VMware vSphere Integrated Containers for vSphere Administrators

vSphere Integrated Containers 1.1

vmware

Table of Contents

vSphere Integrated Containers for vSphere Administrators

vSphere Integrated Containers for vSphere Administrators provides information about how to use VMware vSphere® Integrated Containers™ as a vSphere administrator.

Product version: 1.1

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

Overview of vSphere Integrated Containers for vSphere Administrators

vSphere Integrated Containers enables IT teams to run traditional and container workloads side-by-side on existing infrastructure seamlessly. This overview is intended for vSphere administrators who intend to use vSphere Integrated Containers to manage container workloads in their vSphere environment.

Introduction to Containers, Images and Volumes

The word "container" is an overloaded one these days. When understanding containers and how they relate to vSphere Integrated Containers, it is helpful to distinguish the *runtime* aspect from the *packaging* aspect.

Runtime

At its most basic, a container is simply a sandbox in which a process can run. The sandbox isolates the process from other processes that are running on the same system. A container has a lifecycle which is typically tied to the lifecycle of the process that it is designed to run. If you start a container, it starts its main process and when that process ends, the container stops. The container might have access to some storage. It typically has an identity on a network.

Conceptually, a container represents many of the same capabilities as a VM. The main difference between the two is the abstraction layer:

- A software container is a sandbox within a guest OS and it is up to the guest to provide the container with its dependencies and to enforce isolation. Multiple containers share the guest kernel, networking, and storage. A container does not boot. It is simply a slice of an already-running OS. The OS running the container is called its *host*.
- In contrast, a VM is a sandbox within a hypervisor. It is the hypervisor that provides a VM with its dependencies, such as virtual disks and NICs. A VM has to boot an OS and its lifecycle is typically tied to that of the OS rather than to that of any one process. By design, a VM is strongly isolated from other VMs and its host.

One of the most interesting facets of containers is how they deal with state. Any data that a container writes is non-persistent by default and is lost when that container is deleted. State, however, can persist beyond the lifespan of a container by attaching a *volume* to it or by sending it over a network. Binary dependencies that the container needs, such as OS libraries or application binaries, are encapsulated in *images* . Images are immutable.

Packaging

One of the most significant benefits of containers is that they allow you to package up the entire environment that an application needs and run it anywhere. You can go to Docker Hub, select from hundreds of thousands of applications and run that application anywhere that you have installed Docker on a compatible OS. The packaging encapsulates the binary dependencies, environment variables, volumes, and even the network configuration.

The format of this packaging is called an *image*. An image is a template from which many containers can instantiate. The Docker image format allows for images to be composed in a parent-child relationship, just like a disk snapshot. This image hierarchy allows containers to share common dependencies. For example, you might have a Debian 8 image that has a child image with Java installed. That Java image might have a child with Tomcat installed. The Debian 8 image might have other children, such as PHP, Python, and so on.

The immutability of the image format means that you never modify an image, you always create a new one. The layered nature of the image format means that you can cache commonly-used layers so that you only need to download or upload the layers that you do not already have. It also means that if you want to patch a particular image, you create a new image and then rebuild all of its children.

The main advantage of the image format is its portability. As long as you have a destination that is running a container engine, for example Docker, you can download and run an image on it. This portability is facilitated by a *registry*. A registry is a service that indexes and stores images. You can run your own private image registry that forms part of a development pipeline. You can *push* images to the registry from development, *pull* them into a test environment for verification, and then *pull* them into a production environment.

What is vSphere Integrated Containers?

vSphere Integrated Containers comprises three major components:

- **vSphere Integrated Containers Engine**, a container runtime for vSphere that allows you to provision containers as virtual machines, offering the same security and functionality of virtual machines in VMware ESXi™ hosts or vCenter Server® instances.
- **vSphere Integrated Containers Registry**, an enterprise-class container registry server that stores and distributes container images. vSphere Integrated Containers Registry extends the Docker Distribution open source project by adding the functionalities that an enterprise requires, such as security, identity and management.
- **vSphere Integrated Containers Management Portal**, a container management portal that provides a UI for DevOps teams to provision and manage containers, including the ability to obtain statistics and information about container instances. Cloud administrators can manage container hosts and apply governance to their usage, including capacity quotas and approval workflows.

These components currently support the Docker image format. vSphere Integrated Containers is entirely Open Source and free to use. Support for vSphere Integrated Containers is included in the vSphere Enterprise Plus license.

vSphere Integrated Containers is designed to solve many of the challenges associated with putting containerized applications into production. It directly uses the clustering, dynamic scheduling, and virtualized infrastructure in vSphere and bypasses the need to maintain discrete Linux VMs as container hosts.

vSphere Integrated Containers allows you, the vSphere administrator, to provide a container management endpoint to a user as a service. At the same time, you remain in complete control over the infrastructure that the container management endpoint service depends on. The main differences between vSphere Integrated Containers and a classic container environment are the following:

- vSphere, not Linux, is the container host:
  - Containers are deployed *as* VMs, not *in* VMs.
  - Every container is fully isolated from the host and from the other containers.
  - vSphere provides per-tenant dynamic resource limits within a vCenter Server cluster
- vSphere, not Linux, is the infrastructure:
  - You can select vSphere networks that appear in the Docker client as container networks.
  - Images, volumes, and container state are provisioned directly to VMFS.
- vSphere is the control plane:
  - Use the Docker client to directly control selected elements of vSphere infrastructure.
  - A container endpoint Service-as-a-Service presents as a service abstraction, not as IaaS.

vSphere Integrated Containers is designed to be the fastest and easiest way to provision any Linux-based workload to vSphere, if that workload can be serialized as a Docker image.

What Does vSphere Integrated Containers Do?

vSphere Integrated Containers gives you, the vSphere administrator, the tools to easily make your vSphere infrastructure accessible to users so that they can provision container workloads into production.

Scenario 1: A Classic Container Environment

In a classic container environment:

- A user raises a ticket and says, "I need Docker".
- You provision a large Linux VM and send them the IP address.
- The user installs Docker, patches the OS, configures in-guest network and storage virtualization, secures the guest, isolates the containers, packages the containers efficiently, and manages upgrades and downtime.

In this scenario, what you have provided is similar to a nested hypervisor that they have to manage and which is opaque to you. If you scale that up to one large Linux VM per tenant, you end up creating a large distributed silo for containers.

Scenario 2: vSphere Integrated Containers

With vSphere Integrated Containers:

- A user raises a ticket and says, "I need Docker".
- You identify datastores, networking, and compute on your cluster that the user can use for their Docker environment.
- You use a utility called `vic-machine` to install a small appliance, called a virtual container host (VCH). The VCH represents an authorization to use the infrastructure that you have identified, into which the user can self-provision container workloads.
- The appliance runs a secure remote Docker API, that is the only access that the user has to the vSphere infrastructure.
- Instead of sending your user a Linux VM, you send them the IP address of the appliance, the port of the remote Docker API, and a certificate for secure access.

In this scenario, you have provided the user with a service portal. This is better for the user because they do not have to worry about isolation, patching, security, backup, and so on. It is better for you because every container that the user deploys is a container VM. You can perform vMotion and monitor container VMs just like all of your other VMs.

If the user needs more compute capacity, in Scenario 1, the pragmatic choice is to power down the VM and reconfigure it, or give the user a new VM and let them deal with the clustering implications. Both of these solutions are disruptive to the user. With vSphere Integrated Containers in Scenario 2, you can reconfigure the VCH in vSphere, or redeploy it with a new configuration in a way that is completely transparent to the user.

vSphere Integrated Containers allows you to select and dictate the appropriate infrastructure for the task in hand:

- Networking: You can select multiple port groups for different types of network traffic, ensuring that all of the containers that a user provisions get the appropriate interfaces on the right networks.
- Storage: You can select different vSphere datastores for different types of state. For example, container state is ephemeral and is unlikely to need to be backed up, but volume state almost certainly should be backed up. vSphere Integrated Containers automatically ensures that state gets written to the appropriate datastore when the user provisions a container.

To summarize, vSphere Integrated Containers gives you a mechanism that allows users to self-provision VMs as containers into your virtual infrastructure.

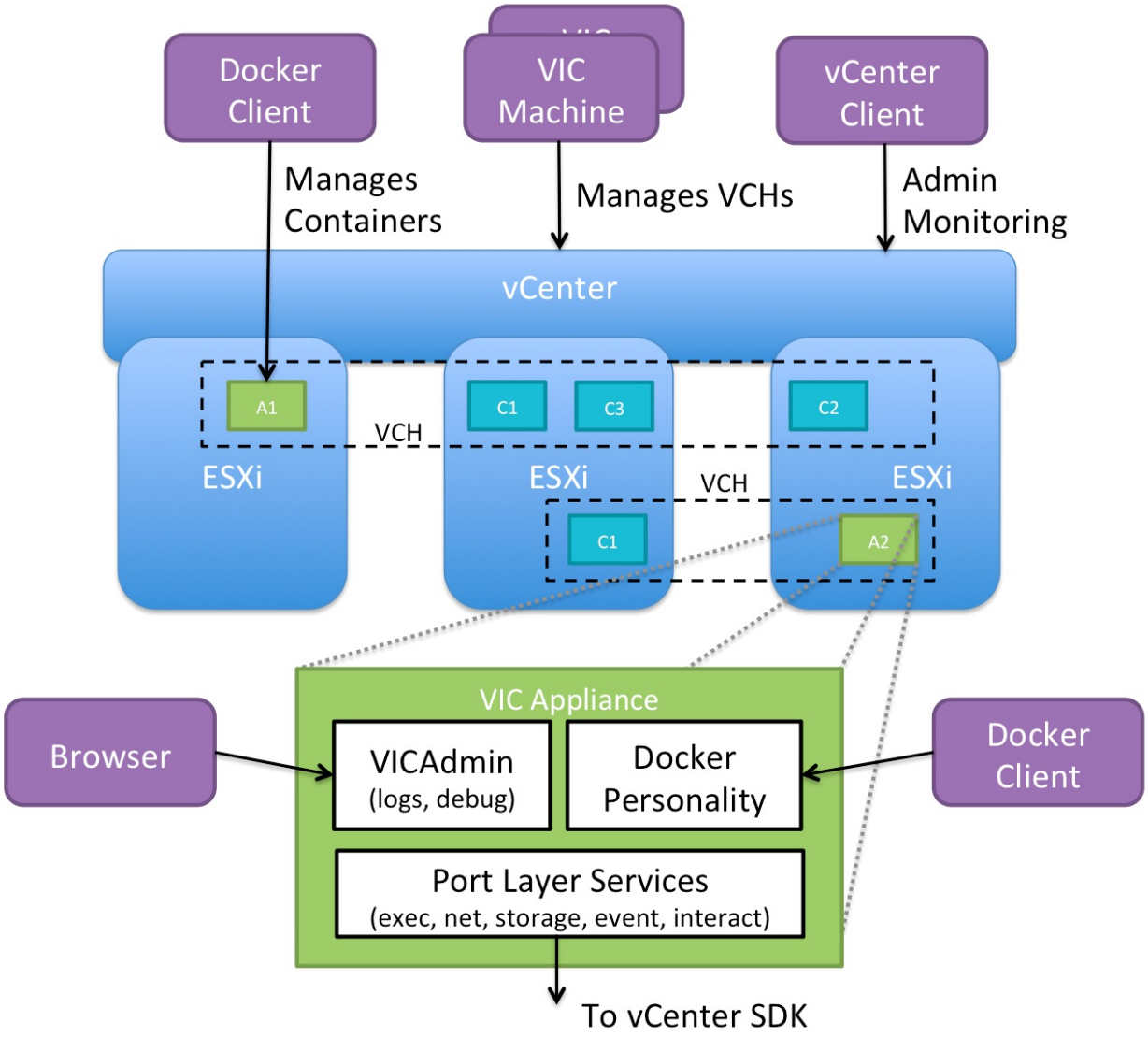
What Is vSphere Integrated Containers Engine For?

vSphere Integrated Containers Engine currently offers a subset of the Docker API. It is designed to specifically address the provisioning of containers into production, solving many of the problems highlighted in What Does vSphere Integrated Containers Engine Do?.

vSphere Integrated Containers Engine exploits the portability of the Docker image format to present itself as an enterprise deployment target. Developers build containers on one system and push them to a registry. Containers are tested by another system and are approved for production. vSphere Integrated Containers Engine can then pull the containers out of the registry and deploy them to vSphere.

vSphere Integrated Containers Engine Concepts

If you consider a Venn diagram with "What vSphere Does" in one circle and "What Docker Does" in another, the overlap is significant. The objective of vSphere Integrated Containers Engine is to take as much of vSphere as possible and layer whatever Docker capabilities are missing on top, reusing as much of Docker's own code as possible. The result should not sacrifice the portability of the Docker image format and should be completely transparent to a Docker client. The following sections describe key concepts and components that make this possible.



Container VMs

The container VMs that vSphere Integrated Containers Engine creates have all of the characteristics of software containers:

- An ephemeral storage layer with optionally attached persistent volumes.
- A custom Linux guest OS that is designed to be "just a kernel" and that needs images to be functional.
- A mechanism for persisting and attaching read-only binary image layers.
- A PID 1 guest agent *tether* extends the control plane into the container VM.
- Various well-defined methods of configuration and state ingress and egress
- Automatically configured to various network topologies.

The provisioned container VM does not contain any OS container abstraction.

- The container VM boots from an ISO that contains the Photon Linux kernel. Note that container VMs do not run the full Photon OS.
- The container VM is configured with a container image that is mounted as a disk.
- Container image layers are represented as a read-only VMDK snapshot hierarchy on a vSphere datastore. At the top of this hierarchy is a read-write snapshot that stores ephemeral state.
- Container volumes are formatted VMDKs that are attached as disks and indexed on a datastore.

- Networks are distributed port groups that are attached as vNICs.

Virtual Container Hosts

A virtual container host (VCH) is the functional equivalent of a Linux VM that runs Docker, but with some significant benefits. A VCH represents the following elements:

- A clustered pool of resource into which to provision container VMs.
- A single-tenant container namespace.
- An isolated Docker API endpoint.
- Authorization to use and configure pre-approved virtual infrastructure.
- A private network that containers are attached to by default.

If you deploy a VCH in a vCenter Server cluster it spans all of the hosts in the cluster, providing the same flexibility and dynamic use of host resources as is the norm.

A VCH is functionally distinct from a traditional container host in the following ways:

- It naturally encapsulates clustering and dynamic scheduling by provisioning to vSphere targets.
- The resource constraints are dynamically configurable with no impact on the containers.
- Containers do not share a kernel.
- There is no local image cache. This is kept on a datastore in the cluster that you specify when you deploy a VCH.
- There is no read-write shared storage

A VCH is a multi-functional appliance that you deploy as a vApp in a vCenter Server cluster or as a resource pool on an ESXi host. The vApp or resource pool provides a useful visual parent-child relationship in the vSphere Client so that you can easily identify the container VMs that are provisioned into a VCH. You can also specify resource limits on the vApp. You can provision multiple VCHs onto a single ESXi host, into a vSphere resource pool, or into a vCenter Server cluster.

The VCH Endpoint VM

The VCH endoint VM is the VM that runs inside the VCH vApp or resource pool. There is a 1:1 relationship between a VCH and a VCH endpoint VM. The VCH endpoint VM provides the following functions:

- Runs the services that a VCH requires.
- Provides a secure remote API to a client.
- Receives Docker commands and translates those commands into vSphere API calls and vSphere infrastructure constructs.
- Provides network forwarding so that ports to containers can be opened on the VCH endoint VM and the containers can access a public network.
- Manages the lifecycle of the containers, the image store, the volume store, and the container state
- Provides logging and monitoring of its own services and of its containers.

The lifecycle of the VCH endpoint VM is managed by a utility called `vic-machine`.

The `vic-machine` Utility

The `vic-machine` utility is a binary for Windows, Linux, and OSX that manages the lifecycle of VCHs. `vic-machine` has been designed for use by vSphere administrators. It takes pre-existing compute, network, storage and a vSphere user as input and creates a VCH as output. It has the following additional functions:

- Creates certificates for Docker client TLS authentication.
- Checks that the prerequisites for VCH deployment are met on the cluster or host, namely that the firewall, licenses, and so on are configured correctly.
- Configures existing VCHs for debugging.
- Lists, inspects, upgrades, and deletes VCHs.

What Is vSphere Integrated Containers Registry?

vSphere Integrated Containers Registry is an enterprise-class registry server that you can use to store and distribute container images. vSphere Integrated Containers Registry allows DevOps administrators to organize image repositories in projects, and to set up role-based access control to those projects to define which users can access which repositories. vSphere Integrated Containers Registry also provides rule-based replication of images between registries, implements Docker Content Trust, and provides detailed logging for project and user auditing.

For a more detailed overview of vSphere Integrated Containers Registry, see Managing Images, Projects, and Users with vSphere Integrated Containers Registry in *vSphere Integrated Containers for DevOps Administrators*.

What Is vSphere Integrated Containers Management Portal?

vSphere Integrated Containers Management Portal is a highly scalable and very lightweight container management platform for deploying and managing container based applications. It is designed to have a small footprint and boot extremely quickly. vSphere Integrated Containers Management Portal is intended to provide DevOps administrators with automated deployment and lifecycle management of containers.

- Rule-based resource management, allowing DevOps administrators to set deployment preferences which let vSphere Integrated Containers Management Portal manage container placement.
- Live state updates that provide a live view of the container system.
- Multi-container template management, that enables logical multi-container application deployments.

For a more information about vSphere Integrated Containers Management Portal, see View and Manage VCHs, Add Registries, and Provision Containers Through the Management Portal in *vSphere Integrated Containers for DevOps Administrators*.

Interoperability of vSphere Integrated Containers with Other VMware Software

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

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

Performing Operations on VCHs and Container VMs in vSphere

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

VMware vRealize® Suite

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

VMware vSphere vMotion®

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

VMware vSphere High Availability

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

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.

VMware vSAN™

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.

Networks Used by vSphere Integrated Containers Engine

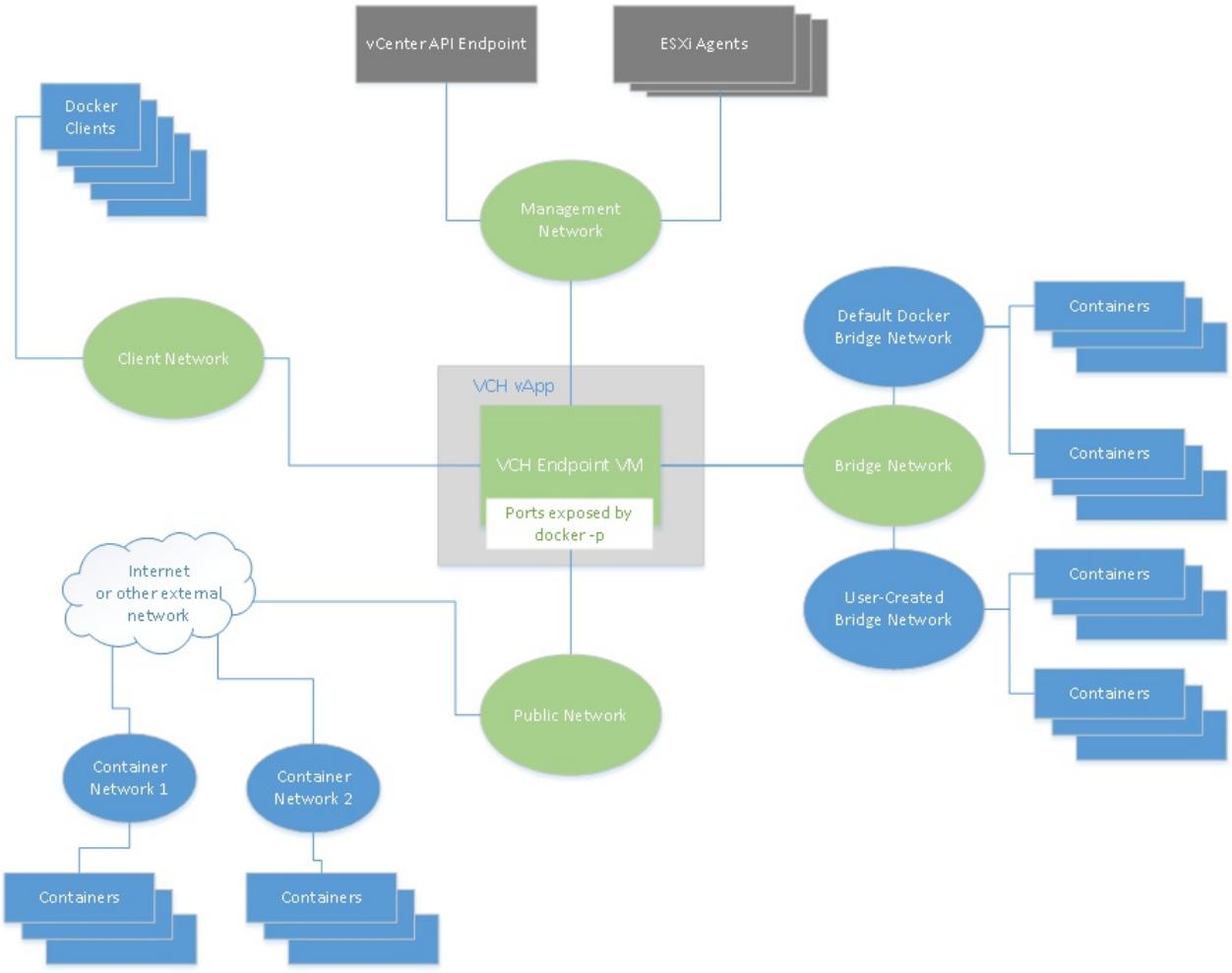
You can configure networks on a virtual container host (VCH) that are tied into the vSphere infrastructure. You define which networks are available to a VCH when you deploy the VCH.

Each network that a VCH uses is a port group on either a vCenter Server instance or ESXi host.

This topic provides an overview of the different network types that virtual container hosts use.

High-Level View of VCH Networking

The image below shows a high-level view of the networks that a VCH uses and how they connect to your vSphere environment and to the Docker environment that container developers use.



The following sections describe each of the VCH network types, that are represented in the image by green elipses. The blue shapes represent Docker objects, and the gray shapes represent vSphere.

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

Management Network

The network for communication between the VCH and vCenter Server and ESXi hosts. The VCH uses this network to provide the `attach` function of the Docker API.

IMPORTANT: Because the management network provides access to your vSphere environment, and because container VMs use this network to communicate with the VCH, always use a secure network for the management network. Ideally, use separate networks for the management network and the container networks.

You define the management network by setting the `--management-network` option when you run `vic-machine create`. For more detailed information about management networks, see the section on the `--management-network` option in VCH Deployment Options.

Public Network

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

You define the public network by setting the `--public-network` option when you run `vic-machine create`. For more detailed information about management networks, see the section on the `--public-network` option in VCH Deployment Options.

Client Network

The network on which the VCH endpoint VM makes the Docker API available to Docker clients. The client network isolates the Docker endpoints from the public network.

You define the Docker management endpoint network by setting the `--client-network` option when you run `vic-machine create`. For more detailed information about Docker management endpoint networks, see the section on the `--client-network` option in VCH Deployment Options.

Bridge Network

The network or networks that container VMs use to communicate with each other. Each VCH requires a unique bridge network. The bridge network is a port group on a distributed virtual switch.

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

You define the bridge networks by setting the `--bridge-network` option when you run `vic-machine create`. For more detailed information about bridge networks, see the section on the `--bridge-network` option in VCH Deployment Options.

Container application developers can also use `docker network create` to create additional bridge networks. These networks are represented by the User-Created Bridge Network in the image above. Additional bridge networks are created by IP address segregation and are not new port groups. You can define a range of IP addresses that additional bridge networks can use by defining the `bridge-network-range` option when you run `vic-machine create`. For more detailed information about how to set bridge network ranges, see the section on the `bridge-network-range` option.

Container Networks

Container networks allow the vSphere administrator to make vSphere networks directly available to containers. This is done during deployment of a VCH by providing a mapping of the vSphere network name to an alias that is used inside the VCH endpoint VM. The mapped networks are then listed as available by the Docker API. Running `docker network ls` shows these 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_` or `docker network connect`. The containers connected to container networks are connected directly to these networks, and traffic does not route though the VCH endpoint VM using NAT.

You can share one network alias between multiple containers. For more detailed information about setting up container networks, see the sections on the `container-network-xxx` options in Virtual Container Host Deployment Options.

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.

The installation process involves several steps.

- Download the OVA from http://www.vmware.com/go/download-vic.
- Deploy the OVA, providing configuration information for vSphere Integrated Containers Registry and vSphere Integrated Containers Management Portal. The OVA deploys an appliance VM that provides the following services:
  - Runs vSphere Integrated Containers Registry
  - Runs vSphere Integrated Containers Management Portal
  - Makes the vSphere Integrated Containers Engine binaries available for download
  - Hosts the vSphere Client plug-in packages for vCenter Server
- Run the scripts to install the vSphere Client plug-ins on vCenter Server.
- Run the command line utility, `vic-machine`, to deploy and manage virtual container hosts.

Download vSphere Integrated Containers

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

Official Releases

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

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

Open Source Builds of the vSphere Integrated Containers Components

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 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.
- Build the latest source version of the vSphere Integrated Containers components:
  - vSphere Integrated Containers Engine
  - vSphere Integrated Containers Registry
  - vSphere Integrated Containers Portal

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

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

Deploy the vSphere Integrated Containers Appliance

You install vSphere Integrated Containers by deploying a virtual appliance. The appliance runs vSphere Integrated Containers Registry and vSphere Integrated Containers Management Portal, and makes the download of the vSphere Integrated Containers Engine binaries available.

Prerequisites

- You downloaded the OVA installer from the official vSphere Integrated Containers downloads page on vmware.com.
- Deploy the appliance to a vSphere environment that meets the minimum system requirements:
    - 2 vCPUs
    - 8GB RAM
    - 80GB free disk space on the datastore

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
    - Destination datacenter or folder
    - Destination host, cluster, or resource pool
    - Accept the end user license agreements (EULA)
    - Disk format and destination datastore
    - 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.

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

    Leave the networking properties blank to use DHCP.

5. Expand **Registry Configuration** to configure the deployment of vSphere Integrated Containers Registry.
    - If you do not want to deploy vSphere Integrated Containers Registry, uncheck the **Deploy Registry** check box.
    - 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 Notary service for vSphere Integrated Containers Registry.
    - In the **Registry Admin Password** text box, set the password for the vSphere Integrated Containers Registry admin account.
    - In the **Database Password** text box, set the password for the root user of the MySQL database that vSphere Integrated Containers Registry uses.
    - Optionally check the **
    - Collection** check box to enable garbage collection when the appliance reboots.
    - To use custom certificates to authenticate connections to vSphere Integrated Containers Registry, optionally paste the content of the appropriate certificate and key files in the **SSL Cert** and **SSL Cert Key** text boxes. Leave the text boxes blank to use auto-generated certificates.
6. Expand **Management Portal Configuration** to configure the deployment of vSphere Integrated Containers Management Portal.
    - If you do not want to deploy vSphere Integrated Containers Management Portal, uncheck the **Deploy Management Portal** check box.
    - 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 and key files in the **SSL Cert** and **SSL Cert Key** text boxes. Leave the text boxes blank to use auto-generated certificates.
7. Expand **Fileserver Configuration** to configure the file server from which you download vSphere Integrated Containers Engine and which publishes the plug-in packages for the vSphere Client.
    - In the **Fileserver Port** text box, optionally change the port on which the vSphere Integrated Containers Engine file server runs.
    - To use custom certificates to authenticate connections to the vSphere Integrated Containers Engine file server, optionally paste the content of the appropriate certificate and key files in the **SSL Cert** and **SSL Cert Key** text boxes. Leave the text boxes blank to use auto-generated certificates.
8. Click **Next** and **Finish** to deploy the vSphere Integrated Containers appliance.
9. When the deployment completes, power on the appliance VM.

What to Do Next

- Go to https://*vic_appliance_address*:9443, download the vSphere Integrated Containers Engine binaries, and start deploying virtual container hosts (VCHs). If the vSphere Integrated Containers appliance uses a different port for the vSphere Integrated Containers Engine file server, replace 9443 with the appropriate port. For information about deploying VCHs, see Using vic-machine to Deploy VCHs.
- Install the vSphere Client plug-ins for vSphere Integrated Containers. You do not need to download the client plug-in file. You pass the URL https://*vic_appliance_address*:*port* to the script that you run to install the vSphere Integrated Containers plug-ins. For information about installing the plug-ins, see Installing the vSphere Client Plug-ins.
- Log in to vSphere Integrated Containers Registry at https://*vic_appliance_address*:443. If the vSphere Integrated Containers appliance uses a different port for vSphere Integrated Containers Registry, replace 443 with the appropriate port.
- Log in to vSphere Integrated Containers Management Portal: https://*vic_appliance_address*:8282. If the vSphere Integrated Containers appliance uses a different port for vSphere Integrated Containers Management Portal, replace 8282 with the appropriate port.

Installing the vSphere Client Plug-Ins

vSphere Integrated Containers provides two UI plug-ins for vSphere:

- Abasic Flex-based plug-in that adds information about virtual container hosts (VCHs) and container VMs in the Flex-based vSphere Web Client. The basic plug-in works with the Flex-based vSphere Web Client for both vSphere 6.0 and 6.5.
- An HTML5 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
- Access the vSphere Integrated Containers View
- Find VCH Information in the vSphere Clients
- Find Container Information in the vSphere Clients

Install the Client Plug-Ins on vCenter Server for Windows

You install the vSphere Client plug-ins for vSphere Integrated Containers by using the Web server that runs in the vSphere Integrated Containers appliance.

- You can install the the Flex-based vSphere Web Client plug-in on vCenter Server 6.0 or 6.5.
- You can install the the HTML5 vSphere Client plug-in on vCenter Server 6.5.

Prerequisites

- You deployed the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- Open a Web browser on a Windows system, go to https://*vic_appliance_address*:9443, and download and unpack the vSphere Integrated Containers Engine package. You must use a Windows system to run the script to install the plug-in on a vCenter Server instance that runs on Windows. For example, download the package to the Windows system on which vCenter Server is running. If the vSphere Integrated Containers appliance uses a different port for the vSphere Integrated Containers Engine file server, replace 9443 with the appropriate port.
- Copy the thumbprint of the Communications Server certificate from https://*vic_appliance_address*:*port*. For information about how to view the certificate thumbprint, see the documentation for your type of browser.

Procedure

1. On the Windows system on which you have downloaded and unpacked vSphere Integrated Containers Engine, open the `\vic\ui\vCenterForWindows\configs` file in a text editor.
2. Enter the IPv4 address or FQDN of the vCenter Server instance on which to install the plug-in.

   ```
   SET target_vcenter_ip=vcenter_server_address
   ```

3. Enter the URL on which the vSphere Integrated Containers appliance publishes the client plug-in bundle.

   ```
   SET vic_ui_host_url=https://vic_appliance_address:port
   ```

4. Provide the SHA-1 thumbprint of the Web server.

   ```
   SET vic_ui_host_thumbprint=thumbprint
   ```

   NOTE: Use colon delimitation in the thumbprint. Do not use space delimitation.
5. Specify the version of the plug-in to install.
   - To install the plug-in for the HTML5 vSphere Client, leave the vCenter Server version set to `6.5`.
   - To install the plug-in for the Flex-based vSphere Web Client, set the vCenter Server version to `6.0`.

     ```
     SET target_vc_version=6.0
     ```

     NOTE: When installing the Flex-based plug-in, you must set the version to `6.0` even if you are running vCenter Server 6.5.
6. Save and close the `configs` file.
7. Open a Windows command prompt, navigate to `\vic\ui\vCenterForWindows`, and run the installer.

   ```
   install.bat
   ```

8. Enter the user name and password for the vCenter Server administrator account.
9. When installation finishes, if you are logged into the vSphere Web Client, log out then log back in again.

What to Do Next

Verify the deployment of the plug-in.

- If you installed the HTML5 plug-in, see Access the vSphere Integrated Containers View in the HTML5 vSphere Client.
- If you installed the Flex plug-in, see Find VCH Information in the vSphere Clients.

If the vSphere Integrated Containers plug-in does not appear, restart the vSphere Client service. For instructions about how to restart the vSphere Client service, see vSphere Integrated Containers Plug-In Does Not Appear.

Install the Client Plug-Ins on a vCenter Server Appliance

You install the vSphere Client plug-ins for vSphere Integrated Containers by using the Web server that runs in the vSphere Integrated Containers appliance.

- You can install the the Flex-based vSphere Web Client plug-in on vCenter Server 6.0 or 6.5.
- You can install the the HTML5 vSphere Client plug-in on vCenter Server 6.5.

Prerequisites

- Go to the vCenter Server Appliance Management Interface (VAMI) at https://*vcsa_address*:5480, click **Access**, and make sure that Bash Shell is enabled.
- You deployed the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- Go to https://*vic_appliance_address*:9443 and download and unpack the vSphere Integrated Containers Engine package. If the vSphere Integrated Containers appliance uses a different port for the vSphere Integrated Containers Engine file server, replace 9443 with the appropriate port.
- Copy the thumbprint of the Communications Server certificate from https://*vic_appliance_address*:*port*. For information about how to view the certificate thumbprint, see the documentation for your type of browser.

Procedure

1. On the system on which you have downloaded and unpacked vSphere Integrated Containers Engine, open the appropriate configuration file in a text editor.
   - For the HTML5 plug-in, open `/vic/ui/HTML5Client/configs`.
   - For the Flex plug-in, open `/vic/ui/VCSA/configs`.
2. Enter the IPv4 address or FQDN of the vCenter Server Appliance on which to install the plug-in.

   ```
   VCENTER_IP="vcenter_server_address"
   ```

3. Enter the URL on which the vSphere Integrated Containers appliance publishes the client plug-in bundle.

   ```
   SET VIC_UI_HOST_URL="https://vic_appliance_address:port"
   ```

4. Provide the SHA-1 thumbprint of the Web server.

   ```
   VIC_UI_HOST_THUMBPRINT="thumbprint"
   ```

   NOTE: Use colon delimitation in the thumbprint. Do not use space delimitation.
5. Save and close the `configs` file.
6. (Optional) If you unpacked vSphere Integrated Containers Engine on a Windows system, open the `/vic/ui/HTML5Client/install.sh` file in a text editor and point `PLUGIN_MANAGER_BIN` to the Windows UI executable.

   Before:

   ```
   if [[ $(echo $OS | grep -i "darwin") ]] ; then
       PLUGIN_MANAGER_BIN="../../vic-ui-darwin"
     else
       PLUGIN_MANAGER_BIN="../../vic-ui-linux"
   ```

   After:

   ```
   if [[ $(echo $OS | grep -i "darwin") ]] ; then
       PLUGIN_MANAGER_BIN="../../vic-ui-darwin"
     else
       PLUGIN_MANAGER_BIN="../../vic-ui-windows"
   ```

7. Open a command prompt, navigate to `/vic/ui/VCSA`, and run the installer.

   ```
   ./install.sh
   ```

   Make sure that `install.sh` is executable by running `chmod` before you run it.

8. Enter the user name and password for the vCenter Server administrator account.
9. When installation finishes, if you are logged into the vSphere Client or vSphere Web Client, log out then log back in again.

What to Do Next

Verify the deployment of the plug-in.

- If you installed the HTML5 plug-in, see Access the vSphere Integrated Containers View in the HTML5 vSphere Client.
- If you installed the Flex plug-in, see Find VCH Information in the vSphere Clients.

If the vSphere Integrated Containers plug-in does not appear, restart the vSphere Client service. For instructions about how to restart the vSphere Client service, see vSphere Integrated Containers Plug-In Does Not Appear.

Access the vSphere Integrated Containers View in the HTML5 vSphere Client

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

IMPORTANT: Do not use the vSphere Client or to perform operations on virtual container host (VCH) appliances or container VMs. Specifically, using the vSphere Client to power off, power on, or delete VCH appliances or container VMs can cause vSphere Integrated Containers Engine to not function correctly. Always use `vic-machine` to perform operations on VCHs. Always use Docker commands to perform operations on containers.

Prerequisites

- You are running vCenter Server 6.5.
- You installed the HTML5 plug-in for vSphere Integrated Containers.

Procedure

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

Result

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

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

Find VCH Information in the vSphere Clients

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

IMPORTANT: Do not use the vSphere Client to perform operations on VCH appliances or container VMs. Specifically, using the vSphere Client to power off, power on, or delete VCH appliances or container VMs can cause vSphere Integrated Containers Engine to not function correctly. Always use `vic-machine` to perform operations on VCHs. Always use Docker commands to perform operations on containers.

Prerequisites

- You deployed a VCH.
- You installed either or both of HTML5 or Flex-based plug-ins for vSphere Integrated Containers.

Procedure

1. Log in to either the HTML5 vSphere Client or the Flex-based vSphere Web Client.
2. On the **Home** page, select **Hosts and Clusters**.
3. Expand the hierarchy of vCenter Server objects to navigate to the VCH vApp.
4. Expand the VCH vApp and select the VCH endpoint VM.
5. Click the **Summary** tab for the VCH endpoint VM and scroll down to the Virtual Container Host portlet.

Result

Information about the VCH appears in the VCH portlet in the **Summary** tab:

- The address of the Docker API endpoint for this VCH
- A link to the VCH Admin portal for the VCH, from which you can obtain health information and download log bundles for the VCH.

Find Container Information in the vSphere Clients

After you have installed either or both of the plug-ins for vSphere Integrated Containers, you can find information about the container VMs that are running in virtual container hosts (VCHs).

IMPORTANT: Do not use the vSphere Client to perform operations on VCH appliances or container VMs. Specifically, using the vSphere Client to power off, power on, or delete VCH appliances or container VMs can cause vSphere Integrated Containers Engine to not function correctly. Always use `vic-machine` to perform operations on VCHs. Always use Docker commands to perform operations on containers.

Prerequisites

- You deployed a VCH and pulled and ran at least one container.
- You installed either or both of HTML5 or Flex-based plug-ins for vSphere Integrated Containers.

Procedure

1. Log in to either the HTML5 vSphere Client or the Flex-based vSphere Web Client.
2. On the **Home** page, select **Hosts and Clusters**.
3. Expand the hierarchy of vCenter Server objects to navigate to the VCH vApp.
4. Expand the VCH vApp and select a container VM.
5. Click the **Summary** tab for the container VM and scroll down to the **Container** portlet.

Result

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.

Using `vic-machine` to Deploy Virtual Container Hosts

You use the vSphere Integrated Containers Engine `vic-machine` utility to deploy virtual container hosts.

Contents of the vSphere Integrated Containers Engine Binaries

After you download and unpack a vSphere Integrated Containers Engine binary bundle, you obtain following files:

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

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

Environment Prerequisites for VCH Deployment

Before you deploy vSphere Integrated Containers Engine, you must ensure that your vSphere infrastructure meets the requirements.

Supported Platforms for `vic-machine`

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

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

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

Supported vSphere Configurations

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

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

Caveats and limitations:

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

License Requirements

vSphere Integrated Containers Engine requires a vSphere Enterprise Plus license.

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

ESXi Host Firewall Requirements

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

- Allow outbound TCP traffic to port 2377 on the endpoint VM, for use by the interactive container shell.

- Allow inbound HTTPS/TCP traffic on port 443, for uploading to and downloading from datastores.

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

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

ESXi Host Storage Requirements for vCenter Server Clusters

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

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

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

General Network Requirements

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

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

vCenter Server Network Requirements

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

- Create a distributed virtual switch with a port group for each VCH, for use as the bridge network. You can create multiple port groups on the same distributed virtual switch, but each VCH requires its own port group for the bridge network.
- Optionally create port groups for use as mapped container networks.
- All hosts in a cluster must be attached to the port groups that you will use for the VCH bridge network and for any mapped container networks.
- Isolate the bridge network and any mapped container networks. You can isolate networks by using a separate VLAN for each network.

For information about bridge networks and container networks, see the `--bridge-network` and `--container-network` options in *VCH Deployment Options*.

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

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

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

Open the Required Ports on ESXi Hosts

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

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

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

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

Prerequisites

- Deploy the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- Download the vSphere Integrated Containers Engine bundle, `vic_1.1.0-version.tar.gz`, from https://*vic_appliance_address*:9443 and unpack it on your working machine. If you configured the appliance to use a different port for the file server, replace 9443 with the appropriate port.

Procedure

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

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

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

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

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

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

- The address of the vCenter Server instance and datacenter, or the ESXi host, on which to deploy the VCH in the `--target` option.
- The user name and password for the vCenter Server instance or ESXi host in the `--user` and `--password` options.
- In the case of a vCenter Server cluster, the name of the cluster in the `--compute-resource` option.

- The thumbprint of the vCenter Server or ESXi host certificate in the `--thumbprint` option, if they use untrusted, self-signed certificates.
- The `--allow` option to open the port.

Deploy a VCH to an ESXi Host with No vCenter Server

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

Prerequisites

- Deploy the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- Download the vSphere Integrated Containers Engine bundle, `vic_1.1.0-version.tar.gz`, from https://*vic_appliance_address*:9443 and unpack it on your working machine. If you configured the appliance to use a different port for the file server, replace 9443 with the appropriate port.
- Create or obtain an ESXi host with the following configuration:
  - One datastore
  - The VM Network is present
  - You can use a nested ESXi host for this example
- Verify that the ESXi host meets the requirements in Environment Prerequisites for VCH Deployment.
- Make sure that the correct firewall port is open on the ESXi host. For information about how to open ports on ESXi hosts, see Open the Required Ports on ESXi Hosts.
- Familiarize yourself with the vSphere Integrated Containers Engine binaries, as described in Contents of the vSphere Integrated Containers Engine Binaries.
- Familiarize yourself with the options of the `vic-machine create` command described in VCH Deployment Options.

Procedure

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

   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. 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
     --force
     ```

   - Windows:

     ```
     $ vic-machine-windows create
     --target esxi_host_address
     --user root
     --password "esxi_host_p@ssword"
     --no-tlsverify
     --force
     ```

   - Mac OS:

     ```
     $ vic-machine-darwin create
     --target esxi_host_address
     --user root
     --password 'esxi_host_p@ssword'
     ```

```
--no-tlsverify
--force
```

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

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

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

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

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

Result

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

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

What to Do Next

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

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

Deploy a VCH to a Basic vCenter Server Cluster

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

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

Prerequisites

- Deploy the vSphere Integrated Containers appliance. For information about deploying the appliance, see Deploy the vSphere Integrated Containers Appliance.
- Download the vSphere Integrated Containers Engine bundle, `vic_1.1.0-version.tar.gz`, from https://*vic_appliance_address*:9443 and unpack it on your working machine. If you configured the appliance to use a different port for the file server, replace 9443 with the appropriate port.
- Create or obtain a vCenter Server instance with the following configuration:
  - One datacenter
  - One cluster with two ESXi hosts and DRS enabled. You can use nested ESXi hosts for this example.
  - A shared datastore, that is accessible by both of the ESXi hosts.
  - The VM Network is present
  - One distributed virtual switch with one port group named `vic-bridge`
- Verify that your vCenter Server instance and both of the ESXi hosts in the cluster meet the requirements in Environment Prerequisites for VCH Deployment.
- Make sure that the correct firewall ports are open on the ESXi hosts. For information about how to open ports on ESXi hosts, see Open the Required Ports on ESXi Hosts.
- Familiarize yourself with the vSphere Integrated Containers Engine binaries, as described in Contents of the vSphere Integrated Containers Engine Binaries.
- Familiarize yourself with the options of the `vic-machine create` command described in VCH Deployment Options.

Procedure

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

   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. 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
     --force
     ```

   - 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
--force
```

- 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
--force
```

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

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

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

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

Result

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

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

What to Do Next

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

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

Verify the Deployment of a VCH

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

IMPORTANT: Do not use the vSphere Client or vSphere Web Client to perform operations on VCH appliances or container VMs. Specifically, using the vSphere Client or vSphere Web Client to power off, power on, or delete VCH appliances or container VMs can cause vSphere Integrated Containers Engine to not function correctly. Always use `vic-machine` to perform operations on VCHs. Always use Docker commands 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 to deploy a VCH to either an ESXi host or to a vCenter Server instance.
- You ran `vic-machine create` with the `--no-tlsverify` option.
- You have installed a Docker client.
- If you deployed the VCH to vCenter Server, connect a vSphere Web Client to that vCenter Server instance.
- If you deployed the VCH to an ESXi host, connect a vSphere Client to that host.

Procedure

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

   The vApp or resource pool contains the VCH endpoint VM.
2. Run the `docker info` command to confirm that you can connect to the VCH.

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

     vSphere Integrated Containers Engine creates a folder 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 Web Client or vSphere Client.
   - vCenter Server: Go to **Hosts and Clusters** and expand the VCH vApp.
   - ESXi host: Go to **Inventory** and expand the VCH resource pool.

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

8. View the container VM files in the vSphere Web Client or vSphere Client.
   - vCenter Server: Go to **Storage** and select the datastore that you designated as the image store.
   - ESXi host: Click the **Summary** tab for the ESXi host, right-click the datastore that you designated as the image store, and select **Browse Datastore**.

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

VCH Deployment Options

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

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

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

- Options for Specifying a Static IP Address for the VCH Endpoint VM
- Options for Configuring a Non-DHCP Network for Container Traffic
- Options to Configure VCHs to Use Proxy Servers
- Advanced Resource Management Options
- Other Advanced Options

vSphere Target Options

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

`--target`

Short name: `-t`

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

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

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

  ```
  --target esxi_host_address
  ```

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

  ```
  --target vcenter_server_address
  ```

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

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

  Wrap the user name or password in single quotes (Linux or Mac OS) or double quotes (Windows) if they include special characters.

  ```
  'vcenter_or_esxi_usern@me':'p@ssword'@vcenter_or_esxi_address
  ```

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

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

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

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

  Wrap the datacenter name in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if it includes spaces.

  ```
  --target vcenter_server_address/'datacenter name'
  ```

`--user`

Short name: `-u`

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

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

```
--user esxi_or_vcenter_server_username
```

Wrap the user name in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if it includes special characters.

```
--user 'esxi_or_vcenter_server_usern@me'
```

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

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

`--password`

Short name: `-p`

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

```
--password esxi_host_or_vcenter_server_password
```

Wrap the password in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if it includes special characters.

```
--password 'esxi_host_or_vcenter_server_p@ssword'
```

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

`--compute-resource`

Short name: `-r`

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

If the vCenter Server instance on which you are deploying a VCH only includes a single instance of a standalone host or a 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 to an ESXi host and you do not specify `--compute-resource`, `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, specify the name of the resource pool:

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

- To deploy to a vCenter Server instance that has more than one standalone host 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 more than one cluster, 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, 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, specify the names of the target cluster and the resource pool:

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

- Wrap the resource names in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if they include spaces:

  ```
  --compute-resource 'cluster name'/'resource pool name'
  ```

`--thumbprint`

Short name: None

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

NOTE If your vSphere environment uses untrusted, self-signed certificates, you can run `vic-machine create` without the `--thumbprint` option by using the `--force` option. However, running `vic-machine create` with the `--force` option rather than providing the certificate thumbprint is not recommended, because it permits man-in-the-middle attacks to go undetected.

To obtain the thumbprint of the vCenter Server or ESXi host certificate, run `vic-machine create` without the specifying the `--thumbprint` or `--force` options. The deployment of the VCH fails, but the resulting error message includes the required certificate thumbprint. You can copy the thumbprint from the error message and run vic-machine create again, including the `--thumbprint` option. If you obtain the thumbprint by other means, use upper-case letters and colon delimitation rather than space delimitation when you specify `--thumbprint`.

```
--thumbprint certificate_thumbprint
```

Security Options

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

- Restrict access to the Docker API with Auto-Generated Certificates
- Restrict access to the Docker API with Custom Certificates
- Do Not Restrict Access to the Docker API

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

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

Restrict Access to the Docker API with Auto-Generated Certificates

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

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

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

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

`--tls-cname`

Short name: None

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

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

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

- Checks for an existing certificate in either a folder that has the same name as the VCH that you are deploying, or in a location that you specify in the `--cert-path` option. If a valid certificate exists that includes the same Common Name attribute as the one that you specify in `--tls-cname`, `vic-machine create` reuses it. Reusing certificates allows you to delete and recreate VCHs for which you have already distributed the certificates to container developers.
- If certificates are present in the certificate folder that include a different Common Name attribute to the one that you specify in `--tls-cname`, `vic-machine create` fails.
- If a certificate folder does not exist, `vic-machine create` creates a folder with the same name as the VCH, or creates a folder in the location that you specify in the `--cert-path` option.
- If valid certificates do not already exist, `vic-machine create` creates the following trusted CA, server, and client certificate/key pairs in the certificate folder:

    - `ca.pem`
    - `ca-key.pem`
    - `cert.pem`
    - `key.pem`
    - `server-cert.pem`
    - `server-key.pem`
- Creates a browser-friendly PFX client certificate, `cert.pfx`, to use to authenticate connections to the VCH Admin portal for the VCH.

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

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

- Activates TLS client verification.

  ```
  DOCKER_TLS_VERIFY=1
  ```

- The path to the client certificates.

  ```
  DOCKER_CERT_PATH=path_to_certs
  ```

- The address of the VCH.

  ```
  DOCKER_HOST=vch_address:2376
  ```

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

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

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

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

`--cert-path`

Short name: none

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

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

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

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

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

`--certificate-key-size`

Short name: `--ksz`

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

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

`--organization`

Short name: None

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

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

```
--organization organization_name
```

Restrict Access to the Docker API with Custom Certificates

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

`--cert`

Short name: none

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

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

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

Wrap the folder names in the paths in single quotes (Linux or Mac OS) or double quotes (Windows) if they include spaces.

```
--cert 'path to certificate file'/certificate_file_name.pem
--key 'path to key file'/key_file_name.pem
```

`--key`

Short name: none

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

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

Wrap the folder names in the paths in single quotes (Linux or Mac OS) or double quotes (Windows) if they include spaces.

```
--cert 'path to certificate file'/certificate_file_name.pem
--key 'path to key file'/key_file_name.pem
```

`--tls-ca`

Short name: `--ca`

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

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

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

Do Not Restrict Access to the Docker API

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

`--no-tlsverify`

Short name: `--kv`

The `--no-tlsverify` option prevents the use of CAs for client authentication. You still require a server certificate if you use `--no-tlsverify`. You can still supply a custom server certificate by using the `--cert` and `--key` options. If you do not use `--cert` and `--key` to supply a custom server certificate, `vic-machine create` generates a self-signed server certificate. If you specify `--no-tlsverify` there is no access control and the VCH is susceptible to man-in-the-middle attacks when connected to Docker clients. However, connections remain encrypted.

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

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

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

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

```
--no-tlsverify
```

`--no-tls`

Short name: `-k`

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

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

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

```
--no-tls
```

Specify Different User Accounts for VCH Deployment and Operation

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

`--ops-user`

Short name: None

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

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

Wrap the user name in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if it includes special characters.

```
--ops-user 'user_n@me'
```

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

`--ops-password`

Short name: None

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

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

Wrap the password in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if it includes special characters.

```
--ops-password 'p@ssword'
```

Private Registry Options

If you use vSphere Integrated Containers Registry, or if container developers need to access Docker images that are stored in other private registry servers, you must configure VCHs to allow them to connect to these private registry servers when you deploy the VCHs. VCHs can connect to both secure and insecure private registry servers.

`--registry-ca`

Short name: `--rc`

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

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

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

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

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

`--insecure-registry`

Short name: `--dir`

An insecure private registry server is a private registry server for Docker images that does not provide TLS. The VCH cannot confirm the identity of the remote system that it is pulling images from and the communication is not encrypted. Setting the `--insecure-registry` option on a VCH informs that VCH that it is authorized to pull images from the designated insecure private registry server. Insecure private registries are not recommended in production environments.

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

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

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

Datastore Options

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

- vSphere Integrated Containers Engine fully supports VMware vSAN datastores.
- vSphere Integrated Containers Engine supports all alphanumeric characters, hyphens, and underscores in datastore paths and datastore names, but no other special characters.
- If you specify different datastores in the different datastore options, and if no single host in a cluster can access all of those datastores, `vic-machine create` fails with an error.

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

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

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

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

`--image-store`

Short name: `-i`

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

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

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

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

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

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

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

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

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

If the folder that you specify in `/path` does not already exist, `vic-machine create` creates it. Wrap the datastore name and path in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if they include spaces:

```
--image-store 'datastore name'/'datastore path'
```

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

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

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

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

`--volume-store`

Short name: `--vs`

The datastore in which to create volumes when container developers use the `docker volume create` or `docker create -v` commands. When you specify the `volume-store` option, you provide the name of the target datastore and a label for the volume store. You can optionally provide a path to a specific folder in the datastore in which to create the volume store. If the folders that you specify in the path do not already exist on the datastore, `vic-machine create` creates the appropriate folder structure.

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

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

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

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.

If you specify an invalid datastore name, `vic-machine create` fails and suggests valid datastores.

IMPORTANT If you do not specify the `volume-store` option, no volume store is created and container developers cannot use the `docker volume create` or `docker create -v` commands.

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

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

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

- If you specify the target datastore and the volume store label, `vic-machine create` creates a folder named `VIC/volumes` at the top level of the target datastore. Any volumes that container developers create will appear in the `VIC/volumes` folder.

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

- If you specify the target datastore, a datastore path, and the volume store label, `vic-machine create` creates a folder named `volumes` in the location that you specify in the datastore path. Any volumes that container developers create will appear in the `path/volumes` folder.

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

- Wrap the datastore name and path in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if they include spaces. The volume store label cannot include spaces.

  ```
  --volume-store 'datastore name'/'datastore path':volume_store_label
  ```

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

  ```
  --volume-store datastore_name/path:volume_store_label_1
  --volume-store datastore_name/path:volume_store_label_2
  [...]
  --volume-store datastore_name/path:volume_store_label_n
  ```

Networking Options

The `vic-machine create` utility allows you to specify different networks for the different types of traffic between containers, the VCH, the external internet, and your vSphere environment. For information about the different networks that VCHs use, see Networks Used by vSphere Integrated Containers Engine.

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

By default, `vic-machine create` obtains IP addresses for VCH endpoint VMs by using DHCP. For information about how to specify a static IP address for the VCH endpoint VM on the client, public, and management networks, see Specify a Static IP Address for the VCH Endpoint VM in Advanced Options.

If your network access is controlled by a proxy server, see Options to Configure VCHs to Use Proxy Servers in Advanced Options.

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.

`--bridge-network`

Short name: `-b`

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

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

In a vCenter Server environment, before you run `vic-machine create`, you must create a distributed virtual switch and a port group. You must add the target ESXi host or hosts to the distributed virtual switch, and assign a VLAN ID to the port group, to ensure that the bridge network is isolated. For information about how to create a distributed virtual switch and port group, see the section on vCenter Server Network Requirements in Environment Prerequisites for VCH Deployment.

You pass the name of the port group to the `bridge-network` option. Each VCH requires its own port group.

IMPORTANT

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

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

The `bridge-network` option is **optional** when you are deploying a VCH to an ESXi host with no 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
```

Wrap the port group name in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if it includes spaces.

```
--bridge-network 'port group name'
```

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

For information about how to specify a range of IP addresses for additional bridge networks, see `bridge-network-range` in Advanced Networking Options.

`--client-network`

Short name: `--cln`

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

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

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

Wrap the port group name in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if it includes spaces.

```
--client-network 'port group name'
```

`--public-network`

Short name: `--pn`

A port group for containers to use to connect to the Internet. VCHs use the public network to pull container images, for example from https://hub.docker.com/. Containers that use use port mapping expose network services on the public interface.

NOTE: vSphere Integrated Containers Engine 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.

If not specified, 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.

```
--public-network port_group
```

Wrap the network name in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if it includes spaces.

```
--public-network 'port group name'
```

`--management-network`

Short name: `--mn`

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

IMPORTANT: Because the management network provides access to your vSphere environment, and because container VMs use this network to communicate with the VCH, always use a secure network for the management network.

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

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

Wrap the network name in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if it includes spaces.

```
--management-network 'port group name'
```

`--container-network`

Short name: `--cn`

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

You can optionally specify one or more container networks. Container networks allow containers to directly attach to a network without having to route through the VCH via network address translation (NAT). Container networks that you add by using the `--container-network` option appear when you run the `docker network ls` command. These networks are available for use by containers. Containers that use these networks are directly attached to the container network, and do not go through the VCH or share the public IP of the VCH.

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

- You can specify a vSphere network as the container network.
- The port group must exist before you run `vic-machine create`.
- You cannot use the same port group as you use for the bridge network.
- You can create the port group on the same distributed virtual switch as the port group that you use for the bridge network.
- If the port group that you specify in the `container-network` option does not support DHCP, see Options for Configuring a Non-DHCP Network for Container Traffic in Advanced Options.
- The descriptive name appears under `Networks` when you run `docker info` or `docker network ls` on the deployed VCH.
- Container developers use the descriptive name in the `--net` option when they run `docker run` or `docker create`.

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

If you do not specify `--container-network`, or if you deploy containers that do not use a container network, the containers' network services are still be available via port mapping through the VCH, by using NAT through the public interface of the VCH.

```
--container-network port_group_name:container_port _group_name
```

Wrap the port group name in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if it includes spaces. The descriptive name cannot include spaces.

```
--container-network 'port group name':container port group name
```

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

General Deployment Options

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

`--name`

Short name: `-n`

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

```
--name vch_name
```

Wrap the name in single quotes (') on Mac OS and Linux and in double quotes (") on Windows if it includes spaces.

```
--name 'vch name'
```

`--memory`

Short name: `--mem`

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

```
--memory 1024
```

`--cpu`

Short name: None

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

```
--cpu 1024
```

`--force`

Short name: `-f`

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

If your vSphere environment uses untrusted, self-signed certificates, you can use the `--force` option to deploy a VCH without providing the thumbprint of the vCenter Server or ESXi host in the `thumbprint` option.

IMPORTANT Running `vic-machine create` with the `--force` option rather than providing the certificate thumbprint is not recommended, because it permits man-in-the-middle attacks to go undetected.

```
--force
```

`--timeout`

Short name: none

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

```
--timeout 5m0s
```

Advanced Options

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

Options for Specifying a Static IP Address for the VCH Endpoint VM

You can specify a static IP address for the VCH endpoint VM on each of the client, public, and management networks. DHCP is used for the endpoint VM for any network on which you do not specify a static IP address.

To specify a static IP address for the endpoint VM on the client, public, or management network, you provide an IP address in the `client/public/management-network-ip` option. If you set a static IP address, you can optionally provide gateway addresses and specify one or more DNS server addresses.

`--dns-server`

Short name: None

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

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

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

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

Short name: None

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

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

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

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

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

You specify addresses as IPv4 addresses with a network mask.

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

You can also specify addresses as resolvable FQDNs.

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

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

Short name: None

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

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

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

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

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

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

In the following example, `--management-network-gateway` informs the VCH that it can reach all of the vSphere management endoints that are in the ranges 192.168.3.0-255 and 192.168.128.0-192.168.131.255 by sending packets to the gateway at 192.168.2.1. Ensure that the address ranges that you specify include all of the systems that will connect to this VCH instance.

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

Options for Configuring a Non-DHCP Network for Container Traffic

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

For information about the container network, see the section on the `container-network` option.

`--container-network-gateway`

Short name: `--cng`

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

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

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

Wrap the port group name in single quotes (Linux or Mac OS) or double quotes (Windows) if it includes spaces.

```
--container-network-gateway 'port group name':gateway_ip_address/subnet_mask
```

`--container-network-dns`

Short name: `--cnd`

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

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

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

Wrap the port group name in single quotes (Linux or Mac OS) or double quotes (Windows) if it includes spaces.

```
--container-network-dns 'port group name':8.8.8.8
```

`--container-network-ip-range`

Short name: `--cnr`

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

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

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

You can also specify the IP range as a CIDR.

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

Wrap the port group name in single quotes (Linux or Mac OS) or double quotes (Windows) if it includes spaces.

```
--container-network-ip-range 'port group name':192.168.100.0/24
```

Options to 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 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.

`--https-proxy`

Short name: `--sproxy`

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

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

`--http-proxy`

Short name: `--hproxy`

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

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

Advanced Resource Management Options

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

`--memory-reservation`

Short name: `--memr`

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

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

`--memory-shares`

Short name: `--mems`

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

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

`--cpu-reservation`

Short name: `--cpur`

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

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

`--cpu-shares`

Short name: `--cpus`

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

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

`--endpoint-cpu`

Short name: none

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

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

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

`--endpoint-memory`

Short name: none

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

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

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

Other Advanced Options

`--bridge-network-range`

Short name: `--bnr`

The range of IP addresses that additional bridge networks can use when container application developers use `docker network create` to create new bridge networks. If you do not specify the `bridge-network-range` option, the IP range for bridge networks is 172.16.0.0/12.

When you specify the bridge network IP range, you specify the IP range as a CIDR. The smallest subnet that you can specify is /16. If you specify an invalid value for `--bridge-network-range`, `vic-machine create` fails with an error.

```
--bridge-network-range 192.168.100.0/16
```

`--base-image-size`

Short name: None

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

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

`--container-store`

Short name: `--cs`

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

`--appliance-iso`

Short name: `--ai`

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

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

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

Wrap the folder names in the path in single quotes (Linux or Mac OS) or double quotes (Windows) if they include spaces.

```
--appliance-iso 'path to ISO file'/appliance.iso
```

--bootstrap-iso

Short name: `--bi`

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

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

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

Wrap the folder names in the path in single quotes (Linux or Mac OS) or double quotes (Windows) if they include spaces.

```
--bootstrap-iso 'path to ISO file'/bootstrap.iso
```

--use-rp

Short name: none

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

```
--use-rp
```

--debug

Short name: `-v`

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

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

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

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

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

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

Advanced Examples of Deploying a VCH

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

For simplicity, these examples use the `--force` option to disable the verification of the vCenter Server certificate, so the `--thumbprint` option is not specified. Similarly, all examples that do not relate explicitly to certificate use specify the `--no-tls` option.

For detailed descriptions of all of the `vic-machine create` options, see VCH Deployment Options.

General Deployment Examples

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

Deploy to a vCenter Server Cluster with Multiple Datacenters and Datastores

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

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

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

This example deploys a VCH with the following configuration:

- Provides the vCenter Single Sign-On user and password in the `--target` option. Note that the user name is wrapped in quotes, because it contains the `@` character. Use single quotes if you are using `vic-machine` on a Linux or Mac OS system and double quotes on a Windows system.
- Deploys a VCH named `vch1` to the cluster `cluster1` in datacenter `dc1`.
- Uses a port group named `vic-bridge` for the bridge network.
- Designates `datastore1` as the datastore in which to store container images, the files for the VCH appliance, and container VMs.

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

```
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--name vch1
--force
--no-tls
```

Deploy to a Specific Standalone Host in vCenter Server

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

This example deploys a VCH with the following configuration:

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

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

Deploy to a Resource Pool on an ESXi Host

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

This example deploys a VCH with the following configuration:

- Specifies the user name and password, image store, and a name for the VCH.
- Designates `rp 1` as the resource pool in which to place the VCH. Note that the resource pool name is wrapped in quotes, because it contains a space. Use single quotes if you are using `vic-machine` on a Linux or Mac OS system and double quotes on a Windows system.

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

Deploy to a Resource Pool in a vCenter Server Cluster

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

This example deploys a VCH with the following configuration:

- Specifies the user name, password, datacenter, image store, bridge network, and name for the VCH.
- Designates `rp 1` in cluster `cluster 1` as the resource pool in which to place the VCH. Note that the resource pool and cluster names are wrapped in quotes, because they contain spaces. Use single quotes if you are using `vic-machine` on a

Linux or Mac OS system and double quotes on a Windows system.

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

Set Limits on Resource Use

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

This example deploys a VCH with the following configuration:

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

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

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

Networking Examples

The examples in this section demonstrate how to direct traffic to and from VCHs and the other elements in your environment, how to set static IPs, how to configure container VM networks, and how to configure a VCH to use a proxy server.

Specify Public, Management, and Client Networks

In addition to the mandatory bridge network, if your vCenter Server environment includes multiple networks, you can direct different types of traffic to different networks.

- You can direct the traffic between the VCH and the Internet to a specific network by specifying the `--public-network` option. Any container VM traffic that routes through the VCH also uses the public network. If you do not specify the `--public-network` option, the VCH uses the VM Network for public network traffic.
- You can direct traffic between ESXi hosts, vCenter Server, and the VCH to a specific network by specifying the `--management-network` option. If you do not specify the `--management-network` option, the VCH uses the public network for management traffic.
- You can designate a specific network for use by the Docker API by specifying the `--client-network` option. If you do not specify the `--client-network` option, the Docker API uses the public network.

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

This example deploys a VCH with the following configuration:

- Specifies the user name, password, datacenter, cluster, image store, bridge network, and name for the VCH.
- Directs public and management traffic to network 1 and Docker API traffic to network 2. Note that the network names are wrapped in quotes, because they contain spaces. Use single quotes if you are using `vic-machine` on a Linux or Mac OS system and double quotes on a Windows system.

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

For more information about the networking options, see the Networking Options section in VCH Deployment Options.

Set a Static IP Address for the VCH Endpoint VM on the Different Networks

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

This example deploys a VCH with the following configuration:

- Specifies the user name, password, datacenter, cluster, image store, bridge network, and name for the VCH.
- Directs public and management traffic to network 1 and Docker API traffic to network 2. Note that the network names are wrapped in quotes, because they contain spaces. Use single quotes if you are using `vic-machine` on a Linux or Mac OS system and double quotes on a Windows system.
- Sets a DNS server for use by the public, management, and client networks.
- Sets a static IP address and subnet mask for the VCH endpoint VM on the public and client networks. Because the management network shares a network with the public network, you only need to specify the public network IP address. You cannot specify a management IP address because you are sharing a port group between the management and public network.
- Specifies the gateway for the public network. If you set a static IP address on the public network, you must also specify the gateway address.
- Does not specify a gateway for the client network. It is not necessary to specify a gateway on either of the client or management networks if those networks are L2 adjacent to their gateways.
- Because this example specifies a static IP address for the VCH endpoint VM on the client network, `vic-machine create` uses this address as the Common Name with which to create auto-generated trusted certificates. Full TLS authentication is implemented by default, so no TLS options are specified.

```
vic-machine-operating_system create
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--public-network 'network 1'
--public-network-ip 192.168.1.10/24
--public-network-gateway 192.168.1.1
```

```
--management-network 'network 1'
--client-network 'network 2'
--client-network-ip 192.168.3.10/24
--dns-server dns_server_address
--force
--name vch1
```

For more information about setting static IP addresses, see the Options for Specifying a Static IP Address for the VCH Endpoint VM in VCH Deployment Options.

Configure a Non-DHCP Network for Container VMs

You can designate a specific network for container VMs to use by specifying the `--container-network` option. Containers use this network if the container developer runs `docker run` or `docker create` specifying the `--net` option with one of the specified container networks when they run or create a container. This option requires a port group that must exist before you run `vic-machine create`. You cannot use the same port group that you use for the bridge network. You can provide a descriptive name for the network, for use by Docker. If you do not specify a descriptive name, Docker uses the vSphere network name. For example, the descriptive name appears as an available network in the output of `docker info` and `docker network ls`.

If the network that you designate as the container network in the `--container-network` option does not support DHCP, you can configure the gateway, DNS server, and a range of IP addresses for container VMs to use.

This example deploys a VCH with the following configuration:

- Specifies the user name, password, datacenter, cluster, image store, bridge network, and name for the VCH.
- Uses the VM Network for the public, management, and client networks.
- Designates a port group named `vic-containers` for use by container VMs that are run with the `--net` option.
- Gives the container network the name `vic-container-network`, for use by Docker.
- Specifies the gateway, two DNS servers, and a range of IP addresses on the container network for container VMs to use.

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

For more information about the container network options, see the `--container-network` and Options for Configuring a Non-DHCP Network for Container Traffic sections in VCH Deployment Options.

Configure a Proxy Server

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

This example deploys a VCH with the following configuration:

- Specifies the user name, password, image store, cluster, bridge network, and name for the VCH.
- Configures the VCH to access the network via an HTTPS proxy server.

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

```
--target 'Administrator@vsphere.local':password@vcenter_server_address/dc1
--compute-resource cluster1
--image-store datastore1
--bridge-network vch1-bridge
--https-proxy https://proxy_server_address:port
--name vch1
--force
--no-tls
```

Specify Volume Stores

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

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

This example deploys a VCH with the following configuration:

- Specifies the user name, password, datacenter, cluster, bridge network, and name for the VCH.
- Specifies the `volumes` folder on `datastore 1` as the default volume store. Creating a volume store named `default` allows container application developers to create anonymous or named volumes by using `docker create -v`.
- Specifies a second volume store named `volume_store_2` in the `volumes` folder on `datastore 2`.
- Note that the datastore names are wrapped in quotes, because they contain spaces. Use single quotes if you are using `vic-machine` on a Linux or Mac OS system and double quotes on a Windows system.

```
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
--name vch1
--force
--no-tls
```

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

Security Examples

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

Use Auto-Generated Trusted CA Certificates

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

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

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

This example deploys a VCH with the following configuration:

- Specifies the user, password, datacenter, image store, cluster, bridge network, and name for the VCH.
- Provides a wildcard domain `*.example.org` as the FQDN for the VCH, for use as the Common Name in the certificate. This assumes that there is a DHCP server offering IP addresses on VM Network, and that those addresses have corresponding DNS entries such as `dhcp-a-b-c.example.com`.
- Specifies a folder in which to store the auto-generated certificates.

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

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

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

Use Custom Server Certificates

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

This example deploys a VCH with the following configuration:

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

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

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

Combine Custom Server Certificates and Auto-Generated Client Certificates

You can create a VCH so that it uses a custom certificate to authenticate with the VCH Admin portal and an auto-generated certificate to authenticate with Docker clients. Specifying the paths to the `key.pem` and `cert.pem` files in the `--key` and `--cert` options is sufficient if the certificate is signed by a CA that is trusted by the system on which you run `vic-machine` and that is also trusted by the users who connect to the VCH. However, if the certificate is signed by a CA that is not trusted by the system, the installer cannot validate the certificate chain and `vic-machine create` fails with a certificate validation warning:

```
Server certificate hostname doesn't match:
x509: cannot validate certificate for 10.17.109.182 because it doesn't contain any IP SANs
```

In this case, if you specify the `--tls-cname` option to match the common name value of the server certificate, `vic-machine create` generates self-signed certificates for Docker client authentication and deployment of the VCH succeeds.

This example deploys a VCH with the following configuration:

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

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

Specify Different User Accounts for VCH Deployment and Operation

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

This example deploys a VCH with the following configuration:

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

```
vic-machine-operating_system create
--target vsphere_admin:vsphere_admin_password@vcenter_server_address/dc1
--compute-resource cluster1/VCH_pool
--image-store datastore1
--bridge-network vch1-bridge
--container-network vic-containers:vic-container-network
--name vch1
--ops-user vsphere_user
--ops-password vsphere_user_password
--force
--no-tls
```

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

Registry Server Examples

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

Authorize Access to an Insecure Private Registry Server

To authorize connections from a VCH to an insecure private registry server, set the `insecure-registry` option. You can specify `insecure-registry` multiple times to allow connections from the VCH to multiple insecure private registry servers.

This example deploys a VCH with the following configuration:

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

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

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

Authorize Access to Secure Registries and vSphere Integrated Containers Registry

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

Deploy a VCH for Use with vSphere Integrated Containers Registry

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

If you did not provide a custom server certificate and private key for the registry to the OVA installer when you deployed the vSphere Integrated Containers appliance, vSphere Integrated Containers Registry auto-generates a Certificate Authority (CA) certificate, a server certificate, and a server private key. You can download the auto-generated CA certificates from the vSphere Integrated Containers Registry interface.

Prerequisites

- You selected the option to deploy vSphere Integrated Containers Registry when you deployed the vSphere Integrated Containers appliance.
- You downloaded the vSphere Integrated Containers Engine bundle from the appliance.

Procedure

1. Obtain the CA certificate of the registry instance or instances to use with this VCH.
   - If you deployed the registry with custom certificates, obtain the certificate from your certificate manager.
   - If you deployed the registry with auto-generated certificates, log in to the vSphere Integrated Containers Registry interface as `admin` user, click the **admin** drop-down menu and click **Download Root Cert**.
   - You can also obtain the certificate by using SCP to copy the certificate file from `/data/harbor/cert` in the vSphere Integrated Containers appliance VM.

     ```
     scp root@vic_appliance_address:/data/harbor/cert/ca.crt ./destination_path
     ```

2. Use `vic-machine create` to deploy a VCH, specifying the registry's CA certificate by using the `--registry-ca` option.

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

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

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

Result

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

Use Different User Accounts for VCH Deployment and Operation

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

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

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

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

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

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

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
`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 vApp > Add virtual machine VirtualMachine > Configuration > Add new disk VirtualMachine > Configuration > Advanced VirtualMachine > Configuration > Add or Remove Device VirtualMachine > Configuration > Remove disk 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

For information about how to create vSphere roles, see vSphere Permissions and User Management Tasks in the vSphere documentation.

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

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

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

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

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

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

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

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.

What to do next

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

Using `vic-machine` to Manage VCHs

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.

Prerequisites

You have downloaded and unpacked the vSphere Integrated Containers Engine binaries.

Procedure

1. On the system on which you downloaded the binaries, navigate to the directory that contains the `vic-machine` utility.
2. Run the `vic-machine version` command.

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

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

Result

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.

`--target`

Short name: `-t`

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

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

  ```
  --target esxi_host_address
  ```

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

  ```
  --target vcenter_server_address
  ```

- You can include the user name and password in the target URL.

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

  Wrap the user name or password in single quotes (Linux or Mac OS) or double quotes (Windows) if they include special characters.

  ```
  'vcenter_or_esxi_usern@me':'p@ssword'@vcenter_or_esxi_address
  ```

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

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

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

`--user`

Short name: `-u`

The username for the ESXi host or vCenter Server instance on which you deployed the VCH. This option is mandatory if you do not specify the username in the `target` option.

```
--user esxi_or_vcenter_server_username
```

Wrap the user name in single quotes (Linux or Mac OS) or double quotes (Windows) if it includes special characters.

```
--user 'esxi_or_vcenter_server_usern@me'
```

`--password`

Short name: `-p`

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

```
--password esxi_host_or_vcenter_server_password
```

Wrap the password in single quotation marks (') on Mac OS and Linux and in double quotation (") marks on Windows if it includes special characters.

```
--password 'esxi_host_or_vcenter_server_p@ssword'
```

`--thumbprint`

Short name: None

The thumbprint of the vCenter Server or ESXi host certificate. Specify this option if your vSphere environment uses untrusted, self-signed certificates. Alternatively, specifying the `--force` option allows you to omit the `--thumbprint` option. 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.

To obtain the thumbprint of the vCenter Server or ESXi host certificate, run `vic-machine` without the specifying the `--thumbprint` or `--force` options. The operation fails, but the resulting error message includes the required certificate thumbprint. You can copy the thumbprint from the error message and run `vic-machine` again, including the `thumbprint` option. If you obtain the thumbprint by other means, use upper-case letters and colon delimitation rather than space delimitation when you specify `--thumbprint`.

```
--thumbprint certificate_thumbprint
```

`--compute-resource`

Short name: `-r`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- Wrap the resource names in single quotes (Linux or Mac OS) or double quotes (Windows) if they include spaces:

  ```
  --compute-resource 'cluster name'/'resource pool name'
  ```

`--name`

Short name: `-n`

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

```
--name vch_appliance_name
```

Wrap the appliance name in single quotes (Linux or Mac OS) or double quotes (Windows) if it includes spaces.

```
--name 'vch appliance name'
```

`--id`

Short name: None

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

```
--id vch_id
```

`--timeout`

Short name: none

The timeout period for performing operations on the VCH. Specify a value in the format `XmYs` if the default timeout is insufficient.

```
--timeout 5m0s
```

List VCHs 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. The `vic-machine ls` command does not include any options in addition to the common options described in Common `vic-machine` Options.

Prerequisites

You have deployed a VCH. If you have not deployed a VCH, `vic-machine ls` returns an empty list.

Procedure

1. On the system on which you run `vic-machine`, navigate to the directory that contains the `vic-machine` utility.
2. Run the `vic-machine ls` command.
    - To obtain a list of all VCHs that are running on an ESXi host or vCenter Server instance, you must provide the address of the target ESXi host or vCenter Server.
    - You must specify the username and optionally the password, either in the `--target` option or separately in the `--user` and `--password` options.
    - If your vSphere environment uses untrusted, self-signed certificates, you must also specify the thumbprint of the vCenter Server instance or ESXi host in the `--thumbprint` option. To obtain the thumbprint of the vCenter Server or ESXi host certificate, run `vic-machine` without the specifying the `--thumbprint` option. The listing of the VCHs fails, but the resulting error message includes the required certificate thumbprint. You can copy the thumbprint from the error message and run `vic-machine` again, including the `--thumbprint` option.

      ```
      $ vic-machine-operating_system ls
      --target esxi_host_address
      --user root
      --password esxi_host_password
      --thumbprint certificate_thumbprint
      ```

      ```
      $ vic-machine-operating_system ls
      --target vcenter_server_username:password@vcenter_server_address
      --thumbprint certificate_thumbprint
      ```

Result

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 Information About a VCH

You can obtain information about a virtual container host (VCH) by using the `vic-machine inspect` command.

The `vic-machine inspect` command does not include any options in addition to the common options described in Common `vic-machine` Options.

Prerequisites

You have deployed a VCH.

Procedure

1. On the system on which you run `vic-machine`, navigate to the directory that contains the `vic-machine` utility.
2. Run the `vic-machine inspect` command.

   The following example includes the options required to obtain information about a named instance of a VCH from a simple vCenter Server environment.

   - You must specify the username and optionally the password, either in the `--target` option or separately in the `--user` and `--password` options.
   - If the VCH has a name other than the default name, `virtual-container-host`, you must specify the `--name` or `--id` option.
   - If multiple compute resources exist in the datacenter, you must specify the `--compute-resource` or `--id` option.
   - If your vSphere environment uses untrusted, self-signed certificates, you must also specify the thumbprint of the vCenter Server instance or ESXi host in the `--thumbprint` option. To obtain the thumbprint of the vCenter Server or ESXi host certificate, run `vic-machine` without the specifying the `--thumbprint` option. The inspection of the VCH fails, but the resulting error message includes the required certificate thumbprint. You can copy the thumbprint from the error message and run `vic-machine` again, including the `--thumbprint` option.

     ```
     $ vic-machine-operating_system inspect
     --target vcenter_server_username:password@vcenter_server_address
     --thumbprint certificate_thumbprint
     --name vch_name
     ```

Result

The `vic-machine inspect` command displays information about the VCH:

- The VCH ID:

  ```
  VCH ID: VirtualMachine:vm-101
  ```

  The vSphere Managed Object Reference, or moref, of the VCH. You can use VCH ID when you run the `vic-machine delete` 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.
  - VCH with full TLS authentication with trusted Certificate Authority certificates:

    ```
    DOCKER_TLS_VERIFY=1
    DOCKER_CERT_PATH=path_to_certificates
    DOCKER_HOST=vch_address:2376
    ```

  - VCH with TLS authentication with untrusted self-signed certificates:

    ```
    DOCKER_HOST=vch_address:2376
    ```

  - VCH with no TLS authentication:

    ```
    DOCKER_HOST=vch_address:2375
    ```

- The Docker command to use to connect to the Docker endpoint.
  - VCH with full TLS authentication with trusted Certificate Authority certificates:

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

  - VCH with TLS authentication with untrusted self-signed certificates:

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

  - VCH with no TLS authentication:

    ```
    docker -H vch_address:2375 info
    ```

Delete a VCH

You delete virtual container hosts (VCHs) by using the `vic-machine delete` command.

For descriptions of the options that `vic-machine delete` includes in addition to the Common `vic-machine` Options , see VCH Delete Options.

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.

Prerequisites

You have deployed a VCH that you no longer require.

Procedure

1. On the system on which you run `vic-machine` , navigate to the directory that contains the `vic-machine` utility.
2. Run the `vic-machine delete` command.

   The following example includes the options required to remove a VCH from a simple vCenter Server environment.

   - You must specify the username and optionally the password, either in the `--target` option or separately in the `--user` and `--password` options.
   - If the VCH has a name other than the default name, `virtual-container-host` , you must specify the `--name` or `--id` option.
   - If multiple compute resources exist in the datacenter, you must specify the `--compute-resource` or `--id` option.
   - If your vSphere environment uses untrusted, self-signed certificates, you must also specify the thumbprint of the vCenter Server instance or ESXi host in the `--thumbprint` option. To obtain the thumbprint of the vCenter Server or ESXi host certificate, run `vic-machine` without the specifying the `--thumbprint` or `--force` options. The deletion of the VCH fails, but the resulting error message includes the required certificate thumbprint. You can copy the thumbprint from the error message and run `vic-machine` again, including the `--thumbprint` option.

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

3. 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` . 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.

   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
   ```

What to Do Next

The `vic-machine delete` command does not modify the firewall. If you do not need to deploy or run further VCHs on the ESXi host or cluster, run `vic-machine update firewall --deny` to close port 2377 on the host or hosts.

VCH Delete Options

The command line utility for vSphere Integrated Containers Engine, `vic-machine`, provides a `delete` command that allows you to cleanly remove virtual container hosts (VCHs).

The `vic-machine delete` command includes one option in addition to the common options described in Common `vic-machine` Options.

`--force`

Short name: `-f`

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 can use the `--force` option to delete a VCH without providing the thumbprint of the vCenter Server or ESXi host in the `--thumbprint` option.

```
--force
```

Access the 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 can only log in to VCH Admin by specifying the username and password of the ESXi host or vCenter Server on which you deployed the VCH. If you deployed the VCH with client and server authentication by using `--tls-cname` or by specifying a static IP address on the client network, you can use the generated `*.pfx` certificate to authenticate with the VCH Admin portal. For information about using the `*.pfx` certificate to log into VCH admin, see Browser-Based Certificate Login and Command Line Certificate Login.

Prerequisites

- You deployed a VCH.
- Obtain the address of the VCH:
  - Copy the address from the output of `vic-machine create` or `vic-machine inspect`.
  - If you deployed the VCH to vCenter Server, copy the address from the **Summary** tab for the endpoint VM in the vSphere Client.
  - If you deployed the VCH to an ESXi host, copy the address from the **Summary** tab for the endpoint VM in the desktop vSphere Client.

Procedure

1. Go to https://*vch_address*:2378.

   If prompted about an insecure or not private connection, click Advanced and follow the prompts to proceed to the portal.

2. Enter the username and password for the vCenter Server instance or ESXi host.

Result

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.

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 --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 specifying custom certificates in the `--key` and `--cert` options, by `--tls-cname` to auto-generate CA certificates, 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 --cert-path` option.

Without Client Authentication

If you deployed the VCH without client authentication by using either of `--no-tls` or `--no-tlsverify`, you can use curl to access the logs but you must first authenticate connections to VCH Admin by using the vSphere username and password.

1. Log in to VCH Admin to gather an authentication cookie for subsequent access:

   ```
   curl -sk https://vch_address:2378/authentication
   -XPOST -F username=vsphere_username
   -F password=vsphere_password
   -D cookies_file
   ```

2. Use the cookie from Step 1 in a `curl` command to access the logs.

   ```
   curl -sk https://vch_address:2378/logs/port-layer.log
   -b cookies_file
   ```

VCH Admin Status Reference

The Web-based administration portal for virtual container hosts (VCHs), VCH Admin, presents status information about a VCH.

If the vSphere environment in which you are deploying a VCH does not meet the requirements, the deployment does not succeed. However, a successfully deployed VCH can stop functioning if the vSphere environment changes after the deployment. If environment changes adversely affect the VCH, the status of the affected component changes from green to yellow.

Virtual Container Host (VCH)

VCH Admin checks the status of the processes that the VCH runs:

- The port layer server, that presents an API of low-level container primitive operations, and implements those container operations via the vSphere APIs.
- VCH Admin server, that runs the VCH Admin portal.
- The vSphere Integrated Containers Engine initialization service and watchdog service for the other components.
- The Docker engine server, that exposes the Docker API and semantics, translating those composite operations into port layer primitives.

Error

- The **VCH** status is yellow.
- The **VCH** status is yellow and an error message informs you that the VCH cannot connect to vSphere.

Cause

- One or more of the VCH processes is not running correctly, or the VCH is unable to connect to vSphere.
- The management network connection is down and the VCH endpoint VM cannot connect to vSphere.

Solution

1. (Optional) If you see the error that the VCH is unable to connect to vSphere, check the VCH management network.
2. In the VCH Admin portal for the VCH, click the link for the **VCH Admin Server** log.
3. Search the log for references to the different VCH processes.

   The different processes are identified in the log by the following names:

   - `port-layer-server`
   - `vicadmin`
   - `vic-init`
   - `docker-engine-server`
4. Identify the process or processes that are not running correctly and attempt to remediate the issues as required.

Registry and Internet Connectivity

VCH Admin checks connectivity on the public network by attempting to connect from the VCH to docker.io and google.com. VCH Admin only checks the public network connection. It does not check other networks, for example the bridge, management, client, or container networks.

Error

The **Registry and Internet Connectivity** status is yellow.

Cause

The public network connection is down.

Solution

Check the **VCH Admin Server** log for references to network issues. Use the vSphere Web Client to remediate the management network issues as required.

Firewall

VCH Admin checks that the firewall is correctly configured on an ESXi host on which the VCH is running. If the VCH is running in a cluster, VCH Admin checks the firewall configuration on all of the hosts in the cluster.

Error

- The **Firewall** status is unavailable.
- The **Firewall** status is yellow and shows the error `Firewall must permit 2377/tcp outbound to use VIC`.

Cause

- The management network connection is down and the VCH endpoint VM cannot connect to vSphere.
- The firewall on the ESXi host on which the VCH is running no longer allows outbound connections on port 2377.
  - The firewall was switched off when the VCH was deployed. The firewall has been switched on since the deployment of the VCH.
  - A firewall ruleset was applied manually to the ESXi host to allow outbound connections on port 2377. The ESXi host has been rebooted since the deployment of the VCH. Firewall rulesets are not retained when an ESXi host reboots.

Solution

- If the **Firewall** status is unavailable:
  - Check the **VCH Admin Server** log for references to network issues.
  - Use the vSphere Web Client to remediate the management network issues as required.
- If you see the error about port 2377, run the `vic-machine update firewall` command on the ESXi host or hosts to allow outbound connections on port 2377. For information about how to run `vic-machine update firewall`, see Open the Required Ports on ESXi Hosts.

License

VCH Admin checks that the ESXi hosts on which you deploy VCHs have the appropriate licenses.

Error

- The **License** status is yellow and shows the error `License does not meet minimum requirements to use VIC`.
- The **License** status is unavailable.

Cause

- The license for the ESXi host or for one or more of the hosts in a vCenter Server cluster on which the VCH is deployed has been removed, downgraded, or has expired since the deployment of the VCH.
- The management network is down, or the VCH endpoint VM is unable to connect to vSphere.

Solution

- If the license does not meet the requirements:
  - If the VCH is running on an ESXi host that is not managed by vCenter Server, replace the ESXi host license with a valid vSphere Enterprise license.

    - If the VCH is running on a standalone ESXi host in vCenter Server, replace the ESXi host license with a valid vSphere Enterprise Plus license.
    - If the VCH is running in a vCenter Server cluster, check that all of the hosts in the cluster have a valid vSphere Enterprise Plus license, and replace any licenses that have been removed, downgraded, or have expired.
- If the **License** status is unavailable:
    - Check the **VCH Admin Server** log for references to network issues.
    - Use the vSphere Web Client to remediate the management network issues as required.

Access vSphere Integrated Containers Engine Log Bundles

vSphere Integrated Containers Engine provides log bundles that you can download from the VCH Admin portal for a virtual container host (VCH).

You access the VCH Admin Portal at https://*vch_address*:2378. For more information about the VCH Admin portal, see Access the 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.

Debugging the VCH

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.

NOTE: Do not confuse the `vic-machine debug` command with the `vic-machine create --debug` option. The `vic-machine debug` command allows you to log into and debug a VCH endpoint VM that you have already deployed. The `vic-machine create --debug` option deploys a new VCH that has increased levels of logging and other modifications, to allow you to debug the environment in which you deploy VCHs. For information about the `vic-machine create --debug` option, see the section on `--debug` in VCH Deployment Options.

- Enable Shell Access to the VCH Endpoint VM
- Authorize SSH Access to the VCH Endpoint VM
- VCH Debug Options

Enable shell access to the VCH Endpoint VM

You can use the `vic-machine debug` command to enable shell access to a virtual container host (VCH) endpoint VM by setting a root password on the VM. Setting a root password enables access to the VCH endpoint VM via the VM console only. If you require SSH access to the VCH endoint VM, rather than just shell access, see Authorize SSH Access to the VCH Endpoint VM.

IMPORTANT: Any changes that you make to a VCH by using `vic-machine debug` are non-persistent and are discarded if the VCH endpoint VM reboots.

For descriptions of the options that `vic-machine debug` includes in addition to the Common `vic-machine` Options, and for information about password expiry periods, see VCH Debug Options.

Prerequisites

You deployed a VCH.

Procedure

1. On the system on which you run `vic-machine`, navigate to the directory that contains the `vic-machine` utility.
2. Run the `vic-machine debug` command.
    - 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
      ```

    - Specify the ID or name of the VCH to debug.
    - Potentially provide the thumbprint of the vCenter Server or ESXi host certificate.
    - Provide a password for the root user on the VCH endpoint VM by specifying the `--rootpw` option. Wrap the password in single quotes (Linux or Mac OS) or double quotes (Windows) if it includes shell characters such as `$`, `!` or `%`.

      ```
      $ 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'
      ```

Result

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. 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.

For descriptions of the options that `vic-machine debug` includes in addition to the Common `vic-machine` Options, and for information about password expiry periods, see VCH Debug Options.

Prerequisites

You deployed a VCH.

Procedure

1. On the system on which you run `vic-machine`, navigate to the directory that contains the `vic-machine` utility.
2. Run the `vic-machine debug` command.
    - 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
      ```

      - Specify the ID or name of the VCH to debug.
      - Potentially provide the thumbprint of the vCenter Server or ESXi host certificate.
      - Specify the `--rootpw` option. Wrap the password in single quotes (Linux or Mac OS) or double quotes (Windows) if it includes shell characters such as `$`, `!` or `%`.
      - Authorize SSH access by specifying `--enable-ssh`.
      - Optionally, 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.

      ```
      $ 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
      ```

Result

The output of the `vic-machine debug` command includes confirmation that SSH access is enabled:

```
### Configuring VCH for debug ####
[...]
SSH to appliance:
ssh root@vch_address
[...]
Completed successfully
```

VCH Debug Options

The command line utility for vSphere Integrated Containers Engine, `vic-machine`, provides a `debug` command that allows you to enable VM console or SSH access to the virtual container host (VCH) endpoint VM, set a password for the root user account, and upload a key file for automatic public key authentication.

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.

IMPORTANT: If you set a password or enable shell access on a VCH endpoint VM, these changes do not persist if you reboot the VM. You must run `vic-machine debug` to reenable access and reset the password each time that the VCH endpoint VM reboots.

The `vic-machine debug` command includes the following options in addition to the common options described in Common `vic-machine` Options.

`--rootpw`

Short name: `--pw`

Set a new password for the root user account on the VCH endpoint VM. 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.

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 you attempt to log in after the password has expired, you see a prompt to change the password, in which case you must run `vic-machine debug --rootpw` again.

Wrap the password in single quotes (Linux or Mac OS) or double quotes (Windows) if it includes shell characters such as `$`, `!` or `%`.

```
--rootpw 'new_p@ssword'
```

`--enable-ssh`

Short name: `--ssh`

Enable an SSH server in the VCH endpoint VM. The `sshd` service runs until the VCH endpoint VM reboots. The `--enable-ssh` 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.

```
--enable-ssh
```

`--authorized-key`

Short name: `--key`

Upload a public key file to `/root/.ssh/authorized_keys` to enable SSH key authentication for the `root` user. Include the name of the `*.pub` file in the path.

```
--authorized-key path_to_public_key_file/key_file.pub
```

Upgrading vSphere Integrated Containers

If you have an existing deployment of vSphere Integrated Containers 1.0, you can upgrade its components to version 1.1.

- You can upgrade vSphere Integrated Containers Registry from version 0.5 to version 1.1.
- To upgrade vSphere Integrated Containers Engine from 0.8 or later to 1.1, upgrade your virtual container hosts (VCHs) individually.
- There is no upgrade for the Flex-based vSphere Web Client plug-in for vSphere Integrated Containers 1.0. Use the plug-in for the HTML5 vSphere Client, which is new in 1.1.

vSphere Integrated Containers 1.0 did not officially support vSphere Integrated Containers Management Portal. You cannot upgrade an existing instance of vSphere Integrated Containers Management Portal to version 1.1.

Upgrade vSphere Integrated Containers Registry

If you deployed version 0.5 of vSphere Integrated Containers Registry (Harbor) with vSphere Integrated Containers 1.0, you can upgrade your existing installation to version 1.1.

Prerequisites

- You have a vSphere Integrated Containers Registry 0.5 installation that you deployed by using the official OVA installer from vSphere Integrated Containers 1.0.
- Deploy the new vSphere Integrated Containers 1.1 appliance by using the OVA installer. For information about using the OVA installer, see Deploy the vSphere Integrated Containers Appliance.
  - Deploy the 1.1 appliance in a location in which it can access the VMDK files of the 0.5 appliance.
  - In the **Appliance Security** section of the **Customize template** page of the installer, do not disable SSH access to the 1.1 appliance.
  - In the **Registry Configuration** section of the installer, make sure that you provide the same passwords in the **Registry Admin Password** and **Database Password** as the 0.5 appliance uses.
- Log in to a vSphere Web Client instance from which you can access both of the vSphere Integrated Containers Registry 0.5 and 1.1 vSphere Integrated Containers appliances.
- 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 Registry 0.5 appliance by selecting **Shut Down Guest OS**.

   IMPORTANT: Do not select **Power Off**.

2. Right-click the new vSphere Integrated Containers 1.1 appliance, and select **Edit Settings**.
3. Click the **New device** drop-down menu, select **Existing Hard Disk**, and click **Add**.
4. Navigate to the VDMK files of the 0.5 appliance, select one of the VMDK files, and click **OK**.

   You can select either VMDK file. The order of selection is not important.

5. Expand the entry for **New Hark disk** and make sure that the new disk is attached as **Virtual Device Node** `SCSI(0:2)`.
6. Click **Existing Hard Disk** > **Add** again, select the other VMDK file, and click **OK**.
7. Make sure that the new disk is attached as **Virtual Device Node** `SCSI(0:3)` and click **OK** to close the Edit Settings window.
8. Power on the vSphere Integrated Containers 1.1 appliance, then use SSH to connect to it as root user.

   ```
   $ ssh root@vic_appliance_address
   ```

9. Run the upgrade script and respond to the prompts until you see confirmation that the upgrade is complete.

   ```
   $ /etc/vmware/harbor/upgrade_from_0.5.sh
   ```

10. Shut down the vSphere Integrated Containers 1.1 appliance, and edit its settings to detach the two VDMK files that you attached above.
11. Power on the vSphere Integrated Containers 1.1 appliance to complete the upgrade.

What to Do Next

Log into the new version of vSphere Integrated Containers Registry at https://*vic_1.1_appliance_address*:443 to verify that the data from your vSphere Integrated Containers Registry 0.5 installation has migrated successfully.

Upgrade a VCH

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 from version 0.8 and above. 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 the `vic-machine create` command.
- 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.

Procedure

1. On the system on which you run `vic-machine`, navigate to the directory that contains the new version of the `vic-machine` utility.
2. Run the `vic-machine upgrade` command.

   The following example includes the options required to upgrade a VCH in a simple vCenter Server environment.

   - You must specify the username and optionally the password, either in the `target` option or separately in the `--user` and `--password` options.
   - If the VCH has a name other than the default name, `virtual-container-host`, you must specify the `--name` or `--id` option.
   - If multiple compute resources exist in the datacenter, you must specify the `--compute-resource` or `--id` option.
   - If your vSphere environment uses untrusted, self-signed certificates, you must also specify the thumbprint of the vCenter Server instance or ESXi host in the `--thumbprint` option. To obtain the thumbprint of the vCenter Server or ESXi host certificate, run `vic-machine` without the specifying the `--thumbprint` or `--force` options. The upgrade of the VCH fails, but the resulting error message includes the required certificate thumbprint. You can copy the thumbprint from the error message and run `vic-machine` again, including the `--thumbprint` option.

     ```
     $ vic-machine-operating_system upgrade
     --target vcenter_server_username:password@vcenter_server_address
     --thumbprint certificate_thumbprint
     --name vch_name
     ```

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
   ```

```
--name vch_name
--timeout 5m0s
```

4. If the upgrade operation continues to fail with error messages, run `vic-machine upgrade` again with the `--force` option.

   If your vSphere environment uses untrusted, self-signed certificates, running `vic-machine upgrade` with the `--force` option allows you to omit the `--thumbprint` option.

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

What to Do Next

Upgrade the HTML5 vSphere Client Plug-In.

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.

`--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
```

Wrap the folder names in the path in single quotes (Linux or Mac OS) or double quotes (Windows) if they include spaces.

```
--appliance-iso 'path to ISO file'/appliance.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
```

Wrap the folder names in the path in single quotes (Linux or Mac OS) or double quotes (Windows) if they include spaces.

```
--bootstrap-iso 'path to ISO file'/ISO_file_name.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.

If your vSphere environment uses untrusted, self-signed certificates, you can use the `--force` option to upgrade a VCH without providing the thumbprint of the vCenter Server or ESXi host in the `thumbprint` option.

```
--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 HTML5 vSphere Client Plug-In

If you have a previous installation of the HTML5 vSphere Client plug-in for vSphere Integrated Containers and you download a new version of vSphere Integrated Containers Engine, you must upgrade the HTML5 plug-in.

NOTE: No new development work is planned for the plug-in for the Flex-based vSphere Web Client. In this and future releases, only the HTML5 vSphere Client will be updated. This release adds no new features to the Flex plug-in. If you installed the Flex plug-in with vSphere Integrated Containers 1.0, there is no upgrade to perform.

The plug-in for the HTML5 vSphere Client is new in vSphere Integrated Containers 1.1. This procedure describes how to upgrade the HTML5 plug-in from version 1.1 to a later 1.x release. For information about installing the HTML5 plug-in for the first time, see Install the HTML5 Plug-In on a vCenter Server Appliance or Install the HTML5 Plug-In on vCenter Server for Windows by Using a Web Server.

Prerequisites

- You deployed an older version of the vSphere Integrated Containers plug-in for the HTML5 vSphere Client.
- You downloaded a new version of vSphere Integrated Containers Engine.
- For information about how to update the `configs` and `install.sh` files, see Install the HTML5 Plug-In on a vCenter Server Appliance or Install the HTML5 Plug-In on vCenter Server for Windows by Using a Web Server.

Procedure

1. If you run vCenter Server on Windows, copy the new version of the `com.vmware.vic-version.zip` file to the appropriate location on your Web server.
2. Update the new version of the `configs` file.
    - vCenter Server Appliance: `vic/ui/HTML5Client/configs`
    - vCenter Server on Windows: `vic/ui/vCenterForWindows/configs`
3. (Optional) If you are upgrading the plug-in on a vCenter Server Appliance and you are working on a Windows system, update the `vic/ui/HTML5Client/install.sh` file to point `PLUGIN_MANAGER_BIN` to the Windows UI executable.
4. Run the `vic/ui/HTML5Client/upgrade.sh` or `vic/ui/vCenterForWindows/upgrade.bat` script.
5. When the upgrade finishes, if you are logged into the vSphere Client, log out then log back in again.

Troubleshooting vSphere Integrated Containers Engine Administration

This information provides solutions for common problems that you might encounter when working with vSphere Integrated Containers.

Check the Status of the vSphere Integrated Containers Services

You can check the status of the vSphere Integrated Containers Registry and vSphere Integrated Containers Management Portal services, and the file server that runs in the appliance, by logging in to the vSphere Integrated Containers appliance.

Prerequisites

You deployed the vSphere Integrated Containers appliance

Procedure

1. Connect to the vSphere Integrated Containers appliance by using SSH.
2. Run one of the following commands to check the status of one of the vSphere Integrated Containers services:
    - vSphere Integrated Containers Registry: `systemctl status harbor`
    - vSphere Integrated Containers Management Portal services: `systemctl status admiral`
    - Embedded file server: `systemctl status fileserver`

Result

The output shows the status of the service that you specified, as well as the most recent log entries.

Status	Description
`active (running)`	The service is running correctly.
`inactive (failed)`	The service failed to start.
`inactive (dead)`	The service is not responding.

What to Do Next

If the status is `inactive (failed)` or `inactive (dead)`, see Restart the vSphere Integrated Containers Services.

Restart the vSphere Integrated Containers Services

You can restart the vSphere Integrated Containers Registry and vSphere Integrated Containers Management Portal services and the file server that run in the appliance by logging in to the vSphere Integrated Containers appliance.

Prerequisites

You deployed the vSphere Integrated Containers appliance.

Procedure

1. Connect to the vSphere Integrated Containers appliance by using SSH.
2. Run one of the following commands to restart one of the vSphere Integrated Containers services:
    - vSphere Integrated Containers Registry: `systemctl restart harbor`
    - vSphere Integrated Containers Management Portal services: `systemctl restart admiral`
    - Embedded file server: `systemctl restart fileserver`

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

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:

- If the server is trusted and you did not specify the certificate thumbprint when you ran `vic-machine create`, specify the `--thumbprint` option, using the thumbprint from the error message.
- If the 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 the new thumbprint in the `--thumbprint` option.

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 `--cert-path`, `--cert`, `--key`, `--appliance-iso`, and `--bootstrap-iso`

For information about when to use quotation marks for different options, see the descriptions of those options in VCH Deployment Options.

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 `--cert-path`. The default value of `--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 `--cert` and `--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 `--cert-path`.
  - Change the value of `--name`. This alters the default `--cert-path`.
  - Specify `--cert-path` explicitly.
  - Delete the existing certificates from `--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 Integrated Containers Plug-In Does Not Appear

After you have installed either of the HTML5 or Flex-based plug-ins for vSphere Integrated Containers, the plug-ins do not appear in the HTML5 vSphere Client or the Flex-based vSphere Web Client.

Problem

The UI plug-in installer reported success, but the plug-in does not appear in the client. Logging out of the client and logging back in again does not resolve the issue.

Cause

- If a previous attempt at installing the vSphere Integrated Containers plug-in failed, the failed installation state is retained in the client cache.
- You installed a new version of the vSphere Integrated Containers plug-in that has the same version number as the previous version, for example a hot patch.

Solution

Restart the client service.

Restart the HTML5 or Flex Client on vSphere 6.5 on Windows

1. Log into the Flex-based vSphere Web Client.
2. Go to **Home** > **System Configuration** > **Services**.
3. Select the service for either the HTML5 or Flex client:
   - HTML5: **VMware vSphere Client**.
   - Flex: **VMware vSphere Web Client**
4. Click the **Restart** button.

Restart the Flex Client on vCenter Server 6.0 on Windows

1. Open Server Manager on the Windows system on which vCenter Server is running.
2. Select **Configuration** > **Services**.
3. Select **VMware vSphere Web Client** and click **Restart**.

Restart the Flex Client on a vCenter Server Appliance

1. Use SSH to log in to the vCenter Server Appliance as `root`.
2. Stop the vSphere Web Client service by running one of the following commands.
   - vCenter Server 6.0:

     ```
     service vsphere-client stop
     ```

   - vCenter Server 6.5:

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

3. Restart the vSphere Web Client service by running one of the following commands.
   - vCenter Server 6.0:

     ```
     service vsphere-client start
     ```

- vCenter Server 6.5:

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

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 Security Reference

The Security Reference provides information to allow you to secure your vSphere Integrated Containers implementation.

Network Security

VMware highly recommends using a secure management network for vSphere Integrated Containers Engine. The container VMs communicate with the endpoint VM over the management network when an interactive shell is required. While the communication is encrypted, the public keys are not validated, which leaves scope for man-in-the-middle attacks. This connection is only used for the interactive console when enabled (stdin/out/err), and not for any other purpose.

External Interfaces, Ports, and Services

The following ports must be open on the VCH appliance.

Endpoint VM

Client interface:

- 2375 insecure port for Docker API access if deployed with `--no-tls`
- 2376 for TLS secured port for Docker API access
- 22 SSH when enabled with `vic-machine debug`
- 2378 VIC admin server health and log access (HTTPS)
- 6060 pprof debug data when enabled with --debug levels

Management interface:

- 2377 incoming connections from container VMs
- 443 outgoing connections established to vSphere target
- 443 outgoing connections established to ESX hosts

Bridge interface:

- 53 DNS server for container name resolution

Public interface:

- any port not listed as used elsewhere can be forwarded to a container VM

Container VM

- 6060 pprof debug data when enabled with --debug levels
- vSphere Integrated Containers Engine does not use ports when not configured for debug

Service Accounts and Privileges {#}

vSphere Integrated Containers Engine does not create service accounts and does not assign privileges. The `--ops-user` and `--ops-password` options allow a VCH to operate with less-privileged credentials than those that are required for deploying a new VCH. For information about the `--ops-user` option and the permissions that it requires, see the descriptions of `--ops-user` in

VCH Deployment Options and Advanced Examples of Deploying a VCH, and the section Use Different User Accounts for VCH Deployment and Operation.

Apply Security Updates and Patches

Download a new version of vSphere Integrated Containers Engine and upgrade your existing VCHs.

Security Related Log Messages

Security-related information for vSphere Integrated Containers Engine appears in `docker-personality.log` and `vicadmin.log`, that you can access from the VCH Admin portal for a VCH.

Sensitive Data

The VMX file of the VCH endpoint VM stores vSphere Integrated Containers Engine configuration information, which allows most of the configuration to be read-only by the guest. The container VMs might hold sensitive application data, such as environment variables for processes, command arguments, and so on.