VMware vSphere Integrated Containers Installation

vSphere Integrated Containers

vmware

Table of Contents

Introduction 0

Installing vSphere Integrated Containers 1

Environment Prerequisites for vSphere Integrated Containers Installation 2

- Create a Private Port Group for Virtual Container Hosts 2.1

Contents of the vSphere Integrated Containers Binaries 3

Deploy a Virtual Container Host 4

- Virtual Container Host Deployment Options 4.1
- Examples of Deploying a Virtual Container Host 4.2

Verify the Deployment of a Virtual Container Host to vCenter Server 5

Verify the Deployment of a Virtual Container Host to an ESXi Host 6

vSphere Integrated Containers Installation

vSphere Integrated Containers Installation provides information about how to install and configure VMware vSphere Integrated Containers.

Product version 0.3.0

NOTE This book is a work in progress.

For an introduction to vSphere Integrated Containers and descriptions of its main components, see *vSphere Integrated Containers Overview*.

Intended Audience

This information is intended for anyone who wants to install, configure, and get started with using vSphere Integrated Containers. The information is written for experienced VMware 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

Installing vSphere Integrated Containers

You install vSphere Integrated Containers by using a command line installer, that deploys virtual container hosts to VMware ESXi™ hosts or VMware vCenter Server® instances. You can connect Docker clients to the virtual container hosts and use the Docker client to work with containers. You can use your vSphere environment to manage the container VMs and container images.

Environment Prerequisites for vSphere Integrated Containers Installation

Before you install vSphere Integrated Containers, you must ensure that your infrastructure meets certain requirements.

Supported Platforms for `vic-machine`

The current builds of the vSphere Integrated Containers installation and management utility, `vic-machine`, have been tested and verified on the following Linux OS, Windows, and Mac OS systems.

Platform	Supported Versions
Windows	7, 10
Mac OS X	10.11 (TBC)
Linux	Ubuntu 15.04, others TBD

Supported vSphere Configurations

You can install vSphere Integrated Containers in the following vSphere setups:

- Standalone ESXi 6.0 host that is not managed by a vCenter Server instance.
- vCenter Server 6.0, managing one or more standalone ESXi 6.0 hosts.
- vCenter Server 6.0, managing a cluster of ESXi 6.0 hosts, with DRS enabled.

In all cases, your ESXi hosts must have at least 8GB of memory.

Deploying vSphere Integrated Containers to a vCenter Server instance that is running in Enhanced Linked Mode is fully supported.

License Requirements

The type of license that vSphere Integrated Containers requires depends on the way in which you deploy the software.

Type of Installation	vSphere Feature Used	Required License
Standalone ESXi host or vCenter Server managing standalone ESXi hosts	Network Serial Port	vSphere Enterprise
vCenter Server managing a cluster	Distributed Virtual Switch	vSphere Enterprise Plus

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

Role and Permissions Requirements

You must use an account with the vSphere Administrator role when you install vSphere Integrated Containers.

Network Requirements

- Use a trusted network for the deployment and use of vSphere Integrated Containers.
- Use a trused network for connections between Docker clients and the virtual container hosts.
- Each virtual container host requires the following network configuration:
  - One IP address, that can be either static or set by using DHCP.
  - One VLAN, if you use VLAN for network isolation.
  - One IP address for each container that you run with the `docker run --net=host` option.

Create a Private Port Group for Virtual Container Hosts

If you install vSphere Integrated Containers in an environment that includes more than one ESXi host, you must create a private port group for container VMs to use to communicate with each other.

You must create the private port group in advance of deploying a virtual container host, and use the `-bridge-network` option to assign the network to the virtual container host.

In an environment with a single ESXi host, the `-bridge-network` option is optional and you do not need to create a port group in advance. For information about when to specify the `-bridge-network` option, see Virtual Container Host Deployment Options.

You create private port groups in the vSphere Web Client.

IMPORTANT vSphere Integrated Containers does not support tagging on private port groups. Do not apply tags to the private port groups that you create.

1. Log into the vSphere Web Client for the vCenter Server instance that manages your host or cluster.
2. In the **Home** view, click **Hosts and Clusters**.
3. Select an ESXi host and click the **Manage** tab.
4. Select **Networking** > **Virtual switches**.
5. Click the icon to **Add host networking**.
6. Select **Virtual Machine Port Group for a Standard Switch** and click **Next**.
7. Select an existing standard switch, or create a new standard switch, and click **Next**.
8. Rename the network for ease identification, and click **Next**.

   For example, rename the network to `VIC_Network`.
9. Click **Finish**.

Contents of the vSphere Integrated Containers Binaries

After you download and unpack a vSphere Integrated Containers binary bundle from https://bintray.com/vmware/vic-repo/build/view#files, you obtain following files:

File	Description
`appliance.iso`	The ISO from which a virtual container host appliance boots.
`bootstrap.iso`	A Photon OS kernel from which container VMs boot.
`vic_machine`	The command line installation and management utility for virtual container hosts.
`vic_machine-darwin`	The Mac OS command line installation and management utility for virtual container hosts.
`vic_machine-linux`	The Linux command line installation and management utility for virtual container hosts.
`vic machine-windows`	The Windows command line installation and management utility for virtual container hosts.
`README`	Contains a link to the vSphere Integrated Containers repository on GitHub.
`LICENSE`	The license file for vSphere Integrated Containers

If you build the vSphere Integrated Containers binaries manually, you find the ISO files and the `vic_machine` utility in the `<git_installation_dir>/vic/bin` folder.

Deploy a Virtual Container Host

You use the `create` command of the `vic-machine` command line utility to deploy a vSphere Integrated Containers virtual container host.

The `vic-machine` utility can deploy a virtual container host in one of the following setups:

- vCenter Server with a cluster
- vCenter Server with one or more standalone ESXi hosts
- A standalone ESXi host

The virtual container host allows you to use an ESXi host or vCenter Server instance as the Docker endpoint for a Docker client. The containers that you pull or create in your Docker client are stored and managed in the vSphere environment.

NOTE The `vic-machine` utility does not add an extension in the vSphere Web Client.

Prerequisites

- Verify that your vSphere infrastructure meets the requirements in Environment Prerequisites for vSphere Integrated Containers Installation.
- If you are deploying a virtual container host in a vSphere environment with more than one ESXi host, create a private port group for container VMs to use to communicate with each other. For information about how to create a private port group, see Create a Private Port Group for Virtual Container Hosts.
- Obtain either a verified build, the latest daily build, or the source code of vSphere Integrated Containers:
  - Download the most recent verified build of vSphere Integrated Containers from https://github.com/vmware/vic/releases and unpack it. This version has been tested and approved, but it does not reflect the most up-to-date version of the code.
  - Download the latest daily build of vSphere Integrated Containers from https://bintray.com/vmware/vic-repo/build/view#files and unpack it. This version reflects the version of the code as it was at the last daily build. It has not been tested or approved.
  - For the very latest version, for example to include changes that you have made since the last daily build, build the vSphere Integrated Containers binaries from the source code.
- Add the folder that contains the vSphere Integrated Containers binaries to the `PATH` environment variable.
- Familiarize yourself with the vSphere Integrated Containers binaries, as described in Contents of the vSphere Integrated Containers Binaries.
- Familiarize yourself with the options of the `create` command of the `vic-machine` utility described in Virtual Container Host Deployment Options.
- For examples of commands to deploy a virtual container host in various vSphere configurations, see Examples of Deploying a Virtual Container Host.

Procedure

1. Open a terminal on the system on which you downloaded and unpacked the vSphere Integrated Containers binary bundle.
2. Navigate to the directory that contains the `vic-machine` utility:
   - If you downloaded the most recent verified build or the latest daily build, go to the location in which you unpacked the vSphere Integrated Containers bundle.
   - If you built the vSphere Integrated Containers binaries, go to `installation_dir/vic/bin`.
3. Run the `create` command of the `vic-machine` utility.

   The following examples include the fewest possible options, for installation in a simple vCenter Server environment with a cluster.

Deploy a virtual container host from a Mac OS system:

```
$ vic-machine-darwin create
--target vcenter_server_address
--image-datastore datastore_name
--user username
--compute-resource /datacenter_name/host/cluster_name/
--bridge-network network_name
```

Deploy a virtual container host from a Linux OS system:

```
$ vic-machine-linux create
--target vcenter_server_address
--image-datastore datastore_name
--user username
--compute-resource /datacenter_name/host/cluster_name/
--bridge-network network_name
```

Deploy a virtual container host from a Windows system:

```
$ vic-machine-windows create
--target vcenter_server_address
--image-datastore datastore_name
--user username
--compute-resource /datacenter_name/host/cluster_name/
--bridge-network network_name
```

At the end of a successful installation, `vic-machine` displays a success message:

```
Initialization of appliance successful
SSH to appliance (default=root:password)
ssh root@vch_address
Log server:
https://vch_address:2378
Connect to docker:docker -H vch_address:2376
--tls --tlscert='./vch_name-cert.pem'
--tlskey='./vch_name.pem' info
```

4. (Optional) Copy the generated certificate and key files to the Docker client system.

   If you did not explicitly disable TLS certificate generation by using the `generate-cert=false` option, and if your Docker client is not on the same system as the one that you used to run `vic-machine`, you must copy the `vch_name-cert.pem` and `vch_name-key.pem` files to the Docker client system.

What to Do Next

If you did not explicitly disable TLS certificate generation by using the `generate-cert=false` option, and if your Docker client is not on the same system as the one that you used to run `vic-machine`, copy the `vch_name-cert.pem` and `vch_name-key.pem` files to the Docker client system.

To test your virtual container host, see Verify the Deployment of a Virtual Container Host to vCenter Server or Verify the Deployment of a Virtual Container Host to an ESXi Host.

Virtual Container Host Deployment Options

The command line utility for vSphere Integrated Containers, `vic-machine`, provides a `create` command that has options to allow you to customize the deployment of a virtual container host to correspond to your vSphere environment.

- Mandatory Options
- Networking Options
- Datastore Options
- Security Options
- Deployment Options

Mandatory Options

The `create` command of the `vic-machine` utility requires you to provide information about where in your vSphere environment to deploy the virtual container host, the user account to use, and the location in which to store files. The options in this section are **mandatory** in all cases.

`target`

The IPv4 address or fully qualified domain name (FQDN) of the ESXi host or vCenter Server instance on which you are installing vSphere Integrated containers.

- If an ESXi host is managed by vCenter Server, you must provide the address of vCenter Server rather than the address of the host.
- If you are installing vSphere Integrated Containers on a vCenter Server instance, you must specify the `user` option.
- If you are installing vSphere Integrated Containers directly on an ESXi host and you do not specify the `user` option, `vic-machine` uses the `root` account for installation.
- If you do not specify the `passwd` option, `vic-machine` prompts you to enter the password.

To facilitate IP address changes in your infrastructure, provide an FQDN whenever possible, rather than an IP address.

```
--target esxi_host_or_vcenter_server_address
```

`user`

The username for the ESXi host or vCenter Server instance on which you are installing vSphere Integrated containers.

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

```
--user esxi_or_vcenter_server_username
```

`compute-resource`

The path to the host, cluster, or resource pool in which to deploy the virtual container host. You must specify the paths in the `govc` format, including the forward slashes at the beginning and end of the paths.

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

  ```
  --compute-resource
  /ha-datacenter/host/localhost.eng.vmware.com/Resources/resource_pool_name/
  ```

- To deploy to an ESXi host that is not managed by vCenter Server, when the host has no existing resource pools, specify the root resource pool:

  ```
  --compute-resource /ha-datacenter/host/localhost.eng.vmware.com/Resources/
  ```

- To deploy to a vCenter Server instance that has one or more hosts but no clusters, specify the IPv4 address or fully qualified domain name (FQDN) of the target host:

  ```
  --compute-resource /datacenter_name/host/host_address/
  ```

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

  ```
  --compute-resource /datacenter_name/host/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 /datacenter_name/host/host_name/Resources/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 /datacenter_name/host/cluster_name/Resources/resource_pool_name/
  ```

`image-datastore`

The datastore in which to store container image files. When you deploy a virtual container host, `vic-machine` creates a folder named `VIC` on the target datastore, in which to store all of the container images that you pull into a virtual container host. The `vic-machine` utility also places the VM files for the virtual container host in the datastore that you designate as the image store, in a folder that has the same name as the virtual container host.

You can designate the same datastore as the image store for multiple virtual container hosts. In this case, only one `VIC` folder is created in the datastore and the container image files are made available to all of the virtual container hosts that use that image store.

NOTE: In the current builds the `container-store` option is not enabled. As a consequence, container VM files are also stored in the datastore that you designate as the image store.

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

Networking Options

The `vic-machine` utility allows you to specify different networks for the different types of traffic between containers, the virtual container host, the external internet, and your vSphere environment.

`bridge-network`

The network that container VMs use to communicate with each other. You can assign the same network to multiple virtual container hosts.

The `bridge-network` option is **optional** when your environment only has one host. For example:

- ESXi host with no vCenter Server
- vCenter Server with no cluster and one ESXi host
- vCenter Server with a cluster and one ESXi host

In a single host environment, if you do not specify `bridge-network`, `vic-machine` creates a standard virtual switch and a port group that each have the same name as the virtual container host. 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 virtual container host.

The `bridge-network` option is **mandatory** in environments with more than one host. For example:

- vCenter Server with no cluster and more than one standalone ESXi host
- vCenter Server with a cluster with more than one ESXi host

In a multiple-host environment, you must create a private port group before you run `vic-machine`, and pass the port group name to the `bridge-network` option. For information about how to create a port group, see Create a Private Port Group for Virtual Container Hosts.

```
--bridge-network network_name
```

`external-network`

The network for containers to use to connect to the Internet. Containers use the external network to pull container images, for example from https://hub.docker.com/, and to publish network services. If you define the external network, you can deploy containers directly on the external interface.

If not specified, containers use the default VM Network for external traffic.

```
--external-network network_name
```

`management-network`

The network that the virtual container host uses to communicate with vCenter Server and ESXi hosts. Container VMs use this network to communicate with the virtual container host.

If not specified, the virtual container host uses the bridge network for management traffic.

```
--management-network network_name
```

Datastore Options

The `vic-machine` utility allows you to specify the datastores in which to store container VM files and container image files.

`image-datastore`

See image-datastore in the section on mandatory options.

`container-datastore`

The datastore in which to store container VM files. When you run a container, container VM files are stored in folders at the top level of the designated datastore. If multiple virtual container hosts use the same container store, all of the container VM files appear at the top level of the container store. You cannot currently designate a specific datastore folder for the VM files of the containers that run in a particular virtual container host.

If you do not specify the `container-store` option, vSphere Integrated Containers stores container VM files in the same datastore that you specify in the mandatory `image-store` option.

NOTE: In the current builds the `container-store` option is not enabled. Container VM files are stored in the datastore that you designate as the image store.

```
--container-datastore datastore_name
```

Security Options

You can configure a virtual container host to use an automatically generated certificate, no certificate, or a Certificate Authority (CA) certificate to authenticate connections with Docker clients.

`generate-cert`

If you do not set the `generate-cert` option, `vic-machine` by default generates a TLS certificate and key for the virtual container host to use to authenticate with a Docker client. Set the `generate-cert` option to `false` if you do not require certificate-based authentication between the virtual container host and the Docker client. If you use the `cert` and `key` options to upload a custom CA certificate, `vic-machine` does not auto-generate a certificate, without requiring you to set `generate-cert` to `false`.

```
--generate-cert=false
```

`cert`

The path to an X.509 certificate for the Docker API to use to authenticate the virtual container host with a Docker client.

- This option is mandatory if your Docker environment uses TLS certificates that are signed by a CA. For information about how to set up a Docker client to use CA certificates, see https://docs.docker.com/engine/security/https/.
- Use this option in combination with the `key` option, that provides the path to the private key file for the CA certificate.

If you use the `cert` and `key` options, `vic-machine` does not automatically generate certificates. Omit this option if your Docker environment does not use certificates that are signed by a CA.

```
--cert path_to_vcenter_server_certificate --key path_to_vcenter_server_key
```

`key`

The path to the private key file for use with a custom CAcertificate. This option is mandatory if your Docker environment uses certificates that are signed by a CA. For information about how to set up a Docker client to use CA certificates, see https://docs.docker.com/engine/security/https/.

Use this option in combination with the `cert` option, that provides the path to an X.509 certificate file.

```
--cert path_to_vcenter_server_certificate --key path_to_vcenter_server_key
```

Deployment Options

The `vic-machine` utility provides options to customize the deployment of virtual container hosts.

`name`

Aname for the virtual container host appliance. If not specified, `vic-machine` sets the name of the virtual container host to `docker-appliance`. If a virtual container host 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, the deployment of the virtual container host fails. Use the `force` option to replace an existing virtual container host with a new one with the same name.

```
--name vch_appliance_name
```

`password`

The password for the user account on the vCenter Server on which you are deploying the virtual container host, 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.

NOTE: If your password contains special characters, you must wrap the password in single quotation marks (').

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

`force`

Forces `vic-machine` to ignore warnings and non-fatal errors and continue with the deployment of a virtual container host. Errors such as an incorrect compute resource still cause the installation to fail. Using the `force` option deletes any existing virtual container host appliances or datastore folders that have the same name as the one you are specifying in the current deployment.

```
--force
```

`timeout`

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

```
--timeout 5m0s
```

`appliance-iso`

The ISO image from which the virtual container host appliance boots. Omit this option to boot the appliance from the default ISO that is included with `vic-machine`. Set this option to boot the appliance from a different ISO file, for example to reinstall an existing virtual container host or to update it to a newer version.

```
--appliance-iso path_to_ISO_file
```

bootstrap-iso

The ISO image from which container VMs boot. Omit this option to boot container VMs from the default Photon OS ISO that is included with `vic-machine`. Set this option to a different ISO file to boot container VMs with an operating system other than Photon OS.

```
--bootstrap-iso path_to_ISO_file
```

appliance-cpu

The number of virtual CPUs for the virtual container host appliance VM. The default is 1. Set this option to increase the number of CPUs in the virtual container host VM, for example if the virtual container host will handle large volumes of containers, or containers that require a lot of processing power.

```
--appliance-cpu number_of_CPUs
```

appliance-memory

The amount of memory for the virtual container host appliance VM. The default is 2048MB. Set this option to increase the amount of memory in the virtual container host VM, for example if the virtual container host will handle large volumes of containers, or containers that consume a lot of memory.

```
--appliance-memory amount_of_memory
```

Examples of Deploying a Virtual Container Host

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

- Deploy a Virtual Container Host Directly on an ESXi Host with no Resource Pools
- Deploy a Virtual Container Host in a vCenter Server Cluster
- Deploy a Virtual Container Host to a Cluster and Specify External, Management, and Container Networks
- Deploy a Virtual Container Host and Specify a Different Datastore for Container VM Files
- Deploy a Virtual Container Host on a Standalone Host in vCenter Server
- Deploy a Virtual Container Host in a Resource Pool on an ESXi Host
- Deploy a Virtual Container Host in a Resource Pool in a vCenter Server Cluster
- Deploy a Virtual Container Host for Use with a Docker Client that Requires CA Certificates
- Deploy a Virtual Container Host with no TLS Authentication
- Deploy a Virtual Container Host with Customized VM Settings

Deploy a Virtual Container Host Directly on an ESXi Host with no Resource Pools

You can install vSphere Integrated Containers directly on an ESXi host that is not managed by a vCenter Server instance. This example provides the miniumum options required to install vSphere Integrated Containers. The installer prompts you for the password for the ESXi host and deploys a virtual container host appliance with the default name `docker-appliance`. When an ESXi host has no resource pools, you must provide the address of the root resource pool in the `compute-resource` option in the `govc` format, exactly as shown in the example. You specify the datastore in which to store the virtual container host files and container image files in the mandatory `image-store` option.

```
vic-machine-darwin-linux-windows create
--target esxi_host_IPv4_address_or_FQDN
--user root
--compute-resource /ha-datacenter/host/localhost.eng.vmware.com/Resources/
--image-datastore datastore_name
```

Deploy a Virtual Container Host in a vCenter Server Cluster

If vCenter Server manages clusters, you use the `compute-resource` option to specify the address of the cluster to which to deploy the virtual container host. You provide the cluster address in the `govc` format.

In clusters that have more than one host, you must use the `bridge-network` option to specify an existing network for container VMs to use to communicate with each other and with the virtual container host.

In addition to the mandatory options for deployment to a cluster, this example sets the vCenter Single Sign-On user and password and gives the virtual container host a custom name, `vch1`.

```
vic-machine-darwin-linux-windows create
--target vcenter_server_IPv4_address_or_FQDN
--user Administrator@vsphere.local
--password vcenter_sso_password
--compute-resource /datacenter_name/host/cluster_name/
--image-datastore datastore_name
--bridge-network network_name
--name vch1
```

Deploy a Virtual Container Host to a Cluster and Specify External, Management, and Container Networks

If your vSphere environment includes multiple networks, you can direct different types of traffic to different networks.

- You direct the traffic between the virtual container host, container VMs, and the internet to a specific network by specifying the `external-network` option. If you do not specify the `external-network` option, the virtual container host uses the default VM Network for external traffic.
- You direct traffic between ESXi hosts, vCenter Server, and the virtual container host to a specific network by specifying the `management-network` option. If you do not specify the `management-network` option, the virtual container host uses the bridge network for management traffic.
- You direct the traffic between container VMs and between container VMs and the virtual container host to a specific network by specifying the `bridge-network` option. This option is mandatory in environments with more than one ESXi host.

To specify networking options, you must provide the name of a network or private port group that exists on the ESXi host before you deploy a virtual container host. For information about creating a private port group, see Create a Private Port Group for Virtual Container Hosts.

In addition to the mandatory options for deployment to a cluster, this example sets the vCenter Single Sign-On user and password, specifies different networks for the different types of traffic, and gives the virtual container host a custom name, `vch1`.

```
vic-machine-darwin-linux-windows create
--target vcenter_server_IPv4_address_or_FQDN
--user Administrator@vsphere.local
--password vcenter_sso_password
--compute-resource /datacenter_name/host/cluster_name/
--image-datastore datastore_name
--bridge-network network_1_name
--management-network network_2_name
--external-network network_3_name
--name vch1
```

Deploy a Virtual Container Host and Specify a Different Datastore for Container VM Files

If your vSphere environment includes multiple datastores, you can specify different datastores for the container image files and the container VM files. If you specify the `container-store` option, the VM files for container VMs are placed in folders at the top level of the designated datastore. The virtual container host files and the container

image files are placed in the datastore that you specify in the `image-store` option. If you do not specify the `container-store` option, the virtual container host uses the `image-store` datastore for container VM files.

In addition to the mandatory options for deployment to a cluster, this example sets the vCenter Single Sign-On user and password, specifies a different datastore for container VM files, and gives the virtual container host a custom name, `vch1`.

```
vic-machine-darwin-linux-windows create
--target vcenter_server_IPv4_address_or_FQDN
--user Administrator@vsphere.local
--password vcenter_sso_password
--compute-resource /datacenter_name/host/cluster_name/
--image-datastore datastore_1_name
--container-datastore datastore_2_name
--bridge-network network_1_name
--name vch1
```

Deploy a Virtual Container Host on a Standalone Host in vCenter Server

If vCenter Server manages 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 virtual container host. You provide the host address in the `govc` format. If the vCenter Server manages more than one standalone ESXi host, you must specify the `bridge-network` option.

In addition to the mandatory options, this example specifies the vCenter Single Sign-On user and password, and gives the virtual container host a custom name, `vch1`.

```
vic-machine-darwin-linux-windows create
--target vcenter_server_IPv4_address_or_FQDN
--user Administrator@vsphere.local
--password vcenter_sso_password
--compute-resource /datacenter_name/host/host_IPv4_address_or_FQDN/
--image-datastore datastore_name
--bridge-network network_name
--name vch1
```

Deploy a Virtual Container Host in a Resource Pool on an ESXi Host

To deploy a virtual container host in a resource pool on an ESXi host that is not managed by vCenter Server, you specify the resource pool address in the `compute-resource` option. You provide the resource pool address in the `govc` format.

This example uses the minimum required options.

```
vic-machine-darwin-linux-windows create
--target esxi_host_IPv4_address_or_FQDN
--user root
--compute-resource
/ha-datacenter/host/localhost.eng.vmware.com/Resources/resource_pool_name/
--image-datastore datastore_name
```

Deploy a Virtual Container Host in a Resource Pool in a vCenter Server Cluster

To deploy a virtual container host in a resource pool in a vCenter Server Cluster, you specify the resource pool address in the `compute-resource` option. You provide the resource pool address in the `govc` format.

In addition to the mandatory options, this example sets the vCenter Single Sign-On user and password, and gives the virtual container host a custom name, `vch1`.

```
vic-machine-darwin-linux-windows create
--target vcenter_server_IPv4_address_or_FQDN
--user Administrator@vsphere.local
--password vcenter_sso_password
--compute-resource /datacenter_name/host/cluster_name/Resources/resource_pool_name/
--image-datastore datastore_name
--bridge-network network_name
--name vch1
```

Deploy a Virtual Container Host for Use with a Docker Client that Requires CA Certificates

If your development environment uses custom CA certificates to authenticate connections between Docker clients and virtual container hosts, use the `cert` and `key` options to provide the paths to an X.509 certificate and its key when you deploy a virtual container host. The paths to the certificate and key files must be relative to the location from which you are deploying a virtual container host.

In addition to the mandatory options for deployment to a cluster, this example sets the vCenter Single Sign-On user and password, and gives the virtual container host a custom name, `vch1`.

```
vic-machine-darwin-linux-windows create
--target vcenter_server_IPv4_address_or_FQDN
--user Administrator@vsphere.local
--password vcenter_sso_password
--compute-resource /datacenter_name/host/cluster_name/
--image-datastore datastore_name
--bridge-network network_name
--cert path_to_certificate_file
--key path_to_certificate_file
--name vch1
```

Deploy a Virtual Container Host with no TLS Authentication

If your development environment does not require authentication of the connections between Docker clients and virtual container hosts, use the `generate-cert=false` option to deactivate TLS in the virtual container host. If you do not specify `generate-cert=false` when you deploy a virtual container host, vic-machine automatically generates a certificate and key file that you must provide to the Docker client when you connect it to this virtual container host.

In addition to the mandatory options for deployment to a cluster, this example sets the vCenter Single Sign-On user and password, and gives the virtual container host a custom name, `vch1`.

```
vic-machine-darwin-linux-windows create
--target vcenter_server_IPv4_address_or_FQDN
--user Administrator@vsphere.local
--password vcenter_sso_password
--compute-resource /datacenter_name/host/cluster_name/
--image-datastore datastore_name
--bridge-network network_name
--name vch1
--generate-cert=false
```

Deploy a Virtual Container Host with Customized VM Settings

If a virtual container host is likely to handle heavy container workloads, you can increase the number of CPUs and the amount of memory in the virtual container host VM by setting the `appliance-cpu` and `appliance-memory` options. For example, set these options if the virtual container host will handle large volumes of containers, containers that consume a lot of memory, or containers that require a lot of processing power. If these options are not set, `vic-machine` deploys a virtual container host with 1 CPU and 2048 MB of memory.

In addition to the mandatory options for deployment to a cluster, this example sets the vCenter Single Sign-On user and password and gives the virtual container host a custom name, `vch1`.

```
vic-machine-darwin-linux-windows create
--target vcenter_server_IPv4_address_or_FQDN
--user Administrator@vsphere.local
--password vcenter_sso_password
--compute-resource /datacenter_name/host/cluster_name/
--image-datastore datastore_name
--bridge-network network_name
--name vch1
--appliance-cpu number_of_CPUs
--appliance-memory amount_of_memory
```

Verify the Deployment of a Virtual Container Host to vCenter Server

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

Prerequisites

- You used `vic-machine create` to deploy a virtual container host to vCenter Server. For information about running `vic-machine`, see Deploy a Virtual Container Host.
- You have installed a Docker client.
- If you did not explicitly disable certificate generation when you deployed the virtual container host, copy the `vch_name-cert.pem` and `vch_name-key.pem` files to the system on which you you are running the Docker client.
- Connect a vSphere Web Client to the vCenter Server instance on which you deployed the virtual container host.

Procedure

1. In the vSphere Web Client, go to **Hosts and Clusters** and select the cluster or host on which you deployed the virtual container host.

   You should see a resource pool with the name that you set for the virtual container host. The resource pool contains the virtual container host appliance VM.

2. In your Docker client, run the `docker info` command to confirm that you can connect to the virtual container host.

   - Without TLS authentication:

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

   - With TLS authentication:

     ```
     $ docker -H vch_address:2376
     --tls
     --tlscert='path_to_cert_file/vch_name-cert.pem'
     --tlskey='path_to_key_file/vch_name-key.pem'
     info
     ```

   You should see confirmation that the Storage Driver is `Portlayer Storage` and the `Name` is `VIC`.

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

   - Without TLS authentication:

     ```
     $ docker -H vch_address:2375 pull busybox:latest
     ```

   - With TLS authentication:

     ```
     $ docker -H vch_address:2376
     --tls
     --tlscert='path_to_cert_file/vch_name-cert.pem'
     --tlskey='path_to_key_file/vch_name-key.pem'
     pull busybox:latest
     ```

4. In the vSphere Web Client, go to **Storage**, select the datastore that you designated as the image store, click **Manage** > **Files**, and navigate to the **VIC** > **Photon** folder.

   You should see a folder for every container image that you have pulled into the virtual container host. The folders contain the container image files.

5. In your Docker client, run the Docker container that you pulled into the virtual container host.

   - Without TLS authentication:

     ```
     $ docker -H vch_address:2375 run busybox
     ```

   - With TLS authentication:

     ```
     $ docker -H vch_address:2376
     --tls
     --tlscert='path_to_cert_file/vch_name-cert.pem'
     --tlskey='path_to_key_file/vch_name-key.pem'
     run busybox
     ```

6. In the vSphere Web Client, go to **Hosts and Clusters** and select the resource pool that contains the virtual container host.

   You should see a VM for every container that you run.

7. In the vSphere Web Client, go to **Storage** and select the datastore that you designated as the container store.

   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.

Verify the Deployment of a Virtual Container Host to an ESXi Host

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

Prerequisites

- You used `vic-machine create` to deploy a virtual container host to a standalone ESXi host. For information about running `vic-machine`, see Deploy a Virtual Container Host.
- You have installed a Docker client.
- If you did not explicitly disable certificate generation when you deployed the virtual container host, copy the `vch_name-cert.pem` and `vch_name-key.pem` files to the system on which you you are running the Docker client.
- Connect a vSphere Client to the ESXi host on which you deployed the virtual container host.

Procedure

1. In the vSphere Web Client, go to **Inventory** and select the host on which you deployed the virtual container host.

   You should see a resource pool with the name that you set for the virtual container host. The resource pool contains the virtual container host appliance VM.

2. In your Docker client, run the `docker info` command to confirm that you can connect to the virtual container host.

   - Without TLS authentication:

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

   - With TLS authentication:

     ```
     $ docker -H vch_address:2376
     --tls
     --tlscert='path_to_cert_file/vch_name-cert.pem'
     --tlskey='path_to_key_file/vch_name-key.pem'
     info
     ```

   You should see confirmation that the Storage Driver is `Portlayer Storage` and the `Name` is `VIC`.

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

   - Without TLS authentication:

     ```
     $ docker -H vch_address:2375 pull busybox:latest
     ```

   - With TLS authentication:

     ```
     $ docker -H vch_address:2376
     --tls
     --tlscert='path_to_cert_file/vch_name-cert.pem'
     --tlskey='path_to_key_file/vch_name-key.pem'
     pull busybox:latest
     ```

4. In the vSphere Client, click the **Summary** tab for the host, right-click the datastore that you designated as the image store, select **Browse Datastore**, and navigate to the **VIC** > **Photon** folder.

   You should see a folder for every container image that you have pulled into the virtual container host. The folders contain the container image files.

5. In your Docker client, run the Docker container that you pulled into the virtual container host.

   - Without TLS authentication:

     ```
     $ docker -H vch_address:2375 run busybox
     ```

   - With TLS authentication:

     ```
     $ docker -H vch_address:2376
     --tls
     --tlscert='path_to_cert_file/vch_name-cert.pem'
     --tlskey='path_to_key_file/vch_name-key.pem'
     run busybox
     ```

6. In the vSphere Client, go to **Inventory** and select the resource pool that contains the virtual container host.

   You should see a VM for every container that you run.

7. In the vSphere Client, click the **Summary** tab for the host, right-click the datastore that you designated as the container 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.