VMware vSphere Integrated Containers Installation

vSphere Integrated Containers

vmware

Table of Contents

vSphere Integrated Containers Installation

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

Product version: 0.5

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 vSphere Integrated Containers installation and management utility, `vic-machine`, has 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 (El Capitan)
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.

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

ESXi Host Requirements

To be valid targets for virtual container hosts and container VMs, standalone ESXi hosts and all ESXi hosts in vCenter Server clusters must meet the following criteria:

- All ESXi hosts must be attached to shared storage for use as container datastores, image datastores, and volume stores.
- The firewall on all ESXi hosts must be configured to allow connections on the back channel.
- All ESXi hosts must be attached to the distributed virtual switch for the bridge network in vCenter Server. For more information about distributed virtual switches, see Network Requirements below.
- All ESXi hosts must be attached to any mapped vSphere networks.

During deployment of virtual container hosts, the `vic-machine` utilty checks that the target ESXi hosts meet the requirements, and issues warnings if they do not.

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
ESXi host	Network Serial Port	vSphere Enterprise
vCenter Server	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 trusted 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.
- In a vCenter Server environment, before you deploy a virtual container host, you must create a distributed virtual switch and a distributed port group. You must add the target ESXi host or hosts to the distributed virtual switch.
  - For information about how to create a distributed port group, see Create a vSphere Distributed Switch in the vSphere 6.0 documentation.
  - For information about how to add hosts to a distributed virtual switch, see Add Hosts to a vSphere Distributed Switch in the vSphere 6.0 documentation.

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-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

When you deploy a virtual container host, `vic-machine` registers the virtual container host as a vSphere extension. Authentication between the virtual container host and vSphere is handled via key pair authentication against the vSphere extension.

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.
- In a vCenter Server environment, before you deploy a virtual container host, you must create a distributed virtual switch and a distributed port group. You must add the target ESXi host or hosts to the distributed virtual switch. For information about how to create a distributed virtual switch and port group, see *Network Requirements* in Environment Prerequisites for vSphere Integrated Containers Installation.
- 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-store datastore_name
   --user username
   --bridge-network network_name
   ```

   Deploy a virtual container host from a Linux OS system:

   ```
   $ vic-machine-linux create
   --target vcenter_server_address
   --image-store datastore_name
   --user username
   --bridge-network network_name
   ```

   Deploy a virtual container host from a Windows system:

   ```
   $ vic-machine-windows create
   --target vcenter_server_address
   --image-store datastore_name
   --user username
   --bridge-network network_name
   ```

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

   ```
   Initialization of appliance successful
   vic-admin portal:
   https://vch_address:2378
   DOCKER_HOST=vch_address:2376
   Connect to docker:
   docker -H vch_address:2376 --tls info
   Installer completed successfully
   ```

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 `no-tls` 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 `no-tls` 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

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, the bridge network to use, and the location in which to store container image files. The options in this section are **mandatory**.

`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 virtual container host.

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

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

- If you are deploying a virtual container host directly on an ESXi host, you must specify the `user` option, or include the user name and password in the target URL. Wrap the user name or password in single quotes (Linux or Mac OS) or double quotes (Windows) if they include special characters.

  ```
  --target esxi_host_username:password@esxi_host_address
  ```

- If you are deploying a virtual container host on a vCenter Server instance, you must specify the `user` option, or include the user name and password in the target URL. Wrap the user name or password in single quotes (Linux or Mac OS) or double quotes (Windows) if they include special characters.

  ```
  --target vcenter_server_username:password@vcenter_server_address
  ```

- If you are deploying a 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 create` fails and suggests the available datacenters that you can specify.

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

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

user

Short name: `-u`

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
```

You can also specify the username in the URL that you pass to `vic-machine create` in the `target` option.

image-store

Short name: `-i`

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 specify a datastore folder to use as the image store in the format `datastore/path`. In this case, the virtual container host uses `path` as the image store instead of using the folder with the same name as the virtual container host. If the folder that you specify does not already exist, `vic-machine create` creates it.

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.

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

NOTES:

- vSphere Integrated Containers supports all alphanumeric characters, hyphens, and underscores in datastore paths and datastore names.
- In the current builds the `container-datastore` 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
```

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

bridge-network

Short name: `-b`

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

IMPORTANT The bridge network is used exclusively by vSphere Integrated Containers. You cannot use the network that you specify in the `bridge-network` option as the target for any of the other `vic-machine create` networking options.

The `bridge-network` option is **optional** when you are deploying a virtual container host 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 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** if you are deploying a virtual container host to vCenter Server.

In a vCenter Server environment, before you run `vic-machine create`, you must create a distributed virtual switch and a distributed port group. You must add the target ESXi host or hosts to the distributed virtual switch. For information about how to create a distributed virtual switch and port group, see *Network Requirements* in Environment Prerequisites for vSphere Integrated Containers Installation.

You pass the name of the distributed port group to the `bridge-network` option. If you specify an invalid network name, `vic-machine create` fails and suggests valid networks.

```
--bridge-network distributed_port_group_name
```

Networking Options

The `vic-machine create` 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`

See bridge-network in the section on mandatory 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.

```
--bridge-network-range 192.168.100.0/24
```

`external-network`

Short name: `--en`

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

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

`management-network`

Short name: `--mn`

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 external network for management traffic. If you specify an invalid network name, `vic-machine create` fails and suggests valid networks.

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

`client-network`

Short name: `--cln`

The network that the virtual container host uses to generate the Docker API. The Docker API only uses this network.

If not specified, the virtual container host uses the external network for client traffic. If you specify an invalid network name, `vic-machine create` fails and suggests valid networks.

```
--client-network network_name
```

`container-network`

Short name: `--cn`

Anetwork for container VMs to use for external communication when you run `docker run` or `docker create` with the `--net` option.

To specify a container network, you provide the name of a distributed port group for the container VMs to use, and a descriptive name for the container network that is used by Docker. If you specify an invalid network name, `vic-machine create` fails and suggests valid networks.

- The distributed port group must exist before you run `vic-machine create` .
- You cannot use the same distributed port group that you use for the bridge network.
- You can create the distributed port group on the same distributed virtual switch as the distributed port group that you use for the bridge network.
- The descriptive name appears under `Networks` when you run `docker info` on the deployed virtual container host.
- Container developers use the descriptive name in the `--net` option when they run `docker run` or `docker create` .

If you do not specify the `container-network` option, or if you run `docker run` or `docker create` without specifying `--net` , container VMs use the bridge network.

If the network that you specify does not support DHCP, you must also specify the `container-network-gateway` option.

```
--container-network distributed_port_group_name:container_network_name
```

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 use the distributed port group that you specify in the `container-network` option.

```
--container-network-gateway distributed_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 use the distributed port group that you specify in the `container-network` option.

```
--container-network-dns distributed_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 do not specify this option, 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 use the distributed port group that you specify in the `container-network` option.

```
--container-network-ip-range distributed_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 distributed_port_group_name:192.168.100.0/24
```

Compute Resource Options

If the vCenter Server instance on which you are deploying a virtual container host only includes a single instance of a standalone host or cluster, `vic-machine create` automatically detects and uses those resources. If you are deploying to an ESXi host that has no resource pools, `vic-machine create` automatically uses the default resource pool. In these cases, you do not need to specify a compute resource when you run `vic-machine create`.

compute-resource

Short name: `-r`

The relative path to the host, cluster, or resource pool in which to deploy the virtual container host. 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.
- An ESXi host includes multiple resource pools.
- You want to deploy the virtual container host to a specific resource pool in your environment.

If you do not specify the `compute-resource` option and multiple possible resources exist, `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 but no clusters, 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
  ```

Datastore Options

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

- vSphere Integrated Containers fully supports VMware Virtual SAN datastores.
- vSphere Integrated Containers supports all alphanumeric characters, hyphens, and underscores in datastore paths and datastore names.

`image-store`

See image-store in the section on mandatory options.

`container-datastore`

Short name: `--cs`

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-datastore` option, vSphere Integrated Containers stores container VM files in the same datastore that you specify in the mandatory `image-store` option.

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

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

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

`volume-store`

Short name: `--vs`

The datastore in which to create volumes when using the `docker volume create` command. 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 you specify an invalid datastore name, `vic-machine create` fails and suggests valid datastores.

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 virtual container host when container application developers run `docker info`. Container application developers also specify the volume store label in the `docker volume create --opt VolumeStore=volume_store_label` option when they create a volume.

- 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 application developers do not need to specify the `--opt VolumeStore=volume_store_label` option when they run `docker volume create`.

  NOTE: If container application developers intend to create anonymous volumes by using `docker create -v`, 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 `volumes` under the `VIC` folder on the target datastore. Any volumes that container application developers create will appear in the `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 `VIC/volumes` in the location that you specify in the datastore path. If the folders that you specify in the path do not already exist on the datastore, `vic-machine create` creates the appropriate folder structure. Any volumes that container application developers create will appear in the `path/VIC/volumes` folder.

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

- You can specify the `volume-store` option multiple times, to create multiple volume stores on the virtual container host.

```
--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
```

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.

`no-tls`

Short name: `-k`

If you do not set the `no-tls` 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 `no-tls` option 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 `no-tls` to `false`.

```
--no-tls
```

`cert`

Short name: none

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`

Short name: none

The path to the private key file for use with a custom CA certificate. 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
```

Appliance Deployment Options

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

`name`

Short name: `-n`

Aname for the virtual container host appliance. If not specified, `vic-machine` sets the name of the virtual container host to `virtual-container-host`. 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`

Short name: `-p`

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 (') on Mac OS and Linux and in double quotation (") marks on Windows.

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

`force`

Short name: `-f`

Forces `vic-machine create` 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.

```
--force
```

`timeout`

Short name: none

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`

Short name: `--ai`

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

Short name: `--bi`

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

Short name: none

The number of virtual CPUs for the virtual container host 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

Short name: none

The amount of memory for the virtual container host 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
```

use-rp

Short name: none

Deploy the virtual container host to a resource pool rather than to a vApp. If you specify this option, `vic-machine create` creates a resource pool with the same name as the virtual container host.

```
--use-rp
```

pool-memory-reservation

Short name: `--pmr`

Reserve a quantity of memory for use by the vApp or resource pool that contains the virtual container host and container VMs. Specify the memory reservation value in MB. If not specified, `vic-machine create` sets the reservation to 1.

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

`pool-memory-limit`

Short name: `--pml`

Limit the amount of memory that is available for use by the vApp or resource pool that contains the virtual container host and container VMs. Specify the memory limit value in MB. If not specified, `vic-machine create` sets the limit to 0 (unlimited).

```
--pool-memory-limit 1024
```

`pool-memory-shares`

Short name: `--pms`

Set memory shares on the vApp or resource pool that contains the virtual container host and container VMs. 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`.

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

`pool-cpu-reservation`

Short name: `--pcr`

Reserve a quantity of CPU capacity for use by the vApp or resource pool that contains the virtual container host and container VMs. Specify the CPU reservation value in MHz. If not specified, `vic-machine create` sets the reservation to 1.

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

`pool-cpu-limit`

Short name: `--pcl`

Limit the amount of CPU capacity that is available for use by the vApp or resource pool that contains the virtual container host and container VMs. Specify the CPU limit value in MHz. If not specified, `vic-machine create` sets the reservation to 0 (unlimited).

```
--pool-cpu-limit 1024
```

`pool-cpu-shares`

Short name: `--pcs`

Set CPU shares on the vApp or resource pool that contains the virtual container host and container VMs. 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`.

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

debug

Short name: `-v`

Provide verbose logging output, for troubleshooting purposes when running `vic-machine create`. 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. Note that setting debug to a value greater than 1 can affect the behavior of `vic-machine create`.

```
--debug 1
```

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

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 `virtual-container-host`. 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
--image-store datastore_name
```

Deploy a Virtual Container Host in a vCenter Server Cluster

If vCenter Server manages more than one cluster, you use the `compute-resource` option to specify the address of the cluster to which to deploy the virtual container host.

You must use the `bridge-network` option to specify an existing distributed port group 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 cluster_name
--image-store datastore_name
--bridge-network distributed_port_group_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 distributed port group that exists in your vCenter Server environment before you deploy a virtual container host. For information about how to create a distributed virtual switch and port group, see *Network Requirements* in Environment Prerequisites for vSphere Integrated Containers Installation.

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 cluster_name
--image-store 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 cluster_name
--image-store datastore_1_name
--container-datastore datastore_2_name
--bridge-network network_1_name
--name vch1
```

It is also possible to specify a specific folder in which to store your images, instead of just a datastore. This can be done by providing a path following the `image-store` option:

```
vic-machine-darwin-linux-windows create
--target vcenter_server_IPv4_address_or_FQDN
--user Administrator@vsphere.local
--password vcenter_sso_password
--compute-resource cluster_name
--image-store datastore_1_name/path/to/some/specific/location
--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 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 virtual container host. 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 host_IPv4_address_or_FQDN
--image-store 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.

This example uses the minimum required options.

```
vic-machine-darwin-linux-windows create
--target esxi_host_IPv4_address_or_FQDN
--user root
--compute-resource resource_pool_name
--image-store 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.

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 cluster_name/resource_pool_name
--image-store 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 CAcertificates 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 cluster_name
--image-store 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 `no-tls` option to deactivate TLS in the virtual container host. If you do not specify `no-tls` 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 cluster_name
--image-store datastore_name
--bridge-network network_name
--name vch1
--no-tls
```

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 cluster_name
--image-store 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 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.

Installing the vSphere Web Client Plug-In for vSphere Integrated Containers

You can install a plug-in that adds information about virtual container hosts and container VMs in the vSphere Web Client.

You can install the plug-in for vSphere Integrated Containers either on a vCenter Server instance that runs on Windows, or on a vCenter Server Appliance.

In the current builds of vSphere Integrated Containers, information about virtual container hosts and container VMs appears in the **Summary** tabs for those VMs.

- Install the vSphere Integrated Containers Plug-In on vCenter Server For Windows by Using a Web Server
- Install the vSphere Integrated Containers Plug-In on vCenter Server For Windows, by Using SFTP
- Install the vSphere Integrated Containers Plug-In on vCenter Server For Windows, without SFTP or a Web Server
- Install the vSphere Integrated Containers Plug-In on a vCenter Server Appliance by Using a Web Server
- Install the vSphere Integrated Containers Plug-In on a vCenter Server Appliance Without Access to a Web Server

Install the vSphere Integrated Containers Plug-In on vCenter Server For Windows by Using a Web Server

If your vCenter Server instance runs on Windows, you can use a Web server to host the vSphere Web Client plug-in for vSphere Integrated Containers.

Prerequisites

- You deployed at least one virtual container host to a vCenter Server instance.
- Download the latest build of the vSphere Integrated Containers plug-in bundle from https://bintray.com/vmware/vic-repo/build-deps/installer-onsite-beta#files.
- You are running a Web server that your vCenter Server instance can access.

Procedure

1. Unpack the `vic-ui-installer.zip` bundle on the Windows machine on which vCenter Server is running.
2. Upload the plug-in bundle to your Web server.

   ```
   unpack_dir\installer\vsphere-client-serenity\com.vmware.vicui.Vicui-0.0.1.zip
   ```

3. Open the `unpack_dir\installer\vCenterForWindows\configs` file in a text editor.
4. Enter the IPv4 address or FQDN of the vCenter Server instance on which to install the plug-in.

   ```
   target_vcenter_ip=vcenter_server_address
   ```

5. Enter the URL of the ZIP file on your Web server.

   ```
   vic_ui_host_url=vic_web_server_location
   ```

6. (Optional) If you used an HTTPS address in `vic_ui_host_url`, provide the SHA-1 thumbprint of the Web server.

   ```
   vic_ui_host_thumbprint=thumbprint
   ```

7. Save and close the `configs` file.
8. Open a command prompt and navigate to `unpack_dir\installer\vCenterForWindows`.
9. Run the installer.

   ```
   install.bat
   ```

   Make sure that you use the correct account to run `install.bat`.
   - If vCenter Server uses the local system account, run `install.bat` with the local system account.
   - If vCenter Server uses a different user account, run `install.bat` with that account.
10. Enter the password for the vCenter Server administrator account.
11. When installation finishes, if you are logged into the vSphere Web Client, log out then log back in again.

Install the vSphere Integrated Containers Plug-In on vCenter Server For Windows by Using SFTP

If your vCenter Server instance runs on a Windows system that has access to an SFTP service, you can use SFTP to install the vSphere Web Client plug-in for vSphere Integrated Containers.

Prerequisites

- You deployed at least one virtual container host to a vCenter Server instance.
- Download the latest build of the vSphere Integrated Containers plug-in bundle from https://bintray.com/vmware/vic-repo/build-deps/installer-onsite-beta#files.
- Your vCenter Server instance is running on a Windows system that has an SFTP service.

Procedure

1. Unpack the `vic-ui-installer.zip` bundle on a Windows system that has access to an SFTP service.
2. Open the `unpack_dir\installer\vCenterForWindow\configs` file in a text editor.
3. Enter the IPv4 address or FQDN of the vCenter Server instance on which to install the plug-in.

   ```
   target_vcenter_ip=vcenter_server_address
   ```

4. Make sure that `sftp_supported` is set to 1.

   ```
   sftp_supported=1
   ```

5. Provide the SFTP user name and password.

   ```
   sftp_username=usernamesftp_password=password
   ```

6. Provide the location of the `vsphere-client-serenity` folder on the vCenter Server Windows system relative to the root folder of the SFTP connection.

   For example, if the root folder of the SFTP connection is `vcenter_server_install_dir\vCenterServer\cfg`, set the relative path as follows:

   ```
   target_vc_packages_path=/vsphere-client/vc-packages/vsphere-client-serenity/
   ```

7. Save and close the `configs` file.
8. Open a command prompt and navigate to `unpack_dir\installer\vCenterForWindows`.
9. Run the installer.

   ```
   install.bat
   ```

   Make sure that you use the correct account to run `install.bat`.
   - If vCenter Server uses the local system account, run `install.bat` with the local system account.
   - If vCenter Server uses a different user account, run `install.bat` with that account.
10. Enter the password for the vCenter Server administrator account.
11. When installation finishes, if you are logged into the vSphere Web Client, log out then log back in again.

Install the vSphere Integrated Containers Plug-In on vCenter Server For Windows, without SFTP or a Web Server

You can install the vSphere Web Client plug-in for vSphere Integrated Containers on a vCenter Server instance that has neither access to an SFTP service nor to a Web Server.

Prerequisites

- You deployed at least one virtual container host to a vCenter Server instance.
- Download the latest build of the vSphere Integrated Containers plug-in bundle from https://bintray.com/vmware/vic-repo/build-deps/installer-onsite-beta#files.

Procedure

1. Unpack the `vic-ui-installer.zip` bundle on the Windows machine on which vCenter Server is running.
2. Copy the `com.vmware.vicui.Vicui-0.0.1` folder into the folder that contains the vSphere Web Client packages.
   - Source folder:

     ```
     unpack_dir\installer\vsphere-client-serenity
     ```

   - Destination folder:

     ```
     instl_dir\vCenterServer\cfg\vsphere-client\vc-packages\vsphere-client-serenity
     ```

     `instl_dir` is the location in which vCenter Server is installed. If the `vc-packages\vsphere-client-serenity` folders do not exist under the `vsphere-client` folder, create them manually.
3. Open the `unpack_dir\installer\vCenterForWindows\configs` file in a text editor.
4. 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
   ```

5. Deactivate SFTP by changing the value of `sftp_supported` to 0.

   ```
   SET sftp_supported=0
   ```

6. Save and close the `configs` file.
7. Open a command prompt and navigate to `unpack_dir\installer\vCenterForWindows`.
8. Run the installer.

   ```
   install.bat
   ```

   Make sure that you use the correct account to run `install.bat`.
   - If vCenter Server uses the local system account, run `install.bat` with the local system account.
   - If vCenter Server uses a different user account, run `install.bat` with that account.
9. Enter the password for the vCenter Server administrator account.
10. When installation finishes, if you are logged into the vSphere Web Client, log out then log back in again.

Install the vSphere Integrated Containers Plug-In on a vCenter Server Appliance by Using a Web Server

If you are running the vCenter Server Appliance, you can use a Web server to host the vSphere Web Client plug-in for vSphere Integrated Containers.

Prerequisites

- You deployed at least one virtual container host to a vCenter Server Appliance instance.
- Download the latest build of the vSphere Integrated Containers plug-in bundle from https://bintray.com/vmware/vic-repo/build-deps/installer-onsite-beta#files.
- You are running a Web server that the vCenter Server Appliance can access.

Procedure

1. Unpack the `vic-ui-installer.zip` bundle on any system.
2. Upload the plug-in bundle to your Web server.

   ```
   unpack_dir\installer\vsphere-client-serenity\com.vmware.vicui.Vicui-0.0.1.zip
   ```

3. (Optional) Connect to the vCenter Server Appliance via SSH and copy the contents of the `vic-ui-installer.zip` bundle to a temporary location on the vCenter Server Appliance.

   Perform this step if you unpacked the `vic-ui-installer.zip` bundle on a Windows system. If you unpacked `vic-ui-installer.zip` on a Mac OS or Linux system, you do not need to copy the files to the vCenter Server Appliance.

4. Open the `unpack_dir\installer\VCSA\configs` file in a text editor.
5. Enter the IPv4 address or FQDN of the vCenter Server instance on which to install the plug-in.

   ```
   VCENTER_IP="vcenter_server_address"
   ```

6. Enter the URL of the ZIP file on your Web server.

   ```
   VIC_UI_HOST_URL="vic_web_server_location"
   ```

7. (Optional) If you used an HTTPS address in `vic_ui_host_url`, provide the SHA-1 thumbprint of the Web server.

   ```
   VIC_UI_HOST_THUMBPRINT="thumbprint"
   ```

8. Save and close the `configs` file.
9. Navigate to `unpack_dir/installer/VCSA`.
10. Run the installer.

    ```
    ./install.sh
    ```

    Make sure that `install.sh` is executable by running `chmod` before you run it.
11. Enter the password for the vCenter Server administrator account.
12. When installation finishes, if you are logged into the vSphere Web Client, log out then log back in again.

Install the vSphere Integrated Containers Plug-In on a vCenter Server Appliance Without Access to a Web Server

If you are running the vCenter Server Appliance and you do not have access to a Web server, you can manually install the vSphere Web Client plug-in for vSphere Integrated Containers.

Prerequisites

- You deployed at least one virtual container host to a vCenter Server instance.
- Download the latest build of the vSphere Integrated Containers plug-in bundle from https://bintray.com/vmware/vic-repo/build-deps/installer-onsite-beta#files.

Procedure

1. Unpack the `vic-ui-installer.zip` bundle on any system.
2. (Optional) Connect to the vCenter Server Appliance via SSH and copy the contents of the `vic-ui-installer.zip` bundle to a temporary location on the vCenter Server Appliance.

   Perform this step if you unpacked the `vic-ui-installer.zip` bundle on a Windows system. If you unpacked `vic-ui-installer.zip` on a Mac OS or Linux system, you do not need to copy the files to the vCenter Server Appliance.
3. Open the `unpack_dir\installer\VCSA\configs` file in a text editor.
4. Enter the IPv4 address or FQDN of the vCenter Server instance on which to install the plug-in.

   ```
   VCENTER_IP="vcenter_server_address"
   ```
5. Save and close the `configs` file.
6. Navigate to `unpack_dir/installer/VCSA`.
7. Run the installer.

   ```
   ./install.sh
   ```

   Make sure that `install.sh` is executable by running `chmod` before you run it.
8. Enter the password for the vCenter Server administrator account.
9. Enter the root password for the vCenter Server Appliance twice.

   The installer requires the root password of the vCenter Server Appliance twice: once to copy the files to the appliance over SSH, and once to set the correct ownership on the files and folders.
10. When installation finishes, if you are logged into the vSphere Web Client, log out then log back in again.