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

vSphere Integrated Containers 1.3

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: 1.3

This documentation applies to all 1.3.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.

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304

www.vmware.com

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 the vSphere Integrated Containers Installer
- Deployment Prerequisites
- Deploy the Appliance
- Download the vSphere Integrated Containers Engine Bundle
- Install the vSphere Client Plug-ins

Download the vSphere Integrated Containers Installer

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.

Deployment Prerequisites for vSphere Integrated Containers

Before you deploy the vSphere Integrated Containers appliance and virtual container hosts (VCHs), you must ensure that the virtual infrastructure in which you are deploying it meets certain requirements.

License Requirements

vSphere Integrated Containers requires a vSphere Enterprise Plus license.

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

Virtual Infrastructure Requirements

The different components of vSphere Integrated Containers have different virtual infrastructure requirements.

vSphere Integrated Containers Appliance Requirements

You deploy the vSphere Integrated Containers appliance on a vCenter Server instance. Deploying the appliance directly on an ESXi host is not supported.

- vCenter Server 6.0 or 6.5.
- ESXi 6.0 or 6.5 for all hosts.
- At least 2 vCPUs.
- At least 8GB RAM.
- At least 80GB free disk space on the datastore. The disk space for the appliance uses thin provisioning.

vSphere Client Requirements

vSphere Integrated Containers provides a basic plug-in for the Flex-based vSphere Web Client and a more feature-complete plug-in for the HTML5 vSphere Client:

- The Flex-based plug-in for vSphere 6.0 and 6.5 has limited functionality and only provides information about VCHs and container VMs.
- The HTML5 plug-in for vSphere 6.5 has a more extensive feature set that allows you to deploy and interact with VCHs. The HTML5 vSphere Client plug-in for vSphere Integrated Containers requires vCenter Server 6.5.0d or later.

Supported Configurations for Virtual Container Host Deployment

You can deploy virtual container hosts (VCHs) in the following types of setup:

- 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 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.

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

All 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.

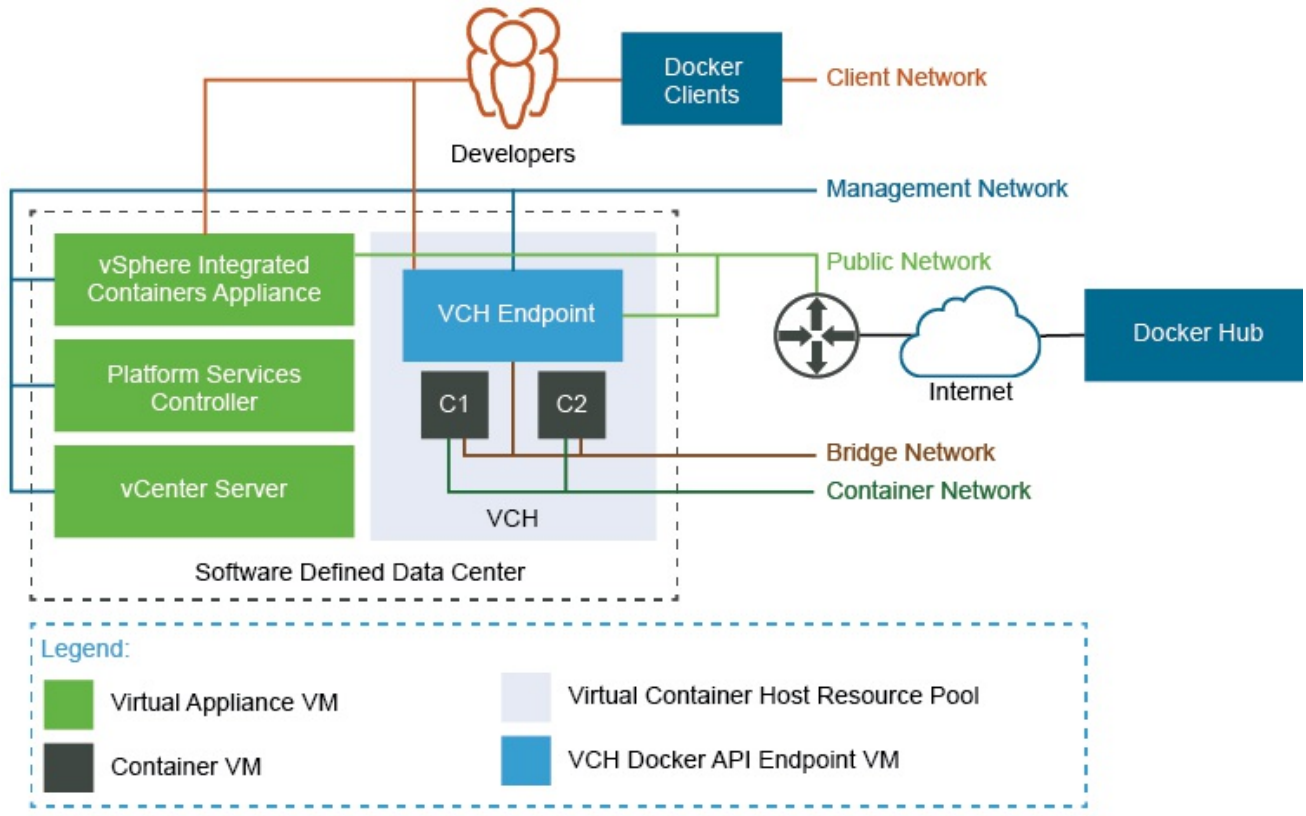
For information about image stores and volumes stores, see Virtual Container Host Storage.

Clock Synchronization

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.

Networking Requirements

The vSphere Integrated Containers appliance requires access to the external Internet, the vSphere Infrastructure, and to the network on which developers connect Docker clients. VCHs connect to multiple different networks, as shown in the image below.



For more information about the networks that VCHs connect to, see Virtual Container Host Networks

IMPORTANT: If you configure a VCH to use separate networks for the public, management, and client networks, these networks must be accessible by the vSphere Integrated Containers appliance.

Networking Requirements for VCH Deployment

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. For more information about the role and requirements of the management network, see Configure 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.
  - For information about bridge networks, see Configure Bridge 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.
- If you use the Create Virtual Container Host wizard to deploy VCHs, you must create and use a port group for the public network. The VCH endpoint VM must be able to obtain an IP address on this port group. If you use `vic-machine` to deploy VCHs, it is still

strongly recommended that use a port group for the public network. Using the default VM Network for the public network instead of a port group prevents vSphere vMotion from moving the VCH endpoint VM between hosts in the cluster. You can use the same port group as the public network for multiple VCHs.

- Optionally create port groups for each of the management and client networks.
- Optionally create port groups for use as mapped container networks. For information about container networks, see Configure Container Networks.
- All hosts in a cluster should be attached to the port groups that you create for the VCH networks 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 how to assign a VLAN ID to a port group, see VMware KB 1003825. For more information about private VLAN, see VMware KB 1010691.

Custom Certificates

If you intend to use 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.

Deploy the vSphere Integrated Containers Appliance

You install vSphere Integrated Containers by deploying a virtual appliance.

The following services run in the vSphere Integrated Containers appliance:

- vSphere Integrated Containers Registry service
- vSphere Integrated Containers Management Portal service
- The file server for vSphere Integrated Containers Engine downloads and installation of the vSphere Client plug-ins
- The `vic-machine` server service, that powers the Create Virtual Container Host wizard in the HTML5 vSphere Client plug-in

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.

Prerequisites

- You downloaded an official build or an open-source build of the OVA installer. For information about where to download the installer, see Download the vSphere Integrated Containers Installer.
- Verify that the environment in which you are deploying the appliance meets the prerequisites described in Deployment Prerequisites for vSphere Integrated Containers.
- 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 and fully qualified domain name (FQDN) for the appliance VM.

    To use DHCP, leave the networking properties blank. If you specify an FQDN, the appliance uses this FQDN to register with the Platform Services Controller and runs the Registry, Management Portal, and file server services at that FQDN.

    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 file server runs.
   - To use custom certificates to authenticate connections to the vSphere Integrated Containers 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 **Configure Example Users** to configure the ready-made example user accounts that vSphere Integrated Containers creates by default 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.

   - Uncheck the **Create Example Users** checkbox if you do not want vSphere Integrated Containers to create user accounts in the Platform Services Controller.
   - In the **Username Prefix for Example Users** text box, optionally modify the prefix of the example user names from the default, `vic`. If you unchecked the **Create Example Users** checkbox, this option is ignored.
   - In the **Password for Example Users** text boxes, modify the password for the example user account from the default, `VicPro!23`. The new password must comply with the password policy for the Platform Services Controller, otherwise the creation of the example user accounts fails. If you unchecked the **Create Example Users** checkbox, this option is ignored.

     IMPORTANT: If you did not uncheck the **Create Example Users** checkbox, it is strongly recommended that you change the default password for the example users.

9. Click **Next** and **Finish** to deploy the vSphere Integrated Containers appliance.
10. 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.

11. (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.

12. Wait for a few minutes to allow the appliance services to start, then in a browser, go to http://*vic_appliance_address* and 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.

13. 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 the following links:

- vSphere Integrated Containers Management Portal
- The download for the vSphere Integrated Containers Engine bundle
- Documentation

What to Do Next

- Download the vSphere Integrated Containers Engine Bundle.
- Install the vSphere Client Plug-ins.
- Log in to vSphere Integrated Containers Management Portal. For information about the management portal, see Configure and Manage vSphere Integrated Containers.
- 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.

Troubleshooting

- If you do not see a green success banner at the top of the Getting Started page after initializing the appliance, the appliance has not initialized correctly. For more information, see Reinitialize the vSphere Integrated Containers Appliance. You should not reinitialize the appliance in any circumstances other than those described in that topic.
- 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.

Download the vSphere Integrated Containers Engine Bundle

After you deploy the vSphere Integrated Containers appliance, you download the vSphere Integrated Containers Engine bundle from the appliance to your usual working machine.

The vSphere Integrated Containers Engine bundle includes:

- Scripts that you run to install, upgrade, or remove the vSphere Client plug-in for vSphere Integrated Containers.
- The `vic-machine` command line utility, that you use to deploy virtual container hosts (VCHs) and manage their lifecycle.

Prerequisites

- You deployed the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- Your working machine runs a 64-bit version of the following Windows, Mac OS, or Linux OS systems.

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

The `vic-machine` utility has been tested and verified on the operating systems above. Other recent 64-bit OS versions should work but are untested.

Procedure

1. In a browser, go to http://*vic_appliance_address*.
2. Scroll down to **Infrastructure deployment tools** and click the link to **download the vSphere Integrated Containers Engine bundle**.
3. Unpack the bundle on your working machine.

Result

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

File	Description
`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-machine-server`	The endpoint for the `vic-machine` API. The `vic-machine` API is currently experimental and unsupported.
`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-ui-darwin`	The OSX executable for the deployment of the vSphere Client plug-in. **NOTE**: Do not run this executable directly.
`vic-ui-linux`	The Linux executable for the deployment of the vSphere Client plug-in. **NOTE**: Do not run this executable directly.
`vic-ui-windows.exe`	The Windows executable for the deployment of the vSphere Client plug-in. **NOTE**: Do not run this executable directly.

`README`	Contains a link to the vSphere Integrated Containers Engine repository on GitHub.
`LICENSE`	The license file.

What to Do Next

- Install the vSphere Client Plug-ins.
- Deploy Virtual Container Hosts.

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 have not installed a previous version of the plug-ins. To upgrade a previous installation, see Upgrade the vSphere Client Plug-Ins on vCenter Server for Windows.
- 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 vSphere Certificate Thumbprints.

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
      ```

3. Delete the vSphere Integrated Containers Engine binaries from the Windows host.

What to Do Next

If you see the error message `At least one plugin is already registered with the target VC`, see Upgrade the vSphere Client Plug-Ins on vCenter Server for Windows.

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.
- You have not installed a previous version of the plug-ins. To upgrade a previous installation, see Upgrade the vSphere Client Plug-Ins on vCenter Server Appliance.
- 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 vSphere Certificate Thumbprints.
- 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. Set the following environment variables:

   - vSphere Integrated Containers appliance address:

     ```
     export VIC_ADDRESS=vic_appliance_address
     ```

   - vSphere Integrated Containers Engine bundle file:

     ```
     export VIC_BUNDLE=vic_v1.3.0.tar.gz
     ```

     If you have installed a different version of the appliance, update `1.3.0` to the appropriate version in the command above. You can see the correct version by going to https://*vic_appliance_address*:9443/files/ in a browser.

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

   Copy and paste the following command as shown:

```
curl -kL https://${VIC_ADDRESS}:9443/files/${VIC_BUNDLE} -o ${VIC_BUNDLE}
```

5. Unpack the vSphere Integrated Containers binaries.

```
tar -zxf ${VIC_BUNDLE}
```

6. 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.

7. 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

If you see the error message `At least one plugin is already registered with the target VC`, see Upgrade the vSphere Client Plug-Ins on vCenter Server Appliance.

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

Deploy Virtual Container Hosts

In vSphere Integrated Containers, you deploy virtual container hosts (VCHs) that serve as Docker API endpoints. VCHs allow Docker developers to provision containers as VMs in your vSphere environment. For a description of the role and function of VCHs, see What is vSphere Integrated Containers Engine? in *Overview of vSphere Integrated Containers*.

After you deploy the vSphere Integrated Containers appliance, you download the vSphere Integrated Containers Engine bundle from the appliance to your usual working machine. The vSphere Integrated Containers Engine bundle includes the `vic-machine` CLI utility. You use `vic-machine` to deploy and manage VCHs at the command line.

The HTML5 vSphere Client plug-in for vSphere Integrated Containers allows you to deploy VCHs interactively from the vSphere Client.

Using the `vic-machine` CLI Utility

After you deploy the vSphere Integrated Containers appliance, you download the vSphere Integrated Containers Engine bundle from the appliance and unpack it on your usual working machine. For information about how to download the bundle, see Download the vSphere Integrated Containers Engine Bundle.

The vSphere Integrated Containers Engine bundle includes the `vic-machine` CLI utility. You use `vic-machine` to deploy and manage virtual container hosts (VCHs) at the command line.

- Running `vic-machine` Commands
- Obtain Certificate Thumbprints
- Set Environment Variables
- Open the Required Ports on ESXi Hosts
- 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 Boot Options

Running `vic-machine` Commands

You run `vic-machine` commands by specifying the appropriate binary for the platform on which you are using `vic-machine`, the `vic-machine` command to run, and multiple options for that command.

You use the `vic-machine create` command to deploy VCHs:

```
vic-machine-windows create --option argument --option argument
```

```
vic-machine-linux create --option argument --option argument
```

```
vic-machine-darwin create --option argument --option argument
```


Basic `vic-machine create` Options

The `vic-machine` options in this section are common to all `vic-machine` commands.

You can set environment variables so that you do not have to specify the `--target`, `--user`, `--password`, and `--thumbprint` options in every `vic-machine` command. For information about setting `vic-machine` environment variables, see Set Environment Variables for Common `vic-machine` Options.

If you use the Create Virtual Container Host wizard, it deploys VCHs to the vCenter Server instance with which the vSphere Integrated Containers appliance is registered, and uses the vSphere credentials with which you are logged in to the vSphere Client. Consequently, when using the Create Virtual Container Host wizard, you do not need to provide any information about the deployment target, vSphere administrator credentials, or vSphere certificate thumbprints.

`--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** when using `vic-machine`.

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.

Usage:

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 user name 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` prompts you to enter the password.

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 user name 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 user name for an account that has the Administrator role on that vCenter Server instance.

You can also specify the user name 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 with reduced privileges 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. For information about using a different account for post-deployment operation, see Configure the Operations User.

Usage:

```
--user esxi_or_vcenter_server_username
```

`--password`

Short name: `-p`

The password for the vSphere administrator 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.

You can also specify the user name 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.

Usage:

```
--password esxi_host_or_vcenter_server_password
```

`--thumbprint`

Short name: None

If your vSphere environment uses untrusted, self-signed certificates to authenticate connections, you must specify the thumbprint of the vCenter Server or ESXi host certificate in the `--thumbprint` option of all `vic-machine` commands. 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.

For information about how to obtain the certificate thumbprint for vCenter Server or an ESXi host, see Obtain vSphere Certificate Thumbprints.

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.

CAUTION: Specifying the `--force` option bypasses safety checks, including certificate thumbprint verification. Using `--force` in this way can expose VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials. Using `--force` can result in unexpected deployment topologies that would otherwise fail with an error. Do not use `--force` in production environments.

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

Usage:

```
--thumbprint certificate_thumbprint
```

`--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.

CAUTION: Specifying the `--force` option bypasses safety checks, including certificate thumbprint verification. Using `--force` in this way can expose VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials. Using `--force` can result in unexpected deployment topologies that would otherwise fail with an error. Do not use `--force` in production environments.

Usage:

```
--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.

Usage:

```
--timeout 5m0s
```

Other `vic-machine create` Options

The `vic-machine create` command provides many more options that allow you to customize the deployment of VCHs to correspond to your vSphere environment and to meet your development requirements.

For information about other VCH deployment options, see the following topics:

- General Virtual Container Host Settings
- Virtual Container Host Compute Capacity

- Virtual Container Host Storage Capacity
- Virtual Container Host Networks
- Virtual Container Host Security
- Configure the Operations User
- Virtual Container Host Boot Options

The options that these topics describe apply to both the `vic-machine` CLI utility and to the Create Virtual Container Host wizard in the vSphere Client.

Specifying Option Arguments

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`.

For example, to deploy a VCH into a cluster named `cluster 1` in a vCenter Server instance that requires the vSphere administrator account `Administrator@vsphere.local`, you must wrap the corresponding option arguments in quotes:

```
vic-machine-linux
--target vcenter_server_address
--user 'Administrator@vsphere.local'
--compute-resource 'cluster 1'
[...]
```

```
vic-machine-windows
--target vcenter_server_address
--user "Administrator@vsphere.local"
--compute-resource "cluster 1"
[...]
```

Obtain vSphere Certificate Thumbprints

If your vSphere environment uses untrusted, self-signed certificates to authenticate connections, you must specify the thumbprint of the vCenter Server or ESXi host certificate in all `vic-machine` commands to deploy and manage virtual container hosts (VCHs). 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. You can set the thumbprint as an environment variable. For information about setting `vic-machine` environment variables, see Set Environment Variables for Common `vic-machine` Options.

If you deploy VCHs from the vSphere Client, the Create Virtual Container Host wizard obtains the thumbprint automatically. However, you might still need to obtain the thumbprint for use in other `vic-machine` commands, for example `vic-machine update firewall` or `vic-machine configure`.

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.

- vCenter Server Appliance or ESXi Host
- Platform Services Controller

vCenter Server Appliance or ESXi Host

You can use SSH and OpenSSL to obtain the certificate thumbprint for a vCenter Server Appliance 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 or to set it as an environment variable.

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 or to set it as an environment variable.

Set Environment Variables for Common `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 vSphere Certificate Thumbprints.

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.

- Create a basic VCH:

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

- List VCHs:

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

- Inspect a VCH:

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

- 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
  ```

For more information about the `vic-machine ls`, `inspect`, `upgrade`, `configure`, and `delete` commands, see Virtual Container Host Administration with `vic-machine`.

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 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.
- Download the vSphere Integrated Containers Engine bundle from the appliance to your usual working machine. For information about how to download the bundle, see Download the vSphere Integrated Containers Engine Bundle.
- 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 vSphere Certificate Thumbprints.

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/datacenter
   --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 a Virtual Container Host to an ESXi Host with No vCenter Server

This topic provides instruction for using `vic-machine` to deploy 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.

The ESXi host to which you deploy the VCH must match the specifications listed in the prerequisites. This example `vic-machine create` command deploys a VCH by using the minimum `vic-machine create` options possible, for demonstration purposes.

Prerequisites

- Deploy the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- Download the vSphere Integrated Containers Engine bundle from the appliance and unpack it on your usual working machine. For information about how to download the bundle, see Download the vSphere Integrated Containers Engine Bundle.
- 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.

  IMPORTANT: Pay particular attention to the Networking Requirements 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 vSphere Certificate Thumbprints.
- Familiarize yourself with the vSphere Integrated Containers Engine binaries, as described in Download the vSphere Integrated Containers Engine Bundle.
- Familiarize yourself with the basic options of the `vic-machine create` command described in Using the `vic-machine` CLI Utility.

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.
- For simplicity, disables the verification of client certificates 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 `vic-machine create` command 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.

Deploy a VCH to a Basic vCenter Server Cluster

This topic provides instructions for using `vic-machine` to deploy 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. This example `vic-machine create` command deploys a VCH by using the minimum `vic-machine create` options possible, for demonstration purposes.

Prerequisites

- Deploy the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- Download the vSphere Integrated Containers Engine bundle from the appliance and unpack it on your usual working machine. For information about how to download the bundle, see Download the vSphere Integrated Containers Engine Bundle.
- 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 VMware vSphere Distributed 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.

  IMPORTANT: Pay particular attention to the Networking Requirements 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 vSphere Certificate Thumbprints.
- Familiarize yourself with the vSphere Integrated Containers Engine binaries, as described in Download the vSphere Integrated Containers Engine Bundle.
- Familiarize yourself with the basic options of the `vic-machine create` command described in Using the `vic-machine` CLI Utility.
- Familiarize yourself with the bridge network and image store, as described in Configure Bridge Networks and Specify the Image Store.

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.
- A vCenter Single Sign-On user and password for a vSphere administrator account, 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.
- For simplicity, disables the verification of client certificates 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 `vic-machine create` command 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.

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 perform operations on VCHs or container VMs directly in the vCenter Server inventory. Specifically, powering off, powering on, or deleting the VCH can cause vSphere Integrated Containers Engine to not function correctly. Always use the vSphere Integrated Containers plug-in for the vSphere Client or `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 or vSphere Integrated Containers Management Portal to perform operations on containers.

Prerequisites

- 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 Client to that vCenter Server instance.
- If you deployed the VCH to an ESXi host, log in to the UI for that host.

Procedure

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

     The resource pool contains the VCH endpoint VM.
2. In a Docker client, run the `docker info` command to confirm that you can connect to the VCH.

   ```
   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.

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

4. View the container image files in the vSphere Client.
   - vCenter Server: Go to **Storage**, right-click the datastore that you designated as the image store, and select **Browse Files**.
   - ESXi host: Go to **Storage** and click **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.

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

7. View the container VMs in the vSphere Client.

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

Virtual Container Host Boot Options

The `vic-machine create` utility provides options that change the location of the ISO files from which virtual container hosts (VCHs) and container VMs boot.

`vic-machine` Options

The options in this topic are only available with the `vic-machine create` command. They are not available in the Create Virtual Container Host wizard in the vSphere Client.

`--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.

Usage:

```
--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.

Usage:

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

Example `vic-machine` Commands

If you moved the `appliance.iso` or `bootstrap.iso` file to a location that is not the folder that contains the `vic-machine` binary, you must point `vic-machine` to those ISO files.

This example `vic-machine create` command deploys a VCH that specifies `--appliance-iso` to direct `vic-machine` to the location in which you stored the `appliance.iso` file.

```
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-tlsverify
--appliance-iso path_to_iso/appliance.iso
```

This example `vic-machine create` command deploys a VCH that specifies `--bootstrap-iso` to direct `vic-machine` to the location in which you stored the `appliance.iso` file.

```
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-tlsverify
--bootstrap-iso path_to_iso/bootstrap.iso
```

Deploy Virtual Container Hosts in the vSphere Client

If you have installed the HTML5 plug-in for vSphere Integrated Containers, you can deploy virtual container hosts (VCHs) interactively in the vSphere Client.

The different options that you configure in the Create Virtual Container Host wizard in the vSphere Client correspond to `vic-machine create` options. The `vic-machine create` options are exposed by the `vic_machine_server` service of the vSphere Integrated Containers appliance. When you use the Create Virtual Container Host wizard, it deploys VCHs to the vCenter Server instance with which the vSphere Integrated Containers appliance is registered, and uses the vSphere credentials with which you are logged in to the vSphere Client. Consequently, when using the Create Virtual Container Host wizard, you do not need to provide any information about the deployment target, vSphere administrator credentials, or vSphere certificate thumbprints.

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.
- Make sure that your virtual infrastructure meets the requirements for VCH deployment. For information about virtual infrastructure requirements, see Deployment Prerequisites for vSphere Integrated Containers.

  IMPORTANT: Pay particular attention to the Networking Requirements for VCH Deployment.

- Make sure that the correct ports are open on all ESXi hosts. For information about how to open ports on ESXi hosts, see Open the Required Ports on ESXi Hosts.

Procedure

1. Log in to the HTML5 vSphere Client with a vSphere administrator account, and click the **vSphere Client** logo in the top left corner.
2. Under Inventories, click **vSphere Integrated Containers**.

   The vSphere Integrated Containers view presents the number of VCHs and container VMs that you have deployed to this vCenter Server instance.

3. Click **vSphere Integrated Containers** in the main panel and select the **Virtual Container Hosts** tab.

   On first connection, if you see the message `Failed to verify the vic-machine server at endpoint https://vic_appliance_address:8443`, perform the following steps to trust the certificate of the `vic-machine` service that is running in the appliance:

   i. Click the link **View API directly in your browser** that appears in step 3 of the error message.
   ii. In the new browser tab that opens, follow your browser's usual procedure to trust the certificate.

   You should see the confirmation message `You have successfully accessed the VCH Management API`.

   iii. Close the new browser tab and click the **Refresh** button in the error message in the **Virtual Container Hosts** tab.

   When you have trusted the certificate, the error message disappears.

4. Click **+ New Virtual Container Host**.

   The Create Virtual Container Host wizard opens.

What to Do Next

See the following topics for instructions about how to fill in the pages of the Create Virtual Container Host wizard:

1. General Settings
2. Compute Capacity
3. Storage Capacity
4. Networks

General Virtual Container Host Settings

When you deploy a virtual container host (VCH), you can configure a name for the VCH, a naming convention for container VMs, and debugging levels.

Options

The sections in this topic each correspond to an entry in the General Settings page of the Create Virtual Container Host wizard and to the corresponding `vic-machine create` options.

VCH Name

Aname for the VCH, that appears the vCenter Server inventory and that you can use in other `vic-machine` commands. The default VCH name is `virtual-container-host`.

Create VCH Wizard

Enter a name for the VCH.

vic-machine Option

`--name`, `-n`

If a VCH of the same name exists on the ESXi host or in the vCenter Server inventory, `vic-machine create` fails with an error. 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
```

Container VM Name Template

Enforce a naming convention for container VMs, that applies a prefix or suffix to the names of all container VMs that run in the 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 whether to use the container name or the container ID for the second part of the container VM display name. If you use the container name, the container VM display names use either the name that Docker generates for the container, or a name that the container developer specifies in `docker run --name` when they run the container.

Create VCH Wizard

1. Optionally enter a container name prefix.
2. Select **Docker name** or **Container ID**.
3. Optionally enter a container name suffix.

vic-machine Option

`--container-name-convention`, `--cnc`

Specify a prefix and/or suffix to apply to container names, and add `-{name}` or `-{id}`, including the curly brackets, to specify whether to use the container name or the container ID for the second part of the container VM display name.

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

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

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

Debug

Deploy the VCH with more verbose levels of logging, and optionally modify the behavior of `vic-machine` for troubleshooting purposes. Specifying a debug level of greater than 0 increases the verbosity of the logging for all aspects of VCH operation, not just deployment. For example, by setting a higher debug level, you increase the verbosity of the logging for VCH initialization, VCH services, container VM initialization, and so on.

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 Debug Running Virtual Container Hosts.

You can set a debugging level of 1, 2, or 3. Setting level 2 or 3 changes the behavior of `vic-machine create` as well as increasing the level of verbosity of the logs:

- `1` Provides verbosity in the logs, with no other changes to `vic-machine` behavior. This is the default setting.
- `2` Exposes servers on more interfaces, launches `pprof` in container VMs.
- `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.

Additionally, deploying a VCH with debug level 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.

Create VCH Wizard

- Leave the default level of 0 for usual deployments.
- Optionally select level 1, 2, or 3 if you need to debug deployment problems.

vic-machine Option

`--debug`, `-v`

Optionally specify a debugging level of `1`, `2`, or `3`. If not specified, the debug level is set to 0 and verbose logging is disabled.

```
--debug 3
```

Syslog

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.

Create VCH Wizard

1. Select **tcp** or **udp** for the transport protocol.
2. Enter the IP address or FQDN of the syslog endpoint.
3. Optionally enter the port on which with syslog endpoint is exposed if it is not the default of 514.

vic-machine Option

`--syslog-address`, no short name

Specify the address and port of the syslog endpoint. 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
```

What to Do Next

If you are using the Create Virtual Container Host wizard, click **Next** to go to the Compute Capacity settings.

Example `vic-machine` Commands

The following examples show `vic-machine create` commands that use the options described in this topic. For simplicity, the examples all use the `--no-tlsverify` option to automatically generate server certificates but disable client authentication. The examples use an existing port group named `vch1-bridge` for the bridge network and designate `datastore1` as the image store, and deploy the VCH to `cluster1` in datacenter `dc1`.

Set a Container Name Convention

This example `vic-machine create` command deploys a VCH that specifies `--container-name-convention` so that the vCenter Server display names of all container VMs include the prefix `vch1-container` followed by the container name.

```
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-tlsverify
--container-name-convention vch1-container-{name}
```

Configure Debugging and Sylog on a VCH

This example `vic-machine create` command deploys a VCH that sets the deployment debugging level to 3 and sends logs to an external syslog endpoint.

```
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 vcenter_server_certificate_thumbprint
--no-tlsverify
--debug 3
--syslog-address tcp://syslog_host_address
```

Virtual Container Host Compute Capacity

When you deploy a virtual container host (VCH), you must select the compute resource in your virtual infrastructure in which to deploy the VCH. You can optionally configure resource usage limits on the VCH.

Options

The sections in this topic each correspond to an entry in the Compute Capacity page of the Create Virtual Container Host wizard and to the corresponding `vic-machine create` options.

Compute Resource

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

NOTE: You cannot deploy a VCH to a specific host in a cluster. You deploy the VCH to the cluster, and DRS manages the placement of the VCH on a host.

Create VCH Wizard

Selecting a compute resource is **mandatory**.

1. Expand the **Compute resource** inventory hierarchy.
2. Select a standalone host, cluster, or resource pool to which to deploy the VCH.

vic-machine Option

`--compute-resource`, `-r`

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
```

CPU

Limit the amount of CPU capacity that is available for use by the VCH resource pool. This limit also applies to the container VMs that run in the VCH resource pool. Specify the CPU capacity in MHz.

Create VCH Wizard

In the **CPU** text box, leave the default value of `Unlimited`, or optionally enter a limit of between 1 and 9779 MHz.

vic-machine Option

`--cpu`, no short name

Specify a CPU limit value of between 1 and 9779 MHz. If not specified, `vic-machine create` sets the limit to 0 (unlimited).

```
--cpu 1024
```

Memory

Limit the amount of memory that is available for use by the VCH resource pool. This limit also applies to the container VMs that run in the VCH resource pool. Specify the memory limit value in MB.

Create VCH Wizard

In the **Memory** text box, leave the default value of `Unlimited`, or optionally enter a limit of between 1 and 21765 MB.

vic-machine Option

`--memory`, `--mem`

Specify a limit of between 1 and 21765 MB. If not specified, `vic-machine create` sets the limit to 0 (unlimited).

```
--memory 1024
```

Advanced Options

When using the Create Virtual Container Host wizard, if you change any of the advanced options, leave the **Advanced** view open when you click **Next** to proceed to the next page.

If you are using `vic-machine`, 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`.

For information about vSphere memory and CPU shares and reservations, see Allocate Memory Resources, and Allocate CPU Resources in the vSphere documentation.

CPU Reservation

Reserve a quantity of CPU capacity for use by the VCH resource pool. This limit also applies to the container VMs that run in the VCH resource pool. Specify the CPU reservation value in MHz.

Create VCH Wizard

1. Expand **Advanced**.
2. In the **CPU reservation** text box, leave the default value of 1, or optionally enter a limit of between 1 and 9779 MHz.

vic-machine Option

`--cpu-reservation`, `--cpur`

Specify a limit of between 1 and 9779 MHz. If not specified, `vic-machine create` sets the reservation to 1.

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

CPU Shares

Set CPU shares on the VCH resource pool. This limit also applies to the container VMs that run in the VCH resource pool.

Create VCH Wizard

1. Expand **Advanced**.
2. In the **CPU shares** text box, leave the default value of **Normal**, or select **Low** or **High**.

vic-machine Option

`--cpu-shares` , `--cpus`

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
```

Memory Reservation

Reserve a quantity of memory for use by the VCH resource pool. This limit also applies to the container VMs that run in the VCH resource pool. Specify the memory reservation value in MB.

Create VCH Wizard

1. Expand **Advanced**.
2. In the **Memory reservation** text box, leave the default value of 1, or optionally enter a limit of between 1 and 21153 MB.

vic-machine Option

`--memory-reservation` , `--memr`

Specify a limit of between 1 and 21153 MB. If not specified, `vic-machine create` sets the reservation to 1.

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

Memory Shares

Set memory shares on the VCH resource pool. This limit also applies to the container VMs that run in the VCH resource pool.

Create VCH Wizard

1. Expand **Advanced**.
2. In the **Memory shares** text box, leave the default value of **Normal** or select **Low** or **High**.

vic-machine Option

`--memory-shares` , `--mems`

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
```

Endpoint VM CPUs

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: In most cases, increase the overall CPU capacity of the VCH resource pool, rather than increasing the number of CPUs on the VCH endpoint VM. This option is mainly intended for use by VMware Support.

Create VCH Wizard

1. Expand **Advanced**.
2. In the **CPUs** text box, leave the default value of 1 or enter a higher number of CPUs.

vic-machine Option

Specify a value of greater than 1. If not specified, `vic-machine create` sets the number of CPUs to 1.

`--endpoint-cpu`, no short name

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

Endpoint VM Memory

The amount of memory for the VCH endpoint VM. 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, increase the overall amount of memory for the VCH resource pool, rather than the amount of memory of the VCH endpoint VM. Use `docker create -m` to set the memory on container VMs. This option is mainly intended for use by VMware Support.

Create VCH Wizard

1. Expand **Advanced**.
2. In the **Memory** text box, leave the default value of 2048 MB, or enter a value of between 1 and 21765 MB.

vic-machine Option

`--endpoint-memory`, no short name

Specify a value of between 1 and 21765 MB. If not specified, `vic-machine create` sets memory to 2048 MB.

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

What to Do Next

If you are using the Create Virtual Container Host wizard, click **Next** to go to the Storage Capacity settings.

Example `vic-machine` Commmands

The following examples show `vic-machine create` commands that use the options described in this topic. For simplicity, the examples all use the `--no-tlsverify` option to automatically generate server certificates but disable client authentication. The examples use an existing port group named `vch1-bridge` for the bridge network and designate `datastore1` as the image store.

Deploy to a vCenter Server Cluster with Multiple Datacenters and Datastores

This example `vic-machine create` command deploys a VCH named `vch1` to the cluster `cluster1` in datacenter `dc1`.

```
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-tlsverify
```

Deploy to a Specific Standalone Host in vCenter Server

This example `vic-machine create` command deploys a VCH on the ESXi host with the FQDN `esxihost1.organization.company.com`.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--bridge-network vch1-bridge
--image-store datastore1
--compute-resource esxihost1.organization.company.com
--name vch1
--thumbprint certificate_thumbprint
--no-tlsverify
```

Deploy to a Resource Pool on an ESXi Host

This example `vic-machine create` command deploys a VCH into a resource pool named `rp 1`. The resource pool name is wrapped in quotes, because it contains a space. It does not specify an image store, assuming that the host in this example only has one datastore.

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

Deploy to a Resource Pool in a vCenter Server Cluster

This example `vic-machine create` command deploys a VCH into a resource pool named `rp 1`. In this example, the resource pool name `rp 1` is unique across all hosts and clusters, so it only specifies 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-tlsverify
```

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-tlsverify
```

Set Limits on Resource Use

This example `vic-machine create` command 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-tlsverify
```

Virtual Container Host Storage Capacity

Virtual container hosts (VCHs) require a datastore in which to store container image files, container VM files, and the files for the VCH itself. You can also specify one or more datastores in which container developers can create container volumes.

- Specify the Image Datastore
- Specify Volume Datastores

Storage Requirements and Limitations

The storage that you select for use as image and volume stores for VCHs must meet the following requirements.

- 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.
- Datastores that you specify as image and volume stores should ideally be accessible to all of the hosts in a cluster. Specifying storage that is not accessible to all of the hosts in a cluster is possible, but might result in all of your container VMs and container volumes being placed on the same host.
    - If you specify different datastores in the different datastore options, and if no single host in a cluster can access all of the datastores, VCH deployment 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, VCH deployment 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.

Specify the Image Datastore

When you deploy a virtual container host (VCH), you must specify a datastore or datastore folder for use as the image store. The image store is the vSphere datastore in which to store container image files, container VM files, and the files for the VCH itself, including a creation log file.

You can also optionally change the base image size for container images.

Options

The sections in this topic each correspond to an entry in the Image Datastore section of the Storage Capacity page of the Create Virtual Container Host wizard, and to the corresponding `vic-machine create` options.

Datastore

If you are deploying the VCH to a vCenter Server cluster, the datastore that you designate as the image store must be shared by at least two, but preferably all, 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). Using non-shared datastores might lead to situations in which all container VMs and image files are stored on a single host.

You can specify a datastore folder to use as the image store. If the folder that you specify does not already exist, `vic-machine create` creates it.

When you deploy a VCH `vic-machine` creates the following set of folders in the image datastore:

- A folder for the VM files of the VCH:

  ```
  datastore_name/vch_name
  ```

  This folder includes a creation log file named `vic-machine_timestamp_create_id.log`,
- A key-value store folder for the VCH:

  ```
  datastore_name/vch_name/kvstores
  ```

- A folder in which to store all of the container images that you pull into the VCH.
  - If you designate the whole datastore as the image store, images are stored in the following location:

    ```
    datastore_name/VIC/vch_uuid/images
    ```

  - If you designate a datastore folder as the image store, images are stored in the following location:

    ```
    datastore_name/path_to_folder/VIC/vcu_uuid/images
    ```

By specifying a datastore folder, you can designate the same datastore folder as the image store for multiple VCHs. Only one `VIC` folder is created in the datastore, but it 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 and run 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 names as the container VMs.

Create VCH Wizard

Specifying an image store is **mandatory**.

1. Select a datastore from the **Datastore** drop-down menu.

   Select a datastore that is shared by at least two, but preferably all, hosts in a cluster.

2. In the **File folder** text box, optionally enter the path to a folder in the specified datastore, to use to store image files.

vic-machine Option

`--image-store`, `-i`

Specifying an image store is **mandatory** if there is more than one datastore in your vSphere environment. If there is only one datastore in your vSphere environment, `vic-machine` uses it automatically and you do not need to specify the datastore. 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.

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

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

To specify a datastore folder to use as the image store, include the path to the folder in the `--image-store` option:

```
--image-store datastore_name/path/to/folder
```

Base Image Size

The size of the base image from which to create other container 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. For information about container base images, see Create a base image in the Docker documentation.

Create VCH Wizard

1. In the **Max Container VM image size** text box, leave the default value of 8, or enter a different value.
2. Select **GB** or **MB**.

vic-machine Option

`--base-image-size`, no short name

Specify a value in GB or MB. If not specified, `vic-machine create` sets the image size to 8 MB.

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

What to Do Next

If you are using the Create Virtual Container Host wizard, scroll down the page to specify Volume Datastores.

Example `vic-machine` Commmand

This example `vic-machine create` command deploys a VCH that uses the folder `vch1_images` in `datastore1` as the image store.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1/vch1_images
--bridge-network vch1-bridge
--name vch1
--thumbprint certificate_thumbprint
--no-tlsverify
```

Specify Volume Datastores

Volume stores for virtual container hosts (VCHs) are datastores in which to create volumes when container developers use the `docker volume create` command or deploy containers that use volumes. You can specify either a datastore that is backed by vSphere or an NFS share point as the volume store.

About Volume Stores

For information about how Docker containers use volumes, see Use volumes in the Docker documentation.

To specify a volume store, you provide the datastore name or NFS mount point, an optional path to a specific folder in that datastore, and a volume store label.

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. The volume store label must be unique.

You can create multiple volume stores for a single VCH.

IMPORTANT: If you do not specify a volume store when you create a VCH, no volume store is created by default and container developers cannot create or run containers that use 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.

If you delete a VCH, by default any volumes that the VCH manages remain available in the volume store. There are different ways to delete volumes that are no longer required:

- Delete the volumes manually in the vSphere Client.
- Run `docker rm -v`.
- Run `vic-machine delete` with the `--force` option.
- Select **Delete persistent and anonymous volumes** when you delete the VCH in the vSphere Client.

vSphere Datastores

If you are deploying the VCH to a vCenter Server cluster, the vSphere datastores that you designate as volume stores should be shared by at least two, but preferably all, ESXi hosts in the cluster. Using non-shared datastores is possible and deployment succeeds, but results in a warning that this configuration limits the use of vSphere features such as vSphere vMotion and DRS.

If you specify a vSphere datastore without specifying a datastore folder, vSphere Integrated Containers Engine 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 folder, vSphere Integrated Containers Engine creates a folder named `volumes` in the location that you specify. If the folders that you specify do not already exist on the datastore, vSphere Integrated Containers Engine creates the appropriate folder structure. Any volumes that container developers create will appear in the `path/volumes` folder.

vSphere Integrated Containers Engine creates the `volumes` folder independently from the folders for VCH files so that you can attach existing volume stores to different VCHs. You can assign an existing volume store that already contains data to a VCH by either creating a new VCH or by running `vic-machine configure --volume-store` on an existing one. You can only assign a volume store to a single VCH at a time.

IMPORTANT: If multiple VCHs 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.

NFS Datastores

If you use NFS volume stores, container developers can share the data in the volumes in the volume stores 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.

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.

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

Anonymous Volumes

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`. Also, some common container images require the presence of a `default` volume store in order to run.

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

Add Volume Datastores

This section describes the Volume Datastores section of the Storage Capacity page of the Create Virtual Container Host wizard, and the corresponding `vic-machine create` option.

Create VCH Wizard

1. Optionally enable anonymous volumes by setting the **Enable anonymous volumes** switch to the green ON position.

   Enabling anonymous volumes automatically adds the label `default` to the first volume datastore.

2. Select a datastore for the first volume store from the **Datastore** drop-down menu.
3. In the **Folder** text box, optionally enter the path to a folder in the specified datastore.

   If the folders that you specify in the path do not already exist on the selected datastore, vSphere Integrated Containers Engine creates the appropriate folder structure.

4. If you did not enable anonymous volumes, or if this is an additional volume store, provide a label for the volume store in the **Volume store name** text box.
5. Optionally click the **+** button to add more volume datastores to the VCH, and repeat the proceeding steps for each additional volume datastore.

NOTE: It is not currently possible to specify an NFS share point as a volume store in the Create Virtual Container Host wizard. If you use the wizard to create VCHs, after deployment, run `vic-machine configure` with the `--volume-store` option to add NFS share points to the VCH. For information about adding volume stores after deployment, see Add Volume Stores.

vic-machine Option

`--volume-store`, `--vs`

To specify a whole vSphere datastore for use as a volume store, provide the datastore name and a volume store label:

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

Optionally use the `ds://` prefix to specify a datastore that is backed by vSphere:

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

To specify a volume store in a datastore folder, add the path to the appropriate folder:

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

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

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

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

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

Use the label `default` to allow container developers to create anonymous volumes:

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

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

You can specify the `--volume-store` option multiple times, and 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
```

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

What to Do Next

If you are using the Create Virtual Container Host wizard, click **Next** to go to the VCH Networks settings.

Example `vic-machine` Commmand

This example `vic-machine create` command deploys a VCH with 3 volume stores:

- A `default` volume store in the `volumes` folder on `datastore 1`.
- A second volume store named `volume_store_2` in the `volumes` folder on `datastore 2`.
- A volume store named `shared_volume` in a NFS share point, from which containers can mount shared volumes.

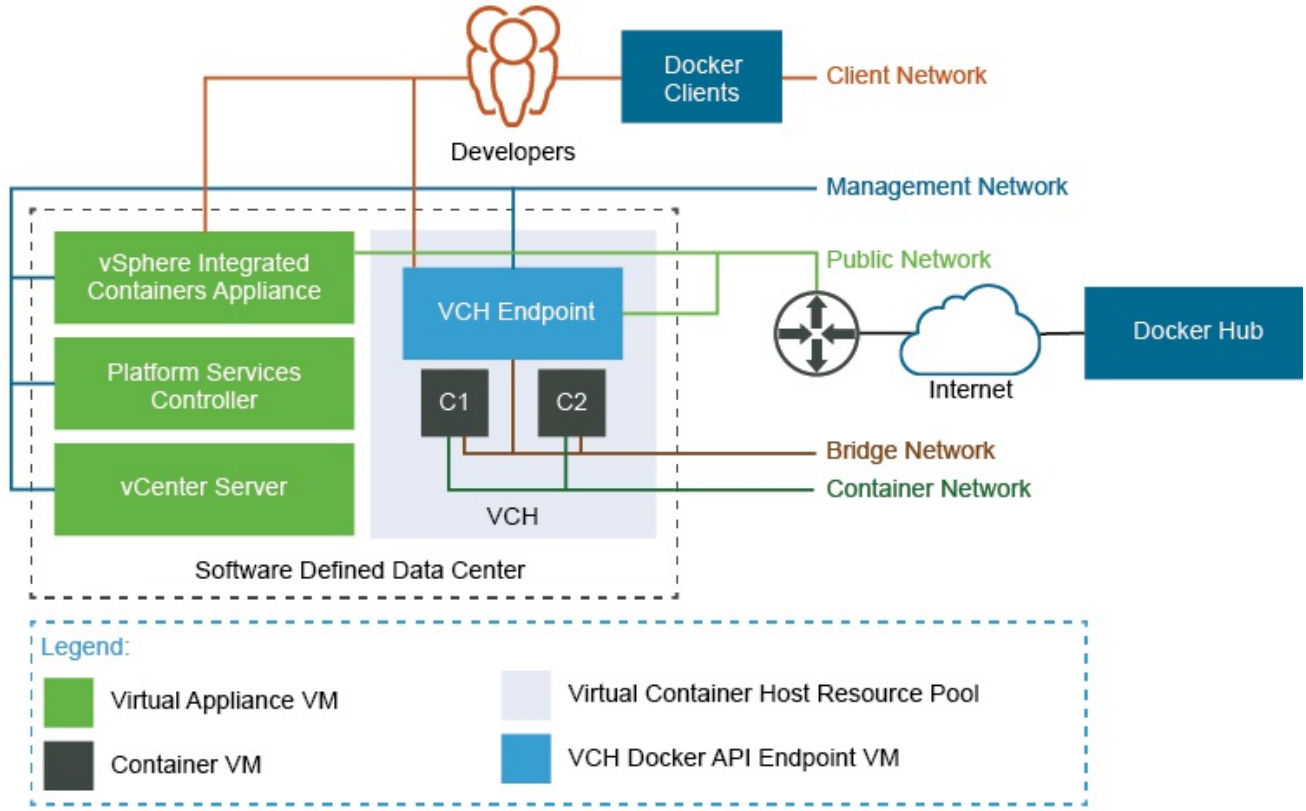
```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--bridge-network vch1-bridge
--image-store 'datastore 1'
--volume-store 'datastore 1'/volumes:default
--volume-store 'datastore 2'/volumes:volume_store_2
--volume-store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

Virtual Container Host Networks

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.

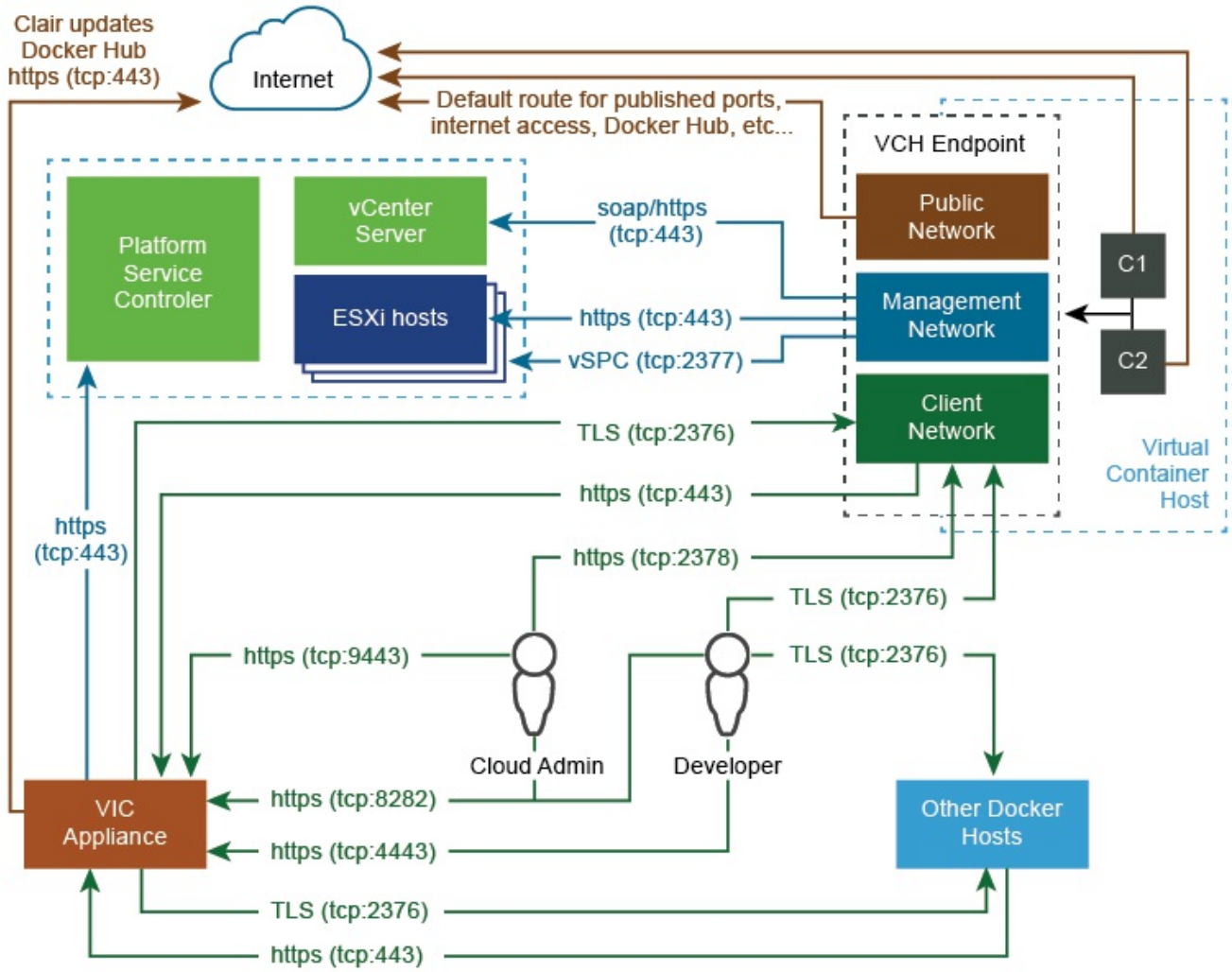
High-Level View of VCH Networking

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.



Ports and Protocols

The image below shows detailed information how different entities that are part of a vSphere Integrated Containers environment communicate with each other.



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 port groups or logical switches in vSphere before you deploy a VCH.

IMPORTANT: All hosts in a cluster should be attached to the port groups that you use for the VCH networks and for any mapped container networks.

For general information VCH networking requirements and how to create a distributed virtual switch and port group, see Networking Requirements for VCH Deployment.

- **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. You must create a dedicated port group for the bridge network for every VCH. For information about VCH bridge networks, see Configure Bridge Networks.
- **Public Network:** The network that container VMs and VCHs use to access the Internet. If you use the Create Virtual Container Host Wizard, you must create a port group for the public network. You cannot use the same port group for the public network as you use for the bridge network. However, you can use the public network port group for the client and management networks. For information about VCH public networks, see Configure the Public Network.
- **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, and provide distinct advantages over using Docker user-defined networks. For information about container networks, including their advantages over Docker user-defined 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 information proxy servers, see Configure VCHs to Use Proxy Servers.

Networking Limitations

In previous releases of vSphere Integrated Containers, VCHs supported a maximum of 3 distinct network interfaces. Due to this limitation, at least two of the public, client, and management networks had to share a network interface and therefore a port group. This limitation has been removed in this release and you can specify a separate network interface for each of the bridge, public, client, management, and container networks.

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 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 on a VCH corresponds to the default bridge network, or `docker0` interface, on a Docker host. Container application developers can use `docker network create` to create additional, user-defined bridge networks when they run containers. For information about default bridge networks and user-defined networks, see Docker container networking in the Docker documentation.

Options

The sections in this topic each correspond to an entry in the Configure Networks page of the Create Virtual Container Host wizard, and to the corresponding `vic-machine create` options.

Bridge Network

Aport group that container VMs use to communicate with each other.

Before you deploy a VCH, 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 Networking Requirements for VCH Deployment.

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 VCH networking options.
- Do not use the bridge network for any other VM workloads.

Create VCH Wizard

Select an existing port group from the **Bridge network** drop-down menu. It is **mandatory** to specify a bridge network.

vic-machine Option

`--bridge-network`, `-b`

You designate the bridge network by specifying the `vic-machine create --bridge-network` option.

The `--bridge-network` option is **mandatory** if you are deploying a VCH to vCenter Server.

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.

```
--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

Arange of IP addresses that additional bridge networks can use when container application developers use `docker network create` to create new user-defined networks. VCHs create these additional user-defined bridge networks by using IP address segregation within a set address range, so user-defined bridge networks do not require you to assign dedicated port groups. By default, all VCHs use the standard Docker range of 172.16.0.0/12 for additional user-defined networks. You can override the default range if that range is already in use in your network. You can reuse the same network address range across all VCHs.

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

Create VCH Wizard

If the default range of 172.16.0.0/12 is in use in your network, enter a new range as a CIDR. For example, enter `192.168.100.0/16`.

vic-machine Option

`--bridge-network-range`, `--bnr`

If the default range of 172.16.0.0/12 is in use in your network, specify a new range in the `--bridge-network-range` option.

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

If you specify an invalid value for `--bridge-network-range`, `vic-machine create` fails with an error.

What to Do Next

If you are using the Create Virtual Container Host wizard, stay on the Configure Networks page and Configure the Public Network settings.

Example `vic-machine` Command

This example `vic-machine create` command deploys a VCH that designates an existing port group named `vch1-bridge` as the bridge network. It specifies 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 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 host network.

Options

The sections in this topic each correspond to an entry in the Configure Networks page of the Create Virtual Container Host wizard, and to the corresponding `vic-machine create` options.

Public Network

You designate a specific port group for traffic from container VMs and the VCH to the Internet by specifying a public network when you deploy the VCH.

IMPORTANT:

- By default, VCHs that you deploy by using `vic-machine` use the standard VM Network for the public network. For deployments to vCenter Server clusters, it is strongly recommended that you create and use a port group for the public network. The VCH endpoint VM must be able to obtain an IP address on this port group. Using the VM Network instead of a port group prevents vSphere vMotion from moving the VCH endpoint VM between hosts in the cluster.
- You can use the same port group as the public network for multiple VCHs.
- The port group must exist before you create the VCH. For information about how to create a VMware vSphere Distributed Switch and a port group, see Create a vSphere Distributed Switch in the vSphere documentation.
- All hosts in a cluster should be attached to the port group. For information about how to add hosts to a vSphere Distributed Switch, see Add Hosts to a vSphere Distributed Switch in the vSphere documentation.
- The Create Virtual Container Host wizard only allows you to select port groups. You cannot select whole networks, for example the standard VM Network. Consequently, if you use the Create Virtual Container Host wizard, you must create a port group for the public network before you deploy the VCH. You cannot use `vic-machine configure` to change the public network setting after you deploy the VCH.

If you do not configure the client and management networks to use specific port groups, those networks use the settings that you specify for the public network.

Create VCH Wizard

Select an existing port group from the **Public network** drop-down menu.

NOTE: If you use the Create Virtual Container Host wizard, specifying a public network is **mandatory**.

vic-machine Option

`--public-network`, `--pn`

Aport 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.

```
--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.

Static IP Address

By default, vSphere Integrated Containers Engine uses DHCP to obtain an IP address for the VCH endpoint VM on the public network. You can optionally configure a static IP address for the VCH endpoint VM on the public network.

- You can only specify one static IP address on a given port group. 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. 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.

Create VCH Wizard

1. Select the **Static** radio button.
2. Enter an IP address with a network mask in the **IP Address** text box, for example `192.168.1.10/24`.

The Create Virtual Container Host wizard only accepts an IP address for the public network. You cannot specify an FQDN.

vic-machine Option

`--public-network-ip`, no short name

You specify addresses as IPv4 addresses with a network mask.

```
--public-network-ip 192.168.1.10/24
```

You can also specify addresses as resolvable FQDNs.

```
--public-network-ip=vch27-team-a.internal.domain.com
```

Gateway

The gateway to use if you specify a static IP address for the VCH endpoint VM on the public network. If you specify a static IP address on the public network, you must specify a gateway for the public network.

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

Create VCH Wizard

Enter the IP address of the gateway in the **Gateway** text box, for example `192.168.1.1`.

vic-machine Option

`--public-network-gateway`, no short name

Specify a gateway address as an IP address without a network mask in the `--public-network-gateway` option.

```
--public-network-gateway 192.168.1.1
```

DNS Server

ADNS server for the VCH endpoint VM to use on the public, client, and management networks.

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

Create VCH Wizard

Enter a comma-separated list of DNS server addresses in the **DNS server** text box, for example `192.168.10.10,192.168.10.11`.

If you are using the Create Virtual Container Host wizard and you set a static IP address on the public network, you must configure a DNS server.

vic-machine Option

`--dns-server`, None

You can specify `--dns-server` multiple times, to configure multiple DNS servers.

```
--dns-server 192.168.10.10
--dns-server 192.168.10.11
```

What to Do Next

If you are using the Create Virtual Container Host wizard, the bridge network and the public network are the only networks that it is mandatory to configure.

- To configure advanced network settings, remain on the Configure Networks page, and see the following topics:
  - Configure the Client Network
  - Configure the Management Network
  - Configure VCHs to Use Proxy Servers
  - Configure Container Networks
- If you have finished configuring the network settings, click **Next** to configure VCH Security settings.

Example `vic-machine` Command

This example `vic-machine create` command deploys a VCH that

- Directs public network traffic to an existing port group named `vch1-public`.
- Sets two DNS servers.

- Sets a static IP address and gateway for the VCH endpoint VM on the public network.
- Does not specify either of the `--management-network` or `--client-network` options. Consequently, management and client traffic also routes over `vch1-public` 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 vch1-public
--public-network-ip 192.168.1.10/24
--public-network-gateway 192.168.1.1
--dns-server 192.168.10.10
--dns-server 192.168.10.11
--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.

Options

The sections in this topic each correspond to an entry in the Configure Networks page of the Create Virtual Container Host wizard, and to the corresponding `vic-machine create` options.

Client Network

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.

- The port group must exist before you create the VCH. For information about how to create a VMware vSphere Distributed Switch and a port group, see Create a vSphere Distributed Switch in the vSphere documentation.
- All hosts in a cluster should be attached to the port group. For information about how to add hosts to a vSphere Distributed Switch, see Add Hosts to a vSphere Distributed Switch in the vSphere documentation.

If you do not specify this option, the VCH uses the public network for client traffic.

Create VCH Wizard

1. Expand the **Advanced** view.
2. Select an existing port group from the **Client network** drop-down menu.

vic-machine Option

`--client-network`, `--cln`

You designate the client network by specifying the `vic-machine create --client-network` option.

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

If you specify an invalid port group name, `vic-machine create` fails and suggests valid port groups.

Static IP Address

By default, vSphere Integrated Containers Engine uses DHCP to obtain an IP address for the VCH endpoint VM on the client network. You can optionally configure a static IP address for the VCH endpoint VM on the client network.

- You can only specify one static IP address on a given port group. If the client network shares a port group with the public

network, you can only specify a static IP address on the public network. All of the networks that share that port group use the IP address that you specify for the public network.

- If you set a static IP address for the VCH endpoint VM on the public network, you must specify the gateway address for the public network. If the client network is L2 adjacent to its gateway, you do not need to specify the corresponding gateway for the client network.
- If the client network shares a port group with the management network, and the public network does not use that port group, you can set a static IP address for the VCH endpoint VM on either of the client or management networks.
- If you assign a static IP address to the VCH endpoint VM on the client network, and you do not specify one of the TLS options, vSphere Integated Containers Engine uses this address as the Common Name with which to auto-generate trusted CA certificates. If you do not specify one of the TLS options, 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 VCH endpoint VM on the client network, vSphere Integated Containers Engine 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 VCH endpoint VM you must set a static IP address on the public network. In this case, vSphere Integated Containers Engine 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 if you had set a static IP on the client network.

You specify the address as an IPv4 address with a network mask.

Create VCH Wizard

1. Select the **Static** radio button.
2. Enter an IP address with a network mask in the **IP Address** text box, for example `192.168.3.10/24`.

The Create Virtual Container Host wizard only accepts an IP address for the client network. You cannot specify an FQDN.

vic-machine Option

`--client-network-ip`, no short name

You specify addresses as IPv4 addresses with a network mask.

```
--client-network-ip 192.168.2.10/24
```

You can also specify the address as a resolvable FQDN.

```
--client-network-ip=vch27-team-a.internal.domain.com
```

Gateway

The gateway to use if you specify a static IP address for the VCH endpoint VM on the client network.

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

Create VCH Wizard

Enter the IP address of the gateway in the **Gateway** text box, for example `192.168.2.1`.

You must enter a gateway address even if the client network is L2 adjacent to the gateway.

vic-machine Option

Specify a gateway address as an IP address without a network mask in the `--client-network-gateway` option. If the client network is L2 adjacent to its gateway, you do not need to specify the gateway.

```
--client-network-gateway 192.168.2.1
```

Routing Destination

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 the client network and that network is not L2 adjacent to its gateway, you must specify the routing destination for that network as a comma-separated list of CIDRs. For example, setting a routing destination of `192.168.2.0/24,192.168.128.0/24` informs the VCH that it can reach all of the vSphere management endoints that are in the ranges 192.168.2.0-255 and 192.168.128.0-192.168.128.255 by sending packets to the specified gateway.

Ensure that the address ranges that you specify include all of the systems that will connect to this VCH instance.

Create VCH Wizard

If you set a static IP address on the client network, optionally enter the routing destination as a comma-separated list of CIDRs in the **Routing destination** text box.

For example, enter `192.168.2.0/24,192.168.128.0/24`.

vic-machine Option

You specify the routing destination or destinations in a comma-separated list in the `--client-network-gateway` option, with the address of the gateway separated from the routing destinations by a colon (`:`).

```
--client-network-gateway 192.168.2.0/24,192.168.128.0/24:192.168.2.1
```

This example informs the VCH that it can reach all of the client network endoints that are in the ranges 192.168.2.0-255 and 192.168.128.0-192.168.128.255 by sending packets to the gateway at 192.168.2.1.

What to Do Next

If you are using the Create Virtual Container Host wizard, the bridge network and the public network are the only networks that it is mandatory to configure.

- To configure further advanced network settings, remain on the Configure Networks page, and see the following topics:
  - Configure the Management Network
  - Configure VCHs to Use Proxy Servers
  - Configure Container Networks
- If you have finished configuring the network settings, click **Next** to configure VCH Security settings.

Example `vic-machine` Command

This example `vic-machine create` command deploys a VCH with the following networking configuration:

- Directs public traffic to `vch1-public` and Docker API traffic to `vch1-client`.
- Sets two DNS servers for use by the public, management, and client networks.
- Sets a static IP address for the VCH endpoint VM on each of the public and client networks.
- Specifies the gateway for the public network.
- 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 vch1-public
--public-network-ip 192.168.1.10/24
--public-network-gateway 192.168.1.1
--client-network vch1-client
--client-network-ip 192.168.2.10/24
--dns-server 192.168.10.10
--dns-server 192.168.10.11
--thumbprint certificate_thumbprint
--name vch1
```

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.

Options

The sections in this topic each correspond to an entry in the Configure Networks page of the Create Virtual Container Host wizard, and to the corresponding `vic-machine create` options.

Management Network

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:

- The port group must exist before you create the VCH. For information about how to create a VMware vSphere Distributed Switch and a port group, see Create a vSphere Distributed Switch in the vSphere documentation.
- All hosts in a cluster should be attached to the port group. For information about how to add hosts to a vSphere Distributed Switch, see Add Hosts to a vSphere Distributed Switch in the vSphere documentation.
- 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.
- Container VMs communicate with the VCH 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 when the interactive console is enabled (`stdin` / `out` / `err`), and not for any other purpose.
- Ideally, use separate networks for the management network and 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 Asymmetric Routes.

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
```

If you do not specify a management network, the VCH uses the public network for management traffic.

Create VCH Wizard

1. Expand the **Advanced** view.
2. Select an existing port group from the **Management network** drop-down menu.

vic-machine Option

`--management-network`, `--mn`

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. If you specify an invalid port group name, `vic-machine create` fails and suggests valid port groups.

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

Static IP Address

By default, vSphere Integrated Containers Engine uses DHCP to obtain an IP address for the VCH endpoint VM on the management network. You can optionally configure a static IP address for the VCH endpoint VM on the management network.

- You can only specify one static IP address on a given port group. If the management network shares a port group with the public network, you can only specify a static IP address on the public network. All of the networks that share that port group use the IP address that you specify for the public network.
- If you set a static IP address for the VCH endpoint VM on the public network, you must specify the gateway address for the public network. If the management network is L2 adjacent to its gateway, you do not need to specify the corresponding gateway for the management network.
- 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.

You specify the address as an IPv4 address with a network mask.

Create VCH Wizard

1. Select the **Static** radio button.
2. Enter an IP address with a network mask in the **IP Address** text box, for example `192.168.3.10/24`.

The Create Virtual Container Host wizard only accepts an IP address for the management network. You cannot specify an FQDN.

vic-machine Option

`--management-network-ip`, no short name

You can specify addresses as IPv4 addresses with a network mask.

```
--management-network-ip 192.168.3.10/24
```

You can also specify addresses as resolvable FQDNs.

```
--management-network-ip=vch27-team-b.internal.domain.com
```

Gateway

The gateway to use if you specify a static IP address for the VCH endpoint VM on the management network.

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

Create VCH Wizard

Enter the IP address of the gateway in the **Gateway** text box, for example `192.168.3.1`.

You must enter a gateway address even if the client network is L2 adjacent to the gateway.

vic-machine Option

`--management-network-gateway`, no short name

Specify a gateway address as an IP address without a network mask. If the client network is L2 adjacent to its gateway, you do not need to specify the gateway.

```
--management-network-gateway 192.168.3.1
```

Routing Destination

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 the management network and that network is not L2 adjacent to its gateway, you must specify the routing destination for that network. You specify a routing destination as a comma-separated list of CIDRs.

For example, setting a routing destination of `192.168.3.0/24,192.168.128.0/24` 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 specified gateway.

Ensure that the address ranges that you specify include all of the systems that will connect to this VCH instance.

Create VCH Wizard

If you set a static IP address and gateway on the management network, optionally enter a comma-separated list of CIDRs in the **Routing destination** text box.

For example, enter `192.168.3.0/24,192.168.128.0/24`.

vic-machine Option

You specify the routing destination or destinations in a comma-separated list in the `--management-network-gateway` option, 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
```

This example 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 192.168.3.1.

```
--management-network-gateway 192.168.3.0/24,192.168.128.0/24:192.168.3.1
```

Asymmetric Routes

You can route incoming connections from ESXi hosts to VCHs over the public network rather than over the management network by configuring asymmetric routes.

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.

Create VCH Wizard

You cannot configure asymmetric routes in the Create Virtual Container Host wizard.

vic-machine Option

`--asymmetric-routes`, `--ar`

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

```
--asymmetric-routes
```

What to Do Next

If you are using the Create Virtual Container Host wizard, the bridge network and the public network are the only networks that it is mandatory to configure.

- To configure further advanced network settings, remain on the Configure Networks page, and see the following topics:
  - Configure the Client Network
  - Configure VCHs to Use Proxy Servers
  - Configure Container Networks
- If you have finished configuring the network settings, click **Next** to configure VCH Security settings.

Example `vic-machine` Command

This example `vic-machine create` command deploys a VCH with the following configuration:

- Directs public, client, and management traffic to networks `vch1-public`, `vch1-client`, and `vch1-management` respectively.
- Sets two DNS servers 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, client, and management networks.
- Specifies the gateway for the public network.
- Specifies a gateway and routing destinations for the client and management networks.

- 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.
- 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
--public-network vch1-public
--public-network-ip 192.168.1.10/24
--public-network-gateway 192.168.1.1
--client-network vch1-client
--client-network-ip 192.168.2.10/24
--client-network-gateway 192.168.2.0/24,192.168.128.0/24:192.168.2.1
--management-network vch1-mgmt
--management-network-ip 192.168.3.10/24
--management-network-gateway 192.168.3.0/24,192.168.128.0/24:192.168.3.1
--dns-server 192.168.10.10
--dns-server 192.168.10.11
--thumbprint certificate_thumbprint
--name vch1
--asymmetric-routes
```

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.

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 Running Virtual Container Hosts.

Options

The sections in this topic each correspond to an entry in the Configure Networks page of the Create Virtual Container Host wizard, and to the corresponding `vic-machine create` options.

HTTP Proxy

The address of the HTTP proxy server through which the VCH accesses image registries when using HTTP.

Create VCH Wizard

Enter the IP address or FQDN of an HTTP proxy in the **HTTP proxy** text box, for example `192.168.3.1`.

vic-machine Option

`--http-proxy`, no short name

Specify the address of the proxy server in the `--http-proxy` option, as either an FQDN or an IP address.

```
--http-proxy http://proxy.example.mycompany.org:80
```

HTTPS Proxy

The address of the HTTPS proxy server through which the VCH accesses image registries when using HTTPS.

Create VCH Wizard

Enter the IP address or FQDN of an HTTPS proxy in the **HTTPS proxy** text box, for example `192.168.3.1`.

vic-machine Option

`--https-proxy`, no short name

Specify the address of the proxy server in the `--https-proxy` option, as either an FQDN or an IP address.

```
--https-proxy https://proxy.example.mycompany.org:443
```

What to Do Next

If you are using the Create Virtual Container Host wizard, the bridge network and the public network are the only networks that it is mandatory to configure.

- To configure further advanced network settings, remain on the Configure Networks page, and see the following topics:
  - Configure the Client Network
  - Configure the Management Network
  - Configure Container Networks
- If you have finished configuring the network settings, click **Next** to configure VCH Security settings.

Example `vic-machine` Command

This example `vic-machine create` command deploys a VCH that accesses 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.example.mycompany.org:443
--name vch1
--thumbprint certificate_thumbprint
--no-tlsverify
```

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`.

Advantages of Container Networks

By using container networks, you can connect container VMs to any specific distributed port group or VMware 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. If containers use port mapping, the containers are accessible over a network via a port on the VCH endpoint VM. If the endpoint VM goes down for any reason, that network connection is no longer available. If you use container networks, containers have their own identity on the container network. Consequently, the network and the container have no dependency on the VCH endpoint VM for execution.
- **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.

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 Running Virtual Container Hosts.

Options

The sections in this topic each correspond to an entry in the Configure Networks page of the Create Virtual Container Host wizard, and to the corresponding `vic-machine create` options.

Container Network

Aport 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 create the VCH. For information about how to create a VMware vSphere Distributed Switch and a port group, see Create a vSphere Distributed Switch in the vSphere documentation.
- All hosts in a cluster should be attached to the port groups that you will use for mapped container networks. For information about how to add hosts to a vSphere Distributed Switch, see Add Hosts to a vSphere Distributed Switch in the vSphere documentation.
- Isolate the mapped container networks by using a separate VLAN for each network.
  - For information about how to assign a VLAN ID to a port group, see VMware KB 1003825.
  - For information about private VLAN, see VMware KB 1010691.
  - For information about VLAN tagging, see VMware KB 1003806.
- You cannot use the same port group as you use for the bridge network.
- You can create the port group on the same vSphere Distributed Switch as the port group that you use for the bridge network.
- If the port group that you specify does not support DHCP, you must configure an IP Address Range for the containers to use.
- 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.

You can specify multiple container networks to add multiple vSphere networks to Docker.

If you do not specify container networks, 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.

Create VCH Wizard

1. Expand the **Advanced** view.
2. Select an existing port group from the **Container network** drop-down menu.
3. In the **Label** text box, enter a descriptive name for use by Docker.

vic-machine Option

`--container-network` `--cn`

You use the `--container-network` option to specify a port group for the container network, and a descriptive name for the network for use by Docker.

```
--container-network port_group_name:descriptive_name
```

You can specify `--container-network` times to add multiple vSphere networks to Docker. If you specify an invalid port group name, `vic-machine create` fails and suggests valid port groups.

IP Address Range

The range of IP addresses that container VMs can use if the port group that you specify as a container network does not support DHCP. If you specify an IP address 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 specify an IP address range if container developers need to deploy containers with static IP addresses.
- If you specify a gateway for a container network but do not specify an IP address range, the IP range for container VMs is the entire subnet that you specify in the gateway.

Create VCH Wizard

1. Select the **Static** radio button.
2. Enter an IP address range or CIDR in the **IP Range** text box.
    - Example IP address range: `192.168.100.2-192.168.100.254`
    - Example CIDR: `192.168.100.0/24`

vic-machine Option

`--container-network-ip-range` , `--cnr`

When you specify the container network IP range, you use the port group that you specify in the `--container-network` option and specify either an IP address range or a CIDR:

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

```
--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.

Gateway

If the port group that you specify as a container network does not support DHCP, you must specify a gateway for the subnet of the container network.

Create VCH Wizard

Enter an IP address with a network mask in the **Gateway** text box, for example `192.168.100.10/24` .

vic-machine Option

`--container-network-gateway` , `--cng`

Specify the IP address and network mask for the gateway in the `--container-network-gateway` option. When you specify the container network gateway, you must use the port group that you specify in the `--container-network` option.

```
--container-network-gateway port_group_name:192.168.100.1/24
```

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.

DNS

If you specify an IP address range and gateway for a container network, it is recommended that you also specify one or more DNS servers.

Create VCH Wizard

Enter a comma-separated list of DNS server addresses in the **DNS server** text box, for example `192.168.100.10,192.168.100.11`.

vic-machine Option

`--container-network-dns`, `--cnd`

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

```
--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.

Firewall Policy

You can configure the trust level of container networks. The following table describes the levels of trust that you can set.

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.

If you do not set a trust level, the default level of trust is `published`. As a consequence, if you do not set a trust level, 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.

You can use `vic-machine configure --container-network-firewall` to change the trust level after deployment of the VCH. For information about configuring container network firewalls, see *Configure Container Network Settings* in Configure Running Virtual Container Hosts.

Create VCH Wizard

Leave the default policy of **Published**, or use the **Firewall policy** drop-down menu to select **Closed**, **Outbound**, **Peers**, or **Open**.

vic-machine Option

`--container-network-firewall` , `--cnf`

You specify the trust level in the `--container-network-firewall` option. You must use the port group that you specify in the `--container-network` option.

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

What to Do Next

If you are using the Create Virtual Container Host wizard, the bridge network and the public network are the only networks that it is mandatory to configure.

- Optionally click the **+** button to add more container networks to the VCH, and repeat the procedures for each additional container network.
- To configure further advanced network settings, remain on the Configure Networks page, and see the following topics:
  - Configure the Client Network
  - Configure the Management Network
  - Configure VCHs to Use Proxy Servers
- If you have finished configuring the network settings, click **Next** to configure VCH Security settings.

Example `vic-machine` Command

This example `vic-machine create` command deploys a VCH with the following configuration:

- Designates a port group and static IP address for the VCH endpoint VM on the public, client, and management networks.
- 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.
- 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
--public-network vch1-public
--public-network-ip 192.168.1.10/24
--public-network-gateway 192.168.1.1
--client-network vch1-client
--client-network-ip 192.168.2.10/24
--client-network-gateway 192.168.2.0/24,192.168.128.0/24:192.168.2.1
--management-network vch1-mgmt
--management-network-ip 192.168.3.10/24
--management-network-gateway 192.168.3.0/24,192.168.128.0/24:192.168.3.1
--dns-server 192.168.10.10
--dns-server 192.168.10.11
--container-network vic-containers:vic-container-network
--container-network-gateway vic-containers:192.168.100.1/24
--container-network-dns vic-containers:192.168.100.10
--container-network-dns vic-containers:192.168.100.11
--container-network-ip-range vic-containers:192.168.100.0/24
--container-network-firewall vic-containers:outbound
```

```
--thumbprint certificate_thumbprint
--name vch1
--asymmetric-routes
```

Virtual Container Host Security

By default, virtual container hosts (VCHs) authenticate connections from Docker API clients by using server and client TLS certificates. This configuration is commonly referred to as `tlsverify` in documentation about containers and Docker.

About TLS Certificates

Acertificate is made up of two parts:

- Apublic certificate part, that is distributed to anyone who needs it
- Aprivate key part, that is kept secret

Paired certificate and key files follow general naming conventions:

- `cert.pem` and `key.pem`
- `<prefix>.pem` and `<prefix>-key.pem`
- `<prefix>-cert.pem` and `<prefix>-key.pem`

For general information about TLS certificates, see https://en.wikipedia.org/wiki/Transport_Layer_Security.

Certificate Usage in Docker

There are four certificates in use in a Docker `tlsverify` configuration:

- **(1)** Aclient certificate, held by the Docker client.
- **(2)** Aserver certificate, held by the server, which in a VCH is the Docker API endpoint.
- **(3)** Acertificate authority (CA), that signs the server certificate.
- **(4)** Another CA, that signs the client certificate and is held by the server.

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 container developers provide explicitly by using the `--tlscacert` option when they run Docker commands. As a part of this validation, the Common Name (CN) in the server certificate must match the name or address of the system from which the Docker client accesses the server. The server certificate must explicitly state at least one of the following in the CN:

- The FQDN of the system from which the Docker client communicates with the server
- The IP address of the system from which the Docker client communicates with the server
- Awildcard domain that matches all of the FQDNs in a specific subdomain.

If the server certificate includes a wildcard domain, all of the systems in that domain can connect to the server. For an example of a domain wildcard, see https://en.wikipedia.org/wiki/Wildcard_certificate#Example.

`DOCKER_CERT_PATH`

Docker clients search for certificates in the `DOCKER_CERT_PATH` location on the system on which the Docker client is running. Docker requires certificate files to have the following names if they are to be consumed automatically from `DOCKER_CERT_PATH`:

File Name	Description

`cert.pem` , `key.pem`	Client certificate **(1)** and private key. client certificate.
`server-cert.pem` , `server-key.pem`	Server certificate **(2)**
`ca.pem`	Public portion of the certificate authority that signed the server certificate **(3)**. Allows the server to confirm that a client is authorized.

For information about how to provide certificates to Docker clients, see Configure the Docker Client for Use with vSphere Integrated Containers.

Virtual Container Host Security Options

As a convenience, vSphere Integrated Containers Engine can optionally generate client **(1)** and server **(2)** certificates. It can also automatically generate one CA that serves as both **(3)** and **(4)**. If you use an automatically generated CA, vSphere Integrated Containers Engine uses that CA to sign both of the client and server certificates. This means that you must provide to the Docker client both the client certificate and the public part of the CA, so that the client can trust the server. You must provide a client certificate and CA for every VCH that a Docker client connects to.

Rather than using an automatically generated CA, in a production deployment you would normally use custom CAs. In this case:

- **(3)** is usually signed by a company certificate that is rooted by a public or corporate trust authority, for example Verisign.
- **(4)** can be unique per client or group of clients. Using the same CA for a group of clients allows each client to have a unique certificate, but allows the group to be authorized as a whole. For example, you could use one CA per VCH, multiple CAs per VCH, or one CA per group of VCHs.

When you deploy a VCH, you must specify the level of security that applies to connections from Docker clients to the Docker API endpoint in the VCH, and whether to use automatically generated or custom certificates, or a combination of both.

Supported Configurations

You can use all automatically generated certificates, all custom certificates, or a combination of both.

NOTE: The Create Virtual Container Host wizard in the vSphere Client does not support automatically generated CA or client certificates. To use automatically generated CA and client certificates, you must use the `vic-machine` CLI utility to deploy VCHs.

The following table provides a summary of the configurations that vSphere Integrated Containers Engine supports, and whether you can implement those configurations in the Create Virtual Container Host wizard in the vSphere Client.

Configuration	Available in vSphere Client?	Examples
Auto-generated server certificate + auto-generated CA + auto-generated client certificate	No	Example
Auto-generated server certificate + custom CA	Yes	Example
Custom server certificate + custom CA	Yes	Example
Custom server certificate + auto-generated CA + auto-generated client certificate	No	Example
Auto-generated server certificate + no client verification	Yes	Example
Custom server certificate + no client verification	Yes	Example
No server or client certificate verification	Yes	Example

The following topics describe how to achieve all of the configurations listed in the table above, by using either the Create Virtual Container Host wizard or the `vic-machine` CLI, or both.

- Virtual Container Host Certificate Options

- Disable Client Verification

The Examples column in the table provides direct links to the relevant example in these topics.

Registry Access

In addition to configuring the level of security to apply to connections from Docker clients to VCHs, you must also configure the level of security to apply to connections from VCHs to registry servers. For example, to use vSphere Integrated Containers Registry, you must configure VCHs accordingly when you deploy them.

For information about configuring VCHs to use registry servers, see Configure Registry Access.

Virtual Container Host Certificate Options

This topic describes the different certificate options that you use when deploying virtual container hosts (VCHs) that implement verification of client certificates. It provides examples of how to combine the options to achieve different configurations.

For information about how to deploy VCHs that do not verify connections from clients, see Disable Client Authentication.

Automatically Generated Certificates

As a convenience, vSphere Integrated Containers Engine provides the option of automatically generating a server certificate for the Docker API endpoint in the 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. To use more finely configured certificates, use custom server certificates.

VCHs accept client certificates if they are signed by a CA that you can optionally provide to the VCH. Alternatively, you can configure a VCH so that vSphere Integrated Containers Engine creates a Certificate Authority (CA) certificate that it uses to automatically generate and sign a single client certificate.

NOTE: The Create Virtual Container Host wizard in the vSphere Client does not support automatically generated CA or client certificates. To use automatically generated CA and client certificates, you must use the `vic-machine` CLI utility to create the VCH.

Custom Certificates

To exercise fine control over the certificates that VCHs use, you must obtain or generate custom certificates yourself before you deploy a VCH. 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. For information about how to create custom certificates for use with Docker, see Protect the Docker daemon socket in the Docker documentation.

Custom certificates must meet the following requirements:

- You must use an X.509 server certificate.
- Server certificates should have the following certificate usages:
  - `KeyEncipherment`
  - `DigitalSignature`
  - `KeyAgreement`

    - `ServerAuth`
- Server keys must not be encrypted.

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.

You can deploy a VCH to use custom certificates in combination with auto-generated certificates, as demonstrated in the Examples.

How to Connect to VCHs with Client Verification

After deployment, the Docker API for VCHs that implement client verification is accessible at https://*vch_dnsname*.example.org:2376.

You must provide the automatically generated `cert.pem`, `key.pem`, and `ca.pem` files to all container developers who need to connect Docker clients to the VCHs.

- If you deploy VCHs by using the Create Virtual Container Host wizard, you must create the `cert.pem` and `key.pem` files manually, using the custom `ca.pem` file to sign them.
- If you deploy VCHs by using `vic-machine`, you can either use the auto-generated client certificate, or by use a client certificate that you create and sign manually.

For example, you can access information about a VCH with client verification by running the following command in the Docker client:

```
docker -H vch_dnsname.example.org.example.org:2376
--tlsverify
--tlscacert=path_to_cert_folder/ca.pem
--tlscert=path_to_cert_folder/cert.pem
--tlskey=path_to_cert_folder/key.pem
info
```

Options

The following sections each correspond to an entry in the Security page of the Create Virtual Container Host wizard if you select the **Docker API Access** tab. Each section also includes a description of the corresponding `vic-machine create` option.

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`.

Common Name (CN)

The IP address, FQDN, or a domain wildcard, for the client system or systems that connect to this VCH, to embed in an automatically generated server certificate.

NOTE: Specifying an FQDN or wildcard assumes that there is a DHCP server offering IP addresses on the client network, and that those addresses have corresponding DNS entries such as `dhcp-a-b-c.example.com`.

Create VCH Wizard

1. For **Source of certificates**, select the **Auto-generate** radio button.
2. In the **Common Name (CN)** text box, enter the IP address, FQDN, or a domain wildcard for the client systems that connect to this VCH.

vic-machine Option

`--tls-cname`, no short name

The IP address, FQDN, or a domain wildcard, for the client system or systems that connect to this VCH.

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

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

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

- On the system on which you run `vic-machine`, checks for an existing server certificate in either a folder that has the same name as the VCH that you are deploying, or in a location that you can optionally specify in the `--tls-cert-path` option. If a valid server certificate exists that includes the same Common Name attribute as the one that you specify in `--tls-cname`, `vic-machine create` reuses that certificate. Reusing certificates allows you to delete and recreate VCHs for which you have already distributed the client 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 in the location from which you run `vic-machine`, or creates a folder in a location that you specify in the `--tls-cert-path` option.
- If valid certificates do not already exist, `vic-machine create` automatically creates a CA and uses that CA to sign and create a client certificate and to sign the server certificate. The CA and client certificate allow the server to confirm the identity of the client. The `vic-machine create` command creates the following 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:

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

For more information about how to connect Docker clients to VCHs, see Configure the Docker Client for Use with vSphere Integrated Containers.

NOTE: 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`. For information about setting a static IP address on the client network, see Configure the Client Network.

Organization (O)

Alist of identifiers to record in automatically generated server certificates, to add basic descriptive information to the server certificate. This information is visible to clients if they inspect the server certificate.

Create VCH Wizard

1. For **Source of certificates**, select the **Auto-generate** radio button.
2. In the **Organization (O)** text box, leave the default setting of the VCH name, or enter a different organization identifier.

vic-machine Option

`--organization`, no short name

If you specify `--tls-cname`, you can optionally specify `--organization`. If not specified, `vic-machine create` uses the name of the VCH as the `organization` value.

NOTE: If you specify a static IP address on the client network, the `client-ip-address` is used for `CommonName` but not for `Organisation`.

```
--organization my_organization_name
```

Certificate Key Size

The size of the key for vSphere Integrated Containers Engine to use when it creates auto-generated certificates. It is not recommended to use key sizes of less than the default of 2048 bits.

Create VCH Wizard

1. For **Source of certificates**, select the **Auto-generate** radio button.
2. In the **Certificate key size** text box, leave the default setting of 2048 bits, or enter a higher value.

vic-machine Option

`--certificate-key-size`, `--ksz`

If you specify `--tls-cname`, you can optionally specify `--certificate-key-size`. If not specified, `vic-machine create` creates keys with default size of 2048 bits.

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

Certificate Path

If you are using the Create Virtual Container Host wizard, the certificate path setting is not applicable.

vic-machine Option

`--tls-cert-path`, none

By default `--tls-cert-path` is a folder in the current directory on the system on which you are running `vic-machine`. The certificate folder takes its name from the VCH name that you specify in the `--name` option. If specified, `vic-machine create` checks in the `--tls-cert-path` folder 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 the folder you specify 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` automatically generates certificates. Automatically generated certificates are saved in the `--tls-cert-path` folder with the standard names. `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.

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

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

Select CA Certificate PEM File

The public portion of a CA that vSphere Integrated Containers Engine uses to validate client certificates. The client certificates are used as credentials for access to the Docker API running in the VCH. This does not need to be the same CA as you use to sign the server certificate, if you use a custom CA to sign server certificates. You can specify multiple CAs.

Create VCH Wizard

If you use the Create Virtual Container Host wizard and you do not disable client verification, it is **mandatory** to upload at least one custom CA file. The Create Virtual Container Host wizard does not support automatic generation of CA files.

1. Leave the **Client Certificates** switch in the green ON position, to enable verification of client certificates.
2. For **Select the certificate pem** file, click **Select** and navigate to an existing `ca.pem` file for the custom CA that you use to sign client certificates.
3. Optionally click **Select** again to upload additional CAs.

vic-machine Option

`--tls-ca`, `--ca`

Specify the path to an existing `ca.pem` file for the custom CA that you use to sign client certificates. Include the filename in the path. You can specify `--tls-ca` multiple times. If not specified, or if no CA exists in the certificate folder on the machine on which you run `vic-machine`, `vic-machine create` automatically generates a CA.

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

Server Certificate

A custom X.509 server certificate for the VCH if you do not use an automatically generated server certificate. This certificate identifies the VCH endpoint VM both to Docker clients and to browsers that connect to the VCH Admin portal. For information about the requirements for server certificates, see Custom Certificates above.

Create VCH Wizard

1. For **Source of certificates**, select the **Existing** radio button.

2. For **Server certificate**, click **Select** and navigate to an existing `server-cert.pem` file.

vic-machine Option

`--tls-server-cert`, no short name

This option is **mandatory** if you use custom server certificates, rather than auto-generated certificates. If you do not use an automatically generated server certificate, use this option in combination with the `--tls-server-key` option, that provides the path to the private key file for the custom server certificate. Include the name of the certificate file in the path.

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.

```
--tls-server-cert path_to_certificate_file/certificate_file_name.pem
```

Server Private Key

The private key file to use with a custom server certificate. This option is mandatory if you specify a custom X.509 server certificate. Include the name of the key file in the path.

IMPORTANT: The key must not be encrypted.

Create VCH Wizard

1. For **Source of certificates**, select the **Existing** radio button.
2. For **Server private key**, click **Select** and navigate to an existing `server-key.pem` file.

vic-machine Option

`--tls-server-key`, no short name

Use this option in combination with the `--tls-server-cert` option. Include the name of the key file in the path.

```
--tls-server-key path_to_key_file/key_file_name.pem
```

Examples

This section provides examples of the combinations of options to use in in the **Docker API Access** tab in the Security page of the Create Virtual Container Host wizard and in `vic-machine create`, for the different security configurations that you can implement when using automatically generated and custom certificates.

- Automatically Generate Server, Client, and CA Certificates
- Automatically Generate Server Certificates and Use Custom CA and Client Certificates
- Use Custom Server and Client Certificates and a Custom CA
- Use a Custom Server Certificate and Automatically Generate a CA and Client Certificate

Automatically Generate Server, Client, and CA Certificates

This example deploys a VCH with the following security configuration:

- Uses an automatically generated server certificate
- Implements client authentication with an automatically generated client certificate
- Uses an automatically generated CA to sign the client and server certificates

Create VCH Wizard

The Create Virtual Container Host wizard does not support automatic generation of CAs and client certificates.

`vic-machine` Command

This example `vic-machine create` command deploys a VCH with the following configuration:

- Provides a wildcard domain, `*.example.org`, for the client systems that will connect to this VCH, for use as the Common Name in the server 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 an empty folder in which to save the auto-generated certificates.
- Sets the certificate's `organization` (`O`) field to `My Organization`.
- Generates a certificate with a key size of 3072 bits.

```
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
--organization 'My Organization'
--certificate-key-size 3072
--thumbprint certificate_thumbprint
--name vch1
```

Result

When you run this command, `vic-machine create` performs the following operations:

- Checks for existing certificates in the folder that you specified in `--tls-cert-path`.
- No existing `server-cert.pem`, `server-key.pem`, or `ca.pem` certificates are present in the folder, so `vic-machine` automatically generates them and saves them in the certificate folder.
- Automatically generates a client certificate and saves it in the certificate folder.
- Uses the automatically generated CA to sign the server and client certificates.
- Automatically generates a `.pfx` certificate to allow access to the VCH Admin portal for this VCH.
- Generates an `env` file that includes the environment variables with which to configure Docker clients that connect to this VCH.

Automatically Generate Server Certificates and Use a Custom CA for Client Certificates

This section provides examples of using both the Create Virtual Container Host wizard and `vic-machine create` to deploy a VCH with the following security configuration:

- Uses an automatically generated server certificate.
- Uploads the CA certificate for a custom CA that you use to sign custom client certificates.
- Implements client authentication with a custom client certificate.

Prerequisites

- Create or obtain the `ca.pem` file for the CA that you use to sign client certificates.
- Use the custom CA to create and sign client certificates.

Create VCH Wizard

1. Leave the **Enable secure access to this VCH** switch in the green ON position.
2. For **Source of certificates**, select the **Auto-generate** radio button.
3. In the **Common Name (CN)** text box, enter the IP address, FQDN, or a domain wildcard for the client systems that connect to this VCH.
4. In the **Organization (O)** text box, leave the default setting of the VCH name, or enter a different organization identifier.
5. In the **Certificate key size** text box, leave the default setting of 2048 bits, or enter a higher value.
6. Leave the **Client Certificates** switch in the green ON position, to enable verification of client certificates.
7. Click **Select** and navigate to an existing `ca.pem` file for the custom CA that you use to sign client certificates.
8. Optionally click **Select** again to upload additional CAs.

`vic-machine` Command

This example `vic-machine create` command deploys a VCH with the following configuration:

- Provides a wildcard domain `*.example.org` as the FQDN for the client systems that connect to the VCH, for use as the Common Name in the automatically generated server certificate.
- Specifies the folder in which to save auto-generated certificates in the `--tls-cert-path` option.
- Sets the certificate's `organization` (`O`) field to `My Organization` .
- Generates certificates with a key size of 3072 bits.
- Provides the path to an existing `ca.pem` file for the CA that you use to sign client 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
--organization 'My Organization'
--certificate-key-size 3072
--tls-ca path_to_folder/ca.pem
--thumbprint certificate_thumbprint
--name vch1
```

Result

When you run this command, `vic-machine create` performs the following operations:

- Checks for existing certificates in the folder that you specified in `--tls-cert-path` .
- Since no existing `server-cert.pem` or `server-key.pem` certificates are present in the folder, `vic-machine` automatically generates them.
- Automatically generates a client certificate, signs it with the custom CA, and saves it in the certificate folder. However, you can use any client certificate that is signed by the CA that you provided to the VCH.

You must provide the custom `cert.pem` , `key.pem` , and `ca.pem` files to all container developers who need to connect Docker clients to this VCH.

Use a Custom Server Certificate and a Custom CA for Client Certificates

This example deploys a VCH with the following security configuration:

- Uses a custom server certificate.
- Implements client authentication with a custom client certificate.
- Uses a custom CA.

Prerequisite

Create or obtain server and client certificates, that you sign by using a custom CA.

Create VCH Wizard

1. Leave the **Enable secure access to this VCH** switch in the green ON position.
2. For **Source of certificates**, select the **Existing** radio button.
3. For **Server certificate**, click **Select** and navigate to an existing `server-cert.pem` file.
4. For **Server private key**, click **Select** and navigate to an existing `server-cert.pem` file.
5. Leave the **Client Certificates** switch in the green ON position, to enable verification of client certificates.
6. Click **Select** and navigate to an existing `ca.pem` file for the custom CA that you use to sign client certificates.
7. Optionally click **Select** again to upload additional CAs.

`vic-machine` Command

This example `vic-machine create` command provides the paths relative to the current location of the `*.pem` files for the custom server certificate and key files, and a custom CA.

```
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 path_to_folder/certificate_file.pem
--tls-server-key path_to_folder/key_file.pem
--tls-ca path_to_folder/ca.pem
--name vch1
--thumbprint certificate_thumbprint
```

Result

When you run this command, `vic-machine create` performs the following operations:

- Uploads the custom server certificate and key to the VCH.
- Uploads the CA to the VCH, to verify client certificates that have been signed by that CA.

You must provide the custom `cert.pem`, `key.pem`, and `ca.pem` files to all container developers who need to connect Docker clients to this VCH.

Use a Custom Server Certificate and Automatically Generate a CA and Client Certificate

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.

Prerequisite

Create or obtain a custom server certificate, that you sign by using a custom CA.

Create VCH Wizard

The Create Virtual Container Host wizard does not support automatic generation of CAs and client certificates.

`vic-machine` Command

This example `vic-machine create` command deploys a VCH with the following configuration:

- 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 auto-generated client 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
```

Result

When you run this command, `vic-machine create` performs the following operations:

- Uploads the `server-cert.pem` or `server-key.pem` to the VCH.
- Automatically generates a CA.
- Uses the CAto create and sign a client certificate.

After deployment, the Docker API for this VCH is accessible at https://dhcp-a-b-c.example.org:2376.

You must provide the automatically generated `cert.pem`, `key.pem`, and `ca.pem` file to all container developers who need to connect Docker clients to this VCH.

What to Do Next

If you are using the Create Virtual Container Host wizard, stay on the Security page and select the **Registry Access tab** to Configure Registry Access.

If you do not require access to a registry server, click **Next** to configure the Operations User.

Disable Client Verification

To deploy a virtual container host (VCH) that does not restrict access to the Docker API but still encrypts communication between clients and the VCH, you can disable client certificate verification. You can also completely disable TLS authentication and encryption on both the client and server sides.

Options

The following sections each correspond to an entry in the Security page of the Create Virtual Container Host wizard if you select the **Docker API Access** tab. Each section also includes a description of the corresponding `vic-machine create` option.

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`.

Disable Client Certificate Verification

Disabling client certificate verification prevents the use of CAs for client authentication. VCHs still require a server certificate if you disable client authentication. You can either supply a custom server certificate or have vSphere Integrated Containers Engine automatically generate one. If you disable client authentication, there is no access control to the VCH from Docker clients, but connections remain encrypted.

If you disable client certificate verification, container developers run Docker commands against the VCH with the `--tls` option. The `DOCKER_TLS_VERIFY` environment variable must not be set. Note that setting `DOCKER_TLS_VERIFY` to 0 or `false` has no effect. For more information about how to connect Docker clients to VCHs, see Configure the Docker Client for Use with vSphere Integrated Containers.

For example, you can access information about a VCH that uses a server certificate but does not perform client verification by running the following command in the Docker client:

```
docker -H vch_dnsname.example.org.example.org:2376 --tls info
```

Create VCH Wizard

Toggle the **Client Certificates** switch to the gray OFF position.

vic-machine Option

`--no-tlsverify`, `--kv`

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

- If you do not specify `--tls-server-cert` and `--tls-server-key`, automatically generates a self-signed server certificate.
- If you specify `--tls-cert-path`, saves the server certificate in the location that you specify.

- 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.

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

```
--no-tlsverify
```

Disable Secure Access

You can completely disable authentication of connections between Docker clients and the VCH. VCHs use neither client nor server certificates. Any Docker client can connect to the VCH if you disable TLS authentication and connections are not encrypted.

IMPORTANT: Disabling secure access is for testing purposes only. Do not disable secure access in production environments.

If you completely disable secure access to the VCH, container developers connect Docker clients to the VCH over HTTP on port 2375, instead of over HTTPS on port 2376. They do not need to specify any TLS options in the Docker command.

For example, you can access information about a VCH that does not use a server or client certificate by running the following command in the Docker client:

```
docker -H vch_dnsname.example.org.example.org:2375 info
```

Create VCH Wizard

At the top of the Security page, toggle the **Enable secure access to this VCH** switch to the gray OFF position.

vic-machine Option

`--no-tls`, `-k`

Run `vic-machine create` with the `--no-tls` option and no other security options. The `--no-tls` option takes no arguments.

```
--no-tls
```

Examples

This section provides examples of the options to use in the **Docker API Access** tab in the Security page of the Create Virtual Container Host wizard and in `vic-machine create`, to create VCHs that disable client certificate verification and that disable secure access completely.

- Automatically Generate a Server Certificate and Disable Client Certificate Verification
- Use Custom Server Certificates and Disable Client Certificate Verification
- Disable Secure Access

Automatically Generate a Server Certificate and Disable Client Certificate Verification

This example deploys a VCH with the following security configuration.

- Uses an automatically generated server certificate.
- Disables client certificate authentication.

Create VCH Wizard

1. Leave the **Enable secure access to this VCH** switch in the green ON position.
2. For **Source of certificates**, select the **Auto-generate** radio button.
3. In the **Common Name (CN)** text box, enter the IP address, FQDN, or a domain wildcard for the client systems that connect to this VCH.
4. In the **Organization (O)** text box, leave the default setting of the VCH name, or enter a different organization identifier.
5. In the **Certificate key size** text box, leave the default setting of 2048 bits, or enter a higher value.
6. Toggle the **Client Certificates** switch to the gray OFF position.

`vic-machine` Command

This example `vic-machine create` command deploys a VCH that specifies `--no-tlsverify` to disable client 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 vch1
--thumbprint certificate_thumbprint
--tls-cert-path path_to_certificate_folder
--no-tlsverify
```

Result

When you run this command, `vic-machine create` performs the following operations:

- Because no other security options are specified, automatically generates a server certificate.
- Saves the server certificate in the location that you specify in `--tls-cert-path`.
- Does not generate a client certificate or CA.

You do not need to provide any certificates to container developers. However, you can provide the generated `env` file, with which they can set environment variables in their Docker client.

Use Custom Server Certificates and Disable Client Certificate Verification

This example deploys a VCH with the following security configuration:

- Uses a custom server certificate.
- Disables client certificate authentication.

Create VCH Wizard

1. Leave the **Enable secure access to this VCH** switch in the green ON position.
2. For **Source of certificates**, select the **Existing** radio button.
3. For **Server certificate**, click **Select** and navigate to an existing `server-cert.pem` file.
4. For **Server private key**, click **Select** and navigate to an existing `server-cert.pem` file.
5. Toggle the **Client Certificates** switch to the gray off position.

`vic-machine` Command

This example `vic-machine create` command provides the paths relative to the current location of the `*.pem` files for the custom server certificate and key files, and specifies the `--no-tlsverify` option to disable client 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 vch1
--thumbprint certificate_thumbprint
--tls-server-cert ../some/relative/path/certificate_file.pem
--tls-server-key ../some/relative/path/key_file.pem
--no-tlsverify
```

Result

When you run this command, `vic-machine create` performs the following operations:

- Uploads the custom server certificate and key to the VCH.
- Does not generate a client certificate or CA.

You do not need to provide any certificates to container developers. However, you can provide the generated `env` file, with which they can set environment variables in their Docker client.

Disable Secure Access

This example completely disables secure access to the VCH. All communication between the VCH and Docker clients is insecure and unencrypted.

Create VCH Wizard

At the top of the Security page, toggle the **Enable secure access to this VCH** switch to the gray OFF position. All other security options are disabled.

`vic-machine` Command

This example deploys a VCH that specifies `--no-tls` to disable client and server 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 vch1
--thumbprint certificate_thumbprint
--no-tls
```

Result

When you run this command, `vic-machine create` does not generate any certificates. Connections to the VCH are possible by using HTTP rather than HTTPS.

You do not need to provide any certificates to container developers. No `env` file is generated, as there are no environment variables to set.

What to Do Next

If you are using the Create Virtual Container Host wizard, stay on the Security page and select the **Registry Access tab** to Configure Registry Access.

If you do not require access to a registry server, click **Next** to configure the Operations User.

Configure Registry Access

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 virtual container hosts (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.

Obtain the vSphere Integrated Containers Registry Certificate

To configure a VCH so that it can connect to vSphere Integrated Containers Registry, you must obtain the registry certificate and pass it to the 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.

Procedure

1. Go to http://*vic_appliance_address*, click the link to **Go to the vSphere Integrated Containers Management Portal**, and log in 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 in the management portal.

2. Go to **Administration** > **Configuration**, and click the link to download the **Registry Root Cert**.

Options

The following sections each correspond to an entry in the Security page of the Create Virtual Container Host wizard if you select the **Registry Access** tab. The **Registry Access** tab is only visible when the **Enable secure access to this VCH** is set to the green ON position. Each section also includes a description of the corresponding `vic-machine create` option.

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`.

Whitelist Registry Mode

You can restrict the registries to which a VCH allows access by setting the VCH in whitelist registry mode. You can allow VCHs to access multiple registries. In whitelist mode, users can only access those registries that you have specified. Users cannot access any registries that are not in the whitelist, even if they are public registries, such as Docker Hub. You can configure a VCH so that it includes both secure and insecure registries in its whitelist.

You can specify whitelisted registries in the following formats:

- IP addresses or FQDN to identify individual registry instances. During deployment, vSphere Integrated Containers Engine 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 vSphere Integrated Containers Engine does not validate CIDR defined ranges during deployment.
- Wildcard domains, for example, *.example.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. Anumeric IP address causes VCHs to perform a reverse DNS lookup to validate against that wild card domain. Note that vSphere Integrated Containers Engine does not validate wildcard domains during deployment.

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 that the VCH holds, you must provide the CAcertificate for that registry to the VCH. If the VCH is running in whitelist mode, you must also add that registry to the whitelist.

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

Whitelisting Insecure Registries

You can add registries that you designate as insecure registries to the whitelist. If you designate a registry as an insecure registry, VCHs do not verify the certificate of that registry when they pull images.

If you add a registry to the whitelist, but you do not specify that registry as an 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.

Create VCH Wizard

1. Select the **Registry Access** tab.
2. Toggle the **Whitelist registry mode** switch to the green ON position.
3. In the **Whitelist registries** text box, enter the IP address or FQDN and port number for the registry server, or enter a wildcard domain.
4. Select **Secure** or **Insecure** from the drop-down menu, to specify whether the registry requires a certificate for access.
5. Optionally click **+** to add more registries to the whitelist.

If you select **Secure** for a given registry, you must also provide a certificate for that registry. For information about providing certificates, see Additional Registry Certificates below.

vic-machine Option

`--whitelist-registry` , `--wr`

If you specify `--whitelist-registry` at least once when you run `vic-machine create` , the VCH runs in whitelist mode.

You use `--whitelist-registry` in combination with the `--registry-ca` or `--insecure-registry` options, to either provide the registry certificate or to allow insecure access to that registry. If you specify a registry as an insecure registry but you do not specify this registry in the whitelist, vSphere Integrated Containers Engine automatically adds the registry to the whitelist only if whitelist mode is activated by specifying at least one other registry in `--whitelist-registry` .

```
--whitelist-registry registry_address
--registry-ca path_to_ca_cert
```

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

Insecure Registry Access

If you designate a registry server as an insecure registry, the VCH does not verify the certificate of that registry when it pulls images. Insecure registries are not recommended in production environments.

If you authorize a VCH to connect to an insecure registry server, the VCH first attempts to access the registry server via HTTPS, then attempts to connect with 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 designate vSphere Integrated Containers Registry instances as insecure registries. Connections to vSphere Integrated Containers Registry always require HTTPS and a certificate.

Create VCH Wizard

1. Leave the **Whitelist registry mode** switch in the gray OFF position.

   If you are using the Create Virtual Container Host wizard and you activate whitelist registry mode, you designate registries as insecure when you add them to the whitelist.

2. In the **IP or FQDN** text box under **Insecure registry access**, enter the IP address or FQDN for the registry server to designate as insecure.

3. If the registry server listens on a specific port, add the port number in the **Port** text box.
4. Optionally click the **+** button to add more registries to the list of insecure registries to which this VCH can connect.

vic-machine Option

`--insecure-registry`, `--dir`

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.

Usage:

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

Additional Registry Certificates

If the VCH is to connect to secure registries, you must provide a CAcertificate that can validate the server certificate of that registry. You can specify multiple CAcertificates for different registries to allow a VCH to connect to multiple secure registries.

IMPORTANT: You must use this option to allow a VCH to connect to a vSphere Integrated Containers Registry instance. vSphere Integrated Containers Registry does not permit insecure connections.

Create VCH Wizard

1. Under **Additional registry certificates**, click **Select** and navigate to an existing certificate file a registry server instance.
2. Optionally click **Select** again to upload additional CAs.

vic-machine Option

`--registry-ca`, `--rc`

You can specify `--registry-ca` multiple times to allow a VCH to connect to multiple secure registries.

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

Examples

This section provides examples of the combinations of options to use in in the **Registry Access** tab in the Security page of the Create Virtual Container Host wizard and in `vic-machine create` commands.

- Authorize Access to a Whitelist of Secure and Insecure Registries
- Authorize Access to Secure and Insecure Private Registries

Authorize Access to a Whitelist of Secure and Insecure Registries

This example deploys a VCH with the following configuration:

- Adds to the whitelist:
  - A single vSphere Integrated Containers Registry instance that is running at 10.2.40.40:443
  - All registries running in the range 10.2.2.1/24
  - All registries in the domain *.example.com
  - A single instance of an insecure registry running at 192.168.100.207, that is not in the IP range or domain specified previously.
- Provides the CA certificate for the vSphere Integrated Containers Registry instance in the whitelist.

Prerequisite

Follow the instructions in Obtain the vSphere Integrated Containers Registry Certificate to obtain the certificate file for your vSphere Integrated Containers Registry instance.

Create VCH Wizard

1. Leave the **Enable secure access to this VCH** switch in the green ON position.
2. Select the **Registry Access** tab.
3. Toggle the **Whitelist registry mode** switch to the green ON position.
4. In the **Whitelist registries** text box, enter `10.2.40.40:443` to add the vSphere Integrated Containers Registry instance to the whitelist.
5. Leave the drop-down menu for this registry set to **Secure**.
6. Click the **+** button, and enter `10.2.2.1/24` to add all registries that are running in that range to the whitelist.
7. Select **Insecure** from the drop-down menu to designate all registries in that range as insecure.
8. Click the **+** button, enter `*.example.com` to add all registries that are running in that domain to the whitelist, and select **Insecure** to designate those registries as insecure.
9. Click the **+** button, enter `192.168.100.207` to add the standalone registry to the whitelist, and select **Insecure** to designate those registries as insecure.
10. Under **Additional registry certificates**, click **Select** and navigate to the CA certificate file for the vSphere Integrated Containers Registry instance that is running at 10.2.40.40:443.

`vic-machine` Command

This example `vic-machine create` command deploys a VCH that uses the `--whitelist-registry`, `--registry-ca`, and `--insecure-registry` options to add a range of registries to its whitelist.

```
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=*.example.com
--registry-ca=/home/admin/mycerts/ca.crt
--insecure-registry=192.168.100.207
--name vch1
--thumbprint certificate_thumbprint
--no-tlsverify
```

Result

The VCH can only access the registries in the IP ranges and domain specified, as well as the standalone insecure registry at 192.168.100.207 and the vSphere Integrated Containers Registry instance at 10.2.40.40:443. It cannot access any other registries, even public registries like Docker Hub.

Authorize Access to Secure and Insecure Private Registries

This example deploys a VCH with the following configuration:

-Allows the VCH to pull images from the following insecure registries:

- All registries in the domain *.example.com
- A single instance of an insecure registry running at 192.168.100.207.
  - Provides the CA certificate for a vSphere Integrated Containers Registry instance.

Prerequisite

Follow the instructions in Obtain the vSphere Integrated Containers Registry Certificate to obtain the certificate file for your vSphere Integrated Containers Registry instance.

Create VCH Wizard

1. Leave the **Enable secure access to this VCH** switch in the green ON position.
2. Select the **Registry Access** tab.
3. Leave the **Whitelist registry mode** switch in the gray OFF position.
4. Under **Insecure registry access**, enter `*.example.com` to allow the VCH to access all registries that are running in that domain.
5. Click the **+** button, and enter `192.168.100.207` to allow the VCH to access the standalone registry at that address.
6. Under **Additional registry certificates**, click **Select** and navigate to the CA certificate file for your vSphere Integrated Containers Registry instance.

`vic-machine` Command

This example `vic-machine create` uses the `--registry-ca` and `--insecure-registry` options to allow access to secure and insecure registries.

```
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 *.example.com
--insecure-registry 192.168.100.207:5000
--registry-ca /home/admin/mycerts/ca.crt
--name vch1
--thumbprint certificate_thumbprint
--no-tlsverify
```

Result

The VCH can access the insecure registries in the domain specified, as well as the standalone insecure registry at 192.168.100.207 and the vSphere Integrated Containers Registry instance. Because whitelist mode is not enabled, it can also access public registries like Docker Hub.

What to Do Next

If you are using the Create Virtual Container Host wizard, click **Next** to configure the Operations User.

Configure the Operations User

A virtual container host (VCH) requires the appropriate permissions in vSphere to perform various tasks during VCH operation. Deployment of a VCH requires a user account with vSphere administrator privileges. However, day-to-day operation of a VCH requires fewer vSphere permissions than deployment.

During deployment of a VCH, vSphere Integrated Containers Engine runs all deployment operations by using the vSphere administrator account that you specify in the `vic-machine create --user` or `--target` options. If you are using the Create Virtual Container Host wizard, it uses the vSphere administrator account with which you are logged into the vSphere Client.

By default, if you deploy a VCH by using `vic-machine`, it runs with the same user account as you used to deploy it. In this case, the VCH uses the vSphere administrator account for post-deployment operations, meaning that it runs with full vSphere administrator privileges. Running with full vSphere administrator privileges is excessive, and potentially a security risk.

To avoid this situation, you should configure a VCH so that it uses different user accounts for deployment and for post-deployment operation by specifying an *operations user* when you deploy the VCH. By specifying an operations user with reduced vSphere privileges, you limit its post-deployment privileges to only those privileges that it needs for day-to-day operation.

If you use the Create Virtual Container Host wizard to deploy VCHs, it is **mandatory** to specify an operations user. If you use `vic-machine`, specifying an operations user is recommended but optional.

How the Operations User Works

If you specify an operations user, `vic-machine` and the VCH behave differently to how they would behave in a default deployment.

Default Behavior

- When you create a VCH by using `vic-machine`, you provide vSphere administrator credentials to `vic-machine create`, either in `--target` or in the `--user` and `--password` options. During deployment, `vic-machine create` uses these credentials to log in to vSphere and create the VCH. The VCH safely and securely stores the vSphere administrator credentials, for use in post-deployment operation.
- When you run other `vic-machine` commands on the VCH after deployment, for example, `vic-machine ls`, `upgrade`, or `configure`, you again provide the vSphere administrator credentials in the `--target` or `--user` and `--password` options. Again, `vic-machine` uses these credentials to log in to vSphere to retrieve the necessary information or to perform upgrade or configuration tasks on the VCH.
- When a container developer creates a container in the VCH, they authenticate with the VCH with their client certificate. In other words, the developer interacts with the VCH via the Docker client, and does not need to provide any vSphere credentials. However, the VCH uses the stored vSphere administrator credentials that you provided during deployment to log in to vSphere to create the container VM and to run operations on it.

Operations User Behavior

- When you create a VCH, the Create Virtual Container Host wizard uses the vSphere administrator credentials with which you logged into vSphere Client to create the VCH. If you use `vic-machine`, you provide vSphere administrator credentials either in `vic-machine create --target` or in the `--user` and `--password` options.
- You also provide the credentials for another vSphere account in the Operations User page of the Create Virtual Container Host wizard or in the `vic-machine create --ops-user` and `--ops-password` options.
- During deployment, vSphere Integrated Containers Engine uses the vSphere administrator credentials to log in to vSphere and create the VCH. The operations user credentials are safely and securely stored in the VCH, for later use in post-deployment operation. In this case, the VCH does not store the vSphere administrator credentials.
- When you perform operations on the VCH after deployment, for example, `vic-machine ls`, `upgrade`, or `configure`, you provide the vSphere administrator credentials in the `--target` or `--user` and `--password` options. This is the same as in the default case. The stored operations user credentials are not used for these operations.
- When a container developer creates a container in the VCH, the VCH uses the stored operations user credentials to log in to vSphere to create the container VM and to run operations on it.

Create a User Account for the Operations User

The user account that you specify as the operations user must exist before you deploy the VCH. The account and must have ready-only non-propagating permissions on the datacenter in which you are deploying the VCH. vSphere Integrated Containers Engine provides an option to automatically assign all of the required roles and permissions to the operations user account. If you prefer to assign roles and permissions manually, see Manually Create a User Account for the Operations User.

You can use the same user account as the operations user for multiple VCHs.

Prerequisite

Log into the Flex-based vSphere Web Client with a vSphere administrator account. You cannot use the HTML5 vSphere Client to create user accounts.

Procedure

1. In Home page of the vSphere Web Client, click **Roles**.
2. Click **Users and Groups** in the Navigator menu.
3. Select the appropriate domain and click the **+** button to add a new user.
4. Enter a user name for the operations user account, for example `vic-ops`.
5. Enter and confirm the password for this account, optionally provide the additional information, and click **OK**.
6. In the Hosts and Clusters view, right-click the datacenter in which to deploy the VCH and select **Add Permission**.
7. Under Users and Groups, select the operations user that you created.
8. Under Assigned Role, select **Read-only** from the drop-down menu.
9. Do not select the Propagate to children checkbox, and click **OK**.

Result

You can use the new user as the operations user account for VCHs. You must use the option to grant any necessary permissions to the user account when you deploy the VCH.

Options

The following sections each correspond to an entry in the Operations User page of the Create Virtual Container Host wizard. Each section also includes a description of the corresponding `vic-machine create` option.

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`.

vSphere User Credentials

AvSphere user account with which the VCH runs after deployment.

The user account that you specify as the operations user must exist before you deploy the VCH. For information about how to create an operations user account, see Create a User Account for the Operations User above.

Create VCH Wizard

1. In the **vSphere user credentials** text box, enter the user name for an existing vSphere user account.

   The user account must have ready-only non-propagating permissions on the datacenter in which you are deploying the VCH, or be a member of a user group on which you have manually configured the correct permissions.

2. Enter the password for the specified user account.

vic-machine Options

`--ops-user`, no short name

`--ops-password`, no short name

The user account that you specify in `--ops-user` must have ready-only non-propagating permissions on the datacenter in which you are deploying the VCH, or be a member of a user group on which you have manually configured the correct permissions.

If you do not specify `--ops-user`, the VCH runs with the vSphere Administrator credentials with which you deploy the VCH, that you specify in either `--target` or `--user`.

If you specify `--ops-user` but you do not specify `--ops-password`, `vic-machine create` prompts you to enter the password for the `--ops-user` account.

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

Grant Any Necessary Permissions

The operations user account must exist before you create a VCH. If you did not manually configure the operations user account with all of the necessary permissions, vSphere Integrated Containers Engine can do this for you.

Create VCH Wizard

- Select the **Grant this user any necessary permissions** check box.
- If you manually added the necessary permissions to the operations user account, do not select the check box.

vic-machine Option

`--ops-grant-perms`, no short name

If you specify `--ops-user`, you can also specify `--ops-grant-perms` so that `vic-machine` automatically grants the necessary vSphere permissions to the operations user account. If you specify `--ops-user` but do not specify `--ops-grant-perms`, you must configure the permissions on the operations user account manually.

The `--ops-grant-perms` option takes no arguments.

```
--ops-grant-perms
```

Example

This example uses the user account `vic-ops@vsphere.local` as the operations user, and automatically grants the necessary permissions to that account.

Prerequisite

Follow the instructions in Create a User Account for the Operations User to create a vSphere user account, `vic-ops@vsphere.local`.

Create VCH Wizard

1. In the **vSphere user credentials** text box, enter `vic-ops@vsphere.local`.
2. Enter the password for `vic-ops@vsphere.local`.
3. Select the **Grant this user any necessary permissions** check box.

vic-machine Command

This example `vic-machine create` command deploys a VCH with the following options:

- Specifies the account `Administrator@vsphere.local` in the `--target` option, to identify the user account with vSphere administrator privileges with which to deploy the VCH.
- Specifies the existing `vic-ops@vsphere.local` user account and its password in the `--ops-user` and `--ops-password` options, to identify the user account with which the VCH runs after deployment.
- Specifies `--ops-grant-perms` to automatically grant the necessary permissions to the `vic-ops@vsphere.local` user account.

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

What to Do Next

If you are using the Create Virtual Container Host wizard, click **Next** to review the configuration that you have made.

Manually Create a User Account for the Operations User

When you deploy a VCH, the user account that you specify as the operations user must have the correct privileges to allow the VCH to perform post-deployment operations. vSphere Integrated Containers Engine provides a mechanism to automatically assign the necessary permissions to the operations user account, but you can also choose to create the user account manually in vSphere. To do so, you create roles, assign privileges to those roles, and assign the roles to the user account to use as the operations 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.

Prerequisite

Log into the Flex-based vSphere Web Client with a vSphere administrator account. You cannot use the HTML5 vSphere Client to create user accounts.

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 Datastore > Low level file operations VirtualMachine.Configuration > Add new disk VirtualMachine.Configuration > Advanced VirtualMachine.Configuration > Remove disk VirtualMachine.Inventory > Create new VirtualMachine.Inventory > Remove
`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 > Assign virtual machine to resource pool 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

3. 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.

   i. Right-click an inventory object and select **Add Permission**.
   ii. Under Users and Groups, select the operations user group that you created.
   iii. Under Assigned Role, assign the appropriate role for each type of inventory object and select the **Propagate to children** check box where necessary.

The following table lists which roles to assign to which type of inventory object, and whether or not to propagate the role.

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

What to Do Next

You can use the user accounts in the user group that you created as operations users for VCHs. When you deploy VCHs you do not need to select the option to grant all necessary permissions in the Create Virtual Container Host wizard, or specify `--ops-grant-perms` in `vic-machine create` commands.

Finish VCH Deployment in the vSphere Client

When you have completed all of the pages in the Create Virtual Container Host wizard, you can review the details of the virtual container host (VCH) that you have configured. You can also copy the generated `vic-machine create` command.

Prerequisites

You completed all of the pages of the Create Virtual Container Host wizard in the vSphere Client.

Procedure

1. In the Summary page, review the details of the VCH that you have configured.

   Expand the entries as necessary to see more configuration details.

2. Scroll down to the bottom of the page to see the generated `vic-machine create` command that results from the configuration that you have specified in the wizard.
3. (Optional) Select a platform from the **Copy CLI command** drop-down menu, and click the clipboard icon to copy the command to the clipboard.

   Copying the `vic-machine create` command allows you to recreate similar VCHs at the command line or by using scripting. The platform that you select corresponds to the system on which you run `vic-machine` commands.

4. Click **Finish** to deploy the VCH.
5. In the **Virtual Container Hosts** tab, click the **>** icon next to the new VCH to follow its deployment progress.

   At the end of a successful deployment, the deployment log shows connection details for this VCH, and a success message.

6. (Optional) When the deployment has succeeded, click the link to the VCH Admin Portal for this VCH.

What to Do Next

The Create Virtual Container Host wizard does not include any escape characters in the generated `vic-machine` command. Consequently, if any of the values that you specified in the wizard include special characters or spaces, you must edit the saved `vic-machine` command to wrap those values in quotes before you can reuse the command to create similar VCHs. For information about using quotes to escape special characters and spaces, see Specifying Option Arguments.

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 minimum configuration:

- The VCH must be able to pull the `dch-photon` image from the vSphere Integrated Containers Registry instance. You must provide the registry's CAcertificate to the VCH so that it can connect to the registry.
- A `dch-photon` container creates an anonymous volume, and as such requires a volume store named `default`.

Example

This example shows how to use both the Create Virtual Container Host wizard and `vic-machine` to create a VCH with the minimum configuration required to deploy a `dch-photon` container.

Prerequisites

1. Log in to the vSphere Integrated Containers Management Portal with a vSphere administrator, Cloud Admin, or DevOps admin user account.
2. Go to **Administration** > **Configuration**, and click the link to download the **Registry Root Cert**.

Create VCH Wizard

1. Log in the HTML5 vSphere Client and go to the **vSphere Integrated Containers** view.
2. Click **vSphere Integrated Containers** in the main panel, select the **Virtual Container Hosts** tab, and click **+ New Virtual Container Host**.
3. On the General Settings page, enter a name for the VCH, for example, `vch_dch_photon`, and click **Next**.
4. On the Compute Capacity page, expand the **Compute resource** inventory hierarchy and select a standalone host, cluster, or resource pool to which to deploy the VCH, and click **Next**.
5. On the Storage Capacity page, select a datastore to use as the Image Datastore.
6. Remain on the Storage Capacity page and configure the volume datastore.
    i. Set the **Enable anonymous volumes** switch to the green ON position.
    ii. Select a datastore to use as a volume datastore.
    iii. Optionally provide the path to a folder in that datastore.
    iv. Click **Next**.
7. On the Configure Networks page, select existing port groups for use as the bridge and public networks, and click **Next**.
8. On the Security page, for simplicity, leave the default options for automatic server certificate generation, and set the **Client Certificates** switch to the gray OFF position to disable client certificate verification.
9. Remain on the Security page, click Registry Access, and under Additional registry certificates, click **Select** to upload the certificate for vSphere Integrated Containers Registry, then click **Next**.
10. On the Operations User page, enter the user name and password for an existing vSphere account, select the **Grant this user any necessary permissions** check box, and click **Next**.
11. On the Summary page, click **Finish**.

vic-machine Command

For simplicity, this example `vic-machine create` command deploys a VCH with the `--no-tlsverify` 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.

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`
- Delete Virtual Container Hosts
- Virtual Container Host Administration Portal

Interoperability of Virtual Container Hosts 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 VCHs with other vSphere features and VMware products.

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 resource pool, the VCH endpoint VM, or container VMs. Always use the vSphere Integrated Containers plug-in for the vSphere Client or `vic-machine delete` to delete VCHs. Always use Docker commands or the vSphere Integrated Containers Management Portal to perform operations on container VMs.
- Manually restarting container VMs can result in incorrect end-times for container operations. Do not manually restart the VCH or container VMs. Always use Docker commands or the vSphere Integrated Containers Management Portal 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. For more information about VCHs and High Availability, see Backing Up Virtual Container Host Data.

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 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 perform operations on VCHs or container VMs directly in the vCenter Server inventory. Specifically, powering off, powering on, or deleting the VCH can cause vSphere Integrated Containers Engine to not function correctly. Always use the vSphere Integrated Containers plug-in for the vSphere Client or `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 or vSphere Integrated Containers Management Portal 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 click the **vSphere Client** logo in the top left corner.
2. Under Inventories, click **vSphere Integrated Containers**.

   The vSphere Integrated Containers view presents the number of VCHs and container VMs that you have deployed to this vCenter Server instance.

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 perform operations on VCHs or container VMs directly in the vCenter Server inventory. Specifically, powering off, powering on, or deleting the VCH can cause vSphere Integrated Containers Engine to not function correctly. Always use the vSphere Integrated Containers plug-in for the vSphere Client or `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 or vSphere Integrated Containers Management Portal 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 resource pool.
4. Expand the VCH resource pool 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

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.

Wrap any option arguments that include spaces or special characters in quotes. For information about option arguments that might require quotes, see Specifying Option Arguments.

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

`--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 in `vic-machine` commands, 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
```

`--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**. You specify this option in the same way as you specify the `vic-machine create --target` option.

`--user`

Short name: `-u`

The ESXi host or vCenter Server user account with which to run the `vic-machine` command. The user account that you specify in `--user` must have vSphere administrator privileges. You specify this option in the same way as you specify `vic-machine create --user`.

`--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. You specify this option in the same way as you specify `vic-machine create --password`.

`--thumbprint`

Short name: None

The thumbprint of the vCenter Server or ESXi host certificate. You specify this option in the same way as you specify `vic-machine create --thumbprint`.

`--compute-resource`

Short name: `-r`

The relative path to the host, cluster, or resource pool in which you deployed the VCH. You specify this option in the same way as you specify `vic-machine create --compute-resource`.

NOTE: If you specify the `id` option in `vic-machine` commands, you do not need to specify the `compute-resource` option.

`--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. You specify this option in the same way as you specify `vic-machine create --name`.

`--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 user name 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 vSphere Certificate Thumbprints.

  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 user name 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 vSphere Certificate Thumbprints.

  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 Running 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. For information about how to obtain the vCenter Server certificate thumbprint, see Obtain vSphere Certificate Thumbprints.

```
$ vic-machine-operating_system configure
    --target vcenter_server_username:password@vcenter_server_address
    --id vch_id
    --thumbprint new_certificate_thumbprint
```

CAUTION: Specifying the `--force` option bypasses safety checks, including certificate thumbprint verification. Using `--force` in this way can expose VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials. Using `--force` can result in unexpected deployment topologies that would otherwise fail with an error. Do not use `--force` in production environments.

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 CAcertificates 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 Connect Virtual Container Hosts to Registries.

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 CAcertificate 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 Virtual Container Hosts Security.

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 Specify Volume Stores 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 Configure the Public Network.

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 the topic on configuring general VCH settings. 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 resource pool 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` memory and CPU reservation and shares options, see Virtual Container Host Compute Capacity.

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.

Debug Running 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 Debug in the topic on configuring general VCH settings.

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
```

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 user name 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 vSphere Certificate Thumbprints.

  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.

```
$ vic-machine-operating_system delete
--target vcenter_server_username:password@vcenter_server_address
--name vch_name
--force
```

CAUTION: Using `--force` in this way exposes VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials. Using `--force` also bypasses other checks, and can result in data loss.

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 user name 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 user name 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 only upgrade vSphere Integrated Containers from version 1.2.x to 1.3.x, or from 1.3.x to a later 1.3.y update release. You cannot upgrade any release earlier than 1.2.x to 1.3.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 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 vSphere Client Plug-Ins on vCenter Server for Windows
- Upgrade the vSphere Client Plug-Ins on vCenter Server Appliance

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 1.2 and 1.3 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.
- 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 have a 1.2.x version of the vSphere Integrated Containers appliance, you can upgrade your existing installation to 1.3.x.

During the upgrade, all configurations transfer to the upgraded appliance.

Prerequisites

- You have completed the pre-upgrade tasks listed in Tasks to Perform Before Upgrading the vSphere Integrated Containers Appliance.
- Deploy the latest 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** check box, 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 script detects your upgrade path correctly, enter `y` to proceed with the upgrade.
        - If the upgrade script detects the upgrade path incorrectly, enter `n` to abort the upgrade and contact VMware support.

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 1.3.x registry instance to the 1.2.x replication endpoint still functions, but it is recommended that you upgrade the target registry.
- 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-ins for the vSphere Client. For information about upgrading the vSphere Client plug-ins, see

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

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 vSphere Certificate Thumbprints.

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 user name 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.

   CAUTION: Specifying the `--force` option bypasses safety checks, including certificate thumbprint verification. Using `--force` in this way can expose VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials. Using `--force` can result in unexpected deployment topologies that would otherwise fail with an error. Do not use `--force` in production environments.

   ```
   $ 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.
- 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.

CAUTION: Specifying the `--force` option bypasses safety checks, including certificate thumbprint verification. Using `--force` in this way can expose VCHs to the risk of man-in-the-middle attacks, in which attackers can learn vSphere credentials. Using `--force` can result in unexpected deployment topologies that would otherwise fail with an error. Do not use `--force` in production environments.

```
--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 vSphere Client Plug-Ins on vCenter Server for Windows

If you have previous installations of the vSphere Client plug-ins for vSphere Integrated Containers, you must upgrade them. This procedure describes how to upgrade existing plug-ins for a vCenter Server running on Windows.

Prerequisites

- You are upgrading the plug-ins 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 1.2.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 1.3.x appliance to a newer 1.3.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 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 vSphere Certificate Thumbprints.

Procedure

1. Run the upgrade script and follow the prompts.

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

   i. Enter the FQDN or 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 upgrade finishes, stop and restart the vSphere Client services.

   NOTE: The Flex-based plug-in has no new features in this release. However, the upgrade script updates the metadata for the Flex-based client. Consequently, you must restart both of the HTML5 and Flex-based clients.

   - HTML5 vSphere Client:

     ```
     service-control --stop vsphere-ui
     ```

     ```
     service-control --start vsphere-ui
     ```

   - Flex-based vSphere Web Client:

     ```
     service-control --stop vsphere-client
     ```

     ```
     service-control --start vsphere-client
     ```

What to Do Next

Log in to the HTML5 vSphere Client, go to the vSphere Integrated Containers view, and verify that the version number reflects the upgrade.

Upgrade the vSphere Client Plug-Ins on vCenter Server Appliance

If you have previous installations of the vSphere Client plug-ins for vSphere Integrated Containers, you must upgrade them. This procedure describes how to upgrade 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 1.2.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 1.3.x appliance to a newer 1.3.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 vSphere Certificate Thumbprints.
- 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. Set the following environment variables:

   - vSphere Integrated Containers appliance address:

     ```
     export VIC_ADDRESS=vic_appliance_address
     ```

   - vSphere Integrated Containers Engine bundle file:

     ```
     export VIC_BUNDLE=vic_v1.3.0.tar.gz
     ```

     If you are upgrading to a different version of the appliance, update `1.3.0` to the appropriate version in the command above. You can see the correct version by going to https://*vic_appliance_address*:9443/files/ in a browser.

4. 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.

   Copy and paste the following command as shown:

   ```
   curl -kL https://${VIC_ADDRESS}:9443/files/${VIC_BUNDLE} -o ${VIC_BUNDLE}
   ```

5. Unpack the vSphere Integrated Containers binaries.

```
tar -zxf ${VIC_BUNDLE}
```

6. Navigate to `/vic/ui/VCSA`, run the upgrade script, and follow the prompts.

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

```
./upgrade.sh
```

   i. Enter the FQDN or 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.
7. When the upgrade finishes, stop and restart the vSphere Client services.

   NOTE: The Flex-based plug-in has no new features in this release. However, the upgrade script updates the metadata for the Flex-based client. Consequently, you must restart both of the HTML5 and Flex-based clients.

   - HTML5 vSphere Client:

     ```
     service-control --stop vsphere-ui
     ```

     ```
     service-control --start vsphere-ui
     ```

   - Flex-based vSphere Web Client:

     ```
     service-control --stop vsphere-client
     ```

     ```
     service-control --start vsphere-client
     ```

What to Do Next

Log in to the HTML5 vSphere Client, 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

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 services that run in the appliance by logging in to the vSphere Integrated Containers appliance. The following services run in the vSphere Integrated Containers appliance:

- vSphere Integrated Containers Registry service
- vSphere Integrated Containers Management Portal service
- The file server for vSphere Integrated Containers Engine downloads and installation of the vSphere Client plug-ins
- The `vic-machine` server service, that powers the Create Virtual Container Host wizard in the HTML5 vSphere Client plug-in

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`
    - `vic-machine` server: `systemctl status vic_machine_server.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 services that run in the appliance by logging in to the vSphere Integrated Containers appliance. The following services run in the vSphere Integrated Containers appliance:

- vSphere Integrated Containers Registry service
- vSphere Integrated Containers Management Portal service
- The file server for vSphere Integrated Containers Engine downloads and installation of the vSphere Client plug-ins
- The `vic-machine` server service, that powers the Create Virtual Container Host wizard in the HTML5 vSphere Client plug-in

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`
    - `vic-machine` server: `systemctl restart vic_machine_server.service`

Back Up and Restore vSphere Integrated Containers

A vSphere Integrated Containers installation is inherently stateful, even if the containers running on it are not. As such, the question of how to back up and restore vSphere Integrated Containers is an important one.

The main components of vSphere Integrated Containers store the following persistent data:

- vSphere Integrated Containers Registry stores immutable image data. You can back up vSphere Integrated Containers Registry state by using VM snapshots and clones.
- vSphere Integrated Containers Management Portal stores user and project metadata. You can back up vSphere Integrated Containers Management Portal state by using VM snapshots and clones.
- Container volumes store persistent data that container VMs use and share. You can back up container volumes by using snapshots and clones only if the container is not running. You restore container volumes by copying virtual disks to a known location.

You can consider all other data in vSphere Integrated Containers to be ephemeral state, that is not suitable for backup.

The following topics describe the types of state that a typical vSphere Integrated Containers installation stores, where it resides, the nature of the data, why you might want to back it up and some alternatives to a backup strategy. The topics describe different approaches to backup and their relative merits.

- Backup and Restore the vSphere Integrated Containers Appliance
- Backing Up Virtual Container Host Data
- Backing Up and Restoring Container Volumes

Back Up and Restore the vSphere Integrated Containers Appliance

The vSphere Integrated Containers appliance runs various services, such as vSphere Integrated Containers Management Portal and vSphere Integrated Containers Registry. In this version of vSphere Integrated Containers, the appliance has two virtual disks attached to it:

- Asystem disk, that contains the operating system and application state of the vSphere Integrated Containers appliance.
- Adata disk, that contains all important persistent data.

The separation of different types of data between disks allows you to upgrade the appliance with an existing data disk from a previous installation. It also allows you to back up and restore the data disk, if necessary.

Snapshots and Clones

You can take a conventional approach to backing up the appliance, in the same way as for any other stateful VM. The appliance disks are not independent of the appliance VM, so if you take a snapshot of the appliance VM, it also takes snapshots of the data and system disks.

NOTE: If you do not take a snapshot the of the memory of the appliance, it comes back up in a powered-off state. This is probably the preferred approach, but it means that the registry is temporarily unavailable while the appliance boots up.

Once you have created a snapshot of the appliance VM, you can clone the snapshot of the data disk, even while the appliance is running. You can use tools like `vmkfstools` to copy the data disk to a backup datastore.

Restoring the Data Disk

You have two choices to restore the data disk:

- Revert the appliance to a VM snapshot.
- Copy a cloned VMDK into the appliance datastore and attach it to the `SCSI(0:1)` virtual device node on the appliance VM.

If you are not restoring the data disk from a live snapshot, you must shut down the appliance before the you restore the disk.

Backing Up Virtual Container Host Data

A virtual container host (VCH) is a pool of virtual resources in which container images are deployed as lightweight VMs. A VCH consists of the VCH endpoint VM, and any number of container VMs. The endpoint VM runs the control plane and container VMs are created, powered on, powered off and deleted in response to container client calls to create, start, stop and delete containers.

This topic provides information about different types of VCH data, what data you should back up, what data does not require backup, and why.

VCH Components and State

To understand the requirements when backing up VCH data, you must first understand state for the different components of a VCH.

Containers

Containers encourage application designers to think about where state belongs and the nature of that state. By design, any part of the container file system that is not a mounted volume is ephemeral by nature and is lost when the container is deleted.

Container Volumes

Assume that a stateful container has one or more volumes attached and a stateless container does not.

- A stateless container that reads and writes to a remote database can be easily scaled up or down and can be run in multiple failure domains behind a load-balancer.
- A stateful container is tied to running in a location from which it can access its volume store. A container volume is intentionally persistent by nature and can exist beyond the lifespan of a container or even of a VCH.

You must also consider whether a volume is sharable between containers. If a volume is an NFS share, it is possible for multiple containers in multiple failure domains to share the same persistent data, provided that they have the correct locking semantics. If a volume is not sharable, the ability for compute resources to move between ESXi hosts is constrained by the use of a shared datastore or by physically copying the data. vSphere makes it possible to live-migrate workloads between hosts in a cluster while keeping persistent data on a shared datastore. For this reason, vSphere Integrated Containers is well-suited to running stateful workloads.

Anonymous Volumes

An anonymous volume is created in the `default` volume store every time a container is run from an image that uses a Dockerfile with a `VOLUME` command. Anonymous volumes are not desirable for production, because you cannot specify the volume size, name, or class of storage. This has implications for your backup strategy, because it might not be clear what anonymous volumes are being used for.

Container Images

Container images are immutable and should persist in a container registry. This is one of the reasons why vSphere Integrated Containers includes vSphere Integrated Containers Registry. When a developer pulls an image, either explicitly or as part of a container execution, the VCH caches the image locally. You can consider the VCH image cache to be ephemeral, even though the containers depend on the images being present in the cache when they run.

Configuration State

When you deploy a VCH, you provide a significant amount of configuration to `vic-machine`, including networks, datastores, credentials, and so on. All of this configuration data is stored in the VMX file for the VCH endpoint VM. Running containers also have a configuration state, which is stored in the VMX file of each container VM.

Containers and the VCH endpoint VM are designed to be stateless with respect to guest configuration. Nothing is persisted in the guest OS. When a container VM starts up, it discovers its state from its VMX file. When a VCH endpoint VM starts up, it discovers both its own state and also the state of the image cache and the existing containers. The stateless nature of the VCH endpoint VM simplifies upgrades. Upgrade is just a case of powering down the VCH endpoint VM, swapping out the ISO from which it boots, and powering it back up again.

High Availability

The VCH is designed so that containers can run independently of the availability of the VCH endpoint VM. However, it is important to note that this is dependent on how the containers are deployed.

The most important consideration is how the networking for the container is configured. If containers use port mapping, the containers are accessible over a network via a port on the VCH endpoint VM. If the endpoint VM goes down for any reason, that network connection is no longer available. This is why vSphere Integrated Containers offers the ability to connect containers directly to a vSphere network by using the `--container-network` option. If you use container networks, containers have their own identity on the container network. Consequently, the network and the container have no dependency on the VCH endpoint VM for execution.

You can configure vSphere High Availability to restore the VCH endpoint VM in the case of an ESXi host failure. If a host goes down and a VCH endpoint VM is lost, the only impact should be a temporary loss of the control plane, namely the ability to manage the lifecycle of containers, networks, and volumes.

Data Integrity

Data integrity for persistent data is extremely important, but a backup strategy is not the only way to ensure data integrity.

Data Replication

vSphere Integrated Containers supports different classes of storage, of which VMware vSAN is an example. vSAN offers built-in redundancy by replicating data to multiple physical drives and can tolerate hardware failure or nodes becoming unavailable. This is particularly useful for persistent volumes.

You can also replicate container image data by installing the vSphere Integrated Containers appliance on vSAN storage. vSphere Integrated Containers Registry also offers the ability to replicate image data to other registry instances.

Data Encryption

Encryption of persistent data is typically provided by the class of data storage that you use. For example, vSAN provides built-in encryption capabilities.

vSphere Integrated Containers makes it easy to specify different classes of datastore for different classes of data, by using the `vic-machine create --image-store` and `--volume-store` options. You specify a single image store in which to store immutable image and ephemeral container state for a VCH. You can then specify any number of volume stores which map to different vSphere datastores, which container developers can specify by their label when they provision a container.

Making Backups

Replication and Encryption are good solutions for protecting data integrity within a given isolation domain. However, if an entire isolation domain is lost or if data becomes corrupted, keeping backups is essential.

The fact that vSphere Integrated Containers volumes are first-class citizens on vSphere datastores means that you can back up container volumes by using any solution that knows how to backup virtual disks.

Conclusions

The best approach to backing up VCH data is to make a clear distinction between persistent state and ephemeral state, and to build well-defined strategies for application deployment, configuration, and backup.

Data that Does Not Require Back Up

Configuration state of container VMs and of the VCH endpoint VM is not a good candidate for backup. This is because configuration state is highly dependent on a tight coupling between itself and the current state of the vSphere environment. As such, there is a strong possibility that a container VM or VCH endpoint VM that you bring back from a backup will not have a configuration state that is consistent with the environment to which you are restoring it. This is not the case with volumes or container images, which can exist independently of any vSphere environment and which can be copied or moved without problems.

Images cached in the VCH Image Store are not good candidates for backup because they are copies of immutable state already stored and backed up in a container registry.

It is important to remember that if the VCH endpoint VM becomes unavailable, this only impacts the ability to manage the lifecycle of the containers running in that VCH. If you have used container networks, the containers themselves continue to run unimpeded. If you cannot resolve the issues affecting the VCH endpoint VM or vSphere environment, you should be able to create a new VCH in a different vSphere environment, deploy new instances of the same container workloads, and switch to those new instances. This depends on having the appropriate load-balancing and data migration in place however, which is typically provided by a higher-level scheduler.

Data that Requires Back Up

In this release of vSphere Integrated Containers, the only VCH data that you should consider for backup is the following:

- Persistent state that is stored in container volumes.

For information about how to backup and restore container volumes, see Backing Up and Restoring Container Volumes.

Backing Up and Restoring Container Volumes

This topic describes the types of volumes that vSphere Integrated Container supports, and provides an example of how containers persist data in container volumes.

vSphere Integrated Containers supports two types of volumes, each of which has different characteristics.

- VMFS virtual disks (VMDKs), mounted as formatted disks directly on container VMs. These volumes are supported on multiple vSphere datastore types, including NFS, iSCSI and VMware vSAN. They are thin, lazy zeroed disks.
- NFS shared volumes. These volumes are distinct from a block-level VMDK on an NFS datastore. They are Linux guest-level mounts of an NFS file-system share.

VMDK Volumes

A VMDK volume comprises two elements:

- A `.vmdk` file which is a formatted virtual disk mounted at a configured location in a container guest file system
- Some metadata that describes the volume.

These volumes are stored directly on a vSphere datastore in a location that you specify when you deploy a virtual container host (VCH).

VMDK volume disks are locked for exclusive use while a container VM runs, so other running containers cannot share them. It is possible to configure multiple containers with the same volume disk, but only one container can run at a time.

Another limitation of VMDK disks is that you cannot clone them while they are in use. You can take snapshots and then clone the snapshots, but vSphere Integrated Containers does not currently have built-in support for doing this. Consequently, you can only clone vSphere Integrated Containers volume disks while a container is not running.

For information about backing up and restoring VMDK volumes, see the following topics.

- Backing Up VMDK Volumes
- Restoring VMDK Volumes

NFS Shared Volumes

NFS volume support is designed for use-cases where multiple containers need read-write access to the same volume.

Taking snapshots and making clones of NFS volumes is handled by the system that provides the NFS server, which should have its own backup strategy. If the NFS server is running as a VM, you can use the same backup strategy as for other stateful VMs.

Example: Persistent Container State

This example uses a VCH with two VMDK volume stores and one NFS volume store to demonstrate how data persists in container volumes.

Procedure

1. Run `vic-machine create` with the following options to deploy a VCH with three volume stores,

   ```
   --volume-store vsanDatastore/volumes/my-vch-data:replicated-encrypted
   --volume-store iSCSI-nvme/volumes/my-vch-logs:default
   --volume-store nfs://10.118.68.164/mnt/nfs-vol?uid=0&gid=0&proto=tcp:shared
   ```

   - The first volume store is on a vSAN datastore and uses the label `replicated-encrypted`. Container developers can create a

volume in that volume store by running the following command:

```
docker volume create --opt VolumeStore=replicated-encrypted myData
```

- The second volume store uses cheaper storage backed by a FreeNAS server mounted using iSCSI. It is used for storing log data. It has the label `default`, which means that any volume that is created without a specifying a volume store is created here.
- The third volume store is an NFS export called `/mnt/nfs-vol` on an NFS server.

2. Browse the three datastores to see the folders that deploying this VCH created.
   - `vsanDatastore/volumes/my-vch-data/volumes`
   - `iSCSI-nvme/volumes/my-vch-logs/volumes`
   - `nfs://10.118.68.164/mnt/nfs-vol/volumes`
3. Run the following commands in the Docker client to create three volumes.

```
$ docker volume create --opt VolumeStore=replicated-encrypted --opt Capacity=10G mydata
```

```
$ docker volume create --opt Capacity=5G mylogs
```

```
$ docker volume create --opt VolumeStore=shared myshared
```

   Note that the second example does not specify a volume store, which implies the use of the `default` volume store.

4. Browse the three datastores to see the files that these commands created.
   - `vsanDatastore/volumes/my-vch-data/volumes/mydata/mydata.vmdk`
   - `vsanDatastore/volumes/my-vch-data/volumes/mydata/ImageMetadata/DockerMetaData`
   - `iSCSI-nvme/volumes/my-vch-logs/volumes/mylogs/mylogs.vmdk`
   - `iSCSI-nvme/volumes/my-vch-logs/volumes/mylogs/ImageMetadata/DockerMetaData`
   - `nfs://10.118.68.164/mnt/nfs-vol/volumes/myshared`
   - `nfs://10.118.68.164/mnt/nfs-vol/volumes_metadata/myshared/DockerMetaData`

     As a vSphere administrator, you would not normally need to know the conventions and contents of these folders, but it is important for the purposes of demonstrating backup and restore.

5. Examine the `DockerMetaData` file of the `mydata` volume.

   You see JSON data that adds some context to this particular disk. This is the same data that would be returned by running `docker volume inspect mydata` in the Docker client:

```
{
"Driver":"local",
"DriverOpts":{
   "Capacity":"10G",
   "VolumeStore":"replicated-encrypted"
},
"Name":"mydata",
"Labels":{
},
"AttachHistory":[
   ""
],
"Image":""
```

```
}
```

6. In the Docker client, mount the volumes to a container and add some data to them.

```
$ docker run -it --name test -v mydata:/data -v mylogs:/logs -v shared:/shared ubuntu
$ echo “some data” > /data/some-data
$ echo “some logs” > /logs/some-logs
$ echo “some shared” > /shared/some-shared
$ exit
```

   This operation creates a new container VM with volumes mounted at the specified locations. The `echo` command sends some data to the `mydata` volume, some logs to the `mylogs` volume, and some shared data to the `shared` volume. The fact that container has exited means that the container VM is now powered off, but the volume disks are still part of its configuration. If you restart the container, the volumes are mounted again.

   If you start a new container with the same volumes configured on it, the new container will be able to see the existing data and modify it. Remember that the `logs` and `data` volumes are exclusive to a single running container, whereas the `shared` volume is not.

7. Run `docker rm test` to delete the container.

8. Run `docker volume ls` to list the available volumes.

   The volumes and the data that they contain are still available after you have deleted the containers.

```
DRIVER              VOLUME NAME
vsphere             mydata
vsphere             mylogs
vsphere             myshared
```

9. Browse the three datastores to see that the files that the container created are still present.

Backing Up VMDK Volumes

Backing up a VMDK volume involves copying the virtual disk to a new location. However, there are some important caveats to this. Most commercial backup solutions focus on making clones or snapshots of a VM. They might not allow you to back up virtual disks on their own. Backup solutions have file-based backup and restore, but whether or not this is the correct approach depends on the characteristics of the datastore.

For example, `tar` and `untar` will work with virtual disks on a FreeNAS, ZFS, or iSCSI setup, but will not work with VMware vSAN. This is because in vSAN, virtual disk data is a hidden object, so only the metadata can be seen. Tools such as `vmkfstools` provide one way to ensure that a virtual disk is cloned properly.

Snapshots vs Clones

Taking a snapshot of a VM or disk allows its state to be frozen in time on the same datastore and potentially restored at a future date. Snapshots can protect against data corruption, but they do not protect against datastore hardware failure or accidental deletion, unless you clone and move the snapshot.

Cloning a VM or a disk performs a deep copy that you can move to a different datastore and bring back in its entirety. This can protect against corruption, hardware failure, and deletion.

Container VMs are not designed for snapshots to be taken, and volume disks are mounted as independent persistent disks. This is by design, so that disks are not deleted when a container is deleted, but the disks are also not subject to snapshots. As such, currently the only way to back up volume disks is to clone them. For more information about cloning volumes, see Datastore Approach below.

Thin vs Fat Disks

vSphere Integrated Containers creates VMDK volumes as thin, lazy zeroed disks. This means that they only take up as much space as they need on the datastore, up to a set capacity limit.

Note that some methods of cloning preserve thin provisioning and some do not. Most file copy approaches to backup will make a byte-for-byte copy, which results in the creation of a fat clone of a thin disk. For more information about the file copy approach to backup, see File Copy Approach below.

The `vmkfstools` utility maintains the thinness of a volume disk when it makes a clone. For more information about `vmkfstools`, see Using `vmkfstools` below.

You should experiment with the clone and backup solutions that you use to be aware of the consequences of making thin disks fat, particularly when it comes to the impact of restoring disks from a backup.

Approaches to Backing Up VMDK Volumes

The following examples show how you can use different approaches to back up the iSCSI and vSAN volume stores used in Example: Persistent Container State in the previous topic.

Datastore Approach

Your backup solution might have the ability to create snapshots or clones of entire datastores. This is a good way to ensure that you have backed up everything, but it also makes it more important to use different datastores for different types of state. For example, it would potentially be a waste of bandwidth to back up ephemeral state or cached immutable image state.

File Copy Approach

A file copy approach is the simplest way to back up volumes, because you can back up an entire volume store by copying the root folder from the datastore. However, as mentioned above, this does not work for all datastore types, and most probably results in fat versions of volume disks.

For example, you can use Veeam to create a File Copy Job that copies the `/vmfs/volumes/iSCSI-nvme/volumes/my-vch-logs` volume store from the example. This will copy the disks, the folders, and the metadata.

By backing up the root folder, you keep the path structure intact. If you restore the volume store to the current VCH or add it to a new VCH, the configuration of the `vic-machine --volume-store` argument matches that of the path you backed up.

Using `vmkfstools`

This example backs up the `vsanDatastore/volumes/my-vch-data` vSAN volume store from the example by cloning the virtual disk to another datastore by using `vmkfstools` and then copying over the metadata.

1. In an ESXi host shell, navigate to the `volumes` folder.

   ```
   cd /vmfs/volumes
   ```

   NOTE: The ESXi host must be able to access both the vSAN and iSCSI datastores.
2. Create a folder in the iSCSI datastore in which to copy the backup.

   ```
   mkdir -p iSCSI-nvme/volumes/my-vch-data-backup/volumes/mydata
   ```

3. Use `vmkfstools` to make a clone of the vSAN volume store disk in the folder that you just created in the iSCSI datastore.

   ```
   vmkfstools -i vsanDatastore/volumes/my-vch-data/volumes/mydata/mydata.vmdk iSCSI-
   nvme/volumes/my-vch-data-backup/volumes/mydata/mydata.vmdk
   ```

4. Copy the volume store metadata into the backup folder.

   ```
   cp -R vsanDatastore/volumes/my-vch-data/volumes/mydata/imageMetadata/ iSCSI-nvme/volumes/my-
   vch-data-backup/volumes/mydata
   ```

This is a rather manual approach to backup, but you could write a script that lists all of the folders in `vsanDatastore/volumes/my-vch-data/volumes` and uses that as an input to another script based on the above commands.

Restoring VMDK Volumes

How to restore volumes is conceptually straightforward. It should be a case of copying the VMDK disks and metadata back to either their original location or to a new location, or restoring the state of an NFS server. However, it is important to understand how vSphere Integrated Containers interacts with the volumes to ensure that this succeeds without error.

Restoring Volumes into an Existing VCH

Restoring a volume into an existing VCH is a question of copying the VMDK disks and metadata back to the expected location on the datastores.

An important caveat is that you cannot overwrite a volume disk that is attached to a currently running container. If you must bring back a volume that has the same name as one that is in use, there are two solutions:

- Rename the cloned volume
- Rename the cloned volume store.

Renaming a Cloned Volume

Renaming a cloned volume requires 3 steps:

1. Rename the volume folder to `newname`.
2. Rename the `.vmdk` file to `newname.vmdk`
3. Set the value of the `name` tag in the `DockerMetadata` file JSON to `newname`.
4. Copy the volume into the volume store.
5. For vSphere Integrated Containers engine to see the new volume, reboot the VCH endpoint VM.

   You need to reboot because the act of copying the volume into the volume store does not notify vSphere Integrated Containers of its existence. Rebooting the VCH endpoint VM us not unusual or drastic. Any reconfiguration of the VCH causes a reboot of the VCH endpoint VM.

Renaming a Cloned Volume Store

You can bring back an entire cloned volume store under a different name.

1. Rename the root folder of the cloned volume store.
2. Copy it onto a datastore that is visible to the VCH.
3. Run `vic-machine configure --volume-store` to add the cloned volume store to the VCH.

Restoring Volumes into a new VCH

Restoring a volume store and then creating a new VCH that can use the volumes is a matter of ensuring that the `vic-machine create --volume-store` argument is correctly configured to point to the volume store. vSphere Integrated Containers is designed to deploy new VCHs onto existing volume stores. You can check whether the volume store has been picked up by running `docker info` and `docker volume ls`.

Troubleshooting vSphere Integrated Containers

This section provides solutions for common problems that you might encounter during operation.

Access and Configure Appliance Logs

You access the logs for the vSphere Integrated Containers appliance by using SSH and navigating to `/storage/log`. You can also configure log retention for the different logs.

Prerequisites

Make sure that SSH access to the appliance is enabled. To enable SSH access to the appliance, see Reconfigure the vSphere Integrated Containers Appliance.

Procedure

1. Use SSH to connect to the appliance as root user.

   ```
   $ ssh root@vic_appliance_address
   ```

   When prompted for the password, enter the appliance password that you specified when you deployed the appliance.
2. To access logs for vSphere Integrated Containers Registry, navigate to `/storage/log/harbor`.

   ```
   $ cd /storage/log/harbor
   ```

   The `/storage/log/harbor` folder contains the log files for the following services:

   - `adminserver.log`: Registry administration service
   - `clair-db.log`: Clair database used for vulnerability scanning of images
   - `clair.log`: Clair service used for vulnerability scanning of images
   - `jobservice.log`: Registry job service log
   - `mysql.log`: Embedded registry database
   - `notary-db.log`: Notary database by Docker Content Trust
   - `notary-server.log`: Notary server used by Docker Content Trust
   - `notary-signer.log`: Notary image signing service used by Docker Content Trust
   - `proxy.log`: Proxy service logs
   - `registry.log`: Registry service logs
   - `ui.log`: User interface logs
3. To configure the log retention for the registry services, edit the `/storage/data/harbor/harbor.cfg` file.

   ```
   $ vi /storage/data/harbor/harbor.cfg
   ```

   The default configuration allows 50 files, up to 200MB each per service.

   i. To set the maximum number of files used for storing logs per service, change the `log_rotate_count` property value to the desired number.
   ii. To set the maximum size in MB per file, change the `log_rotate_size` property value to the desired number.
4. To access logs for the `vic-machine-server` service, navigate to `/storage/log/vic-machine-server`.

   ```
   $ cd /storage/log/vic-machine-server
   ```

   The default configuration allows 10 files, up to 1GB each.
5. To configure the `vic-machine-server` service log retention, edit the `/etc/logrotate.d/vic-machine-server` file.

   ```
   $ vi /etc/logrotate.d/vic-machine-server
   ```

   The default configuration allows 50 files, up to 200MB each per service.

i. To set the maximum number of files used for storing logs, change the `rotate` property value to the desired number.
ii. To set the maximum size in GB per file, change the `size` property value to the desired number.

6. To access logs for vSphere Integrated Containers Management Portal, navigate to `/storage/log/admiral`.

```
$ cd /storage/log/admiral
```

7. To configure the management portal log retention, edit the `/etc/vmware/admiral/logging-vic.properties` file.

```
$ vi /etc/vmware/admiral/logging-vic.properties
```

The default configuration allows 5 files, up to 1GB each.

i. To set the maximum number of files used for storing logs, change the `java.util.logging.FileHandler.count` property value to the desired number.
ii. To set the maximum size in bytes per file, change the `java.util.logging.FileHandler.limit` property value to the desired number.

Access Virtual Container Host Log Bundles

Virtual container hosts (VCHs) provide log bundles that you can download from the VCH Admin portal.

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.

Troubleshooting VCH Creation Errors

During a creation of a VCH, a log file named `vic-machine_<timestamp>_create_<id>.log` populates. You can find that file on the target datastore in a folder with the same name as the VCH that you specified for the `vic-machine create` command.

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

Appliance Console Shows OVF Error on Boot

When you boot the vSphere Integrated Containers appliance, the VM console screen displays the error `unable to unmarshal ovf environment`.

Problem

When you see this error in the appliance VM console, the appliance is in an unusable state.

Cause

The `ovfenv` data for the appliance is corrupted or missing.

Solution

Perform the following steps to rewrite corrupted or missing `ovfenv` data.

1. In the Flex-based vSphere Client, right-click the appliance VM and select **Power** > **Shut Down Guest OS**.
2. Right-click the appliance again and select **Edit Settings**.
    - In the Flex-based vSphere Web Client, select **vApp Options** and click **OK**.
    - In the HTML5 vSphere Client, select **VM Options** and click **OK**.
3. Verify under **Recent Tasks** that a `Reconfigure virtual machine` task has run on the appliance.
4. Power on the appliance.
5. Open the appliance VM console to verify that the error message does not appear.

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 vSphere Certificate Thumbprints.

- 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 without problems, but for other users the connections fail.

Cause

Some users are attempting to access the Getting Started page, file server, and vSphere Integrated Containers Management Portal 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, and vSphere Integrated Containers Management Portal 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.

Default Volume Store Error

When you create or run a container, the Docker operation fails with an error about a missing volume store.

Problem

Running the container fails with error:

```
docker: Error response from daemon: No volume store named (default) exists.
```

Cause

By default, vic-machine create does not create a volume store when the vSphere administrator deploys a VCH. To run containers from images that use volumes, the vSphere administrator must specify a volume store named default when they deploy the VCH.

Solution

Deploy a VCH by using the `vic-machine create --volume-store` option to create a VCH with a volume store named `default`. See Specify Volume Stores.

Use `docker volume inspect` to get information about the volume.

Logging in to vSphere Integrated Containers Registry from Docker Fails

When you run `docker login vic_registry_address` to log in to vSphere Integrated Containers Registry from the Docker client, you see a `401 Unauthorized` error.

Problem

This error occurs even though you have correctly configured the Docker client with the registry certificate and you have successfully logged in to this registry instance before.

Cause

This issue can occur if the vSphere Integrated Containers appliance has experienced a hard reboot rather than a soft reboot of the guest OS.

Solution

In the vSphere Client, restart the vSphere Integrated Containers appliance by selecting **Power** > **Restart Guest OS**.

vSphere Integrated Containers Security Reference

The Security Reference provides information to allow you to secure your vSphere Integrated Containers implementation.

Service Accounts, Privileges, and User Authentication

vSphere Integrated Containers does not create service accounts and does not assign any vSphere privileges. The vSphere Integrated Containers appliance uses vCenter Single Sign-On user accounts to manage user authentication. You can optionally create example Single Sign-On user accounts for vSphere Integrated Containers Management Portal when you deploy the appliance. For information about the example user accounts, see User Authentication and Deploy the vSphere Integrated Containers Appliance.

VCH Authentication with vSphere

Using `vic-machine` to deploy and manage virtual container hosts (VCHs) requires a user account with vSphere administrator privileges. The `vic-machine create --ops-user` and `--ops-password` options allow a VCH to operate with less-privileged credentials than those that are required to deploy a new VCH. For information about the `--ops-user` option and the permissions that it requires, see Configure the Operations User.

When deploying VCHs, you must provide the certificate thumbprint of the vCenter Server or ESXi host on which you are deploying the VCH. For information about how to obtain and verify vSphere certificate thumbprints, see Obtain vSphere Certificate Thumbprints. Be aware that it is possible to use the `--force` option to run `vic-machine` commands that bypass vSphere certificate verification. For information about the `--force` option, see `--force` in the topic on running `vic-machine` commands.

Docker Client Authentication with VCHs

VCHs authenticate Docker API clients by using client certificates. For information about VCHs and client authentication, see Virtual Container Host Security. Be aware that it is possible to use the `--no-tlsverify` and `--no-tls` options to deploy VCHs that do not authenticate client connections. For information about the `--no-tlsverify` and `--no-tls` options, see Disable Certificate Authentication.

Network Security

All connections to vSphere Integrated Containers Management Portal and Registry are encrypted and secured by HTTPS.

VMware highly recommends using a secure network for the VCH management network. For more information about connections to VCHs in general and the management network in particular, see Virtual Container Host Networks and Configure the Management Network.

External Interfaces, Ports, and Services

The following ports must be open on the vSphere Integrated Containers appliance, VCH endpoint VMs, and container VMs:

ESXi Hosts

ESXi hosts must have the following firewall configuration for VCH deployment:

- 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.

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

vSphere Integrated Containers Appliance

The vSphere Integrated Containers appliance makes the core vSphere Integrated Containers services available.

Port	Protocol	Description
443	HTTPS	Connections to vSphere Integrated Containers Registry from vSphere Integrated Containers Management Portal, VCHs, and Docker clients
4443	HTTPS	Connections to the Docker Content Trust service for vSphere Integrated Containers Registry
8282	HTTPS	Connections to vSphere Integrated Containers Management Portal UI and API
9443	HTTPS	Connections to the appliance intialization and Getting Started page, vSphere Integrated Containers Engine download, and vSphere Client plug-in installer

VCH Endpoint VM

The different network interfaces on a VCH expose different services on different ports. For an overview of the different network interfaces on a VCH, see Virtual Container Host Networks.

Public Interface

Container developers can forward any VCH port that is not used elsewhere to a container VM. For more information about the VCH public interface, see Configure the Public Network.

Bridge Interface

For information about the VCH bridge interface, see Configure Bridge Networks.

Port	Protocol	Description
53	TCP	Connections from the VCH to DNS servers for container name resolution

Client Interface

For information about the VCH client interface, see Configure the Client Network.

Port	Protocol	Description
22	SSH	Connections to the VCH when using `vic-machine debug --enable-ssh` or `vic-machine create/configure --debug 3`.
2375	HTTP	Insecure port for Docker API access if VCH is deployed with `--no-tls`
2376	HTTPS	Secure port for Docker API access if VCH is not deployed with `--no-tls`
2378	HTTPS	Connections to the VCH Administration Portal server
6060	HTTPS	Exposes `pprof` debug data about the VCH if the VCH is running with `vic-machine create --debug` or `vic-machine configure --debug` enabled

For information about VCH TLS options, see Virtual Container Host Security. For information about how debugging VCHs affects VCH behavior, see , see Debug in the topic on configuring general VCH settings and Debug Running Virtual Container Hosts.

Management Interface

For information about the VCH management interface, see Configure the Management Network.

Port	Protocol	Description
443	HTTPS	Outgoing connections from the VCH to vCenter Server and ESXi hosts
2377	HTTPS	Incoming connections from container VMs to the VCH

Container VMs

If container developers do not explicitly expose ports, container VMs do not expose any ports if they are not running in debug mode.

Port	Protocol	Description
6060	HTTPS	Exposes `pprof` debug data about a container VM when a VCH is running with `vic-machine create --debug` enabled

Security Updates and Patches

Download a new version of vSphere Integrated Containers and upgrade your existing appliances, vSphere Client plug-ins, and VCHs. For information about installing security patches, see Upgrading vSphere Integrated Containers.

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. For information about accessing VCH logs, see Access Virtual Container Host Log Bundles.

There are no specific security-related logs for the vSphere Integrated Containers appliance. To access logs for the appliance, see Access vSphere Integrated Containers Appliance Logs.

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 Management Portal securely stores the credentials for access to VCHs, Docker hosts, and registries. Any private elements of those credentials, such as passwords or private keys, are kept encrypted in the vSphere Integrated Containers Management Portal data store.

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 vSphere Certificate Thumbprints.	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 Virtual Container Host Security.	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.