VMware vSphere Integrated Containers Engine for vSphere Administrators

vSphere Integrated Containers Engine 0.5.5

vmware

Table of Contents

vSphere Integrated Containers Engine for vSphere Administrators

vSphere Integrated Containers Engine for vSphere Administrators provides information about how to install and configure VMware vSphere Integrated Containers Engine.

Product version: 0.5.5

NOTE This book is a work in progress.

Intended Audience

This information is intended for vSphere® Administrators who must manage a vSphere Integrated Containers Engine implementation in their vSphere environment. 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 assumed.

VMware, Inc. 3401 Hillview Ave. Palo Alto, CA94304

www.vmware.com

vSphere Integrated Containers Engine Architecture

vSphere Integrated Containers Engine exists in a vSphere environment, allowing you to manage containers like virtual machines. The architecture consists of these components:

- vCenter Server management tools: Monitor and manage container virtual machines alongside regular virtual machines.
- Trusted networks: Deploy and use vSphere Integrated Containers Engine and create connections between Docker clients and virtual container hosts.
- Virtual SAN datastores: Specify Virtual SAN datastores in which to store container images, container VM files, container volumes, and the virtual container host vApp.
- Docker API appliance virtual machine: The vSphere Integrated Containers Engine installer deploys a vApp, referred to as the virtual container host. You point Docker clients to this appliance for use as the Docker endpoint.
- Docker container virtual machines: Using Photon OS technology, you create and provision multiple container virtual machines directly from a template. The Docker daemon runs outside the container virtual machine. The container is a x86 hardware virtualized virtual machine with a process ID, container interfaces and mounts.

Virtual Container Host

The virtual container host appliance is backed by a Photon OS kernel that provides a virtual container endpoint backed by a vSphere vApp that allows you to control and consume container services.

You can access a Docker API endpoint for development and map ports for client connections to run containers as required.

vSphere resource management handles container placement within the virtual container host, so that a virtual container host can be served by an entire vSphere cluster or by a fraction of the same cluster. The only resources consumed by a container host in the cluster are the resources consumed by the container VMs that run in it.

You can reconfigure the virtual container host with no impact to containers running in it. The virtual container host is not limited by the kernel version or by the operating system that the containers are running.

You can deploy multiple virtual container hosts in an environment, depending on your business needs, including allocating separate resources for development, testing, and production.

You can configure virtual container hosts, giving your development team access to a large virtual container host, or sub-allocate smaller virtual container hosts for individual developers.

Each virtual container host maintains a cache of container images, which you download from either the public Docker Hub or a private registry.

The virtual container host maintains filesystem layers inherent in container images by mapping to discrete VMDK files, all of which are housed in vSphere datastores on VSAN, NFS, or local disks.

You deploy a virtual container host using the CLI installer, then access virtual container host endpoints remotely through a Docker command line interface or other API client.

vSphere Web Client Plugin

You can monitor virtual container hosts and container VMs by using the vSphere Integrated Containers Engine plug-in for the vSphere Web Client

Docker Client

Docker clients communicate with the virtual container host, not with each container, so you can see aggregated pools of vSphere resources, including storage and memory allocations.

You can pull standard container images from the Docker hub or from a private registry.

You can create, run, stop, and delete containers using standard docker commands and verify these actions in the vSphere Web Client.

vSphere Integrated Containers Engine Interoperability with Other VMware Software

IT administrators use vCenter Server to view and manage containers. vSphere Integrated Containers Engine works seamlessly with VMware products.

vRealize Suite

The vRealize Suite is available for health monitoring, performance analysis, and compliance across private and public clouds to move businesses faster.

vSphere High Availability, Fault Tolerance, and vSphere vMotion

IT teams can assure service-level agreements for container workloads with VMware vSphere Distributed Resource Scheduler as well as reduce planned and unplanned downtime with VMware vSphere High Availability and VMware vSphere vMotion.

You can apply vSphere High Availability and Fault Tolerance to both the container VMs and the virtual container host, so that containers and the virtual container host can power on or off independently of each other.

You can also restart or upgrade the virtual container host without needing to take the containers offline during the process. You do not require a native agent on the ESXi host. The appliance VM does not need to be running for vMotion to occur. Clusters with non-container VMs can also vMotion with fully automated DRS.

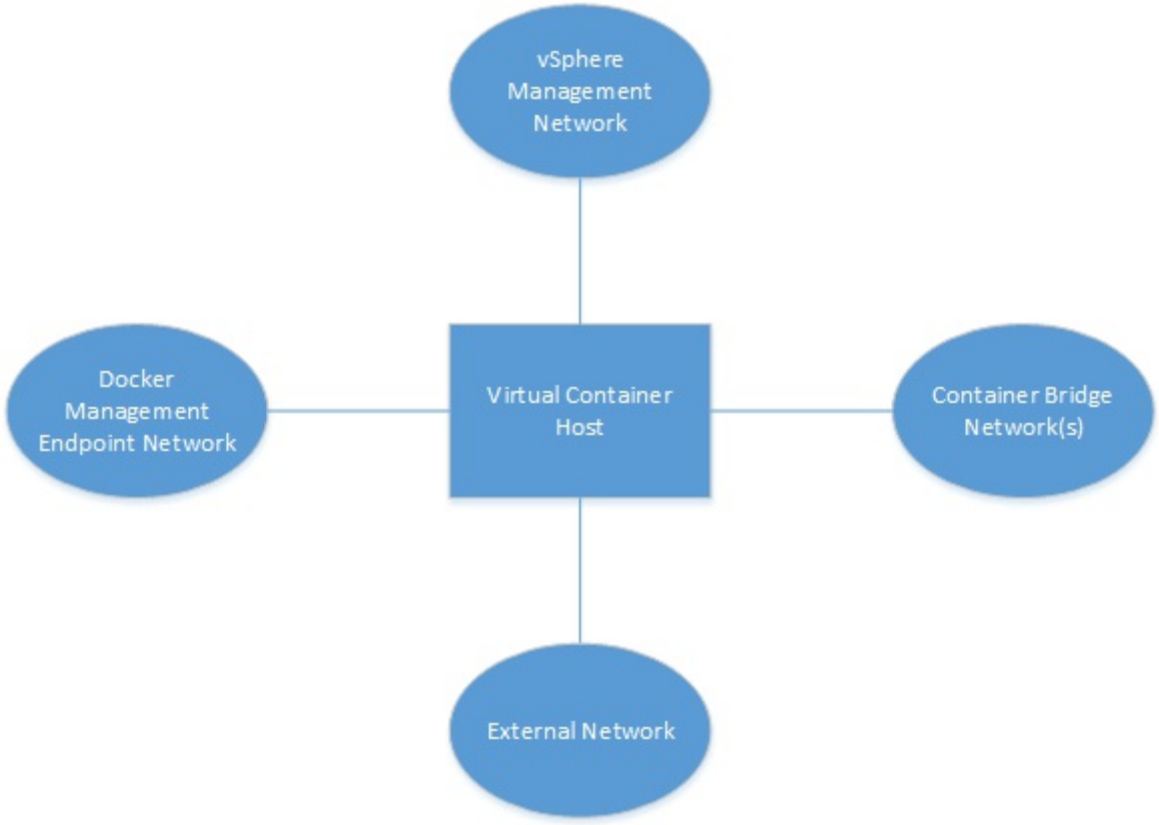
NOTE: In the current builds of vSphere Integrated Containers Engine, support for vMotion is not yet fully implemented.

VMware Virtual SAN

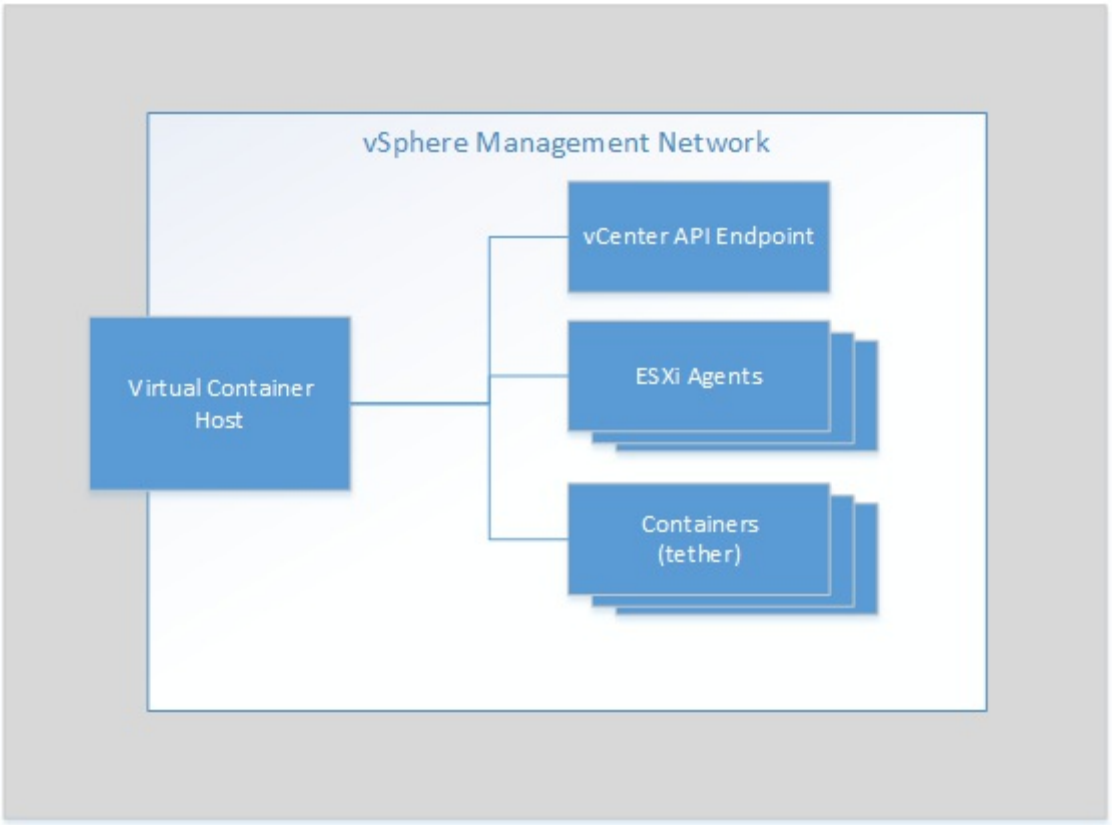
The virtual container host maintains filesystem layers inherent in container images by mapping to discrete VMDK files, all of which are housed in vSphere datastores on VSAN, NFS, or local disks.

vSphere Integrated Containers Engine Network Overview

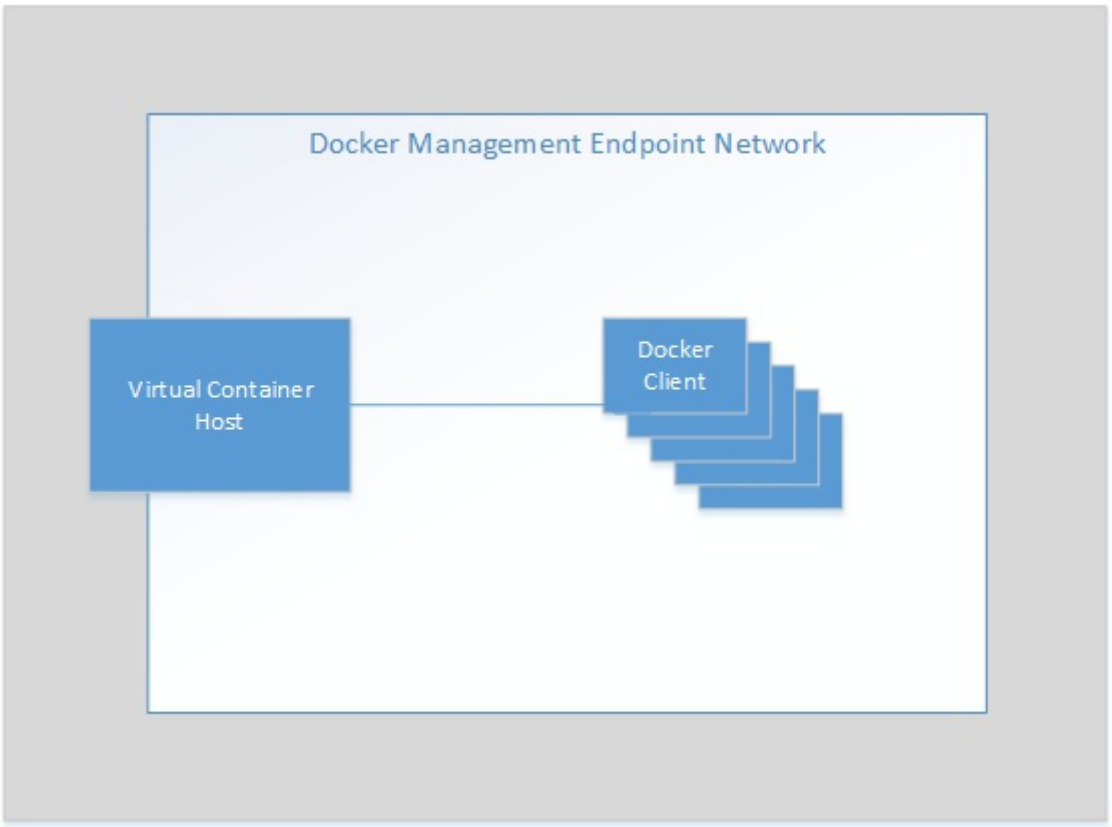
vSphere Container Host connects to four network types.



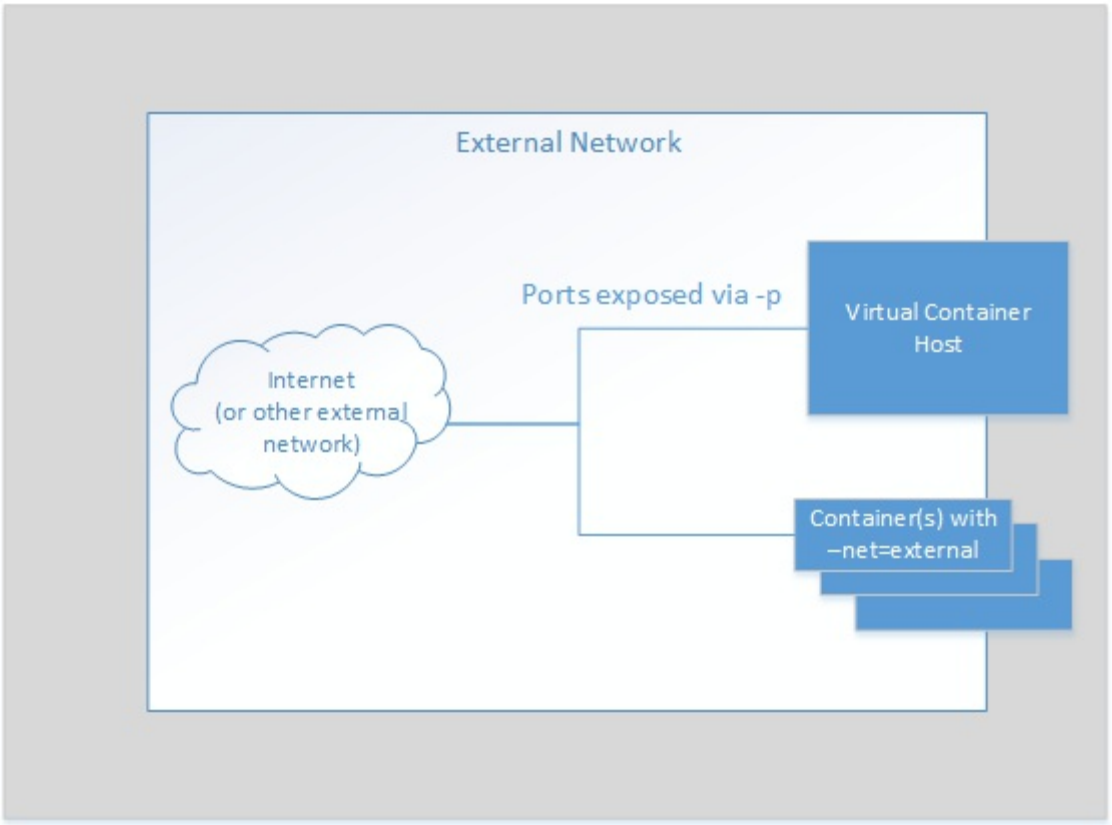
- vSphere Management Network: to communicate with vCenter and ESXi hosts. This network also serves as a tether within the containers to communicate with the vSphere Integrated Containers virtual container host.



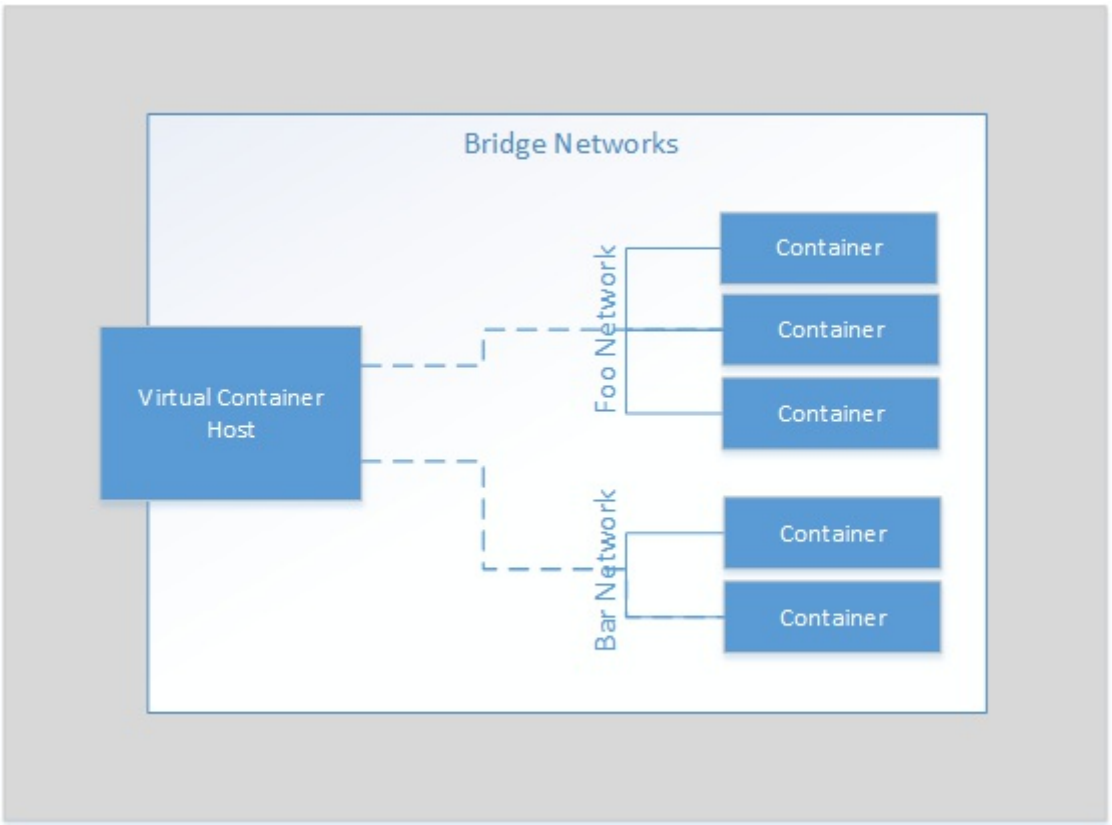
- Docker Management Endpoint Network: to connect to Docker clients and isolate the Docker endpoints from the more public external network.



- External Network: to connect to the internet. Containers can use this external network to publish network services. After defining the external network, you can deploy containers directly on the external interface.



- Container Bridge Network: to allow containers to communicate with each other.



The Port Layer

You can configure networks that are tied into the vSphere infrastructure. Pre-configured networks available to a virtual container host are determined by the networks that are part of the provisioning or added when you reconfigure the virtual container host.

The port layer augments vSphere API with low level, platform-specific primitives to allow you to implement a simple container engine:

- Port Layer Execution: Handles container management, such as create, start, and stop.
- Port Layer Interaction: Handles interaction with a running container.
- Port Layer Networking: Handles specific vSphere NSX network mappings into the Docker network namespace as well as mapping existing network entities such as database servers into the Docker container namespace with defined aliases.
- Port Layer Storage: Provides storage manipulation, including container image storage, layering with volume creation and manipulation. imagec, the docker registry client library, uses this component to translate registry images into a layered format that VMDK disk chains can use directly.

Tether Process

The tether process is a minimal agent in the container VM that starts and stops processes and provides monitoring statistics.

Virtual Container Host Administration

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

- Obtain vic-machine Version Information
- List Virtual Container Hosts and Obtain their IDs
- Obtain Information About a Virtual Container Host
- Delete a Virtual Container Host

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-darwin-linux-windows version
   ```

Result

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

List Virtual Container Hosts and Obtain Their IDs

You can obtain a list of the virtual container hosts that are running in vCenter Server or on an ESXi host by using the `vic-machine ls` command.

The `vic-machine ls` command lists virtual container hosts with their IDs. You can use virtual container host IDs when you run the `vic-machine inspect` and `vic-machine delete` commands. Using virtual container host IDs reduces the number of options that you need to specify when you use `vic-machine inspect` and `vic-machine delete`.

Prerequisites

You have deployed at least one virtual container host.

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 virtual container hosts 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.

   ```
   $ vic-machine-darwin-linux-windows ls
   --target esxi_host esxi_host_address
   --user root
   --password esxi_host_passwor
   ```

   ```
   $ vic-machine-darwin-linux-windows ls
   --target vcenter_server_username:password@vcenter_server_address
   ```

Result

The `vic-machine ls` command lists the virtual container hosts that are running on the ESXi host or vCenter Server instance that you specified.

- Virtual container hosts running on an ESXi host:

  ```
  ID      PATH                                          NAME
  1       /ha-datacenter/host/host_name/Resources       vch_1
  2       /ha-datacenter/host/host_name/Resources       vch_2
  [...]   [...]                                         [...]
  n       /ha-datacenter/host/host_name/Resources       vch_n
  ```

- Virtual container hosts running on a standalone host that is managed by vCenter Server:

  ```
  ID          PATH                                            NAME
  vm-id_1     /datacenter/host/host_address/Resources         vch_1
  vm-id_2     /datacenter/host/host_address/Resources         vch_2
  [...]       [...]                                           [...]
  vm-id_n     /datacenter/host/host_address/Resources         vch_n
  ```

- Virtual container hosts running in a vCenter Server cluster:

```
ID           PATH                                              NAME
vm-id_1      /datacenter/host/cluster_name/Resources            vch_1
vm-id_2      /datacenter/host/cluster_name/Resources            vch_2
[...]        [...]                                             [...]
vm-id_n      /datacenter/host/cluster_name/Resources            vch_n
```

If virtual container hosts are deployed in resource pools on hosts or clusters, the resource pool names appear after `Resources` in the `PATH`.

Virtual Container Host List Options

The command line utility for vSphere Integrated Containers Engine, `vic-machine`, provides an `ls` command that lists the virtual container hosts that are running on an ESXi host or vCenter Server instance. The options that `vic-machine ls` provides allow you to list the virtual container hosts that are running in specific areas of your vSphere environment.

`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 virtual container hosts. This option is 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 virtual container hosts 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 ls` prompts you to enter the password.

- If you deployed the virtual container hosts on a vCenter Server instance that includes more than one datacenter, you can include the datacenter name in the target URL. If you include a datacenter name, `vic-machine ls` lists all of the virtual container hosts that are running in that datacenter. If you do not include a datacenter name, `vic-machine ls` lists all of the virtual container hosts that are running on that vCenter Server instance, for all datacenters.

  ```
  --target vcenter_server_address
  ```

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

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

user

Short name: `-u`

The username for the ESXi host or vCenter Server instance on which you deployed the virtual container hosts. 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 virtual container hosts, or the password for the ESXi host if you deployed directly to an ESXi host. If not specified, `vic-machine ls` 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'
```

compute-resource

Short name: `-r`

The relative path to a resource pool on an ESXi host, or to a host, cluster, or resource pool in vCenter Server, in which you deployed virtual container hosts.

If you specify the `compute-resource` option when the target is a vCenter Server instance that has more than one datacenter, you must include the datacenter name in the `target` option.

If you do not specify the `compute-resource` option, `vic-machine ls` lists all of the virtual container hosts that are running in all of the resource pools on the target ESXi host, or in all of the hosts, clusters, and resource pools on the target vCenter Server instance.

- To list the virtual container hosts in a specific resource pool on an ESXi host that is not managed by vCenter Server, specify the name of the resource pool:

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

- To list the virtual container hosts on a specific standalone host that is managed by vCenter Server, specify the IPv4 address or fully qualified domain name (FQDN) of the host:

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

- To list the virtual container hosts in a specific vCenter Server cluster, specify the name of the target cluster:

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

- To list the virtual container hosts 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
  ```

- To list the virtual container hosts 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'
  ```

debug

Short name: `-v`

Provide verbose logging output, for troubleshooting purposes when running `vic-machine ls`. If not specified, the `debug` value is set to 0 and verbose logging is disabled. Provide a value of 1 or greater to increase the verbosity of the logging.

```
--debug 1
```

Obtain Information About a Virtual Container Host

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

Prerequisites

You have deployed a virtual container host.

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 virtual container host from a simple vCenter Server environment.

   ```
   $ vic-machine-darwin-linux-windows inspect
   --target vcenter_server_username:password@vcenter_server_address
   --name vch_name
   ```

Result

The `vic-machine inspect` command displays the connection information about the virtual container host:

```
vic-admin portal:
https://vch_address:2378
DOCKER_HOST=vch_address:2376
Connect to docker:
docker -H vch_address:2376 --tls info
Completed successfully
```

Virtual Container Host Inspect Options

The command line utility for vSphere Integrated Containers Engine, `vic-machine`, provides an `inspect` command that allows you to see information about virtual container hosts. The options that `vic-machine inspect` requires depend on the location in your vSphere environment in which you deployed the virtual container host.

`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 virtual container host. This option is 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 virtual container host 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 inspect` prompts you to enter the password.

- If you deployed the virtual container host on a vCenter Server instance that includes more than one datacenter, include the datacenter name in the target URL. If you omit the datacenter name or include an invalid datacenter, `vic-machine inspect` 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 virtual container host. 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 virtual container host, or the password for the ESXi host if you deployed directly to an ESXi host. If not specified, `vic-machine inspect` prompts you to enter the password.

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

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'
```

compute-resource

Short name: `-r`

The relative path to the host, cluster, or resource pool in which you deployed the virtual container host. 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.
- The ESXi host includes multiple resource pools.
- You deployed the virtual container host 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 inspect` fails and suggests valid targets for `compute-resource` in the failure message.

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

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

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

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

- If the virtual container host 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 virtual container host 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 virtual container host to inspect. This option is mandatory if the virtual container host to inspect has a name other than the default name, `virtual-container-host`, and you do not use the `id` option. Specify `--name` with exactly the same value that you used when you ran `vic-machine create`.

```
--name vch_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 ID of the virtual container host to inspect, for example `vm-100`. You obtain the ID of a virtual container host by running `vic-machine ls`. If you specify the `id` option, you do not need to specify the `name` or `compute-resource` options.

```
--id vch_id
```

`debug`

Short name: `-v`

Provide verbose logging output, for troubleshooting purposes when running `vic-machine inspect`. If not specified, the `debug` value is set to 0 and verbose logging is disabled. Provide a value of 1 or greater to increase the verbosity of the logging.

```
--debug 1
```

Delete a Virtual Container Host

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

Prerequisites

You have deployed a virtual container host 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 named instance of a virtual container host from a simple vCenter Server environment.

   ```
   $ vic-machine-darwin-linux-windows delete
   --target vcenter_server_username:password@vcenter_server_address
   --name vch_name
   ```

3. If the delete operation fails with a message about container VMs that are powered on, run `vic-machine delete` again with the `--force` option.

   CAUTION Running `vic-machine delete` with the `--force` option removes all running container VMs that the virtual container host manages, as well as any associated volumes and volume stores.

   ```
   $ vic-machine-darwin-linux-windows delete
   --target vcenter_server_username:password@vcenter_server_address
   --name cluster_name
   --force
   ```

Virtual Container Host 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. The options that `vic-machine delete` requires depend on the location in your vSphere environment in which you deployed the virtual container host.

`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 virtual container host. This option is 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 virtual container host 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 delete` prompts you to enter the password.

- If you deployed the virtual container host 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 delete` 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 virtual container host. 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 virtual container host, or the password for the ESXi host if you deployed directly to an ESXi host. If not specified, `vic-machine delete` 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'
```

compute-resource

Short name: `-r`

The relative path to the host, cluster, or resource pool in which you deployed the virtual container host. 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.
- The ESXi host includes multiple resource pools.
- You deployed the virtual container host 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 delete` fails and suggests valid targets for `compute-resource` in the failure message.

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

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

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

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

- If the virtual container host 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 virtual container host 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 virtual container host appliance to delete. This option is mandatory if the virtual container host to delete 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`.

```
--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 ID of the virtual container host to delete, for example `vm-100`. You obtain the ID of a virtual container host by running `vic-machine ls`. If you specify the `id` option, you do not need to specify the `name` or `compute-resource` options.

```
--id vch_id
```

`force`

Short name: `-f`

Forces `vic-machine delete` to ignore warnings and continue with the deletion of a virtual container host. Any running container VMs and any volume stores associated with the virtual container host are deleted. Errors such as an incorrect compute resource still cause the deletion to fail.

- If you do not specify `force` and the virtual container host contains running container VMs, the deletion fails with a warning.
- If you do not specify `force` and the virtual container host has volume stores, the deletion of the virtual container host 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.

```
--force
```

`timeout`

Short name: none

The timeout period for deleting the virtual container host. Specify a value in the format `XmYs` if the default timeout of 3m0s is insufficient.

```
--timeout 5m0s
```

`debug`

Short name: `-v`

Provide verbose logging output, for troubleshooting purposes when running `vic-machine delete`. If not specified, the `debug` value is set to 0 and verbose logging is disabled. Provide a value of 1 or greater to increase the verbosity of the logging.

```
--debug 1
```

Find Virtual Container Host Information in the vSphere Web Client

After you have installed the vSphere Web Client plug-in for vSphere Integrated Containers Engine, you can find information about virtual container hosts in the vSphere Web Client.

Prerequisites

- You deployed a virtual container host.
- You installed the vSphere Web Client plug-in for vSphere Integrated Containers Engine.

Procedure

1. In the vSphere Web Client Home page, select **Hosts and Clusters**.
2. Expand the hierarchy of vCenter Server objects to navigate to the virtual container host vApp.
3. Expand the virtual container host vApp and select the virtual container host endpoint VM.
4. Click the **Summary** tab for the virtual container host VM and scroll down to the Virtual Container Host portlet.

Result

Information about the virtual container host appears in the Virtual Container Host portlet in the **Summary** tab:

- The address of the Docker API endpoint for this virtual container host
- A link to the vic-admin portal for the virtual container host, from which you can obtain health information and download log bundles for the virtual container host.

Find Container Information in the vSphere Web Client

After you have installed the vSphere Web Client plug-in for vSphere Integrated Containers Engine, you can use the vSphere Web Client to find information about containers that are running in virtual container hosts.

Prerequisites

- You deployed a virtual container host and pulled and ran at least one container.
- You installed the vSphere Web Client plug-in for vSphere Integrated Containers Engine.

Procedure

1. In the vSphere Web Client Home page, select **Hosts and Clusters**.
2. Expand the hierarchy of vCenter Server objects to navigate to the virtual container host vApp.
3. Expand the virtual container host vApp and select a container VM.
4. Click the **Summary** tab for the container VM and scroll down to the **Container** portlet.

Result

Information about the container 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.

Troubleshooting vSphere Integrated Containers Engine Administration

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

Running `vic-machine ls` on an ESXi Host Fails with an Error

When you use `vic-machine ls` to list virtual container hosts and you specify the address of an ESXi host in the `target` option, the operation fails with an error.

Problem

Listing virtual container hosts fails with the error message:

```
Target is managed by vCenter server "vcenter_server_address",
please change --target to vCenter server address or select a standalone ESXi
```

Cause

You set the `target` option to the address of an ESXi host that is managed by a vCenter Server instance.

Solution

Set the `target` option to the address of the vCenter Server instance that manages the ESXi host on which the virtual container hosts are running.

Deleting or Inspecting a VCH Fails with a Resource Pool Error

When you use `vic-machine delete` or `vic-machine inspect` to delete or inspect a virtual container host and you specify the address of an ESXi host in the `target` option, the operation fails with a resource pool error.

Problem

Deleting or inspecting a virtual container host fails with the error message:

```
Failed to get VCH resource pool "/ha-datacenter/host/localhost./Resources/vch_name":
resource pool '/ha-datacenter/host/localhost./Resources/vch_name' not found
```

Cause

You set the `target` option to the address of an ESXi host that is managed by a vCenter Server instance.

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`.