Install, Deploy, and Maintain the VMware vSphere Integrated Containers Infrastructure

vSphere Integrated Containers 1.2

vmware®

Table of Contents

Install, Deploy, and Maintain the vSphere Integrated Containers Infrastructure

Install, Deploy, and Maintain the vSphere Integrated Containers Infrastructure provides information about how to use VMware vSphere® Integrated Containers™ as a vSphere administrator.

Product version: 1.2

This documentation applies to all 1.2.x releases.

Intended Audience

This information is intended for VMware vSphere® administrators who want to install and set up vSphere Integrated Containers. The information is written for experienced vSphere administrators who are familiar with virtual machine technology and datacenter operations. Knowledge of container technology and Docker is useful.

Installing vSphere Integrated Containers

You install vSphere Integrated Containers by deploying an OVA appliance. The OVA appliance provides access to all of the vSphere Integrated Containers components and services.

- Download vSphere Integrated Containers
- Deploy the vSphere Integrated Containers Appliance
- Install the vSphere Client plug-ins on vCenter Server
- Open the Required Ports on ESXi Hosts

Download vSphere Integrated Containers

You can download different versions of vSphere Integrated Containers, that have different levels of stability and support.

Official Releases

To obtain the latest official release of vSphere Integrated Containers, go to the official vSphere Integrated Containers downloads page on vmware.com and download the OVA installer. The OVA installer allows you to deploy all of the vSphere Integrated Containers components.

Full support of vSphere Integrated Containers requires the vSphere Enterprise Plus license. To make a support request, contact VMware Global Support.

Open Source Builds of vSphere Integrated Containers

You can obtain open source builds of vSphere Integrated Containers Engine, vSphere Integrated Containers Portal, and vSphere Integrated Containers Registry, that have different levels of stability.

- Download recent builds of the vSphere Integrated Containers OVA installer. Builds happen approximately weekly. You need a Google account to access these builds.
- Download tagged open source software (OSS) versions of the vSphere Integrated Containers components that have been tested and released to the open source community, but that might not reflect the most up-to-date version of the code:
  - vSphere Integrated Containers Engine
  - vSphere Integrated Containers Registry
  - vSphere Integrated Containers Portal
- Download built vSphere Integrated Containers Engine binaries. Builds usually happen after every successful merge into the source code. These builds have been minimally tested for integration. You need a Google account to access these builds.
- Build the latest source version of the vSphere Integrated Containers components:
  - vSphere Integrated Containers Engine
  - vSphere Integrated Containers Registry
  - vSphere Integrated Containers Portal

IMPORTANT: Open source builds are not supported by VMware Global Support.

- You can obtain community support for open source builds by reporting bugs and creating issues on Github.
- For general questions, visit the vSphere Integrated Containers channel on Slack.com. If you do not have an @vmware.com or @emc.com email address, sign up at https://code.vmware.com/home to get an invitation.

Deploy the vSphere Integrated Containers Appliance

You install vSphere Integrated Containers by deploying a virtual appliance. The appliance runs the vSphere Integrated Containers Registry and vSphere Integrated Containers Management Portal services, and publishes the downloads of the vSphere Integrated Containers Engine binaries.

NOTE: This topic reflects changes to the deployment procedure that were introduced in vSphere Integrated Containers 1.2.1. Do not attempt to deploy version 1.2.0.

Prerequisites

- You downloaded an official build or an open-source build of the OVA installer.

  - Download official builds from the vSphere Integrated Containers downloads page on vmware.com.
  - Download open-source builds from the vSphere Integrated Containers repository on Google Cloud Platform.
- Deploy the appliance to a vCenter Server instance. Deploying the appliance directly on an ESXi host is not supported.
- Deploy the appliance to a vCenter Server system that meets the minimum system requirements:

  - 2 vCPUs
  - 8GB RAM
  - 80GB free disk space on the datastore
- Ensure that all vCenter Server instances and ESXi hosts in the environment in which you are deploying the appliance have network time protocol (NTP) running. Running NTP prevents problems arising from clock skew between the vSphere Integrated Containers appliance, virtual container hosts, and the vSphere infrastructure.
- **IMPORTANT**: If you intend to use a custom certificates, vSphere Integrated Containers Management Portal requires the TLS private key to be supplied as a PEM-encoded PKCS#8-formatted file. For information about how to convert keys to the correct format, see Converting Keys for Use with vSphere Integrated Containers Management Portal.
- You can deploy multiple vSphere Integrated Containers appliances to the same vCenter Server instance. Also, if a Platform Services Controller manages multiple vCenter Server instances, you can deploy multiple appliances to different vCenter Server instances that share that Platform Services Controller.
- Use the Flex-based vSphere Web Client to deploy the appliance. You cannot deploy OVA files from the HTML5 vSphere Client or from the legacy Windows client.

Procedure

1. In the vSphere Web Client, right-click an object in the vCenter Server inventory, select **Deploy OVF template**, and navigate to the OVA file.
2. Follow the installer prompts to perform basic configuration of the appliance and to select the vSphere resources for it to use.

   - Accept or modify the appliance name
   - Select the destination datacenter or folder
   - Select the destination host, cluster, or resource pool
   - Accept the end user license agreements (EULA)
   - Select the disk format and destination datastore

- Select the network that the appliance connects to
3. On the **Customize template** page, under **Appliance Security**, set the root password for the appliance VM and optionally uncheck the **Permit Root Login** checkbox.

   Setting the root password for the appliance is mandatory.

   IMPORTANT: You require SSH access to the vSphere Integrated Containers appliance to perform upgrades. You can also use SSH access in exceptional cases that you cannot handle through standard remote management or CLI tools. Only use SSH to access the appliance when instructed to do so in the documentation, or under the guidance of VMware GSS.

4. Expand **Networking Properties** and optionally configure a static IP address for the appliance VM.

   To use DHCP, leave the networking properties blank.

   IMPORTANT: If you set a static IP address for the appliance, use spaces to separate DNS servers. Do not use comma separation for DNS servers.

5. Expand **Registry Configuration** to configure the deployment of vSphere Integrated Containers Registry.

   - In the **Registry Port** text box, optionally change the port on which to publish the vSphere Integrated Containers Registry service.
   - In the **Notary Port** text box, optionally change the port on which to publish the Docker Content Trust service for vSphere Integrated Containers Registry.
   - Optionally check the **Garbage Collection** check box to enable garbage collection on the registry when the appliance reboots.

6. Expand **Management Portal Configuration** to configure the deployment of vSphere Integrated Containers Management Portal.

   - In the **Management Portal Port** text box, optionally change the port on which to publish the vSphere Integrated Containers Management Portal service.
   - To use custom certificates to authenticate connections to vSphere Integrated Containers Management Portal, optionally paste the content of the appropriate certificate, key, and Certificate Authority (CA) files in the **SSL Cert**, **SSL Cert Key**, and **CA Cert** text boxes.

     IMPORTANT: Provide the TLS private key as a PEM-encoded PKCS#8-formatted file.

   - Leave the text boxes blank to use auto-generated certificates.

7. Expand **Fileserver Configuration** to configure the file server from which you download the vSphere Integrated Containers Engine binaries, and which publishes the plug-in packages for the vSphere Client.

   - In the **Fileserver Port** text box, optionally change the port on which the vSphere Integrated Containers Engine file server runs.
   - To use custom certificates to authenticate connections to the vSphere Integrated Containers Engine file server, optionally paste the content of the appropriate certificate and key files in the **SSL Cert** and **SSL Cert Key** text boxes. The file server supports RSA format for TLS private keys.
   - Leave the text boxes blank to use auto-generated certificates.

8. Expand **Demo VCH Installer Wizard Configuration** to optionally change the port on which the interactive web installer for virtual container hosts (VCHs) runs.

9. Expand **Configure Example Users** to configure ready-made vSphere Integrated Containers user accounts in the Platform Services Controller.

   You can use these accounts to test the different user personas that can access vSphere Integrated Containers Management Portal and Registry.

   - Optionally uncheck the **Create Example Users** checkbox to disable the creation of example user accounts.
   - In the **Username Prefix for Example Users** text box, optionally modify the prefix of the example user names from the default, `vic` .
   - In the **Password for Example Users** text boxes, optionally modify the password for the example user account from the default, `VicPro!23` .

10. Click **Next** and **Finish** to deploy the vSphere Integrated Containers appliance.
11. When the deployment completes, power on the appliance VM.

    If you deployed the appliance so that it obtains its address via DHCP, go to the **Summary** tab for the appliance VM and note the address.

12. (Optional) If you provided a static network configuration, view the network status of the appliance.

    i. In the **Summary** tab for the appliance VM, launch the VM console
    ii. In the VM console, press the right arrow key.

    The network status shows whether the network settings that you provided during the deployment match the settings with which the appliance is running. If there are mismatches, power off the appliance and select **Edit Settings** > **vApp Options** to correct the network settings.

13. In a browser, go to http://*vic_appliance_address* and when prompted, enter the connection details for the vCenter Server instance on which you deployed the appliance.

    - The address and single sign-on credentials of vCenter Server.
    - If vCenter Server is managed by an external Platform Services Controller, enter the FQDN and administrator domain for the Platform Services Controller. If vCenter Server is managed by an embedded Platform Services Controller, leave the External PSC text boxes empty.

    IMPORTANT: The installation process requires the single sign-on credentials to register vSphere Integrated Containers Management Portal and Registry with the Platform Services Controller and to tag the appliance VM for use in Docker content trust. The vSphere Integrated Containers Management Portal and Registry services cannot start if you do not complete this step.

14. Click **Continue** to initialize the appliance.

Result

- You see the vSphere Integrated Containers Getting Started page at http://*vic_appliance_address*. The Getting Started page includes links to the vSphere Integrated Containers Management Portal, the Demo VCH Installer Wizard, the download for the vSphere Integrated Containers Engine bundle, and to documentation.
- If you see the error `Failed to register with PSC. Please check the vSphere user domain PSC settings and try again` , see the procedure in vSphere Integrated Containers Appliance Fails to Register with PSC to register vSphere Integrated Containers with the Platform Services Controller.
- If the initialization fails with any other error, see Reinitialize the vSphere Integrated Containers Appliance. You should not reinitialize the appliance in any circumstances other than those described in that topic.

What to Do Next

Access the different vSphere Integrated Containers components from the vSphere Integrated Containers Getting Started page at http://*vic_appliance_address*.

- Click the link to go to the **vSphere Integrated Containers Management Portal**. For information about how to use vSphere Integrated Containers Management Portal, see Configure and Manage vSphere Integrated Containers.
- Scroll down to **Infrastructure deployment tools** and click the link to go to the **Demo VCH Installer Wizard**. For information about how to use the interactive demo VCH installer, see Deploy a Virtual Container Host Interactively.
- Scroll down to **Infrastructure deployment tools** and click the link to **download the vSphere Integrated Containers Engine bundle**. The vSphere Integrated Containers Engine bundle allows you to perform the following tasks:

  - Use `vic-machine` to configure the firewalls on all ESXi hosts to permit VCH deployment. For information about how to configure the firewalls on ESXi hosts, see Open the Required Ports on ESXi Hosts.
  - Install the vSphere Client plug-ins for vSphere Integrated Containers. For information about installing the plug-ins, see Installing the vSphere Client Plug-ins.
  - Use `vic-machine` to deploy production VCHs. For information about deploying VCHs with `vic-machine`, see Deploy Virtual Container Hosts with `vic-machine` .

- To remove security warnings when you connect to the Getting Started page or management portal, see Obtain the Thumbprints and CA Files of the vSphere Integrated Containers Appliance Certificates and Verify and Trust vSphere Integrated Containers Appliance Certificates.

- If you see a certificate error when you attempt to go to http://*vic_appliance_address*, see Browser Rejects Certificates with `ERR_CERT_INVALID` Error.

- If necessary, you can reconfigure the appliance after deployment by editing the settings of the appliance VM. For information about reconfiguring the appliance, see Reconfigure the vSphere Integrated Containers Appliance.

Installing the vSphere Client Plug-Ins

vSphere Integrated Containers provides a basic plug-in for the Flex-based vSphere Web Client on vCenter Server 6.0 or 6.5. vSphere Integrated Containers provides a plug-in with more complete functionality for the HTML5 vSphere Client. The HTML5 vSphere Client is only available with vSphere 6.5.

You can deploy the plug-ins on a vCenter Server instance that runs on Windows, or on a vCenter Server Appliance.

For information about the Flex-based vSphere Web Client and the HTML5 vSphere Client for vSphere 6.5, see Introduction to the vSphere Client in the vSphere 6.5 documentation.

- Install the Client Plug-Ins on vCenter Server for Windows
- Install the Client Plug-Ins on a vCenter Server Appliance

Install the Client Plug-Ins on vCenter Server for Windows

To install the vSphere Client plug-ins for vSphere Integrated Containers, you log in to the Windows system on which vCenter Server runs and run a script. The script registers an extension with vCenter Server, and instructs vCenter Server to download the plug-in files from the file server in the vSphere Integrated Containers appliance.

The installer installs a basic plug-in for the Flex-based vSphere Web Client on vCenter Server 6.0 or 6.5 and a plug-in with more complete functionality for the HTML5 vSphere Client on vCenter Server 6.5.

Prerequisites

- The HTML5 plug-in requires vCenter Server 6.5.0d or later. The HTML5 plug-in does not function with earlier versions of vCenter Server 6.5.0.
- The vCenter Server instance on which to install the plug-in runs on Windows. If you are running a vCenter Server appliance instance, see Install the Client Plug-Ins on a vCenter Server Appliance.
- You deployed the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- Log in to the Windows system on which vCenter Server is running. You must perform all of the steps in this procedure on this Windows system.

  IMPORTANT: The upgrade script does not function if you have set the `VIC_MACHINE_THUMBPRINT` environment variable on the system on which you run the script. Delete the `VIC_MACHINE_THUMBPRINT` environment variable before running the script.

- In a Web browser, go to http://*vic_appliance_address*, scroll down to Infrastructure Deployment Tools, click the link to **download the vSphere Integrated Containers Engine bundle**, and unpack it on the Desktop.
- Obtain the vCenter Server certificate thumbprint. For information about how to obtain and verify the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

Procedure

1. Run the install script and follow the prompts.

   ```
   %USERPROFILE%\Desktop\vic\ui\vCenterForWindows\install.bat
   ```

   i. Enter the IP address of the vCenter Server instance.
   ii. Enter the user name and password for the vCenter Server administrator account.
   iii. Enter **yes** if the vCenter Server certificate thumbprint is legitimate, and wait for the install process to finish.
2. When the installation finishes, stop and restart the services of your management clients.
   i. Restart the HTML5 vSphere Client service.

   ```
   service-control --stop vsphere-ui && service-control --start vsphere-ui
   ```

   ii. Restart the Flex-based vSphere Web Client service.

   ```
   service-control --stop vsphere-client && service-control --start vsphere-client
   ```

What to Do Next

To verify the deployment of the plug-ins, see VCH Administration in the vSphere Client.

Install the Client Plug-Ins on a vCenter Server Appliance

You install the vSphere Client plug-ins for vSphere Integrated Containers by logging into the vCenter Server appliance and running a script. The script registers an extension with vCenter Server, and instructs vCenter Server to download the plug-in files from the file server in the vSphere Integrated Containers appliance.

The installer installs a basic plug-in for the Flex-based vSphere Web Client on vCenter Server 6.0 or 6.5 and a plug-in with more complete functionality for the HTML5 vSphere Client on vCenter Server 6.5.

Prerequisites

- The HTML5 plug-in requires vCenter Server 6.5.0d or later. The HTML5 plug-in does not function with earlier versions of vCenter Server 6.5.0.
- You are installing the plug-ins on a vCenter Server appliance instance. If you are running vCenter Server on Windows, see Install the Client Plug-Ins on vCenter Server for Windows.
- Go to the vCenter Server Appliance Management Interface (VAMI) at https://*vcsa_address*:5480, log in as the appliance `root` user, then click **Access**, and make sure that SSH Login and Bash Shell are enabled.
- You deployed the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- Obtain the vCenter Server certificate thumbprint. For information about how to obtain and verify the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.
- The system on which you run the script is running `awk`.

IMPORTANT: The upgrade script does not function if you have set the `VIC_MACHINE_THUMBPRINT` environment variable on the system on which you run the script. Delete the `VIC_MACHINE_THUMBPRINT` environment variable before running the script.

Procedure

1. Connect as root user to the vCenter Server Appliance by using SSH.

   ```
   $ ssh root@vcsa_address
   ```

2. Start bash.

   ```
   $ shell
   ```

3. Use `curl` to copy the vSphere Integrated Containers Engine binaries from the vSphere Integrated Containers appliance file server to the vCenter Server Appliance.

   ```
   $ curl -kL  http://vic_appliance_address/files/vic_1.2.x.tar.gz -o vic_1.2.x.tar.gz
   ```

   NOTE: Update `vic_1.2.x` to the appropriate version in the command above and in the next step. You can see the full file name by going to http://*vic_appliance_address*/files/ in a browser.

4. Unpack the vSphere Integrated Containers binaries.

```
$ tar -zxf vic_1.2.x.tar.gz
```

5. Navigate to `/vic/ui/VCSA` , run the installer script, and follow the prompts.

```
$ cd vic/ui/VCSA
```

```
$ ./install.sh
```

    i. Enter the IP address of the vCenter Server instance.
   ii. Enter the user name and password for the vCenter Server administrator account.
  iii. Enter **yes** if the vCenter Server certificate thumbprint is legitimate, and wait for the install process to finish.

6. When the installation finishes, stop and restart the services of your management clients.
    i. Restart the HTML5 vSphere Client service.

```
$ service-control --stop vsphere-ui && service-control --start vsphere-ui
```

   ii. Restart the Flex-based vSphere Web Client service.

```
$ service-control --stop vsphere-client && service-control --start vsphere-client
```

What to Do Next

To verify the deployment of the plug-ins, see VCH Administration in the vSphere Client.

Open the Required Ports on ESXi Hosts

ESXi hosts communicate with the virtual container hosts (VCHs) through port 2377 via Serial Over LAN. For the deployment of a VCH to succeed, port 2377 must be open for outgoing connections on all ESXi hosts before you run `vic-machine create` to deploy a VCH. Opening port 2377 for outgoing connections on ESXi hosts opens port 2377 for inbound connections on the VCHs.

The `vic-machine` utility includes an `update firewall` command, that you can use to modify the firewall on a standalone ESXi host or all of the ESXi hosts in a cluster.

You use the `--allow` and `--deny` flags to enable and disable a firewall rule named `vSPC`. When enabled, the `vSPC` rule allows all outbound TCP traffic from the target host or hosts. If you disable the rule, you must configure the firewall via another method to allow outbound connections on port 2377 over TCP. If you do not enable the rule or configure the firewall, vSphere Integrated Containers Engine does not function, and you cannot deploy VCHs.

The `vic-machine create` command does not modify the firewall. Run `vic-machine update firewall --allow` before you run `vic-machine create`.

Prerequisites

- Deploy the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- In a Web browser, go to http://*vic_appliance_address*, scroll down to Infrastructure Deployment Tools, click the link to **download the vSphere Integrated Containers Engine bundle**, and unpack it on your working machine.
- If your vSphere environment uses untrusted, self-signed certificates, you must specify the thumbprint of the vCenter Server instance or ESXi host in the `--thumbprint` option. For information about how to obtain the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

Procedure

1. Open a terminal on the system on which you downloaded and unpacked the vSphere Integrated Containers Engine binary bundle.
2. Navigate to the directory that contains the `vic-machine` utility:
3. Run the `vic-machine update firewall` command.

   To open the appropriate ports on all of the hosts in a vCenter Server cluster, run the following command:

   ```
   $ vic-machine-operating_system update firewall
   --target vcenter_server_address
   --user "Administrator@vsphere.local"
   --password vcenter_server_password
   --compute-resource cluster_name
   --thumbprint thumbprint
   --allow
   ```

   To open the appropriate ports on an ESXi host that is not managed by vCenter Server, run the following command:

```
$ vic-machine-operating_system update firewall
--target esxi_host_address
--user root
--password esxi_host_password
--thumbprint thumbprint
--allow
```

The `vic-machine update firewall` command in these examples specifies the following information:

- The address of the vCenter Server instance and datacenter, or the ESXi host, on which to deploy the VCH in the `--target` option.
- The user name and password for the vCenter Server instance or ESXi host in the `--user` and `--password` options.
- In the case of a vCenter Server cluster, the name of the cluster in the `--compute-resource` option.
- The thumbprint of the vCenter Server or ESXi host certificate in the `--thumbprint` option, if they use untrusted, self-signed certificates.

  Use upper-case letters and colon delimitation in the thumbprint. Do not use space delimitation.

- The `--allow` option to open the port.

Deploy Virtual Container Hosts Interactively

The vSphere Integrated Containers appliance provides an interactive web installer from which you can deploy a basic virtual container host (VCH). This VCH has limited cabilities and is for demonstration purposes only, to allow you to start experimenting with vSphere Integrated Containers.

IMPORTANT: The demo VCH has the minimum configuration that deployment to vCenter Server requires. Only the bridge network, public network, image store, and compute resource options are currently supported. As a consequence, the demo VCH has the following limitations:

- It does not implement any TLS authentication options, and as such is completely unsecured. Do not use the demo VCH in production environments.
- It only supports DHCP. You cannot assign a static IP address to the VCH on any of its interfaces.
- It has no option create a volume store. Any containers that create volumes will fail to start if you run them in the demo VCH.

To deploy VCHs for use in production environments, use the `vic-machine` command line utility.

Prerequisites

- You deployed the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- You opened port 2377 for outgoing connections on all ESXi hosts in your vCenter Server environment. For information about opening port 2377, see Open the Required Ports on ESXi Hosts.
- You must create two distributed port groups, one each for the bridge and public networks, on the vCenter Server instance on which to deploy the VCH. For information about how to create a distributed virtual switch and a port group, see Create a vSphere Distributed Switch and Add Hosts to a vSphere Distributed Switch in the vSphere documentation.
- Obtain the vCenter Server certificate thumbprint. For information about how to obtain and verify the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

NOTE: When using `vic-machine` to deploy VCHs, if you do not specify a network or port group for the public network, the VCH uses the VM Network by default. However, because the VM Network might not be present, the demo VCH requires that you create a port group for the public network.

Procedure

1. In a Web browser, go to http://*vic_appliance_address*, scroll down to Infrastructure Deployment Tools, click the link **Go to the Demo VCH Installer Wizard**, and trust the certificate.
2. Enter the IP address and administrator credentials of the vCenter Server instance on which the vSphere Integrated Containers appliance is running and click **Login**.
3. Use the drop-down menus to select the appropriate resources for each of the required resources.

Option	Description
Bridge Network	Select an existing distributed port group for container VMs use to communicate with each other.
Public Network	Select a different distributed port group for container VMs use to connect to the

Public Network	internet.
Image Store	Select a datastore in which to store container images that you pull into the VCH.
Compute Resource	Select the host, cluster, or resource pool in which to deploy the VCH.

4. (Optional) Modify the name of the VCH to create.

5. Enter the vCenter Server certificate thumbprint into the **Thumbprint** text box and click **Execute**.

   If you leave **Thumbprint** empty, the deployment of the VCH fails, and the certificate thumbprint of the target vCenter Server appears under **Execution Output**. Verify that the thumbprint is valid. If it is valid, copy and paste the thumprint in **Thumbprint** and click **Execute** again.

You can monitor the progress of the VCH deployment under **Execution Output**. Stay on the Installer page until the command finishes. Logs might stop streaming if you switch to other tabs or windows.

Result

At the end of a successful deployment, **Execution Output** displays information about the new VCH.

What to Do Next

- Connect a Docker client to the VCH and run Docker commands against it. For information about running Docker commands against a VCH, see Verify the Deployment of a VCH.
- Copy the generated command output under **Create Command** and use it as a template for use with `vic-machine` to create production VCHs. For information about how to deploy production VCHs, see Using `vic-machine` to Deploy Virtual Container Hosts.
- Go to http://*vic_appliance_address*, click the link to **Go to the vSphere Integrated Containers Management Portal**, enter the vCenter Server Single Sign-On credentials, and go to **Home** > **Clusters**. After creating a VCH, the web installer adds the VCH to the project `default-project` in the management portal instance that is running in the vSphere Integrated Containers appliance.

Deploy Virtual Container Hosts with `vic-machine`

After you deploy the vSphere Integrated Containers appliance, go to http://*vic_appliance_address* in a Web browser, scroll down to Infrastructure Deployment Tools, click the link to **download the vSphere Integrated Containers Engine bundle**, and unpack it on your working machine.

The vSphere Integrated Containers Engine binaries include the `vic-machine` utility, that you use to deploy virtual container hosts (VCHs).

- Contents of the vSphere Integrated Containers Engine Binaries
- Environment Prerequisites for VCH Deployment
- Obtain Certificate Thumbprints
- Deploy a VCH to an ESXi Host with No vCenter Server
- Deploy a VCH to a Basic vCenter Server Cluster
- Verify the Deployment of a VCH
- VCH Deployment Options
- Set Environment Variables
- Advanced Examples of Deploying a VCH
- VCH for Use with Registry
- VCH for Use with `dch-photon`
- Use Different User Accounts for VCH Deployment and Operation

Contents of the vSphere Integrated Containers Engine Binaries

After you deploy the vSphere Integrated Containers appliance, you download the vSphere Integrated Containers Engine bundle from http://*vic_appliance_address*.

When you unpack the vSphere Integrated Containers Engine bundle, you obtain following files:

File	Description
`appliance.iso`	The Photon based boot image for the virtual container host (VCH) endpoint VM.
`bootstrap.iso`	The Photon based boot image for the container VMs.
`ui/`	A folder that contains the files and scripts for the installation of the vSphere Client plug-in.
`vic-machine-darwin`	The OSX command line utility for the deployment and management of VCHs.
`vic-machine-linux`	The Linux command line utility for the deployment and management of VCHs.
`vic-machine-windows.exe`	The Windows command line utility for the deployment and management of VCHs.
`vic-ui-darwin`	The OSX executable for the deployment of the vSphere Client plug-in. **NOTE**: Do not run this executable directly.[1]
`vic-ui-linux`	The Linux executable for the deployment of the vSphere Client plug-in. **NOTE**: Do not run this executable directly.[1]
`vic-ui-windows.exe`	The Windows executable for the deployment of the vSphere Client plug-in. **NOTE**: Do not run this executable directly.[1]
`README`	Contains a link to the vSphere Integrated Containers Engine repository on GitHub.
`LICENSE`	The license file.

[1] For information about how to install the vSphere Client plug-in, see Installing the vSphere Client Plug-Ins.

Environment Prerequisites for VCH Deployment

Before you deploy virtual container hosts (VCHs), you must ensure that your vSphere infrastructure meets the requirements.

- Supported Platforms for `vic-machine`
- Supported vSphere Configurations
- License Requirements
- ESXi Host Firewall Requirements
- ESXi Host Storage Requirements for vCenter Server Clusters
- Networking Requirements

Supported Platforms for `vic-machine`

The vSphere Integrated Containers management utility, `vic-machine`, has been tested and verified on the following 64-bit Windows, Mac OS, and Linux OS systems.

Platform	Supported Versions
Windows	7, 10
Mac OS X	10.11 (El Capitan)
Linux	Ubuntu 16.04 LTS

Other recent 64-bit OS versions should work but are untested.

Supported vSphere Configurations

You can deploy vSphere Integrated Containers Engine in the following vSphere setups:

- vCenter Server 6.0 or 6.5, managing a cluster of ESXi 6.0 or 6.5 hosts, with VMware vSphere Distributed Resource Scheduler™ (DRS) enabled.
- vCenter Server 6.0 or 6.5, managing one or more standalone ESXi 6.0 or 6.5 hosts.
- Standalone ESXi 6.0 or 6.5 host that is not managed by a vCenter Server instance.

Caveats and limitations:

- VMware does not support the use of nested ESXi hosts, namely running ESXi in virtual machines. Deploying vSphere Integrated Containers Engine to a nested ESXi host is acceptable for testing purposes only.
- If you deploy a virtual container host (VCH) onto an ESXi host that is not managed by vCenter Server, and you then move that host into a cluster, the VCH might not function correctly.

License Requirements

vSphere Integrated Containers Engine requires a vSphere Enterprise Plus license.

All of the ESXi hosts in a cluster require an appropriate license. Deployment fails if your environment includes one or more ESXi hosts that have inadequate licenses.

ESXi Host Firewall Requirements

To be valid targets for VCHs and container VMs, ESXi hosts must have the following firewall configuration:

- Allow outbound TCP traffic to port 2377 on the endpoint VM, for use by the interactive container shell.
- Allow inbound HTTPS/TCP traffic on port 443, for uploading to and downloading from datastores.

These requirements apply to standalone ESXi hosts and to ESXi hosts in vCenter Server clusters.

For information about how to open ports on ESXi hosts, see Open the Required Ports on ESXi Hosts.

ESXi Host Storage Requirements for vCenter Server Clusters

ESXi hosts in vCenter Server clusters must meet the following storage requirements in order to be usable by a VCH:

- Be attached to the datastores that you will use for image stores and volume stores.
- Have access to shared storage to allow VCHs to use more than one host in the cluster.

For information about image stores and volumes stores, see the Datastore Options section of *VCH Deployment Options*.

Networking Requirements

The following network requirements apply to deployment of VCHs to standalone ESXi hosts and to vCenter Server:

- Use a trusted network for the deployment and use of vSphere Integrated Containers Engine.
- Use a trusted network for the management network.
- Connections between Docker clients and the VCH are encrypted via TLS unless you explicitly disable TLS. The client network does not need to be trusted.
- Each VCH requires an IPv4 address on each of the networks that it is connected to. The bridge network is handled internally, but other interfaces must have a static IP configured on them, or be able to acquire one via DHCP.
- Each VCH requires access to at least one network, for use as the public network. You can share this network between multiple VCHs. The public network does not need to be trusted.

The following network requirements apply to the deployment of VCHs to vCenter Server:

- Create a distributed virtual switch with a port group for each VCH, for use as the bridge network. You can create multiple port groups on the same distributed virtual switch, but each VCH requires its own port group for the bridge network.
- Optionally create port groups for use as mapped container networks, or for the public, management, and client

networks.

- All hosts in a cluster must be attached to the port groups that you will use for the VCH bridge network and for any mapped container networks.
- Isolate the bridge network and any mapped container networks. You can isolate networks by using a separate VLAN for each network.

For information about bridge networks and container networks, see Configure Bridge Networks and Configure Container Networks.

For information about how to create a distributed virtual switch and a port group, see Create a vSphere Distributed Switch in the vSphere documentation.

For information about how to add hosts to a distributed virtual switch, see Add Hosts to a vSphere Distributed Switch in the vSphere documentation.

For information about how to assign a VLAN ID to a port group, see VMware KB 1003825. For more information about private VLAN, see VMware KB 1010691.

Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host

If your vSphere environment uses untrusted, self-signed certificates, you must specify the thumbprint of the vCenter Server or ESXi host certificate in the `--thumbprint` option of each `vic-machine` command. If your vSphere environment uses trusted certificates that are signed by a known Certificate Authority (CA), you do not need to specify the `--thumbprint` option.

If you do not specify the `--thumbprint` option, `vic-machine` commands fail with certificate verification errors that include the thumbprint of the target. In this case, you should verify that the thumbprint in the error message is valid before attempting to run `vic-machine` again.

You can use either SSH and OpenSSL or the Platform Services Controller to obtain certificate thumbprints, either before you run `vic-machine` commands, or to confirm that a thumbprint in an error message is valid.

After you obtain the certificate thumbprint from vCenter Server or an ESXi host, you can set it as a `vic-machine` environment variable so that you do not have to specify `--thumbprint` in every command. For information about setting `vic-machine` environment variables, see Set Environment Variables for Key vic-machine Options.

Obtain Certificate Thumbprints from a vCenter Server Appliance or ESXi Host

You can use SSH and OpenSSL to obtain the certificate thumbprint for a vCenter Server Appiance instance or an ESXi host.

1. Use SSH to connect to the vCenter Server Appliance or ESXi host as `root` user.

   ```
   $ ssh root@vcsa_or_esxi_host_address
   ```

2. Use `openssl` to view the certificate fingerprint.

   - vCenter Server Appliance:

     ```
     openssl x509 -in /etc/vmware-vpx/ssl/rui.crt -fingerprint -sha1 -noout
     ```

   - ESXi host:

     ```
     openssl x509 -in /etc/vmware/ssl/rui.crt -fingerprint -sha1 -noout
     ```

3. Copy the certificate thumbprint for use in the `--thumbprint` option of `vic-machine` commands.

Obtain Certificate Thumbprints from Platform Services Controller

You can obtain a vCenter Server certificate thumbprint by logging into the Platform Services Controller for that vCenter Server instance.

1. Log in to the Platform Services Controller interface.

    - Embedded Platform Services Controller: https://*vcenter_server_address*/psc
    - Standalone Platform Services Controller: https://*psc_address*/psc

2. Select **Certificate Management** and enter a vCenter Single Sign-On password.

3. Select **Machine Certificates**, select a certificate, and click **Show Details**.
4. Copy the thumbprint for use in the `--thumbprint` option of `vic-machine` commands.

Deploy a VCH to an ESXi Host with No vCenter Server

This topic provides instructions for deploying a virtual container host (VCH) to an ESXi host that is not managed by vCenter Server. This is the most straightforward way to deploy a VCH, and is ideal for testing.

Prerequisites

- Deploy the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- In a Web browser, go to http://*vic_appliance_address*, scroll down to Infrastructure Deployment Tools, click the link to **download the vSphere Integrated Containers Engine bundle**, and unpack it on your working machine.
- Create or obtain an ESXi host with the following configuration:
  - One datastore
  - The VM Network is present
  - You can use a nested ESXi host for this example
- Verify that the ESXi host meets the requirements in Environment Prerequisites for VCH Deployment.
- Make sure that the correct firewall port is open on the ESXi host. For information about how to open ports on ESXi hosts, see Open the Required Ports on ESXi Hosts.
- Obtain the ESXi host certificate thumbprint. For information about how to obtain the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.
- Familiarize yourself with the vSphere Integrated Containers Engine binaries, as described in Contents of the vSphere Integrated Containers Engine Binaries.
- Familiarize yourself with the options of the `vic-machine create` command described in VCH Deployment Options.

Procedure

1. Open a terminal on the system on which you downloaded and unpacked the vSphere Integrated Containers Engine binary bundle.
2. Navigate to the directory that contains the `vic-machine` utility:
3. Run the `vic-machine create` command.

   In these examples, the password is wrapped in quotes because it contains `@` .

   - Linux OS:

     ```
     $ vic-machine-linux create
     --target esxi_host_address
     --user root
     --password 'esxi_host_p@ssword'
     --no-tlsverify
     --thumbprint esxi_certificate_thumbprint
     ```

   - Windows:

     ```
     $ vic-machine-windows create
     --target esxi_host_address
     --user root
     ```

```
--password "esxi_host_p@ssword"
--no-tlsverify
--thumbprint esxi_certificate_thumbprint
```

- Mac OS:

```
$ vic-machine-darwin create
--target esxi_host_address
--user root
--password 'esxi_host_p@ssword'
--no-tlsverify
--thumbprint esxi_certificate_thumbprint
```

The `vic-machine create` command in this example specifies the minimum information required to deploy a VCH to an ESXi host:

- The address of the ESXi host on which to deploy the VCH, in the `--target` option.
- The ESXi host `root` user and password in the `--user` and `--password` options.
- Disables the verification of clients that connect to this VCH by specifying the `--no-tlsverify` option.
- Specifies the thumbprint of the ESXi host certificate by specifying the `--thumbprint` option.

Because the ESXi host only has only one datastore and uses the VM Network network, `vic-machine create` automatically detects and uses those resources.

When deploying to an ESXi host, `vic-machine create` creates a standard virtual switch and a port group for use as the container bridge network, so you do not need to specify any network options if you do not have specific network requirements.

This example deploys a VCH with the default name `virtual-container-host` .

Result

At the end of a successful deployment, `vic-machine` displays information about the new VCH:

```
Initialization of appliance successful
VCH Admin Portal:
https://vch_address:2378
Published ports can be reached at:
vch_address
Docker environment variables:
DOCKER_HOST=vch_address:2376
Environment saved in virtual-container-host/virtual-container-host.env
Connect to docker:
docker -H vch_address:2376 --tls info
Installer completed successfully
```

What to Do Next

To test your VCH, see Verify the Deployment of a VCH.

For examples of commands to deploy a VCH in various other vSphere configurations, see Advanced Examples of Deploying a VCH.

Deploy a VCH to a Basic vCenter Server Cluster

This topic provides instructions for deploying a virtual container host (VCH) in a very basic vCenter Server environment. This basic deployment allows you to test vSphere Integrated Containers Engine with vCenter Server before attempting a more complex deployment that corresponds to your real vSphere environment.

The vCenter Server instance to which you deploy the VCH must match the specifications listed in the prerequisites.

Prerequisites

- Deploy the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- In a Web browser, go to http://*vic_appliance_address*, scroll down to Infrastructure Deployment Tools, click the link to **download the vSphere Integrated Containers Engine bundle**, and unpack it on your working machine.
- Create or obtain a vCenter Server instance with the following configuration:
  - One datacenter
  - One cluster with two ESXi hosts and DRS enabled. You can use nested ESXi hosts for this example.
  - A shared datastore, that is accessible by both of the ESXi hosts.
  - The VM Network is present
  - One distributed virtual switch with one port group named `vic-bridge`
- Verify that your vCenter Server instance and both of the ESXi hosts in the cluster meet the requirements in Environment Prerequisites for VCH Deployment.
- Make sure that the correct firewall ports are open on the ESXi hosts. For information about how to open ports on ESXi hosts, see Open the Required Ports on ESXi Hosts.
- Obtain the vCenter Server certificate thumbprint. For information about how to obtain the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.
- Familiarize yourself with the vSphere Integrated Containers Engine binaries, as described in Contents of the vSphere Integrated Containers Engine Binaries.
- Familiarize yourself with the options of the `vic-machine create` command described in VCH Deployment Options.

Procedure

1. Open a terminal on the system on which you downloaded and unpacked the vSphere Integrated Containers Engine binary bundle.
2. Navigate to the directory that contains the `vic-machine` utility:
3. Run the `vic-machine create` command.

   In these examples, the user name is wrapped in quotes because it contains `@` .

   - Linux OS:

     ```
     $ vic-machine-linux create
     --target vcenter_server_address
     --user 'Administrator@vsphere.local'
     --password vcenter_server_password
     --bridge-network vic-bridge
     --image-store shared_datastore_name
     ```

```
--no-tlsverify
--thumbprint vcenter_server_certificate_thumbprint
```

- Windows:

```
$ vic-machine-windows create
--target vcenter_server_address
--user "Administrator@vsphere.local"
--password vcenter_server_password
--bridge-network vic-bridge
--image-store shared_datastore_name
--no-tlsverify
--thumbprint vcenter_server_certificate_thumbprint
```

- Mac OS:

```
$ vic-machine-darwin create
--target vcenter_server_address
--user 'Administrator@vsphere.local'
--password vcenter_server_password
--bridge-network vic-bridge
--image-store shared_datastore_name
--no-tlsverify
--thumbprint vcenter_server_certificate_thumbprint
```

The `vic-machine create` command in this example specifies the minimum information required to deploy a VCH to vCenter Server:

- The address of the vCenter Server instance on which to deploy the VCH, in the `--target` option.
- The vCenter Single Sign-On user and password in the `--user` and `--password` options.
- The port group named `vic-bridge`, for use as the container bridge network.
- The name of the shared datastore to use as the image store, in which to store container images.
- Disables the verification of clients that connect to this VCH by specifying the `--no-tlsverify` option.
- Specifies the thumbprint of the vCenter Server host certificate by specifying the `--thumbprint` option.

Because the vCenter Server instance only has one datacenter and one cluster, and uses the VM Network network, `vic-machine create` automatically detects and uses these resources.

This example deploys a VCH with the default name `virtual-container-host`.

Result

At the end of a successful deployment, `vic-machine` displays information about the new VCH:

```
Initialization of appliance successful
VCH Admin Portal:
https://vch_address:2378
```

```
Published ports can be reached at:
vch_address
Docker environment variables:
DOCKER_HOST=vch_address:2376
Environment saved in virtual-container-host/virtual-container-host.env
Connect to docker:
docker -H vch_address:2376 --tls info
Installer completed successfully
```

What to Do Next

To test your VCH, see Verify the Deployment of a VCH.

For examples of commands to deploy a VCH in various other vSphere configurations, see Advanced Examples of Deploying a VCH.

Verify the Deployment of a VCH

After you have deployed a virtual container host (VCH), you can verify the deployment by connecting a Docker client to the VCH and running Docker operations. You can check the results in the vSphere Client or vSphere Web Client.

IMPORTANT: Do not use the vSphere Client or to perform operations on virtual container hosts or container VMs. Specifically, using the vSphere Client to power off, power on, or delete the VCH vApp or VCH endpoint VM can cause vSphere Integrated Containers Engine to not function correctly. Always use `vic-machine` to perform operations on VCHs. The vSphere Client does not allow you to delete container VMs, but do not use the vSphere Client to power container VMs on or off. Always use Docker commands to perform operations on containers.

Prerequisites

- You deployed a VCH in one of the following ways:

    - You followed the instructions in Deploy a Demo VCH to deploy a basic VCH with no security.
    - You followed the instructions in Deploy a VCH to an ESXi Host with No vCenter Server or Deploy a VCH to a Basic vCenter Server Cluster, specifying the `--no-tlsverify` option.
- You have installed a Docker client.
- If you deployed the VCH to vCenter Server, connect a vSphere Web Client to that vCenter Server instance.
- If you deployed the VCH to an ESXi host, connect a vSphere Client to that host.

Procedure

1. View the VCH appliance in the vSphere Web Client or vSphere Client.

    - vCenter Server: Go to **Hosts and Clusters** in the vSphere Web Client and select the cluster or host on which you deployed the VCH. You should see a vApp with the name that you set for the VCH.
    - ESXi host: Go to **Inventory** in the vSphere Client and select the host on which you deployed the VCH. You should see a resource pool with the name that you set for the VCH.

    The vApp or resource pool contains the VCH endpoint VM.

2. Run the `docker info` command to confirm that you can connect to the VCH.

    - Demo VCH:

      ```
      docker -H vch_address:2376 info
      ```

    - Deployment with `vic-machine create`:

      ```
      docker -H vch_address:2376 --tls info
      ```

      You should see confirmation that the Storage Driver is `vSphere Integrated Containers Backend Engine`.

3. Pull a Docker container image into the VCH, for example, the `BusyBox` container.

    - Demo VCH:

```
docker -H vch_address:2376 pull busybox
```

- Deployment with `vic-machine create` :

```
docker -H vch_address:2376 --tls pull busybox
```

4. View the container image files in the vSphere Web Client or vSphere Client.

   - vCenter Server: Go to **Storage**, right-click the datastore that you designated as the image store, and select **Browse Files**.
   - ESXi host: Click the **Summary** tab for the ESXi host, right-click the datastore that you designated as the image store, and select **Browse Datastore**.

   vSphere Integrated Containers Engine creates a folder that has the same name as the VCH, that contains a folder named `VIC` in which to store container image files.

5. Expand the `VIC` folder to navigate to the `images` folder. The `images` folder contains a folder for every container image that you pull into the VCH. The folders contain the container image files.

6. In your Docker client, run the Docker container that you pulled into the VCH.

   - Demo VCH:

   ```
   docker -H vch_address:2376 run --name test busybox
   ```

   - Deployment with `vic-machine create` :

   ```
   docker -H vch_address:2376 --tls run --name test busybox
   ```

7. View the container VMs in the vSphere Web Client or vSphere Client.

   - vCenter Server: Go to **Hosts and Clusters** and expand the VCH vApp.
   - ESXi host: Go to **Inventory** and expand the VCH resource pool.

   You should see a VM for every container that you run, including a VM named `test-container_id` .

8. View the container VM files in the vSphere Web Client or vSphere Client.

   - vCenter Server: Go to **Storage** and select the datastore that you designated as the image store.
   - ESXi host: Click the **Summary** tab for the ESXi host, right-click the datastore that you designated as the image store, and select **Browse Datastore**.

   At the top-level of the datastore, you should see a folder for every container that you run. The folders contain the container VM files.

VCH Deployment Options

The command line utility for vSphere Integrated Containers Engine, `vic-machine`, provides a `create` command with options that allow you to customize the deployment of virtual container hosts (VCHs) to correspond to your vSphere environment.

- vSphere Target Options
- Security Options
- Private Registry Options
- Datastore Options
- Networking Options
- General Deployment Options

To allow you to fine-tune the deployment of VCHs, `vic-machine create` provides Advanced Options.

- Advanced Resource Management Options
- Other Advanced Options

NOTE: Wrap any option arguments that include spaces or special characters in quotes. Use single quotes if you are using `vic-machine` on a Linux or Mac OS system and double quotes on a Windows system.

Option arguments that might require quotation marks include the following:

- User names and passwords in `--target`, or in `--user` and `--password`.
- Datacenter names in `--target`.
- VCH names in `--name`.
- Datastore names and paths in `--image-store` and `--volume-store`.
- Network and port group names in all networking options.
- Cluster and resource pool names in `--compute-resource`.
- Folder names in the paths for `--tls-cert-path`, `--tls-server-cert`, `--tls-server-key`, `--appliance-iso`, and `--bootstrap-iso`.

vSphere Target Options

The `create` command of the `vic-machine` utility requires you to provide information about where in your vSphere environment to deploy the VCH and the vCenter Server or ESXi user account to use.

You can set environment variables for the `--target`, `--user`, `--password`, and `--thumbprint` options. For information about setting environment variables, see Set Environment Variables for Key `vic-machine` Options.

`--target`

Short name: `-t`

The IPv4 address, fully qualified domain name (FQDN), or URL of the ESXi host or vCenter Server instance on which you are deploying a VCH. This option is always **mandatory**.

To facilitate IP address changes in your infrastructure, provide an FQDN whenever possible, rather than an IP address. If `vic-machine create` cannot resolve the FQDN, it fails with an error.

- If the target ESXi host is not managed by vCenter Server, provide the address of the ESXi host.

  ```
  --target esxi_host_address
  ```

- If the target ESXi host is managed by vCenter Server, or if you are deploying to a cluster, provide the address of vCenter Server.

  ```
  --target vcenter_server_address
  ```

- You can include the user name and password in the target URL. If you are deploying a VCH on vCenter Server, specify the username for an account that has the Administrator role on that vCenter Server instance.

  ```
  --target vcenter_or_esxi_username:password@vcenter_or_esxi_address
  ```

  If you do not include the user name in the target URL, you must specify the `user` option. If you do not specify the `password` option or include the password in the target URL, `vic-machine create` prompts you to enter the password.

  You can configure a VCH so that it uses a non-administrator account for post-deployment operations by specifying the `--ops-user` option.

- If you are deploying a VCH on a vCenter Server instance that includes more than one datacenter, include the datacenter name in the target URL. If you include an invalid datacenter name, `vic-machine create` fails and suggests the available datacenters that you can specify.

  ```
  --target vcenter_server_address/datacenter_name
  ```

`--user`

Short name: `-u`

The username for the ESXi host or vCenter Server instance on which you are deploying a VCH.

If you are deploying a VCH on vCenter Server, specify a username for an account that has the Administrator role on that vCenter Server instance.

```
--user esxi_or_vcenter_server_username
```

You can specify the username in the URL that you pass to `vic-machine create` in the `target` option, in which case the `user` option is not required.

You can configure a VCH so that it uses a non-administrator account for post-deployment operations by specifying the `--ops-user` option. If you do not specify `--ops-user`, VCHs use the vSphere administrator account that you specify in `--user` for general post-deployment operations.

`--password`

Short name: `-p`

The password for the user account on the vCenter Server on which you are deploying the VCH, or the password for the ESXi host if you are deploying directly to an ESXi host. If not specified, `vic-machine` prompts you to enter the password during deployment.

```
--password esxi_host_or_vcenter_server_password
```

You can also specify the username and password in the URL that you pass to `vic-machine create` in the `target` option, in which case the `password` option is not required.

`--compute-resource`

Short name: `-r`

The host, cluster, or resource pool in which to deploy the VCH.

If the vCenter Server instance on which you are deploying a VCH only includes a single instance of a standalone host or cluster, `vic-machine create` automatically detects and uses those resources. In this case, you do not need to specify a compute resource when you run `vic-machine create`. If you are deploying the VCH directly to an ESXi host and you do not use `--compute-resource` to specify a resource pool, `vic-machine create` automatically uses the default resource pool.

You specify the `--compute-resource` option in the following circumstances:

- A vCenter Server instance includes multiple instances of standalone hosts or clusters, or a mixture of standalone hosts and clusters.
- You want to deploy the VCH to a specific resource pool in your environment.

If you do not specify the `--compute-resource` option and multiple possible resources exist, or if you specify an invalid resource name, `vic-machine create` fails and suggests valid targets for `--compute-resource` in the failure message.

- To deploy to a specific resource pool on an ESXi host that is not managed by vCenter Server, specify the name of the resource pool:

```
--compute-resource  resource_pool_name
```

- To deploy to a vCenter Server instance that has multiple standalone hosts that are not part of a cluster, specify the IPv4 address or fully qualified domain name (FQDN) of the target host:

```
--compute-resource host_address
```

- To deploy to a vCenter Server with multiple clusters, specify the name of the target cluster:

```
--compute-resource cluster_name
```

- To deploy to a specific resource pool on a standalone host that is managed by vCenter Server, or to a specific

resource pool in a cluster, if the resource pool name is unique across all hosts and clusters, specify the name of the resource pool:

```
--compute-resource resource_pool_name
```

- To deploy to a specific resource pool on a standalone host that is managed by vCenter Server, if the resource pool name is not unique across all hosts, specify the IPv4 address or FQDN of the target host and name of the resource pool:

```
--compute-resource host_name/resource_pool_name
```

- To deploy to a specific resource pool in a cluster, if the resource pool name is not unique across all clusters, specify the full path to the resource pool:

```
--compute-resource cluster_name/Resources/resource_pool_name
```

`--thumbprint`

Short name: None

The thumbprint of the vCenter Server or ESXi host certificate. Specify this option if your vSphere environment uses untrusted, self-signed certificates. If your vSphere environment uses trusted certificates that are signed by a known Certificate Authority (CA), you do not need to specify the `--thumbprint` option.

If you run `vic-machine` without the specifying the `--thumbprint` option and the operation fails, the resulting error message includes the certificate thumbprint. Always verify that the thumbprint in the error message is valid before attempting to run the command again.

For information about how to obtain the certificate thumbprint either before running `vic-machine` or to verify a thumbprint from a `vic-machine` error message, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

You can bypass certificate thumbprint verification by specifying the `--force` option instead of `--thumbprint`.

CAUTION: It is not recommended to use `--force` to bypass thumbprint verification in production environments. Using `--force` in this way exposes VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials.

Use upper-case letters and colon delimitation in the thumbprint. Do not use space delimitation.

```
--thumbprint certificate_thumbprint
```

Security Options

The security options that `vic-machine create` provides allow for 3 broad categories of security:

- Restrict access to the Docker API with Auto-Generated Certificates
- Restrict access to the Docker API with Custom Certificates

- Do Not Restrict Access to the Docker API

You can also configure a VCH to use different user accounts for deployment and operation.

NOTE: Certain options in this section are exposed in the `vic-machine create` help if you run `vic-machine create --extended-help`, or `vic-machine create -x`.

Restrict Access to the Docker API with Auto-Generated Certificates

As a convenience, `vic-machine create` provides the option of generating a client certificate, server certificate, and certificate authority (CA) as appropriate when you deploy a VCH. The generated certificates are functional, but they do not allow for fine control over aspects such as expiration, intermediate certificate authorities, and so on.

vSphere Integrated Containers Engine authenticates Docker API clients by using client certificates. This configuration is commonly referred to as `tlsverify` in documentation about containers and Docker. A client certificate is accepted if it is signed by a CA that you provide by specifying one or more instances of the `--tls-ca` option. In the case of the certificates that `vic-machine create` generates, `vic-machine create` creates a CA and uses it to create and sign a single client certificate.

When using the Docker client, the client validates the server either by using CAs that are present in the root certificate bundle of the client system, or that are provided explicitly by using the `--tlscacert` option when running Docker commands. As a part of this validation, the server certificate must explicitly state at least one of the following, and must match the name or address that the client uses to access the server:

- The FQDN used to communicate with the server
- The IP address used to communicate with the server
- A wildcard domain that matches all of the FQDNs in a specific subdomain. For an example of a domain wildcard, see https://en.wikipedia.org/wiki/Wildcard_certificate#Example.

`--tls-cname`

Short name: None

The FQDN or IP address to embed in an auto-generated server certificate. Specify an FQDN, IP address, or a domain wildcard. If you provide a custom server certificate by using the `--tls-server-cert` option, you can use `--tls-cname` as a sanity check to ensure that the certificate is valid for the deployment.

If you do not specify `--tls-cname` but you do set a static address for the VCH on the client network interface, `vic-machine create` uses that address for the Common Name, with the same results as if you had specified `--tls-cname=x.x.x.x`. For information about setting a static IP address on the client network, see Specify a Static IP Address for the VCH Endpoint VM.

When you specify the `--tls-cname` option, `vic-machine create` performs the following actions during the deployment of the VCH:

- Checks for an existing certificate in either a folder that has the same name as the VCH that you are deploying, or in a location that you specify in the `--tls-cert-path` option. If a valid certificate exists that includes the same Common Name attribute as the one that you specify in `--tls-cname`, `vic-machine create` reuses it. Reusing certificates allows you to delete and recreate VCHs for which you have already distributed the certificates to container developers.

- If certificates are present in the certificate folder that include a different Common Name attribute to the one that you specify in `--tls-cname` , `vic-machine create` fails.
- If a certificate folder does not exist, `vic-machine create` creates a folder with the same name as the VCH, or creates a folder in the location that you specify in the `--tls-cert-path` option.
- If valid certificates do not already exist, `vic-machine create` creates the following trusted CA, server, and client certificate/key pairs in the certificate folder:
  - `ca.pem`
  - `ca-key.pem`
  - `cert.pem`
  - `key.pem`
  - `server-cert.pem`
  - `server-key.pem`
- Creates a browser-friendly PFX client certificate, `cert.pfx` , to use to authenticate connections to the VCH Admin portal for the VCH.

NOTE: The folder and file permissions for the generated certificate and key are readable only by the user who created them.

Running `vic-machine create` with the `--tls-cname` option also creates an environment file named `vch_name.env` , that contains Docker environment variables that container developers can use to configure their Docker client environment:

- Activates TLS client verification.

```
DOCKER_TLS_VERIFY=1
```

- The path to the client certificates.

```
DOCKER_CERT_PATH=path_to_certs
```

- The address of the VCH.

```
DOCKER_HOST=vch_address:2376
```

You must provide copies of the `cert.pem` and `key.pem` client certificate files and the environment file to container developers so that they can connect Docker clients to the VCH. If you deploy the VCH with the `--tls-cname` option, container developers must configure the client appropriately with one of the following options:

- By using the following `tlsverify` , `tlscert` , and `tlskey` Docker options, adding `tlscacert` if a custom CA was used to sign the server certificate.
- By setting `DOCKER_CERT_PATH=/path/to/client/cert.pem` and `DOCKER_TLS_VERIFY=1` .

```
 --tls-cname vch-name.example.org
```

```
 --tls-cname *.example.org
```

`--tls-cert-path`

Short name: none

By default `--tls-cert-path` is a folder in the current directory, that takes its name from the VCH name that you specify in the `--name` option. `vic-machine create` checks in `--tls-cert-path` for existing certificates with the standard names and uses those certificates if they are present:

- `server-cert.pem`
- `server-key.pem`
- `ca.pem`

If `vic-machine create` does not find existing certificates with the standard names in `--tls-cert-path`, or if you do not specify certificates directly by using the `--tls-server-cert`, `--tls-server-key`, and `--tls-ca` options, `vic-machine create` generates certificates. Generated certificates are saved in the `--tls-cert-path` folder with the standard names listed. `vic-machine create` additionally generates other certificates:

- `cert.pem` and `key.pem` for client certificates, if required.
- `ca-key.pem`, the private key for the certificate authority.

```
--tls-cert-path 'path_to_certificate_folder'
```

`--certificate-key-size`

Short name: `--ksz`

The size of the key for `vic-machine create` to use when it creates auto-generated trusted certificates. You can optionally use `--certificate-key-size` if you specify `--tls-cname`. If not specified, `vic-machine create` creates keys with default size of 2048 bits. It is not recommended to use key sizes of less than 2048 bits.

```
--certificate-key-size 3072
```

`--organization`

Short name: None

A list of identifiers to record in certificates generated by `vic-machine`. You can optionally use `--organization` if you specify `--tls-cname`. If not specified, `vic-machine create` uses the name of the VCH as the organization value.

NOTE: The `client-ip-address` is used for `CommonName` but not for `Organisation`.

```
--organization organization_name
```

Restrict Access to the Docker API with Custom Certificates

To exercise fine control over the certificates that VCHs use, obtain or generate custom certificates yourself before you deploy a VCH. Use the `--tls-server-key`, `--tls-server-cert`, and `--tls-ca` options to pass the custom certificates to `vic-machine create`.

IMPORTANT: PKCS#7 certificates do not work with `vic-machine` . For information about how to convert certificates to the correct format, see Converting Certificates for Use with vSphere Integrated Containers Engine.

`--tls-server-cert`

Short name: none

The path to a custom X.509 server certificate. This certificate identifies the VCH endpoint VM both to Docker clients and to browsers that connect to the VCH Admin portal.

- This certificate should have the following certificate usages:
  - `KeyEncipherment`
  - `DigitalSignature`
  - `KeyAgreement`
  - `ServerAuth`
- This option is mandatory if you use custom TLS certificates, rather than auto-generated certificates.
- Use this option in combination with the `--tls-server-key` option, that provides the path to the private key file for the custom certificate.
- Include the names of the certificate and key files in the paths.
- If you use trusted custom certificates, container developers run Docker commands with the `--tlsverify` , `--tlscacert` , `--tlscert` , and `--tlskey` options.

```
--tls-server-cert path_to_certificate_file/certificate_file_name.pem
--tls-server-key path_to_key_file/key_file_name.pem
```

`--tls-server-key`

Short name: none

The path to the private key file to use with a custom server certificate. This option is mandatory if you specify the `--tls-server-cert` option, that provides the path to a custom X.509 certificate file. Include the names of the certificate and key files in the paths.

IMPORTANT: The key must not be encrypted.

```
--tls-server-cert path_to_certificate_file/certificate_file_name.pem
--tls-server-key path_to_key_file/key_file_name.pem
```

`--tls-ca`

Short name: `--ca`

You can specify `--tls-ca` multiple times, to point `vic-machine create` to a file that contains the public portion of a CA. `vic-machine create` uses these CAs to validate client certificates that are offered as credentials for Docker API access. This does not need to be the same CA that you use to sign the server certificate.

```
--tls-ca path_to_ca_file
```

NOTE: The `--tls-ca` option appears in the extended help that you see by running `vic-machine-os create --extended-help` or `vic-machine-os create -x`.

Do Not Restrict Access to the Docker API

To deploy a VCH that does not restrict access to the Docker API, use the `--no-tlsverify` option. To completely disable TLS authentication, use the `--no-tls` option.

`--no-tlsverify`

Short name: `--kv`

The `--no-tlsverify` option prevents the use of CAs for client authentication. You still require a server certificate if you use `--no-tlsverify`. You can still supply a custom server certificate by using the `--tls-server-cert` and `--tls-server-key` options. If you do not use `--tls-server-cert` and `--tls-server-key` to supply a custom server certificate, `vic-machine create` generates a self-signed server certificate. If you specify `--no-tlsverify` there is no access control, however connections remain encrypted.

When you specify the `--no-tlsverify` option, `vic-machine create` performs the following actions during the deployment of the VCH.

- Generates a self-signed server certificate if you do not specify `--tls-server-cert` and `--tls-server-key`.
- Creates a folder with the same name as the VCH in the location in which you run `vic-machine create`.
- Creates an environment file named `vch_name.env` in that folder, that contains the `DOCKER_HOST=vch_address` environment variable, that you can provide to container developers to use to set up their Docker client environment.

If you deploy a VCH with the `--no-tlsverify` option, container developers run Docker commands with the `--tls` option, and the `DOCKER_TLS_VERIFY` environment variable must not be set. Note that setting `DOCKER_TLS_VERIFY` to 0 or `false` has no effect.

The `--no-tlsverify` option takes no arguments.

```
--no-tlsverify
```

`--no-tls`

Short name: `-k`

Disables TLS authentication of connections between the Docker client and the VCH. VCHs use neither client nor server certificates.

Set the `no-tls` option if you do not require TLS authentication between the VCH and the Docker client. Any Docker client can connect to the VCH if you disable TLS authentication and connections are not encrypted.

If you use the `no-tls` option, container developers connect Docker clients to the VCH via port 2375, instead of via port 2376.

```
--no-tls
```

Specify Different User Accounts for VCH Deployment and Operation

Because deploying a VCH requires greater levels of permissions than running a VCH, you can configure a VCH so that it uses different user accounts for deployment and for operation. In this way, you can limit the day-to-day operation of a VCH to an account that does not have full administrator permissions on the target vCenter Server.

`--ops-user`

Short name: None

A vSphere user account with which the VCH runs after deployment. If not specified, the VCH runs with the vSphere Administrator credentials with which you deploy the VCH, that you specify in either `--target` or `--user`.

```
--ops-user user_name
```

The user account that you specify in `--ops-user` must exist before you deploy the VCH. For information about the permissions that the `--ops-user` account requires, see Use Different User Accounts for VCH Deployment and Operation.

`--ops-password`

Short name: None

The password or token for the operations user that you specify in `--ops-user`. If not specified, `vic-machine create` prompts you to enter the password for the `--ops-user` account.

```
--ops-password password
```

Private Registry Options

If you use vSphere Integrated Containers Registry, or if container developers need to access Docker images that are stored in other private registry servers, you must configure VCHs to allow them to connect to these private registry servers when you deploy the VCHs. VCHs can connect to both secure and insecure private registry servers. You can also configure VCHs so that they can only access images from a whitelist of approved registries.

`--registry-ca`

Short name: `--rc`

The path to a CA certificate that can validate the server certificate of a private registry. You can specify `--registry-ca` multiple times to specify multiple CA certificates for different registries. This allows a VCH to connect to multiple registries.

The use of registry certificates is independent of the Docker client security options that you specify. For example, it is possible to use the `--no-tls` option to disable TLS authentication between Docker clients and the VCH, and to use the `--registry-ca` option to enable TLS authentication between the VCH and a private registry.

You must use this option to allow a VCH to connect to vSphere Integrated Containers Registry. For information about how to obtain the CA certificate from vSphere Integrated Containers Registry, see Deploy a VCH for Use with vSphere Integrated Containers Registry.

```
--registry-ca path_to_ca_cert_1
--registry-ca path_to_ca_cert_2
```

NOTE: The `--registry-ca` option appears in the extended help that you see by running `vic-machine-os create --extended-help` or `vic-machine-os create -x`.

--insecure-registry

Short name: `--dir`

If you set the `--insecure-registry` option, the VCH does not verify the certificate of that registry when it pulls images. Insecure private registries are not recommended in production environments.

If you authorize a VCH to connect to an insecure private registry server, the VCH attempts to access the registry server via HTTP if access via HTTPS fails. VCHs always use HTTPS when connecting to registry servers for which you have not authorized insecure access.

NOTE: You cannot configure VCHs to connect to vSphere Integrated Containers Registry instances as insecure registries. Connections to vSphere Integrated Containers Registry always require HTTPS and a certificate.

You can specify `--insecure-registry` multiple times if multiple insecure registries are permitted. If the registry server listens on a specific port, add the port number to the URL

```
--insecure-registry registry_URL_1
--insecure-registry registry_URL_2:port_number
```

--whitelist-registry

You can restrict the registries to which a VCH allows access by setting the `--whitelist-registry` option. You can specify `--whitelist-registry` multiple times to allow access to multiple registries. If you specify `--whitelist-registry` at least once, the VCH runs in whitelist mode. In whitelist mode, users can only access those registries that you have specified in the `--whitelist-registry` option. Users cannot access any registries that are not in the whitelist, even if they are public registries, such as Docker Hub.

You can specify whitelisted registries in the following formats:

- IP addresses or FQDN to identify individual registry instances. During deployment, `vic-machine` validates the IP address of the registry.
- CIDR formatted ranges, for example, 192.168.1.1/24. If you specify a CIDR range, the VCH adds to the whitelist any IP addresses within that subnet. Note that `vic-machine` does not validate CIDR defined ranges during deployment.
- Wildcard domains, for example, . *.company.com. If you specify a wildcard domain, the VCH adds to the whitelist any IP addresses or FQDNs that it can validate against that domain. A numeric IP address causes VCHs to

perform a reverse DNS lookup to validate against that wild card domain. Note that `vic-machine` does not validate wildcard domains during deployment.

You use `--whitelist-registry` in combination with the `--registry-ca` and `--insecure-registry` options. You can configure a VCH so that it includes both secure and insecure registries in its whitelist.

Whitelisting Secure Registries

VCHs include a base set of well-known certificates from public CAs. If a registry requires a certificate to authenticate access, and if that registry does not use one of the CAs in the VCH, you must provide the CA certificate for that registry in the `--registry-ca` option. You must also specify that registry in the `--whitelist-registry` option if the VCH is running in whitelist mode.

- If you provide a certificate in the `--registry-ca` option but you do not specify that registry in the `--whitelist-registry` option, the VCH does not allow access to that registry.
- If you specify a registry in the `--whitelist-registry` option, but you do not provide a certificate in `--registry-ca` and the registry's CA is not in the set of well-known certificates in the VCH, the VCH does not allow access to that registry.

```
--whitelist-registry registry_address
--registry-ca path_to_ca_cert_1
```

Whitelisting Insecure Registries

You can add registries that you designate as insecure registries to the whitelist by specifying both of the `--insecure-registry` and `--whitelist-registry` options.

- If you specify a registry in the `--whitelist-registry` option, but you do not specify that registry in `--insecure-registry`, the VCH attempts to verify the registry by using certificates. If it does not find a certificate, the VCH does not allow access to that registry.
- If you specify a registry in the `--insecure-registry` option but you do not specify this registry in `--whitelist-registry`, `vic-machine` adds the registry to the whitelist only if at least one other registry is specified in `--whitelist-registry`.

```
--whitelist-registry registry_address
--insecure-registry registry_address
```

Datastore Options

The `vic-machine` utility allows you to specify the datastore in which to store container image files, container VM files, and the files for the VCH. You can also specify datastores in which to create container volumes.

- vSphere Integrated Containers Engine fully supports VMware vSAN datastores.
- vSphere Integrated Containers Engine supports all alphanumeric characters, hyphens, and underscores in datastore paths and datastore names, but no other special characters.

- If you specify different datastores in the different datastore options, and if no single host in a cluster can access all of those datastores, `vic-machine create` fails with an error.

  ```
  No single host can access all of the requested datastores.
  Installation cannot continue.
  ```

- If you specify different datastores in the different datastore options, and if only one host in a cluster can access all of them, `vic-machine create` succeeds with a warning.

  ```
  Only one host can access all of the image/container/volume datastores.
  This may be a point of contention/performance degradation and HA/DRS
  may not work as intended.
  ```

- VCHs do not support datastore name changes. If a datastore changes name after you have deployed a VCH that uses that datastore, that VCH will no longer function.

`--image-store`

Short name: `-i`

The datastore in which to store container image files, container VM files, and the files for the VCH. The `--image-store` option is **mandatory** if there is more than one datastore in your vSphere environment. If there is only one datastore in your vSphere environment, the `--image-store` option is not required.

If you do not specify the `--image-store` option and multiple possible datastores exist, or if you specify an invalid datastore name, `vic-machine create` fails and suggests valid datastores in the failure message.

If you are deploying the VCH to a vCenter Server cluster, the datastore that you designate in the `image-store` option must be shared by at least two ESXi hosts in the cluster. Using non-shared datastores is possible, but limits the use of vSphere features such as vSphere vMotion® and VMware vSphere Distributed Resource Scheduler™ (DRS).

To specify a whole datastore as the image store, specify the datastore name in the `--image-store` option:

```
--image-store datastore_name
```

If you designate a whole datastore as the image store, `vic-machine` creates the following set of folders in the target datastore:

- `datastore_name/VIC/vch_uuid/images`, in which to store all of the container images that you pull into the VCH.
- `datastore_name/vch_name`, that contains the VM files for the VCH.
- `datastore_name/vch_name/kvstores`, a key-value store folder for the VCH.

You can specify a datastore folder to use as the image store by specifying a path in the `--image-store` option</code>:

```
--image-store datastore_name/path
```

If the folder that you specify in `/path` does not already exist, `vic-machine create` creates it.

If you designate a datastore folder as the image store, `vic-machine` creates the following set of folders in the target datastore:

- `datastore_name/path/VIC/vcu_uuid/images` , in which to store all of the container images that you pull into the VCH.
- `datastore_name/vch_name` , that contains the VM files for the VCH. This is the same as if you specified a datastore as the image store.
- `datastore_name/vch_name/kvstores` , a key-value store folder for the VCH. This is the same as if you specified a datastore as the image store.

By specifying the path to a datastore folder in the `--image-store` option, you can designate the same datastore folder as the image store for multiple VCHs. In this way, `vic-machine create` creates only one `VIC` folder in the datastore, at the path that you specify. The `VIC` folder contains one `vch_uuid/images` folder for each VCH that you deploy. By creating one `vch_uuid/images` folder for each VCH, vSphere Integrated Containers Engine limits the potential for conflicts of image use between VCHs, even if you share the same image store folder between multiple hosts.

When container developers create containers, vSphere Integrated Containers Engine stores the files for container VMs at the top level of the image store, in folders that have the same name as the containers.

--volume-store

Short name: `--vs`

The datastore in which to create volumes when container developers use the `docker volume create` command. You can specify either a datastore that is backed by vSphere or an NFS share point as the volume store.

If you are deploying the VCH to a vCenter Server cluster, vSphere datastores that you designate in the `volume-store` option should be shared by at least two ESXi hosts in the cluster. Using non-shared datastores is possible and `vic-machine create` succeeds, but it issues a warning that this configuration limits the use of vSphere features such as vSphere vMotion and DRS.

If you use NFS volume stores, container developers can share the data in those volumes between containers by attaching the same volume to multiple containers. For example, you can use shared NFS volume stores to share configuration information between containers, or to allow containers to access the data of another container. To use shared NFS volume stores, it is recommended that the NFS share points that you designate as the volume stores be directly accessible by the network that you use as the container network. For information about container networks, see the description of the `--container-network` option.

The label that you specify is the volume store name that Docker uses. For example, the volume store label appears in the information for a VCH when container developers run `docker info` . Container developers specify the volume store label in the `docker volume create --opt VolumeStore=volume_store_label` option when they create a volume.

IMPORTANT: The volume store label must be unique.

If you specify an invalid vSphere datastore name or an invalid NFS share point URL, `vic-machine create` fails and suggests valid datastores.

IMPORTANT If you do not specify the `volume-store` option, no volume store is created and container developers cannot create containers with volumes. You can add volume stores to a VCH after deployment by running `vic-machine configure --volume-store` . For information about adding volume stores after deployment, see Add Volume Stores in Configure Virtual Container Hosts.

- To specify a vSphere datastore, provide the datastore name and the volume store label.

  ```
  --volume-store datastore_name:volume_store_label
  ```

  You can optionally use the `ds://` prefix to specify a datastore that is backed by vSphere.

  ```
  --volume-store ds://datastore_name:volume_store_label
  ```

  If you specify a vSphere datastore without specifying a path to a specific datastore folder, `vic-machine create` creates a folder named `VIC/volumes` at the top level of the target datastore. Any volumes that container developers create will appear in the `VIC/volumes` folder.

- If you specify a vSphere datastore and a datastore path, `vic-machine create` creates a folder named `volumes` in the location that you specify in the datastore path. If the folders that you specify in the path do not already exist on the datastore, `vic-machine create` creates the appropriate folder structure. Any volumes that container developers create will appear in the `path/volumes` folder.

  ```
  --volume-store datastore_name/datastore_path:volume_store_label
  ```

  The `vic-machine create` command creates the `volumes` folder independently from the folders for VCH files so that you can share volume stores between VCHs. If you delete a VCH, any volumes that the VCH managed will remain available in the volume store unless you specify the `--force` option when you delete the VCH. You can then assign an existing volume store that already contains data to a newly created VCH.

  IMPORTANT: If multiple VCHs will use the same datastore for their volume stores, specify a different datastore folder for each VCH. Do not designate the same datastore folder as the volume store for multiple VCHs.

- To specify an NFS share point as a volume store, use the `nfs://` prefix and the path to a shared mount point.

  IMPORTANT: When container developers run `docker info` or `docker volume ls` against a VCH, there is currently no indication whether a volume store is backed by vSphere or by an NFS share point. Consequently, you should include an indication that a volume store is an NFS share point in the volume store label.

  ```
  nfs://datastore_name/path_to_share_point:nfs_volume_store_label
  ```

- You can specify the `volume-store` option multiple times, to create multiple volume stores for the VCH.

- You can also specify the URL, UID, GID, and access protocol of a shared NFS mount point when you specify an NFS share point.

  ```
  --volume-store nfs://datastore_address/path_to_share_point?
  uid=1234&gid=5678&proto=tcp:nfs_volume_store_label
  ```

- If you do not specify a UID and GID, vSphere Integrated Containers Engine uses the `anon` UID and GID when creating and interacting with the volume store. The `anon` UID and GID is 1000.

- You cannot specify the root folder of an NFS server as a volume store.

- If you only require one volume store, set the volume store label to `default` . If you set the volume store label to `default` , container developers do not need to specify the `--opt VolumeStore=volume_store_label` option when they run `docker volume create` .

  NOTE: If container developers intend to use `docker create -v` to create containers that are attached to anonymous or named volumes, you must create a volume store with a label of `default` .

  ```
    --volume-store datastore_name:default
  ```

  ```
    --volume-store nfs://datastore_name/path_to_share_point:default
  ```

- You can specify the `volume-store` option multiple times, to create multiple volume stores for the VCH. You can add a mixture of vSphere datastores and NFS share points to a VCH.

  ```
    --volume-store datastore_name/path:volume_store_label_1
    --volume-store datastore_name/path:volume_store_label_2
    --volume-store nfs://datastore_name/path_to_share_point:nfs_volume_store_label
  ```

General Deployment Options

The `vic-machine` utility provides options to customize the VCH.

`--name`

Short name: `-n`

A name for the VCH. If not specified, `vic-machine` sets the name of the VCH to `virtual-container-host` . If a VCH of the same name exists on the ESXi host or in the vCenter Server inventory, or if a folder of the same name exists in the target datastore, `vic-machine create` creates a folder named `vch_name_1` . If the name that you provide contains unsupported characters, `vic-machine create` fails with an error.

```
  --name vch_name
```

`--memory`

Short name: `--mem`

Limit the amount of memory that is available for use by the VCH vApp in vCenter Server, or for the VCH resource pool on an ESXi host. This limit also applies to the container VMs that run in the VCH vApp or resource pool. Specify the memory limit value in MB. If not specified, `vic-machine create` sets the limit to 0 (unlimited).

```
  --memory 1024
```

`--cpu`

Short name: None

Limit the amount of CPU capacity that is available for use by the VCH vApp in vCenter Server, or for the VCH resource pool on an ESXi host. This limit also applies to the container VMs that run in the VCH vApp or resource pool. Specify the CPU limit value in MHz. If not specified, `vic-machine create` sets the limit to 0 (unlimited).

```
--cpu 1024
```

`--force`

Short name: `-f`

Forces `vic-machine create` to ignore warnings and non-fatal errors and continue with the deployment of a VCH. Errors such as an incorrect compute resource still cause the deployment to fail.

If your vSphere environment uses untrusted, self-signed certificates, you can bypass certificate thumbprint verification by specifying the `--force` option instead of `--thumbprint`.

CAUTION: It is not recommended to use `--force` to bypass thumbprint verification in production environments. Using `--force` in this way exposes VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials.

```
--force
```

`--timeout`

Short name: none

The timeout period for uploading the vSphere Integrated Containers Engine files and ISOs to the ESXi host, and for powering on the VCH. Specify a value in the format `XmYs` if the default timeout of 3m0s is insufficient.

```
--timeout 5m0s
```

Advanced Options

The options in this section are exposed in the `vic-machine create` help if you run `vic-machine create --extended-help`, or `vic-machine create -x`.

Specify a Static IP Address for the VCH Endpoint VM

Advanced Resource Management Options

You can set limits on the memory and CPU shares and reservations on the VCH. For information about memory and CPU shares and reservations, see Allocate Memory Resources, and Allocate CPU Resources in the vSphere documentation.

`--memory-reservation`

Short name: `--memr`

Reserve a quantity of memory for use by the VCH vApp in vCenter Server, or for the VCH resource pool on an ESXi host. This limit also applies to the container VMs that run in the VCH vApp or resource pool. Specify the memory reservation value in MB. If not specified, `vic-machine create` sets the reservation to 1.

```
--memory-reservation 1024
```

`--memory-shares`

Short name: `--mems`

Set memory shares on the VCH vApp in vCenter Server, or on the VCH resource pool on an ESXi host. This limit also applies to the container VMs that run in the VCH vApp or resource pool. Specify the share value as a level or a number, for example `high`, `normal`, `low`, or `163840`. If not specified, `vic-machine create` sets the share to `normal`.

```
--memory-shares low
```

`--cpu-reservation`

Short name: `--cpur`

Reserve a quantity of CPU capacity for use by the VCH vApp in vCenter Server, or for the VCH resource pool on an ESXi host. This limit also applies to the container VMs that run in the VCH vApp or resource pool. Specify the CPU reservation value in MHz. If not specified, `vic-machine create` sets the reservation to 1.

```
--cpu-reservation 1024
```

`--cpu-shares`

Short name: `--cpus`

Set CPU shares on the VCH vApp in vCenter Server, or on the VCH resource pool on an ESXi host. This limit also applies to the container VMs that run in the VCH vApp or resource pool. Specify the share value as a level or a number, for example `high`, `normal`, `low`, or `163840`. If not specified, `vic-machine create` sets the share to `normal`.

```
--cpu-shares low
```

`--endpoint-cpu`

Short name: none

The number of virtual CPUs for the VCH endpoint VM. The default is 1. Set this option to increase the number of CPUs in the VCH endpoint VM.

NOTE Always use the `--cpu` option instead of the `--endpoint-cpu` option to increase the overall CPU capacity of the VCH vApp, rather than increasing the number of CPUs on the VCH endpoint VM. The `--endpoint-cpu` option is mainly intended for use by VMware Support.

```
--endpoint-cpu number_of_CPUs
```

`--endpoint-memory`

Short name: none

The amount of memory for the VCH endpoint VM. The default is 2048MB. Set this option to increase the amount of memory in the VCH endpoint VM if the VCH will pull large container images.

NOTE With the exception of VCHs that pull large container images, always use the `--memory` option instead of the `--endpoint-memory` option to increase the overall amount of memory for the VCH vApp, rather than on the VCH endpoint VM. Use `docker create -m` to set the memory on container VMs. The `--endpoint-memory` option is mainly intended for use by VMware Support.

```
--endpoint-memory amount_of_memory
```

Other Advanced Options

`--base-image-size`

Short name: None

The size of the base image from which to create other images. You should not normally need to use this option. Specify the size in `GB` or `MB`. The default size is 8GB. Images are thin-provisioned, so they do not usually consume 8GB of space.

```
--base-image-size 4GB
```

`--container-store`

Short name: `--cs`

The `container-store` option is not enabled. Container VM files are stored in the datastore that you designate as the image store.

`--appliance-iso`

Short name: `--ai`

The path to the ISO image from which the VCH appliance boots. Set this option if you have moved the `appliance.iso` file to a folder that is not the folder that contains the `vic-machine` binary or is not the folder from which you are running `vic-machine`. Include the name of the ISO file in the path.

NOTE: Do not use the `--appliance-iso` option to point `vic-machine` to an `--appliance-iso` file that is of a different version to the version of `vic-machine` that you are running.

```
--appliance-iso path_to_ISO_file/appliance.iso
```

--bootstrap-iso

Short name: `--bi`

The path to the ISO image from which to boot container VMs. Set this option if you have moved the `bootstrap.iso` file to a folder that is not the folder that contains the `vic-machine` binary or is not the folder from which you are running `vic-machine`. Include the name of the ISO file in the path.

NOTE: Do not use the `--bootstrap-iso` option to point `vic-machine` to a `--bootstrap-iso` file that is of a different version to the version of `vic-machine` that you are running.

```
--bootstrap-iso path_to_ISO_file/bootstrap.iso
```

--use-rp

Short name: none

Deploy the VCH appliance to a resource pool on vCenter Server rather than to a vApp. If you specify this option, `vic-machine create` creates a resource pool with the same name as the VCH.

NOTE: If you specify both of the `--ops-user` and `--use-rp` options when you create a VCH, you must specify an additional permission when you create the roles for the operations user. For information about operations user roles and permissions, see Use Different User Accounts for VCH Deployment and Operation.

```
--use-rp
```

--debug

Short name: `-v`

Deploy the VCH with more verbose levels of logging, and optionally modify the behavior of `vic-machine` for troubleshooting purposes. Specifying the `--debug` option increases the verbosity of the logging for all aspects of VCH operation, not just deployment. For example, by setting the `--debug` option, you increase the verbosity of the logging for VCH initialization, VCH services, container VM initialization, and so on. If not specified, the `--debug` value is set to 0 and verbose logging is disabled.

NOTE: Do not confuse the `vic-machine create --debug` option with the `vic-machine debug` command, that enables access to the VCH endpoint VM. For information about `vic-machine debug`, see Debugging the VCH.

When you specify `vic-machine create --debug`, you set a debugging level of 1, 2, or 3. Setting `--debug` to 2 or 3 changes the behavior of `vic-machine create` as well as increasing the level of verbosity of the logs:

- `--debug 1` Provides verbosity in the logs, with no other changes to `vic-machine` behavior. This is the default setting.
- `--debug 2` Exposes servers on more interfaces, launches `pprof` in container VMs.
- `--debug 3` Disables recovery logic and logs sensitive data. Disables the restart of failed components and prevents container VMs from shutting down. Logs environment details for user application, and collects application output in the log bundle. This is the maximum supported debugging level.

Additionally, deploying a VCH with a `--debug 3` enables SSH access to the VCH endpoint VM console by default, with a root password of `password`, without requiring you to run the `vic-machine debug` command. This functionality enables you to perform targeted interactive diagnostics in environments in which a VCH endpoint VM failure occurs consistently and in a fashion that prevents `vic-machine debug` from functioning.

IMPORTANT: There is no provision for persistently changing the default root password. Only use this configuration for debugging in a secured environment.

--syslog-address

Short name: None

Configure a VCH so that it sends the logs in the `/var/log/vic` folder on the VCH endpoint VM to a syslog endpoint that is not located in the VCH. The VCH also sends container logs to the same syslog endpoint.

You specify the address and port of the syslog endpoint in the `--syslog-address` option. You must also specify whether the transport protocol is UDP or TCP. If you do not specify a port, the default port is 514.

```
--syslog-address udp://syslog_host_address[:port]
```

```
--syslog-address tcp://syslog_host_address[:port]
```

--container-name-convention

Short name: `--cnc`

Enforce a naming convention for container VMs, that applies a prefix to the names of all container VMs that run in a VCH. Applying a naming convention to container VMs facilitates organizational requirements such as chargeback. The container naming convention applies to the display name of the container VM that appears in the vSphere Client, not to the container name that Docker uses.

You specify the container naming convention by providing a prefix to apply to container names, and adding `-{name}` or `-{id}` to specify whether to use the container name or the container ID for the second part of the container VM display name. If you specify `-{name}`, the container VM display names use either the name that Docker generates, or a name that the container developer specifies in `docker run --name` when they run the container.

```
--container-name-convention cVM_name_prefix-{name}
```

```
--container-name-convention cVM_name_prefix-{id}
```

Virtual Container Host Networking

You can configure networks on a virtual container host (VCH) that tie your Docker development environment into the vSphere infrastructure. You define which networks are available to a VCH when you use `vic-machine create` to deploy the VCH.

- High-Level View of VCH Networking
- Understanding Docker and VCH Networking
- VCH Networks
- Networking Limitations
- Host Firewall Configuration

High-Level View of VCH Networking

The image below shows how VCH networks connect to your vSphere environment, to vSphere Integrated Containers Registry and Management Portal, and to public registries, such as Docker Hub.
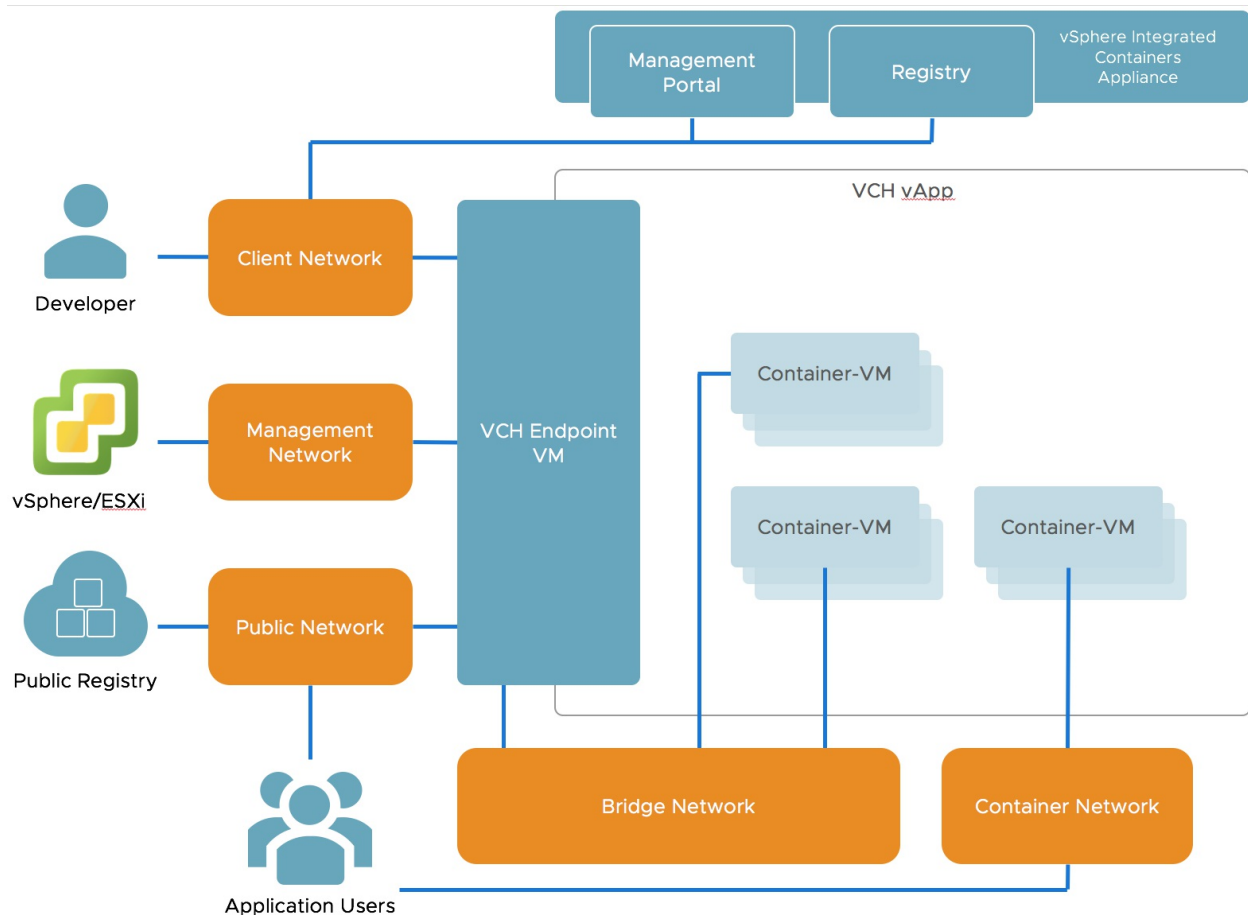
Understanding Docker and VCH Networking

To understand how you can configure networks on VCHs, you first must understand how networking works in Docker.

For an overview of Docker networking in general, and an overview of networking with vSphere Integrated Containers in particular, watch the Docker Networking Options and vSphere Integrated Containers Networking Overview videos on the VMware Cloud-Native YouTube Channel:

See also Docker container networking in the Docker documentation.

VCH Networks

You can direct traffic between containers, the VCH, the external Internet, and your vSphere environment to different networks. Each network that a VCH uses is a distributed port group or an NSX logical switch on either a vCenter Server instance or an ESXi host. You must create the port groups or logical switches in vSphere before you deploy a VCH. For information about how to create a distributed virtual switch and port group, see the section on Networking Requirements in Environment Prerequisites for VCH Deployment.

- **Public Network:** The network that container VMs and VCHs use to access the Internet. For information about VCH public networks, see Configure the Public Network.
- **Bridge Networks**: In Docker terminology, the VCH bridge network corresponds to the default bridge network on a Docker host. You can also create additional bridge networks, that correspond to Docker user-defined networks. For information about VCH bridge networks, see Configure Bridge Networks.
- **Client Network**: You can isolate traffic between Docker clients and the VCH from traffic on the public network by specifying a dedicated network for client connections. For information about VCH client networks, see Configure the Client Network.
- **Management Network**: You can also isolate the traffic between the VCH and vCenter Server and ESXi hosts from traffic on the public network by specifying a dedicated management network. For information about VCH

management networks, see Configure the Management Network.

- **Container Networks**: User-defined networks that you can use to connect container VMs directly to a routable network. Container networks allow vSphere administrators to make vSphere networks directly available to containers. Container networks are specific to vSphere Integrated Containers and have no equivalent in regular Docker. For information about container networks, see Configure Container Networks.

You can configure static IP addresses for the VCH on the different networks, and configure VCHs to use proxy servers. For more information about static IP addresses and proxy servers, see Specify a Static IP Address for the VCH Endpoint VM and Configure VCHs to Use Proxy Servers.

Networking Limitations

A VCH supports a maximum of 3 distinct network interfaces. The bridge network requires its own port group, so at least two of the public, client, and management networks must share a network interface and therefore a port group. Container networks do not go through the VCH, so they are not subject to this limitation. This limitation will be removed in a future release

Host Firewall Configuration

When you specify different network interfaces for the different types of traffic, `vic-machine create` checks that the firewalls on the ESXi hosts allow connections to port 2377 from those networks. If access to port 2377 on one or more ESXi hosts is subject to IP address restrictions, and if those restrictions block access to the network interfaces that you specify, `vic-machine create` fails with a firewall configuration error:

```
Firewall configuration incorrect due to allowed IP restrictions on hosts:
"/ha-datacenter/host/localhost.localdomain/localhost.localdomain"
Firewall must permit dst 2377/tcp outbound to the VCH management interface
```

For information about how to open port 2377, see Open the Required Ports on ESXi Hosts.

Configure the Public Network

The public network is the network that container VMs and the virtual container host (VCH) use to connect to the Internet. VCHs use the public network to pull container images from public registries, for example from Docker Hub. Container VMs that use port mapping expose network services on the public network. In Docker terminology, the public network corresponds to the `eth0` network on a Docker host.

- `vic-machine` Option
- Example `vic-machine` Command

`vic-machine` Option

You designate a specific network for traffic from container VMs and the VCH to the Internet by specifying the `vic-machine create --public-network` option when you deploy the VCH.

`--public-network`

Short name: `--pn`

A port group that container VMs and VCHs use to connect to the Internet. Ports that containers that are connected to the default bridge network expose with `docker create -p` are made available on the public interface of the VCH endpoint VM via network address translation (NAT), so that containers can publish network services.

NOTE: vSphere Integrated Containers adds a new capability to Docker that allows you to directly map containers to a network by using the `--container-network` option. This is the recommended way to deploy container services with vSphere Integrated Containers. For more information, see Configure Container Networks.

Usage:

```
--public-network port_group_name
```

If you do not specify this option, containers use the VM Network for public network traffic. If you specify an invalid port group name, `vic-machine create` fails and suggests valid port groups.

Example `vic-machine` Command

This example deploys a VCH with the following configuration:

- Specifies the target vCenter Server instance, the vCenter Server user name, password, datacenter and cluster, an image store, a port group for the bridge network, a name for the VCH, and the thumbprint of the vCenter Server certificate.
- Secures connections to the Docker API with an automatically generated server certificate, without client certificate verification, by setting `--no-tlsverify`.
- Directs public network traffic to an existing port group named `network 1`.

- Does not specify either of the `--management-network` or `--client-network` options. Consequently, management and client traffic also routes over `network 1` because those networks default to the public network setting if they are not set.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--public-network 'network 1'
--name vch1
--thumbprint certificate_thumbprint
--no-tlsverify
```

Configure Bridge Networks

Bridge networks are the network or networks that container VMs use to communicate with each other. Every virtual container host (VCH) must have a unique bridge network. In Docker terminology, the bridge network corresponds to the default bridge network on a Docker host. You can also create additional bridge networks, that correspond to Docker user-defined networks.

IMPORTANT: Do not use the bridge network for any other VM workloads, or as a bridge for more than one VCH.

Container application developers can use `docker network create` to create additional, user-defined bridge networks when they run containers. VCHs create additional user-defined bridge networks by using IP address segregation within an address range that you specify, so user-defined bridge networks do not require new port groups.

- `vic-machine` Options
- Example `vic-machine` Command

`vic-machine` Options

You designate the bridge network by specifying the `vic-machine create --bridge-network` option. You can also provide a range of IP addresses for use by additional user-defined bridge networks by specifying the `bridge-network-range` option. For information about how to set bridge network ranges, see the section on the `--bridge-network-range` option.

`--bridge-network`

Short name: `-b`

A port group that container VMs use to communicate with each other.

The `--bridge-network` option is **mandatory** if you are deploying a VCH to vCenter Server. Before you run `vic-machine create`, you must create a distributed virtual switch and a port group for the bridge network. You must add the target ESXi host or hosts to the distributed virtual switch, and assign a VLAN ID to the port group, to ensure that the bridge network is isolated. For information about how to create a distributed virtual switch and port group, see the section on Networking Requirements in Environment Prerequisites for VCH Deployment.

The `--bridge-network` option is **optional** if you are deploying a VCH to an ESXi host that is not managed by vCenter Server. In this case, if you do not specify `--bridge-network`, `vic-machine` creates a virtual switch and a port group that each have the same name as the VCH. You can optionally specify this option to assign an existing port group for use as the bridge network for container VMs. You can also optionally specify this option to create a new virtual switch and port group that have a different name to the VCH.

IMPORTANT

- Do not specify the same port group as the bridge network for multiple VCHs. Sharing a port group between VCHs might result in multiple container VMs being assigned the same IP address.
- Do not use the bridge network port group as the target for any of the other `vic-machine create` networking options.

- If you intend to use the `--ops-user` option to use different user accounts for deployment and operation of the VCH, you must place the bridge network port group in a network folder that has the `Read-Only` role with propagation enabled. For more information about the requirements when using `--ops-user`, see Use Different User Accounts for VCH Deployment and Operation.

Usage:

```
--bridge-network port_group_name
```

If you do not specify `--bridge-network` or if you specify an invalid port group name, `vic-machine create` fails and suggests valid port groups.

`--bridge-network-range`

Short name: `--bnr`

A range of IP addresses that additional bridge networks can use when container application developers use `docker network create` to create new bridge networks.

When you specify the bridge network IP range, you specify the IP range as a CIDR. The smallest subnet that you can specify is /16.

Usage:

```
--bridge-network-range network_address/subnet
```

If you do not specify the `bridge-network-range` option, the IP range for bridge networks is 172.16.0.0/12. If you specify an invalid value for `--bridge-network-range`, `vic-machine create` fails with an error.

Example `vic-machine` Command

This example deploys a VCH with the following configuration:

- Specifies the target vCenter Server instance, the vCenter Server user name, password, datacenter and cluster, an image store, a name for the VCH, and the thumbprint of the vCenter Server certificate.
- Secures connections to the Docker API with an automatically generated server certificate, without client certificate verification, by setting `--no-tlsverify`.
- Designates an existing port group named `vch1-bridge` as the bridge network.
- Designates IP addresses in the range 192.168.100.0/16 for use by user-defined bridge networks.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--bridge-network-range 192.168.100.0/16
```

```
--name vch1
--thumbprint certificate_thumbprint
--no-tlsverify
```

Configure the Client Network

The client network is the network on which the VCH endpoint VM makes the Docker API available to Docker clients. By designating a specific client network, you isolate Docker endpoints from the public network. Virtual container hosts (VCHs) access vSphere Integrated Containers Management Portal and vSphere Integrated Containers Registry over the client network.

- `vic-machine` Option
- Example `vic-machine` Command

`vic-machine` Option

You designate a specific network for traffic between Docker clients and the VCH by specifying the `vic-machine create --client-network` option when you deploy the VCH.

`--client-network`

Short name: `--cln`

A port group on which the VCH makes the Docker API available to Docker clients. Docker clients use this network to issue Docker API requests to the VCH.

Usage:

```
--client-network port_group_name
```

If you do not specify this option, the VCH uses the public network for client traffic. If you specify an invalid port group name, `vic-machine create` fails and suggests valid port groups.

Example `vic-machine` Command

This example deploys a VCH with the following configuration:

- Specifies the target vCenter Server instance, the vCenter Server user name, password, datacenter and cluster, an image store, a port group for the bridge network, a name for the VCH, and the thumbprint of the vCenter Server certificate.
- Secures connections to the Docker API with an automatically generated server certificate, without client certificate verification, by setting `--no-tlsverify`.
- Directs public traffic to an existing port group named `network 1` and Docker API and management traffic to `network 2`.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
```

```
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--public-network 'network 1'
--management-network 'network 2'
--client-network 'network 2'
--name vch1
--thumbprint certificate_thumbprint
--no-tlsverify
```

Configure the Management Network

The management network is the network on which the VCH endpoint VM connects to vCenter Server and ESXi hosts. By designating a specific management network, you isolate connections to vSphere resources from the public network. The VCH uses this network to provide the `attach` function of the Docker API.

- `vic-machine` Options
- Example `vic-machine` Commands

`vic-machine` Options

You designate a specific network for traffic between the VCH and vSphere resources by specifying the `vic-machine create --management-network` option when you deploy the VCH. You can also route incoming connections from ESXi hosts to VCHs over the public network rather than over the management network by specifying the `--asymmetric-routes` option.

`--management-network`

Short name: `--mn`

A port group that the VCH uses to communicate with vCenter Server and ESXi hosts. Container VMs use this network to communicate with the VCH.

IMPORTANT: Because the management network provides access to your vSphere environment, and because container VMs use this network to communicate with the VCH, always use a secure network for the management network. Ideally, use separate networks for the management network and the container networks. The most secure setup is to make sure that VCHs can access vCenter Server and ESXi hosts directly over the management network, and that the management network has route entries for the subnets that contain both the target vCenter Server and the corresponding ESXi hosts. If the management network does not have route entries for the vCenter Server and ESXi host subnets, you must configure asymmetric routing. For more information about asymmetric routing, see the section on the `--asymmetric-routes` option.

When you create a VCH, `vic-machine create` checks that the firewall on ESXi hosts allows connections to port 2377 from the management network of the VCH. If access to port 2377 on ESXi hosts is subject to IP address restrictions, and if those restrictions block access to the management network interface, `vic-machine create` fails with a firewall configuration error:

```
Firewall configuration incorrect due to allowed IP restrictions on hosts:
"/ha-datacenter/host/localhost.localdomain/localhost.localdomain"
Firewall must permit dst 2377/tcp outbound to the VCH management interface
```

For information about how to open port 2377, see Open the Required Ports on ESXi Hosts.

NOTE: If the management network uses DHCP, `vic-machine` checks the firewall status of the management network before the VCH receives an IP address. It is therefore not possible to fully assess whether the firewall permits the IP address of the VCH. In this case, `vic-machine create` issues a warning.

```
Unable to fully verify firewall configuration due to DHCP use on management network
VCH management interface IP assigned by DHCP must be permitted by allowed IP settings
Firewall allowed IP configuration may prevent required connection on hosts:
"/ha-datacenter/host/localhost.localdomain/localhost.localdomain"
Firewall must permit dst 2377/tcp outbound to the VCH management interface
```

Usage:

```
--management-network port_group_name
```

If you do not specify this option, the VCH uses the public network for management traffic. If you specify an invalid port group name, `vic-machine create` fails and suggests valid port groups.

`--asymmetric-routes`

Short name: `--ar`

Allows incoming connections from ESXi hosts to VCHs over the public network rather than over the management network. This option allows containers on bridge networks to indirectly access assets on the management or client networks via the public interface, if those assets are routable from the public network. If the management network does not have route entries for the vCenter Server and ESXi host subnets, and you do not set `--asymmetric-routes`, containers that run without specifying `-d` remain in the starting state.

In this scenario, use the `--asymmetric-routes` option to allow management traffic from ESXi hosts to the VCH to pass over the public network. By setting the `--asymmetric-routes` option, you set reverse path forwarding in the VCH endpoint VM to loose mode rather than the default strict mode. For information about reverse path forwarding and loose mode, see https://en.wikipedia.org/wiki/Reverse_path_forwarding.

Usage:

```
--asymmetric-routes
```

The `--asymmetric-routes` option takes no arguments. If you do not set `--asymmetric-routes`, all management traffic is routed over the management network.

Example `vic-machine` Commands

This example deploys a VCH with the following configuration:

- Specifies the target vCenter Server instance, the vCenter Server user name, password, datacenter and cluster, an image store, a port group for the bridge network, a name for the VCH, and the thumbprint of the vCenter

Server certificate.

- Secures connections to the Docker API with an automatically generated server certificate, without client certificate verification, by setting `--no-tlsverify` .
- Directs public and management traffic to an existing port group named `network 1` and Docker API traffic to `network 2` .
- Does not specify the `--public-network` option, so traffic from container VMs and the VCH default to the VM Network.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--management-network 'network 1'
--client-network 'network 2'
--name vch1
--thumbprint certificate_thumbprint
--no-tls
```

The following example deploys a similar VCH, but specifies `--asymmetric-routes` to allow incoming connections from ESXi hosts to VCHs over the public network rather than over the management network.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--management-network 'network 1'
--client-network 'network 2'
--name vch1
--thumbprint certificate_thumbprint
--no-tlsverify
--asymmetric-routes
```

Configure Container Networks

Container networks are vSphere networks that the vSphere administrator makes directly available to container VMs. When you deploy a virtual container host (VCH), you provide a mapping of the vSphere network name to an alias that the VCH endpoint VM uses. You can share one network alias between multiple containers.

The mapped networks are available for use by the Docker API. Running `docker network ls` lists the container networks, and container developers can attach them to containers in the normal way by using commands such as `docker run` or `create` with the `--network=mapped-network-name` option, or `docker network connect`.

By using container networks, you can connect container VMs to any specific distributed port group or NSX logical switch, which gives the container VMs their own dedicated connection to the network. Container networks allow containerized applications to get their own routable IP address and become first class citizens of your datacenter. Using container networks provides you with the following advantages:

- **No single point of failure**: Every container VM has its own dedicated network connection, so even if the VCH endpoint VM fails there is no outage for your applications.
- **Network bandwidth sharing** Every container VM gets its own network interface and all of the bandwidth it can provide is available to the application. Traffic does not route though the VCH endpoint VM via network address translation (NAT) and containers do not share the public IP of the VCH.
- **No NAT conflicts**: There is no need for port mapping because every container VM gets its own IP address. Container services are directly exposed on the network without NAT, so applications that once could not run on containers can now run by using vSphere Integrated Containers.
- **No port conflicts**: Since every container VM gets its own IP address, you can have multiple application containers that require an exclusive port running on the same VCH.

Because container networks do not go through the VCH, the limitation of VCHs only supporting a maximum of 3 distinct network interfaces does not apply container networks.

- Basic `vic-machine` Option
- Configure Non-DHCP Container Networks
- Configure the Firewall on Container Networks
- Example `vic-machine` Command

NOTE: You can add or reconfigure container networks after you have deployed a VCH by using the `vic-machine configure --container-network` options. For information about adding or reconfiguring container networks, see Configure Container Network Settings in Configure Virtual Container Hosts.

Basic `vic-machine` Option

You designate a specific network for use by container VMs by specifying the `vic-machine create --container-network` option when you deploy the VCH.

```
--container-network
```

Short name: `--cn`

A port group for container VMs to use for external communication when container developers run `docker run` or `docker create` with the `--net` option.

IMPORTANT: For security reasons, whenever possible, use separate port groups for the container network and the management network.

To specify a container network, you provide the name of a port group for the container VMs to use, and an optional descriptive name for the container network for use by Docker. If you do not specify a descriptive name, Docker uses the vSphere network name.

- The port group must exist before you run `vic-machine create`.
- You cannot use the same port group as you use for the bridge network.
- You can create the port group on the same distributed virtual switch as the port group that you use for the bridge network.
- If the port group that you specify in the `container-network` option does not support DHCP, see Configure Non-DHCP Container Networks.
- The descriptive name that you provide appears under `Networks` when you run `docker info` or `docker network ls` on the deployed VCH. The descriptive name cannot include spaces. The descriptive name is optional unless the port group name contains spaces. If the port group name contains spaces, you must specify a descriptive name.
- Container developers use the descriptive name in the `--net` option when they run `docker run` or `docker create`.
- If you use shared NFS share points as volumes stores, it is recommended to make the NFS target accessible from the container network. If you use NFS volume stores and you do not specify a container network, containers use NAT to route traffic to the NFS target through the VCH endpoint VM. This can create potential bottlenecks and a single point of failure.
- If you intend to use the `--ops-user` option to use different user accounts for deployment and operation of the VCH, you must place any container network port groups in a network folder that has the `Read-Only` role with propagation enabled. For more information about the requirements when using `--ops-user`, see Use Different User Accounts for VCH Deployment and Operation.

Usage:

```
--container-network port_group_name:docker_display_name
```

You can specify `--container-network` multiple times to add multiple vSphere networks to Docker.

If you do not specify `--container-network`, or if you deploy containers that do not use a container network, the containers' network services are still be available via port mapping through the VCH, by using NAT through the public interface of the VCH. If you specify an invalid port group name, `vic-machine create` fails and suggests valid port groups.

Configure Non-DHCP Container Networks

If the network that you specify in the `--container-network` option does not support DHCP, you must specify the `--container-network-gateway` option. You can optionally specify one or more DNS servers and a range of IP addresses for container VMs on the container network.

`--container-network-gateway`

Short name: `--cng`

The gateway for the subnet of the container network. This option is required if the network that you specify in the `--container-network` option does not support DHCP. Specify the gateway in the format `container_network:subnet`. If you specify this option, it is recommended that you also specify the `--container-network-dns` option.

When you specify the container network gateway, you must use the port group that you specify in the `--container-network` option.

Usage:

```
--container-network-gateway port_group_name:gateway_ip_address/subnet_mask
```

If you specify `--container-network-gateway` but you do not specify `--container-network`, or if you specify a different port group to the one that you specify in `--container-network`, `vic-machine create` fails with an error.

`--container-network-dns`

Short name: `--cnd`

The address of a DNS server for the container network. This option is recommended if the network that you specify in the `--container-network` option does not support DHCP.

When you specify the container network DNS server, you must use the port group that you specify in the `--container-network` option.

Usage:

```
--container-network-dns port_group_name:8.8.8.8
```

You can specify `--container-network-dns` multiple times, to configure multiple DNS servers. If you specify `--container-network-dns` but you do not specify `--container-network`, or if you specify a different port group to the one that you specify in `--container-network`, `vic-machine create` fails with an error.

`--container-network-ip-range`

Short name: `--cnr`

The range of IP addresses that container VMs can use if the network that you specify in the `container-network` option does not support DHCP. If you specify `--container-network-ip-range`, VCHs manage the addresses for containers within that range. The range that you specify must not be used by other computers or VMs on the network. You must also specify `--container-network-ip-range` if container developers need to deploy containers with static IP addresses. If you specify `container-network-gateway` but do not specify `--container-network-ip-range`, the IP range for container VMs is the entire subnet that you specify in `--container-network-gateway`.

When you specify the container network IP range, you must use the port group that you specify in the `--container-network` option.

Usage:

```
--container-network-ip-range port_group_name:192.168.100.2-192.168.100.254
```

You can also specify the IP range as a CIDR.

```
--container-network-ip-range port_group_name:192.168.100.0/24
```

If you specify `--container-network-ip-range` but you do not specify `--container-network`, or if you specify a different port group to the one that you specify in `--container-network`, `vic-machine create` fails with an error.

Configure the Firewall on Container Networks

You can configure the trust level of container networks by setting the `--container-network-firewall` option.

`--container-network-firewall`

Short name: `--cnf`

The `--container-network-firewall` option allows you to set the following levels of trust.

Trust Level	Description
`closed`	No traffic can come in or out of the container interface, even if developers expose ports on containers.
`outbound`	Only outbound connections are permitted. Use this setting if the VCH will host applications that consume but do not provide services.
`peers`	Only connections to other containers with the same `peers` interface are permitted. To enforce the `peers` trust level, you must set the `--container-network-ip-range` on the container network. The VCH applies a network rule so that container traffic is only allowed over that IP range. If you do not specify an IP range, the container network uses DHCP and there is no way that the VCH can determine whether or not a container at a given IP address is a peer to another container. In this case, the VCH defaults to the `open` setting, and it treats all connections as peer connections. Use the `peers` setting for container VMs that need to communicate with each other but not with the external world.
`published`	Only connections to published ports is permitted.
`open`	All traffic is permitted and developers can decide which ports to expose.

Usage:

```
--container-network-firewall port_group_name:trust_level
```

If you do not set `--container-network-firewall`, the default level of trust is `published`. As a consequence, if you do not set `--container-network-firewall`, container developers must explicitly specify `-p 80` in `docker run` and `docker create` commands to publish port 80 on a container. Obliging developers to specify the ports to expose improves security

and gives you more awareness of your environment and applications.

Example `vic-machine` Command

This example deploys a VCH with the following configuration:

- Specifies the target vCenter Server instance, the vCenter Server user name, password, datacenter and cluster, an image store, a port group for the bridge network, a name for the VCH, and the thumbprint of the vCenter Server certificate.
- Secures connections to the Docker API with an automatically generated server certificate, without client certificate verification, by setting `--no-tlsverify`.
- Uses the default VM Network for the public, management, and client networks.
- Designates a port group named `vic-containers` for use by container VMs that are run with the `--net` option.
- Gives the container network the name `vic-container-network`, for use by Docker.
- Specifies the gateway, two DNS servers, and a range of IP addresses on the container network for container VMs to use.
- Opens the firewall on the container network for outbound connections.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--container-network vic-containers:vic-container-network
--container-network-gateway vic-containers:gateway_ip_address/24
--container-network-dns vic-containers:dns1_ip_address
--container-network-dns vic-containers:dns2_ip_address
--container-network-ip-range vic-containers:192.168.100.0/24
--container-network-firewall vic-containers:outbound
--name vch1
--thumbprint certificate_thumbprint
--no-tlsverify
```

Specify a Static IP Address for the VCH Endpoint VM

By default, `vic-machine create` obtains IP addresses for virtual container host (VCH) endpoint VMs by using DHCP. You can specify a static IP address for the VCH endpoint VM on the client, public, and management networks. DHCP is used for the endpoint VM for any network on which you do not specify a static IP address.

- `vic-machine` Options
- Example `vic-machine` Command

`vic-machine` Options

You can configure a static IP address for a VCH endpoint VM on the different networks by specifying the `vic-machine create --client-network-ip`, `--public-network-ip`, and `--management-network-ip` options when you deploy the VCH. You can also specify one or more DNS servers and gateway addresses by using the `--dns-server`, `--client-network-gateway`, `--public-network-gateway`, and `--management-network-gateway` options.

`--dns-server`

Short name: None

A DNS server for the VCH endpoint VM to use on the client, public, or management networks. You can specify `dns-server` multiple times, to configure multiple DNS servers.

- If you specify `dns-server`, `vic-machine create` uses the same `--dns-server` setting for all three of the client, public, and management networks.
- If you do not specify `dns-server` and you specify a static IP address for the VCH endpoint VM on all three of the client, public, and management networks, `vic-machine create` uses the Google public DNS service.
- If you do not specify `dns-server` and you use DHCP for all of the client, public, and management networks, `vic-machine create` uses the DNS servers that DHCP provides.

Usage:

```
--dns-server=172.16.10.10
--dns-server=172.16.10.11
```

`--client-network-ip`, `--public-network-ip`, `--management-network-ip`

Short names: None

A static IP address for the VCH endpoint VM on the public, client, or management network.

You specify a static IP address for the endpoint VM on the public, client, or management networks by using the `--public-network-ip`, `client-network-ip`, and `management-network-ip` options.

- You can only specify one static IP address on a given port group. If more than one of the client, public, or management networks share a port group, you can only specify a static IP address on one of those networks. All of the networks that share that port group use the IP address that you specify.
- If you set a static IP address for the VCH endpoint VM on the public network, you must specify a corresponding gateway address by using the `--public-network-gateway` option. If the management and client networks are L2 adjacent to their gateways, you do not need to specify the corresponding gateways for those networks.
- If either of the client or management networks shares a port group with the public network, you can only specify a static IP address on the public network.
- If either or both of the client or management networks do not use the same port group as the public network, you can specify a static IP address for the endpoint VM on those networks by using `--client-network-ip` or `--management-network-ip`, or both. In this case, you must specify a corresponding gateway address by using `client/management-network-gateway`.
- If the client and management networks both use the same port group, and the public network does not use that port group, you can set a static IP address for the endpoint VM on either or both of the client and management networks.
- If you assign a static IP address to the VCH endpoint VM on the client network by setting the `--client-network-ip` option, and you do not specify one of the TLS options, `vic-machine create` uses this address as the Common Name with which to auto-generate trusted CA certificates. If you do not specify `--tls-cname`, `--no-tls` or `--no-tlsverify`, two-way TLS authentication with trusted certificates is implemented by default when you deploy the VCH with a static IP address on the client network. If you assign a static IP address to the endpoint VM on the client network, `vic-machine create` creates the same certificate and environment variable files as described in the `--tls-cname` option.

    IMPORTANT: If the client network shares a port group with the public network you cannot set a static IP address for the endpoint VM on the client network. To assign a static IP address to the endpoint VM you must set a static IP address on the public network by using the `--public-network-ip` option. In this case, `vic-machine create` uses the public network IP address as the Common Name with which to auto-generate trusted CA certificates, in the same way as it would for the client network.

You specify addresses as IPv4 addresses with a network mask.

Usage:

```
--public-network-ip 192.168.X.N/24
--management-network-ip 192.168.Y.N/24
--client-network-ip 192.168.Z.N/24
```

You can also specify addresses as resolvable FQDNs.

```
--public-network-ip=vch27-team-a.internal.domain.com
--management-network-ip=vch27-team-b.internal.domain.com
--client-network-ip=vch27-team-c.internal.domain.com
```

If you do not specify an IP address for the endpoint VM on a given network, `vic-machine create` uses DHCP to obtain an IP address for the endpoint VM on that network.

`--client-network-gateway` , `--public-network-gateway` , `--management-network-gateway`

Short names: None

The gateway to use if you use `--public/client/management-network-ip` to specify a static IP address for the VCH endpoint VM on the public, client, or management networks. If you specify a static IP address on the public network, you must specify a gateway by using the `--public-network-gateway` option. If the management and client networks are L2 adjacent to their gateways, you do not need to specify the gateway for those networks.

You specify gateway addresses as IP addresses without a network mask.

Usage:

```
--public-network-gateway 192.168.X.1
```

The default route for the VCH endpoint VM is always on the public network. As a consequence, if you specify a static IP address on either of the management or client networks and those networks are not L2 adjacent to their gateways, you must specify the routing destination for those networks in the `--management-network-gateway` and `--client-network-gateway` options. You specify the routing destination or destinations in a comma-separated list, with the address of the gateway separated from the routing destinations by a colon (:).

```
--management-network-gateway routing_destination_1,
routing_destination_2:gateway_address
```

```
--client-network-gateway routing_destination_1,
routing_destination_2:gateway_address
```

In the following example, `--management-network-gateway` informs the VCH that it can reach all of the vSphere management endoints that are in the ranges 192.168.3.0-255 and 192.168.128.0-192.168.128.255 by sending packets to the gateway at 192.168.2.1. Ensure that the address ranges that you specify include all of the systems that will connect to this VCH instance.

```
--management-network-gateway 192.168.3.0/24,192.168.128.0/24:192.168.2.1
```

Example `vic-machine` Command

If you specify networks for any or all of the public, management, and client networks, you can deploy the VCH so that the VCH endpoint VM has a static IP address on one or more of those networks.

This example deploys a VCH with the following configuration:

- Specifies the target vCenter Server instance, the vCenter Server user name, password, datacenter and cluster, an image store, a port group for the bridge network, a name for the VCH, and the thumbprint of the vCenter Server certificate.

- Directs public and management traffic to network 1 and Docker API traffic to network 2.
- Sets a DNS server for use by the public, management, and client networks.
- Sets a static IP address and subnet mask for the VCH endpoint VM on the public and client networks. Because the management network shares a network with the public network, you only need to specify the public network IP address. You cannot specify a management IP address because you are sharing a port group between the management and public network.
- Specifies the gateway for the public network. If you set a static IP address on the public network, you must also specify the gateway address.
- Does not specify a gateway for the client network. It is not necessary to specify a gateway on either of the client or management networks if those networks are L2 adjacent to their gateways.
- Because this example specifies a static IP address for the VCH endpoint VM on the client network, `vic-machine create` uses this address as the Common Name with which to create auto-generated trusted certificates. Full TLS authentication is implemented by default, so no TLS options are specified.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--public-network 'network 1'
--public-network-ip 192.168.1.10/24
--public-network-gateway 192.168.1.1
--management-network 'network 1'
--client-network 'network 2'
--client-network-ip 192.168.3.10/24
--dns-server dns_server_address
--thumbprint certificate_thumbprint
--name vch1
```

Configure VCHs to Use Proxy Servers

If access to the Internet or to your private image registries requires the use of a proxy server, you must configure a virtual container host (VCH) to connect to the proxy server when you deploy it. The proxy is used only when pulling images, and not for any other purpose.

IMPORTANT: Configuring a VCH to use a proxy server does not configure proxy support on the containers that this VCH runs. Container developers must configure proxy servers on containers when they create them.

- `vic-machine` Options
- Example `vic-machine` Command

NOTE: You can add, reconfigure, or remove proxy servers after you have deployed a VCH by using the `vic-machine configure --https-proxy` and `--http-proxy` options. For information about adding, reconfiguring, or removing proxy servers, see Add, Configure, or Remove Proxy Servers in Configure Virtual Container Hosts.

`vic-machine` Options

You configure a VCH to use a proxy server by specifying either of the `vic-machine create --https-proxy` or `--http-proxy` options when you deploy the VCH.

`--https-proxy`

Short name: None

The address of the HTTPS proxy server through which the VCH accesses image registries when using HTTPS. Specify the address of the proxy server as either an FQDN or an IP address.

Usage:

```
--https-proxy https://proxy_server_address:port
```

`--http-proxy`

Short name: None

The address of the HTTP proxy server through which the VCH accesses image registries when using HTTP. Specify the address of the proxy server as either an FQDN or an IP address.

Usage:

```
--http-proxy http://proxy_server_address:port
```

Example `vic-machine` Command

If your network access is controlled by a proxy server, you must configure a VCH to connect to the proxy server when you deploy it, so that it can pull images from external sources.

This example deploys a VCH with the following configuration:

- Specifies the target vCenter Server instance, the vCenter Server user name, password, datacenter and cluster, an image store, a port group for the bridge network, a name for the VCH, and the thumbprint of the vCenter Server certificate.
- Secures connections to the Docker API with an automatically generated server certificate, without client certificate verification, by setting `--no-tlsverify`.
- Configures the VCH to access the network via an HTTPS proxy server.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--https-proxy https://proxy_server_address:port
--name vch1
--thumbprint certificate_thumbprint
--no-tlsverify
```

Set Environment Variables for Key `vic-machine` Options

If you deploy multiple virtual container hosts (VCHs) to the same vCenter Server instance or ESXi host, you can simplify `vic-machine` commands by setting environment variables for certain key `vic-machine` options.

You can set environment variables for the following `vic-machine` options.

Option	Variable	Description
`--target`	`VIC_MACHINE_TARGET`	The address of the vCenter Server instance or ESXi host on which you are deploying VCHs.
`--user`	`VIC_MACHINE_USER`	The user name for the vSphere account that you use when running `vic-machine` commands. Use an account with administrator privileges.
`--password`	`VIC_MACHINE_PASSWORD`	The password for the vSphere user account.
`--thumbprint`	`VIC_MACHINE_THUMBPRINT`	The thumbprint of the vCenter Server or ESXi host certificate.

NOTE: You cannot include the vSphere user name and password in the `VIC_MACHINE_TARGET` environment variable. You must either specify the user name and password in the `VIC_MACHINE_USER` and `VIC_MACHINE_PASSWORD` environment variables, or use the `--user` and `--password` options when you run `vic-machine`.

For information about how to obtain the vCenter Server certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

When you run any of the different `vic-machine` commands, `vic-machine` checks whether environment variables are present in the system. If you have set any or all of the environment variables, `vic-machine` automatically uses the values from those environment variables. You only need to specify the additional `vic-machine` options.

The following examples show some simplified `vic-machine` commands that you can run if you set all four environment variables.

- List VCHs:

  ```
  vic-machine-operating_system ls
  ```

- Inspect a VCH:

  ```
  vic-machine-operating_system inspect --id vm-123
  ```

- Create a basic VCH:

  ```
  vic-machine-operating_system create --bridge-network vic-bridge --no-tls --name vch-no-
  tls
  ```

- Upgrade a VCH:

  ```
  vic-machine-operating_system upgrade --id vm-123
  ```

- Configure a VCH, for example to add a new volume store:

  ```
  vic-machine-operating_system configure --id vm-123 --volume-store
  datastore_name/datastore_path:default
  ```

- Delete a VCH:

  ```
  vic-machine-operating_system delete --id vm-123
  ```

Advanced Examples of Deploying a VCH

This topic provides examples of the options of the `vic-machine create` command to use when deploying virtual container hosts (VCHs) in various vSphere configurations.

- General Deployment Examples
  - Deploy to a vCenter Server Cluster with Multiple Datacenters and Datastores
  - Deploy to a Specific Standalone Host in vCenter Server
  - Deploy to a Resource Pool on an ESXi Host
  - Deploy to a Resource Pool in a vCenter Server Cluster
  - Set Limits on Resource Use
- Specify One or More Volume Stores
- Security Examples
  - Use Auto-Generated Trusted CA Certificates
  - Use Custom Server Certificates
  - Combine Custom Server Certificates and Auto-Generated Client Certificates
  - Specify Different User Accounts for VCH Deployment and Operation
- Registry Server Examples
  - Authorize Access to an Insecure Private Registry Server
  - Authorize Access to Secure Registries and vSphere Integrated Containers Registry
  - Authorize Access to a Whitelist of Registries

For simplicity, all examples that do not relate explicitly to certificate use specify the `--no-tls` option.

For detailed descriptions of all of the `vic-machine create` options, see VCH Deployment Options. For information about how to obtain the certificate thumbprint before running `vic-machine`, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

NOTE: Wrap any option arguments that include spaces or special characters in quotes. Use single quotes if you are using `vic-machine` on a Linux or Mac OS system and double quotes on a Windows system.

Option arguments that might require quotation marks include the following:

- User names and passwords in `--target`, `--user`, `--password`, `--ops-user`, and `--ops-password`.
- Datacenter names in `--target`.
- VCH names in `--name`.
- Datastore names and paths in `--image-store` and `--volume-store`.
- Network and port group names in all networking options.
- Cluster and resource pool names in `--compute-resource`.
- Folder names in the paths for `--tls-cert-path`, `--tls-server-cert`, `--tls-server-key`, `--appliance-iso`, and `--bootstrap-iso`.

General Deployment Examples

The examples in this section demonstrate the deployment of VCHs in different vSphere environments.

Deploy to a vCenter Server Cluster with Multiple Datacenters and Datastores

If vCenter Server has more than one datacenter, you specify the datacenter in the `--target` option.

If vCenter Server manages more than one cluster, you use the `--compute-resource` option to specify the cluster on which to deploy the VCH.

When deploying a VCH to vCenter Server, you must use the `--bridge-network` option to specify an existing port group for container VMs to use to communicate with each other. For information about how to create a distributed virtual switch and port group, see the section on vCenter Server Network Requirements in Environment Prerequisites for VCH Deployment.

This example deploys a VCH with the following configuration:

- Provides the vCenter Single Sign-On user and password in the `--target` option. The user name is wrapped in quotes, because it contains the `@` character.
- Deploys a VCH named `vch1` to the cluster `cluster1` in datacenter `dc1` .
- Uses a port group named `vic-bridge` for the bridge network.
- Designates `datastore1` as the datastore in which to store container images, the files for the VCH appliance, and container VMs.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--name vch1
--thumbprint certificate_thumbprint
--no-tls
```

Deploy to a Specific Standalone Host in vCenter Server

If vCenter Server manages multiple standalone ESXi hosts that are not part of a cluster, you use the `--compute-resource` option to specify the address of the ESXi host to which to deploy the VCH.

This example deploys a VCH with the following configuration:

- Specifies the user name, password, image store, bridge network, and name for the VCH.
- Deploys the VCH on the ESXi host with the FQDN `esxihost1.organization.company.com` in the datacenter `dc1` . You can also specify an IP address.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--image-store datastore1
--bridge-network vch1-bridge
--compute-resource esxihost1.organization.company.com
--name vch1
--thumbprint certificate_thumbprint
```

```
--no-tls
```

Deploy to a Resource Pool on an ESXi Host

To deploy a VCH in a specific resource pool on an ESXi host that is not managed by vCenter Server, you specify the resource pool name in the `--compute-resource` option.

This example deploys a VCH with the following configuration:

- Specifies the user name and password, image store, and a name for the VCH.
- Designates `rp 1` as the resource pool in which to place the VCH. The resource pool name is wrapped in quotes, because it contains a space.

```
vic-machine-operating_system create
--target root:password@esxi_host_address
--compute-resource 'rp 1'
--image-store datastore1
--name vch1
--thumbprint certificate_thumbprint
--no-tls
```

Deploy to a Resource Pool in a vCenter Server Cluster

To deploy a VCH in a resource pool in a vCenter Server cluster, you specify the resource pool in the `compute-resource` option.

This example deploys a VCH with the following configuration:

- Specifies the user name, password, datacenter, image store, bridge network, and name for the VCH.
- Designates `rp 1` as the resource pool in which to place the VCH. In this example, the resource pool name `rp 1` is unique across all hosts and clusters, so you only need to specify the resource pool name.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource 'rp 1'
--image-store datastore1
--bridge-network vch1-bridge
--name vch1
--thumbprint certificate_thumbprint
--no-tls
```

If the name of the resource pool is not unique across all clusters, for example if two clusters each contain a resource pool named `rp 1`, you must specify the full path to the resource pool in the `compute-resource` option, in the format *cluster_name*/Resources/*resource_pool_name*.

```
vic-machine-operating_system create
```

```
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource 'cluster 1'/Resources/'rp 1'
--image-store datastore1
--bridge-network vch1-bridge
--name vch1
--thumbprint certificate_thumbprint
--no-tls
```

Set Limits on Resource Use

To limit the amount of system resources that the container VMs in a VCH can use, you can set resource limits on the VCH vApp.

This example deploys a VCH with the following configuration:

- Specifies the user name, password, image store, cluster, bridge network, and name for the VCH.
- Sets resource limits on the VCH by imposing memory and CPU reservations, limits, and shares.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--memory 1024
--memory-reservation 1024
--memory-shares low
--cpu 1024
--cpu-reservation 1024
--cpu-shares low
--name vch1
--thumbprint certificate_thumbprint
--no-tls
```

For more information about setting resource use limitations on VCHs, see the Advanced Deployment Options and Advanced Resource Management Options sections in VCH Deployment Options.

Specify Volume Stores

If container application developers will use the `docker volume create` command to create containers that use volumes, you must create volume stores when you deploy VCHs. You specify volume stores in the `--volume-store` option. You can specify `--volume-store` multiple times to create multiple volume stores.

When you create a volume store, you specify the name of the datastore to use and an optional path to a folder on that datastore. You also specify a descriptive name for that volume store for use by Docker.

This example deploys a VCH with the following configuration:

- Specifies the user name, password, datacenter, cluster, bridge network, and name for the VCH.
- Specifies the `volumes` folder on `datastore 1` as the default volume store. Creating a volume store named `default` allows container application developers to create anonymous or named volumes by using `docker create -v`.
- Specifies a second volume store named `volume_store_2` in the `volumes` folder on `datastore 2`.
- Specifies a volume store named `shared_volume` in a NFS share point, from which containers can mount shared volumes.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--bridge-network vch1-bridge
--image-store 'datastore 1'
--volume-store 'datastore 1'/volumes:default
--volume-store 'datastore 2'/volumes:volume_store_2
--volume-store nfs://nfs_store/path/to/share/point:shared_volume
--name vch1
--thumbprint certificate_thumbprint
--no-tls
```

For more information about volume stores, see the volume-store section in VCH Deployment Options.

Security Examples

The examples in this section demonstrate how to configure a VCH to use Certificate Authority (CA) certificates to enable `TLSVERIFY` in your Docker environment, and to allow access to insecure registries of Docker images.

Use Auto-Generated Trusted CA Certificates

You can deploy a VCH that implements two-way authentication with trusted auto-generated TLS certificates that are signed by a CA.

To automatically generate a server certificate that can pass client verification, you must specify the Common Name (CN) for the certificate by using the `--tls-cname` option. The CN should be the FQDN or IP address of the server, or a domain with a wildcard. The CN value must match the name or address that clients will use to connect to the server. You can use the `--organization` option to add basic descriptive information to the server certificate. This information is visible to clients if they inspect the server certificate.

If you specify an existing CA file with which to validate clients, you must also provide an existing server certificate that is compatible with the `--tls-cname` value or the IP address of the client interface.

This example deploys a VCH with the following configuration:

- Specifies the user, password, datacenter, image store, cluster, bridge network, and name for the VCH.
- Provides a wildcard domain `*.example.org` as the FQDN for the VCH, for use as the Common Name in the certificate. This assumes that there is a DHCP server offering IP addresses on VM Network, and that those

addresses have corresponding DNS entries such as `dhcp-a-b-c.example.com`.

- Specifies a folder in which to store the auto-generated certificates.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--tls-cname *.example.org
--tls-cert-path path_to_cert_folder
--thumbprint certificate_thumbprint
--name vch1
```

The Docker API for this VCH will be accessible at `https://dhcp-a-b-c.example.com:2376`.

For more information about using auto-generated CA certificates, see the section Restrict Access to the Docker API with Auto-Generated Certificates in VCH Deployment Options.

Use Custom Server Certificates

You can create a VCH that uses a custom server certificate, for example a server certificate that has been signed by Verisign or another public root. You use the `--tls-server-cert` and `--tls-server-key` options to provide the paths to a custom X.509 certificate and its key when you deploy a VCH. The paths to the certificate and key files must be relative to the location from which you are running `vic-machine create`.

This example deploys a VCH with the following configuration:

- Specifies the user name, password, image store, cluster, bridge network, and name for the VCH.
- Provides the paths relative to the current location of the `*.pem` files for the custom server certificate and key files.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--tls-server-cert ../some/relative/path/certificate_file.pem
--tls-server-key ../some/relative/path/key_file.pem
--name vch1
--thumbprint certificate_thumbprint
```

For more information about using custom server certificates, see the section Restrict Access to the Docker API with Custom Certificates in VCH Deployment Options.

Combine Custom Server Certificates and Auto-Generated Client Certificates

You can create a VCH with a custom server certificate by specifying the paths to custom `server-cert.pem` and `server-key.pem` files in the `--tls-server-cert` and `--tls-server-key` options. The key should be un-encrypted. Specifying the `--tls-server-cert` and `--tls-server-key` options for the server certificate does not affect the automatic generation of client certificates. If you specify the `--tls-cname` option to match the common name value of the server certificate, `vic-machine create` generates self-signed certificates for Docker client authentication and deployment of the VCH succeeds.

This example deploys a VCH with the following configuration:

- Specifies the user name, password, image store, cluster, bridge network, and name for the VCH.
- Provides the paths relative to the current location of the `*.pem` files for the custom server certificate and key files.
- Specifies the common name from the server certificate in the `--tls-cname` option. The `--tls-cname` option is used in this case to ensure that the certificate is valid for the resulting VCH, given the network configuration.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--tls-server-cert ../some/relative/path/certificate_file.pem
--tls-server-key ../some/relative/path/key_file.pem
--tls-cname cname_from_server_cert
--name vch1
--thumbprint certificate_thumbprint
```

Specify Different User Accounts for VCH Deployment and Operation

When you deploy a VCH, you can use different vSphere user accounts for deployment and for operation. This allows you to run VCHs with lower levels of privileges than are required for deployment.

This example deploys a VCH with the following configuration:

- Specifies the image store and name for the VCH.
- Specifies *vsphere_admin* in the `--target` option, to identify the user account with vSphere Administrator privileges with which to deploy the VCH.
- Specifies *vsphere_user* and its password in the `--ops-user` and `--ops-password` options, to identify the user account with which the VCH runs. The user account that you specify in `--ops-user` must is different to the vSphere Administrator account that you use for deployment, and must exist before you deploy the VCH.
- Specifies a resource pool in which to deploy the VCH in the `--compute-resource` option.
- Specifies the VCH port groups in the `--bridge-network` and `--container-network` options.

```
vic-machine-operating_system create
--target vsphere_admin:vsphere_admin_password@vcenter_server_address/dc1
--compute-resource cluster1/VCH_pool
--image-store datastore1
--bridge-network vch1-bridge
```

```
--container-network vic-containers:vic-container-network
--name vch1
--ops-user vsphere_user
--ops-password vsphere_user_password
--thumbprint certificate_thumbprint
--no-tls
```

For information about the permissions that the `--ops-user` account requires, and the permissions to set on the resource pool for the VCH and on the network folders, see Use Different User Accounts for VCH Deployment and Operation.

Registry Server Examples

The examples in this section demonstrate how to configure a VCH to use a private registry server, for example vSphere Integrated Containers Registry.

Authorize Access to an Insecure Private Registry Server

To authorize connections from a VCH to a private registry server without verifying the certificate of that registry, set the `--insecure-registry` option. If you authorize a VCH to connect to an insecure private registry server, the VCH attempts to access the registry server via HTTP if access via HTTPS fails. VCHs always use HTTPS when connecting to registry servers for which you have not authorized insecure access. You can specify `insecure-registry` multiple times to allow connections from the VCH to multiple insecure private registry servers.

NOTE: You cannot configure VCHs to connect to vSphere Integrated Containers Registry instances as insecure registries. Connections to vSphere Integrated Containers Registry always require HTTPS and a certificate.

This example deploys a VCH with the following configuration:

- Specifies the user name, password, image store, cluster, bridge network, and name for the VCH.
- Authorizes the VCH to pull Docker images from the insecure private registry servers located at the URLs *registry_URL_1* and *registry_URL_2*.
- The registry server at *registry_URL_2* listens for connections on port 5000.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--insecure-registry registry_URL_1
--insecure-registry registry_URL_2:5000
--name vch1
--thumbprint certificate_thumbprint
--no-tls
```

For more information about configuring VCHs to allow connections to insecure private registry servers, see the section on the `--insecure-registry` option in VCH Deployment Options.

Authorize Access to Secure Registries and vSphere Integrated Containers Registry

For an example of how to use `--registry-ca` to authorize access to vSphere Integrated Containers Registry or to another secure registry, see Deploy a VCH for Use with vSphere Integrated Containers Registry.

Authorize Access to a Whitelist of Registries

To restrict the registries to which a VCH allows access, set the `--whitelist-registry` option. You can specify `--whitelist-registry` multiple times to add multiple registries to the whitelist. You use `--whitelist-registry` in combination with the `--registry-ca` and `--insecure-registry` options.

This example deploys a VCH with the following configuration:

- Specifies the user name, password, image store, cluster, bridge network, and name for the VCH.
- Adds to the whitelist:

```
- The single registry instance running at 10.2.40.40:443
- All registries running in the range 10.2.2.1/24
- All registries in the domain *.mycompany.com
```

- Provides the CA certificate for the registry instance 10.2.40.40:443.
- Adds a single instance of an insecure registry to the whitelist by specifying `--insecure-registry`.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--whitelist-registry="10.2.40.40:443"
--whitelist-registry=10.2.2.1/24
--whitelist-registry=*.mycompany.com
--registry-ca=/home/admin/mycerts/ca.crt
--insecure-registry=192.168.100.207
--name vch1
--thumbprint certificate_thumbprint
--no-tls
```

For more information about configuring VCHs to allow connections to a whitelist of registries, see the section on the `--whitelist-registry` option in VCH Deployment Options.

Deploy a VCH for Use with vSphere Integrated Containers Registry

To use vSphere Integrated Containers Engine with vSphere Integrated Containers Registry, you must obtain the registry certificate and pass it to a virtual container host (VCH) when you create that VCH.

When you deployed the vSphere Integrated Containers appliance, vSphere Integrated Containers Registry auto-generated a Certificate Authority (CA) certificate. You can download the registry CA certificate from the vSphere Integrated Containers Management Portal.

Prerequisites

- You downloaded the vSphere Integrated Containers Engine bundle from http://*vic_appliance_address*.
- Obtain the vCenter Server or ESXi host certificate thumbprint. For information about how to obtain the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

Procedure

1. Log in to the vSphere Integrated Containers Management Portal with a vSphere administrator or Cloud administrator user account.

   vSphere administrator accounts for the Platform Service Controller with which vSphere Integrated Containers is registered are automatically granted Cloud Admin access.

2. Go to **Administration** > **Configuration**, and click the link to download the **Registry Root Cert**.
3. Use `vic-machine create` to deploy a VCH, specifying the registry's CA certificate by using the `--registry-ca` option.

   You can configure the VCH to connect to multiple registries by specifying `--registry-ca` multiple times.

   For simplicity, this example deploys a VCH with the `--no-tls` flag, so that container application developers do not need to use a TLS certificate to connect a Docker client to the VCH. However, the connection between the VCH and the registry still requires certificate authentication.

   ```
   vic-machine-operating_system create
   --target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
   --compute-resource cluster1
   --image-store datastore1
   --bridge-network vch1-bridge
   --name vch_registry
   --thumbprint vcenter_server_certificate_thumbprint
   --no-tlsverify
   --registry-ca=cert_path/ca.crt
   ```

   Optionally, you can use the `--whitelist-registry` option to limit this VCH so that it can only access certain registries. This example limits access to registries in your company's domain, but you could specify the address of a specific registry, or a CIDR range of addresses.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--name vch_registry
--thumbprint vcenter_server_certificate_thumbprint
--no-tlsverify
--registry-ca=cert_path/ca.crt
--whitelist-registry *.mycompany.com
```

Result

The VCH has a copy of the registry certificate and can connect to this vSphere Integrated Containers Registry instance.

Deploy a Virtual Container Host for Use with `dch-photon`

This version of vSphere Integrated Containers includes an image repository named `dch-photon`, that is pre-loaded in the `default-project` in vSphere Integrated Containers Registry.

The `dch-photon` image allows container developers to deploy a standard Docker container host that runs in a Photon OS container. Container developers can use this Docker engine to perform operations in standard Docker. For example, developers can use `dch-photon` containers to perform operations that virtual container hosts (VCHs) do not support in this version of vSphere Integrated Containers, such as `docker build` and `docker push`.

For container developers to be able to deploy containers from the `dch-photon` image, you must deploy VCHs with a specific configuration.

Prerequisites

- You downloaded the vSphere Integrated Containers Engine bundle from http://*vic_appliance_address*.
- Obtain the vCenter Server or ESXi host certificate thumbprint. For information about how to obtain the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

Procedure

1. Log in to the vSphere Integrated Containers Management Portal with a vSphere administrator, Cloud Admin or DevOps admin user account.

   vSphere administrator accounts for the Platform Services Controller with which vSphere Integrated Containers is registered are automatically granted Cloud Admin access.

2. Go to **Administration** > **Configuration**, and click the link to download the **Registry Root Cert**.
3. Use `vic-machine create` to deploy a VCH.

   - The VCH must be able to pull the `dch-photon` image from the vSphere Integrated Containers Registry instance. You must specify the registry's CA certificate by using the `--registry-ca` option.
   - A `dch-photon` container creates an anonymous volume, and as such requires named `default`.

   For simplicity, this example deploys a VCH with the `--no-tls` flag, so that container application developers do not need to use a TLS certificate to connect a Docker client to the VCH. However, the connection between the VCH and the registry still requires certificate authentication.

   ```
   vic-machine-operating_system create
   --target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
   --compute-resource cluster1
   --image-store datastore1
   --bridge-network vch-bridge
   --name vch_dch_photon
   --thumbprint vcenter_server_certificate_thumbprint
   --no-tlsverify
   --registry-ca cert_path/ca.crt
   ```

```
--volume-store datastore_name:default
```

You could also specify `--volume-store nfs://datastore_name/path_to_share_point:default` to designate an NFS share point as the default volume store.

Result

The VCH that you deployed can access vSphere Integrated Containers Registry, and has a volume store named `default` . It is ready for container developers to use with `dch-photon` containers.

Use Different User Accounts for VCH Deployment and Operation

A virtual container host (VCH) appliance requires the appropriate permissions in vSphere to perform various tasks during VCH operation.

During deployment of a VCH, `vic-machine` uses the vSphere account that you specify in either of the `vic-machine create --user` or `--target` options for all deployment operations. Deployment of a VCH requires a user account with vSphere Administrator privileges.

Day-to-day operation of a VCH requires fewer permissions than deployment. You can configure a VCH so that it uses different user accounts for deployment and for operation by using the `vic-machine create --ops-user` and `--ops-password` options when you deploy the VCH. By specifying `--ops-user`, you can limit the post-deployment permissions of the VCH to only those vSphere operations that it needs. If you do not specify `--ops-user`, the VCH runs with full vSphere Administrator privileges.

After deployment, a VCH must have permission to perform the following operations:

- Create, modify, and delete VMs within its resource pool
- Reconfigure the endpoint VM
- Validate host firewall configuration and system licenses

When you deploy a VCH, a user account that you specify in `--ops-user` must have the correct privileges to allow the VCH to perform these operations. vSphere Integrated Containers Engine does not currently create the required roles, so to assign privileges to the `--ops-user` user account, you must manually create user roles in vSphere before you deploy the VCH. You assign privileges to those roles, and assign the roles to the user account to use in `--ops-user`.

- For information about how to create vSphere roles, see vSphere Permissions and User Management Tasks in the vSphere documentation.
- For information about how to assign permissions to objects in the vSphere Inventory, see Add a Permission to an Inventory Object in the vSphere documentation.

Procedure

1. In the vSphere Web Client, create a user group, for example `VIC Ops Users`, and add the appropriate user accounts to the user group.

   The best practice when assigning roles in vSphere is to assign the roles to user groups and then to add users to those groups, rather than assigning roles to the users directly.

2. Go to **Administration** > **Roles** and create one role for each type of inventory object that VCHs need to access.

   It is possible to create a single role, but by creating multiple roles you keep the privileges of the VCH as granular as possible.

Role to Create	Required Permissions
VCH - vcenter	Datastore > Configure datastore
VCH - datacenter	Datastore > Configure datastore

VCH - datacenter	Datastore > Configure datastore Datastore > Low level file operations
VCH - datastore	Datastore > AllocateSpace Datastore > Browse datastore Datastore > Configure datastore Datastore > Remove file Datastore > Low level file operations Host > Configuration > System management
VCH - network	Network > Assign network
VCH - endpoint	dvPort group > Modify dvPort group > Policy operation dvPort group > Scope operation Resource > Add virtual machine * vApp > Add virtual machine VirtualMachine > Configuration > Add existing disk VirtualMachine > Configuration > Add new disk VirtualMachine > Configuration > Add or remove device VirtualMachine > Configuration > Advanced VirtualMachine > Configuration > Modify device settings VirtualMachine > Configuration > Remove disk VirtualMachine > Configuration > Rename VirtualMachine > Guest operations > Guest operation program execution VirtualMachine > Interaction > Device connection VirtualMachine > Interaction > Power off VirtualMachine > Interaction > Power on VirtualMachine > Inventory > Create new VirtualMachine > Inventory > Remove VirtualMachine > Inventory > Register VirtualMachine > Inventory > Unregister

* If you use both of the `--ops-user` and `--use-rp` options when you create a VCH, you must include the **Resource** > **Add virtual machine** permission in the `VCH - endpoint` role. The **vApp** > **Add virtual machine** permission is not required if you deploy the VCH with the `--use-rp` option.

3. Go to **Networking**, create a network folder, and place the distributed virtual switches that the VCHs will use for the bridge network and any container networks into that folder.

   The parent object of distributed virtual switches that the VCH uses as the bridge network and container networks must be set to `Read-Only`, with **Propagate to Children** enabled. By placing distributed virtual switches in a network folder, you avoid setting an entire datacenter to `Read-Only`. This restriction only applies to the bridge network and container networks. When you specify the `vic-machine create --bridge-network` and `--container-network` options, include the full inventory path to the networks in the following format:

   ```
   datacenter/network/network_folder/port_group_name
   ```

4. (Optional) Go to **Hosts and Clusters** and create a resource pool in which to deploy VCHs.

   By creating a resource pool for VCHs, you can set the correct permissions on just that resource pool rather than on an entire host or cluster. You specify this resource pool in the `vic-machine create --compute-resource` option when you deploy the VCH. For a more granular application of privileges, you can also apply the permissions directly to VCH vApps after deployment, rather than to a resource pool.

5. In each of the **Hosts and Clusters**, **Storage**, and **Networking** views, select inventory objects and assign the user group and the appropriate role to each one.

Inventory Object	Role to Assign	Propagate
Top-level vCenter Server instance	`VCH - vcenter`	No
Datacenters	`VCH - datacenter`	No
Clusters. All datastores in the cluster inherit permissions from the cluster.	`VCH - datastore`	Yes
Standalone VMware vSAN datastores	`VCH - datastore`	No
Standalone datastores	`VCH - datastore`	No
Network folders	`Read-only`	Yes
Port groups	`VCH - network`	No
Resource pools for VCHs	`VCH - endpoint`	Yes
VCH vApps, for a very granular application of privileges	`VCH - endpoint`	Yes

What to do next

Use `vic-machine create --ops-user=<user_account>` to deploy VCHs that operate with restricted privileges. Ensure that the various vSphere inventory objects that you specify as arguments have the user group with the appropriate role. For an example of a `vic-machine` command with the `--ops-user` option, see the section *Specify Different User Accounts for VCH Deployment and Operation* Advanced Examples of Deploying a VCH.

Virtual Container Host Administration

You can monitor and perform administration tasks on virtual container hosts (VCHs) in the vSphere Client, by using `vic-machine` , and in the VCH Admin Portal.

- Interoperability
- Virtual Container Host Administration in the vSphere Client
- Virtual Container Host Administration with `vic-machine`
- Virtual Container Host Administration Portal

Interoperability of vSphere Integrated Containers with Other VMware Software

vSphere administrators can use vSphere to view and manage the vSphere Integrated Containers appliance, virtual container hosts (VCHs), and container VMs. You can use any vSphere feature to manage the vSphere Integrated Containers appliance without affecting its behavior.

This topic describes the interoperability of vSphere Integrated Containers Engine with other vSphere features and VMware products.

- Performing Operations on VCHs and Container VMs in vSphere
- VMware vRealize® Suite
- VMware vSphere vMotion®
- VMware vSphere High Availability
- VMware NSX®
- Maintenance Mode
- Storage
- Enhanced Linked Mode Environments
- vSphere Features Not Supported in This Release

Performing Operations on VCHs and Container VMs in vSphere

- If you restart a VCH endpoint VM, it comes back up in the same state that it was in when it shut down.
- If you use DHCP on the client network, the IP address of the VCH endpoint VM might change after a restart. Use `vic-machine inspect` to obtain the new IP address.
- Do not manually delete a VCH vApp, the VCH endpoint VM, or container VMs. Always use `vic-machine delete` to delete VCHs and use Docker commands to perform operations on container VMs.
- Manually restarting container VMs, either individually or by manually restarting the VCH vApp, can result in incorrect end-times for container operations. Do not manually restart the vApp or container VMs. Always use Docker commands to perform operations on container VMs.

VMware vRealize Suite

Your organization could use VMware vRealize Automation to provide a self-provisioning service for VCHs, by using the vRealize Automation interface or APIs to request VCHs. At the end of the provisioning process, vRealize Automation would communicate the VCH endpoint VM address to the requester. If you deploy VCHs with TLS authentication, `vic-machine create` generates a file named `vch_name.env`. The `env` file contains Docker environment variables that are specific to the VCH. vRealize Automation could potentially provide the `env` file at the end of a provisioning process for VCHs.

VMware vSphere vMotion

You can use vMotion to move VCHs without needing to take the container VMs offline. The VCH endpoint VM does not need to be running for vMotion to occur on the container VMs. Clusters with a mix of container VMs and non-container VMs can use vMotion with fully automated DRS.

VMware vSphere High Availability

You can apply vSphere High Availability to clusters on which VCHs and container VMs run. If the host on which a VCH or container VMs are running goes offline, the VCH and container VMs migrate to another host in the cluster. VCHs restart on the new host immediately. Container VMs that were running before the migration restart one by one, after the VCH has restarted.

VMware NSX

You can deploy the vSphere Integrated Containers appliance on an NSX network. VCHs require distributed port groups and a bridge network. You can deploy VCHs to NSX networks if those networks are configured to provide distributed port groups. You can use NSX networks for the management, bridge, public, and container networks.

Maintenance Mode

In a cluster with fully automated DRS, if you put a host into maintenance mode, DRS migrates the VCHs and container VMs to another host in the cluster. Putting hosts into maintenance mode requires manual intervention in certain circumstances:

- If VCHs and container VMs are running on a standalone ESXi host, you must power off the VCHs and container VMs before you put the host into maintenance mode.
- If container VMs have active `docker attach` sessions, you cannot put the host into maintenance mode until the `attach` sessions end.

Storage

VCHs maintain file system layers inherent in container images by mapping to discrete VMDK files, all of which can be housed in shared vSphere datastores, including vSAN, NFS, Fibre Channel, and iSCSI datastores.

Enhanced Linked Mode Environments

You can deploy VCHs in Enhanced Linked Mode environments. Any vCenter Server instance in the Enhanced Linked Mode environment can access VCH and container VM information.

vSphere Features Not Supported in This Release

vSphere Integrated Containers Engine does not currently support the following vSphere features:

- vSphere Storage DRS™: You cannot configure VCHs to use Storage DRS datastore clusters. However, you can specify the path to a specific datastore within a Storage DRS datastore cluster by specifying the full inventory path to the datastore in the `vic-machine create --image-store` option. For example, `--image-store /dc1/datastore/my-storage-pod/datastore1` . You can also specify the relative path from a datastore folder in a datacenter, for example `--image-store my-storage-pod/datastore1` .
- vSphere Fault Tolerance: vSphere Integrated Containers does not implement vSphere Fault Tolerance. However, VCH processes that stop unexpectedly do restart automatically, independently of vSphere Fault Tolerance.
- vSphere Virtual Volumes™: You cannot use Virtual Volumes as the target datastores for image stores or volume stores.
- Snapshots: Creating and reverting to snapshots of the VCH endpoint VM or container VMs can cause vSphere Integrated Containers Engine not to function correctly.

Virtual Container Host Administration in the vSphere Client

vSphere Integrated Containers provides a basic plug-in for the Flex-based vSphere Web Client on vCenter Server 6.0 or 6.5. vSphere Integrated Containers provides a plug-in with more complete functionality for the HTML5 vSphere Client. The HTML5 vSphere Client is only available with vSphere 6.5.

- View All VCH and Container Information
- View Individual VCH and Container Information

View All VCH and Container Information in the HTML5 vSphere Client

If you have installed the HTML5 plug-in for vSphere Integrated Containers, you can find information about your vSphere Integrated Containers deployment in the HTML5 vSphere Client.

IMPORTANT: Do not use the vSphere Client or to perform operations on virtual container hosts or container VMs. Specifically, using the vSphere Client to power off, power on, or delete the VCH vApp or VCH endpoint VM can cause vSphere Integrated Containers Engine to not function correctly. Always use `vic-machine` to perform operations on VCHs. The vSphere Client does not allow you to delete container VMs, but do not use the vSphere Client to power container VMs on or off. Always use Docker commands to perform operations on containers.

NOTE: More functionality will be added to the vSphere Integrated Containers view in future releases.

Prerequisites

- You are running vCenter Server 6.5.0d or later. The vSphere Integrated Containers view does not function with earlier versions of vCenter Server 6.5.0.
- You installed the HTML5 plug-in for vSphere Integrated Containers.

Procedure

1. Log in to the HTML5 vSphere Client and go to the **Home** page.
2. Click **vSphere Integrated Containers**.

   The vSphere Integrated Containers view presents the number of VCHs and container VMs that you have deployed.

3. Click **vSphere Integrated Containers** in the main panel and select the **Summary** tab.

   The **Summary** tab shows the version of vSphere Integrated Containers that you are running and the number of VCHs.

4. Select the **Virtual Container Hosts** tab.

   The **Virtual Container Hosts** tab provides information about the VCHs that are registered with this vCenter Server instance:

   - Lists all VCHs by name. Click the VCH name to go to the Summary tab for the VCH endpoint VM.
   - Indicates that the VCH is running correctly.
   - Displays the `DOCKER_HOST` environment variable that container developers use to connect to this VCH.
   - Provides the link to the VCH Admin Portal for this VCH.

5. Select the **Containers** tab.

   The **Containers** tab shows information about all of the container VMs that are running in this vCenter Server instance, for all VCHs:

   - Lists all containers by name.
   - Indicates whether the container VM is powered on or off.

- Provides information about the memory, CPU, and storage consumption of the container VM.
- Lists the port number and the protocol of any mapped ports that the container VM exposes.
- Provides links to the Summary tabs for the VCH that manages the container VM and for the VM itself.
- Displays the image from which this container VM was created.

View Individual VCH and Container Information in the vSphere Clients

After you have installed the client plug-ins for vSphere Integrated Containers, you can find information about individual virtual container hosts (VCHs) and container VMs in the HTML5 vSphere Client or the Flex-based vSphere Web Client.

IMPORTANT: Do not use the vSphere Client or to perform operations on virtual container hosts or container VMs. Specifically, using the vSphere Client to power off, power on, or delete the VCH vApp or VCH endpoint VM can cause vSphere Integrated Containers Engine to not function correctly. Always use `vic-machine` to perform operations on VCHs. The vSphere Client does not allow you to delete container VMs, but do not use the vSphere Client to power container VMs on or off. Always use Docker commands to perform operations on containers.

Prerequisites

- You deployed a VCH and at least one container VM.
- You installed the plug-ins for vSphere Integrated Containers.

Procedure

1. Log in to either the HTML5 vSphere Client or the Flex-based vSphere Web Client.
2. On the **Home** page, select **Hosts and Clusters**.
3. Expand the hierarchy of vCenter Server objects to navigate to the VCH vApp.
4. Expand the VCH vApp and select the VCH endpoint VM.

   Information about the VCH appears in the **Virtual Container Host** portlet in the **Summary** tab:

   - The `DOCKER_HOST` environment variable that container developers use to connect to this VCH.
   - The link to the VCH Admin Portal for this VCH.
5. Select a container VM.

   Information about the container VM appears in the **Container** portlet in the **Summary** tab:

   - The name of the running container. If the container developer used `docker run -name container_name` to run the container, `container_name` appears in the portlet.
   - The image from which the container was deployed.
   - If the container developer used `docker run -p port` to map a port when running the container, the port number and the protocol appear in the portlet.

Virtual Container Host Administration with `vic-machine`

The `vic-machine` utility provides commands that allow you to manage existing virtual container hosts (VCHs).

- Obtain `vic-machine` Version Information
- Common `vic-machine` Options
- List Virtual Container Hosts and Obtain their IDs
- Obtain General Virtual Container Host Information and Connection Details
- Obtain Virtual Container Host Configuration Information
- Configure Virtual Container Hosts
- Delete Virtual Container Hosts
- Debug Virtual Container Hosts

Obtain `vic-machine` Version Information

You can obtain information about the version of `vic-machine` by using the `vic-machine version` command.

The `vic-machine version` command has no arguments.

Example

```
$ vic-machine-operating_system version
```

Output

The `vic-machine` utility displays the version of the instance of `vic-machine` that you are using.

```
vic-machine-operating_system
version vic_machine_version-vic_machine_build-git_commit
```

- `vic_machine_version` is the version number of this release of vSphere Integrated Containers Engine.
- `vic_machine_build` is the build number of this release.
- `tag` is the short `git commit` checksum for the latest commit for this build.

Common `vic-machine` Options

This section describes the options that are common to all `vic-machine` commands. The common options that `vic-machine` requires relate to the vSphere environment in which you deployed the virtual container host (VCH), and to the VCH itself.

NOTE: Wrap any option arguments that include spaces or special characters in quotes. Use single quotes if you are using `vic-machine` on a Linux or Mac OS system and double quotes on a Windows system.

Option arguments that might require quotation marks include the following:

- User names and passwords in `--target`, or in `--user` and `--password`.
- Datacenter names in `--target`.
- VCH names in `--name`.
- Datastore names and paths in `--image-store`.
- Cluster and resource pool names in `--compute-resource`.

You can set environment variables for the `--target`, `--user`, `--password`, and `--thumbprint` options. For information about setting environment variables, see Set Environment Variables for Key `vic-machine` Options.

`--target`

Short name: `-t`

The IPv4 address, fully qualified domain name (FQDN), or URL of the ESXi host or vCenter Server instance on which you deployed the VCH. This option is always **mandatory**.

- If the target ESXi host is not managed by vCenter Server, provide the address of the host.

  ```
  --target esxi_host_address
  ```

- If the target ESXi host is managed by vCenter Server, or if you deployed the VCH to a cluster, provide the address of vCenter Server.

  ```
  --target vcenter_server_address
  ```

- You can include the user name and password in the target URL. The user account that you specify must have vSphere administrator privileges.

  ```
  --target vcenter_or_esxi_username:password@vcenter_or_esxi_address
  ```

  If you do not include the user name in the target URL, you must specify the `user` option. If you do not specify the `password` option or include the password in the target URL, `vic-machine` prompts you to enter the password.

- If you deployed the VCH on a vCenter Server instance that includes more than one datacenter, include the datacenter name in the target URL. If you include an invalid datacenter name, `vic-machine` fails and suggests the available datacenters that you can specify.

```
    --target vcenter_server_address/datacenter_name
```

`--user`

Short name: `-u`

The ESXi host or vCenter Server user account with which to run the `vic-machine` command. This option is mandatory if you do not specify the username in the `target` option. The user account that you specify in `--user` must have vSphere administrator privileges.

```
  --user esxi_or_vcenter_server_username
```

`--password`

Short name: `-p`

The password for the user account on the vCenter Server on which you deployed the VCH, or the password for the ESXi host if you deployed directly to an ESXi host. If not specified, `vic-machine` prompts you to enter the password.

```
  --password esxi_host_or_vcenter_server_password
```

`--thumbprint`

Short name: None

The thumbprint of the vCenter Server or ESXi host certificate. Specify this option if your vSphere environment uses untrusted, self-signed certificates. If your vSphere environment uses trusted certificates that are signed by a known Certificate Authority (CA), you do not need to specify the `--thumbprint` option.

If you run `vic-machine` without the specifying the `--thumbprint` option and the operation fails, the resulting error message includes the certificate thumbprint. Always verify that the thumbprint in the error message is valid before attempting to run the command again.

For information about how to obtain the certificate thumbprint either before running `vic-machine` or to verify a thumbprint from a `vic-machine` error message, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

You can bypass certificate thumbprint verification by specifying the `--force` option instead of `--thumbprint`.

CAUTION: It is not recommended to use `--force` to bypass thumbprint verification in production environments. Using `--force` in this way exposes VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials.

Use upper-case letters and colon delimitation in the thumbprint. Do not use space delimitation.

```
  --thumbprint certificate_thumbprint
```

`--compute-resource`

Short name: `-r`

The relative path to the host, cluster, or resource pool in which you deployed the VCH. Specify `--compute-resource` with exactly the same value that you used when you ran `vic-machine create`. You specify the `compute-resource` option in the following circumstances:

- vCenter Server includes multiple instances of standalone hosts or clusters, or a mixture of standalone hosts and clusters.
- You deployed the VCH in a specific resource pool in your environment.

If you specify the `id` option, you do not need to specify the `compute-resource` option.

If you do not specify the `compute-resource` or `id` options and multiple possible resources exist, `vic-machine` fails and suggests valid targets for `compute-resource` in the failure message.

- If the VCH is in a specific resource pool on an ESXi host, specify the name of the resource pool:

    ```
    --compute-resource  resource_pool_name
    ```

- If the VCH is on a vCenter Server instance that has more than one standalone host but no clusters, specify the IPv4 address or fully qualified domain name (FQDN) of the target host:

    ```
    --compute-resource host_address
    ```

- If the VCH is on a vCenter Server with more than one cluster, specify the name of the target cluster:

    ```
    --compute-resource cluster_name
    ```

- If the VCH is in a specific resource pool on a standalone host that is managed by vCenter Server, specify the IPv4 address or FQDN of the target host and name of the resource pool:

    ```
    --compute-resource host_name/resource_pool_name
    ```

- If the VCH is in a specific resource pool in a cluster, specify the names of the target cluster and the resource pool:

    ```
    --compute-resource cluster_name/resource_pool_name
    ```

`--name`

Short name: `-n`

The name of the VCH. This option is mandatory if the VCH has a name other than the default name, `virtual-container-host`, or if you do not use the `id` option. Specify `--name` with exactly the same value that you used when you ran `vic-machine create`. This option is not used by `vic-machine ls`.

```
--name vch_appliance_name
```

`--id`

Short name: None

The vSphere Managed Object Reference, or moref, of the VCH, for example `vm-100` . You obtain the ID of a VCH by running `vic-machine ls` . If you specify the `id` option, you do not need to specify the `--name` or `--compute-resource` options. This option is not used by `vic-machine create` or `vic-machine version` .

```
--id vch_id
```

`--timeout`

Short name: none

The timeout period for performing operations on the VCH. Specify a value in the format `XmYs` if the default timeout is insufficient.

```
--timeout 5m0s
```

List Virtual Container Hosts and Obtain Their IDs

You can obtain a list of the virtual container hosts (VCHs) that are running in vCenter Server or on an ESXi host by using the `vic-machine ls` command. The `vic-machine ls` command lists VCHs with their IDs, names, and versions, and informs you whether upgrades are available for the VCHs.

The `vic-machine ls` command does not include any options in addition to the common options described in Common `vic-machine` Options.

- To obtain a list of all VCHs that are running on an ESXi host or vCenter Server instance, you must provide the address of the target ESXi host or vCenter Server.
- You must specify the username and optionally the password, either in the `--target` option or separately in the `--user` and `--password` options.
- If your vSphere environment uses untrusted, self-signed certificates, you must specify the thumbprint of the vCenter Server instance or ESXi host in the `--thumbprint` option. For information about how to obtain the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

  Use upper-case letters and colon delimitation in the thumbprint. Do not use space delimitation.

Example

This example specifies the vCenter Server credentials in the `--target` option.

```
$ vic-machine-operating_system ls
--target vcenter_server_username:password@vcenter_server_address
--thumbprint certificate_thumbprint
```

Output

The `vic-machine ls` command lists the VCHs that are running on the ESXi host or vCenter Server instance that you specified.

```
ID         PATH      NAME     VERSION     UPGRADE STATUS
vm-101     path      vch_1    version     Upgradeable to version
vm-102     path      vch_2    version     Up to date
[...]
vm-n       path      vch_n    version     Up to date
```

- The IDs are the vSphere Managed Object References, or morefs, for the VCH endpoint VMs. You can use VCH IDs when you run the `vic-machine inspect`, `debug`, `upgrade`, and `delete` commands. Using VCH IDs reduces the number of options that you need to specify when you run those commands.
- The `PATH` value depends on where the VCH is deployed:
  - ESXi host that is not managed by vCenter Server:

```
/ha-datacenter/host/host_name/Resources
```

- Standalone host that is managed by vCenter Server:

```
/datacenter/host/host_address/Resources
```

- vCenter Server cluster:

```
/datacenter/host/cluster_name/Resources
```

If VCHs are deployed in resource pools on hosts or clusters, the resource pool names appear after `Resources` in the path. You can use the information in `PATH` in the `--compute-resource` option of `vic-machine` commands.

- The `VERSION` value shows the version of `vic-machine` that was used to create the VCH. It includes the release version, the build number and the short Git commit checksum, in the format `vch_version-vch_build-git_commit`.

- The `UPGRADE STATUS` reflects whether the current version of `vic-machine` that you are using is the same as the one that you used to deploy a VCH. If the version or build number of the VCH does not match that of `vic-machine`, `UPGRADE STATUS` is `Upgradeable to vch_version-vch_build-git_commit`.

Obtain General Virtual Container Host Information and Connection Details

You can obtain general information about a virtual container host (VCH) and its connection details by using the `vic-machine inspect` command.

In addition to the common options described in Common `vic-machine` Options, the `vic-machine inspect` command only includes one option, `--tls-cert-path`.

- You must specify the username and optionally the password, either in the `--target` option or separately in the `--user` and `--password` options.
- If the VCH has a name other than the default name, `virtual-container-host`, you must specify the `--name` or `--id` option.
- If multiple compute resources exist in the datacenter, you must specify the `--compute-resource` or `--id` option.
- If your vSphere environment uses untrusted, self-signed certificates, you must specify the thumbprint of the vCenter Server instance or ESXi host in the `--thumbprint` option. For information about how to obtain the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

  Use upper-case letters and colon delimitation in the thumbprint. Do not use space delimitation.

- If the VCH implements server and client authentication (`tlsverify`) and uses a non-default location to store its certificates, specify the `--tls-cert-path` option. If you do not specify `--tls-cert-path`, `vic-machine inspect` looks for valid certificates in `$PWD`, `$PWD/$vch_name` and `$HOME/.docker`.

Examples

The following example includes the options required to obtain information about a named instance of a VCH from a simple vCenter Server environment.

```
$ vic-machine-operating_system inspect
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --name vch_name
```

The following example includes the `--tls-cert-path` option, for a VCH that stores client certificates in a non-default location.

```
$ vic-machine-operating_system inspect
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --name vch_name
    --tls-cert-path path_to_certificates
```

Output

The `vic-machine inspect` command displays general information about the VCH, its version and upgrade status, and details about how to connect to the VCH:

- The VCH ID:

  ```
  VCH ID: VirtualMachine:vm-101
  ```

  The vSphere Managed Object Reference, or moref, of the VCH. You can use the VCH ID when you run the `vic-machine delete`, `configure`, or `debug` commands. Using a VCH ID reduces the number of options that you need to specify when you run those commands.
- The version of the `vic-machine` utility and the version of the VCH that you are inspecting.

  ```
  Installer version: vic_machine_version-vic_machine_build-git_commit
  VCH version: vch_version-vch_build-git_commit
  ```

- The upgrade status of the VCH:

  ```
  VCH upgrade status:
  Installer has same version as VCH
  No upgrade available with this installer version
  ```

  If `vic-machine inspect` reports a difference between the version or build number of `vic-machine` and the version or build number of the VCH, the upgrade status is `Upgrade available`.
- The address of the VCH Admin portal for the VCH.

  ```
  VCH Admin Portal:
  https://vch_address:2378
  ```

- The address at which the VCH publishes ports.

  ```
  vch_address
  ```

- The Docker environment variables that container developers can use when connecting to this VCH, depending on the the level of security that the VCH implements.
    - VCH with server and client authentication (`tlsverify`):

      ```
      DOCKER_TLS_VERIFY=1
      DOCKER_CERT_PATH=path_to_certificates
      DOCKER_HOST=vch_address:2376
      ```

      If `vic-machine inspect` is unable to find the appropriate client certificates, either in the default location or in a location that you specify in the `--tls-cert-path` option, the output includes a warning.

```
Unable to find valid client certs
DOCKER_CERT_PATH must be provided in environment or certificates specified
individually via CLI arguments
```

- VCH with TLS server authentication but without client authentication (`no-tlsverify`):

  ```
  DOCKER_HOST=vch_address:2376
  ```

- VCH with no TLS authentication (`no-tls`):

  ```
  DOCKER_HOST=vch_address:2375
  ```

- The Docker command to use to connect to the Docker endpoint, depending on the the level of security that the VCH implements.
  - VCH with server and client authentication (`tlsverify`):

    ```
    docker -H vch_address:2376 --tlsverify info
    ```

  - VCH with TLS server authentication but without client authentication (`no-tlsverify`):

    ```
    docker -H vch_address:2376 --tls info
    ```

  - VCH with no TLS authentication (`no-tls`):

    ```
    docker -H vch_address:2375 info
    ```

Obtain Virtual Container Host Configuration Information

You can obtain information about the configuration of a virtual container host (VCH) by using the `vic-machine inspect config` command. The `inspect config` command provides details of the options with which the VCH was deployed with `vic-machine create` or subsequently reconfigured with `vic-machine configure`.

The `inspect config` command only includes one option, `--format`, the value of which can be either `verbose` or `raw`.

- `verbose` : Provides an easily readable list of the options with which the VCH was deployed. If you do not specify `--format`, `config` provides verbose output by default.
- `raw` : Provides the options with which the VCH was deployed in command line option format. You can copy or pipe the output into a `vic-machine create` command, to create an identical VCH.

Verbose Example

The following example obtains the configuration of a VCH by using its VCH ID. It does not specify `--format`, so the command provides verbose output.

```
$ vic-machine-operating_system inspect config
    --target 'Administrator@vsphere.local':password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
```

Output

By default, the `vic-machine inspect config` command lists the options with which the VCH was deployed in the easily readable verbose format.

```
Target VCH created with the following options:


        --target=vcenter_server_address
        --thumbprint=certificate_thumbprint
        --name=vch1
        --compute-resource=/datacenter_name/host/vcenter_server_address/Resources
        --ops-user=Administrator@vsphere.local
        --image-store=ds://datastore1
        --volume-store=ds://datastore1/volumes:default
        --volume-store=ds://datastore1/volumes:vol1
        --bridge-network=vic-bridge
        --public-network=vic-public
        --memory=1024
```

```
        --cpu=1024
```

In addition to the minimum required `vic-machine create` options, the VCH in this example was deployed with two
volume stores, named `default` and `vol1`, a specific public network, and constraints on memory and cpu usage.
Also, because the VCH was not deployed with the `--ops-user` option, `config` lists `--ops-user` as
Administrator@vsphere.local, which is the same user account as the one that was used to deploy the VCH.

Raw Example

The following example specifies the `--format raw` option.

```
$ vic-machine-operating_system inspect config
    --target 'Administrator@vsphere.local':password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --format raw
```

Output

The `vic-machine inspect config` command lists the options with which the VCH was deployed in command line format.

```
--target=vcenter_server_address --thumbprint=certificate_thumbprint --name=vch1 --compute-
resource=/datacenter_name/host/vcenter_server_address/Resources --ops-
user=Administrator@vsphere.local --image-store=ds://datastore1 --volume-
store=ds://datastore1%5Cvolumes%5Ctlsverify:default --volume-
store=ds://datastore1%5Cvolumes%5Cconfigtest:vol1 --bridge-network=vic-bridge --public-
network=vic-public --memory=1024 --cpu=1024
```

Configure Virtual Container Hosts

You can configure certain settings on an existing virtual container host (VCH) by using the `vic-machine configure` command.

When you run `vic-machine configure`, you use the options described in Common `vic-machine` Options to identify the VCH to configure. In addition to these options, the `vic-machine configure` command provides options that allow you to perform the following modifications on VCHs:

- Update vCenter Server Credentials
- Update vCenter Server Certificates
- Add or Update Registry Server Certificates
- Update Security Configuration
- Add Volume Stores
- Add and Reset DNS Servers
- Configure Container Network Settings
- Add, Configure, or Remove Proxy Servers
- Configure Debug Mode
- Configure CPU and Memory Allocations
- Reset Upgrade or Configuration Progress

To see the current configuration of a VCH before you configure it, and to check the new configuration, run `vic-machine inspect config` before and after you run `vic-machine configure`. For information about running `vic-machine inspect config`, see Obtain VCH Configuration Information.

IMPORTANT: Running `vic-machine inspect config` before you run `vic-machine configure` is especially important if you are adding registry certificates, volume stores, DNS servers, or container networks to a VCH that already includes one or more of those elements. When you add registry certificates, volume stores, DNS servers, or container networks to a VCH, you must specify the existing configuration as well as any new configurations in separate instances of the appropriate `vic-machine inspect config` option.

When you run a `vic-machine configure` operation, `vic-machine` takes a snapshot of the VCH endpoint VM before it makes any modifications to the VCH. However, `vic-machine` does not remove the snapshot when the configuration operation finishes. You must manually remove the snapshot, after verifying that the configuration operation was successful.

Update vCenter Server Credentials

If the vCenter Server credentials change after the deployment of a VCH, you must update that VCH with the new credentials. The VCH will not function until you update the credentials.

You provide the new vCenter Server credentials in the `vic-machine configure --ops-user` and `--ops-password` options. You use the `vic-machine configure --ops-user` and `--ops-password` options to update the credentials even if you did not specify the `vic-machine create --ops-user` and `--ops-password` options during the initial deployment of the VCH. If you

did not specify `vic-machine create --ops-user` and `--ops-password` during the deployment of the VCH, by default the VCH uses the values from `vic-machine create --user` and `--password` for the `--ops-user` and `--ops-password` settings, and it uses these credentials for day-to-day, post-deployment operation.

For example, if you specified `--user Administrator@vsphere.local` in the `vic-machine create` command, and you did not set the `vic-machine create --ops-user` and `--ops-password` options, the VCH automatically sets `--ops-user` to Administrator@vsphere.local and uses this account for post-deployment operations. Consequently, if the password for Administrator@vsphere.local changes, you must specify the `vic-machine configure --ops-user` and `--ops-password` options to update the password. This example specifies the `--user` and `--password` options to log into vCenter Server, and then specifies `--ops-user` and `--ops-password` to update those settings in the VCH.

```
$ vic-machine-operating_system configure
    --target vcenter_server_address
    --user Administrator@vsphere.local
    --password password
    --thumbprint certificate_thumbprint
    --id vch_id
    --ops-user Administrator@vsphere.local
    --ops-password new_password
```

You can also use the `vic-machine configure --ops-user` and `--ops-password` options to configure an operations user on a VCH that was not initially deployed with that option. Similarly, you can use `--ops-user` and `--ops-password` to change the operations user account on a VCH that was deployed with an operations user account, or to update the password for a previously specified operations user account. This example specifies the credentials to log into vCenter Server in the `--target` option, rather than in `--user` and `--password`.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --ops-user new_operations_user_account
    --ops-password password
```

Update vCenter Server Certificates

If the vCenter Server certificate changes, you must update any VCHs running on that vCenter Server instance, otherwise they will no longer function.

To update the certificate, provide the new certificate thumbprint to the VCH in the `--thumbprint` option:

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --id vch_id
    --thumbprint new_certificate_thumbprint
```

If you run `vic-machine configure` with the `--force` option and you do not specify `--thumbprint`, `vic-machine` updates the thumbprint automatically.

CAUTION: It is not recommended to use `--force` to automatically update thumbprints in production environments. Using `--force` in this way exposes VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials. For information about how to obtain the vCenter Server certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

Add or Update Registry Server Certificates

If a VCH requires access to a new vSphere Integrated Containers Registry instance, or to another private registry, you can add new registry CA certificates by using the `vic-machine configure --registry-ca` option. You also use the `vic-machine configure --registry-ca` option if the certificate for an existing registry changes.

The `vic-machine configure --registry-ca` option functions in the same way as the equivalent `vic-machine create --registry-ca` option. For information about the `vic-machine create --registry-ca` option, see Private Registry Options in VCH Deployment Options.

This example updates the certificate for a registry that this VCH already uses.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --registry-ca path_to_new_ca_cert_for_existing_registry
```

If you are adding registry certificates to a VCH that already has one or more registry certificates, you must also specify each existing registry certificate in a separate instance of `--registry-ca`. This example passes the CA certificate for a new registry to a VCH and specifies the existing certificate for a registry that this VCH already uses.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --registry-ca path_to_ca_cert_for_existing_registry
    --registry-ca path_to_ca_cert_for_new_registry
```

NOTE: Unlike `vic-machine create`, the `vic-machine configure` command does not provide an `--insecure-registry` option.

Update Security Configuration

You can configure the security settings of a VCH by using the different TLS options of the `vic-machine configure` command.

- To configure TLS authentication with automatically generated certificates on a VCH that currently implements no TLS authentication, or to regenerate automatically generated certificates, use the `vic-machine configure --tls-cname` option.
- To configure TLS authentication with custom certificates on a VCH that currently implements no TLS authentication, or that uses automatically generated certificates, or to replace existing custom certificates, use the `vic-machine configure --tls-server-cert` and `--tls-server-key` options.
- To disable verification of client certificates, use the `vic-machine configure --no-tlsverify` option.
- To change the location in which to search for and store certificates, use the `vic-machine configure --tls-cert-path` option.

The `vic-machine configure` TLS options function in the same way as the equivalent `vic-machine create` options. For information about the `vic-machine create` security options, see Security Options in VCH Deployment Options.

NOTE: The `vic-machine configure` command does not include an equivalent to `vic-machine create --tls-ca` option.

This example sets the `vic-machine configure --tls-cname` option to implement TLS authentication with automatically generated server and client certificates. Before the configuration, the VCH either has no authentication or uses automatically generated certificates that you want to regenerate. The `--tls-cert-path` option specifies the folder in which to store the generated certificate.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --tls-cname *.example.com
    --tls-cert-path path_to_cert_folder
```

This example uses the `vic-machine configure --tls-server-cert` and `--tls-server-key` options to implement TLS authentication with custom certificates. Before the configuration, the VCH either has no TLS authentication, or it uses automatically generated certificates, or it uses custom certificates that require replacement.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --tls-server-cert path_to_cert/certificate_name.pem
    --tls-server-key path_to_key/key_name.pem
```

This example sets `--no-tlsverify` to disable the verification of client certificates on a VCH that implements client and server authentication.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --no-tlsverify
```

Add Volume Stores

You can add volume stores to VCHs by using the `vic-machine configure --volume-store` option. You can add vSphere datastores and NFS datastores with shared mount points to a VCH.

The `vic-machine configure --volume-store` option functions in the same way as the equivalent `vic-machine create --volume-store` option. For information about the `vic-machine create --volume-store` option, see `--volume-store` in VCH Deployment Options.

If you are adding volume stores to a VCH that already has one or more volume stores, you must specify each existing volume store in a separate instance of `--volume-store`.

This example adds a new NFS volume store to a VCH. The VCH already has an existing volume store with the label `default`, that is backed by a vSphere datastore.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --volume-store datastore_name/datastore_path:default
    --volume-store nfs://datastore_name/path_to_share_point:nfs_volume_store_label
```

NOTE: The current version of vSphere Integrated Containers does not allow you to remove volume stores from a VCH.

Add and Reset DNS Servers

If you deployed the VCH with a static IP address, you can add DNS servers or reset them to the default by using the `vic-machine configure --dns-server` option.

The `vic-machine configure --dns-server` option functions in the same way as the equivalent `vic-machine create --dns-server` option. For information about the `vic-machine create --dns-server` option, see `--dns-server` in Specify a Static IP Address for the VCH Endpoint VM.

If you are adding DNS servers to a VCH that already includes one or more DNS servers, you must also specify each existing DNS server in a separate instance of `--dns-server`. This example adds a new DNS server, `dns_server_2`, to a VCH that already uses `dns_server_1`.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --dns-server dns_server_1
    --dns-server dns_server_2
```

To reset the DNS servers on a VCH to the default, set the `vic-machine configure --dns-server` option to `""` .

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --dns-server ""
```

NOTE: The `vic-machine configure` command does not include options to set a static IP address on a VCH that uses DHCP.

Configure Container Network Settings

If containers that run in a VCH require a dedicated network for external communication, you can add one or more container networks to the VCH by using the `vic-machine configure --container-network` options. You can specify `--container-network` multiple times to add multiple container networks.

The `vic-machine configure --container-network` options function in the same way as the equivalent `vic-machine create` options. For information about the `vic-machine create` container network options, Configure Container Networks.

This example adds a new container network to a VCH. It designates a port group named `vic-containers` for use by container VMs, gives the container network the name `vic-container-network` for use by Docker, specifies the gateway, two DNS servers, and a range of IP addresses on the container network for container VMs to use.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --container-network vic-containers:vic-container-network
    --container-network-gateway vic-containers:gateway_ip_address/24
    --container-network-ip-range vic-containers:192.168.100.0/24
    --container-network-dns vic-containers:dns1_ip_address
    --container-network-dns vic-containers:dns2_ip_address
```

If you are adding container networks to a VCH that already includes one or more container networks, you must also specify each existing container network in separate instances of the `--container-network` options. This example adds a new DHCP container network named `vic-containers-2` to the VCH from the example above.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --container-network vic-containers:vic-container-network
    --container-network-gateway vic-containers:gateway_ip_address/24
    --container-network-ip-range vic-containers:192.168.100.0/24
```

```
    --container-network-dns vic-containers:dns1_ip_address
    --container-network-dns vic-containers:dns2_ip_address
    --container-network vic-containers-2:vic-container-network-2
```

You can also configure the trust level of the container network firewall by setting the `--container-network-firewall` option. This example opens the firewall for outbound connections on the two container networks from the preceding examples.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --container-network vic-containers:vic-container-network
    --container-network-gateway vic-containers:gateway_ip_address/24
    --container-network-ip-range vic-containers:192.168.100.0/24
    --container-network-dns vic-containers:dns1_ip_address
    --container-network-dns vic-containers:dns2_ip_address
    --container-network-firewall vic-containers:outbound
    --container-network vic-containers-2:vic-container-network-2
    --container-network-firewall vic-containers-2:outbound
```

For information about the trust levels that you can set, see `--container-network-firewall` in VCH Deployment Options.

You cannot modify or delete an existing container network on a VCH.

Add, Configure, or Remove Proxy Servers

If access to the Internet or to private registry servers changes to pass through a proxy server, you configure a VCH to use the new proxy server by using the `vic-machine configure --https-proxy` and `--http-proxy` options. You also use the `vic-machine configure --https-proxy` and `--http-proxy` options if an existing proxy server changes.

The `vic-machine configure --https-proxy` and `--http-proxy` options function in the same way as the equivalent `vic-machine create` options. For information about the `vic-machine create --https-proxy` and `--http-proxy` options, see Configure VCHs to Use Proxy Servers.

This example configures a VCH to use a new HTTPS proxy server.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --https-proxy https://new_proxy_server_address:port
```

To remove a proxy server from a VCH, set the `vic-machine configure --https-proxy` or `--http-proxy` options to `""`.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --https-proxy ""
```

Configure Debug Mode

To enable or disable debug mode on a VCH, you use the `vic-machine configure --debug` option. You can also use `vic-machine configure --debug` to increase or decrease the level of debugging on a VCH that is already running in debug mode.

The `vic-machine configure --debug` option functions in the same way as the equivalent `vic-machine create --debug` option. For information about the `vic-machine create --debug` option, see `--debug` in VCH Deployment Options. By default, `vic-machine create` deploys VCHs with debugging level 0.

This example increases the level of debugging to level 3, either on a VCH that is running with a lower level of debugging, or on a VCH that is not running in debug mode.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --debug 3
```

This example sets the `--debug` option to 0, to disable debug mode on a VCH.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --debug 0
```

Configure CPU and Memory Allocations

If a VCH requires more resources, or if it consumes too many resources, you can configure CPU and memory allocations on the VCH vApp by using the different `vic-machine configure --memory` and `--cpu` options.

The `vic-machine configure` options for memory and CPU allocations function in the same way as the equivalent `vic-machine create` options. For information about the `vic-machine create --cpu` and `--memory` options, see General Deployment Options in VCH Deployment Options. For information about the memory and CPU reservation and shares options, see Advanced Resource Management Options.

This example configures a VCH to impose memory and CPU reservations, limits, and shares.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --memory 1024
    --memory-reservation 1024
    --memory-shares low
    --cpu 1024
    --cpu-reservation 1024
    --cpu-shares low
```

NOTE: If you set limits on memory and CPU usage that are too low, the `vic-machine configure` operation might fail because it is unable to restart the VCH.

This example removes all limitations on memory and CPU use from a VCH.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --memory 0
    --memory-reservation 0
    --memory-shares normal
    --cpu 0
    --cpu-reservation 0
    --cpu-shares normal
```

Reset Upgrade or Configuration Progress

If an attempt to upgrade or configure a VCH was interrupted before it could complete successfully, any further attempts to run `vic-machine upgrade` or `vic-machine configure` fail with the error `another upgrade/configure operation is in progress`. This happens because `vic-machine upgrade` and `vic-machine configure` set an `UpdateInProgress` flag on the VCH endpoint VM that prevents other operations on that VCH while the upgrade or configuration operation is ongoing. If an upgrade or configuration operation is interrupted before it completes, this flag persists on the VCH indefinitely.

To clear the flag so that you can attempt further `vic-machine upgrade` or `vic-machine configure` operations, run `vic-machine configure` with the `--reset-progress` option.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --thumbprint certificate_thumbprint
    --id vch_id
    --reset-progress
```

IMPORTANT: Before you run `vic-machine configure --reset-progress`, check in Recent Tasks in the vSphere Client that there are indeed no update or configuration operations in progress on the VCH endoint VM.

Delete Virtual Container Hosts

You delete virtual container hosts (VCHs) by using the `vic-machine delete` command.

The `vic-machine delete` includes one option in addition to the Common `vic-machine` Options, `--force` .

- You must specify the username and optionally the password, either in the `--target` option or separately in the `--user` and `--password` options.
- If the VCH has a name other than the default name, `virtual-container-host` , you must specify the `--name` or `--id` option.
- If multiple compute resources exist in the datacenter, you must specify the `--compute-resource` or `--id` option.
- Specifying the `--force` option forces `vic-machine delete` to ignore warnings and continue with the deletion of a VCH. Any running container VMs and any volume stores associated with the VCH are deleted. Errors such as an incorrect compute resource still cause the deletion to fail.

    - If you do not specify `--force` and the VCH contains running container VMs, the deletion fails with a warning.
    - If you do not specify `--force` and the VCH has volume stores, the deletion of the VCH succeeds without deleting the volume stores. The list of volume stores appears in the `vic-machine delete` success message for reference and optional manual removal.
- If your vSphere environment uses untrusted, self-signed certificates, you must specify the thumbprint of the vCenter Server instance or ESXi host in the `--thumbprint` option. For information about how to obtain the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

    Use upper-case letters and colon delimitation in the thumbprint. Do not use space delimitation.

When you delete a VCH that uses TLS authentication with trusted Certificate Authority (CA) certificates, `vic-machine delete` does not delete the certificates or the certificate folder, even if you specify the `--force` option. Because `vic-machine delete` does not delete the certificates, you can delete VCHs and create new ones that reuse the same certificates. This is useful if you have already distributed the client certificates for VCHs that you need to recreate.

The `vic-machine delete` command does not modify the firewall on ESXi hosts. If you do not need to deploy or run further VCHs on the ESXi host or cluster after you have deleted VCHs, run `vic-machine update firewall --deny` to close port 2377 on the host or hosts.

Example

The following example includes the options required to remove a VCH from a simple vCenter Server environment.

```
$ vic-machine-operating_system delete
--target vcenter_server_username:password@vcenter_server_address
--thumbprint certificate_thumbprint
--name vch_name
```

If the delete operation fails with a message about container VMs that are powered on, run `docker stop` on the containers and run `vic-machine delete` again. Alternatively, run `vic-machine delete` with the `--force` option.

CAUTION Running `vic-machine delete` with the `--force` option removes all running container VMs that the VCH manages, as well as any associated volumes and volume stores. It is not recommended to use the `--force` option to remove running containers.

```
$ vic-machine-operating_system delete
--target vcenter_server_username:password@vcenter_server_address
--thumbprint certificate_thumbprint
--name vch_name
--force
```

If your vSphere environment uses untrusted, self-signed certificates, running `vic-machine delete` with the `--force` option allows you to omit the `--thumbprint` option.

CAUTION: It is not recommended to use `--force` to bypass thumbprint verification in production environments. Using `--force` in this way exposes VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials.

```
$ vic-machine-operating_system delete
--target vcenter_server_username:password@vcenter_server_address
--name vch_name
--force
```

Debug Virtual Container Hosts

By default, all shell access to the virtual container host (VCH) endpoint VM is disabled. Login shells for all users are set to `/bin/false` . The `vic-machine` utility provides a `debug` command that allows you to enable shell access to the VCH endpoint VM, either by using the VM console or via SSH.

In addition to the Common `vic-machine` Options, `vic-machine debug` provides the `--rootpw` , `--enable-ssh` and `--authorized-key` options, which are described in the following sections.

- Enable Shell Access to the VCH Endpoint VM
- Authorize SSH Access to the VCH Endpoint VM

NOTE: Do not confuse the `vic-machine debug` command with the `vic-machine create --debug` or `vic-machine configure --debug` options. The `vic-machine debug` command allows you to log into and debug a VCH endpoint VM that you have already deployed. The `vic-machine create --debug` option deploys a new VCH that has increased levels of logging and other modifications, to allow you to debug the environment in which you deploy VCHs. For information about the `vic-machine create --debug` option, see the section on `--debug` in VCH Deployment Options.

Enable shell access to the VCH Endpoint VM

You can use the `vic-machine debug` command to enable shell access to a virtual container host (VCH) endpoint VM by setting a root password on the VM. Setting a root password enables access to the VCH endpoint VM via the VM console only. If you require SSH access to the VCH endoint VM, rather than just shell access, see Authorize SSH Access to the VCH Endpoint VM.

IMPORTANT: Any changes that you make to a VCH by using `vic-machine debug` are non-persistent and are discarded if the VCH endpoint VM reboots.

In addition to the Common `vic-machine` Options, `vic-machine debug` provides the `--rootpw` , `--enable-ssh` and `--authorized-key` options.

- You must specify the vSphere target and its credentials, either in the `--target` option or separately in the `--user` and `--password` options.

  The credentials that you provide must have the following privilege on the endpoint VM:

  ```
  Virtual machine.Guest Operations.Guest Operation Program Execution
  ```

- You must specify the ID or name of the VCH to debug.

- You might need to provide the thumbprint of the vCenter Server or ESXi host certificate. Use upper-case letters and colon delimitation in the thumbprint. Do not use space delimitation.
- You enable shell access by specifying a password for the root user on the VCH endpoint VM in the `--rootpw` option. Setting a password on the VCH allows you to access the VCH by using the VM console. If you also set the `--enable-ssh` option, you can use this password to connect to the VCH by using SSH. Wrap the password in quotes if it includes shell characters such as `$` , `!` or `%` .

  ```
  --rootpw 'new_p@ssword'
  ```

- When you use the password to log in to a VCH, you see the message that the password will expire in 0 days. To obtain a longer expiration period, use the Linux `passwd` command in the endpoint VM to set a new password. If the password expires, the VCH does not revert to the default security configuration from before you ran `vic-machine debug` . If you attempt to log in using an interactive password via the terminal or SSH, you see a prompt to change the password. If you are using an SSH key, you cannot log in until you either change the password or run `vic-machine debug` again.

Example

This example sets a password to allow shell access to the VCH.

```
$ vic-machine-operating_system debug
    --target vcenter_server_or_esxi_host_address
    --user vcenter_server_or_esxi_host_username
```

```
    --password vcenter_server_or_esxi_host_password
    --id vch_id
    --thumbprint certificate_thumbprint
    --rootpw 'new_p@ssword'
```

Output

The output of the `vic-machine debug` command includes confirmation that SSH access is enabled:

```
### Configuring VCH for debug ####
[...]
SSH to appliance:
ssh root@vch_address
[...]
Completed successfully
```

Authorize SSH Access to the VCH Endpoint VM

You can use the `vic-machine debug` command to enable shell access to a virtual container host (VCH) endpoint VM by setting a root password on the VM. Setting a root password enables access to the VCH endpoint VM via the VM console. If you authorize SSH access to the VCH endpoint VM, you can edit system configuration files that you cannot edit by running `vic-machine` commands. You can also use `debug` to authorize SSH access to the VCH endpoint VM. You can optionally upload a key file for public key authentication when accessing the endpoint VM by using SSH.

IMPORTANT: Any changes that you make to a VCH by using `vic-machine debug` are non-persistent and are discarded if the VCH endpoint VM reboots.

- You must specify the vSphere target and its credentials, either in the `--target` option or separately in the `--user` and `--password` options.

  The credentials that you provide must have the following privilege on the endpoint VM:

  ```
  Virtual machine.Guest Operations.Guest Operation Program Execution
  ```

- You must specify the ID or name of the VCH to debug.
- You might need to provide the thumbprint of the vCenter Server or ESXi host certificate. Use upper-case letters and colon delimitation in the thumbprint. Do not use space delimitation.
- To enable SSH access, you mush enable shell access by specifying the `--rootpw` option. Wrap the password in quotes if it includes shell characters such as `$`, `!` or `%`.
- You authorize SSH access by specifying `--enable-ssh`. The `sshd` service runs until the VCH endpoint VM reboots. The `--enable-ssh` option takes no arguments.
- If you have already enabled SSH access but the password that you set has expired, and you then rerun `--enable-ssh` without specifying `--rootpw`, the password expiry is set to 1 day in the future and the password is preserved.
- Optionally, you can specify the `--authorized-key` option to upload a public key file to `/root/.ssh/authorized_keys` folder in the endpoint VM. Include the name of the `*.pub` file in the path.

  ```
  --authorized-key path_to_public_key_file/key_file.pub
  ```

Example

This example authorizes SSH access and provides a public key file.

```
$ vic-machine-operating_system debug
    --target vcenter_server_or_esxi_host_address
    --user vcenter_server_or_esxi_host_username
    --password vcenter_server_or_esxi_host_password
    --id vch_id
```

```
        --thumbprint certificate_thumbprint
        --enable-ssh
        --rootpw 'new_p@ssword'
        --authorized-key path_to_public_key_file/key_file.pub
```

Output

The output of the `vic-machine debug` command includes confirmation that SSH access is enabled:

```
### Configuring VCH for debug ####
[...]
SSH to appliance:
ssh root@vch_address
[...]
Completed successfully
```

VCH Administration Portal

vSphere Integrated Containers Engine provides a Web-based administration portal for virtual container hosts (VCHs), called VCH Admin.

If you deployed the VCH with `--no-tls` or `--no-tlsverify` , you log in to VCH Admin by specifying the username and password of the ESXi host or vCenter Server on which you deployed the VCH. If you deployed the VCH with client and server authentication by using `--tls-cname` or by specifying a static IP address on the client network, you can use the generated `*.pfx` certificate to authenticate with the VCH Admin portal. For information about using the `*.pfx` certificate to log into VCH admin, see Browser-Based Certificate Login and Command Line Certificate Login.

You access the VCH Admin portal in the following places:

- In the HTML5 vSphere Client, go to **Home** > **vSphere Integrated Containers** > **vSphere Integrated Containers** > **Virtual Container Hosts** and click the link to the VCH Admin portal.
- In the HTML5 vSphere Client or Flex-based vSphere Web Client, got to **Hosts and Clusters**, select a VCH endpoint VM, and click the link to the VCH Admin portal in the **Summary** tab.
- Copy the address of the VCH Admin portal from the output of `vic-machine create` or `vic-machine inspect` .

After you log in, the VCH Admin portal displays information about the VCH and the environment in which is running:

- Status information about the VCH, registry and Internet connections, firewall configuration, and license. For information about these statuses and how to remedy error states, see the VCH Status Reference.
- The address of the Docker endpoint.
- The system time of the VCH. This is useful to know because clock skews between VCHs and client systems can cause TLS authentication to fail. For information about clock skews, see Connections Fail with Certificate Errors when Using Full TLS Authentication with Trusted Certificates.
- The remaining capacity of the datastore that you designated as the image store. If the VCH is unable to connect to vSphere, the datastore information is not displayed.
- Live logs and log bundles for different aspects of the VCH. For information about the logs, see Access vSphere Integrated Containers Engine Log Bundles.

If you see a certificate error when you attempt to log in to the VCH Administration Portal, see Browser Rejects Certificates with `ERR_CERT_INVALID` Error.

Browser-Based Certificate Login

If you deployed the VCH with client and server authentication by using `--tls-cname` or by specifying a static IP address on the client network, you can use browser-based certificate authentication to access the VCH Admin Portal. In this way, you do not need to provide the vSphere credentials each time that you log in to VCH Admin.

Prerequisites

- You deployed a VCH with `--tls-cname` or a static IP address for the VCH on the client network.
- Use Firefox. Currently, this feature is only supported with Firefox.
- Locate the file named `cert.pfx` on the system on which you ran `vic-machine create`. The `cert.pfx` is located in either of the following locations:

    - In the folder with the same name as the VCH, in the directory from which you ran `vic-machine create`.
    - In a folder that you specified in the `vic-machine create --tls-cert-path` option.

Procedure

1. In Firefox, select `Tools` > `Options` and select `Advanced`.
2. Click `View Certificates`.
3. Click `Import`.
4. Browse to the `cert.pfx` file and click `Open`.
5. Click `OK`.

    Do not enter a password when prompted.

Result

You see a message stating that the certificate was successfully installed. With the VCH certificate installed in your browser, you can navigate to https://*vch_address*:2378/ or to one of the log pages without having to enter the vSphere credentials.

Command Line Certificate Login

You can use certificate-based authentication with tools such as `curl` or `wget` to access the VCH Admin log server.

With TLS Client Authentication

-If you deployed the VCH with client authentication by using `--tls-cname` or by specifying a static IP address on the client network, you can point `curl` to the `cert.pem` and `key.pem` files for the VCH. The following example authenticates connections to the `port-layer.log` file.

```
curl https://vch_address:2378/logs/port-layer.log
--key ./cert_folder/key.pem
--certificate ./cert_folder/cert.pem
```

NOTE: If your certificates are self-signed, you might also need to specify the `curl -k` flag.

In the example above, *cert_folder* is either of the following locations:

- The folder with the same name as the VCH, in the directory from which you ran `vic-machine create`.
- A folder that you specified in the `vic-machine create --tls-cert-path` option.

Without Client Authentication

If you deployed the VCH without client authentication by using either of `--no-tls` or `--no-tlsverify`, you can use curl to access the logs but you must first authenticate connections to VCH Admin by using the vSphere username and password.

1. Log in to VCH Admin to gather an authentication cookie for subsequent access:

```
curl -sk https://vch_address:2378/authentication
-XPOST -F username=vsphere_username
-F password=vsphere_password
-D cookies_file
```

2. Use the cookie from Step 1 in a `curl` command to access the logs.

```
curl -sk https://vch_address:2378/logs/port-layer.log
-b cookies_file
```

VCH Admin Status Reference

The Web-based administration portal for virtual container hosts (VCHs), VCH Admin, presents status information about a VCH.

If the vSphere environment in which you are deploying a VCH does not meet the requirements, the deployment does not succeed. However, a successfully deployed VCH can stop functioning if the vSphere environment changes after the deployment. If environment changes adversely affect the VCH, the status of the affected component changes from green to yellow.

Virtual Container Host (VCH)

VCH Admin checks the status of the processes that the VCH runs:

- The port layer server, that presents an API of low-level container primitive operations, and implements those container operations via the vSphere APIs.
- VCH Admin server, that runs the VCH Admin portal.
- The vSphere Integrated Containers Engine initialization service and watchdog service for the other components.
- The Docker engine server, that exposes the Docker API and semantics, translating those composite operations into port layer primitives.

Error

- The **VCH** status is yellow.
- The **VCH** status is yellow and an error message informs you that the VCH cannot connect to vSphere.

Cause

- One or more of the VCH processes is not running correctly, or the VCH is unable to connect to vSphere.
- The management network connection is down and the VCH endpoint VM cannot connect to vSphere.

Solution

1. (Optional) If you see the error that the VCH is unable to connect to vSphere, check the VCH management network.
2. In the VCH Admin portal for the VCH, click the link for the **VCH Admin Server** log.
3. Search the log for references to the different VCH processes.

   The different processes are identified in the log by the following names:

   - `port-layer-server`
   - `vicadmin`
   - `vic-init`
   - `docker-engine-server`
4. Identify the process or processes that are not running correctly and attempt to remediate the issues as required.

Registry and Internet Connectivity

VCH Admin checks connectivity on the public network by attempting to connect from the VCH to docker.io and google.com. VCH Admin only checks the public network connection. It does not check other networks, for example the bridge, management, client, or container networks.

Error

The **Registry and Internet Connectivity** status is yellow.

Cause

The public network connection is down.

Solution

Check the **VCH Admin Server** log for references to network issues. Use the vSphere Web Client to remediate the management network issues as required.

Firewall

VCH Admin checks that the firewall is correctly configured on an ESXi host on which the VCH is running. If the VCH is running in a cluster, VCH Admin checks the firewall configuration on all of the hosts in the cluster.

Error

- The **Firewall** status is unavailable.
- The **Firewall** status is yellow and shows the error `Firewall must permit 2377/tcp outbound to use VIC`.

Cause

- The management network connection is down and the VCH endpoint VM cannot connect to vSphere.
- The firewall on the ESXi host on which the VCH is running no longer allows outbound connections on port 2377.

  - The firewall was switched off when the VCH was deployed. The firewall has been switched on since the deployment of the VCH.
  - A firewall ruleset was applied manually to the ESXi host to allow outbound connections on port 2377. The ESXi host has been rebooted since the deployment of the VCH. Firewall rulesets are not retained when an ESXi host reboots.

Solution

- If the **Firewall** status is unavailable:
  - Check the **VCH Admin Server** log for references to network issues.
  - Use the vSphere Web Client to remediate the management network issues as required.

- If you see the error about port 2377, run the `vic-machine update firewall` command on the ESXi host or hosts to allow outbound connections on port 2377. For information about how to run `vic-machine update firewall`, see Open the Required Ports on ESXi Hosts.

License

VCH Admin checks that the ESXi hosts on which you deploy VCHs have the appropriate licenses.

Error

- The **License** status is yellow and shows the error `License does not meet minimum requirements to use VIC`.
- The **License** status is unavailable.

Cause

- The license for the ESXi host or for one or more of the hosts in a vCenter Server cluster on which the VCH is deployed has been removed, downgraded, or has expired since the deployment of the VCH.
- The management network is down, or the VCH endpoint VM is unable to connect to vSphere.

Solution

- If the license does not meet the requirements:
  - If the VCH is running on an ESXi host that is not managed by vCenter Server, replace the ESXi host license with a valid vSphere Enterprise license.
  - If the VCH is running on a standalone ESXi host in vCenter Server, replace the ESXi host license with a valid vSphere Enterprise Plus license.
  - If the VCH is running in a vCenter Server cluster, check that all of the hosts in the cluster have a valid vSphere Enterprise Plus license, and replace any licenses that have been removed, downgraded, or have expired.
- If the **License** status is unavailable:
  - Check the **VCH Admin Server** log for references to network issues.
  - Use the vSphere Web Client to remediate the management network issues as required.

Upgrading vSphere Integrated Containers

You can upgrade vSphere Integrated Containers from version 1.1.x to 1.2.x, or from 1.2.x to a later 1.2.y update release.

You cannot upgrade any of the components of vSphere Integrated Containers 1.0, namely vSphere Integrated Containers Engine 0.8 and vSphere Integrated Containers Registry 0.5, to 1.2.x. Similarly, you cannot upgrade an instance of vSphere Integrated Containers Management Portal (Admiral) that predates version 1.1.0 to version 1.2.x.

You upgrade vSphere Integrated Containers in three stages:

Upgrade the vSphere Integrated Containers Appliance

Upgrading the appliance upgrades both vSphere Integrated Containers Registry and vSphere Integrated Containers Management portal.

- For information about the data that migrates during upgrade, see Data That Migrates During vSphere Integrated Containers Appliance Upgrade.
- For information about how to prepare for upgrade, see Tasks to Perform Before Upgrading the vSphere Integrated Containers Appliance.
- For information about upgrading the appliance, see Upgrade the vSphere Integrated Containers Appliance.

Upgrade Virtual Container Hosts

After you have upgraded the appliance, you can download the new version of the vSphere Integrated Containers Engine bundle. To upgrade vSphere Integrated Containers Engine, you upgrade the virtual container hosts (VCHs) individually.

For information about upgrading VCHs, see Upgrade Virtual Container Hosts.

Upgrade the vSphere Client Plug-Ins

After you have upgraded the appliance and downloaded the vSphere Integrated Containers Engine bundle, you can upgrade the HTML5 vSphere Client plug-in.

For information about upgrading the vSphere Client plug-in, see the topic that corresponds to the type of vCenter Server that you use.

- Upgrade the Plug-Ins on vCenter Server for Windows
- Upgrade the Plug-Ins on vCenter Server Appliance

Data That Migrates During vSphere Integrated Containers Appliance Upgrade

To upgrade an older version of the appliance, you deploy a new appliance instance. The appliance upgrade process migrates vSphere Integrated Containers Registry and Management Portal data from the older appliance to the new appliance. Due to changes in the data model, user identity management, and the merging of the user interfaces between versions 1.1.x and 1.2.x of vSphere Integrated Containers Registry and Management Portal, not all data can migrate when you upgrade the vSphere Integrated Containers appliance from 1.1.x to 1.2.x.

vSphere Integrated Containers Management Portal Data

Type of data	Migrated?	More Information
Virtual Container Hosts (VCHs)	Yes	VCHs that you added in vSphere Integrated Containers Management Portal 1.1.x migrate to the new version. VCH certificates and credentials also migrate. Upgrading the appliance does not upgrade the VCHs themselves. For information about upgrading VCHs, see Upgrade Virtual Container Hosts.
Placement policies	No	You cannot modify placements and placement zones in vSphere Integrated Containers 1.2.x. This version manages placements internally when you create or modify projects.
Projects	Yes	Projects that you created in vSphere Integrated Containers Management Portal 1.1.x appear in the **Projects** view of the upgraded interface alongside projects that you created in vSphere Integrated Containers Registry 1.1.x. If you renamed the default project in 1.1.x, the project name reverts to `default-project` after the upgrade. vSphere Integrated Containers Management Portal project names must respect the rules described in Tasks to Perform Before Upgrading the vSphere Integrated Containers Appliance.
Registries	Yes	Registries that you added in version 1.1.x migrate to the new version. Registry certificates and credentials also migrate.
Applications	Yes	Applications that you created in version 1.1.x migrate to the new version.
Containers and templates	Yes	Containers and templates that you created in version 1.1.x migrate to the new version. Containers that are running in VCHs also migrate.
Networks and volumes	Yes	Networks and volumes that you created in version 1.1.x migrate to the new version.

vSphere Integrated Containers Registry Data

Type of data	Migrated?	More Information
System	Yes	If you changed the login token expiration period or deselected the checkbox to verify remote registry certificates during replication, these settings persist

Configuration		after upgrade.
Email Settings	No	vSphere Integrated Containers 1.2.x does not implement email notifications.
Projects	Yes	All existing projects migrate.
Replication endpoints and rules	Yes	All replication endpoints and rules migrate. If the replication endpoints are vSphere Integrated Containers Registry instances, upgrade those instances to 1.2.x. Replication of images from the 1.2.x registry instance to the 1.1.x replication endpoint still functions, but it is recommended that you upgrade the target registry.
Users	No	vSphere Integrated Containers 1.2.x imports users and user groups from the Platform Services Controller. In addition, the roles that you assign to users are different in version 1.2.x. If vSphere Integrated Containers Registry 1.1.x uses database authentication, local registry users cannot migrate to the Platform Services Controller. You can recreate these users in the new version as local users in the Platform Services Controller. If vSphere Integrated Containers Registry 1.1.x uses LDAP authentication, in the new version you must add the appropriate LDAP server as an identity source in the Platform Services Controller. In either case, you must import the users from the Platform Services Controller into vSphere Integrated Containers Management Portal after the upgrade.
Image repositories	Yes	All of the images in a project migrate.
Logs	Yes	Project logs migrate.
Notary	Yes and no	If you used Docker Content Trust, or Notary, to sign images, the signatures include the address of the registry. If the address of the appliance, and therefore of the registry, changed during the upgrade, you must resign the images with the new address. If the address did not change, no action is required.

Tasks to Perform Before Upgrading the vSphere Integrated Containers Appliance

To ensure a successful upgrade, you must perform several tasks before upgrading the vSphere Integrated Containers appliance. These pre-upgrade tasks are necessary due to differences in implementation between versions 1.1 and 1.2 of the vSphere Integrated Containers Registry and Management Portal, in particular the merging of the user interfaces and the transition to the Platform Services Controller for identity management.

- Make sure that the previous version of the vSphere Integrated Containers appliance allows SSH connections.

    1. Select the appliance VM in the Hosts and Clusters view of the vSphere client
    2. Select **Edit Settings** > **vApp Options**.
    3. Expand **Appliance Security** and make sure that **Permit Root Login** is set to `True`.
    4. If **Permit Root Login** is set to `False`, power off the appliance VM, and edit the settings to enable it.

- If the previous version of vSphere Integrated Containers Registry uses local database authentication for identity management, make a record of all of the users that exist in the database. These users cannot migrate to the Platform Services Controller, so you must recreate them after the upgrade.
- If the previous version of vSphere Integrated Containers Management Portal includes projects in **Management** > **Policies** > **Placements**, project names must respect certain rules for the upgrade to succeed. Project names can only include the following characters:

    - Alpha-numeric characters
    - Lower-case letters
    - Periods, dashes, and underscores
    - Maximum length of 254 characters

        If project names include upper-case characters or special characters other than periods, dashes, and underscores, or exceed 254 characters, you must edit them before you perform the upgrade.

- Ensure that all vCenter Server instances and ESXi hosts in the environment in which you are deploying the appliance have network time protocol (NTP) running. Running NTP prevents problems arising from clock skew between the vSphere Integrated Containers appliance, virtual container hosts, and the vSphere infrastructure.
- Back up the appliance by using your usual backup tools.

Upgrade the vSphere Integrated Containers Appliance

If you deployed a 1.1.x version of the vSphere Integrated Containers appliance, you can upgrade your existing installation to 1.2.x. You can also upgrade a 1.2.x appliance to a later 1.2.y update release.

Upgrading the vSphere Integrated Containers appliance upgrades vSphere Integrated Containers Registry and vSphere Integrated Containers Management Portal. For information about the vSphere Integrated Containers Registry and Management Portal data that migrates during upgrade, see Data That Migrates During vSphere Integrated Containers Appliance Upgrade.

NOTE: This topic reflects changes to the upgrade procedure that were introduced in vSphere Integrated Containers 1.2.1. Do not attempt to upgrade to version 1.2.0.

Prerequisites

- You have completed the pre-upgrade tasks listed in Tasks to Perform Before Upgrading the vSphere Integrated Containers Appliance.
- Deploy the new version of the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.

  IMPORTANT:

  - Do not disable SSH access to the new appliance. You require SSH access to the appliance during the upgrade procedure.
  - When the OVA deployment finishes, do not power on the new appliance. Attempting to perform the upgrade procedure on a new appliance that you have already powered on and initialized causes vSphere Integrated Containers Management Portal and Registry not to function correctly and might result in data loss.
- Use the Flex-based vSphere Web Client to deploy the appliance. You cannot deploy OVA files from the HTML5 vSphere Client or from the legacy Windows client.

- Deploy the appliance to the same vCenter Server instance as the one on which the previous version is running, or to a vCenter Server instance that is managed by the same Platform Services Controller.
- Log in to the vSphere Client for the vCenter Server instance on which the previous version is running and on which you deployed the new version.

Procedure

1. Shut down the older vSphere Integrated Containers appliance by selecting **Shut Down Guest OS**.

   IMPORTANT: Do not select **Power Off**.

2. Right-click the older vSphere Integrated Containers appliance, and select **Edit Settings**.
3. Hover your pointer over **Hard disk 2**, click the **Remove** button, and click **OK**.

   - Hard disk 2 is the larger of the two disks.
   - **IMPORTANT**: Do not check the **Delete files from this datastore** checkbox.
4. Right-click the new vSphere Integrated Containers appliance, and select **Edit Settings**.

5. Hover your pointer over **Hard disk 2**, click the **Remove** button, and check the **Delete files from this datastore**

checkbox, and click **OK**.

6. In the **Storage** view of the vSphere Client, move the disk from the previous appliance into the datastore folder of the new appliance.

    i. Navigate to the VDMK files of the previous appliance.
    ii. Select the VMDK file with the file name that ends in `_1`
    iii. Click **Move to...**, and move it into the datastore folder of the new appliance.

7. In the **Hosts and Clusters** view of the vSphere Client, right-click the new appliance and select **Edit Settings** again to add the disk from the old appliance to the new appliance.

    ◦ Flex-based vSphere Web Client: Click the **New device** drop-down menu, select **Existing Hard Disk**, and click **Add**.
    ◦ HTML5 vSphere Client: Click the **Add New Device** button and select **Existing Hard Disk**.

8. Navigate to the datastore folder into which you moved the disk, select the VMDK file from the previous appliance, and click **OK**.

9. Expand **New Hard Disk** and make sure that the Virtual Device Node for the disk is set to **SCSI(0:1)**, then click **OK**.

10. Power on the new vSphere Integrated Containers appliance and note its address.

IMPORTANT: Do not go to the Getting Started page of the appliance. Logging in to the Getting Started page for the first time initializes the appliance. Initialization is only applicable to new installations and causes upgraded appliances not to function correctly.

11. Use SSH to connect to the new appliance as root user.

```
$ ssh root@new_vic_appliance_address
```

When prompted for the password, enter the appliance password that you specified when you deployed the new version of the appliance.

12. Navigate to the upgrade script and run it.

```
$ cd /etc/vmware/upgrade
```

```
$ ./upgrade.sh
```

As the script runs, respond to the prompts to provide the following information:

    i. Enter the address of the vCenter Server instance on which you deployed the new appliance.
    ii. Enter the Single Sign-On user name and password of a vSphere administrator account. The script requires these credentials to register the new version of vSphere Integrated Containers with the VMware Platform Services Controller.
    iii. If vCenter Server is managed by an external Platform Services Controller, enter the FQDN of the Platform Services Controller. If vCenter Server is managed by an embedded Platform Services Controller, press Enter without entering anything.
    iv. If vCenter Server is managed by an external Platform Services Controller, enter the administrator domain for

the Platform Services Controller. If vCenter Server is managed by an embedded Platform Services Controller, press Enter without entering anything.

    v. Verify that the upgrade script has detected your upgrade path correctly.

       If the upgrade script detects that you are performing an upgrade from 1.1.x to 1.2.y, it migrates 1.1.x data to the new data model for 1.2.y. If it detects an upgrade from 1.2.x to 1.2.y, the data model is the same, so no data migration is required.

- If the script detects your upgrade path correctly, enter `y` to proceed with the upgrade.
- If the upgrade script detects the upgrade path incorrectly, for example if it asks you to proceed with an upgrade from 1.1.x to 1.2.y when you are upgrading 1.2.x to 1.2.y, enter `n` to abort the upgrade and follow the procedure in Troubleshooting vSphere Integrated Containers Appliance Upgrade.

13. When you see confirmation that the upgrade has completed successfully, go to http://*vic_appliance_address*, click the link to **Go to the vSphere Integrated Containers Management Portal**, and use vCenter Server Single Sign-On credentials to log in.

    ◦ In the **Home** tab of the vSphere Integrated Containers Management Portal, check that all existing applications, containers, networks, volumes, and virtual container hosts have migrated successfully.
    ◦ In the **Administration** tab, check that projects, registries, repositories, and replication configurations have migrated successfully.

14. When you have confirmed that the upgrade succeeded, delete the appliance VM for the previous version from the vCenter Server inventory.

What to Do Next

- If, in the previous version, you configured vSphere Integrated Containers Registry instances as replication endpoints, upgrade those registry instances. Replication of images from the 1.2.x registry instance to the 1.1.x replication endpoint still functions, but it is recommended that you upgrade the target registry.
- Due to changes in the data model, user identity management, and the merging of the user interfaces for vSphere Integrated Containers Registry and Management Portal in version 1.2.x, you must perform some manual tasks after you upgrade the appliance. For information about post-upgrade tasks, see Tasks to Perform After Upgrading the vSphere Integrated Containers Appliance.
- Download the vSphere Integrated Containers Engine bundle and upgrade your VCHs. For information about upgrading VCHs, see Upgrade Virtual Container Hosts.
- Upgrade the vSphere Integrated Containers plug-in for the HTML5 vSphere Client. For information about upgrading the vSphere Client plug-in, see

    ◦ Upgrade the HTML5 vSphere Client Plug-In on vCenter Server for Windows
    ◦ Upgrade the HTML5 vSphere Client Plug-In on a vCenter Server Appliance

Tasks to Perform After Upgrading the vSphere Integrated Containers Appliance

Due to changes in the data model, user identity management, and the merging of the user interfaces for vSphere Integrated Containers Registry and Management Portal in version 1.2.x, you must perform some manual tasks after you upgrade the vSphere Integrated Containers appliance.

- Import users from the Platform Services Controller into vSphere Integrated Containers Management Portal. For information about users and roles in this version of vSphere Integrated Containers, see the following topics:

    - User Authentication
    - Add Cloud Administrators
    - Add Viewers, Developers, or DevOps Administrators to Projects

- If, in version 1.1.x of the appliance, you manually added the vSphere Integrated Containers Registry instance to vSphere Integrated Containers Management Portal, and if the address of the appliance changed during the upgrade, two instances of vSphere Integrated Containers Registry appear in the **Administration** > **Registries** > **Source Registries** view in the new version. The registry named `default-vic-registry` is the new registry instance that is running in the new appliance, that is automatically registered with the management portal. Data from the registry that was running in the previous appliance has migrated to this instance. A registry instance with the name and address from the old, and now defunct, appliance is present in the list of registries. Delete this instance from the list.

- If you added the same virtual container host (VCH) to more than one placement zone in vSphere Integrated Containers Management Portal 1.1.x, multiple instances of that VCH appear in the **Home** > **Infrastructure** > **Container Hosts** view in the new version. Delete the duplicates.

Upgrade Virtual Container Hosts

You upgrade virtual container hosts (VCHs) by downloading a new version of vSphere Integrated Containers Engine and running the `vic-machine upgrade` command.

You can use `vic-machine upgrade` to upgrade VCHs to newer versions. You can run `vic-machine upgrade` on VCHs that are either running or powered off. When you upgrade a running VCH, the VCH goes temporarily offline, but container workloads continue as normal during the upgrade process. Upgrading a VCH does not affect any mapped container networks that you defined by setting the `vic-machine create --container-network` option. The following operations are not available during upgrade:

- You cannot access container logs
- You cannot attach to a container
- NAT based port forwarding is unavailable

IMPORTANT: Upgrading a VCH does not upgrade any existing container VMs that the VCH manages. For container VMs to boot from the latest version of `bootstrap.iso`, container developers must recreate them.

For descriptions of the options that `vic-machine upgrade` includes in addition to the Common `vic-machine` Options , see VCH Upgrade Options.

Prerequisites

- You deployed one or more VCHs with an older version of `vic-machine` .
- You downloaded a new version of the vSphere Integrated Containers Engine bundle.
- Run the `vic-machine ls` command by using the new version of `vic-machine` to see the upgrade status of all of the VCHs that are running on a vCenter Server instance or ESXi host. For information about running `vic-machine ls` , see List VCHs and Obtain Their IDs.
- Optionally note the IDs of the VCHs.
- Obtain the vCenter Server or ESXi host certificate thumbprint. For information about how to obtain the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

Procedure

1. On the system on which you run `vic-machine` , navigate to the directory that contains the new version of the `vic-machine` utility.
2. Run the `vic-machine upgrade` command.

   The following example includes the options required to upgrade a VCH in a simple vCenter Server environment.

   - You must specify the username and optionally the password, either in the `target` option or separately in the `--user` and `--password` options.
   - If the VCH has a name other than the default name, `virtual-container-host` , you must specify the `--name` or `--id` option.
   - If multiple compute resources exist in the datacenter, you must specify the `--compute-resource` or `--id` option.
   - If your vSphere environment uses untrusted, self-signed certificates, you must also specify the thumbprint of the vCenter Server instance or ESXi host in the `--thumbprint` option.

Use upper-case letters and colon delimitation in the thumbprint. Do not use space delimitation.

```
$ vic-machine-operating_system upgrade
--target vcenter_server_username:password@vcenter_server_address
--thumbprint certificate_thumbprint
--id vch_id
```

3. If the upgrade operation fails with error messages, run `vic-machine upgrade` again, specifying a timeout longer than 3 minutes in the `--timeout` option.

```
$ vic-machine-operating_system upgrade
--target vcenter_server_username:password@vcenter_server_address
--thumbprint certificate_thumbprint
--id vch_id
--timeout 5m0s
```

4. If the upgrade operation continues to fail with error messages, run `vic-machine upgrade` again with the `--force` option.

   If your vSphere environment uses untrusted, self-signed certificates, running `vic-machine upgrade` with the `--force` option allows you to omit the `--thumbprint` option.

   CAUTION: It is not recommended to use `--force` to bypass thumbprint verification in production environments. Using `--force` in this way exposes VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials.

```
$ vic-machine-operating_system upgrade
--target vcenter_server_username:password@vcenter_server_address
--id vch_id
--timeout 5m0s
--force
```

5. (Optional) To roll back an upgraded VCH to the previous version, or to revert a VCH that failed to upgrade, run `vic-machine upgrade` again with the `--rollback` option.

```
$ vic-machine-operating_system upgrade
--target vcenter_server_username:password@vcenter_server_address
--id vch_id
--rollback
```

Result

During the upgrade process, `vic-machine upgrade` performs the following operations:

- Validates whether the configuration of the existing VCH is compatible with the new version. If not, the upgrade fails.

157

- Uploads the new versions of the `appliance.iso` and `bootstrap.iso` files to the VCH. There is no timeout for this stage of the upgrade process, so that the ISO files can upload over slow connections.
- Creates a snapshot of the VCH endpoint VM, to use in case the upgrade fails and has to roll back.
- Boots the VCH by using the new version of the `appliance.iso` file.
- Deletes the snapshot of the VCH endpoint VM once the upgrade has succeeded.
- After you upgrade a VCH, any new container VMs will boot from the new version of the `bootstrap.iso` file.
- If the upgrade times out while waiting for the VCH service to start, the upgrade fails and rolls back to the previous version.
- If the upgrade fails with the error `another upgrade/configure operation is in progress`, a previous attempt at upgrading the VCH might have been interrupted without rolling back. In this case, run `vic-machine configure` with the `--reset-progress` option. For information about `vic-machine configure --reset-progress`, see Reset Upgrade or Configuration Progress.

What to Do Next

Upgrade the HTML5 vSphere Client plug-in.

- Upgrade the HTML5 vSphere Client Plug-In on vCenter Server for Windows
- Upgrade the HTML5 vSphere Client Plug-In on a vCenter Server Appliance

VCH Upgrade Options

The command line utility for vSphere Integrated Containers Engine, `vic-machine`, provides an `upgrade` command that allows you to upgrade virtual container hosts (VCHs) to a newer version.

The `vic-machine upgrade` command includes the following options in addition to the common options described in [Common `vic-machine` Options](#).

NOTE: Wrap any option arguments that include spaces or special characters in quotes. Use single quotes if you are using `vic-machine` on a Linux or Mac OS system and double quotes on a Windows system.

`--appliance-iso`

Short name: `--ai`

The path to the new version of the ISO image from which to upgrade the VCH appliance. Set this option if you have moved the `appliance.iso` file to a folder that is not the folder that contains the `vic-machine` binary or is not the folder from which you are running `vic-machine`. Include the name of the ISO file in the path.

NOTE: Do not use the `--appliance-iso` option to point `vic-machine` to an `--appliance-iso` file that is of a different version to the version of `vic-machine` that you are running.

```
--appliance-iso path_to_ISO_file/ISO_file_name.iso
```

`--bootstrap-iso`

Short name: `--bi`

The path to the new version of the ISO image from which to upgrade the container VMs that the VCH manages. Set this option if you have moved the `bootstrap.iso` file to a folder that is not the folder that contains the `vic-machine` binary or is not the folder from which you are running `vic-machine`. Include the name of the ISO file in the path.

NOTE: Do not use the `--bootstrap-iso` option to point `vic-machine` to a `--bootstrap-iso` file that is of a different version to the version of `vic-machine` that you are running.

```
--bootstrap-iso path_to_ISO_file/bootstrap.iso
```

`--force`

Short name: `-f`

Forces `vic-machine upgrade` to ignore warnings and continue with the upgrade of a VCH. Errors such as an incorrect compute resource still cause the upgrade to fail.

You can bypass certificate thumbprint verification by specifying the `--force` option instead of `--thumbprint`.

CAUTION: It is not recommended to use `--force` to bypass thumbprint verification in production environments. Using `--force` in this way exposes VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials.

```
--force
```

--rollback

Short name: None

Rolls a VCH back to its previous version, for example if upgrade failed. Before starting the upgrade process, `vic-machine upgrade` takes a snapshot of the existing VCH. The upgrade process deletes older snapshots from any previous upgrades. The `--rollback` option reverts an upgraded VCH to the snapshot of the previous deployment. Because `vic-machine upgrade` only retains one snapshot, you can only use `--rollback` to revert the VCH to the version that immediately precedes the most recent upgrade.

```
--rollback
```

Upgrade the Plug-Ins on vCenter Server for Windows

If you have a previous installation of the plug-ins for vSphere Integrated Containers, you must upgrade them. This procedure describes how to upgrade an existing plug-ins for vCenter Server on Windows.

Prerequisites

- You are upgrading the HTML5 plug-in on a vCenter Server instance that runs on Windows. If you are running a vCenter Server appliance instance, see Upgrade the HTML5 vSphere Client Plug-In on a vCenter Server Appliance.
- You deployed the vSphere Integrated Containers plug-ins with vSphere Integrated Containers 1.1.x. For information about installing the plug-ins for the first time, see Install the Client Plug-Ins on vCenter Server for Windows.
- You upgraded an existing vSphere Integrated Containers 1.2.x appliance to a newer 1.2.y version. For information about upgrading the vSphere Integrated Containers appliance, see Upgrade the vSphere Integrated Containers Appliance.
- Log in to the Windows system on which vCenter Server is running. You must perform all of the steps in this procedure on this Windows system.

  IMPORTANT: The upgrade script does not function if you have set the `VIC_MACHINE_THUMBPRINT` environment variable on the system on which you run the script. Delete the `VIC_MACHINE_THUMBPRINT` environment variable before running the script..

- Go to http://*upgraded_vic_appliance_address* in a Web browser, download the new version of the vSphere Integrated Containers Engine package, `vic_1.2.y.tar.gz` , and unpack it on the Desktop.
- Obtain the vCenter Server certificate thumbprint. For information about how to obtain and verify the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

Procedure

1. Run the upgrade script and follow the prompts.

   ```
   %USERPROFILE%\Desktop\vic\ui\vCenterForWindows\upgrade.bat
   ```

   i. Enter the IP address of the vCenter Server instance.
   ii. Enter the user name and password for the vCenter Server administrator account.
   iii. Enter **yes** if the vCenter Server certificate thumbprint is legitimate, and wait for the install process to finish.
2. When the installation finishes, stop and restart the services of your management clients.
   i. Restart the HTML5 vSphere Client service.

   ```
   service-control --stop vsphere-ui
   ```

   ```
   service-control --start vsphere-ui
   ```

   ii. Restart the Flex-based vSphere Web Client service.

```
service-control --stop vsphere-client
```

```
service-control --start vsphere-client
```

What to Do Next

Log in to the management clients, go to the vSphere Integrated Containers view, and verify that the version number reflects the upgrade.

Upgrade the Plug-Ins on vCenter Server Appliance

If you have a previous 1.1.x installation of the plug-ins for vSphere Integrated Containers, you must upgrade them. This procedure describes how to upgrade an existing plug-ins for a vCenter Server Appliance.

Prerequisites

- You are upgrading the plug-ins on a vCenter Server appliance instance. If you are running vCenter Server on Windows, see Upgrade the HTML5 vSphere Client Plug-In on vCenter Server for Windows.
- You deployed the vSphere Integrated Containers plug-ins with vSphere Integrated Containers 1.1.x. For information about installing the plug-ins for the first time, see Install the Client Plug-Ins on a vCenter Server Appliance.
- You upgraded an existing vSphere Integrated Containers 1.2.x appliance to a newer 1.2.y version. For information about upgrading the vSphere Integrated Containers appliance, see Upgrade the vSphere Integrated Containers Appliance.
- Go to the vCenter Server Appliance Management Interface (VAMI) at https://*vcsa_address*:5480, click **Access**, and make sure that Bash Shell is enabled.
- Obtain the vCenter Server certificate thumbprint. For information about how to obtain and verify the certificate thumbprint, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.
- The system on which you run the script is running `awk` .

IMPORTANT: The upgrade script does not function if you have set the `VIC_MACHINE_THUMBPRINT` environment variable on the system on which you run the script. Delete the `VIC_MACHINE_THUMBPRINT` environment variable before running the script.

Procedure

1. Connect as root user to the vCenter Server Appliance by using SSH.

   ```
   ssh root@vcsa_address
   ```

2. Use `curl` to copy the new vSphere Integrated Containers Engine binaries from the file server in the upgraded vSphere Integrated Containers appliance to the vCenter Server Appliance.

   ```
   curl -k https://upgraded_vic_appliance_address:9443/files/vic_1.2.x.tar.gz -o
   vic_1.2.x.tar.gz
   ```

   NOTE: Update `vic_1.2.x` to the appropriate version in the command above and in the next step.
3. Unpack the vSphere Integrated Containers binaries.

   ```
   tar -zxf vic_1.2.x.tar.gz
   ```

4. Navigate to `/vic/ui/VCSA` , run the upgrade script, and follow the prompts.

   ```
   cd vic/ui/VCSA
   ```

```
./upgrade.sh
```

      i. Enter the IP address of the vCenter Server instance.

     ii. Enter the user name and password for the vCenter Server administrator account.

    iii. Enter **yes** if the vCenter Server certificate thumbprint is legitimate, and wait for the install process to finish.

    iv. (Optional) If the version that you try to install is same or older than the one already installed, enter **yes** to force reinstall and wait for the process to finish.

5. When the installation finishes, stop and restart the services of your management clients.

      i. Restart the HTML5 vSphere Client service.

```
service-control --stop vsphere-ui
```

```
service-control --start vsphere-ui
```

     ii. Restart the Flex-based vSphere Web Client service.

```
service-control --stop vsphere-client
```

```
service-control --start vsphere-client
```

What to Do Next

Log in to the management clients, go to the vSphere Integrated Containers view, and verify that the version number reflects the upgrade.

Manage the vSphere Integrated Containers Appliance

This section provides information about how to manage your vSphere Integrated Containers appliance.

- Reconfigure the vSphere Integrated Containers Appliance
- Reinitialize the Appliance
- Check the Status of the vSphere Integrated Containers Services
- Restart the vSphere Integrated Containers Services

Reconfigure the vSphere Integrated Containers Appliance

After you have deployed the vSphere Integrated Containers appliance, you can reconfigure the settings that you provided to the OVA installer during deployment. You can also reconfigure the appliance VM itself, for example to expand the amount of storage for vSphere Integrated Containers Registry, or to increase memory and processing power.

Prerequisites

- Log in to a vSphere Web Client instance for the vCenter Server instance on which the vSphere Integrated Containers appliance is running.
- If you use vSphere 6.5, log in to the Flex-based vSphere Web Client, not the HTML5 vSphere Client.

Procedure

1. Shut down the vSphere Integrated Containers appliance by selecting **Shut Down Guest OS**.

   IMPORTANT: Do not select **Power Off**.

2. Right-click the new vSphere Integrated Containers appliance, and select **Edit Settings**.
3. In the **Virtual Hardware** tab, reconfigure the appliance VM as necessary.

   - Increase the number of CPUs
   - Increase the amount of RAM
   - Increase the size of hard disk 2 to expand the storage for vSphere Integrated Containers Registry

4. Click **vApp Options** to modify the settings that you provided when you used the OVA installer to deploy the appliance.

   - In **Appliance Security**, update the password for the appliance root account, enable or disable SSH log in.
   - Reconfigure **Networking Properties** to set a static IP address, update the network configuration, or remove all settings to enable DHCP.
   - Reconfigure **Registry Configuration** to enable or disable vSphere Integrated Containers Registry, change the ports on which the registry publishes services, update the admin and database passwords, enable or disable garbage collection, or update the certificate and key.
   - Reconfigure **Management Portal Configuration** to enable or disable vSphere Integrated Containers Management Portal, change the port on which the portal publishes services, or update the certificate and key.
   - Reconfigure **File Server Configuration** to change the port on which the file server publishes the vSphere Integrated Containers Engine download, or update the certificate and key.

   NOTE: It is not recommended to modify the **Deployment** and **Authoring** settings.

5. Click **OK** to close the Edit Settings window.

6. Power on the vSphere Integrated Containers appliance to complete the reconfiguration.

Result

When the appliance powers on, the new settings are automatically applied.

Reinitialize the vSphere Integrated Containers Appliance

After you power on the vSphere Integrated Containers appliance for the first time, you are prompted to enter the vCenter Server credentials and Platform Services Controller settings. This allows the OVA installer to perform two tasks to initialize the appliance:

- Tag the appliance VM for content trust
- Register the appliance with the Platform Services Controller

After initialization, the vSphere Integrated Containers Getting Started page should display a success message at the top of the page. In this case, no action is necessary.

The Getting Started Page includes a button labeled **Re-Initialize the vSphere Integrated Containers Appliance**.

IMPORTANT: Only reinitialize the appliance in the following circumstances:

- The initialization of the appliance did not succeed and the Getting Started page includes a red error alert instead of a green success alert. For example, you see the error `Failed to locate VIC Appliance. Please check the vCenter Server provided and try again`.
- You need to re-tag the appliance for Docker content trust. For more information, see Re-Tag the vSphere Integrated Containers Appliance.

Re-Tag the vSphere Integrated Containers Appliance

vSphere Integrated Containers Registry implements Docker content trust to sign images. As such, the vSphere Integrated Containers appliance requires a specific VM tag to identify it as a content trust source.

In some cases, you might need to re-tag the appliance VM, for example if the tag has been accidentally deleted, or if the tagging of the VM failed during the first initialization of the appliance. If the tagging of the VM failed during initialization, you see the error `Failed to locate VIC Appliance. Please check the vCenter Server provided and try again` in the Getting Started page.

Procedure

1. In the Hosts and Clusters view of the vSphere Client right-click the OVA VM and select **Tags & Custom Attributes** > **Remove Tag**.
2. Check that the `ProductVM` tag is present in the list of tags and click **Cancel**.
3. If the `ProductVM` tag is missing, go to the vSphere Integrated Containers Getting Started page at http://*vic_appliance_address*.
4. Scroll to the bottom of the page and click the **Re-Initialize the vSphere Integrated Containers Appliance** button.
5. Enter the vCenter Server address and credentials and click **Continue**.
6. After initialization, check in the vSphere Client that the `ProductVM` tag is present.

Check the Status of the vSphere Integrated Containers Services

You can check the status of the vSphere Integrated Containers Registry and vSphere Integrated Containers Management Portal services, and the file server that runs in the appliance, by logging in to the vSphere Integrated Containers appliance.

Prerequisites

You deployed the vSphere Integrated Containers appliance

Procedure

1. Connect to the vSphere Integrated Containers appliance by using SSH.
2. Run one of the following commands to check the status of one of the vSphere Integrated Containers services:

   - vSphere Integrated Containers Registry: `systemctl status harbor.service`
   - vSphere Integrated Containers Management Portal services: `systemctl status admiral.service`
   - Embedded file server: `systemctl status fileserver.service`
   - Demo VCH Installer Wizard: `systemctl status engine_installer.service`

Result

The output shows the status of the service that you specified, as well as the most recent log entries.

Status	Description
`active (running)`	The service is running correctly.
`inactive (failed)`	The service failed to start.
`inactive (dead)`	The service is not responding.

What to Do Next

If the status is `inactive (failed)` or `inactive (dead)`, see Restart the vSphere Integrated Containers Services.

Restart the vSphere Integrated Containers Services

You can restart the vSphere Integrated Containers Registry and vSphere Integrated Containers Management Portal services and the file server that run in the appliance by logging in to the vSphere Integrated Containers appliance.

Prerequisites

You deployed the vSphere Integrated Containers appliance.

Procedure

1. Connect to the vSphere Integrated Containers appliance by using SSH.
2. Run one of the following commands to restart one of the vSphere Integrated Containers services:

   - vSphere Integrated Containers Registry: `systemctl restart harbor.service`
   - vSphere Integrated Containers Management Portal services: `systemctl restart admiral.service`
   - Embedded file server: `systemctl restart fileserver.service`
   - Demo VCH Installer Wizard: `systemctl restart engine_installer.service`

Troubleshooting vSphere Integrated Containers

This section provides solutions for common problems that you might encounter during operation.

- Access vSphere Integrated Containers Engine Log Bundles
- VCH Deployment Times Out
- VCH Deployment Fails with a Certificate Verification Error
- VCH Deployment Fails with Missing Common Name Error Even When TLS Options Are Specified Correctly
- VCH Deployment Fails with Firewall Validation Error
- VCH Deployment Fails with Certificate cname Mismatch
- VCH Deployment Fails with Docker API Endpoint Check Failed Error
- VCH Deployment with a Shared NFS Datastore Fails with an Error About No Single Host Being Able to Access All Datastores
- vSphere Client Plug-In Scripts Fail with No Error Message
- vSphere Integrated Containers Plug-Ins Not Deploying Correctly
- Some Users Cannot Access vSphere Integrated Containers Services
- Deleting or Inspecting a VCH Fails with a Not a VCH or Resource Pool Not Found Error
- Connections Fail with Certificate Errors when Using Full TLS Authentication with Trusted Certificates
- vSphere Integrated Containers Appliance VM Password Refused

Access vSphere Integrated Containers Engine Log Bundles

vSphere Integrated Containers Engine provides log bundles that you can download from the VCH Admin portal for a virtual container host (VCH).

You access the VCH Admin Portal at https://*vch_address*:2378. For more information about the VCH Admin portal, see VCH Administration Portal.

To aid in troubleshooting errors, you can download different log bundles:

- **Log Bundle** contains logs that relate specifically to the VCH that you created.
- **Log Bundle with container logs** contains the logs for the VCH and also includes the logs regarding the containers that the VCH manages.

  NOTE: If the VCH is unable to connect to vSphere, logs that require a vSphere connection are disabled, and you see an error message. For information about accessing logs manually, see Collecting Logs Manually below.

- Live logs (tail files) allow you to view the current status of how components are running.

  - **Docker Personality** is the interface to Docker. When configured with client certificate security, it reports unauthorized access attempts to the Docker server web page.
  - **Port Layer Service** is the interface to vSphere.
  - **Initialization & watchdog** reports:

    ```
    - Network configuration
    - Component launch status for the other components
    - Reports component failures and restart counts
    ```

    At higher debug levels, the component output is duplicated in the log files for those components, so `init.log` includes a superset of the log data.

    Note: This log file is duplicated on the datastore in a file in the endpoint VM folder named `tether.debug`, to allow the debugging of early stage initialization and network configuration issues.

  - **Admin Server** includes logs for the VCH admin server, may contain processes that failed, and network issues. When configured with client certificate security, it reports unauthorized access attempts to the admin server web page.

Live logs can help you to see information about current commands and changes as you make them. For example, when you are troubleshooting an issue, you can see whether your command worked or failed by looking at the live logs.

You can share the non-live version of the logs with administrators or VMware Support to help you to resolve issues.

Logs also include the `vic-machine` commands used during VCH deployment to help you resolve issues.

Collecting Logs Manually

If the VCH Admin portal is offline, use `vic-machine debug` to enable SSH on the VCH and use `scp -r` to capture the logs from `/var/log/vic/`.

Setting the Log Size Cap

The log size cap is set at 20MB. If the size exceeds 20 MB, vSphere Integrated Containers Engine compresses the files and saves a history of the last two rotations. The following files are rotated:

`/var/log/vic/port-layer.log`

`/var/log/vic/init.log`

`/var/log/vic/docker-personality.log`

`/var/log/vic/vicadmin.log`

vSphere Integrated Containers Engine logs any errors that occur during log rotation.

VCH Deployment Times Out

When you use `vic-machine create` to deploy a virtual container host (VCH), the operation times out.

Problem

Deployment fails with a timeout error that states that the context deadline has been exceeded.

```
Failed to power on appliance context deadline exceeded. Exiting...
vic-machine-linux failed: Create timed out:
if slow connection, increase timeout with --timeout
```

Causes

This error can have different causes:

- The connection between the system on which you are running `vic-machine` and vCenter Server is slow. The upload of the ISO files exceeds the default 3 minute timeout.
- The upload of the ISO files succeeds but the VCH fails to obtain an IP address.
- The VCH obtained an IP address, but the VCH service does not start or the VCH cannot connect to the Docker API.

Solutions

1. Set the `vic-machine --timeout` option to allow more time for the ISOs to upload.

   For example, set `--timeout 10m` or `--timeout 20m`.

2. If the ISO upload succeeds with a longer timeout period but the operation still times out, check the DHCP service to make sure than an IP address is available for the VCH.
3. If the DHCP service is working and the operation still times out, see VCH Deployment Fails with Docker API Endpoint Check Failed Error

vSphere Integrated Containers Appliance Fails to Register with PSC

The vSphere Integrated Containers appliance deploys successfully, but the initialization of the appliance fails.

Problem

When you enter the vCenter Server credentials at the login prompt on the vSphere Integrated Containers Getting Started page, you see a red alert with the message `Failed to register with PSC. Please check the vSphere user domain PSC settings and try again`.

Cause

- You are deploying vSphere Integrated Containers 1.2.0.
- Your vSphere environment uses an external Platform Services Controller instance that is not embedded in the vCenter Server instance to which you deployed the appliance.

Solution

If you are performing a fresh installation of vSphere Integrated Containers, download and deploy version 1.2.1 or later. vSphere Integrated Containers 1.2.1 allows you to register the appliance with an external Platform Services Controller.

If an attempt to upgrade vSphere Integrated Containers 1.1.x to version 1.2.0 failed because vCenter Server is managed by an external Platform Services Controller, perform the following procedure.

1. Use SSH to connect to the appliance VM as `root` user, using the password that you specified during the OVA deployment.

   ```
   ssh root@vic_appliance_address
   ```

2. Run a command to register vSphere Integrated Containers Registry with the external Platform Services Controller.

   Specify `harbor` in the `--clientName` parameter and set the following parameters according to your vSphere environment:

   - `--tenant` : The user domain, for example, `vsphere.local` .
   - `--username` : A vCenter Server user name with administrator privileges, for example, `administrator@vsphere.local` .
   - `--password` : The password for the vCenter Server user account, formatted appropriately to escape special characters in the shell

- `--domainController` : The FQDN of the Platform Service Controller instance. If the Platform Service Controller instance is hosted on the vCenter Server host, for example `https://$vCenterFQDN/psc` , the value for this parameter should just be the FQDN of the vCenter host.
- `--admiralUrl` : The address of vSphere Integrated Containers Management Portal, for example `https://vic_appliance_IP:8282` .
- `--defaultUserPrefix` : (Optional) The prefix for the user names of the example users. If you did not change this during OVA deployment, the default is `vic` . If you specify `--defaultUserPrefix` you must also specify `--defaultUserPassword` .
- `--defaultUserPassword` : (Optional) The password for the example users. The password must follow the rules set for vSphere. If you did not change this during OVA deployment, the default is `VicPro!23` . If you specify `--defaultUserPassword` you must also specify `--defaultUserPrefix` .

In the following example, parameters that you must update are highlighted in **bold**. Copy the other parameters as they are shown in the example.

```
java -jar /etc/vmware/admiral/admiral-auth-psc-1.2.0-SNAPSHOT-command.jar
--command=register
--version=6.0
--configDir=/etc/vmware/psc
--clientName=harbor
--tenant=yourdomain
--username=administrator@yourdomain
--password='p@ssw0rd'
--domainController=psc_fqdn
--admiralUrl=https://vic_appliance_IP:8282
--defaultUserPrefix=vic
--defaultUserPassword='VicPro!23'
```

3. Run the command again to register vSphere Integrated Containers Engine.

Change the `--clientName` parameter to `engine` . Specify all of the other parameters with the same values as you used in the previous step.

```
java -jar /etc/vmware/admiral/admiral-auth-psc-1.2.0-SNAPSHOT-command.jar
--command=register
--version=6.0
--configDir=/etc/vmware/psc
--clientName=engine
--tenant=yourdomain
--username=administrator@yourdomain
--password='p@ssw0rd'
--domainController=psc_fqdn
--admiralUrl=https://vic_appliance_IP:8282
--defaultUserPrefix=vic
--defaultUserPassword='VicPro!23'
```

4. Run the command a third time to register vSphere Integrated Containers Management Portal.

    Change the `--clientName` parameter to `admiral` . Specify all of the other parameters with the same values as you used in the previous two steps.

    ```
    java -jar /etc/vmware/admiral/admiral-auth-psc-1.2.0-SNAPSHOT-command.jar
      --command=register
      --version=6.0
      --configDir=/etc/vmware/psc
      --clientName=admiral
      --tenant=yourdomain
      --username=administrator@yourdomain
      --password='p@ssw0rd'
      --domainController=psc_fqdn
      --admiralUrl=https://vic_appliance_IP:8282
      --defaultUserPrefix=vic
      --defaultUserPassword='VicPro!23'
    ```

5. After you have run the command 3 times, run the following command.

    ```
    touch /registration-timestamps.txt
    ```

    This command prevents the login window from reappearing on subsequent visits to the Getting Started page.

Result

The appliance is registered with the external Platform Services Controller and the vSphere Integrated Containers services are available at https://*vic_appliance_address*:8282.

VCH Deployment Fails with a Certificate Verification Error

When you use `vic-machine create` to deploy a virtual container host (VCH), the deployment fails with a certificate verification error, noting that it `failed to create validator`.

Problem

Deployment of the VCH fails during the validation of the configuration that you provided:

```
Failed to verify certificate for target=vcenter_server_or_esxi_host
(thumbprint=vc_or_esxi_cert_thumbprint)
Create cannot continue: failed to create validator
vic-machine-platform.exe failed: x509: certificate signed by unknown authority
```

Cause

The certificate on the vCenter Server or ESXi host that you specified in the `--target` option cannot be validated on the client system.

Solution

If the certificate was signed by a certificate authority (CA), add that CA to the trusted roots for the client system.

If the CA should not be generally trusted, or the certificate is self-signed, obtain the thumbprint of the vCenter Server instance or ESXi host. For information about how to obtain the certificate thumbprint either before running `vic-machine` or to verify a thumbprint from a `vic-machine` error message, see Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.

- If the server is trusted and you did not specify the certificate thumbprint when you ran `vic-machine create`, run `vic-machine create` again, specifying the `--thumbprint` option.
- If a thumbprint that you specified in `--thumbprint` does not match the server certificate reported in the error message:

  1. Remove the thumbprint from the `vic-machine create` command. **WARNING:** A thumbprint mismatch could mean the server you have connected to is not the intended target and might have be spoofed.
  2. Validate that the change in server certificate is legitimate
  3. Re-run `vic-machine create`, specifying a new thumbprint in the `--thumbprint` option.

     Use upper-case letters and colon delimitation in the thumbprint. Do not use space delimitation.

Browser Rejects Certificates with `ERR_CERT_INVALID` Error

Attempts to connect to vSphere Integrated Containers web interfaces fail with certificate errors in Google Chrome browsers.

Problem

When you attempt to access the vSphere Integrated Containers Getting Started page, vSphere Integrated Containers Management Portal, or the administration portal for a virtual container host (VCH), Google Chrome rejects the connection with an `ERR_CERT_INVALID` error and a warning similar to the following:

```
Web_address normally uses encryption to protect your information. When Google Chrome tried
to connect to web_address this time, the website sent back unusual and incorrect
credentials...

You cannot visit web_address right now because the website sent scrambled credentials that
Google Chrome cannot process...
```

This issue only affects Google Chrome. Other browsers do not report certificate errors.

Cause

You have already accepted a client certificate or a generated Certificate Authority (CA) for a previous instance of the vSphere Integrated Containers appliance or for a VCH that had the same FQDN or IP address as the new instance.

Solution

1. Search the keychain on the system where the browser is running for client certificates or CAs that are issued to the FQDN or IP address of the vSphere Integrated Containers appliance or VCH.

   Auto-generated vSphere Integrated Containers appliance and VCH certificates are issued by **Self-signed by VMware, Inc**.

2. Delete any client certificates or CAs for older instances of vSphere Integrated Containers appliances or VCHs.

3. Clear the browser history, close, and restart Chrome.
4. Connect to the vSphere Integrated Containers Getting Started page, vSphere Integrated Containers Management Portal, or VCH Administration portal again, verify the certificate, and trust it if it is valid.

For information about how to verify certificates for the vSphere Integrated Containers appliance, see Verify and Trust vSphere Integrated Containers Appliance Certificates.

VCH Deployment Fails with Missing Common Name Error Even When TLS Options Are Specified Correctly

If you deploy a virtual container host (VCH) and you have specified one of the `vic-machine create --tls-cname` , `--no-tlsverify` , or `--no-tls` options, or you set a static IP address on the client network, the deployment fails with an error about the certificate Common Name being missing.

Problem

Deployment fails during the validation of the configuration that you provided, even if you did specify a TLS option or you set a static IP address on the client network. For example:

```
$ vic-machine-windows create
--target 'Administrator@vsphere.local:password'@vcenter_server
--bridge-network vic bridge --no-tls
### Installing VCH ####
[...]
Common Name must be provided when generating certificates for client
authentication:
[...]
Create cannot continue: unable to generate certificates
--------------------
vic-machine-windows.exe failed: provide Common Name for server certificate
```

If you include a TLS option at the beginning of the `vic-machine create` command rather than the end, you see the following error:

```
$ vic-machine-windows create
--target 'Administrator@vsphere.local:password'@vcenter_server
 --no-tls
--bridge-network vic bridge
### Installing VCH ####
[...]
Unknown argument: bridge
--------------------
vic-machine-windows.exe failed: invalid CLI arguments
```

Cause

String values that you provided for certain options contain spaces or special characters that you did not escape with quotations marks. The `vic-machine create` input validator validates the arguments that you provide only as far as the argument that includes the space or special character. If you specify the TLS option before the argument with the space or special character, `vic-machine create` throws the correct error message. However, if you specify the TLS option after the argument that includes the space or special character, the `vic-machine create` validator stops before it reaches the TLS option, and so throws the error about the missing Common Name.

Solution

Wrap any arguments that contain spaces or special characters in single quotation marks (') on Mac OS and Linux and in double quotation marks (") on Windows.

Option arguments that might require quotation marks include the following:

- User names and passwords in `--target`, or in `--user` and `--password`
- Datacenter names in `--target`.
- VCH names in `--name`.
- Datastore names and paths in `--image-store` and `--volume-store`.
- Network and port group names in all networking options.
- Cluster and resource pool names in `--compute-resource`.
- Folder names in the paths for `--tls-cert-path`, `--tls-server-cert`, `--tls-server-key`, `--appliance-iso`, and `--bootstrap-iso`.

VCH Deployment Fails with Firewall Validation Error

When you use `vic-machine create` to deploy a virtual container host (VCH), deployment fails because firewall port 2377 is not open on the target ESXi host or hosts.

Problem

Deployment fails with a firewall error during the validation phase:

```
Firewall must permit dst 2377/tcp outbound to the VCH management interface
```

Cause

ESXi hosts communicate with the VCHs through port 2377 via Serial Over LAN. For deployment of a VCH to succeed, port 2377 must be open for outgoing connections on all ESXi hosts before you run `vic-machine create`. Opening port 2377 for outgoing connections on ESXi hosts opens port 2377 for inbound connections on the VCHs.

Solution

The `vic-machine` utility includes an `update firewall` command, that you can use to modify the firewall on the ESXi host or the ESXi hosts in a cluster. For information about how to use the `update firewall` command, see Open the Required Ports on ESXi Hosts.

VCH Deployment Fails with Certificate `cname` Mismatch

When you use `vic-machine create` to deploy a virtual container host (VCH), the deployment fails with an error about the certificate `cname` value.

Problem

Deployment fails during the validation of the configuration that you provided:

```
Provided cname does not match that in existing server certificate: cname
Unable to load certificates: cname option doesn't match existing server certificate
in certificate path path_to_certificate
```

Cause

`vic-machine create` attempts to re-use certificates that it finds in `--tls-cert-path`. The default value of `--tls-cert-path` derives from the value that you specify in `--name`. If you are deploying a VCH from the same location and with the same name as a previous VCH, `vic-machine create` reuses the old certificates. This behavior is intentional, to allow you to easily redeploy a VCH without requiring you to re-issue client certificates to users.

Before reusing the existing certificates, `vic-machine` confirms that the existing certificate is valid given the options supplied for the new deployment. The options that influence this in order of priority are:

- `--tls-cname` if specified, or
- `--client-ip-address`, or
- `--public-ip-address` if the client and public network roles share an interface.

The error message means that the existing certificate has a Common Name attribute that differs from the value derived from the options detailed above.

Solution

- To reuse the certificates directly, change `--tls-cname`, `--client-ip-address`, or `--public-ip-address` to match the Common Name in the existing certificate.

- If you want to reuse the Certificate Authority so that client certificates remain valid, but you need to provide a different IP address:

  1. Manually generate the server certificates by using `openssl`, signing them with the existing CA.
  2. Use the `--tls-server-cert` and `--tls-server-key` options to pass the newly generated certificates to `vic-machine create`.

- If you do not want to reuse the certificates, choose one of the following options:

    - Change the location from which you run `vic-machine`. This alters the default `--tls-cert-path`.
    - Change the value of `--name`. This alters the default `--tls-cert-path`.
    - Specify `--tls-cert-path` explicitly.
    - Delete the existing certificates from `--tls-cert-path`

VCH Deployment Fails with Docker API Endpoint Check Failed Error

When you use `vic-machine create` to deploy a virtual container host (VCH), deployment fails because `vic-machine` cannot contact the Docker API endpoint.

Problem

Deployment fails with with the error:

```
Docker API endpoint check failed:
API may be slow to start - try to connect to API after a few minutes:
    Run docker -H 192.168.218.160:2376 --tls info
    If command succeeds, VCH is started. If command fails, VCH failed to install - see
documentation for troubleshooting.
```

Cause

During deployment, `vic-machine` checks that the endpoint VM is reachable from Docker clients. If this check fails, `vic-machine create` fails with an error. This error can be caused by the Docker API being slow to start or because it has failed to start.

Solution

The solution to choose depends on whether the API is slow to start or whether it failed to start.

Docker API is Slow to Start

Wait for a few minutes, then run the `docker info` command to test the responsiveness of the Docker API.

If `docker info` succeeds, it shows information about the VCH, including confirmation that the storage driver is vSphere Integrated Containers.

```
Storage Driver: vSphere Integrated Containers version Backend Engine
```

This output means that the VCH is running correctly and can now accept Docker commands.

If `docker info` times out, it means that the Docker API did not start.

Docker API Did Not Start

If the Docker API was not responsive when you ran `docker info`, download the VCH log bundle and examine the logs to determine why the deployment failed. Collecting the vSphere log bundle might also be useful for troubleshooting.

- For information about how to download VCH logs by using the VCH Admin Portal, see Access the VCH Admin Portal in *vSphere Integrated Containers Engine Administration*.
- For information about how to collect VCH logs manually, see Access vSphere Integrated Containers Engine Log Bundles in *vSphere Integrated Containers Engine Administration*.

VCH Deployment with a Shared NFS Datastore Fails with an Error About No Single Host Being Able to Access All Datastores

Deploying a virtual container host (VCH) to a cluster, and specifying a shared NFS datastore as the image store, fails with the error `No single host can access all of the requested datastores.`

Problem

This error occurs even if all of the hosts in the cluster do appear to have access to the shared NFS datastore.

Cause

VCHs require datastores to be writable. The shared NFS datastore is possibly mounted as read-only.

Solution

To see whether a datastore is writable or read-only, consult `mountInfo` in the Managed Object Browser (MOB) of the vCenter Server instance to which you are deploying the VCH.

1. Go to https://*vcenter_server_address*/mob/.
2. Click **content**.
3. Click **group-xx (Datacenters)** in the `rootFolder` row.
4. Click the managed object reference (MoRef) of your datacenter in the `childEntity` row.
5. Click the MoRef of the shared NFS datastore in the `datastore` row.
6. Click the `DatastoreHostMount` link in the `host` row.
7. Click `mountInfo` and check the `accessMode` value.
8. If the `accessMode` value is `readOnly`, unmount the datastore from vCenter Server and remount it with `readWrite` permissions.

vSphere Client Plug-In Scripts Fail with No Error Message

When you run the scripts to install or upgrade the vSphere Integrated Containers plug-ins for the vSphere Client, the operation fails immediately.

Problem

Running the `install`, `uninstall`, or `upgrade` scripts fails immediately after you enter the password for vCenter Server. The output of the script is `Error! Could not register plugin with vCenter Server. Please see the message above`. However, there is no error message above.

Cause

You have set the `VIC_MACHINE_THUMBPRINT` environment variable on the system on which you are running the script. The presence of the `VIC_MACHINE_THUMBPRINT` environment variable causes the script to skip the verification of the vCenter Server certificate thumbprint. This causes the script to fail.

Solution

Delete the `VIC_MACHINE_THUMBPRINT` environment variable, or run the script on a different system.

vSphere Integrated Containers Plug-Ins Not Deploying Correctly

After you have installed the plug-ins for vSphere Integrated Containers, the HTML5 vSphere Client plug-in appears but is empty, or the plug-ins do not appear at all in one or both of the HTML5 vSphere Client or the Flex-based vSphere Web Client.

Problem

The UI plug-in installer reported success, but you experience one of the following problems:

- The HTML5 plug-in appears in the vSphere Client, but the vSphere Integrated Containers Summary, Virtual Container Hosts, and Containers tabs are empty.
- The plug-ins do not appear in the client at all.

Logging out of the client and logging back in again does not resolve the problem.

Causes

If the vSphere Integrated Containers plug-in appears in the HTML5 client but the tabs are empty, you are not running the correct version of vCenter Server 6.5.0. The vSphere Integrated Containers HTML5 plug-in requires vCenter Server 6.5.0d or later.

If the plug-ins do not appear at all:

- A previous attempt at installing the vSphere Integrated Containers plug-ins failed, and the failed installation state was retained in the client cache.
- You installed a new version of the vSphere Integrated Containers plug-ins that has the same version number as the previous version, for example a hot patch.

Solutions

If the vSphere Integrated Containers plug-in appears in the HTML5 client but the tabs are empty, upgrade vCenter Server to version 6.5.0d or later.

If the plug-ins do not appear at all, restart the vSphere Client services.

Restart the HTML5 Client on vCenter Server on Windows

1. Log into the Windows system on which vCenter Server is running.
2. Open a command prompt as Administrator.
3. Use the `service-control` command-line utility to stop and then restart the vSphere Client service.

   ```
   "C:\Program Files\VMware\vCenter Server\bin\service-control" --stop vsphere-ui
   ```

```
"C:\Program Files\VMware\vCenter Server\bin\service-control" --start vsphere-ui
```

Restart the Flex Client on vCenter Server on Windows

1. Log into the Windows system on which vCenter Server is running.
2. Open a command prompt as Administrator.
3. Use the `service-control` command-line utility to stop and then restart the vSphere Client service.

```
"C:\Program Files\VMware\vCenter Server\bin\service-control" --stop vsphere-client
```

```
"C:\Program Files\VMware\vCenter Server\bin\service-control" --start vsphere-client
```

Restart the HTML5 Client on a vCenter Server Appliance

1. Use SSH to log in to the vCenter Server Appliance as `root`.
2. Use the `service-control` command-line utility to stop the vSphere Client service.

```
service-control --stop vsphere-ui
```

3. Restart the vSphere Client service.

```
service-control --start vsphere-ui
```

Restart the Flex Client on a vCenter Server Appliance

1. Use SSH to log in to the vCenter Server Appliance as `root`.
2. Use the `service-control` command-line utility to stop the vSphere Web Client service.

```
service-control --stop vsphere-client
```

3. Restart the vSphere Web Client service.

```
service-control --start vsphere-client
```

Some Users Cannot Access vSphere Integrated Containers Services

Attempts to connect to the Web-based services of the vSphere Integrated Containers appliance fail for certain users but not for others. The appliance and its services are running correctly.

Problem

The appliance and its services are running correctly. Some users can access the Getting Started page, file server, and vSphere Integrated Containers Management Portal, and Demo VCH Installer Wizard without problems, but for other users the connections fail.

Cause

Some users are attempting to access the Getting Started page, file server, vSphere Integrated Containers Management Portal, and Demo VCH Installer Wizard from client systems that have IP addresses in the range 172.17.0.0-172.22.0.0/16.

vSphere Integrated Containers appliance use the 172.17.0.0-172.22.0.0/16 networks internally. The routing table in the appliance contains routes for these subnets, which causes issues if users attempt to access vSphere Integrated Containers services from an address for which the appliance has a directly connected route. Attempts to connect to the appliance from client systems with IP addresses in the range 172.17.0.0-172.22.0.0/16 fail because the appliance routes return traffic to itself instead of to the client system.

Solution

Access the Getting Started page, file server, vSphere Integrated Containers Management Portal, and Demo VCH Installer Wizard from client systems with IP addresses that are not in the 172.17.0.0-172.22.0.0/16 subnets.

Deleting or Inspecting a VCH Fails with a Not a VCH or Resource Pool Not Found Error

When you use `vic-machine delete` or `vic-machine inspect` to delete or inspect a virtual container host (VCH) and you specify the address of an ESXi host in the `target` option, the operation fails with "an error stating that the target is not a VCH or that the resource pool cannot be found".

Problem

Deleting or inspecting a VCH fails with one of the following error messages:

```
### Inspecting VCH ####
Not a VCH
Failed to get Virtual Container Host vch_name
Not a VCH
--------------------
vic-machine-os failed: inspect failed
```

```
### Inspecting VCH ####
Failed to get VCH resource pool "path_to_resource_pool":
resource pool 'path_to_resource_pool' not found
Failed to get Virtual Container Host vch_name
resource pool 'path_to_resource_pool' not found
--------------------
vic-machine-os failed: inspect failed
```

```
### Removing VCH ####
Not a VCH
Failed to get Virtual Container Host vch_name
Not a VCH
--------------------
vic-machine-os failed: delete failed
```

```
### Removing VCH ####
Failed to get VCH resource pool "path_to_resource_pool":
resource pool 'path_to_resource_pool' not found
Failed to get Virtual Container Host vch_name
resource pool 'path_to_resource_pool' not found
--------------------
```

```
vic-machine-os failed: delete failed
```

Cause

You set the `target` option to the address of an ESXi host that is managed by a vCenter Server instance. If there are multiple ESXi hosts in a cluster, the error that you see depends on the host that you specify in the `target` option.

- If you set the `target` option to the ESXi host on which the VCH is running, you see the error `Not a VCH, Failed to get Virtual Container Host`.
- If you set the `target` option to an ESXi host in the cluster that is not the one on which the VCH is running, you see the error `Not a VCH, Failed to get VCH resource pool`.

Solution

1. Run `vic-machine ls` with the `target` option set to the same ESXi host.

   The `vic-machine ls` operation fails but informs you of the address of the vCenter Server instance that manages the ESXi host.

2. Run `vic-machine delete` or `vic-machine inspect` again, setting the `target` option to the address of the vCenter Server instance that was returned by `vic-machine ls`.

Connections Fail with Certificate Errors when Using Full TLS Authentication with Trusted Certificates

Connections to a virtual container host (VCH) that uses full TLS authentication with trusted Certificate Authority (CA) certificates fail with certificate errors.

Problem

- `vic-machine` operations on a VCH result in a "bad certificate" error:

  ```
  Connection failed with TLS error "bad certificate"
  check for clock skew on the host
  Collecting host-227 hostd.log
  vic-machine-windows.exe failed: tls: bad certificate
  ```

  NOTE: `vic-machine` tolerates a 1 day skew. A skew of 1 day might result in a different certificate error than time skew.
- Connections to the VCH Admin portal for the VCH fail with an `ERR_CERT_DATE_INVALID` error.
- Connections to the VCH from Docker clients fail with a `bad certificate` error.

Cause

There is potentially a clock skew between the VCH and the system from which you are connecting to the VCH.

Solution

1. Go to the VCH Admin portal for the VCH at https://*vch_address*:2378 and check the System Time under **VCH Info**.
2. If the system time of the VCH is wrong, run `vic-machine debug` to enable SSH access to the VCH.

   For information about enabling SSH on a VCH, see Authorize SSH Access to the VCH Endpoint VM.

3. Connect to the VCH endpoit VM by using SSH.
4. Use the `date --set` Linux command to set the system clock to the correct date and time.

   The two most common date formats are the following:

   - Unix Time Stamp: `date --set='@1480969133'`
   - `YYYYMMDD HH:MM` format: `date --set="20161205 14:31"`

To prevent this issue recurring on VCHs that you deploy in the future, verify that the host time is correct on the ESXi host on which you deploy VCHs. For information about verifying time synchronization on ESXi hosts, see VMware KB 1003736.

vSphere Integrated Containers Appliance VM Password Refused

After successfully deploying the vSphere Integrated Containers appliance OVA, you cannot use SSH to log in to the vSphere Integrated Containers appliance VM.

Problem

The root password that you specified during the OVA deployment is refused when you attempt to log in to the appliance by using SSH or the virtual machine console.

Cause

A startup process failed on the first boot of the appliance. This caused the appliance password that you specified during deployment of the OVA not to be set.

Solution

1. Use SSH or the VM console to log in to the appliance VM as `root`.

   ```
   $ ssh root@vic_appliance_address
   ```

2. When prompted, enter the default password.

   ```
   2RQrZ83i79N6szpvZNX6
   ```

3. When prompted to change the root password, set it to the same password that you set during the OVA deployment.

   If the password you used during OVA deployment is rejected, set a different password. For example, it might be rejected because it is based on dictionary words. In this case, you can chose a different, throwaway password to get past this step.

4. When you are logged in, reboot the appliance.

   ```
   $ reboot now
   ```

Result

After the reboot, the password is set to the one that you specified during OVA deployment, even if you had to specify a different, throwaway password in order to log in. When the initial password that you specified during OVA deployment expires, the next time that you log in you must set a new password that complies with the strength check.

vSphere Integrated Containers Security Reference

The Security Reference provides information to allow you to secure your vSphere Integrated Containers implementation.

- Network Security
- External Interfaces, Ports, and Services
- Service Accounts and Privileges
- Apply Security Updates and Patches
- Security Related Log Messages
- Sensitive Data

Network Security

VMware highly recommends using a secure management network for vSphere Integrated Containers Engine. The container VMs communicate with the endpoint VM over the management network when an interactive shell is required. While the communication is encrypted, the public keys are not validated, which leaves scope for man-in-the-middle attacks. This connection is only used for the interactive console when enabled (stdin/out/err), and not for any other purpose.

External Interfaces, Ports, and Services

The following ports must be open on the VCH appliance.

Endpoint VM

Client interface:

- 2375 insecure port for Docker API access if deployed with `--no-tls`
- 2376 for TLS secured port for Docker API access
- 22 SSH when enabled with `vic-machine debug`
- 2378 VIC admin server health and log access (HTTPS)
- 6060 pprof debug data when enabled with --debug levels

Management interface:

- 2377 incoming connections from container VMs
- 443 outgoing connections established to vSphere target
- 443 outgoing connections established to ESX hosts

Bridge interface:

- 53 DNS server for container name resolution

Public interface:

- any port not listed as used elsewhere can be forwarded to a container VM

Container VM

- 6060 pprof debug data when enabled with --debug levels
- vSphere Integrated Containers Engine does not use ports when not configured for debug

Service Accounts and Privileges

vSphere Integrated Containers Engine does not create service accounts and does not assign privileges. The `--ops-user` and `--ops-password` options allow a VCH to operate with less-privileged credentials than those that are required for deploying a new VCH. For information about the `--ops-user` option and the permissions that it requires, see the descriptions of `--ops-user` in VCH Deployment Options and Advanced Examples of Deploying a VCH, and the section Use Different User Accounts for VCH Deployment and Operation.

Apply Security Updates and Patches

Download a new version of vSphere Integrated Containers Engine and upgrade your existing VCHs.

Security Related Log Messages

Security-related information for vSphere Integrated Containers Engine appears in `docker-personality.log` and `vicadmin.log` , that you can access from the VCH Admin portal for a VCH.

Sensitive Data

The VMX file of the VCH endpoint VM stores vSphere Integrated Containers Engine configuration information, which allows most of the configuration to be read-only by the guest. The container VMs might hold sensitive application data, such as environment variables for processes, command arguments, and so on.

vSphere Integrated Containers Certificate Reference

vSphere Integrated Containers authenticates connections to its various components by using TLS certificates. In some cases, the certificates are always automatically generated and self-signed. In other cases, you have the option of providing custom certificates.

This topic provides a reference of all of the certificates that vSphere Integrated Containers uses.

Component	Certificate Type	Purpose	Used By
vCenter Server or ESXi host	Self-signed or custom	Required for installation of the vSphere Client plug-ins and deployment and management of virtual container hosts (VCHs). See Obtain the Certificate Thumbprint of vCenter Server or an ESXi Host.	vSphere administrator
vSphere Integrated Containers Management Portal	Self-signed or custom	Authenticates connections from browsers to vSphere Integrated Containers Management Portal. If you use custom certificates, vSphere Integrated Containers Management Portal requires you to provide the TLS private key as an unencrypted PEM-encoded PKCS#8-formatted file. For information about how to convert certificates to PKCS8 format, see Converting Keys for Use with vSphere Integrated Containers. For information about how to obtain auto-generated appliance certificates, see Obtain the Thumbprints and CA Files of the vSphere Integrated Containers Appliance Certificates and Verify and Trust vSphere Integrated Containers Appliance Certificates.	Cloud and DevOps admininistrators, developers
vSphere Integrated Containers Registry	Self-signed	Authenticates connections to vSphere Integrated Containers Registry instances from Docker clients, replication of projects between registry instances, and registration of additional registry instances in the management portal. For information about how to obtain the registry certificate, see Configure System Settings.	Cloud and DevOps admininistrators, developers
vSphere Integrated Containers file server	Self-signed or custom	Authenticates connections to the Getting Started page, downloads of vSphere Integrated Containers Engine binaries, and the installation of vSphere Client plug-ins. For information about how to obtain auto-generated appliance certificates, see Obtain the Thumbprints and CA Files of the vSphere Integrated Containers Appliance Certificates and Verify and Trust vSphere Integrated Containers Appliance Certificates.	vSphere administrator, Cloud and DevOps admininistrators, developers
VCH	None, self-signed, or custom	Authenticates connections from Docker clients to VCHs. If you use custom certificates, `vic-machine` requires you to supply each X.509 certificate in a separate file, using PEM encoding. PKCS#7 is not supported. For information about how to convert certificates to PEM format, see Converting Certificates for Use with vSphere Integrated Containers. For general information about how `vic-machine` uses certificates, see VCH Deployment Options.	vSphere administrator, Cloud and DevOps admininistrators, developers

VCH Administration Portal	None, self-signed, or custom	Authenticates connections from browsers to the administration portals of individual VCHs. See VCH Administration Portal.	vSphere administrator

Converting Keys for Use with vSphere Integrated Containers Management Portal

To convert a PKCS#1 key to PKCS8 format for use with vSphere Integrated Containers Management Portal, make sure there is no whitespace at the end of the key and run one of the following commands:

- PEM-encoded PKCS#1 to PEM-encoded PKCS#8

```
$openssl pkcs8 -topk8 -inform PEM -outform PEM -nocrypt -in key.der -out key.pkcs8.pem
```

- DER-encoded PKCS#1 to PEM-encoded PKCS#8

```
$ openssl pkcs8 -topk8 -inform DER -outform PEM -nocrypt -in key.der -out key.pkcs8.pem
```

- DER-encoded PKCS#8 to PEM-encoded PKCS#8

```
$ openssl pkcs8 -inform DER -outform PEM -nocrypt -in key.pkcs8.der -out key.pkcs8.pem
```

Converting Certificates for Use with vSphere Integrated Containers Engine

To unwrap a PKCS#7 key for use with `vic-machine`, run the following command:

```
$ openssl pkcs7 -print_certs -in cert_name.pem -out chain.pem
```

Obtain the Thumbprints and CA Files of the vSphere Integrated Containers Appliance Certificates

If you do not provide custom certificates during deployment, the OVA installer generates certificates for the vSphere Integrated Containers Management Portal and the vSphere Integrated Containers file server. These certificates authenticate connections to the Getting Started page, vSphere Integrated Containers Management Portal, and the vSphere Integrated Containers Engine bundle and vSphere Client plug-in downloads. If you deploy the appliance with automatically generated certificates, the certificates are self-signed by an automatically generated Certificate Authority (CA).

The vSphere administrator obtains the thumbprints and CA files and passes them to other users who need to access the Getting Started page or the vSphere Integrated Containers Management Portal.

Procedure

1. Use SSH to connect to the vSphere Integrated Containers appliance as `root` user.

   ```
   $ ssh root@vic_appliance_address
   ```

2. Use `openssl` to view the certificate fingerprint of the file server.

   The file server certificate authenticates access to the Getting Started page, including the downloads for the vSphere Integrated Containers Engine bundle and the vSphere Client plug-in.

   ```
   openssl x509 -in  /opt/vmware/fileserver/cert/server.crt -noout -sha1 -fingerprint
   ```

3. Use `openssl` to view the certificate fingerprint of the management portal.

   The management portal certificate authenticates access to the vSphere Integrated Containers Management Portal.

   ```
   openssl x509 -in  /data/admiral/cert/server.crt -noout -sha1 -fingerprint
   ```

4. Take a note of the two thumbprints and close the SSH session.

5. Use `scp` to copy the CA file for the file server to your local machine.

   ```
   scp root@vic_appliance_address:/opt/vmware/fileserver/cert/ca.crt
   /path/on/local_machine/folder1
   ```

6. Use `scp` to copy the CA file for the management portal to your local machine.

   ```
   scp root@vic_appliance_address:/data/admiral/cert/ca.crt /path/on/local_machine/folder2
   ```

Be sure to copy the two files to different locations, as they are both named `ca.crt`.

You can share the thumbprints and CA files with users who need to connect to the vSphere Integrated Containers Management Portal or downloads. For information about how to verify the thumbprints and trust the CAs, see Verify and Trust vSphere Integrated Containers Appliance Certificates.