Red Hat Enterprise Linux 8.0 Beta

Deploying Red Hat Enterprise Linux on public cloud platforms

A public cloud deployment guide

Last Updated: 2019-02-08

Red Hat Enterprise Linux 8.0 Beta Deploying Red Hat Enterprise Linux on public cloud platforms

A public cloud deployment guide

Legal Notice

Abstract

This document provides concepts and procedures for deploying a Red Hat Enterprise Linux 8.0 Beta virtual machine (VM) on several major public cloud platforms. If you have created VMs for public clouds using other Linux distributions, or if you are only looking for required packages and drivers, review the individual topics provided to find what you need.

Table of Contents

THIS IS A BETA VERSION!

Thank you for your interest in Red Hat Enterprise Linux 8.0 Beta. Be aware that:

- Beta code should not be used with production data or on production systems.
- Beta does not include a guarantee of support.
- Feedback and bug reports are welcome. Discussions with your account representative, partner contact, and Technical Account Manager (TAM) are also welcome.
- Upgrades to or from a Beta are not supported or recommended.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so:

- For simple comments on specific passages, make sure you are viewing the documentation in the Multi-page HTML format. Highlight the part of text that you want to comment on. Then, click the **Add Feedback** pop-up that appears below the highlighted text, and follow the displayed instructions.
- For submitting more complex feedback, create a Bugzilla ticket:
  1. Go to the Bugzilla website.
  2. As the Component, use **Documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.

CHAPTER 1. DEPLOYING A VIRTUAL MACHINE ON MICROSOFT AZURE

The following topics provide step-by-step guidance for deploying a Red Hat Enterprise Linux 8.0 Beta virtual machine (VM) on Microsoft Azure.

Prerequisites

- You need a Microsoft Azure account and a Red Hat Customer Portal account to successfully complete the steps.
- Enroll in Red Hat Cloud Access. Red Hat Cloud Access allows you to move your Red Hat subscriptions from physical or on-premise systems onto Microsoft Azure with full support from Red Hat.

Additional resources

- Red Hat Cloud Access
- Red Hat Azure Portal

1.1. OTHER METHODS FOR GETTING AN IMAGE

In addition to the manual steps provided in the following sections, there are other ways to get a Microsoft Azure VM image. These methods are listed below. Note that while these methods may be quicker, it may be useful to complete the manual steps and learn about the required packages and configuration changes necessary for Red Hat Enterprise Linux VMs operating in Microsoft Azure.

- **Composer**
  Composer provides a pre-built image for creating a VM for Microsoft Azure. See Preparing Composer for Creating Azure VHD Images.
- **Microsoft Azure Marketplace**
  Many Red Hat Enterprise Linux images are available from the Microsoft Azure Marketplace. Note that there are VM usage cost differences between using your Red Hat subscriptions through the Red Hat Cloud Access program (called Bring Your Own License, or BYOL, in Microsoft Azure) and using a Pay-As-You-Go marketplace image from Microsoft. See Billing options in the Azure Marketplace for more information.

1.2. USING A CUSTOM BASE IMAGE

To configure a public cloud VM (or instance), you start with a base (starter) VM image. Once the base VM image is created, you modify configuration settings and add the packages required for the VM (or instance) to operate on the public cloud platform. Further configuration changes may be made after the image is uploaded and operating. This is typically required when the VM is used for a specific application.

The recommended base VM image to use for all public cloud platforms is the **Red Hat Enterprise Linux 8.0 Beta KVM Guest Image** downloaded from the Red Hat Customer Portal. The KVM Guest Image is pre-configured with the following cloud configuration settings.

- *The root account is disabled.* For several public cloud platforms, you temporarily enable root account access to make configuration changes and install packages required for the public cloud. The instructions for temporarily enabling root account access are provided in this guide.

- *A user account named* ***`cloud-user`*** *is pre-configured on the image.* The **`cloud-user`** account has sudo access.
- *The image has* ***`cloud-init`*** *installed and enabled.* **`cloud-init`** is a service that handles provisioning of the VM (or instance) at initial boot.

You can choose to use a custom Red Hat Enterprise Linux ISO image. However, note that when using a custom ISO image, additional configuration tasks may be necessary for the resulting VM to operate on the public cloud platform.

Additional resources

Red Hat Enterprise Linux

1.2.1. Required system packages

The procedure assumes you are creating a VM image for Azure using Red Hat Enterprise Linux or Fedora. To successfully complete the procedure, you need to have the packages listed in the following table installed. Note that these packages are located in the base **`fedora`** repository (if using Fedora).

Table 1.1. System Packages

Package	Repository	Description
libvirt	rhel-8-for-x86_64-appstream-beta-rpms	Open source API, daemon, and management tool for managing platform virtualization
virt-install	rhel-8-for-x86_64-appstream-beta-rpms	A command-line utility for building VMs
libguestfs	rhel-8-for-x86_64-appstream-beta-rpms	A library for accessing and modifying virtual machine file systems
libguestfs-tools	rhel-8-for-x86_64-appstream-beta-rpms	System administration tools for virtual machines; includes the guestfish utility

1.2.2. Azure virtual machine configuration settings

Microsoft Azure VMs must have the following configuration settings. Some of these settings are enabled during the initial VM creation. Other settings are set when provisioning the VM image for Microsoft Azure. Keep these settings in mind as you move through the procedure and refer back to them if you need to.

Table 1.2. VM Configuration Settings

Setting	Recommendation
ssh	ssh must be enabled to provide remote access to your Azure VMs.

Setting	Recommendation
dhcp	The primary virtual adapter should be configured for dhcp (IPv4 only).
Swap Space	Do not create a dedicated swap file or swap partition. Swap space may be configured in the Windows Azure Linux Agent.
NIC	Choose **virtio** for the primary virtual network adapter.
encryption	Users may use Azure Disk Encryption (ADE) for on-demand (Pay-As-You-Go) operating system disks on Red Hat Enterprise Linux (RHEL) 7.2 or later versions in Microsoft Azure. Data disks may be encrypted for any supported version of RHEL on Azure. ADE uses a new encryption method in RHEL 7.6, but the extension for enabling ADE on RHEL 7.6 is not available at this time. If you need to use ADE on RHEL 7.6, choose an earlier version of RHEL, encrypt the virtual machine (VM), then upgrade to RHEL 7.6. This VM will continue to use the pre-RHEL 7.6 encryption method. OS disk encryption is not supported for RHEL VMs in Microsoft Azure when using Red Hat Cloud Access VMs, RHEL 6 or any version of RHEL where the Red Hat shipped packages were modified.

1.2.3. Creating a base image from a KVM Guest image

Red Hat and the open source community continually optimize the KVM Guest image for virtualized environments. Once you have the image configured, you can use this image as a template for creating additional VMs in Microsoft Azure.

Procedure

1. Download the latest Red Hat Enterprise Linux 8.0 Beta KVM Guest image from the Red Hat Customer Portal.
2. Create and start a basic Red Hat Enterprise Linux VM. See 2.2. Creating virtual machines for instructions. When creating the VM, use the following configuration settings.
   - Change the default memory and CPUs to the capacity settings you want for the VM.
   - Select **virtio** for the virtual network interface.
3. Shut down the new VM after a login prompt appears.
4. Set up root access to the VM. From your system, use **`virt-customize`** to generate a root password for the VM.

```
# virt-customize -a <guest-image-path> --root-password password:
<PASSWORD>
```

Example:

```
# virt-customize -a /var/lib/libvirt/images/rhel-guest-image-8.0-
120.x86_64.qcow2 --root-password password:redhat!
[   0.0] Examining the guest ...
[ 103.0] Setting a random seed
[ 103.0] Setting passwords
[ 112.0] Finishing off
```

5. Verfiy root access by starting the RHEL VM and logging in as **root**.

6. Once you are logged in as root, go to Configuring the base image for Microsoft Azure.

1.2.4. Creating a base image from an ISO image

The following procedure lists the steps and initial configuration requirements for creating a custom ISO image for Microsoft Azure.

Procedure

1. Download the latest Red Hat Enterprise Linux 8.0 Beta Binary DVD ISO image from the Red Hat Customer Portal.

2. When creating the base VM from the ISO image, use the following initial configuration settings:
   - Choose the memory and CPUs you want to use for the VM.
   - Select **virtio** for the virtual network interface.

3. Make the following additional installation selection and modifications.
   - Select **Minimal Install**.
   - For **Installation Destination**, select **Custom Storage Configuration**. Use the following configuration information to make your selections.
     - Remove **swap space**. Swap space is configured on the physical blade server in Azure by the WALinuxAgent. This step is completed later in the configuration section.
     - Verify at least 500 MB for /**boot**. The remaining space may be used for root /.
     - Logical Volume Management (LVM) may be used but Standard Partitions are preferred. See Logical Volume Management (LVM) Support in Microsoft Azure.
     - File System: xfs, ext4, or ext3 may be used.
   - Set a generic host name and verify that **ens3** is enabled.
   - Create a root password.

4. When installation is complete, reboot the VM and log in to the root account.

5. Once you are logged in as root, go to Configuring the base image for Microsoft Azure.

1.3. CONFIGURING THE BASE IMAGE FOR MICROSOFT AZURE

The base image requires configuration changes to serve as your gold Red Hat Enterprise Linux 8.0 Beta VM image in Microsoft Azure. The following sections provide the additional configuration changes required.

1.3.1. Installing Hyper-V device drivers

Microsoft provides network and storage device drivers as part of their Linux Integration Services for Hyper-V package. Hyper-V device drivers may need to be installed on the VM image prior to provisioning it as a Microsoft Azure VM. Use the **lsinitrd | grep hv** command to verify that the drivers are installed.

Procedure

1. Enter the following **grep** command to determine if all of the required Hyper-V device drivers are installed.

```
# lsinitrd | grep hv
```

In the example below, all the drivers are installed.

```
# lsinitrd | grep hv
drwxr-xr-x   2 root     root            0 Aug 12 14:21
usr/lib/modules/3.10.0-932.el7.x86_64/kernel/drivers/hv
-rw-r--r--   1 root     root        31272 Aug 11 08:45
usr/lib/modules/3.10.0-
932.el7.x86_64/kernel/drivers/hv/hv_vmbus.ko.xz
-rw-r--r--   1 root     root        25132 Aug 11 08:46
usr/lib/modules/3.10.0-
932.el7.x86_64/kernel/drivers/net/hyperv/hv_netvsc.ko.xz
-rw-r--r--   1 root     root         9796 Aug 11 08:45
usr/lib/modules/3.10.0-
932.el7.x86_64/kernel/drivers/scsi/hv_storvsc.ko.xz
```

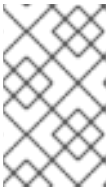
NOTE

One or more drivers may already exist in the environment. All drivers must be present. Complete the remaining steps if all drivers are not listed as shown in the example.

2. Create a file named **dracut.conf** in **/etc/dracut.conf.d**.

3. Add the following driver parameters to the **dracut.conf** file.

```
add_drivers+=" hv_vmbus "
add_drivers+=" hv_netvsc "
add_drivers+=" hv_storvsc "
```



NOTE

Note the spaces before and after the quotes, for example, **add_drivers+=" hv_vmbus "**. This ensures that unique drivers are loaded in the event that other Hyper-V drivers already exist in the environment.

4. Regenerate the **intramfs** image.

```
# dracut -f -v --regenerate-all
```

5. Verify that the drivers are installed by running the **lsinitrd | grep hv** command.

1.3.2. Making additional configuration changes

The VM requires further configuration changes to operate in Microsoft Azure. Complete the following steps to make these changes.

Procedure

1. If necessary, power on the VM.
2. Stop the **cloud-init** service (if present).

```
# systemctl stop cloud-init
```

3. Remove the **cloud-init** software.

```
# yum remove cloud-init
```

4. Edit the **/etc/ssh/sshd_config** file and enable password authentication.

```
PasswordAuthentication yes
```

5. Set a generic host name.

```
# hostnamectl set-hostname localhost.localdomain
```

6. Edit (or create) the **/etc/sysconfig/network-scripts/ifcfg-eth0** file. Use only the parameters listed below.

NOTE

The **ifcfg-eth0** file does not exist on the RHEL 8 DVD ISO image and needs to be created.

```
DEVICE="eth0"
ONBOOT="yes"
BOOTPROTO="dhcp"
TYPE="Ethernet"
USERCTL="yes"
PEERDNS="yes"
IPV6INIT="no"
```

7. Remove all persistent network device rules (if present).

```
# rm -f /etc/udev/rules.d/70-persistent-net.rules
# rm -f /etc/udev/rules.d/75-persistent-net-generator.rules
# rm -f /etc/udev/rules.d/80-net-name-slot-rules
```

8. Set **ssh** to start automatically.

```
# systemctl enable sshd
# systemctl is-enabled sshd
```

9. Modify the kernel boot parameters.

   a. Add **crashkernel=256** to the start of the **GRUB_CMDLINE_LINUX** line in **/etc/default/grub** file. If **crashkernel=auto** is present, change it to **crashkernel=256M**.

   b. Add the following lines to the end of the **GRUB_CMDLINE_LINUX** line (if not present).

```
earlyprintk=ttyS0
console=ttyS0
rootdelay=300
```

   c. Remove the following options, if they are present.

```
rhgb
quiet
```

10. Regenerate the **grub.cfg** file.

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

11. Register the VM and enable the Red Hat Enterprise Linux 8.0 Beta repository.

```
# subscription-manager register --auto-attach
```

12. Install and enable the Windows Azure Linux Agent (WALinuxAgent). The WALinuxAgent is included in Red Hat Enterprise Linux 8.0 Beta Application Stream (AppStream). See Using Application Stream for more information.

```
# yum install WALinuxAgent -y
# systemctl enable waagent
```

13. Edit the following lines in the **/etc/waagent.conf** file to configure swap space for provisioned VMs. Set swap space for whatever is appropriate for your provisioned VMs.

```
Provisioning.DeleteRootPassword=n
ResourceDisk.Filesystem=ext4
ResourceDisk.EnableSwap=y
ResourceDisk.SwapSizeMB=2048
```

14. Unregister the VM from Red Hat Subscription Manager.

```
# subscription-manager unregister
```

15. Prepare the VM for Microsoft Azure provisioning by cleaning up the existing provisioning details. Azure reprovisions the VM in Azure. This command generates warnings, which is expected.

```
# waagent -force -deprovision
```

16. Clean the shell history and shut down the VM.

```
# export HISTSIZE=0
# poweroff
```

1.4. CONVERTING THE IMAGE TO A FIXED VHD FORMAT

All Azure VM images must be in a fixed VHD format. The image must be aligned on a 1 MB boundary before it is converted to VHD. This section describes how to convert the image from **qcow2** to a fixed **VHD** format and align the image if necessary. Once you have converted the image, it can be uploaded to Microsoft Azure.

Procedure

Verifying the image size

1. Convert the image from **qcow2** to **raw** format.

```
$ qemu-img convert -f qcow2 -O raw <image-xxx>.qcow2 <image-xxx>.raw
```

2. Create a shell script using the contents below.

```
#!/bin/bash
MB=$((1024 * 1024))
size=$(qemu-img info -f raw --output json "$1" | gawk 'match($0,
/"virtual-size": ([0-9]+),/, val) {print val[1]}')
rounded_size=$((($size/$MB + 1) * $MB))
if [ $(($size % $MB)) -eq  0 ]
then
 echo "Your image is already aligned. You do not need to resize."
 exit 1
fi
echo "rounded size = $rounded_size"
export rounded_size
```

3. Run the script. The name **align.sh** is used in the example.

```
$ sh align.sh <image-xxx>.raw
```

- If the message *"Your image is already aligned. You do not need to resize."*is displayed, proceed to the following step.
- If a value is displayed, your image is not aligned. Go to Aligning the image and convert it to a fixed VHD format. Use the value displayed to resize the image.

4. Use the command below to convert the file to a fixed **VHD** format. The additional option **force_size** must be added when using Fedora 22 or later.
RHEL server or workstation (using qemu-image version 1.5.3)

```
$ qemu-img convert -f raw -o subformat=fixed -O vpc <image-xxx>.raw
<image.xxx>.vhd
```

Fedora 22 or later (using qemu-img version 2.6 or later)

```
$ qemu-img convert -f raw -o subformat=fixed,force_size -O vpc
<image-xxx>.raw <image.xxx>.vhd
```

Once converted, the VHD file is ready for uploading to Microsoft Azure.

Aligning the image

Complete the steps below only if the raw file is not aligned.

1. Resize the raw file using the rounded value displayed when you ran the verification script.

```
$ qemu-img resize -f raw <image-xxx>.raw <rounded-value>
```

2. Convert the raw image file to a VHD format.
RHEL server or workstation (using qemu-image version 1.5.3)

```
$ qemu-img convert -f raw -o subformat=fixed -O vpc <image-xxx>.raw
<image.xxx>.vhd
```

Fedora 22 or later (using qemu-img version 2.6 or later)

```
$ qemu-img convert -f raw -o subformat=fixed,force_size -O vpc
<image-xxx>.raw <image.xxx>.vhd
```

Once converted, the VHD file is ready for uploading to Microsoft Azure.

1.5. INSTALLING THE AZURE CLI

Complete the following steps to install the Azure command-line interface (Azure CLI 2.0). Azure CLI 2.0 is a Python-based utility that is used to create and manage VMs in Microsoft Azure.

Prerequisites

- You need to have an account with Microsoft Azure before you can use Azure CLI.
- The Azure CLI installation requires Python 2.7.x or Python 3.x and OpenSSL 1.0.2.

Procedure

1. Import the Microsoft repository key.

```
$ sudo rpm --import
https://packages.microsoft.com/keys/microsoft.asc
```

2. Create a local Azure CLI repository entry.

```
$ sudo sh -c 'echo -e "[azure-cli]\nname=Azure
CLI\nbaseurl=https://packages.microsoft.com/yumrepos/azure-
cli\nenabled=1\ngpgcheck=1\ngpgkey=https://packages.microsoft.com/ke
ys/microsoft.asc" > /etc/yum.repos.d/azure-cli.repo'
```

3. Update the **yum** package index.

```
$ yum check-update
```

4. Install the Azure CLI.

```
$ sudo yum install azure-cli
```

5. Run the Azure CLI.

```
$ az
```

Additional resources

- Azure CLI
- Azure CLI command reference

1.6. CREATING RESOURCES IN MICROSOFT AZURE

Complete the procedures in the following sections to upload the **vhd** file, create a gold Azure custom image, and start a RHEL VM in Microsoft Azure.

Procedure

1. Enter the following command to authenticate your system with Microsoft Azure and log in.

```
$ az login
```

Example:

```
[clouduser@localhost]$ az login
To sign in, use a web browser to open the page
https://aka.ms/devicelogin and enter the code FDMSCMETZ to
authenticate.
  [
    {
      "cloudName": "AzureCloud",
      "id": "",
      "isDefault": true,
      "name": "",
      "state": "Enabled",
      "tenantId": "",
      "user": {
        "name": "",
        "type": "user"
      }
    }
  ]
```

2. Create a resource group in an Azure region.

```
$ az group create --name <resource-group> --location <azure-region>
```

-

Example:

```
[clouduser@localhost]$ az group create --name azrhelclirsgrp --
location southcentralus
{
  "id": "/subscriptions//resourceGroups/azrhelclirsgrp",
  "location": "southcentralus",
  "managedBy": null,
  "name": "azrhelclirsgrp",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

3. Create a storage account. See SKU Types for more information about valid SKU values.

```
$ az storage account create -l <azure-region> -n <storage-account-
name> -g <resource-group> --sku <sku_type>
```

Example:

```
[clouduser@localhost]$ az storage account create -l southcentralus -
n azrhelclistact -g azrhelclirsgrp --sku Standard_LRS
{
  "accessTier": null,
  "creationTime": "2017-04-05T19:10:29.855470+00:00",
  "customDomain": null,
  "encryption": null,
  "id":
"/subscriptions//resourceGroups/azrhelclirsgrp/providers/Microsoft.S
torage/storageAccounts/azrhelclistact",
  "kind": "Storage",
  "lastGeoFailoverTime": null,
  "location": "southcentralus",
  "name": "azrhelclistact",
  "primaryEndpoints": {
    "blob": "https://azrhelclistact.blob.core.windows.net/",
    "file": "https://azrhelclistact.file.core.windows.net/",
    "queue": "https://azrhelclistact.queue.core.windows.net/",
    "table": "https://azrhelclistact.table.core.windows.net/"
},
"primaryLocation": "southcentralus",
"provisioningState": "Succeeded",
"resourceGroup": "azrhelclirsgrp",
"secondaryEndpoints": null,
"secondaryLocation": null,
"sku": {
  "name": "Standard_LRS",
  "tier": "Standard"
},
"statusOfPrimary": "available",
"statusOfSecondary": null,
```

```
"tags": {},
  "type": "Microsoft.Storage/storageAccounts"
}
```

4. Get the storage account connection string.

```
$ az storage account show-connection-string -n <storage-account-
name> -g <resource-group>
```

Example:

```
[clouduser@localhost]$ az storage account show-connection-string -n
azrhelclistact -g azrhelclirsgrp
{
  "connectionString":
"DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;Acco
untName=azrhelclistact;AccountKey=NreGk...=="
}
```

5. Export the connection string. Copy the connection string and paste it into the following command. This string connects your system to the storage account.

```
$ export AZURE_STORAGE_CONNECTION_STRING="<storage-connection-
string>"
```

Example:

```
[clouduser@localhost]$ export
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;Endp
ointSuffix=core.windows.net;AccountName=azrhelclistact;AccountKey=Nr
eGk...=="
```

6. Create the storage container.

```
$ az storage container create -n <container-name>
```

Example:

```
[clouduser@localhost]$ az storage container create -n
azrhelclistcont
{
  "created": true
}
```

7. Create a virtual network.

```
$ az network vnet create -g <resource group> --name <vnet-name> --
subnet-name <subnet-name>
```

Example:

```
[clouduser@localhost]$ az network vnet create --resource-group
azrhelclirsgrp --name azrhelclivnet1 --subnet-name azrhelclisubnet1
```

```
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
      "10.0.0.0/16"
      ]
  },
  "dhcpOptions": {
    "dnsServers": []
  },
  "etag": "W/\"\"",
  "id":
"/subscriptions//resourceGroups/azrhelclirsgrp/providers/Microsoft.N
etwork/virtualNetworks/azrhelclivnet1",
  "location": "southcentralus",
  "name": "azrhelclivnet1",
  "provisioningState": "Succeeded",
  "resourceGroup": "azrhelclirsgrp",
  "resourceGuid": "0f25efee-e2a6-4abe-a4e9-817061ee1e79",
  "subnets": [
    {
      "addressPrefix": "10.0.0.0/24",
      "etag": "W/\"\"",
      "id":
"/subscriptions//resourceGroups/azrhelclirsgrp/providers/Microsoft.N
etwork/virtualNetworks/azrhelclivnet1/subnets/azrhelclisubnet1",
      "ipConfigurations": null,
      "name": "azrhelclisubnet1",
      "networkSecurityGroup": null,
      "provisioningState": "Succeeded",
      "resourceGroup": "azrhelclirsgrp",
      "resourceNavigationLinks": null,
      "routeTable": null
    }
  ],
  "tags": {},
  "type": "Microsoft.Network/virtualNetworks",
  "virtualNetworkPeerings": null
  }
}
```


Additional resources

- Azure Managed Disks Overview
- SKU Types

1.7. UPLOADING AND CREATING AN AZURE GOLD IMAGE

Complete the following steps to upload the VHD file to your container and create a gold Azure custom image.

NOTE

The exported storage connection string does not persist after a system reboot. If any of commands in the following steps fail, export the connection string again. See Step 5 in Creating resources in Microsoft Azure.

Procedure

1. Upload the **vhd** file to the storage container. It may take several minutes. To get a list of storage containers, enter **`az storage container list`**.

```
$ az storage blob upload --account-name <storage-account-name> --
container-name <container-name> --type page --file <path-to-vhd> --
name <image-name>.vhd
```

Example:

```
[clouduser@localhost]$ az storage blob upload --account-name
azrhelclistact --container-name azrhelclistcont --type page --file
rhel-image-8.vhd --name rhel-image-8.vhd
Percent complete: %100.0
```

2. Get the URL for the uploaded **vhd** file. You will use this URL in the following step.

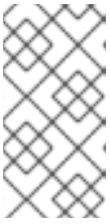
```
$ az storage blob url -c <container-name> -n <image-name>.vhd
```

Example:

```
[clouduser@localhost]$ az storage blob url -c azrhelclistcont -n
rhel-image-8.vhd
"https://azrhelclistact.blob.core.windows.net/azrhelclistcont/rhel-
image-8.vhd"
```

3. Create the gold Azure custom image.

```
$ az image create -n <gold-image-name> -g <resource-group> -l
<azure-region> --source <URL> --os-type linux
```



NOTE

The command may return the error "Only blobs formatted as VHDs can be imported." This error may mean that the image was not aligned to the nearest 1 MB boundary before it was converted to VHD. See Converting the image to fixed VHD format for more information.

Example:

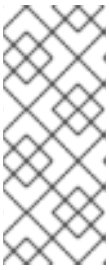
```
[clouduser@localhost]$ az image create -n rhel8 -g azrhelclirsgrp2 -
l southcentralus --source
https://azrhelclistact.blob.core.windows.net/azrhelclistcont/rhel-
image-8.vhd --os-type linux
```

1.8. CREATING AND STARTING THE VM IN MICROSOFT AZURE

The following steps provide the minimum command options to create a managed-disk Azure VM from the image. See az vm create for additional options.

Procedure

1. Enter the following command to create the VM.



NOTE

The option **`--generate-ssh-keys`** creates a private/public key pair. Private and public key files are created in **`~/.ssh`** on your system. The public key is added to the **`authorized_keys`** file on the VM for the user specified by the **`--admin-username`** option. See Other authentication methods for additional information.

```
$ az vm create -g <resource-group> -l <azure-region> -n <vm-name> --
vnet-name <vnet-name> --subnet <subnet-name> --size Standard_A2 --
os-disk-name <simple-name> --admin-username <administrator-name> --
generate-ssh-keys --image <path-to-image>
```

Example:

```
[clouduser@localhost]$ az vm create -g azrhelclirsgrp2 -l
southcentralus -n rhel-azure-vm-1 --vnet-name azrhelclivnet1 --
subnet azrhelclisubnet1  --size Standard_A2 --os-disk-name vm-1-
osdisk --admin-username clouduser --generate-ssh-keys --image rhel8

{
  "fqdns": "",
  "id":
"/subscriptions//resourceGroups/azrhelclirsgrp/providers/Microsoft.C
ompute/virtualMachines/rhel-azure-vm-1",
  "location": "southcentralus",
  "macAddress": "",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "<public-IP-address>",
  "resourceGroup": "azrhelclirsgrp2"
```

Note the **`publicIpAddress`**. You need this to log in to the VM in the following step.

2. Start an SSH session and log in to the VM.

```
[clouduser@localhost]$ ssh  -i /home/clouduser/.ssh/id_rsa
clouduser@<public-IP-address>.
The authenticity of host ',<public-IP-address>' can't be
established.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '<public-IP-address>' (ECDSA) to the list
of known hosts.

[clouduser@rhel-azure-vm-1 ~]$
```

-

If you see a user prompt, you have successfully deployed your Azure VM.

You can now go to the Microsoft Azure portal and check the audit logs and properties of your resources. You can manage your VMs directly in the Microsoft Azure portal. If you are managing multiple VMs, you should use the Azure CLI. The Azure CLI provides a powerful interface to your resources in Azure. Enter **`az --help`** in the CLI or see the Azure CLI Command Reference to learn more about the commands you use to manage your VMs in Microsoft Azure.

1.9. OTHER AUTHENTICATION METHODS

While recommended for increased security, the use of the Azure-generated key pair is not a requirement. The following examples show two other methods for SSH authentication.

Example 1: These command options provision a new VM without generating a public key file. They allow SSH authentication using a password.

```
$ az vm create -g <resource-group> -l <azure-region> -n <vm-name> --vnet-
name <vnet-name> --subnet <subnet-name> --size Standard_A2 --os-disk-name
<simple-name> --authentication-type password --admin-username
<administrator-name> --admin-password <ssh-password> --image <path-to-
image>
```

```
$ ssh <admin-username>@<public-ip-address>
```

Example 2: These command options provision a new Azure VM and allow SSH authentication using an existing public key file.

```
$ az vm create -g <resource-group> -l <azure-region> -n <vm-name> --vnet-
name <vnet-name> --subnet <subnet-name> --size Standard_A2 --os-disk-name
<simple-name> --admin-username <administrator-name> --ssh-dest-key-path
<path-to-existing-ssh-key> --image <path-to-image>
```

```
$ ssh -i <path-to-existing-ssh-key> <admin-username>@<public-ip-address>
```

CHAPTER 2. DEPLOYING AN EC2 INSTANCE ON AMAZON WEB SERVICES

The following topics provide step-by-step guidance for deploying a Red Hat Enterprise Linux 8.0 Beta EC2 instance on Amazon Web Services (AWS).

Prerequisites

- Enroll in Red Hat Cloud Access. Red Hat Cloud Access allows you to move your Red Hat subscriptions from physical or on-premise systems onto AWS with full support from Red Hat.
- **Important:** You need to sign up for AWS and set up your AWS resources first. See Setting Up with Amazon EC2.

Additional resources

- Red Hat Cloud Access
- Red Hat Enterprise Linux on Amazon EC2
- Setting Up with Amazon EC2

2.1. ABOUT RED HAT ENTERPRISE LINUX IMAGES ON AWS

There are several ways to get a Red Hat Enterprise Linux 8.0 Beta image from Red Hat to use on AWS.

- **Red Hat Cloud Access**
  - Once you enroll in the Red Hat Cloud Access program with eligible Red Hat subscriptions, you can access a library of Red Hat gold images shared with you on AWS. See Getting a Red Hat Cloud Access gold image for instructions.
- **Composer** provides a pre-built image for creating an EC2 instance for AWS. See Preparing Composer for Creating AWS AMI Images.
- The **AWS Marketplace** provides many Red Hat Enterprise Linux images. See the AWS Marketplace for additional information about available Red Hat images.

Additional resources

- Red Hat Cloud Access
- Red Hat Enterprise Linux on Amazon EC2 - FAQs

2.2. LISTING RED HAT CLOUD ACCESS GOLD IMAGES

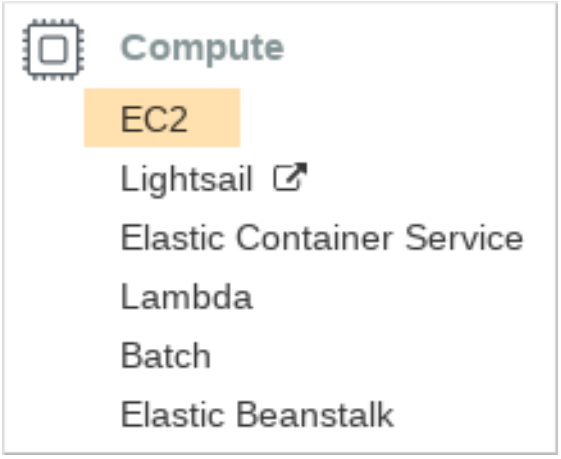
Complete the following steps to list Red Hat Cloud Access gold AWS images.

Prerequisites

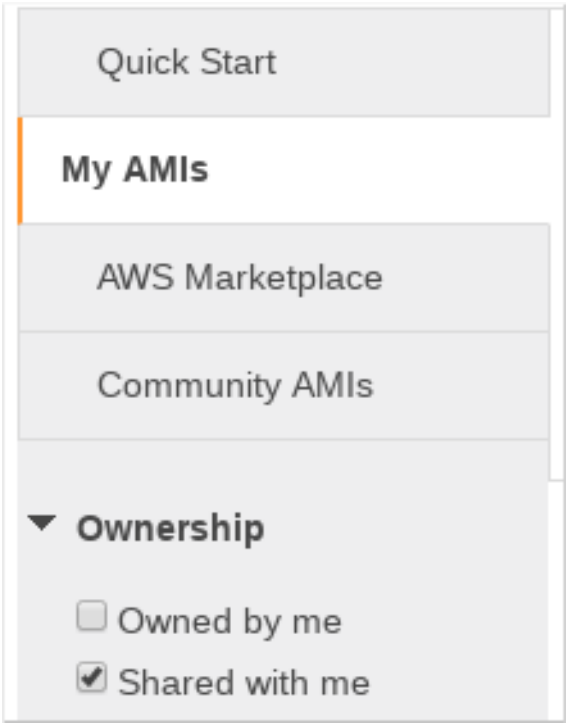
Enroll in the Red Hat Cloud Access program. Once you are enrolled and have available subscriptions, you are able to access Red Hat gold images.

Procedure

1. Launch the AWS Console.
2. Select **EC2** in the **Compute** category under services.

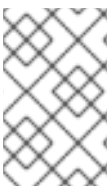


3. Click **Launch Instance** and select **My AMIs**.
4. Check the **Shared with Me** box. The Red Hat gold images made available through Red Hat Cloud Access are listed.



2.3. CREATING AN EC2 INSTANCE USING THE AWS CONSOLE

Complete the following steps to create an EC2 instance using the AWS console.



NOTE

The following procedure provides general instructions for creating a basic Red Hat Enterprise Linux instance. The steps are based on Launching an Instance Using the Launch Instance Wizard. Refer to the AWS procedure for details and updates.

Prerequisites

Set up your AWS resources before completing the procedure. See Setting Up with Amazon EC2 for details. You should not start the procedure if resources have not been created.

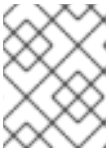
Procedure

1. Launch the AWS Management Console.
2. Select **EC2** in the **Compute** category under services.
3. Click **Launch Instances**.
4. Choose an Amazon Machine Image (AMI). Note that the Red Hat Cloud Access program provides a library of shared **Red Hat gold images**. See Listing Red Hat Cloud Access gold images for instructions.
5. Choose an **Instance Type** that meets your capacity requirements. The recommended capacity for a base Red Hat Enterprise Linux instance is **General purpose, t2.medium**. See Amazon EC2 Pricing for additional details.
6. Click **Next: Configure Instance Details**.
7. Enter the **Number of instances** you want to create.
8. For **Network**, select the VPC you created when setting up your AWS environment. Select a subnet for the instance or create a new subnet.
9. Select **Enable** for Auto-assign Public IP.

NOTE

The selections above are the minimum configuration options necessary to create a basic instance. You may want to review additional options based on your application requirements.

10. Click **Next: Add Storage**. Verify that the default storage is sufficient.
11. Click **Next: Add Tags**. Add a label for the instance(s). This can be anything that is simple and easy to identify later. Make sure to adhere to AWS tagging guidelines.
12. Click **Next: Configure Security Group**. Select the security group you created when setting up your AWS environment.
13. Click **Review and Launch**. Verify your selections.
14. Click **Launch**. You are prompted to select an existing key pair or create a new key pair. Select the key pair you created when setting up your AWS environment.



NOTE

Verify that the permissions for your private key are correct. Use the command options **`chmod 400 <keyname>.pem`** to change the permissions if necessary.

15. Click **Launch Instances**.

16. Click **View Instances**. You can name the instance(s) here. You may want to use the RHEL host name.

You can now launch an SSH session to your instance(s) by selecting an instance and clicking **Connect**. Use the example provided for **A standalone SSH client**.