Run Charon-SSP for the first time

1. In the GNOME desktop, choose **Applications** > **System Tools** > **Charon-SSP Manager** to open Charon-SSP Manager.
2. On the **Login** tab, in the **IP address** field, use **127.0.0.**1 as the localhost IP address. Leave the password blank for now, then click **Connect**.



3. When prompted, enter a new password.

4. Click **OK**. The Charon-SSP Manager main window opens:



3.6 Set up the Charon-SSP license

You need a license to run Charon-SSP. If you're setting up a full license, it's a multi-step process. First you must generate a .c2v file in the Linux VM from the command line, then send this file to Stromasys Support. They process it and return a final .v2c file, which you must upload to the Linux VM and apply to Charon-SSP Manager. This section provides the steps.

To get a trial license, contact Stromasys (https://www.stromasys.com/contact/).

> **Note:**
>
> This section does **not** apply if you are using the Azure Marketplace template. If you are, refer to the documentation on the Charon-SSP Azure Marketplace page.
>
> To get the right license for your hardware, work with Stromasys. Stromasys can also help eliminate the process of getting the v2c file.

1. As **stromadmin**, run the following commands at the prompt (`[stromadmin@host ~]$`) to generate the .c2v file:

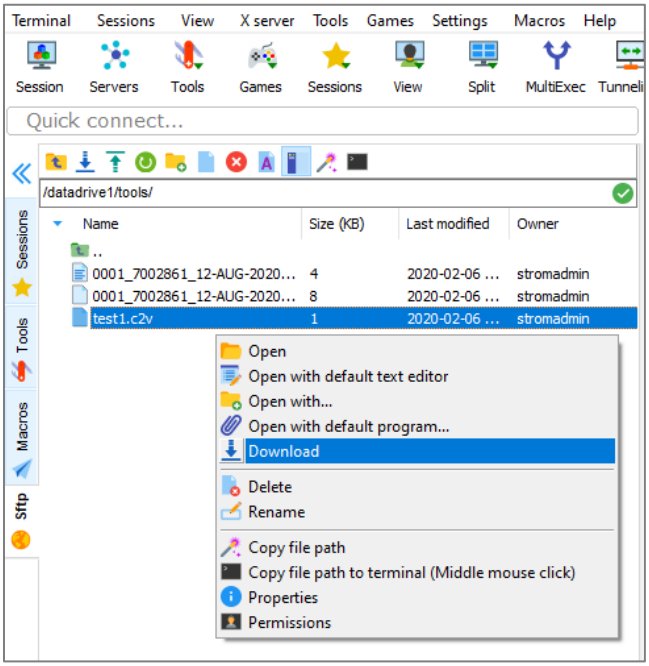
```
cd /opt/charon-agent/ssp-agent/utils/license

./hasp_srm_view -fgp /datadrive1/tools/test1.c2v
```

2. Download the .c2v file to your local machine from the VM. To do this, use the MobaXterm SFTP feature as follows:

   a. In MobaXTerm, click the **SFTP** tab.

b. Navigate to the **/datadrive1/tools/** folder.

c. Right-click the .c2v you just generated and choose **Download**.

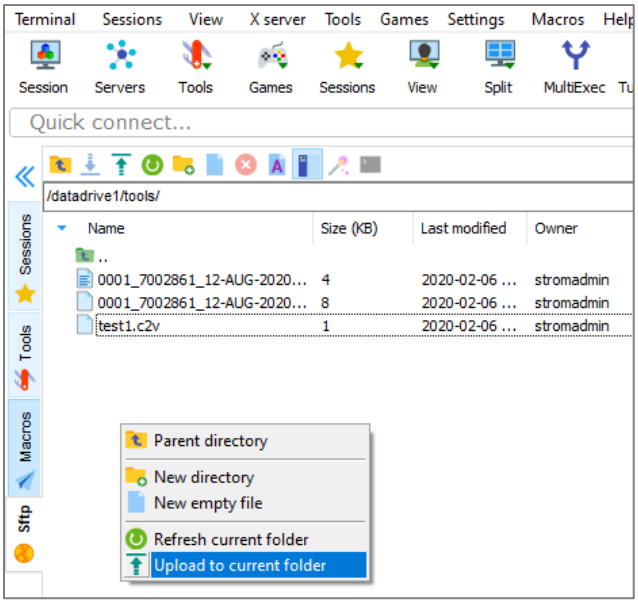


3. Send the .c2v file to Stromasys Support. When they return the .v2c file, upload it as follows:

a. In MobaXTerm, click the **SFTP** tab.

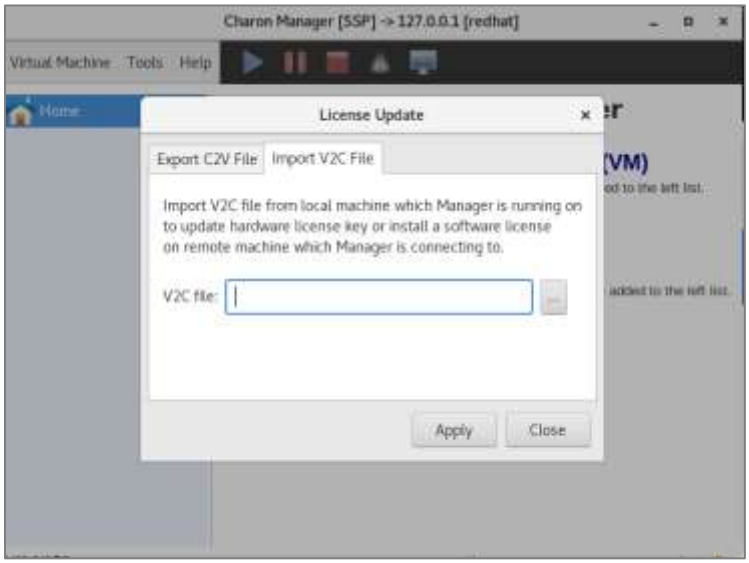
b. Navigate to the **/datadrive1/tools/** folder.

c. Right-click the white space and select **Upload to current folder**.

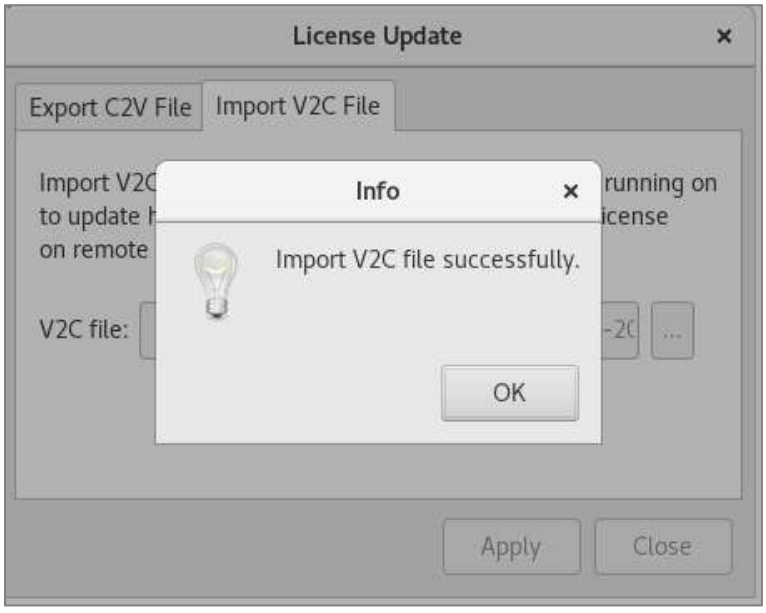


d. Select the .v2c file from your local hard disk and upload it.

4. When the .v2c file is uploaded, open Charon-SSP Manager in the GNOME desktop.
5. Go to **File** > **Tools** > **License Tools** > **License Update**.
6. In the **License Update** window, click the **...** button and select the .v2c file in the **/datadrive1/tools/** folder.



7. Click **Apply**. The following message appears:



4 Create a new VM and boot from the Solaris ISO

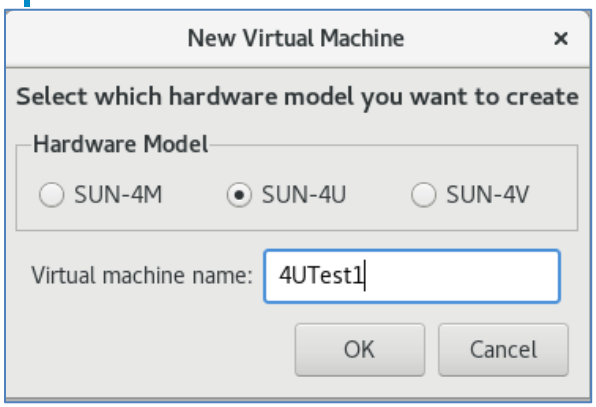
Now you can create a VM in the Charon-SSP Manager. This step requires you to create a virtual disk and mount the Solaris installation ISO as a CD-ROM drive. With that in place, you can install Solaris 10, format the virtual disk, and set up networking. To finish the networking setup, this guide creates a new user on the Solaris VM.

4.1 Configure the VM settings and virtual disks

1. Open the Charon-SSP Manager in the GNOME desktop and click **Create a New Virtual Machine (VM)**.

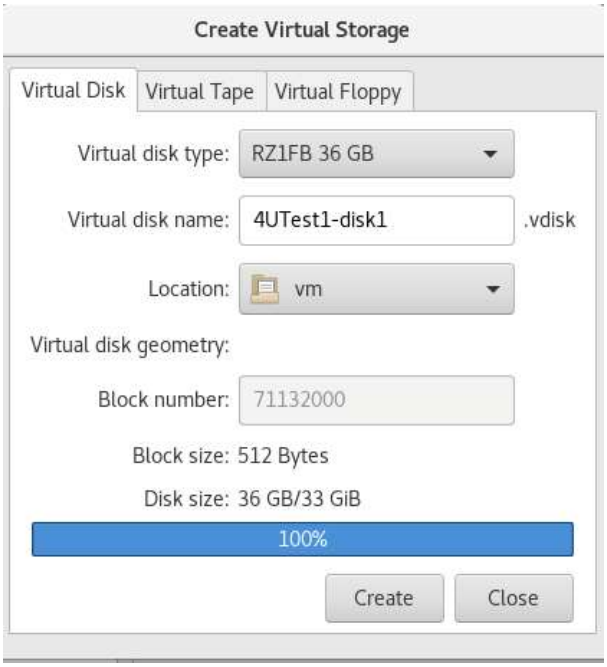
2. In the **New Virtual Machine** window, select the hardware model you want. This guide uses **SUN-4U**. Enter a VM name, such as **4UTest1**, then click **OK**.

> **Note:** Only SUN-4U and SUN-4V support Solaris 10—SUN-4M does not.



3. Go to the Charon-SSP Manager main screen. In the navigation menu, right-click **4UTest1** (or the name you entered for the VM), then choose **Virtual Machine Settings**.

4. For the **Virtual Machine Settings** options, use the following:
    - **Choose this when host Hyper-Threading is enabled or VM is in a virtual environment:** Check this box.
    - **Number of CPU:** 2 Cores.
    - **Power options:** Performance.
    - **Memory:** 16 GB.
      For details about mapping the hardware and performance requirements between the host VM and the Solaris child VM, refer to the Stromasys Charon-SSP documentation.
    - **Graphics:** Disable (uncheck).
5. In the SCSI section, click **Create Virtual Storage**. On the **Virtual Disk** tab, choose the following settings. Note that you must choose at least a 36-GB disk type for Solaris 10.
    - **Virtual disk type:** RZ1FB 36 GB
    - **Virtual disk name:** 4UTest1-disk1.vdisk
    - **Location:** /datadrive1/vm/



6. Click **Create** to provision the disk, then click **Close** to return to the **Virtual Machine Settings**.
7. Click **Add** to add the first of the two devices that are needed—the virtual disk you created.
8. In the **Add SCSI Device** window, choose the following settings:
    - **SCSI bus:** Primary SCSI Bus
    - **SCSI ID:** 1
    - **LUN ID:** 0

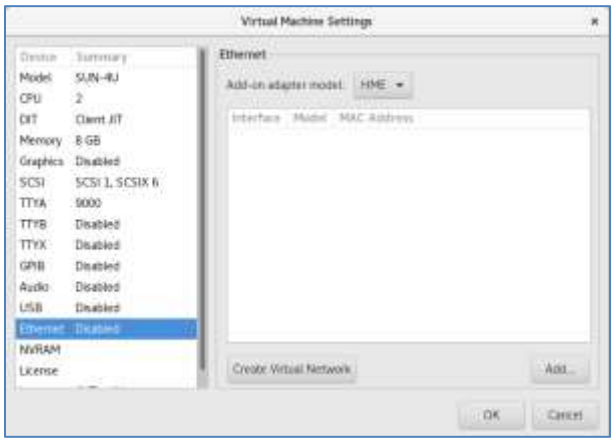
- **Removable:** OFF
- **SCSI device type:** Virtual Disk
- **SCSI device path:** /datadrive1/vm/4UTest1-disk1.vdisk

9. Click **OK**, then click **Add** to create the next device needed—a CD-ROM drive mapping to the Solaris 10 ISO you downloaded. Use the following settings:

   - **SCSI bus:** External SCSI Bus
   - **SCSI ID:** 6
   - **LUN ID:** 0
   - **Removable:** OFF
   - **SCSI device type:** Virtual CDROM
   - **SCSI device path:** /datadrive1/downloads/sol-10-u11-ga-sparc-dvd.iso

10. Click **OK** to return to **Virtual Machine Settings**.

11. In the **Device** list, click **Ethernet**. For **Add-on adapter model**, choose **HME**. Make sure that no interfaces are added to the list (that happens later), then click **OK**.

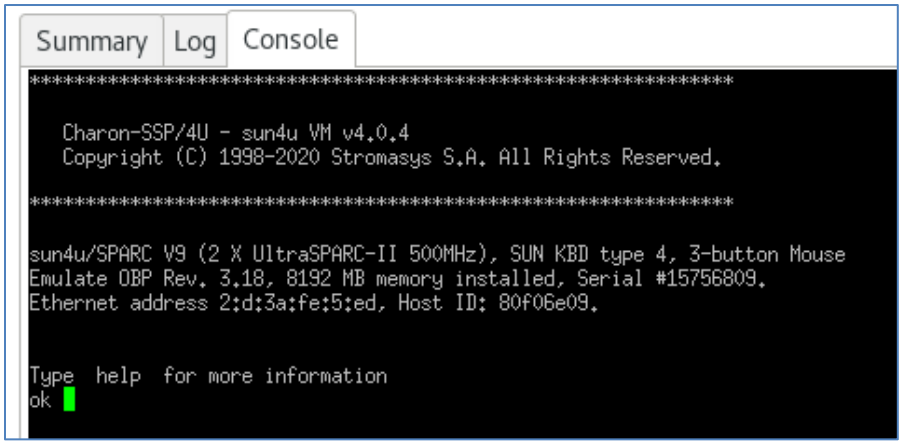


12. On the main **Charon-SSP Manager** screen, right-click **4UTest1** (or the name of your VM) and choose **Run Virtual Machine**.

4.2 Install Solaris 10 and format the virtual disk

The next step is to install the Solaris 10 distribution on the VM and the virtual disk that you provisioned in an earlier step.

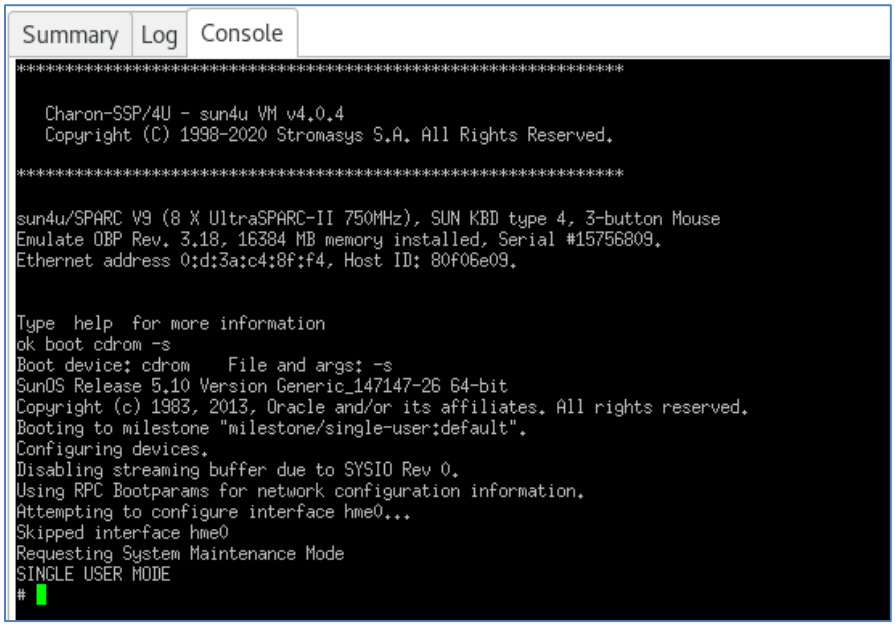
1. Start the VM. On the **Console** tab, make sure you see the Charon-SSP/4U banner and the **ok** prompt.



2. At the prompt, type the **boot cdrom -s** command and press **Enter**.

   If you want, you can also verify that the disks are preset by entering the **probe-scsi** command at the prompt.

   The **-s** parameter boots Solaris and displays a command-line prompt. This takes a few minutes to run and boot up. You should see something like this:



3. At the prompt, type **format**.
4. For **AVAILABLE DISK SELECTIONS**, enter **0**.
5. For **Label Disk Now?**, enter **y**.
6. At the **format>** prompt that now appears, type **format** and press **Enter**.
7. When asked to continue formatting, type **y** and press **Enter**. Disregard the message that says how long it takes. The format process typically takes no more than 10 minutes.

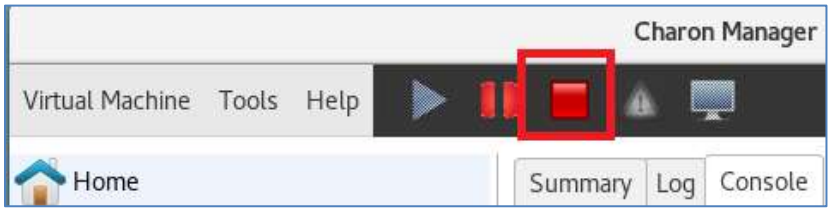
```
Solaris installation program exited.
#
#
#
#
# format
Searching for disks...done

c0t1d0: configured with capacity of 34.18GB


AVAILABLE DISK SELECTIONS:
       0. c0t1d0 <SEAGATE-ST336607LC-0200 cyl 37968 alt 2 hd 59 sec 32>
          /pci@1f,4000/scsi@3/sd@1,0
Specify disk (enter its number): 0
selecting c0t1d0
[disk formatted]
Disk not labeled.  Label it now? y


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name
        !<cmd>     - execute <cmd>, then return
        quit
format> format

Ready to format.  Formatting cannot be interrupted
and takes 3112 minutes (estimated).  Continue? y
Beginning format. The current time is Fri Feb  7 00:33:20 2020

Formatting...
done

Verifying media...
        pass 0 - pattern = 0xc6dec6de
   357/22/10
```

When the formatting is finished, the **format>** prompt is displayed like this:

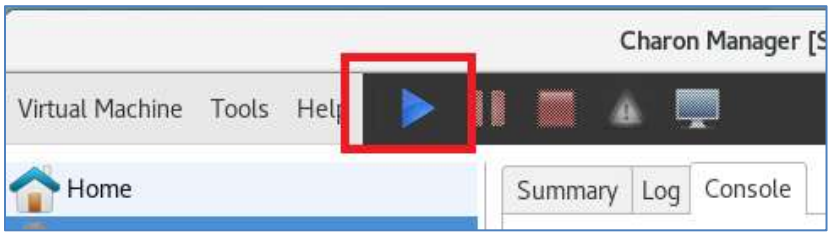
```
Formatting...
done

Verifying media...
        pass 0 - pattern = 0xc6dec6de
   17780/39/76

        pass 1 - pattern = 0x6db6db6d
   17780/39/76

Total of 0 defective blocks repaired.
format>
```

8. Type **quit** to end the format program.

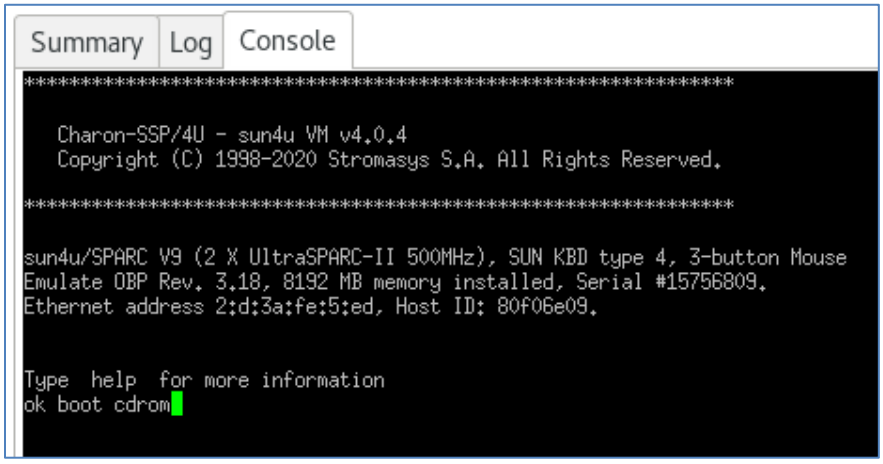
9. Click the red stop icon on the Charon-SSP Manager toolbar to stop the VM, the first step in restarting the installation process. You will go through the installation wizard again—this time, installing to a formatted disk.



10. After the VM fully stops, click the blue **Start** button in the Charon-SSP Manager toolbar to start the VM. When the VM loads, the # prompt appears.



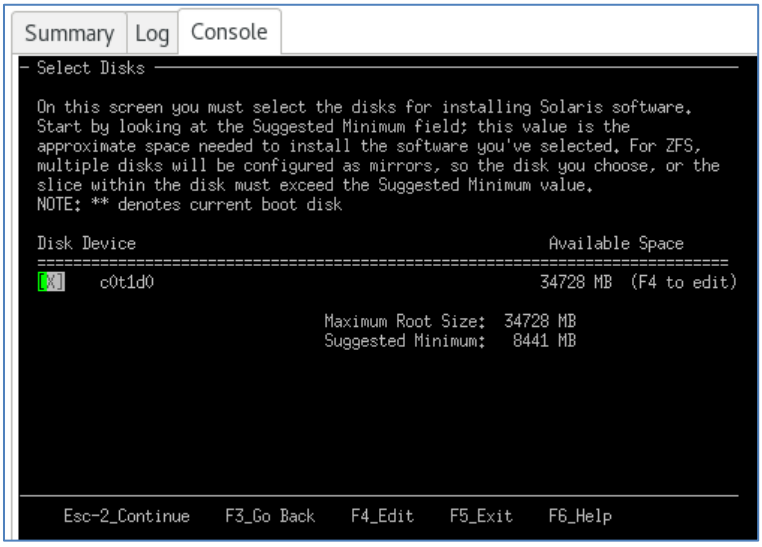
11. At the prompt, type **boot cdrom** and press **Enter** to start the installation process again from the mounted ISO virtual CD ROM.



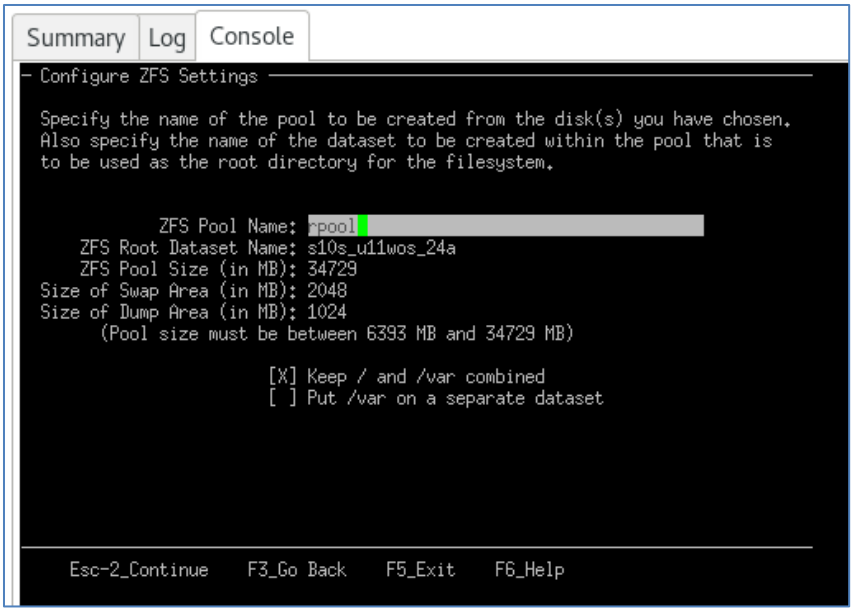
12. At the **Select a Language** question, enter the values as follows. To move to the next screen of the installer, press **ESC + F2**.

- **Select a Language:** 0 English
- **What type of terminal are you using?:** 12 (X Terminal Emulator)
- **Networked:** No (You will set up the network later.)
- **Hostname:** stromtest1
- **Continent:** Americas
- **Countries and Regions:** United States

- **Time Zone:** Pacific
- **Date and Time:** Leave the defaults.
- **Root password:** Enter an easy-to-remember password.
- **Confirm Root password:** Reenter the password.
- **Remote services enabled:** Yes
- **Installation Type (Standard or Flash):** Standard
- **iSCSI:** Install on non-iSCSI target
- **Eject a CD/DVD Automatically?:** Manually eject CD/DVD
- **Reboot After Installation?:** Manual Reboot
- **Choose Media:** CD/DVD
- **License:** Accept License
- **Select Geographic Region:** North America / U.S.A. (UTF-8)
- **Select System Locale:** U.S.A. (UTF-8)
- **Additional Product:** None
- **Choose Filesystem Type:** ZFS
- **Select Software:** Entire Distribution
- **Select Disks:** c0t1d0 (the only option)

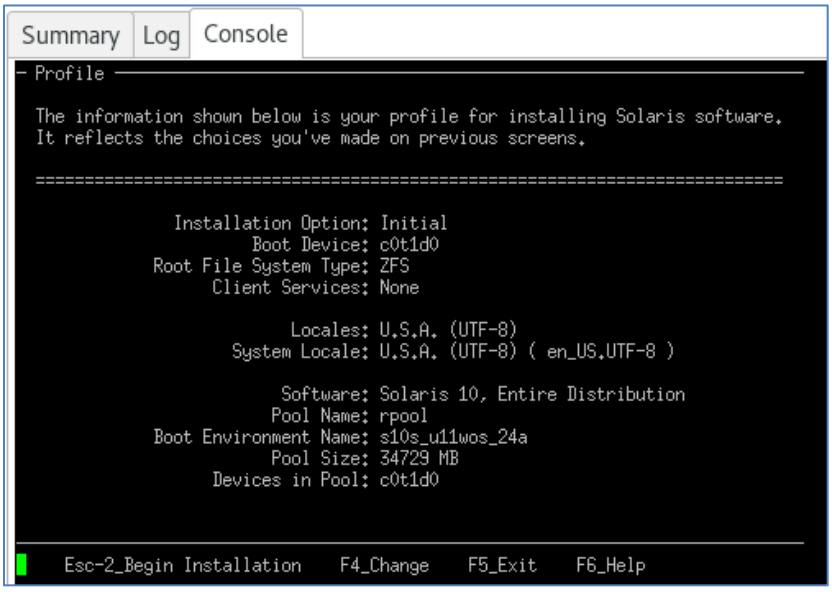


- **Preserve Data:** Continue (which means do not preserve).
- **Configure ZFS Settings:** Leave the defaults.



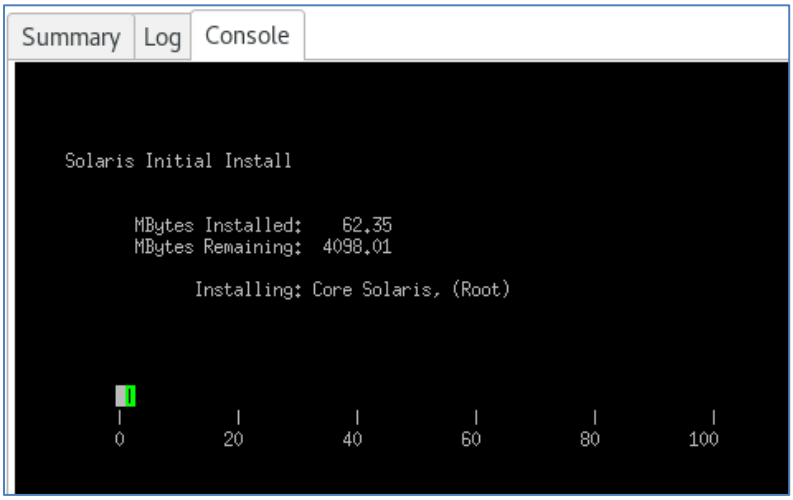
- **Mount Remote File Systems?:** Continue (which means do not mount them).

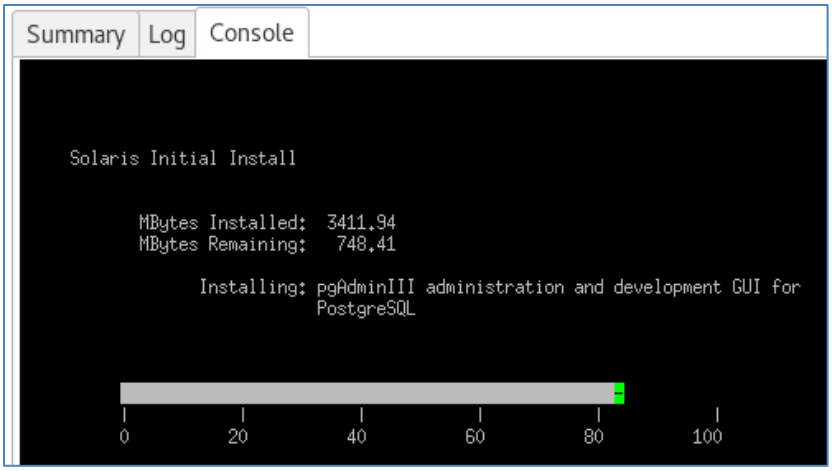
  On the next screen you should see something like the following summary. Press **ESC + 2** (or **F2**) to continue the installation.



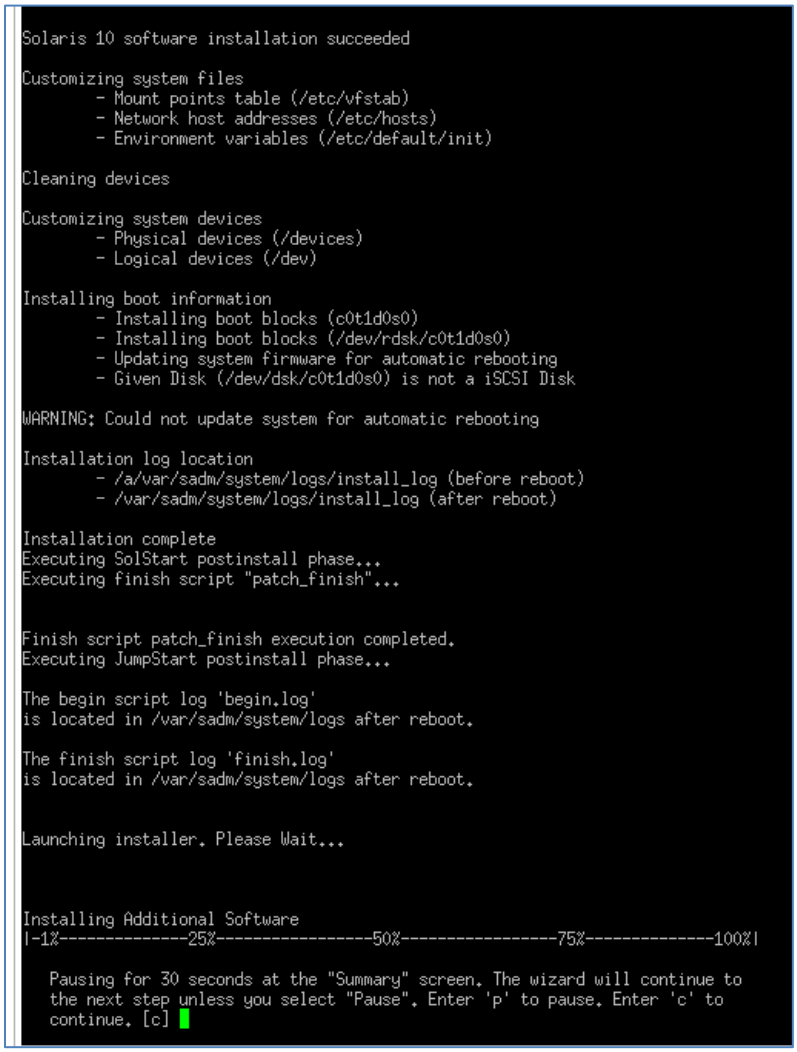
13. When a warning appears about changing the default boot device, press **ESC +2** (or **F2**) to continue. This is normal.

The installation progress screen appears. Installation time varies but typically takes approximately15 minutes.

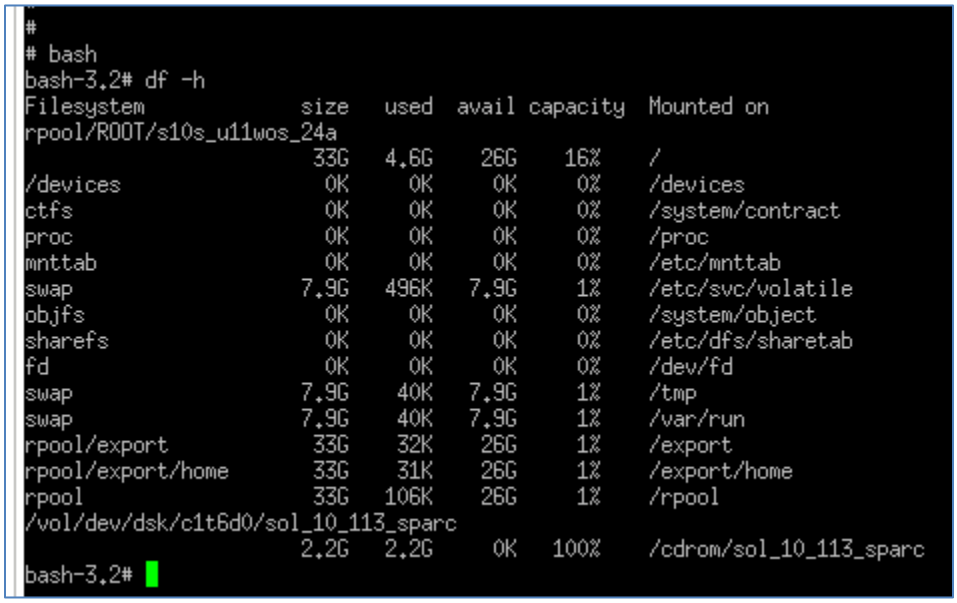




14. When the installer finishes and the wizard prompts you to continue, type **c** and press **Enter** to continue and reboot the VM.



15. When the VM reboots and you are prompted to log on, type **root** and enter the password provided for the root user during the installation process.
16. At the **#** prompt, test the installation by typing **bash** to display a bash prompt, and then typing **df -h**. You should see a screen similar this:



17. To shut down the VM so you can add a new network interface, type the following command at the bash prompt:

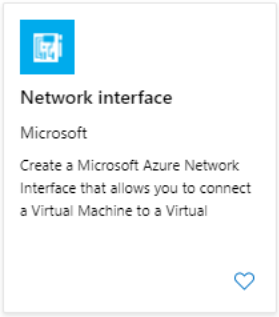
```
shutdown -i5 -y -g0
```

18. After the Solaris VM fully and safely shuts down, stop the emulator and quit the Charon-SSP Manager.

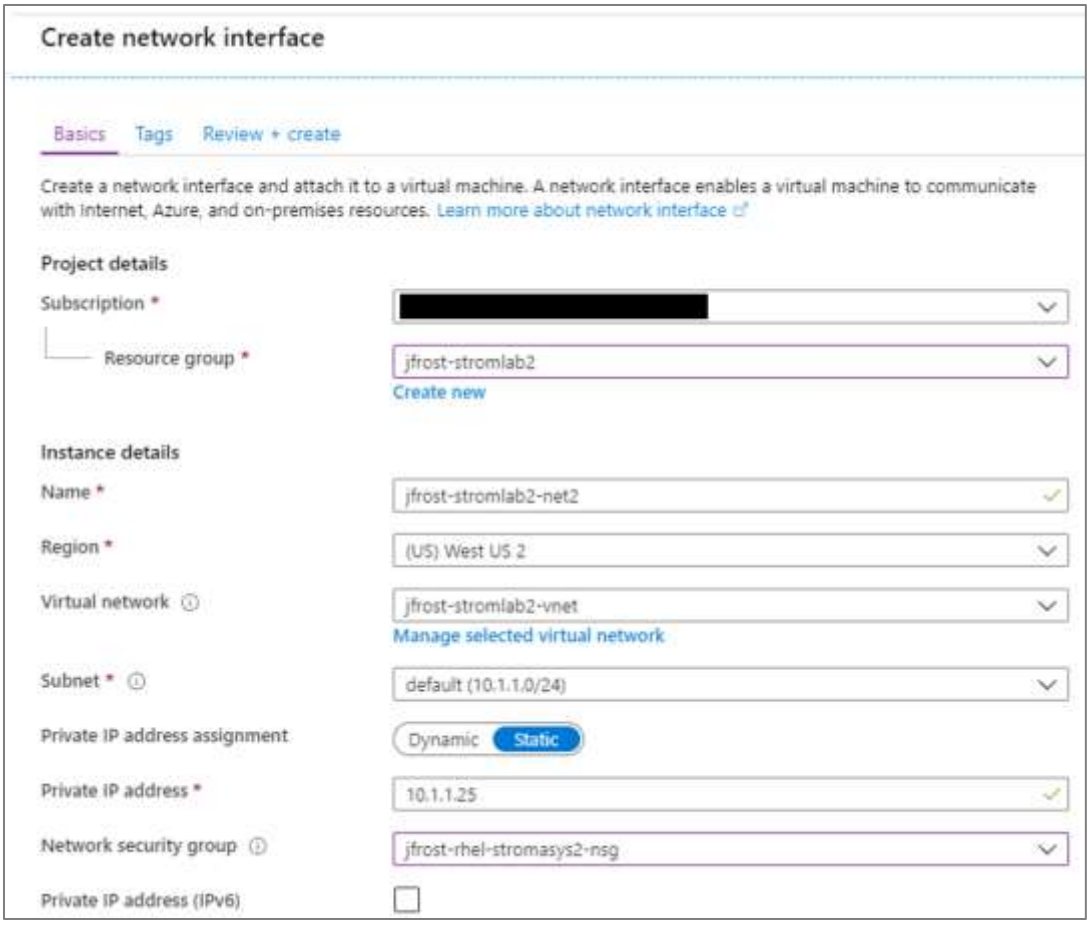
4.3 Set up networking for the Solaris VM on Azure

The next step is to create an additional network interface in Azure and attach it to your Linux VM. This network interface, and the private IP address you assign it, becomes the adapter used by the Solaris VM.

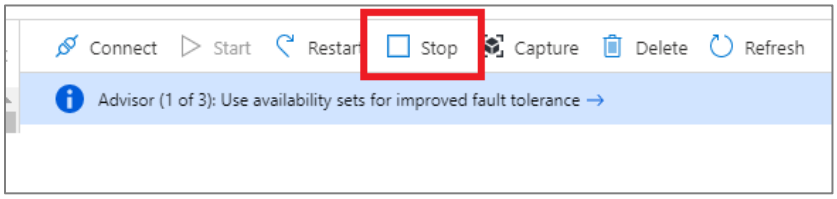
1. On your main laptop web browser, go to the Azure portal (portal.azure.com).
2. Click **Create a resource**.
3. In the search box, search for **network interface**. In the results, choose the **Network interface** option from Microsoft as shown, then click **Create**.



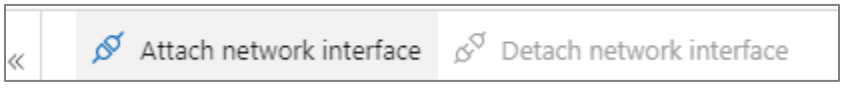
4. Use the following settings for your new network interface:
   - **Subscription:** Choose the Azure subscription you have been working in.
   - **Resource Group:** Choose the resource group you created at the beginning of this guide.
   - **Name:** Provide a unique and easy-to-reference name.
   - **Region:** Use the same region as all other artifacts you created.
   - **Virtual Network:** Choose the existing virtual network you created earlier for the Linux VM.
   - **Network Security Group:** Choose the NSG assigned to the other network interface in this same virtual network.
   - **Private IP address assignment:** Static.
   - **Private IP address:** Use an address in the same subset in the CIDR range within the subnet. For example, if the CIDR range is 10.1.1.0/24 and the current Linux VM has assigned 10.1.1.4 to its private IP, you can use 10.1.1.25 for the new network interface. *This IP address is very important*—it gets assigned to the Solaris VM.
   - **Private IP address (IPv6):** Clear the check box.



5. Click **Review + create**, check your settings, then click **Create**.
6. To attach the new network interface to the VM, shut down the Linux VM by going to its overview and clicking **Stop**.



7. If prompted to save the public IP address, select the option to save it and click **OK**.
8. After the VM fully shuts down, click **Networking** on the portal menu.
9. Under **Networking**, click **Attach network interface** on the toolbar.



10. In the **Attach network interface** box, choose the new network interface you just created and click **OK**.

11. After the network interface is successfully attached, click **Overview** to go back to the **Linux VM Overview**.

4.3.1 Set up the network

1. Click **Start** on the toolbar to start the Linux VM.
2. After the Linux VM starts, use MobaXterm and SSH to connect to the Linux VM and set up the network. First, stop the Network Manager by running the following commands at the prompt (`[stromadmin@host ~]$`):

```
sudo systemctl stop NetworkManager

sudo systemctl disable NetworkManager
```

3. On the Linux VM, run the **ifconfig -a** command to get the MAC address of the new network interface. Copy this MAC address for use in the next step (Notepad is handy for this). Its name is typically something like **eth1**, where **eth0** is the original network interface.

```
[stromadmin@host ~]$ ifconfig -a

enP41118p0s2: flags=6211<UP,BROADCAST,RUNNING,SLAVE,MULTICAST>  mtu 1500
        ether 00:0d:3a:fe:05:ed  txqueuelen 1000  (Ethernet)
        RX packets 173527  bytes 157737720 (150.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 606052  bytes 286490505 (273.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.1.4  netmask 255.255.255.0  broadcast 10.1.1.255
        inet6 fe80::20d:3aff:fefe:5ed  prefixlen 64  scopeid 0x20<link>
        ether 00:0d:3a:fe:05:ed  txqueuelen 1000  (Ethernet)
        RX packets 374982  bytes 242245199 (231.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 527624  bytes 281977560 (268.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=323<UP,BROADCAST,RUNNING,PROMISC>  mtu 1500
        ether 00:0d:3a:6e:4b:68  txqueuelen 1000  (Ethernet)
        RX packets 1937  bytes 229466 (224.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2672  bytes 251716 (245.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 391011  bytes 3400960266 (3.1 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 391011  bytes 3400960266 (3.1 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

4. Verify that the eth0 network file exists. In a later step, you will copy it to create network file for the **eth1** network, which is needed to persist the network settings in case of a reboot. To do this, first use the **su** command to switch to the root user.

```
[stromadmin@host ~]$ su
Password: ******
```

5. At the prompt (`[root@host ~]$`), change to the network-scripts directory:

```
cd /etc/sysconfig/network-scripts
```

6. At the prompt (`[root@host network-scripts]$`), run the following commands:

```
cd /etc/sysconfig/network-scripts

ls -la ifcfg*
```

   In the first line of the results, you can see **ifcfg-eth0**, verifying that the eth0 network file exists.

```
-rw-------. 1 root root 178 Feb  6 04:40 ifcfg-eth0
-rw-r--r--. 1 root root 166 Feb  5 18:04 ifcfg-eth0.bak
-rw-r--r--. 1 root root 254 Aug 24  2018 ifcfg-lo
```

7. At the prompt, run the following command to copy the eth0 network file to create the eth1 file:

```
cp ifcfg-eth0 ifcfg-eth1
```

8. Open the **eth1** file using vi so you can edit it:

```
vi ifcfg-eth1
```

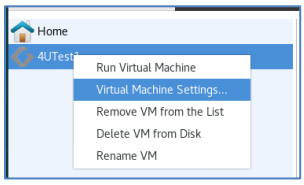
9. Edit the file to look like the following example:

```
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
USERCTL=no
PEERDNS=yes
IPV6INIT=no
PERSISTENT_DHCLIENT=yes
NM_CONTROLLED=yes
DHCP_HOSTNAME=jfrost-rhel-stromasys2
ZONE=public
```

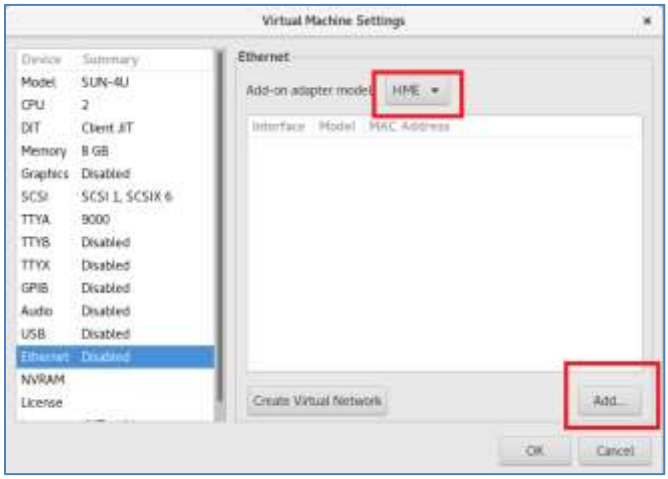
10. Save file and quit vi.

4.3.2 Edit the Solaris VM settings

1. In Charon-SSP Manager, make sure the Solaris VM is stopped so that you can edit the settings for the Solaris VM.
2. Right-click the name of the Solaris VM and choose **Virtual Machine Settings**.

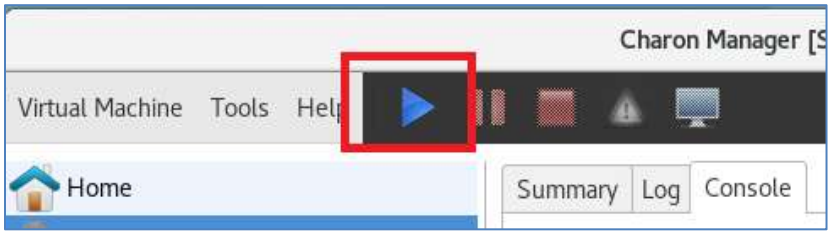


3. Under **Device**, click **Ethernet**. Make sure that **HME** is selected for the add-on adapter model, then click **Add.**



4. In the **Add Ethernet Adapter** window, choose **eth1** for the interface, check the **Set MAC address** box, then enter the MAC address you saved earlier for **eth1** on the Linux VM. Click **OK**.

5. Click **OK** to close **Virtual Machine Settings**.
6. Click the **Start VM** button to start the Solaris VM in the Charon-SSP Manager.



7. At the **ok** prompt, type **boot disk1** and press **Enter**.
8. When prompted to log on, type **root** and press **Enter**, then type the password for root specified during the Solaris installation process.
9. At the **#** prompt, type **bash** to display a bash prompt, then type **ifconfig -a** and press **Enter**. You should see something like this:

```
bash-3.2# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
bash-3.2#
```

10. To open the network interface and verify it, at the bash prompt, use the following commands:

```
ifconfig hme0 plumb
...
ifconfig -a
```

The output should look something like this:

11. To verify that the link is up, at the bash prompt, run the following command:

```
dladm show-dev
```

The output should look something like this:

4.3.3 Configure the network settings

1. To configure the IP address and subnet mask for hme0 interface, use the static private IP address you specified when creating the second network interface in the Azure portal. At the bash prompt, run the following:

```
ifconfig hme0 10.1.1.25 netmask 255.255.255.0 up
```

2. Add the gateway to the network configuration. In Azure, the gateway is usually X.X.X.1 for your subnet. For example, if your IP address is 10.1.1.25, your gateway should be 10.1.1.1. At the bash prompt, use the following command:

```
route add default 10.1.1.1
```

3. Update the permissions for the /etc/hosts file so that you can edit it and make these settings persist in case the Solaris VM is rebooted. To do this, at the bash prompt, run:

```
chmod 644 /etc/hosts
```

4. Edit the /etc/hosts file in vi to add the line **10.1.1.25 stromtest2** and remove other stromtest2 references on the other lines:

```
vi /etc/hosts
```

The hosts file should look something like this:

5. Create a file named **hostname.hme0** in the /etc/ folder:

```
vi /etc/hostname.hme0
```

6. In vi, add the following line in the new hostname.hme0 file, then save and exit vi:

```
10.1.1.25 netmask 255.255.255.0 up
```

7. At the bash prompt, create a file named **defaultrouter** in the /etc/ folder and open it in vi:

```
vi /etc/defaultrouter
```

8. In vi, add the following line in the new defaultrouter file, then save and exit vi.

```
10.1.1.1
```

9. At the bash prompt, create a file named **netmask** in the /etc/ folder and open it in vi:

```
vi /etc/netmask
```

10. In vi, add the following line in the new netmask file, then save and exit vi.

```
255.255.255.0
```

11. At the bash prompt, restart the network service to save the settings:

```
svcadm restart network
```

4.3.4 Test the network

1. On the Solaris VM, ping the network to test that it's working. At the bash prompt, run:

```
ping 10.1.1.4
```

   The "is alive" message tells you that the network communication is working.

```
10.1.1.4 is alive
```

2. To further test, on the Linux VM, start a MobaXterm SSH session.
3. At the prompt (`[stromadmin@host ~]$`), ping the network like this:

```
ping 10.1.1.25 -c 4
```

   If successful, you should see results like this:

```
PING 10.1.1.25 (10.1.1.25) 56(84) bytes of data.
64 bytes from 10.1.1.25: icmp_seq=1 ttl=255 time=0.799 ms
64 bytes from 10.1.1.25: icmp_seq=2 ttl=255 time=217 ms
64 bytes from 10.1.1.25: icmp_seq=3 ttl=255 time=0.930 ms
64 bytes from 10.1.1.25: icmp_seq=4 ttl=255 time=0.904 ms

--- 10.1.1.25 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.799/55.112/217.817/93.937 ms
```

4. To set up DNS on the Solaris VM, at the bash prompt, open the following .conf file in vi:

```
vi /etc/nsswitch.conf
```

5. Search the nsswitch.conf file for the line starting with "hosts:" and edit it, if necessary, to make sure it looks like the following, and save your changes:

```
hosts: files dns
```

6. At the bash prompt, open the following resolv.conf file in vi:

```
vi /etc/resolv.conf
```

7. Add the following line to the resolv.conf file and save your changes:

```
nameserver 8.8.8.8
```

8. Restart the network service by running the following command from the bash prompt:

```
svcadm restart network
```

9. Test it by opening Firefox in Solaris and trying to access a website by its DNS name.
10. With the network setup complete, reboot the Solaris VM to make sure the network configuration settings are intact. To reboot safely, at the bash prompt, run:

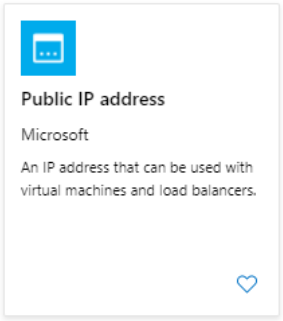
```
init 6
```

11. When the Solaris VM restarts, sign in as **root** with the root password.
12. Repeat steps 1 through 3 to ping the network. Make sure you see the same results so you know that the networking configuration is working correctly.

4.4 Set up and attach a public IP for the Solaris VM

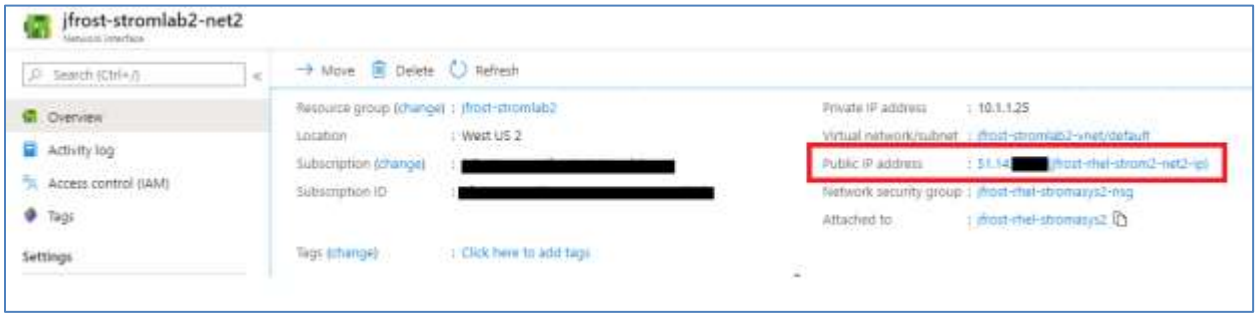
To remotely connect to the Solaris VM, you must associate a public IP address with the new network interface you created in an earlier step. You can then use SSH and a tool such as MobaXterm from your local laptop.

1. On your main laptop web browser, go to the Azure portal (portal.azure.com).
2. Click **Create a resource**.
3. In the search box, search for **public ip**. In the search results, choose the **Public IP address** option from Microsoft shown:



4. Click **Create**, then use the following settings for your new network interface:
   - **IP Version:** IPv4
   - **SKU:** Basic
   - **Name:** Enter a unique and easy-to-reference name.
   - **IP address assignment:** Static
   - **Idle timeout (minutes):** 4
   - **DNS name label:** Enter a short, unique, and easy-to-remember name.
   - **Subscription:** Choose the Azure subscription you have been working in.
   - **Resource Group:** Choose the resource group you created earlier.
   - **Location:** Choose the same region where you created the other artifacts.
5. Click **Create** to create the public IP address.
6. Click **Go to resource** to go to the overview settings for the Public IP artifact.
7. Click the **Associate** button on the toolbar, and for **Resource type**, choose **Network interface**.
8. In the **Network interface** box, choose the network Interface that you created earlier as the second Linux VM network interface. Click **OK**.

9. When the public IP address has finished its association process, go to **Network Interface Overview** and note the public IP address that is displayed on the top right. You need this for a later step.



4.5 Set up a new user in Solaris and connect

These steps create a new user that you can use to connect directly to the Solaris VM from your laptop using the public IP address. This guide uses MobaXterm with SSH.

1. Use Remote Desktop Connection Manager to connect to your Linux VM.
2. Open the Charon-SSP Manager and start the Solaris VM.
3. Sign on using the **root** user.
4. At the **#** prompt, type **bash** to display a bash prompt.
5. At the bash prompt, create a new directory:

```
mkdir /export/home/soladmin
```

6. At the bash prompt, run the following commands to create a new user called soladmin and set its password. You will use this account to connect via SSH to the Solaris VM going forward.

```
useradd -d /export/home/soladmin -m soladmin
passwd soladmin
```

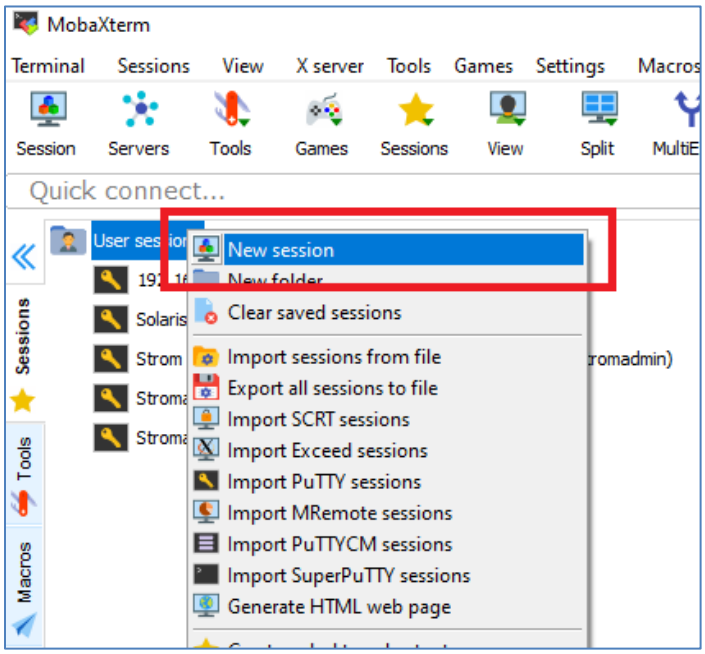
7. To test SSH and the new user, open a terminal in your Linux VM by choosing **Applications** > **Favorites** > **Terminal**.
8. At the prompt (`[stromadmin@host ~]$`), create an SSH connection and enter your password:

```
ssh soladmin@10.1.1.25
```

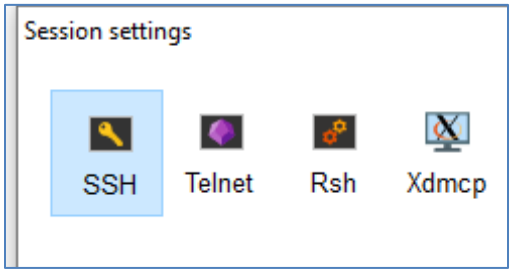
9. At the prompt, run the **df -h** command to display the disk resources so that you can verify you're seeing the Solaris VM, not the Linux VM:

```
$ df -h
Filesystem             size   used  avail capacity  Mounted on
rpool/ROOT/s10s_u11wos_24a
                        33G   4.6G    26G    16%    /
/devices                 0K     0K     0K     0%    /devices
ctfs                     0K     0K     0K     0%    /system/contract
proc                     0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
swap                   8.1G   416K   8.1G     1%    /etc/svc/volatile
objfs                    0K     0K     0K     0%    /system/object
sharefs                  0K     0K     0K     0%    /etc/dfs/sharetab
fd                       0K     0K     0K     0%    /dev/fd
swap                   8.1G    40K   8.1G     1%    /tmp
swap                   8.1G    40K   8.1G     1%    /var/run
rpool/export            33G    32K    26G     1%    /export
rpool/export/home       33G    35K    26G     1%    /export/home
rpool                   33G   106K    26G     1%    /rpool
/vol/dev/dsk/c1t6d0/sol_10_113_sparc
                        2.2G   2.2G     0K   100%    /cdrom/sol_10_113_sparc
```

10. On your laptop, open MobaXterm. On the **Sessions** tab, right-click **User sessions** and choose **New session**.

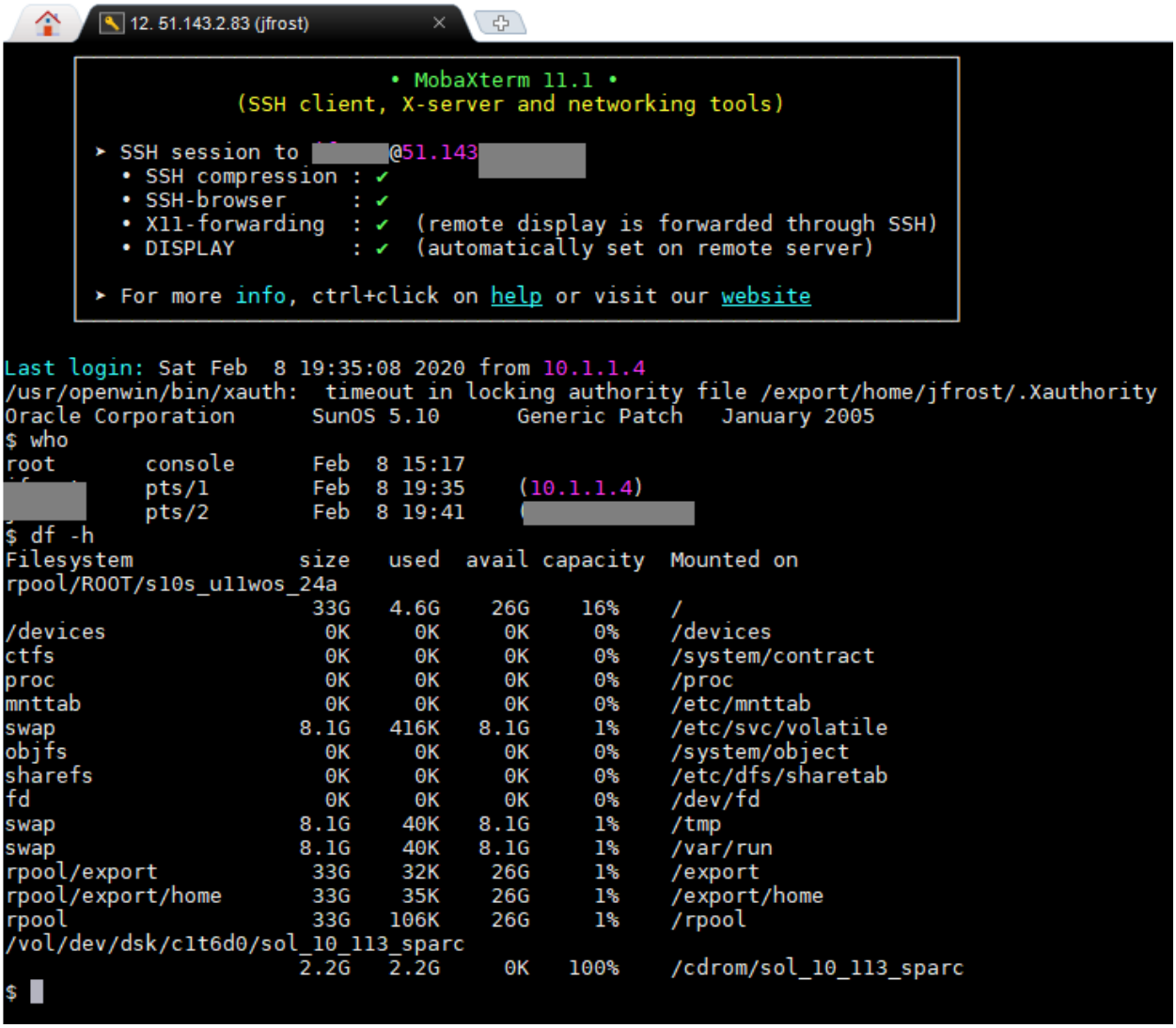


11. In the **Session settings** window, click **SSH**.



12. In the **Basic SSH settings** section, for the **Remote host** field, enter the newly create public IP address you copied earlier.
13. Select the **Specify username** check box and enter **soladmin**, the username you created earlier.
14. Leave port 22 as is.
15. Click the **Bookmark settings** tab. In the **Session name** field, enter a unique name for this session to use in a later step, then click **OK**.
16. In the MobaXterm session window that appears, enter the password for the **soladmin** user when prompted. If asked to save the password, click **Yes**.

17. Run the **who** and **df -h** commands to make sure you are on the Solaris VM. You should see a session like this, confirming that you have successfully set up the network, created a new user, and accessed the Solaris VM from your local laptop:



5 Set up graphic device emulation and remote access via XDMCP on the Solaris VM

This section shows you how to configure the Solaris VM in Charon-SSP to emulate a graphical desktop console—either the Common Desktop Environment (CDE) or Java Desktops. You must also configure the Solaris VM to allow access over XDMCP connections. XDMCP provides remote access to the Solaris VM and gives you a graphic desktop over a remote connection.

For added security and simplicity, a Windows VM in Azure is used as a hop server that connects to the Solaris VM in Azure over XDMCP.

5.1 Set up graphical device emulation

1. On your Linux VM, use Remote Desktop Connection Manager to create an RDP connection to the VM.
2. If not already open, open the Charon-SSP Manager.
3. If the Solaris VM is still running, shut it down. To do so safely, at the bash prompt, use the **shutdown -i5 -y -g0** command.

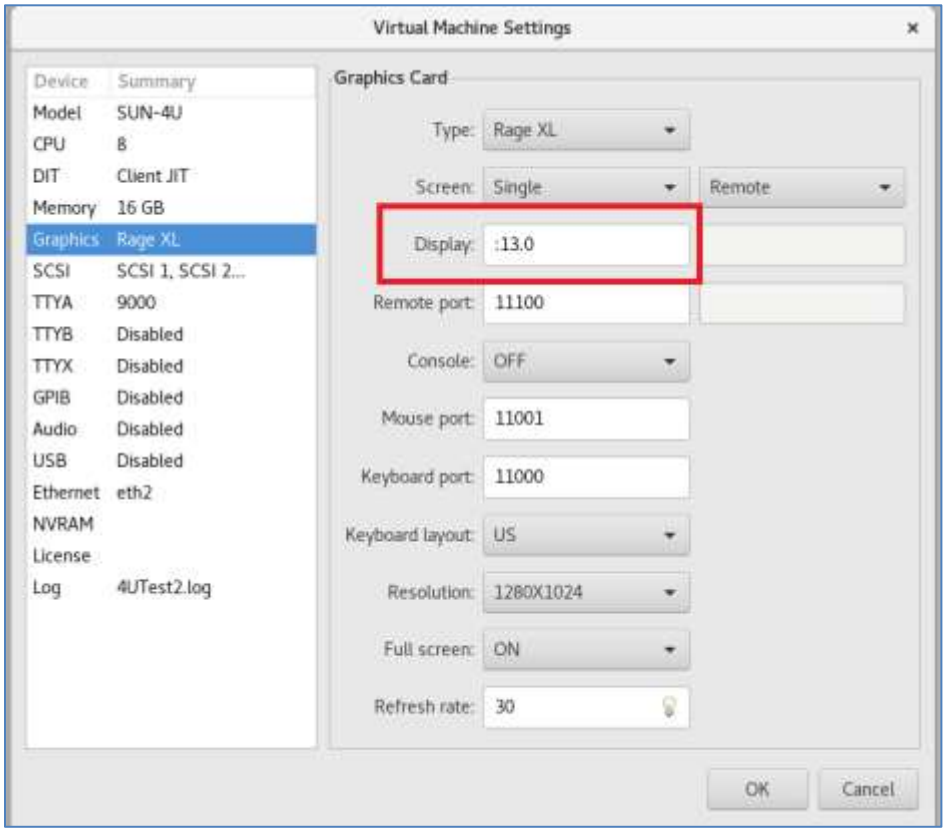
```
bash-3.2# shutdown -i5 -y -g0
Shutdown started. Wed Mar 4 PST 2020

Changing to init state 5 - please wait
Broadcast Message From root (console) on stromtest3 Wed Mar 4
THE SYSTEM stromtest3 IS BEING SHUT DOWN NOW ! ! !
Log off now or risk your Files being damaged

bash-3.2# svc.startd: The system is coming down. Please wait.
svc.startd: 104 system services are now being stopped.
Mar 4 11:08:35 stromtest3 syslogd: going down on signal 15
```

4. To make sure the emulator picks up the configuration changes, click the **Stop** button.
5. On the Linux VM, open a terminal window and run the **echo $DISPLAY** command to get the DISPLAY environmental variable value. Make a note of the first number that's shown. You need this for the next step.

```
[stromadmin@host ~]$ echo $DISPLAY
:13:0
```

6. In the Charon-SSP Manager menu, right-click **Solaris VM** and choose **Virtual Machine Settings**.

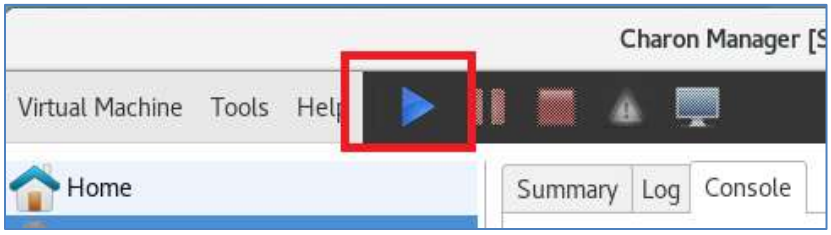
7. Under **Device**, choose **Graphics**, then edit the **Display** option to match the DISPLAY variable returned in the previous step. For example, if the value was **:13:0**, enter that.



8. Make sure the following settings are used, then click **OK**:

- **Type:** Rage XL
- **Screen:** Single, Remote
- **Console:** OFF

9. In the Charon-SSP Manager window, click the **Start** button for the Solaris VM.



10. When the Solaris VM is up, at the **ok** prompt type **boot disk1 –r** and press **Enter**. The **–r** argument ensures that Solaris configures the graphics device.
11. When prompted, sign on as **root** and press **Enter**, then use the password for root specified during the Solaris installation process.
12. At the prompt, run the **bash** command to get the bash shell.
13. At the bash prompt, copy the Xservers file from the /usr/dt/config/ folder to the /etc/dt/config/ folder by running the following commands:

```
mkdir /etc/dt

mkdir /etc/dt/config

cp /usr/dt/config/Xservers /etc/dt/config/Xservers

cd /etc/dt/config
```

14. To check which framebuffer devices are available, at the bash prompt, list all the devices:

```
ls -l /dev/fb*
```

15. Note the one that looks like **/dev/fb0** or similar. You need this information to edit the Xservers file.

```
bash-3.2# ls -l /dev/fb*
lrwxrwxrwx  1 root root  8 Feb 22 21:55 /dev/fb0 -> fbs/m640

/dev/fbs:
total 1
lrwxrwxrwx  1 root root  42 Feb 22 21:55 m640 ->
../../devices/pci@1c,2000/SUNW,m64B@1:m640
#
```

16. Use vi to edit the Xservers file and update it based on the information from the previous step:

```
vi /etc/dt/config/Xservers
```

17. At the end of the file, find this line:

```
:0   Local local_uid@console root /usr/X11/bin/Xserver :0 -nobanner
```

Replace it with the following line using the information from the previous steps.

```
:0   Local local_uid@console root /usr/openwin/bin/Xsun :0 -dev /dev/fb0
```

18. Save the file and quit vi (press **Esc**, type **wq** and press **Enter**).
19. At the bash prompt, configure the cde-login service by running:

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

svcadm restart cde-login
```

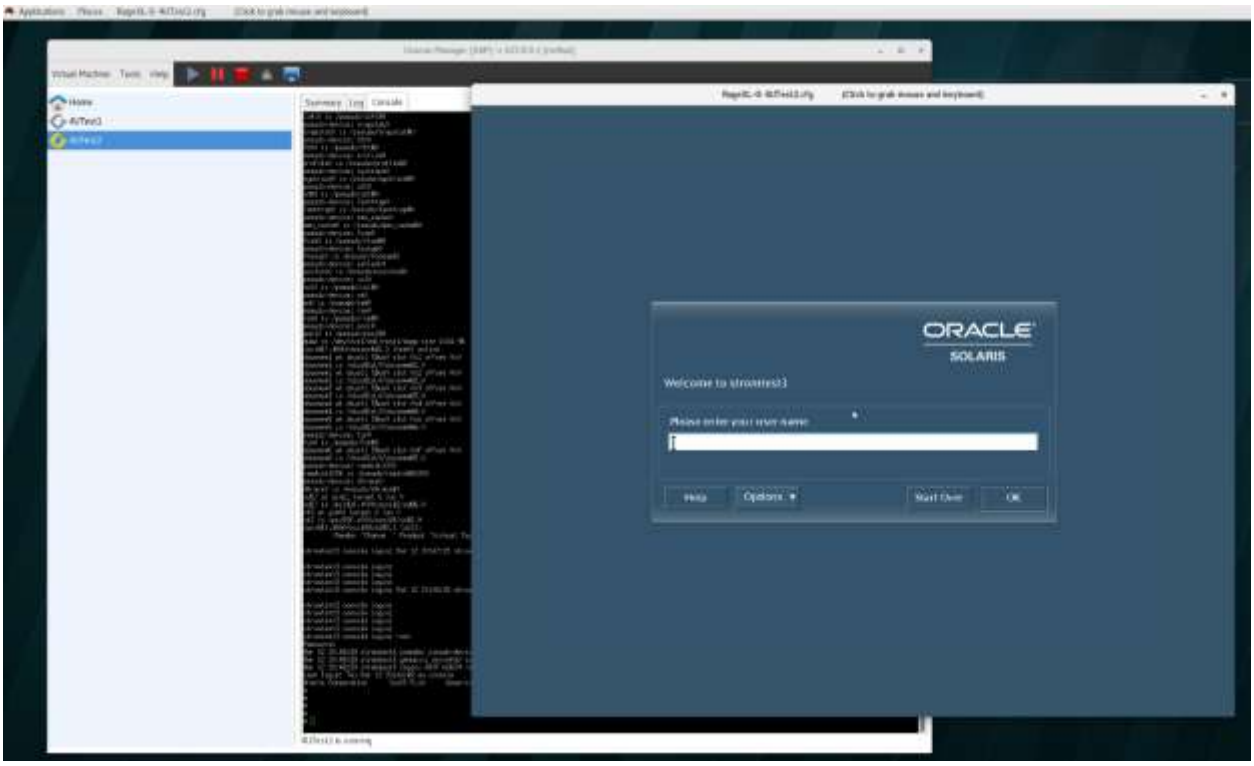
20. At the bash prompt, use the **/usr/dt/bin/dtconfig –reset** command to reset and start the dtlogin service.

```
bash-3.2# # /usr/dt/bin/dtconfig -reset
done
dtlogin config resources reloaded.
# /usr/dt/bin/dtconfig -e
done
desktop auto-start enabled.
#
```

21. At the bash prompt, restart your Solaris VM:

```
init 6
```

When the VM restarts, the graphics emulation window appears after a short wait and displays the Solaris start screen, where you can sign in.



5.2 Create a hop server for secure access to the Solaris VM

It's a good practice to set up a hop server that you can use to connect to the Solaris VM in Azure over XDMCP. Traffic is not encrypted in XDMCP, although you can buy products for securing XDMCP via SSH. However, that setup is beyond the scope of this guide. In addition, XDMCP requires multiple ports to be opened between client and server, which makes it even more challenging to connect from an on-premises environment to a cloud environment.

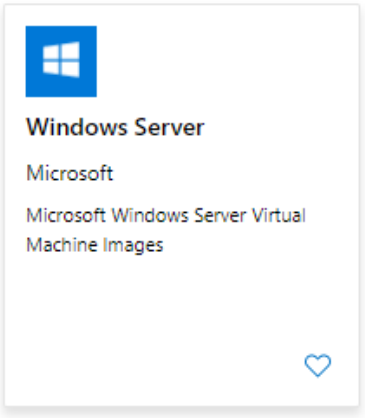
An easier way to add security is to create a Windows VM to act as the hop server running an XDMCP client, such as MobaXterm, and use it to connect via RDP from on-premises. When the Windows RDP session opens, you can make an XDMCP connection to the Solaris VM using MobaXterm.

The following steps walk you through this setup.

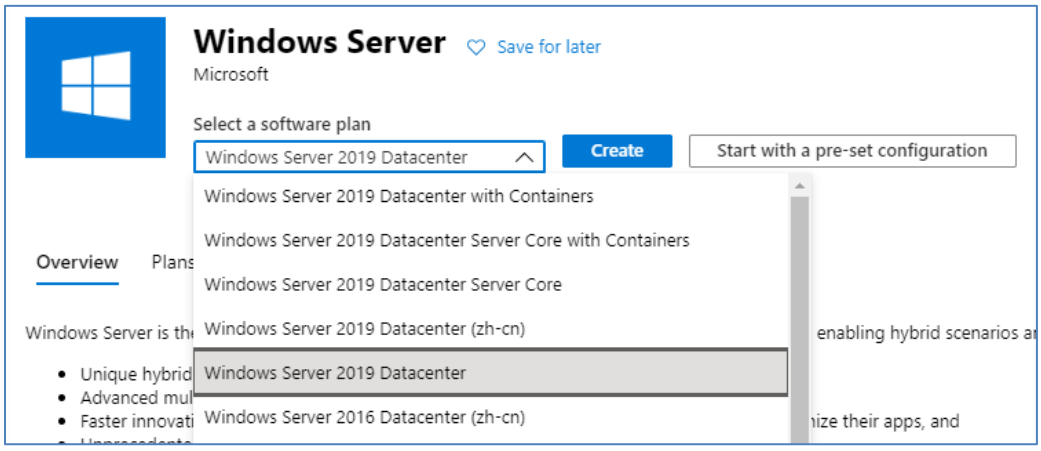
5.2.1 Provision a Windows VM

1. On your main laptop web browser, go to the Azure portal (portal.azure.com).
2. On the portal menu, click **Resource groups**.
3. Under **Resource groups**, click the resource group you have been using, then click **Add**.

4. In the search box, search for **windows server**. In the results, select the Windows Server template shown, then click **Create**.

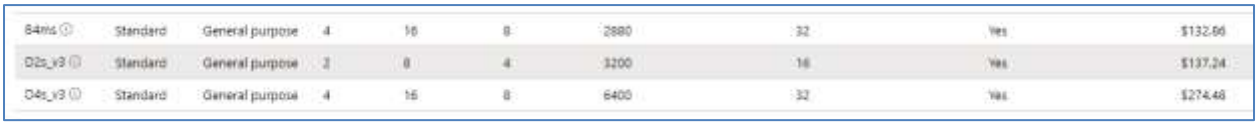


5. Under **Select a software plan**, choose **Windows Server 2019 Datacenter**.



6. Click **Create**. Under **Create a virtual machine**, go to the **Basics** tab and use the following settings:

   - **Subscription:** Use the subscription you created for your new resource group.
   - **Resource Group:** Choose the Resource Group you just created.
   - **Virtual Machine Name:** Provide a unique name as the host name for this VM. This guide uses **jfrost-win-st1**.
   - **Region:** Choose the location where you want this VM deployed in Azure.
   - **Availability Options:** *Optional*. For now, choose **No infrastructure redundancy required**.
   - **Image:** Choose **Windows Server 2019 Datacenter**.
   - **Azure Spot Instance:** No
   - **Size:** Choose an appropriately sized VM. A hop server doesn't run many services, so you can choose a smaller size. For example, a D2s_v3 VM with two CPU Cores and 8 GB RAM is a good choice.



   - **Username:** Choose an easy to remember username, such as **stromadmin**.
   - **Password:** Choose a password that meets the security requirements.
   - **Public Inbound Ports:** Choose **None**. You will set up the network security group rules later.

7. Click **Next: Disks** to go to the **Disk Setup** screen. For **OS disk type**, choose Premium SSD. Leave the default settings in the rest of the fields.

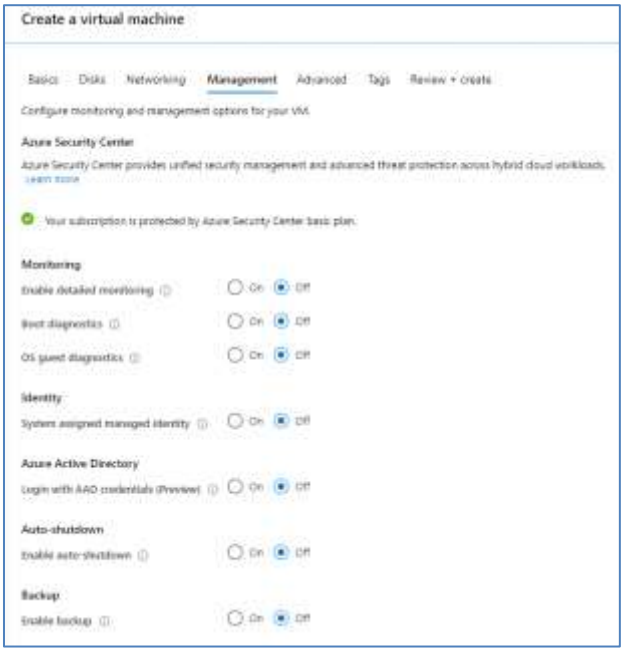
8. Click **Next: Networking** to go to the **Networking** tab. Choose the following settings:

   - **Virtual network:** Choose the virtual network used by the Linux VM in an earlier step.
   - **Subnet:** Use the same subnet used by the Linux VM in an earlier step.
   - **Public IP:** If your organization policies allow public IPs, allow the wizard to create a new one. You can use it to connect your laptop via RDP to the Windows hop server.
   - **NIC network security group:** Choose **None**. In an earlier step, you attached a network security group to this subnet, so you don't need to specify another. You can share that one, which avoids having to set up more port open rules.
   - **Public inbound ports:** Choose **None**.
   - **Accelerated Networking:** Choose **On**. If this option is not available for the VM size you chose, that's fine. This setting can stay off.
   - **Load balancing:** Choose **No**.

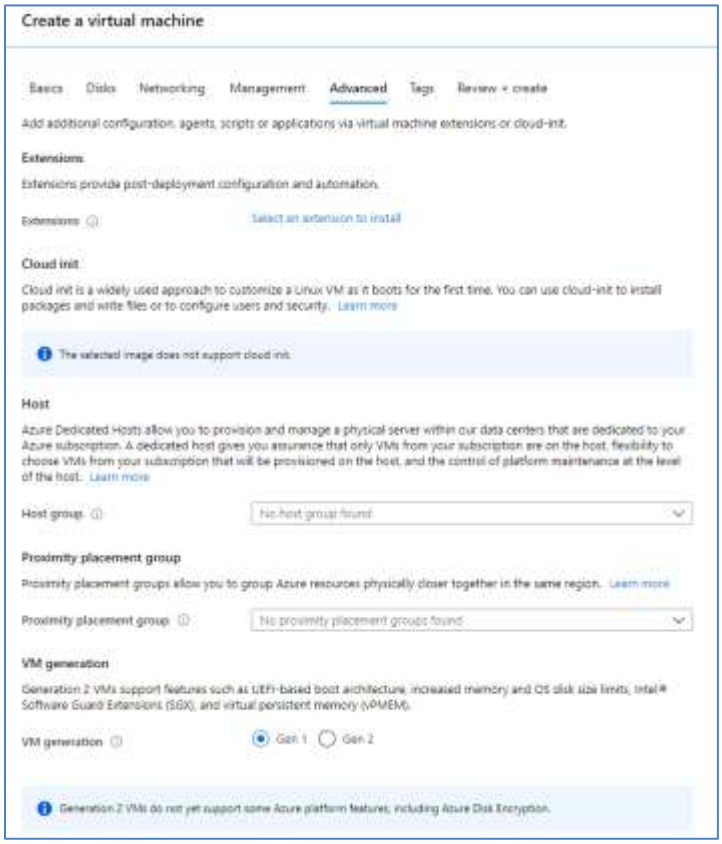
9. Click **Next: Management**. On the **Management** tab, use the following settings:

   - **Enable detailed monitoring:** Off

- **Boot diagnostics:** Off
- **OS guest diagnostics:** Off
- **System assigned managed identity:** Off
- **Login with AAD credentials:** Off
- **Enable auto-shutdown:** Off
- **Enable backup:** Off



10. Click **Next: Advanced**. On the **Advanced** tab, use all the default settings.



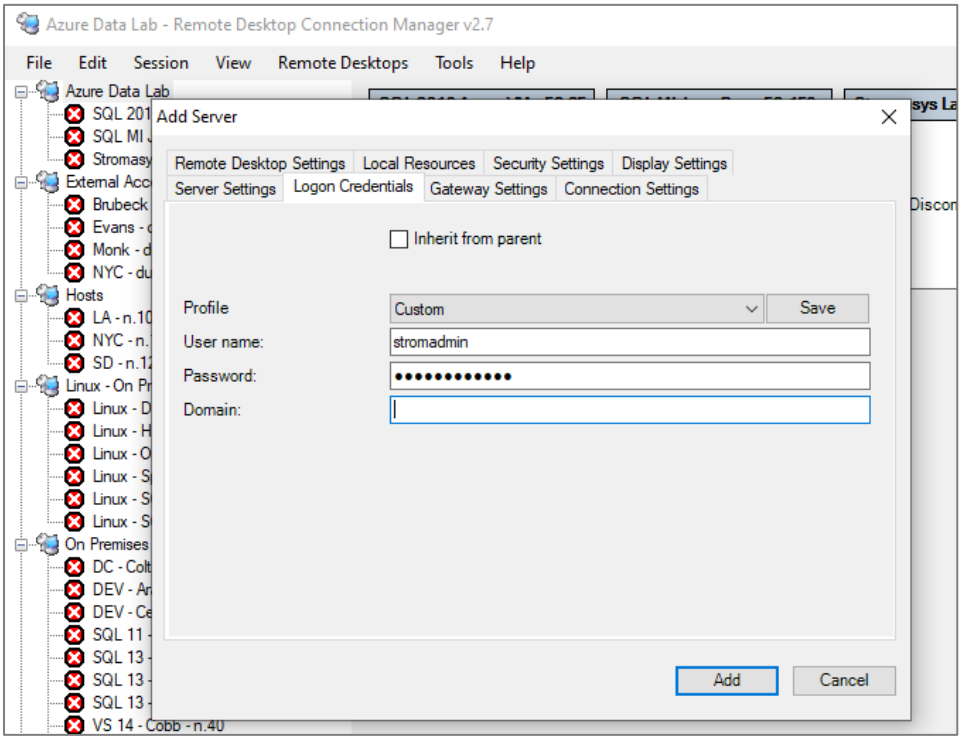
11. Click **Review + create**, verify your settings, then click **Create** to start creating the VM.

5.2.2 Set up a remote desktop connection for the hop server

From your Windows laptop, you can use Remote Desktop Connection Manager just as you did for the Linux VM to create a remote desktop connection. This guide uses Remote Desktop Connection Manager 2.7.

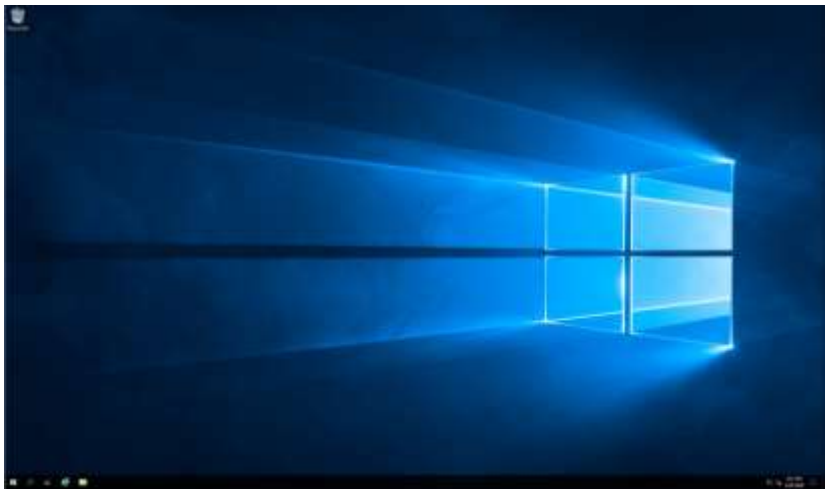
1. Open Remote Desktop Connection Manager and go to **Edit** > **Add Server**.
2. For **Server Name**, enter the public IP address of the Windows VM from an earlier step.
3. For **Display Name**, enter a helpful name to identify the VM.
4. Click the **Logon Credentials** tab and make sure **Inherit from parent i**s not selected. Use the following settings:
   - For **Profile**, chose **Custom**.
   - For **User name**, enter **stromadmin** (or the username you created earlier).
   - For **Password**, enter the Linux admin password you created earlier.

- For **Domain**, delete any value here to make it blank.



5. Click **Add** to add the new server.

6. In the menu, right-click the new server and choose **Connect Server**. If a Remote Desktop Connection message prompts you about certificate errors, click **Yes** to continue. When the remote connection is made, the Windows desktop appears something like this:



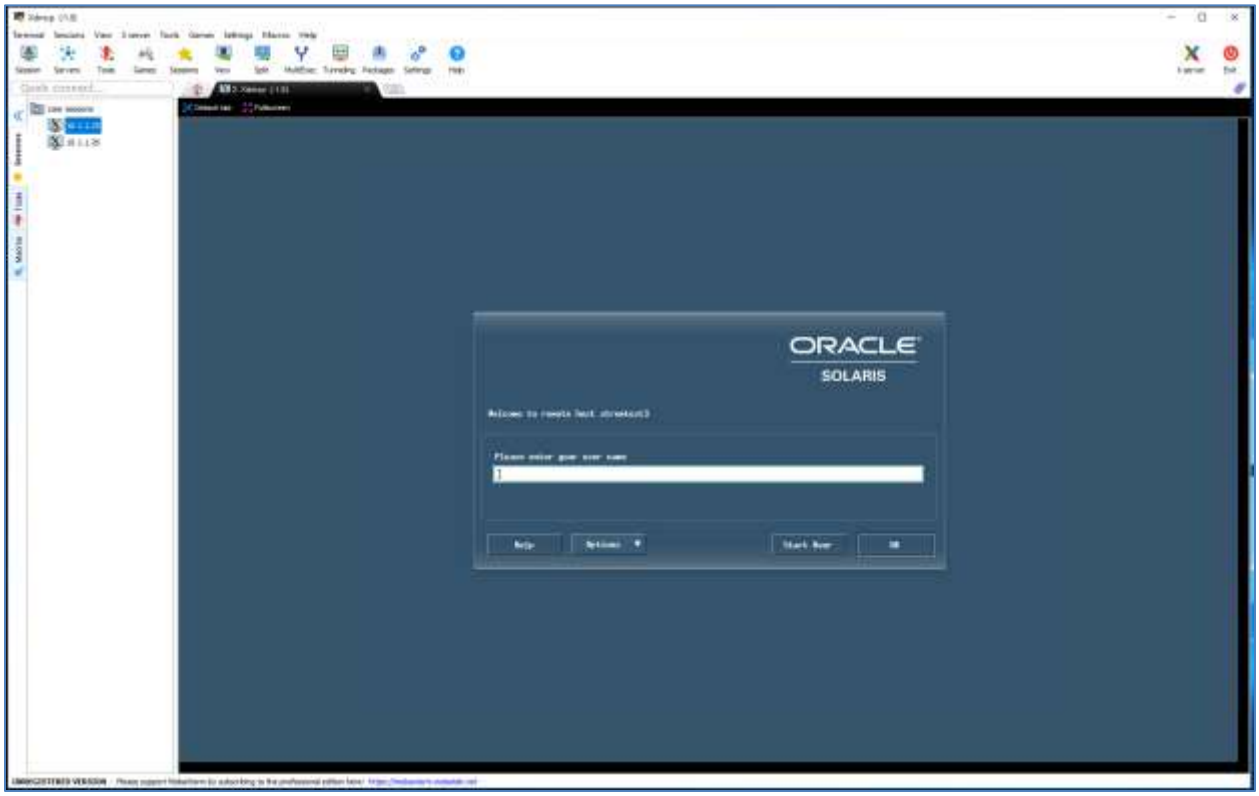
5.2.3 Set up MobaXterm and XDMCP on the hop server

It's useful to add MobaXterm to the Windows hop server as you did for your on-premises laptop. That makes it easy to set up XDMCP connections. You can use other tools to do this, but this guide shows how to download and install MobaXterm.

When it comes to browsers in Windows Server 2019, we recommend using the new chromium-based version of Microsoft Edge, and not Internet Explorer.

> **Note:** If you can't open the links for Edge or MobaXterm, you may need to turn off the Internet Explorer Enhanced Security Configuration (IE ESC). On the **Start** menu, go to **Server Manager** > **Local Server** > **IE Enhanced Security Configuration** and set both options to **Off**.

1. In the RDP session for the Windows hop server, open MobaXterm, create a new session, and select **Xdmcp**.
2. Select **Specify server to connect to** and, for the value, type the private IP address of the Solaris VM, which should be **10.1.1.25**.
3. Click **OK** to have MobaXterm make the XDMCP connection to the Solaris VM. A black screen appears and the hourglass cursor. After a moment, the Solaris sign-on screen appears:



4. For the username and password, use the **soladmin** account created earlier for testing and click **OK**.

   Now you can remotely connect to your Solaris VM using XDMCP from an easy-to-access Windows hop server. For **Password**, enter the password you created for the VM.

5. If prompted to save password, choose **Yes** if you want. You should now be at the bash shell prompt for the VM.

5.3 Set up Solaris virtual tape device emulation and Azure Files Storage

In this section, you set up the virtual tape device for the Solaris VM. This guide uses Azure Files storage as the physical location that the virtual tape device maps to.

You can also use Azure Managed Disks as the physical location for the virtual tape device files, but Azure Files storage has a few advantages. For starters, it's more economical. In addition, Azure Files is a fully managed service. It provides geographic backup options that make it easy to set up disaster recovery options—in keeping with the spirit of tape devices.

For more information, see Introduction to the core Azure Storage services in the Azure documentation.

To get started, you must create a new Azure storage account, then set the host Linux VM to read the storage account as a mounted drive. Then you can use the Stromasys Charon-SSP Virtual Tape features to create a virtual tape device that maps its physical location to Azure Files. The last step is to test tape device and make sure it's useable.

5.3.1 Create an Azure storage account

1. On your main laptop web browser, go to the Azure portal (portal.azure.com).
2. On the portal menu, click **Resource groups**. Select the resource group you've been using and click **Add**.

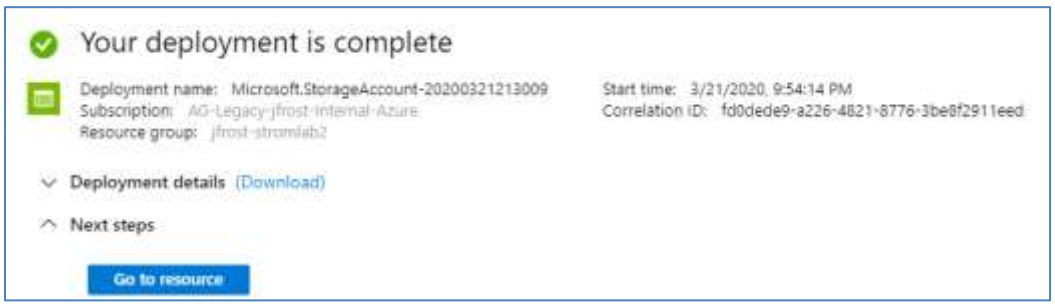
3. In the search box, type **azure storage account**, and in the search results, choose the **Storage account – blob, file, table, queue** template shown and click **Create**.

4. Under **Create storage account**, on the **Basics** tab, use the following settings:

   - **Subscription:** The subscription in which you created your new resource group.
   - **Resource Group:** Choose the Resource Group you just created.
   - **Storage Account Name:** Choose a unique name for this storage account. (This guide uses **jfroststromablob**.)
   - **Location:** The geographic location where you deployed your other artifacts.
   - **Performance:** Standard
   - **Account kind:** StorageV2
   - **Replication:** LRS—or choose another option for replication. For details, see Azure Storage redundancy.
   - **Access tier:** Hot

5. Leave the rest of the settings as they are, then click **Review + create**.

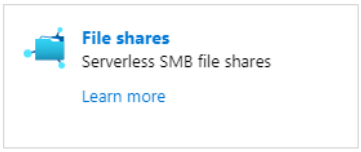
6. Review your settings, then click **Create** and wait for the storage account to be provisioned.



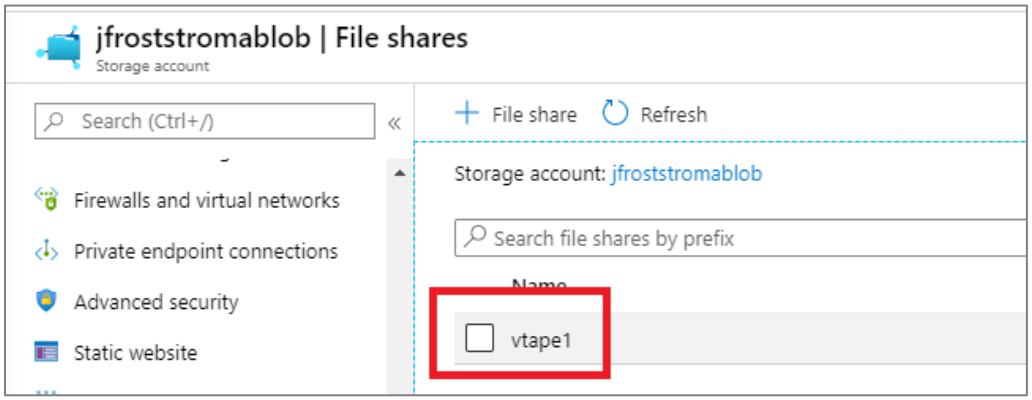
5.3.2 Create a file share

The next step is to create an Azure file share within your storage account. This share is used as the object that you can mount as a drive on the Linux host VM.

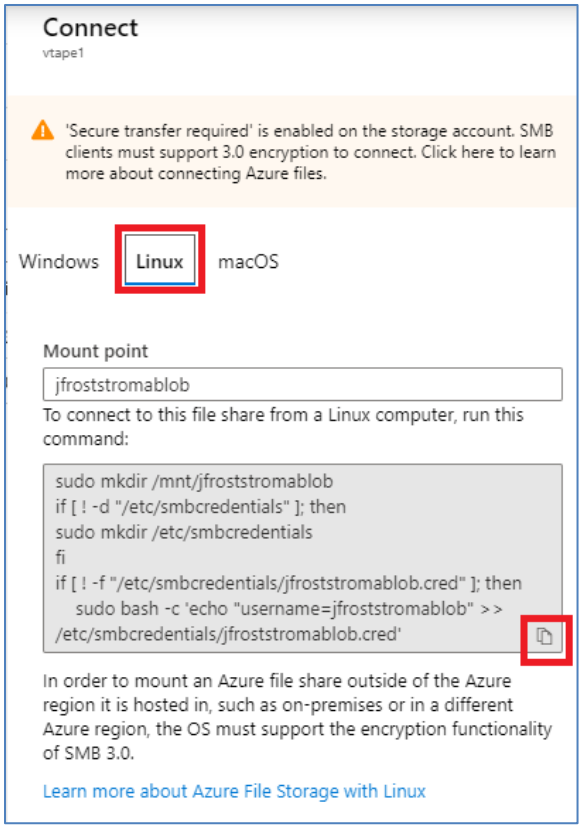
1. In the Azure portal, click **Go to resource.**
2. On the **Overview** of the new storage account, click the **File shares** link.



3. On the toolbar, click **+ File share**, then use the following settings:
   - **Name:** vtape1
   - **Quota:** 5,000 GB
4. Click **Create.**
5. Under **File shares**, click the **vtape1** file share item.



6. Click **Connect**.
7. Under **Connect**, click **Linux**. Then, to copy the bash shell script command, click the Copy ( ) icon as shown:



8. Open Notepad and paste these commands, then save the file somewhere handy on your laptop. You need these commands later when you mount this file share to the Linux VM.

5.3.3 Mount the file share to the Linux host VM as a drive

Using the script you just copied, you can mount the Azure file share to the host Linux VM. To do this, you need to install cifs-utils, a Linux utility package that provides a means for mounting SMB/CIFS shares on a Linux system.

You can then run each of the script lines to execute the mount. It is important to note that you can run the copied commands as a single shell script, but the steps below run one line a time. It's easier to debug any issues if you do this.

1. On your laptop, use MobaXterm to connect over SSH to access the Linux host VM and get a bash shell command line.
2. At the prompt (`[stromadmin@host ~]$`), run the following command to install the cifs-utils package. (If prompted at any point, type **y** and press **Enter**.)

```
sudo yum install cifs-utils
```

3. When the cifs-utils package is installed, run the commands that you copied in the previous section to mount the Azure file share. Run the **mkdir**, **bash**, **chmod**, and **mount** commands one line at a time to make debugging easier like this:

```
sudo mkdir /mnt/jfroststromablob

if [ ! -d "/etc/smbcredentials" ]; then

sudo mkdir /etc/smbcredentials

fi
if [ ! -f "/etc/smbcredentials/jfroststromablob.cred" ]; then

sudo bash -c 'echo "username=jfroststromablob" >>
/etc/smbcredentials/jfroststromablob.cred'

sudo bash -c 'echo
"password=NIxY96S6KabDpL78Cy1rqleFlEs4XI7vqdLidkX+pIyqLJJWjwHy+O1bNC1Yq3Q6MEtdtO7
vV08HzIAXtQ2TAw==" >> /etc/smbcredentials/jfroststromablob.cred'

fi

sudo chmod 600 /etc/smbcredentials/jfroststromablob.cred

sudo bash -c 'echo "//jfroststromablob.file.core.windows.net/vtape1
/mnt/jfroststromablob cifs
nofail,vers=3.0,credentials=/etc/smbcredentials/jfroststromablob.cred,dir_mode=07
77,file_mode=0777,serverino" >> /etc/fstab'

sudo mount -t cifs //jfroststromablob.file.core.windows.net/vtape1
/mnt/jfroststromablob -o
vers=3.0,credentials=/etc/smbcredentials/jfroststromablob.cred,dir_mode=0777,file
_mode=0777,serverino
```

4. If the following error message appears while running the mounting commands, reinstall cifs-utils using the **sudo yum install cifs-utils** command. Errors like this typically happen when the package isn't installed correctly.

```
mount: wrong fs type, bad option, bad superblock on
//jfroststromablob.file.core.windows.net/vtape1,
       missing codepage or helper program, or other error
       (for several filesystems (e.g. nfs, cifs) you might
       need a /sbin/mount.<type> helper program)

       In some cases useful info is found in syslog - try
       dmesg | tail or so.
```

5. To verify that the file share was mounted correctly, run the **df -h** command. If successful, the mounted drive is listed. Its size is 5.0 TB (the quota placed on it when created) as shown here:

```
[stromadmin@host ~]$ df -h
Filesystem                                      Size  Used Avail Use% Mounted on
/dev/sda2                                        32G   11G   21G  35% /
devtmpfs                                        126G     0  126G   0% /dev
tmpfs                                           126G   42M  126G   1% /dev/shm
tmpfs                                           126G  4.0G  123G   4% /run
tmpfs                                           126G     0  126G   0%
/sys/fs/cgroup
/dev/sdc1                                       1.0T   72G  953G   7%
/datadrive1
/dev/sda1                                       497M  108M  390M  22% /boot
//jfroststromablob.file.core.windows.net/vtape1  5.0T   35G  5.0T   1%
/mnt/jfroststromablob
/dev/sdb1                                       504G  2.1G  477G   1%
/mnt/resource
tmpfs                                            26G  112K   26G   1%
/run/user/1000
tmpfs                                            26G     0   26G   0%
/run/user/986
```

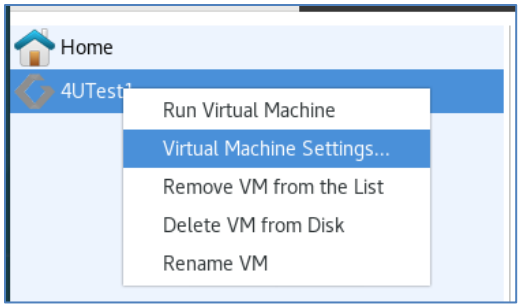
Now you have an Azure file share you can access in the Linux VM. In the next section, you configure Charon-SSP to add a virtual tape device to your Solaris VM, Then from the Solaris VM, you can run tape commands to execute a backup. The backup physically resides on the Azure file share and takes advantage of the redundancy and economy that Azure Storage offers.

5.4 Create a Solaris virtual tape device and run a backup

This section shows you how Azure Files, when attached to the host Linux VM, can act as the data storage or Solaris virtual tape backups. First, you create a virtual tape device in the Charon-SSP manager and attach that device to your Solaris VM. Then you can execute a few tape device commands to perform a backup on the Solaris VM's virtual hard disk primary partition.

5.4.1 Create the virtual tape device

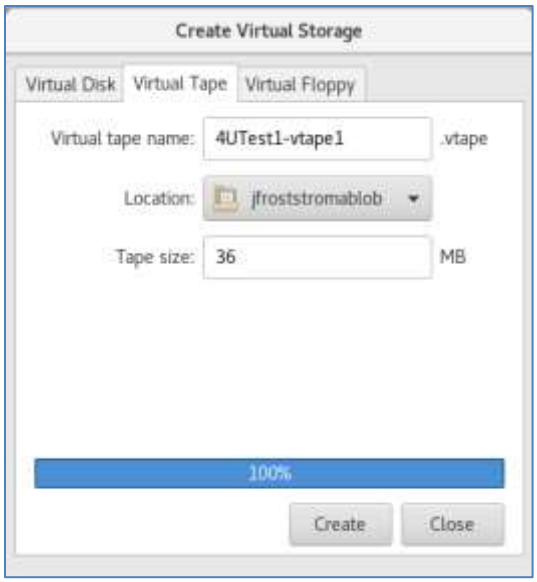
1. Using Remote Desktop Manager, connect to your Linux Host VM.
2. Go to **Charon-SSP Manager** and make sure the Solaris VM is stopped. This is necessary so you can edit the VM settings and add the virtual tape device.
3. Right-click the name of the Solaris VM and choose **Virtual Machine Settings**.



4. Under **Device**, click **SCSI**.
5. In the **SCSI** options, click **Create Virtual Storage**, click the **Virtual Tape** tab, and choose the following settings:
   - **Virtual disk name:** 4UTest-vtape.vtape
   - **Location:** /mnt/jfroststromablob

   > **IMPORTANT:** Make sure to choose the mounted location of the Azure File storage account. This enables you to use Azure Files as the physical storage location.

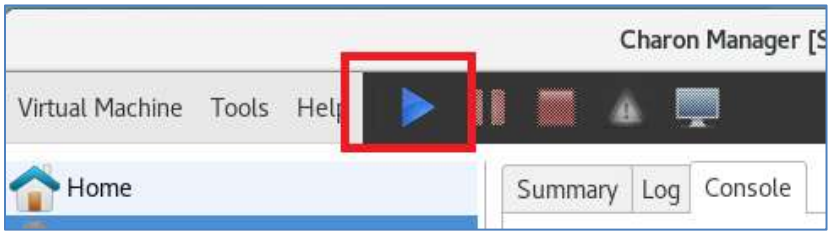
   - **Tape size:** 36 MB (Note that the tape size can grow as needed.)
6. Click **Create** to provision the disk, and when it's finished, click **Close**.



7. In the **Virtual Machine Settings** window, click **Add** to open the **Add SCSI Device** window, where you can add the virtual tape device. Choose the following settings:
   - **SCSI bus:** Primary SCSI Bus
   - **SCSI ID:** 2
   - **LUN ID:** 0
   - **Removable:** ON

- **SCSI device type:** Virtual Tape
- **SCSI device path:** /mnt/jfroststromablob/4UTest-vtape.vtape

8. Click **OK** to return to the **Virtual Machine Settings** window, and click **OK**.
9. In the **Charon-SSP Manager** main window, click the **Start** button for the Solaris VM:



10. When the Solaris VM has started up, type **boot disk1 –r** at the **ok** prompt and press **Enter**. The **–r** argument ensures that Solaris configures the device.
11. When prompted to sign in, use **root** for the username and the password for root that you specified during the Solaris installation process.

5.4.2 Run Solaris backup to the virtual tape device

For this step, you can continue to use the console command line in Charon-SSP Manager. However, for a better experience, you can use MobaXterm and connect to the Solaris VM using the public IP address you assigned earlier.

The following steps use MobaXterm for the SSH connection and the **soladmin** user created earlier.

1. Sign in to the Solaris VM as **soladmin** and run the bash **su** command to become the root user:

```
# bash
Bash-3.2# su
Password: ******
Bash-3.2#
```

2. To verify that the virtual tape device is mounted correctly, at the bash prompt, run:

```
mt -f /dev/rmt/0
```

   You may need to run the **mt** command a few times to fully initialize the tape device. The test is complete when you see results that look like this:

```
Unconfigured Drive: Vendor 'Charon  ' Product 'Virtual Tape   ' tape drive:
   sense key(0x0)= No Additional Sense   residual= 0   retries= 0
   file no= 0   block no= 0
```

3. Get the correct name of the partition as follows. This is needed to back up the entire disk partition.

   a. Run the **format** command.

```
Bash-3.2# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
0. c0t1d0 <COMPAQ-RZ1FB-0200 cyl 17781 alt 2 hd 40 sec 100>
/pci@1f,4000/scsi@3/sd@1,0
Specify disk (enter its number): 0
selecting c0t1d0
[disk formatted]
/dev/dsk/c0t1d0s0 is part of active ZFS pool rpool. Please see zpool(1M).
/dev/dsk/c0t1d0s2 is part of active ZFS pool rpool. Please see zpool(1M).
```

   b. In the results, locate the partition and slice you need to backup. In this case, **0. c0t1d0** is a partition (shown immediately below AVAILABLE DISK SELECTIONS). The last two lines are the two main partition slices active in the ZFS pool, **c0t1d0s0** and **c0t1d0s2**.

   c. At the **format** > prompt, run the **partition** command.

   d. At the **partition** > prompt, run the **print** command.

   e. In the results, locate the slice. Its tag is either **root** or **backup**. In this case, you want the root partition slice (Part 0):

```
Current partition table (original):
Total disk cylinders available: 17781 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 17780       33.91GB    (17781/0/0) 71124000
  1 unassigned    wm       0                0         (0/0/0)            0
  2     backup    wm       0 - 17780       33.91GB    (17781/0/0) 71124000
  3 unassigned    wm       0                0         (0/0/0)            0
  4 unassigned    wm       0                0         (0/0/0)            0
  5 unassigned    wm       0                0         (0/0/0)            0
  6 unassigned    wm       0                0         (0/0/0)            0
  7 unassigned    wm       0                0         (0/0/0)            0
```

f. Calculate the full slice name needed to run the backup. In this case, it's **c0t1d0s0**.

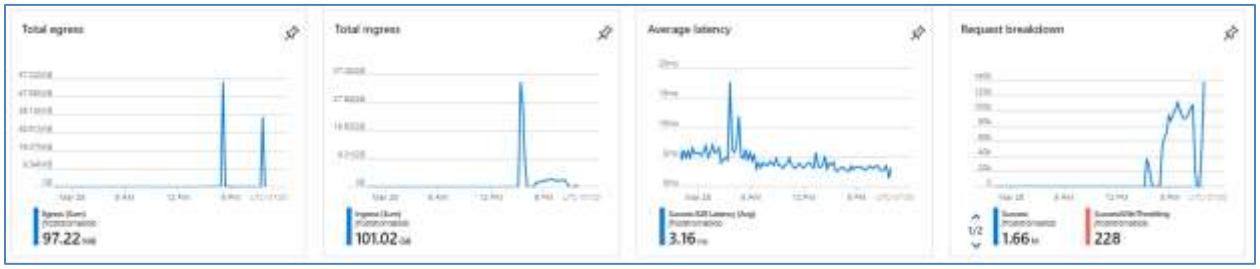
g. To return to the bash prompt, at the **partition>** prompt, type **quit** and press **Enter**. Then repeat at the **format>** prompt.

4. Run the **dd** command on the **c0t1d0s0** partition slice as shown. When complete, the number of records in and records out is displayed.

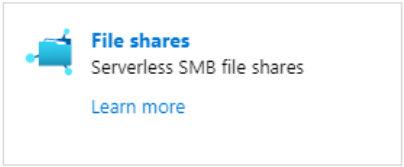
```
Bash-3.2# dd if=/dev/dsk/c0t1d0s0 of=/dev/rmt/0 bs=960k
555+1 records in
555+1 records out
```

5. To check the progress of the backup, go to the Azure portal and click the **jfroststromablob** resource for this storage account.

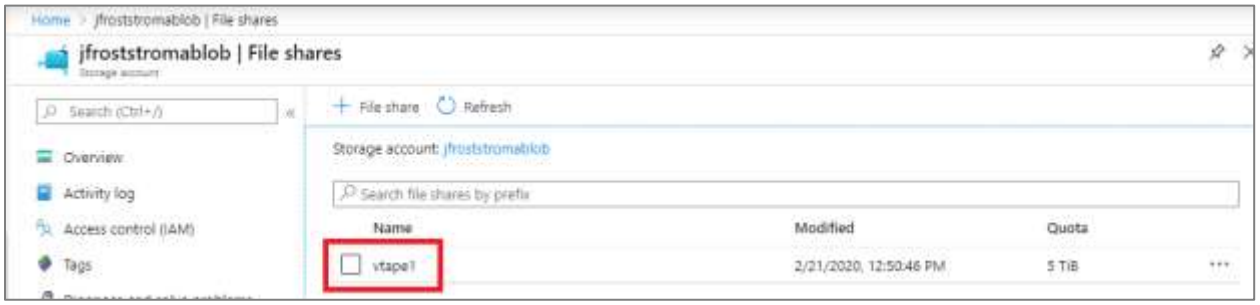
6. On the **Overview** of the **jfroststromablob** resource, note the metrics, such as **Total ingress** and **Request breakdown**. They show the activity on the storage account.



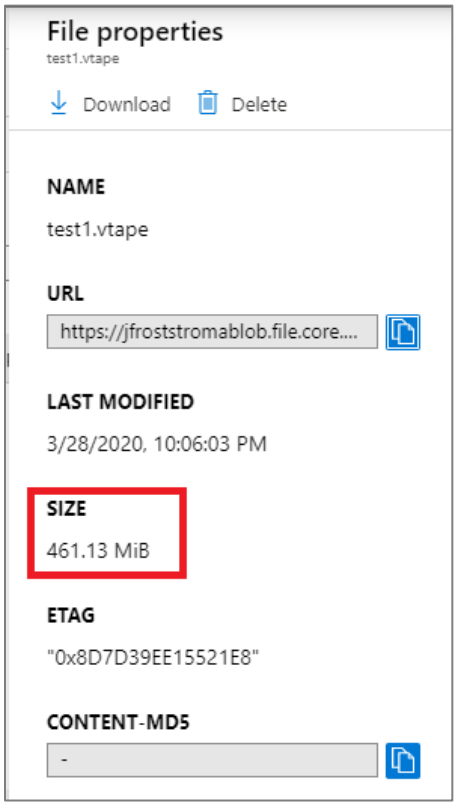
7. To see the progress of the growth of the .vtape file, on the **Overview**, click **File shares**:



8. Click the **vtape1** item listed for the **jfroststromablob** resource:



9. In the **vtape1** settings, click the **test1.vtape** file and notice the **File properties** box that opens. Refer to the **File size** field. The value updates as the backup continues, showing you the progress.



> **Note:** A backup can take several hours to finish depending on the size of the partition. For example, it can take up to five hours for a 33-GB partition that is backed up on a 64-core VM with a 1-TB SSD managed disk to house the partition being backed up.

That's it!

You have successfully created a virtual tape device, attached it to the Solaris VMa and run a backup operation onto that tape using a storage account as the physical storage service.

6 References

In creating this guide, we found the following resources helpful.

Stromasys and Solaris

- Stromasys Charon-SSP documentation
- "How to Configure the Firewall on Oracle Solaris"
  in Chapter 5 of *Securing the Network in Oracle Solaris 11.4*
- Chapter 3, "Setting Up and Using a Tape Drive"
  in *Solaris Handbook for Sun Peripherals*
- "Solaris Backup and Restoration Utilities"
  in Chapter 7 of *Solaris 10 System Administration Exam Prep*
- "Obtaining Disk Information"
  in Chapter 6 of *Managing Devices in Oracle Solaris 11.3*
- Some basic commands and tips for Solaris 10 / 11 servers

Azure

- Use the portal to attach a data disk to a Linux VM
- Optimize your Linux VM on Azure

Linux and open source

- How to Choose Your Red Hat Enterprise Linux File System
- EXT4 vs XFS for Oracle, which one performs better?
- What's Barriers, how to enable/disable it on Linux